- 欢迎来到 The Carpentries Etherpad！

This pad is synchronized as you type, so that everyone viewing this page sees the same text. This allows you to collaborate seamlessly on documents.

Use of this service is restricted to members of The Carpentries community; this is not for general purpose use (for that, try https://etherpad.wikimedia.org).

Users are expected to follow our code of conduct: https://docs.carpentries.org/topic_folders/policies/code-of-conduct.html

---

Welcome to this Software Carpentry Workshop!

- Workshop site: https://sebranchett.github.io/2022-06-14-vu-online/
- Automating Tasks with the Unix Shell: https://sebranchett.github.io/shell-novice/
- Plotting and Programming in Python: https://swcarpentry.github.io/python-novice-gapminder
- Version Control with Git: https://southampton-rsg.github.io/2020-05-21-southampton-swc/novice/git/index.html
    - Link for clone (we will do this together): https://github.com/JuliaScheel/version-control-novice
    - Tutorial for personal token: https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token
    - Git Cheat Sheet: https://education.github.com/git-cheat-sheet-education.pdf

If you have not completed pre-survey, please take a moment if we haven't started yet! https://carpentries.typeform.com/to/wi32rS?slug=2022-06-14-vu-online&typeform-source=sebranchett.github.io

Other information about this session:

Zoom Link: https://vu-live.zoom.us/j/97233043852?pwd=VnRhSGJPMndXVmp4WmU4b1ZrbkZuZz09

Day 4

Please sign in: Name (pronouns)

- Julia Scheel
- Stephanie van de Sandt

- Matthias, (helper, windows)
- Meron Vermaas
- Koen (helper)
- Lena Karvovskaya (helper)
- Sophie Arnoult (helper)
- Erika
- Wei Li
- Jiaxian Wu
- Yuwei Ding
- Susan Branchett, TU Delft (helper)
- Peggy van Minkelen
- Lynn Bouwknegt
- Yiyuan Li
- Irfan Setiawan
- Ting Liu, VU
- Vien Ho
- Lei Chen,VU
- Yuzhi Yang
- Egle Karmaziene
- Alex van der Jagt (he/him)
- eduardo dias
- Francisco Dray
- Juan Du
- Grace Chang-Byrne, VU
- Lanping Tang (vu)
- Shuyi Li, VU
- vicky vanthf

Great things today

Nobody is left behind if they have a problem.
Really nice to see how branching can be used.
I have tried 2 introduction to git courses online & was always confused as there was too much theory & the demonstration was too fast - git environment itself is not friendly for someone used to graphical interface interactions with machine. I thought the course was well designed to give me confidence in navigating the git environment after I knew what everything meant.

Next time please..

The pace was a bit slow this time.
Show how to resolve a conflict please.
Check visual (font) as some of the symbols `~` may not seem clear when presented (it looked a bit grainy & looked more like `--`)
Can prepare more concrete examples to input instead of saying that we can put anything in an environment, as concrete activities (this could be a real function!) enhances confidence in the purpose of learning this skill. However for the git instruction I can see that this was slightly less necessary than for python lab.

Speed check, p2lease vote by writing + (plus) sign below

go-faster: ++
go-slower:
okay:

Font size check, please vote by writing + (plus) sign below
enlarge: +
okay:

Script:
cd Documents
cd Coding
git clone https://github.com/JuliaScheel/version-control-novice

###configure git (setting your system user name, email, & editing environment)
git config --global user.name "Julia Scheel"
git config --global user.email "julischeel@gmail.com"
git config --global core.editor "nano -w"
git config --list #check on - list configuration settings

###move to folder to track
cd version-control-novice
###initiate git
ls #list files
git init
ls
ls -a #list hidden files also

####check status
git status
git log #check commits and history

make changes to climate_analysis.py
nano climate_analysis.py. #open file in nano editor
*make chnage in nano
*press cntrl-x to initiate exit; press "y" to save; & then `enter` to accept file name & to exit nano editor
git status
git add climate_analysis.py #stages the change (adds to tree branch), but not yet committed!
git status
git commit -m "added title"
git status

