Learning Journal for FOAR705

Week 1:

Objective: Restore a file from a 6 month old back up
Action: Search on Google Drive for “Knowledge of Self and Others” the Mind and World essay I wrote over a year ago
Error: None
Result: Success, document found and opened

Objective: Look at project management tools and form opinions
Action: Observe ‘project management’ I am currently doing for my wedding on shareable google sheets. Although the way it is formatted is useful for me to understand, it can be hard to communicate what I mean to other people. I briefly looked into Trello, Jira and Asana. The visual nature of Asana seems much clearer as a form of communication.
Error: None
Result: Asana definitely seems like a more useful project management tool than what I currently use.

Week 2:

Overall Objective: Figure out what LaTex is and how to use it.

Objective: Use LaTex
Action:

1. Create an overleaf Account
2. Create a “Test LaTex” file

Error: None
Result: Success

Objective: Play around with LaTex
Action:

1. Change “\author” to “Georgia Rutherford”.
2. Delete \maketitle to see what happens

Error: Need to hit “recompile” for the example document to update.
Result: Apparently deleting \maketitle deletes the text that was above it in the document. I am unsure why.

Objective: See what happens if I put text above the \begin document
Action: Type \section{Hello World} above \begin document
Error: Using “/” instead of “\” makes it plain text instead of a section.
Using “()” instead of “{}” does not work (not entirely sure why but it looked weird).

Result: It made a page with that section before the other pages. This included the title even though the text for the title was above the “\section{Hello World}. This must be because of the “\maketitle” command

Objective: Make paragraphs
Action:

1. Press enter twice to change line
2. Write “\paragraph{}” on a new line and then continue text on the line below to add a larger break.

Error: Need to do two breaks to change the line of the text (otherwise the text is written next to the other text).
Doing three breaks doesn’t change the line further (still just turns up underneath).
Writing “\paragraph” changes the text inside the “{}” into bold.
I don’t like the indentations on the document, not sure how to change
Result: Not entirely sure if above is the proper way to make paragraphs, but it seems to work. Need to research further.

Overall Result: I have a much better idea on how to use LaTex and the process has been much more simple than I anticipated. However, there is still a lot I’m confused about. For example, I am not extremely fond of the formatting and would like to learn more about how to change that. At this stage I probably know enough to be able to complete the first assignment.

Overall Objective: Do Scoping exercise in LaTex

Objective: Create a LaTex document on overleaf
Actions:

1. Open a blank text
2. Name it Scoping Exercise
3. Change “\author” from my email address to “Georgia Rutherford”
4. Recompile

Error: none
Result: Successfully made document

Objective: Create a list of headings and subheadings on my document
Actions:

1. Write “\section{}” for A Day in the Life, Pains, and Gains
2. Write “\subsection{}” for Pains I Encounter, Pain Relievers, Gains I Would Like, and Gain Creators

3. Recompile

Error: realised after that I can use the “rich text” to put headings and subheadings in even quicker.
Result: Headings and subheadings created

Objective: Enter relevant information under each heading
Action: Entered in text
Error: I realised partway through that I can use “Rich Text” to very easily add bullet points and some formatting.
Tried to put in quotation marks and one of them came out backwards. Need to use ‘ ’ and ` ` keys instead of “” in LaTeX
Result: Finished Exercise

Overall Result: Finished Scoping Exercise

Overall Objective: Complete Data Carpentry exercises

Objective: Identify what is wrong with this spreadsheet/the steps you would need to take to clean up the two tabs and put them together
Things that are wrong:

- Graph labeled Plots uses yes, y, no and N
- Uses a key that a computer won’t be able to translate (yellow in first tab, asterisk in second)
- Some of the cells are left blank
- Page 2 title is not in the graph
- Asterisk used twice in the second tab (could be confusing as to which they refer to)
- Second tab has a mix of numbers and words in graph (1, yes, no, yes)
- Spelling errors (errth) and different ways of spelling (Mabati_sloping vs mabatisloping)
- Data errors (-99 rooms)
- Livestock owned and numbers should be separate

Steps needed to take to put them together:

- You would need to change so they are using all the same values (example: all yes/no or all numbers for livestock)
- You would need to correct spelling so there are no errors and everything matches
- Fix data errors
- Make changes on a separate document

Objective: make a list of some of the types of metadata that should be recorded about this dataset

- What counts as an item for items_owned
- Who was interviewed and how they were chosen
- What does no_meals mean (number of meals? Per day?)
- Wall type on what? (House?)
- What does no_members mean? (number of members? of household?)
- Location of villages
- Which year were the months_lack_food in?

Overall Results: I have read to the end of formatting problems. I was surprised to see that what I thought was a data error, the use of numbers like -99, was actually meant to convey a null value. This highlighted to me the issue of being incredibly clear with your data. I was also interested to read about how merging cells can be problematic for the computer seeing associations between data. Out of all the problems listed, merging cells is probably the one I would be most likely to commit.

Overall Objective: Submit LaTex Scoping Exercise

Objective: Fix an error I just found in the LaTex code “overfull \hbox”
Action:

1. Google what “overfull \hbox” means and how to fix it
2. Add \linebreak to move the word “Throughout” so it isn’t split in two

Error: Initially tried to just move throughout to the line below but it caused an indentation. Using \linebreak moves the word down without causing the indentation.
Result: The error was just highlighting that the word “Throughout” was being split in two. This didn’t really bother me but it was cool to learn how I can add a line break without having the indentation. Having tested it further, I can add \linebreak twice and start a new paragraph without the indentation.
However, having thought about it a bit more I realise that \linebreak will add the line break there even if my formatting changes and there is no longer a need for the line break. I am unsure as to how to fix the issue of “Throughout” being split without causing this second issue.
At this stage I’m going to leave “Throughout” as it is and look more into how to fix this “overfull \hbox” issue for the future.

Objective: Download .tex file
Action:

1. Download PDF
2. Google how to download .tex file from overleaf
3. Download .tex file
4. Unzip file

Error: I was lost for a little bit about where to find the button to download the file as .tex since it wasn’t near the PDF download button. Turns out it was in the menu bar under ‘Source’

Result: Both PDF and .tex files downloaded

Objective: Submit Exercises
Action:

1. Rename from “main” to “GeorgiaRutherford-ScopingExercise”
2. Submit .tex on cloudstor
3. Submit PDF on ilearn

Result: Submitted

Overall Result: Submitted LaTex Scoping exercise

Overall Objective: Data Problems in philosophy

One problem with the use of data in my field is that sometimes data is ignored. It is common among philosophers writing on the topic of work to write about 'the end of work' in society as though it is inevitable. However, there are statistics that show employment is actually very stable at the moment. This issue is discussed in 'The Return of Work in Critical Theory' on page 36 by Christophe Dejours, Jean-Philippe Deranty, Emmanuel Renault and Nicholas Smith.

The second issue of data use in my field also comes from the topic of work. A large amount of data exists around people's experiences of work and some of this data is contradictory or can be interpreted in different ways. This issue is discussed by Marie Jahoda in 'Work, Employment, and Unemployment'.

Overall Result: success

Week 3

Overall Objective: Complete Data Carpentry activities for this week
Objective: Extract the components of the date to new columns
Action:

1. Title columns Day, Month, Year
2. In B2 write Day=($A2)
3. Drag from the corner of B2 to the bottom of the column
4. In C2 write Month=($A2)
5. Drag from the corner of C2 to the bottom of the column
6. In D2 write Day=($A2)
7. Drag from the corner of D2 to the bottom of the column

Result: Success. I was previously unaware of the Day=, Month=, and Year= functions.
Objective: add another data point in the interview_date column by typing 17/11
Action:

1. Write 17/11
2. Continue the day/month/year columns to include the new entry

Result: Success, A16 now displays 17/11/2019. I was also previously unaware excel auto-completed dates like this.

Objective: Apply a new data validation rule to one of the other numeric columns
Action

1. Click on column G (rooms)
2. Click Data
3. Click Data Validation
4. Click allow - whole number
5. Restrict to between 1-30
6. Change input message. Title: Invalid number. Input message: Must be whole number between 1-30

Result: Success

Objective: Apply a new data validation rule to one of the other categorical columns
Action:

1. Choose column B: Village
2. Click Data
3. Click data validation
4. Click allow - list
5. In source type God, Rauca, Chirodzo
6. Change input message. Title: Invalid Village. Input message: Only God, Rauca and Chirodzo accepted

Objective: Export data
Action:

1. Click File
2. Click save as
3. Change format to .csv

Error: For the SAFI_dates file an error popped up saying I couldn't save both tabs. This was okay because I only used one of the tabs. Still good to note for future reference.
Error: when I reopen the document it doesn't appear to have saved the data validation I did, so I will try again.
Result: I think You need to save the document as an .xlsx file to save data validation, and then as .csv so you have access to the data if excel stops supporting the .xlsx files

Objective: Export the csv. View it in a text editor like Atom.io, Sublime Text, or notepad++ Think about the benefits of an always-readable and not tied to a subscription or specific program data format.
Action: I don't have those programs downloaded on my computer, but I opened the .csv in just normal notepad and it was still recognisable. The .xlsx file, on the other hand, was complete gibberish. Having a file not ties to any specific program would mean that you can access it if you

lose access to the program used, and that it is shareable with everyone no matter what program they use.

Overall Result: Data Carpentry Task Done

Overall Objective: Complete Scoping II: Computational Analysis in LaTex

Objective: Create document
Action:

1. Open Overleaf
2. Click new project
3. Click blank project
4. Title: Scoping II: Computational Analysis

Error: None
Result: objective complete

Objective: Create sections for Decomposition, Pattern Recognition, Algorithm Design
Action:

1. Write \section{...} for each topic
2. Click Recompile

Error: two of the topic headings did not appear as sections because without thinking I had capitalised the S is \Section
Solution: uncapitalise the S
Result: Success

Objective: Type information into each section
Action:

1. Add subsections for note organisation and referencing in each section
2. Write information

Error: None
Result: Success

Overall result: Scoping II complete

Overall Objective: Check out Bibtex

Objective: Create a bibtex file
Action:

1. Go to overleaf
2. Create a new project to test things in
3. Create a new .bib file

Error: None
Result: Success

Objective: Cite something
Action:

1. Type in test.bib: @book{carruthers2011opacity,
   title={The opacity of mind: An integrative theory of self-knowledge},
   author={Carruthers, Peter},
   year={2011},
   publisher={OUP Oxford}
   }

2. Type \cite{carruthers2011opacity} in main.tex file

Error: received error “You have cited something which is not included in your bibliography.”
Solution: I had to specify a .bib file and which bibliography style to use in the .tex file.
Action:

1. Type \bibliographystyle{plain}
2. Type \bibliography{test}

Result: Success. The bibliography appeared at the end of the document and the citation is in the main body of the text

Objective: Cite the page number as well
Action

1. Change \cite{carruthers2011opacity} to \cite[p.~31]{carruthers2011opacity}

Error: None
Result: Success. Citation now says [1, p. 31]

Objective: Change citation to a format more like my usual style.
Action:

1. Download an APA style bibtex style from https://www.reed.edu/cis/help/LaTeX/bibtexstyles.html
2. Upload file to project
3. Delete \usepackage[utf8]{inputenc}
4. Write \usepackage[natbib]
5. Change \bibliographystyle{plain} to \bibliographystyle{apa-good}

Error: Literally everything broke and it couldn’t compile. I clearly missed some steps. I’m going to take a break and come back tomorrow.

Objective: Start new bibtex test
Action

1. Open new project
2. Create .bib file
3. Put carruthers reference info into .bib file
4. Type \bibliographystyle{plain} in .tex file
5. Type \bibliography{test} in .tex file

6. Type \cite{carruthers2011opacity}

Error: Reference not found
Solution: Read my learning journal. Be confused because I did everything the same as last time. Realise that the .bib file is named Test.bib this time and that it is cap sensitive. Change to \bibliography{Test}
Result: Success

Objective: Change to APA style reference
Action:

1. Google how to use APA bibtex, click on http://homepage.stat.uiowa.edu/~rlenth/ALPHA/apa-tutorial.pdf
2. Write \usepackage[natbib]
3. Don't delete \usepackage[utf8]{inputenc} this time
4. Change \bibliographystyle{plain} to \bibliographystyle{apalike}

Result: Success in changing the format of the reference. Although it came up as Carruthers (2011) and I would prefer (Carruthers, 2011).

Overall Result: There is definitely some promise here and I think I will find bibtex very useful in the future, but I am still interested in the digitising handwritten notes idea. So for now I am going to leave this and look further into character recognition.

Overall Objective: Check out existing written to digitised text software
Objective: Download something to test
Action:

1. Google 'open source character recognition'
2. Find out about Google Tesseract from https://opensource.com/life/15/9/open-source-extract-text-images "The technology extracts text from images, scans of printed text, and even handwriting"
3. Click on https://opensource.google.com/projects/tesseract
4. Click on https://github.com/tesseract-ocr/tesseract
5. Download tesseract file
6. Unzip tesseract file

Error: Lots of files within the tesseract file. Will need to google how to actually use this software.
Objective: Download Tesseract and use it

Action:

1. Google how to install https://www.bl.uk/britishlibrary/~/media/bl/global/early%20indian%20printed%20books/training%20resources/installing%20and%20using%20tesseract%20ocr.pdf
2. Go to https://github.com/tesseract-ocr/tesseract/wiki
3. Then go to https://github.com/UB-Mannheim/tesseract/wiki
4. Download 64bit installer
5. Go to https://github.com/tesseract-ocr/tesseract/releases

6. Download zip of source code
7. Unzip Source code
8. Rename JPG file to test1
9. Put JPG file into tesseract-4.00.00alpha folder

Error: Can't find the folder. Probably because I downloaded the newer version. I will try put the JPG file into the Tesseract-OCR folder instead

10. Open Command prompt

Error: At this stage I got very confused and asked my IT trained fiance with help on how to use command prompt. .