###make another change to climate_analysis.py
nano climate_analysis.py
*make change
git diff #to see what was changed in file we just accessed
git add climate_analysis.py
git commit -m "add rainfall processing placeholder" (after saving a change in nano, you must save in git. here you also add comment on the change)
git status
check history

```
git log  #see all changes made so far
git diff HEAD~1 climate_analysis.py #shows differences of current version to the last version
git diff HEAD~2 climate_analysis.py #shows differences of current version to the second last version
git diff <commit ID> #shows differences of current version to this speccific version


##### remove and bring back stuff
rm climate_analysis.py #removes this file
ls
git status
git checkout climate_analysis.py #brings this file back
ls
cat climate_analysis.py

git log climate_analysis.py
git checkout HEAD climate_analysis.py

git log
git diff <paste commit id like a4wdht.....>

git log
git diff <paste only the first few characters of commit id>

ls
rm climate_analysis.py
ls

git status
git checkout climate_analysis.py
ls

cat climate_analysis.py

git log
git log climate_analysis.py

<Move to GitHub account in a browser>
Create a new repository
Repository name: version-control-novices
Description: this is the carpetries version-control repository
Keep it Public
Don't change anything else yet...
Click 'Create Repository'

follow instructions for:
...or push an existing repository from the command line
got to git/terminal and paste command

git remote remove origin
```

... and try paste again:
git remote add origin https:://github.com/<your git name>/version-control-novices.git

git remote --verbos # to see how this repository is connected to GitHub
git status
git push origin main

Follow this tutorial:
https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token

In GitHub, go to settings
On left, click on 'Developer settings'
On left, click on 'Personal access tokens'
Click on 'Generate new token'
Give it a name
Select 'repo' and 'admin-repo-hook'
'Generate token'
'Save'

In git/terminal...
git push origin main

You will be asked for:
1.) username: GitHub username
2.) E-Mail: GitHub e-mail
3.) username: GitHub username
4.) Token

==BRANCHING==

In terminal:
git branch
git branch dev
to create a branch called 'dev'
git branch
git checkout dev
git branch

nano rainfall_conversion.py
to create a new file
add some text, a comment for example
^X
y
<Enter>
to save it
git status
ls
you can now see the new file in the list

```
git add rainfall_conversion.py
git commit -m "added rainfall_conversion.py"
git status
git log

# go back to main branch
git checkout main
ls
# new file is not in this main branch, only in dev branch

git checkout dev
git push origin dev
# this pushes the dev branch to GitHub

# we want to merge the branch `dev` with the `main` in GitHub
# move to GitHub
Select your repository 'version-control-novices'
click on 'Compare and pull request' button
always good to add a descriptive comment
Click 'Create pull request' button
Click 'Merge pull request' button
Click 'Confirm merge' button

# go back to git/terminal
git branch
git checkout main
ls
git pull origin main
# this pulls changes from GitHub to local computer
ls  #here you should be able to see you local main has been changed to reflect the merge in Github (with 3 files within)

Testing 'git commit' without inline comment
Use nano to make a dummy change
git add newfile.py
git commit
# automatically takes you to text editor to make extended comment

Excercise walk through:
git branch
git checkout dev
git branch doc
git checkout doc
nano README.md
# add text with title, what's in the repository, etc.
^X
y
<Enter>
```

```
git branch  # 3 branches with '* doc' to indicate we are in doc
git status

git add README.md
git commit -m "added a README file"

git push origin doc

# back in GitHub and code
Compare and pull request
change so merges with dev instead of main (base:)
Create pull request
Check the Files changed
Back to Conversation
Merge pull request
Confirm merge

# back to git/terminal
git branch
git checkout dev
ls
git pull origin dev
ls
```

Day 4 Questions

Q: What does this line do "git commit -m "add rainfall processing placeholder””

- its a commit with a changelog message, you commit the changes to your git (commit), the -m creates a label that tells you what you've changed exactly. You can add whatever you want, its for you and helps to see the difference between the two versions

Q: How to set colours in Mac Terminal?