11. Firstly I need to be in administrator to have permission to edit files in program files folder. (Right click on command prompt icon, right click command prompt, click run as administrator)
12. Next I need to change directory from windows/system32 to the 'working folder' (Type cd "C:\Program Files\Tesseract-OCR")

Error: Can use tab to switch quickly instead of typing out everything

13. Next I need to type what I want it to run (tesseract.exe) what I want it to use as input (test1.jpg) and what I want it to call the output (outputtest). This looks like C:\Program Files\Tesseract-OCR>tesseract.exe test1.jpg outputtest

Error: At first I forgot what I called the jpg file and wrote test.jpg instead of test1.jpg so it failed and I had to go back and change it

Result: I successfully converted the text of the image into text. However the conversion itself was terrible. The phrase "Philosophical Paper -4500 words min" Was changed to "Vb) losoghicad Xager â€œ4500 words in"

Objective: Retry handwritten to digital with tesseract. This time by myself and with much more simple text

Action:

1. Rename JPG file to test2
2. Place test2.jpg in Tesseract-OCR folder
3. Open command prompt in administrator
4. Change directory to Tesseract-OCR
5. Type tesseract.exe test2.jpg outputtest2

Result: Successfully ran through the process by myself, but the test went even worse. The phrase "Hello World" was changed to an arrow pointing upwards.

Overall Result: After having done some more research it doesn't seem like the technology is developed enough to be useful for me. I could spend a lot of time and effort to teach Tesseract my handwriting in particular, but even then I couldn't find any evidence that it could be close enough to 100% effective at this stage. I think this task is just too large for me to complete this semester.

Week 4

Overall Objective: Scoping of my note taking process
After discussing my proof of concept with Shawn I have decided to look closer at my actual note taking process. I usually take handwritten notes while I am in class, for first quick readings of texts, or to organise my thoughts about essay structures. However, when I am doing a more close reading of a text I tend to move to a digital space as I am more likely to use this work in my essay drafts. For this analysis I am going to be focusing on note taking once it has reached this digital stage. Specifically I am going to analyse my note taking of Saul, J. (2012) Politically Significant Terms and Philosophy of Language

1. Open pdf
2. Open Google docs
3. Read through section
4. Copy interesting quote
5. Paste interesting quote into document
6. Note page number
7. In brackets note why quote is interesting
8. Make seperate section in document titles "Notes"
9. Under notes heading write my more general thoughts of the argument/how I would respond
10. Save document

Overall Result: This process wastes time switching between documents and copy/pasting. It is also sometimes difficult to see how my notes connect to the broader argument of the paper.

Overall objective: Check out hypothes.is

Objective: download hypothes.is
Action:

1. Go to site hypothes.is
2. Sign up
3. Download browser extension

Error: none
Result: downloaded

Objective: Make an annotation
Action:

1. Open PDF of Saul, J. (2012) Politically Significant Terms and Philosophy of Language

Error: Unable to annotate on page
Solution: First I have to click on the browser extension “h.” to turn on annotations for that page.

2. Select title
3. Click on the “ at the side of the page
4. In provided space write “title of the paper” as a test
5. Tag annotation with Saul, language, gender, sex

Note: Can’t annotate emails or google docs. Don’t think there is anything I can do to change that.

Note: The scroll at the side of the page shows exactly where they are and how many notations are above or below the scroll.
Result: Success, annotation made

Objective: Retrieve annotations
Action:

1. Go back to hypothes.is
2. Click on section of annotations just made

Note: Clicking on one of the tags takes me to a list of work other people have annotated using that tag
Result: It seems I can visit annotations in context (which takes me to the PDF I was reading) and share a link to the annotations with other people, but there doesn't seem to be a way to download the annotations so I have access to them when I am offline. Downloading the PDF downloads it without the web browser add ons.

Objective: Export Annotations
Action:

1. Google Export annotations
2. Read https://web.hypothes.is/blog/viewing-and-exporting-hypothesis-annotations/
3. Go to https://jonudell.info/h/facet/?max=50
4. Copy "Share annotations" link
5. Past link into the URL section
6. Click HTML

Error: "Nothing found for this query". This could have been for two reasons. One, I think I made my annotations private. Two, the PDF I was annotating required access to ILearn and this could have been blocking access.

1. Change annotations to public and try actions 4-6 again

Error: "Nothing found for this query"

1. Open a stanford encyclopedia of philosophy entry for self consciousness and make annotations
2. Repeat steps 4-6

Error: "Nothing found for this query"

1. Copy link of the stanford encyclopedia of philosophy web page rather than the "Share Annotations" link

Success: The site found my annotations

2. Click the save button

Error: Did not save

3. Turn off adblocker and click the save button

Result: Annotations Can be saved as either a HTML, CSV or JSON

Objective: Commit to GitHub
Action:

1. Rename file from "Hypothesis" to "Hypothesis-Test"

2. Open Github
3. Make “Notes” repository
4. Click “Upload Files”
5. Click “Choose your Files”
6. Choose file to upload
7. Create commit name and description

Error: Because it is a HTML file, it saves in GitHub as the web code rather than how I want to see it
Solution: According to: https://stackoverflow.com/questions/8446218/how-to-see-an-html-page-on-github-as-a-normal-rendered-html-page-to-see-preview , to view a HTML file I can paste the link to http://htmlpreview.github.io/
Error: While the above solution worked to a point, I was unable to expand some sections in the file through this viewer
Solution: I could redownload singular files to expand the sections and see what is in them. According to: https://stackoverflow.com/questions/4604663/download-single-files-from-github?fbclid=IwAR2xbCoTISbhxiXH3QfkwE0CxzavIyGchUDapdKs18oPw7K9K_W2nT7iC0c
To redownload singular documents within a repository I right click on “Raw” and click “Save Link As”
Result: HTML file successfully saved to GitHub and successfully viewed afterwards. However, this process is by no means quick and easy.

Overall Result: Hypothes.is seems like an extremely useful resource. I like the ease annotating on the document I am using, being able to see exactly where the annotations are in the document, clearly seeing how my annotations fit into the document as a whole, and being able to tag my annotations. Using Hypothes.is would save me a lot of time switching between documents to read and then take notes. It would also save me time copy/pasting quotes.

While the note taking process itself is much better while using hypothes.is, the saving process is more complicated. The process of exporting annotations is important to me because I want to be able to back them up and have access to them at all times.

Out of all my exporting options I find the HTML file to be the clearest to read, however this has issues when saving to GitHub. I currently rely on google drive for version control and back ups of my notes, however this also has issues with reading HTML files.