- In the Terminal menu, select:
    - Edit > Show colours (Shift-Cmd-C)
- colour profiles can be chosen from a list, in the Menu:
    - Terminal > Preferences > Profiles
- The terminal should also allow for colours:
    - Terminal > Preferences > Profiles > Advanced: "Declare terminal as: xterm-256color"

Q: I would still like to learn how to make a side-along script of the commands like what Susan did on day 1?

Q: any good tips (or links to good tips) on setting up the environment for easy work? & how to check historical system changes? or best practices to log system changes that melds well with workflow (while programming) that can be easily referred to by self later on? (minimal environment switching, ease of search, naming conventions...etc).

Day 3

Please sign in: Name (pronouns)

- Stephanie van de Sandt
- Matthias Mimault, (Instructor, windows) The James Hutton Institute, Scotland
- Julia Scheel (she/her)
- Elisa Rodenburg (she/her). Helper, VU Amsterdam
- Tijmen Münker (Helper)
- Meron Vermaas
- Erika
- Yuwei Ding
- Ting Liu, VU
- eduardo dias
- Irfan Setiawan,VU
- Wei Li
- Lynn Bouwknegt
- Yuzhi Yang, VU
- Vien Ho
- Jiaxian Wu, VU
- Peggy van Minkelen
- Lanping Tang (vu)
- Yiyuan Li
- Juan Du
- Vicky
- Francisco Dray
- Koen Leuveld (helper)
- Egle Karmaziene
- Alex van der Jagt (he/him)
- Alessandra Gomes
- Shuyi Li
- Grace Chang-Byrne (VU)

Great things today
The current mode is great, one teacher and several helpers. Hopefully there will be more this kind of software Carpentry Workshop! Thank you all for this amazing workshop!
The 'demo-mode' was nice. I guess I will need some more time to practise on my own
Great python instructor Matthias!
I liked how Matthias had evidently "over practiced" so he could have a smooth flow during the instruction. It's a good tip!

Next time please..
Sometimes typing at the bottom of the screen in jupyter lab is hard to see (zoom bar covers the bottom).

Font size check, please vote by writing + (plus) sign below
enlarge: ++
okay: +

Speed check, please vote by writing + (plus) sign below
go-faster:
go-slower: +
okay: ++

Day 3 Questions

Q: Can Python be used for more than data analysis, for example, data mining?

- Yes, it is very widely applicable and data scarping/machine learning and whatever you want is possible. The sky is the limit!
- The Computational Linguistics and Text Mining Lab at the VU has a course on Python for text analysis: https://github.com/cltl/python-for-text-analysis

Q:

FYI: The Computational Linguistics and Text Mining Lab at the VU has a course on Python for text analysis: https://github.com/cltl/python-for-text-analysis

Q:to access the 0 in the list within the list, would you still index the same? In the example: another_list = [1, 2, 3, 4, [0, 4]] where there is a list in a list

- You first need to access the list in the list so: another_list [4]
- This returns [0,4], so to get only the zero, we need to access the first element of the second list: print(another_list [4][0])

Q: Can you append multiple values?

- sequentually, yes

Q: What is the difference between "append"and "extend"?

- append adds one value, extend adds a list
- Both are the same speed
- + operator also works (you have to assign it to primes of course), but extending/appending is faster if you execute the command many times in your script.

Q: Can we omit the 0 at the end? Does typing 3 instead of 3.0 make any difference?

- It's the difference between a float and an integer. While python automatically adjusts between integers and floats, it's good practice to be conscious about the difference between the two.
- In older versions this adjusting between float and integer was not automatic. Additionally, in some cases you will want to sepcifically assign the type of variable (so there are no rounding errors).

Also, some progarmming languages require you to define what type of variable you have.

Q: There is an red 'out[56]' in my output after I run the code in line[56], what does it mean? I just run the 'pd.read_csv' code as the same as in the course.

- The output [out]linenumber is colored red, not because of an error, but to differentiate from the blue [] input. It's just color coding to help!