My note taking process using Hypothes.is would be:

1. Open document that needs annotating
2. Turn on hypothes.is browser extension
3. Highlight section of interesting text
4. Make a note as to why it is interesting in provided space
5. Tag annotations

6. Make more general notes in “Page Notes” section
7. Finish annotating document
8. Copy URL
9. Open https://jonudell.info/h/facet/?max=50
10. Paste URL in the space provided
11. Click Save document as HTML
12. Rename file from “hypothesis” to something more appropriate
13. Open GitHub
14. Open “Notes” repository
15. Click “Upload files”
16. Click “Choose your files”
17. Choose file to upload
18. Create commit name and description

To view the file:

19. Copy URL
20. Paste URL into http://htmlpreview.github.io/

Or redownload the file:

21. Right click Raw
22. Click Save Link As

Both of these options have issues. However, using hypothes.is to take notes could still be a good option if it were possible to automate steps 8-18 and have detailed commit names/descriptions.

For this task I would want a tool that:

1. Allows for user input of URL
2. Allows for user input of file name
3. Allows for user input of description
4. Opens https://jonudell.info/h/facet/?max=50
5. Enters URL
6. Creates HTML file
7. Saves HTML file
8. Renames HTML file to previously inputted name
9. Saves file to output folder
10. Opens github
11. Opens “Notes” repository
12. Goes to “Upload files”
13. Goes to “Choose your files”
14. Choose file to upload
15. Names previously input file name
16. Creates description previously input

By saving the file both on the computer and in github I would have ease of opening the file as well as having backups.

Week 5

Note: Having browsed the Facet website a bit I don't think it has an API I can use for the above tasks. For this reason I am going to change my project to focus on getting my notes off hypothes.is and reformatting those notes to be human readable. For this task I would want a tool that:

1. Allows for input of url
2. Allows for input of file name
3. Interacts with hypothes.is API to get annotation information
4. Outputs annotation information in file
5. Reformats annotation information in a specific way

For future goals I may want a tool that analyses my notes and outputs trends. However, for now I will focus on the above process.

Overall Objective: Get some information off hypothes.is

Objective: Get API key
Result: I thought I needed an API key, but according to https://h.readthedocs.io/en/latest/api/authorization/ "API requests which only read public data do not require authorization". All of the annotations I have made so far with hypothes.is are public. Therefore I am going to try to continue without an API key. If it turns out I do need one there are two ways listed on this site on how to get them.

Objective: Use SWAN on Cloudstor
Action:

1. Open Cloudstor
2. Click on SWAN
3. Start session
4. Open Terminal
5. Use command ls and cd to move around

Result: I know how to use the terminal in SWAN. I will need to do more research in how to use R or Python on this, but I am happy for now that I know something, and am familiarising myself with the layout of SWAN.

Objective: Get some information off hypothes.is
Action:

1. Google hypothes.is API
2. This page https://h.readthedocs.io/en/latest/api-reference/v1/#tag/annotations has a section on getting annotations. It states the command GET /annotations/{id} and https://api.hypothes.is/api/annotations/{id}. When I click the "visit annotation in context" link on hypothes.is, it changes the end of the URL to include /#annotations:K_JrMsbiEem5hutSGynvDw. I assume this is the annotation ID

3. Type https://api.hypothes.is/api/annotations/{K_JrMsbiEem5hutSGynvDw} into a search bar

Error: Either the resource you requested doesn't exist, or you are not currently authorized to see it
Solution: Remove the {} and just type
https://api.hypothes.is/api/annotations/K_JrMsbiEem5hutSGynvDw
Result: Success, I got some information off hypthes.is. It isn't very clear, but I expected that. This is a section of it. {"id": "K_JrMsbiEem5hutSGynvDw", "created": "2019-08-25T02:43:51.327033+00:00", "updated": "2019-08-30T08:46:31.519316+00:00", "user": "acct:georgiarutherford@hypothes.is", "uri": "https://plato.stanford.edu/entries/self-consciousness/", "text": "People to look at for my idea of the self and other: Fichte (1794\u20131795; Wood 2006), Hegel (1807; Pippin 2010), and, from a somewhat different perspective, Mead (1934; Aboulafia 1986)"

Objective: Get annotation information through the terminal
Action:

1. Open terminal in SWAN
2. Type get https://api.hypothes.is/api/annotations/K_JrMsbiEem5hutSGynvDw

Error: bash: get: command not found
Solution: I must have made a sigh of defeat, because my fiance walked over had a look at what I was doing and said "try wget" which worked. Writing the "w" will make it get from the web.
Solution is to write wget https://api.hypothes.is/api/annotations/K_JrMsbiEem5hutSGynvDw
Result: Success, the text output from the command was:
--2019-08-30 12:04:45-- https://api.hypothes.is/api/annotations/K_JrMsbiEem5hutSGynvDw
Resolving api.hypothes.is (api.hypothes.is)... 104.20.215.15, 104.20.214.15, 2606:4700:10::6814:d70f, ...
Connecting to api.hypothes.is (api.hypothes.is)|104.20.215.15|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/json]
Saving to: 'K_JrMsbiEem5hutSGynvDw.1'
[<=>
] 1,915 --.-K/s in 0s
2019-08-30 12:04:46 (16.8 MB/s) - 'K_JrMsbiEem5hutSGynvDw.1' saved [1915]
A file has been downloaded to SWAN, when I type ls I can see that it is there.
Note: I tried the exact same process on git-bash and it didn't work

Overall Result: Very successful. Not only have I gotten information from hypothes.is, but I was able to do so using terminal.

Overall Objective: Reformat the file so it is more human readable.

Objective: Rename file K_JrMsbiEem5hutSGynvDw
Action:

1. Type mv ~/K_JrMsbiEem5hutSGynvDw ~/hypothesistestnotes.txt

Result: Success

Objective: Move hypothesistestnotes.txt into its own folder
Action:

1. Type mkdir Notes
2. Type mv ~/hypothesistestnotes.txt ~/Notes/hypothesistestnotes.txt
3. Type ls ~/Notes

Result: Success, Notes folder has been made and hypothesistestnotes.txt has been moved into it

Objective: Copy file
Action:

1. Type cp hypothesistestnotes.txt Notes

Error: I made a file called notes instead of making a file in notes. I renamed it to hypothesiscopy.txt
Result: Now I can mess around with it without changing my original file

Objective: Change file type to csv
Action:

1. Type mv hypothesiscopy.txt hypothesiscopy.csv
2. Open the csv document

Result: Successfully made a csv document. The way sections are separated because of commas is useful because the information is separated by commas, such as *"id": "K_JrMsbiEem5hutSGynvDw", "created": "2019-08-25T02:43:51.327033+00:00"*. However, this is also problematic because the annotations and quotes themselves also use commas, such as *Hegel (1807; Pippin 2010), and, from a somewhat different perspective, Mead (1934; Aboulafia 1986)*.