Q: sometimes my cell does not show color syntax and it does not run. then I make a new cell and it works. do you know what is happening?

Q: Are there mathematical functions built into Python? Like calculating the mean. Is it ok if the name of our defined function is the same as that of the existing function?

- yes there are quite a number of 'standard' functions, and if you create your own function with the same name, it will overwrite that standard function

Q:does function generally use 'return' instead of 'print'?

- it depends on what you want to do, if it's just something to show you what is going on, you can use print. If you want to use a value outside of that function, you should use return

Q: will the system notify you if you overwrite a standard function?"

- No, you have to be careful. You could overwrite any function.
- Also, when using modules, they could partially overwrite each other if you are not careful

Q: Can you use "print" in a function?

- yes, e.g.: def print_date(year, month, day):

FYI: https://peps.python.org/pep-0008/ (guideline how to style your code, e.g. how to use capitalization and improve the readability and consistency)

Questions concerning day 2:
Q: I am not sure if I understand a previous section right. What we did is to get only the numerical years. However, its not enough only do "years=data.columns.str.strip('gdpPercap_')" because the variable is still "object" rather than "integer"? So we do "data.columns=years.astype('int')"?

- Indeed, the strip() method remove some characters from a string and return a string, non an integer. This is why we use then the method astype to convert a list of strings (several strings) into a list of integers

Day 2

Please sign in: Name (pronouns)

- Matthias Mimault, (Instructor, windows) The James Hutton Institute, Scotland
- Stephanie
- Meron
- Elisa (she/her)
- Francisco Dray
- Le Yu
- Wei Li
- Peggy
- Yuwei Ding
- Lynn
- Vien Ho
- Irfan Setiawan (VU)
- Erika
- Yuzhi Yang
- Jiaxian Wu
- Vicky Vanthof
- Grace Chang-Byrne (VU)
- eduardodias
- Ting Liu, VU
- Yiyuan Li
- Juan Du
- Lanping Tang, VU
- Egle Karmaziene, VU
- Alex van der Jagt (he/him)
- Alessandra Gomes
- Julia Scheel
- Shuyi Li

Day 2 Questions:
Speed check, please v ote by writing + (plus) sign below
go-faster:
go-slower:
okay: ++

Q: Do I need to be online to work with Jupyter notebooks?

- no, you work locally, but you can be online, depends what you're doing and the requirements

Q: Where is the data?

- https://swcarpentry.github.io/python-novice-gapminder/files/python-novice-gapminder-data.zip

Q: How to move to anoher cell?

- shift + ENTER -> execute and go the next one
- deselect: with the escape key, and then use the arrow keys to move between cells
- new cell above: press a

- new cell below: press b

Q: Does the UI element (|> button) do the same as shift-enter?

- yes

Q: What is the kernel?

- Kernel is sth like the shell, interface of cells

Q: While we're working on print. I noticed that there was an automatic space between age and “years old" . Does the print function not forcibly concatenate?

- if you mean the space followed after the comma that's actually part of the print statement's syntax and won't influence the string
- additionally, you can influence this, using sep='*separator*' as an additional input to print, for example: print('test','example',sep='') removes the space

Tip for everyone: under Settings, you can tick "Auto Close Brackets", which will automatically close parentheses and quotes. It makes your life easier :)

Q: Why do we get the first four letters if we refer to 0-3

- the array counting starts at 0 in Python. The first position is 0, the second is 1 and so on

Q: both single and double quotes work as long as they come in pairs?

- Yes, but for now use single quotes

Q: I am not clear that when need to use quotes, and when do not need to? for example, len() in cell 9 and cell10

- A variable containing text, so=called strings requires that you use quotes. The variable name itself should not have quotes: you use quotes for strings, that means words like 'helium', but atom_name is a variable name

Q: How to print helium AND sodium?

- print several variables, better not use the same variable name
- can I use a vector? You could, but we'll talk about that later.

Q: What are the round brackets and square brackets for? When to use what?

- () = arguments of functions, processsing values
- [] = access brackets, they try to access the value inside the list

Q: How does the auto-completion work? Historcial? Does it refer to the overall notebook?

- its in the same environment, same session. not like shell, no memory of previous session. You need to execute everything before.

Q: What was the command for auto completion?

- You can press the Tab key on your keyboard
- You need to start typing the beginning of the variable before you press tab (at least one letter is needed, then all options are shown)

Q: How can we convert a string into a number / integer?

- int('2') string to number
- str(2) number to string
- note that int('letters') does not make sense, and also does not work

To take notes in the notebook: use # to add comments

Q: I tried to add "type" on the result of line 30 (print(str(1)+'2')), but this gave a NoneType response. What does this mean?

- The type of print() is not a type, it's the print output
- type of the contents in the print would give you the desired information
- use print(type(str(1)+'2')) -> <class 'str'>

Use this cheat sheet for Markdown: https://www.markdownguide.org/cheat-sheet/

Q: What is the difference between markdown and the " / # marks?

- markdown change the language, differnet type of writing text (tell the interpreter that its NOT python)
- # inserts comments in a code cell, in markdown it is a title

Q: How to make a list in markdown?

- use - before what you want as bullet point

Q: What is the 'raw' option under 'code' & 'markdown'?

- simple text, will not interpreted as markdown or text

Q: Why does swapcase not require the double underscores?

- underscores are a convention of developers, its a convention of name, its how the developers settled on how to use swapcase
- : import function

Q: What does the dot after a variable stand for?

- the dot functions as a separator after which all internal function options can be accessed
- for example:
    - nunicorn_age = 99.74731
    - print(nunicorn_age.__round__(2))

In your notebook, defining a variable will not output it. Only by printing it using print() will it be shown on your screen.

A few shortcuts:
The b key will make a new cell below the currently selected cell.
The a key will make one above.
The x key will delete the current cell.
The z key will undo your last cell operation (which could be a deletion, creation, etc).
The y key will turn into Python mode
The m key will turn into Markdown mode
The d key twice will also delete the current cell

Q: I wonder the "JupyterLab" only can be used on an online website? I did find this when I open the Anaconda, while only see "Jupyter Notebook".

- It can be used offline and online. But in our examples, everything is local and all just happens on your own computer, nobody can reach it. In the url it says "localhost", that means it's your own private network!
- Q so you can use jupyter lab on a chromebook that has only has a browser?
- Yes

Q: How long will a library be "remembered" in the notebook?

- As long as the session takes
- You can import as many libraries as you want (they can be overwritten if they have the same name though)

Q: Is there a way to see all the libraries that have been imported into a session or general python environment, & /or aliases that have been created?

- Within the library numpy: import numpy as np import sys import os modules = dir() print(modules)

Use Google in case of questions. Stackoverflow may be a good source to get help! (e.g. how to know loaded libraries in Python)

Q: Is bases[..] in cell 79 a function? (bases[random.randrange(len(bases))])

- its a variable with a string

Q: How to know use "()" or not. For example, "data.columns" not includes "()" while "data.describe()" includes "()"?

- describe is a function, thus you have to use the ()

If you call data, your notebook should be in the same folder as your notebook. Otherwise be more precise when you call data and indicate the location of the data (e.g. desktop/data/file.csv)

Q: Why is it .columns and not .columns()? Does that mean it just never takes any parameters?

- it's essentially not a function so it doesn't take parameters. it's a field of the dataframe telling us what the columns are

Q: How to call help for a specific function? using a key combination

- shift + tab

Q: What is meaning of iloc and loc?

- iloc accesses the dataframe by position and loc by index (which could also be a string/name)
- two representations of the data, one is accessed by place, the other one by name

Q: how would we also get the country, for those min and max?