Overall result: I changed the format so it was a little more human readable. It still isn't very clear though.

Overall objective: make a script that will download annotations for me

Objective: Get first line of script to run
Action:

1. Type nano DownloadNotes.sh
2. Type wget https://api.hypothes.is/api/annotations/"$1"
3. Save with ctrl o and exit with
4. Test bash DownloadNotes.sh K_JrMsbiEem5hutSGynvDw

Error: DownloadNotes.sh: line 1: unexpected EOF while looking for matching `""
DownloadNotes.sh: line 2: syntax error: unexpected end of file

Solution: I copy and pasted the command that I wrote in my learning journal. This error was solved by deleting the “$1” and writing it again in SWAN
Result: Success, annotations downloaded. I will delete this file (using rm) and write the rest of my script.

Objective: rename file
Action:

1. Type mv “$1” “$2”
2. Save with ctrl o, exit with ctrl x
3. Type bash DownloadNotes.sh K_JrMsbiEem5hutSGynvDw programtest.txt
4. Type ls to test see if it was successful

Error: I ran into the same problem as last time.
Solution: I need to stop copy/pasting from my learning journal
Result: Success, the file is downloaded and renamed

Objective: Add comments
Action:

1. Type:
   # Will download notes from hypothesis and rename them for you
   # Usage: bash DownloadNotes.sh annotationID Filename
2. Give comments to my fiance and get him to run the program to see if they are understandable by someone else

Result: Success, although I had to tell him what the annotation ID is.

Overall Result: Success, I made my first script

Overall Objective: Delete some text automatically from the hypothesiscopy.txt file

Notes: My file currently looks like this:
{"id": "K_JrMsbiEem5hutSGynvDw", "created": "2019-08-25T02:43:51.327033+00:00", "updated": "2019-08-30T08:46:31.519316+00:00", "user": "acct:georgiarutherford@hypothes.is", "uri": "https://plato.stanford.edu/entries/self-consciousness/", "text": "People to look at for my idea of the self and other: Fichte (1794\u20131795; Wood 2006), Hegel (1807; Pippin 2010), and, from a somewhat different perspective, Mead (1934; Aboulafia 1986)", "tags": ["Hegel", "self-consciousness", "self"], **"group": "__world__", "permissions": {"read": ["group:__world__"], "admin": ["acct:georgiarutherford@hypothes.is"], "update": ["acct:georgiarutherford@hypothes.is"], "delete": ["acct:georgiarutherford@hypothes.is"]}, "target": [{"source": "https://plato.stanford.edu/entries/self-consciousness/", "selector": [{"type": "RangeSelector", "endOffset": 1166, "startOffset": 835, "endContainer": "/div[1]/div[2]/div[3]/div[1]/div[1]/div[4]/p[11]", "startContainer": "/div[1]/div[2]/div[3]/div[1]/div[1]/div[4]/p[11]"}, {"end": 14815, "type": "TextPositionSelector", "start": 14484}, {"type": "TextQuoteSelector",** "exact": "Another,

related tradition has argued that an awareness of subjects\nother than oneself is a necessary condition of self-consciousness (see\n \u00a74.4).\n Historical variations on such a view can be found in Fichte\n(1794\u20131795; Wood 2006), Hegel (1807; Pippin 2010), and, from a\nsomewhat different perspective, Mead (1934; Aboulafia 1986).", **"prefix": "n objective world (see\n \u00a74.3).\n ", "suffix": "\n\n\nFichte offers the most influe"}]}], "document": {"title": ["Self-Consciousness"]}, "links": {"html": "https://hypothes.is/a/K_JrMsbiEem5hutSGynvDw", "incontext": "https://hyp.is/K_JrMsbiEem5hutSGynvDw/plato.stanford.edu/entries/self-consciousness/", "json": "https://hypothes.is/api/annotations/K_JrMsbiEem5hutSGynvDw"}, "flagged": false, "hidden": false, "user_info": {"display_name": null}}**
Ideally I would want my program to be able to delete all the text I have put in bold above and then separate out the rest of the text into sections (ID, Created, Updated, User, URL, Text, Tags, Exact). I will also want to keep the original text file for reference in case the program deletes something it shouldn't. This will use a command input.txt > output.txt
My research on how to edit text file has led me to this https://www.gnu.org/software/sed/manual/sed.html#Overview .

Objective: Use sed command to edit my text file
Action:

1. Type sed '/"group":/,/"TextQuoteSelector"/' hypothesiscopy.txt > testdelete.txt

Error: File made with nothing in it. Might be because I forgot to use the d command

1. Type sed '/"group":/,/"TextQuoteSelector"/d' hypothesiscopy.txt > testdelete.txt

Error: bash: Type: command not found. Testdelete.txt made with nothing in it

1. Google some more because I am clearly missing something.
2. Read https://unix.stackexchange.com/questions/243207/how-can-i-delete-everything-until-a-pattern-and-everything-after-another-pattern
3. Type grep -o '"id".*"group"' hypothesiscopy.txt > testdelete.txt
4. Type grep -o '"exact".*"prefix"' hypothesiscopy.txt >> testdelete.txt

Result: Kind of a success, my testdelete.txt document now just shows:

"id": "K_JrMsbiEem5hutSGynvDw", "created": "2019-08-25T02:43:51.327033+00:00", "updated": "2019-08-30T08:46:31.519316+00:00", "user": "acct:georgiarutherford@hypothes.is", "uri": "https://plato.stanford.edu/entries/self-consciousness/", "text": "People to look at for my idea of the self and other: Fichte (1794\u20131795; Wood 2006), Hegel (1807; Pippin 2010), and, from a somewhat different perspective, Mead (1934; Aboulafia 1986)", "tags": ["Hegel", "self-consciousness", "self"], "group"
"exact": "Another, related tradition has argued that an awareness of subjects\nother than oneself is a necessary condition of self-consciousness (see\n \u00a74.4).\n Historical variations on such a view can be found in Fichte\n(1794\u20131795; Wood 2006), Hegel (1807; Pippin 2010), and, from a\nsomewhat different perspective, Mead (1934; Aboulafia 1986).", "prefix"

This is the text I want to keep except the “group” and “prefix”, however I need them because for the code I need the words that will always be at the end of the text I want to copy. The text before them will change depending on the tags or quotes I use.

Objective: Delete “group” and “prefix” as well
Action:

1. Type grep -o '"id".*"group"' | sed ‘“group”d’ hypothesiscopy.txt > testdelete.txt

Error: sed: -e expression #1, char 1: unknown command: `"'
Solution: the sed command needs to use ‘/^”group”d/ to indicate a word

1. Type grep -o '"id".*"group"' | sed '/^"group"/d' hypothesiscopy.txt > testdelete.txt

Error: Didn’t do what I wanted. I had to cancel the task because it didn’t complete
Solution: I’m going to try this in a less complicated way and see if I can just delete “group” somehow

1. Type sed '/^"group"/d' hypothesiscopy.txt > testdelete.txt

Error: text still in text.

1. Google again: https://www.cyberciti.biz/faq/howto-sed-exact-match-and-delete-words-within-a-file/
2. Try sed 's/word-to-find//g' input.file > output.file
3. Type sed 's/”group”//g' hypthesiscopy.txt > testdelete.txt

Result: Success, the word group is gone

Objective: put some of these commands together
Action:

1. Type nano DownloadReformatTest.sh
2. Type:
   # Will download notes from hypothesis, rename original, move info into a new file without unwanted sections
   # Usage: bash DownloadReformatTest.sh annotationID Filename
   wget https://api.hypothes.is/api/annotations/"$1"
   grep -o '"id".*"group"' “$1” > output.txt
   sed 's/”group”//g' output.txt > “$2”
   grep -o '"exact".*"prefix"' “$1” > output.txt
   sed 's/”prefix”//g' output.txt >> “$2”
   rm “output.txt”
   Mv “$1” “$2-original”
3. Test by typing bash DownloadReformatTest.sh K_JrMsbiEem5hutSGynvDw testdeleteprogram.txt

Error: sed: -e expression #1, char 13: unterminated `s' command. Original document is saved, and testdeleteprogram.txt was made with the first section of text.
Solution: The error seems to be with line 5, it says: sed 's?"prefix"//g' output.txt >> "$2". I think I made a typo. Changing this to sed ‘s/ worked.
Error: the original copy is called testdeleteprogram.txt-original
Solution: change all the “$2” to “$2.txt” and the last one to “$2-original.txt”

Result: Success

Overall Result: Successfully made a script that downloads annotations from hypothes.is and delete a bunch of unwanted output. I doubt my script is the quickest way of doing this, but it is what made sense to my mind.

Overall Objective: Format further

Objective: Separate out my content using the grep command I learnt above
Action:

1. Type:
   wget https://api.hypothes.is/api/annotations/"$1"
   grep -o '"id".*","' "$1" > “$2”.txt
   grep -o '"created".*","' "$1" >> “$2”.txt
   grep -o '"updated".*","' "$1" >> “$2”.txt
   grep -o '"user".*","' "$1" >> “$2”.txt
   grep -o '"uri".*","' "$1" >> “$2”.txt
   grep -o '"text".*","' "$1" >> “$2”.txt
   grep -o '"tags".*","' "$1" >> “$2”.txt
   grep -o '"exact".*","' "$1" >> “$2”.txt
   mv “$1” “$2-original.txt”
2. Test by typing bash ReformatTest2.sh K_JrMsbiEem5hutSGynvDw reformattest

Error: reformattest.txt has no text
Solution: I left in the “” from the last script, but that was apart of the actual word. Take these out
Error: The wild card * means all, so it does this for all the , in the text. I want it to just do the next one. But even if I could do this I would run into the same problem I had using the csv. That is, commas are used in my quotations as well as separating categories
Solution: I am going to try to separate it by the words and then delete the first instance of every duplicated word.

1. First I will see if I can separate out by words by using the script
   wget https://api.hypothes.is/api/annotations/"$1"
   grep -o '"id".*"created"' "$1" > “$2”.txt
   grep -o '"created".*"updated"' "$1" >> “$2”.txt
   grep -o '"updated".*"user"' "$1" >> “$2”.txt
   grep -o '"user".*"uri"' "$1" >> “$2”.txt
   grep -o '"uri".*"text"' "$1" >> “$2”.txt
   grep -o '"text".*"tags"' "$1" >> “$2”.txt
   grep -o '"tags".*"exact"' "$1" >> “$2”.txt
   grep -o '"exact".*"prefix"' "$1" >> “$2”.txt
   mv “$1” “$2-original.txt”
2. Test by typing bash ReformatTest2.sh K_JrMsbiEem5hutSGynvDw reformattest

Error: mv: cannot stat ‘“K_JrMsbiEem5hutSGynvDw”’: No such file or directory
Solution: I hadn’t reformatted some of the “” after pasting them from my learning journal

3. Retype the “” and test by typing bash ReformatTest2.sh K_JrMsbiEem5hutSGynvDw reformattest

Error: a lot of the info I don’t want is back in there
Solution: change step 8 to be from “tags” to “group” instead of “exact”
Result: Success, all the information is on a new line. However, each line also has an extra work at the end, such as "id": "K_JrMsbiEem5hutSGynvDw", "created"

Objective: Remove the last word from each line
Action:

1. Type:
   wget https://api.hypothes.is/api/annotations/"$1"
   grep -o '"id".*"created"' "$1" > output.txt
   sed 's/"created"//g' output.txt > "$2".txt
   grep -o '"created".*"updated"' "$1" > output.txt
   sed 's/"updated"//g' output.txt >> "$2".txt
   grep -o '"updated".*"user"' "$1" > output.txt
   sed 's/"user"//g' output.txt >> "$2".txt
   grep -o '"user".*"uri"' "$1" > output.txt
   sed 's/"uri"//g' output.txt >> "$2".txt
   grep -o '"uri".*"text"' "$1" > output.txt
   sed 's/"text"//g' output.txt >> "$2".txt
   grep -o '"text".*"tags"' "$1" > output.txt
   sed 's/"tags"//g' output.txt >> "$2".txt
   grep -o '"tags".*"group"' "$1" > output.txt
   sed 's/"group"//g' output.txt >> "$2".txt
   grep -o '"exact".*"prefix"' "$1" > output.txt
   sed 's/"prefix"//g' output.txt >> "$2".txt
   rm output.txt
   mv "$1" "$2-original.txt"
2. Test by typing bash ReformatTest2.sh K_JrMsbiEem5hutSGynvDw reformattest

Result: Succes

Output in the reformattest-original.txt is:
{"id": "K_JrMsbiEem5hutSGynvDw", "created": "2019-08-25T02:43:51.327033+00:00", "updated": "2019-08-30T08:46:31.519316+00:00", "user": "acct:georgiarutherford@hypothes.is", "uri": "https://plato.stanford.edu/entries/self-consciousness/", "text": "People to look at for my idea of the self and other: Fichte (1794\u20131795; Wood 2006), Hegel (1807; Pippin 2010), and, from a somewhat different perspective, Mead (1934; Aboulafia 1986)", "tags": ["Hegel", "self-consciousness", "self"], "group": "__world__", "permissions": {"read": ["group:__world__"], "admin": ["acct:georgiarutherford@hypothes.is"], "update": ["acct:georgiarutherford@hypothes.is"], "delete": ["acct:georgiarutherford@hypothes.is"]}, "target": [{"source": "https://plato.stanford.edu/entries/self-consciousness/", "selector": [{"type": "RangeSelector", "endOffset": 1166, "startOffset": 835, "endContainer":