- print(data.index)

Day 1

Please sign in: Name (pronouns)

- e.g. Stephanie van de Sandt, UBVU
- Susan Branchett, TU Delft
- Meron
- Matthias, (helper) The James Hutton Institute, Scotland
- Sophie (Helper)
- Elisa, Vrije Universiteit Amsterdam
- Wei Li
- Erika
- Le Yu
- Lynn
- Ting Liu, VU
- Francisco
- Alex van der Jagt (he/him)
- Juan Du
- Lucie
- Bei Wu
- eduardo dias
- Jiaxian Wu
- Vicky Vanthof
- Yuwei Ding
- Yahua Zi, Vrije Universiteit Amsterdam
- Yuzhi Yang
- Alessandra Gomes
- Grace Chang-Byrne (VU)
- Irfan Setiawan
- Lei Chen VU

- Yiyuan Li, Vrije Universiteit Amsterdam
- Lanping Tang (VU)
- Vien Ho (I put my name in the wrong part on the first day)
- Egle Karmaziene, VU
- Peggy
- Shuyi Li

Day 1 - Questions

Link to course material: https://swcarpentry.github.io/shell-novice/data/shell-lesson-data.zip (for people with multiple devices)
Are these commands only for bash environment or generic?

- Generic, but for bash shell

Bash in English?

- Bourne-again shell) is the command. Simply put, it's a text-interpreted command. It was written by Command Brian Fox for the project, an alternative to Bash (your UNIX shell) Bourne shell.

Can we search inside the terminal?

- Yes! This will be introduced tonight :D

The difference between ls and dir:

- git bash -> ls
- dir for Windows laptops

What does -F mean?
The long option name is --classify.... I think they just ran out of letters :)

I can't see it, there doesn't seem to be such a file or directory, the command ls -F Desktop can't access the Desktop

Q: After the command "ls -F Desktop", some files in Chinese names do not show correctly. May I ask how to correct it (like how to set "UTF-8" in the shell?)

- what does typing "locale" give you? it will become quite technical to fix this one way or another though. To configure a specific locale parameter, edit the appropriate variable. E.g.: sudo update-locale LC_TIME=en_IN.UTF-8

If my file is in the /D, but now I'm in the /c/user, how can I turn to /D?

- can you use cd or cd ~ to go to your home folder? from there you can navigate to /D. Let me know if it works@
- you can also do: cd /D , IF /D starts with a slash or ~

would it make sense to maybe share the exercises for the day the day before so we can read it first? So we already have something to mentally "search for" when in lesson.

Can the instructor when in flow take a moment to pause between new concepts - so we can take a moment to try it out. I noticed I missed some content when exploring the command.

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?

1. cd .
2. cd /
3. cd /home/amanda
4. cd ../..
5. cd ~
6. cd home
7. cd ~/data/..
8. cd
9. cd ..

Excercise 2

Using the filesystem diagram below, if pwd displays /Users/thing, what will
‘ls -F ../backup’ display?
1. ../backup: No such file or directory
2. 2012-12-01 2013-01-08 2013-01-27
3. 2012-12-01/ 2013-01-08/ 2013-01-27/
4. original/ pnas_final/ pnas_sub/
https://swcarpentry.github.io/shell-novice/02-filedir/index.html#relative-path-resolution

Key points

- The file system is responsible for managing information on the disk.
- Information is stored in files, which are stored in directories (folders).
- Directories can also store other directories, which then form a directory tree.
- pwd prints the user’s current working directory.
- ls [path] prints a listing of a specific file or directory; ls on its own lists the current working directory.
- cd [path] changes the current working directory.
- Most commands take options that begin with a single -.
- Directory names in a path are separated with / on Unix, but \ on Windows.
- / on its own is the root directory of the whole file system.
- An absolute path specifies a location from the root of the file system.
- A relative path specifies a location starting from the current location.
- . on its own means ‘the current directory’; .. means ‘the directory above the current one’.

Moving Files

•This is the current situation:
$ ls -F
analyzed/ raw/
$ ls -F analyzed
fructose.dat glucose.dat maltose.dat sucrose.dat
$ cd analyzed

•sucrose.dat and maltose.dat are in the ‘analyzed’ folder, but should be in the ‘raw’ folder. What is the command to correct the situation:

mv sucrose.dat maltose.dat ____/____

Excercise: Removing safely
What happens when we execute rm -i thesis_backup/quotations.txt? Why would we want this protection when using rm?

mv sucrose.dat maltose.dat ____/____

Excercise: Matching a pattern
•When run in the proteins directory, which command(s) will produce this output?
ethane.pdb methane.pdb

1.ls *t*ane.pdb
2.ls *t?ne.*
3.ls *t??ne.pdb
4.ls ethane.*
Think about it first, then try it, then discuss…

Excercise: Reproduce folder structure

•You’re starting a new experiment and would like to duplicate the directory structure from your previous experiment for consistency. Your new directory structure looks like this:
2022-06-14/
└── data
  ├── processed
  └── rawot
•Which set of commands can be used to achieve this?
•Look up the meaning of ‘mkdir –p’ and repeat the exercise

Key points

- cp [old] [new] copies a file.
- mkdir [path] creates a new directory.
- mv [old] [new] moves (renames) a file or directory.
- rm [path] removes (deletes) a file.

- matches zero or more characters in a filename, so *.txt matches all files ending in .txt.
- ? matches any single character in a filename, so ?.txt matches a.txt but not any.txt.
- Use of the Control key may be described in many ways, including Ctrl-X, Control-X, and ^X.
- The shell does not have a trash bin: once something is deleted, it's really gone.
- Most files' names are something.extension. The extension isn't required, and doesn't guarantee anything, but is normally used to indicate the type of data in the file.
- Depending on the type of work you do, you may need a more powerful text editor than Nano.

About wc:

- wc -m filename -> counts characters
- wc -c filename -> counts bytes

This will generally only give the same result for ASCII encoded text!

More on wc without input:

- in this case it reads the standard input stream, so for example you could type something and then Ctrl+d will execute the wc

Excercise Pip
•A file called animals.csv (in the shell-lesson-data/exercise-data/animal-counts folder) contains the following data:
2012-11-05,deer,5
2012-11-05,rabbit,22
2012-11-05,raccoon,7
2012-11-06,rabbit,19
2012-11-06,deer,2
2012-11-06,fox,4
…
•What text passes through each of the pipes and the final redirect in the pipeline below? Note, the sort -r command sorts in reverse order.
cat animals.csv | head -n 5 | tail -n 3 | sort -r > final.txt
Hint: build the pipeline up one command at a time to test your understanding

Key Points

- wc counts lines, words, and characters in its inputs.
- cat displays the contents of its inputs.
- sort sorts its inputs.
- head displays the first 10 lines of its input.
- tail displays the last 10 lines of its input.
- command > [file] redirects a command's output to a file (overwriting any existing content).
- command >> [file] appends a command's output to a file.
- [first] | [second] is a pipeline: the output of the first command is used as the input to the second.
- The best way to use the shell is to use pipes to combine simple single-purpose programs (filters).

Key Points

- Save commands in files (usually called shell scripts) for re-use.
- bash [filename] runs the commands saved in a file.
- $@ refers to all of a shell script's command-line arguments.
- $1, $2, etc., refer to the first command-line argument, the second command-line argument, etc.
- Place variables in quotes if the values might have spaces in them.
- Letting users decide what files to process is more flexible and more consistent with built-in Unix commands.

Great things today

Breakout room exercises

Strong and comprehensive introduction to navaigating with commandline. Loops and scripts could have gotten more attention.

Helpers

+ helpers in breakout rooms

was a good pace. (but I might be slower than average). I have done things in terminal blindly following instructions & was always kind of tentative so was really useful to get the scope of what these commands actually do - & the warnings!

Next time please...

- it was too slow in the begining

Breakout rooms were not really engaging.

Check if everybody is on the same page more often inbetween. Because when you aren't, you cannot type along

Screensharing in breakout room didn't work

would like a moment to practice / think individually before starting group discussion.

update 20220623:

- it's possible to annotate command line if using the terminal environment **iterm**

    - ref: https://iterm2.com/features.html
    - can also toggle time stamp in there