"/div[1]/div[2]/div[3]/div[1]/div[1]/div[4]/p[11]", "startContainer": "/div[1]/div[2]/div[3]/div[1]/div[1]/div[4]/p[11]"}, {"end": 14815, "type": "TextPositionSelector", "start": 14484}, {"type": "TextQuoteSelector", "exact": "Another, related tradition has argued that an awareness of subjects\nother than oneself is a necessary condition of self-consciousness (see\n \u00a74.4).\n Historical variations on such a view can be found in Fichte\n(1794\u20131795; Wood 2006), Hegel (1807; Pippin 2010), and, from a\nsomewhat different perspective, Mead (1934; Aboulafia 1986).", "prefix": "n objective world (see\n \u00a74.3).\n ", "suffix": "\n\n\nFichte offers the most influe"}]}], "document": {"title": ["Self-Consciousness"]}, "links": {"html": "https://hypothes.is/a/K_JrMsbiEem5hutSGynvDw", "incontext": "https://hyp.is/K_JrMsbiEem5hutSGynvDw/plato.stanford.edu/entries/self-consciousness/", "json": "https://hypothes.is/api/annotations/K_JrMsbiEem5hutSGynvDw"}, "flagged": false, "hidden": false, "user_info": {"display_name": null}}

Output in the reformattest.txt is:
"id": "K_JrMsbiEem5hutSGynvDw",
"created": "2019-08-25T02:43:51.327033+00:00",
"updated": "2019-08-30T08:46:31.519316+00:00",
"user": "acct:georgiarutherford@hypothes.is",
"uri": "https://plato.stanford.edu/entries/self-consciousness/",
"text": "People to look at for my idea of the self and other: Fichte (1794\u20131795; Wood 2006), Hegel (1807; Pippin 2010), and, from a somewhat different perspective, Mead (1934; Aboulafia 1986)",
"tags": ["Hegel", "self-consciousness", "self"],
"exact": "Another, related tradition has argued that an awareness of subjects\nother than oneself is a necessary condition of self-consciousness (see\n \u00a74.4).\n Historical variations on such a view can be found in Fichte\n(1794\u20131795; Wood 2006), Hegel (1807; Pippin 2010), and, from a\nsomewhat different perspective, Mead (1934; Aboulafia 1986).",

Overall result: I am very happy with my results so far. I will have to be very careful about using individual words in quotations as if the word is id, created, updated, user, uri, text, tags, exact, group or prefix it would mess up my code. I do not foresee this becoming too much of an issue, but I have made sure my code keeps a copy of the original document just in case.

Data Carpentry

The Unix Shell

Week 4
Overall Objective: Complete up to the end of Working Files and Directories

Objective: Set up on my home computer
Action:

1. Download data-shell.zip
2. Move to desktop
3. Unzip data-shell.zip
4. Open https://gitforwindows.org/
5. Click download
6. Click open git bash
7. Type cd

Error: None
Result: Complete

Objective: Read through Introducing the Shell to remind me of what we went over in class
Result:
$ is a prompt to input instructions
ls is a list of the contents of the current directory
Mis-typed commands are the most common reason for errors
Useful command discussed in class was ctrl l which clears the screen

Objective: Exploring More ls Flags
Action:

1. Type ls -1

Error: I put a space between the - and the 1 to start with. Solution was to take away that space.
Error: Just realised that it is an L and not a 1
Solution: Type ls -l

2. Type ls - l -h

Result: List is clearer, command -h makes it human readable

Objective: Listing Recursively and By Time
Action:

1. Type ls -R -t

Result: Files sorted by the time of last change. Lots of text appeared on the screen and showed no sign of stopping, so I closed and reopened it to move on.

Objective: See inside desktop
Action:

1. Type ls -F Desktop

Error: ls: cannot access 'Desktop': No such file or directory
Solution: As I reopened git-bash I was in the wrong directory. I had to type cd again to get back to the home directory and then it worked.
Result: Success, the data-shell folder is in the list

Objective: Look in data-shell
Action:

1. Type ls -F Desktop/data-shell

Error: The displayed list is just data-shell/
Solution: Because of how I unzipped the file there is a folder data-shell called data-shell. To solve this type ls -F Desktop/data-shell/data-shell to see inside that second folder
Result:Success

Objective: Change directory
Action

1. cd desktop
2. cd data-shell
3. cd data-shell
4. cd data

Note: The instructions say it doesn't print anything but this is incorrect for me. It tells me which directory I am in:

```
georg@DESKTOP-LFLG4GE MINGW64 ~
$ cd desktop
georg@DESKTOP-LFLG4GE MINGW64 ~/desktop
$ cd data-shell
georg@DESKTOP-LFLG4GE MINGW64 ~/desktop/data-shell
$ cd data-shell
georg@DESKTOP-LFLG4GE MINGW64 ~/desktop/data-shell/data-shell
$ cd data
georg@DESKTOP-LFLG4GE MINGW64 ~/desktop/data-shell/data-shell/data
```

Typing pwd does also tell me which directory I am in.
Result: Success

Objective: Move up a directory
Action:

1. Type cd . .

Error: bash: cd: too many arguments
Solution: I originally thought there was a space between the two .'s. Remove this space and it works.

Objective: Move to home directory and then back into data
Action:

1. Type cd on its own (takes you back to the home directory)
2. Type cd Desktop/data-shell/data-shell/data (move back to data in one step)
3. Check where I am with pwd and ls -F

Result: Success

Objective: Absolute vs Relative Paths
To get back to their home directory they could use option 5, 8 or 9
Error: they could also use option 7 which is cd~/data/..

Result: Almost correct, the solution said option 7 would also work but is needlessly complicated

Objective: Relative Path Resolution
The command ls -F ../backup would display the fourth option because .. would take you back up into users and then /backup would take you from users into backup. Then the list of files in backup is original/ pnas_final/ pnas_sub/.
Result: Correct

Objective: ls Reading Comprehension
Both option 2 and 3 would result in that output
Result: Correct

Objective: Make a directory
Action
1. Type mkdir thesis
2. Check that directory was made by typing ls -F

Result: success thesis directory is in the list

Objective: Create a text file
Action
1. Type cd thesis to move into the thesis directory
2. Type nano draft.txt to make a new file
3. Type text
4. Press ctrl O to save
5. Press ctrl X to quit out of text editor

Result: Success

Objective: Creating Files a Different Way
Action:
1. Type touch my_file.txt
2. Type ls -l

This made a file which contains no data. I am really not sure when you would want to make a file like this.
Result: Apparently some programs require there to be an already existing empty file which they can populate. So the touch command allows you to make blank text files.

Objective: Rename file
Action:
1. Type mv thesis/draft.txt thesis/quotes.txt
2. Type ls thesis

Result: Success. Although the command mv means “move” it can be used to rename

Objective: Move quotes.txt to current directory

Action:

1. Type mv thesis/quotes.txt

Error: mv: missing destination file operand after 'thesis/quotes.txt'
Solution: I forgot to say where it was going. Type mv thesis/quotes.txt . to move to current directory
Result: Success

Objective: Moving to current folder
The answer will be mv ../analyzed/sucrose.dat ../analyzed/maltose.dat .
Result: Success

Objective: Renaming Files
The only correct option is 2.
Result: Success

Objective: Moving and Copying
The second option is correct, as the file protiens-saved.dat was copied to the directory above the one we are looking into
Result: Success

Objective: Using rm Safely
Typing rm -i thesis_backup/quotations.txt will ask for confirmation before deleting the file. This is useful so we don't accidentally delete important things.
Result: Success. Also interesting to note that Unix shell doesn't have a trash bin. Also rm will not delete a directory unless we also type -r. For example rm -r thesis would delete the thesis directory.

Objective: Copy with Multiple Filenames
Action:

1. Type cp amino-acids.txt animals.txt backup/
2. Type ls -F backup
3. Type cp amino-acids.txt animals.txt morse.txt

The first set of files were copied to the backup directory. The second set of files received the error "cp: target 'morse.txt' is not a directory".
Result: Success. The last item on a list of things to copy must be where to copy them to.

Objective: List filenames matching a pattern
Option 3 produces ethane.pdb methane.pdb.
Result: Success

Objective: More on Wildcards
$ cp *dataset* backup/datasets
$ cp *calibration.txt backup/calibration

$ cp 2015-11-* send_to_bob/all_november_files/
$ cp *23 send_to_bob/all_datasets_created_on_a_23rd/
Error: I forgot that she only wanted to send bob datasets. So the last command should be $ cp *-23-datasets send_to_bob/all_datasets_created_on_a_23rd/.

Objective: Organizing Directories and Files
Action:

1. Type cp fructose.dat sucrose.dat analyzed/
2. Type rm fructose.dat sucrose.dat

Result: I think my commands probably would have worked, but the solution was mv *.dat analyzed. I was copying the files and then removing them afterwards instead of just moving them. Their solution would have been quicker.

Objective: Reproduce a folder structure
The first two options would work . The second option would take a bit longer as it involves actually moving between the directories.
The third option would fail as the 2016-05-20 directory has not been made yet.
In the fourth option they only changed directory into the 2016-05-20 directory before making the other two directories. Therefore they would not be inside the data directory as desired.
Result: Success

Overall Result: Completed up to the end of Working Files and Directories

Week 5

Overall Objective: Finish the Unix Shell

Objective: What Does sort -n Do?
Typing sort -n makes it sort numerically.
Result: Success

Objective: What Does >> Mean?
Typing >> adds the text to the existing file, whereas > overwrites the text.
Result: Success

Objective: Appending Data
Action:

1. Type head -n 3 animals.txt > animals-subset.txt

Error: head: cannot open 'animals.txt' for reading: No such file or directory
Solution: I forgot to change directories, I was still in molecules. Delete animals-subset.txt file using rm, move to data using cd ~/Desktop/data-shell/data-shell/data/.

1. Type head -n 3 animals.txt > animals-subset.txt
2. Type tail -n 2 animals.txt >> animals-subset.txt

Answer: Option 3, the first 3 and the last 2 lines of animals.txt are copied to animals-subset.txt
Result: Success

Objective: Piping Commands Together
Option 4 is correct. First you want to get the number of lines, then you want to sort it numerically so the top numbers are the smallest, then you want to see the first 3, and you use the pipe to connect them.
Result: Success

Objective: Pipe Reading Comprehension
The answer is 2012-11-05,raccoon, 2012-11-06,deer, 2012-11-06,rabbit. The first step shows what is in the list, the second step takes the first 5 options, the third step takes the last 3 from those 5, and the last step sorts the in the reverse order.
Result: Success

Objective: Pipe Construction
Type cut -d , -f 2 animals.txt | uniq
Error: I need to sort first because the uniq command just cuts out adjacent lines
Solution: Type cut -d , -f 2 animals.txt | sort | uniq
Result: Success

Objective: Which Pipe?
Option 4 is the correct answer. The first step cuts based on the comma and then shows the second column, the next step sorts, the final step shows a count of all animals with the same name next to each other.
Result: Success

Objective: Wildcard Expressions
You would need to write them out separately as *A.txt or *B.txt. This would show an error if there were no files with the A or B.
Result: Success

Objective: Removing Unneeded Files
The solution is rm *.txt as this would remove all files that end with .txt.
Result: Success

Objective: Variables in Loops
The first code outputs a list of the .pdb documents each time it encounters a document ending in .pdb.
The second code lists the file it is encountering every time it encounters a .pdb file.
Result: Success

Objective: Limiting Sets of Files

1. Option for is the correct answer. Everytime the code encounters a file starting with c, it lists that file.
2. Option 4 is the correct answer. Everytime the code encounters a file containing the letter c, it lists that file.

Result: Success

Objective: Saving to a File in a Loop - Part One
Option 1 is correct. The > command was used and so the file will be overwritten each loop until the last. To avoid this the >> command should have been used.
Result: Success

Objective: Saving to a File in a Loop - Part Two
Option 3 is correct. Since the >> command was used all files ending in .pdb will be saved to the file all.pdb.
Result: Success

Objective: Doing a Dry Run
We would want to run the second option because it will echo what is in the quotations onto the screen, rather than echoing something into the file analysed-$file.
Result: Success

Objective: Nested Loops
A folder is made for each species/temperature combination.
Result: Success

Objective: List Unique Species
Action:

1. Type species.sh

Error: I forgot to write the nano and got the error bash: species.sh: command not found
Solution: Type nano species.sh

2. Type:

```
# Usage: bash species.sh filename
cut -d , -f 2 "$1" | sort | uniq
```

Error: I keep trying to click to move which point of the text I am editing, you have to use arrow keys.
Result: I missed the part where it said it needed to be able to take any number of file names.
The real solution is a loop:

```
# Script to find unique species in csv files where species is the second data field
# This script accepts any number of file names as command line arguments
# Loop over all files
for file in $@
do
        echo "Unique species in $file:"
```

```
        # Extract species names
        cut -d , -f 2 $file | sort | uniq
done
```

Objective: Why Record Commands in the History Before Running Them?
If something goes very wrong you can see what happened without actually repeating it.
Result: Success

Objective: Variables in Shell Scripts
Option 2. The first line is head -n 1 *.pdb, this will get the first line of all files ending in .pdb. The second line is, tail -n 1 .pdb, this will get the last line of all files ending in .pdb.
Result: Success

Objective: Find the Longest File With a Given Extension
The script would look something like:
wc -l $1/*.$2
sort -n
tail -n 2
head -n 1
Error: I keep forgetting that you can use pipelines to put things all on one line. I believe my version would still work if you were running it from a file.
Result: Success

Objective: Script Reading Comprehension
The first script will repeat any files with a . in the name.
The second script allows you to input 3 different files and it will show the contents of those files.
The last script will repeat all .pdb files.
Result: Success

Objective: Debugging Scripts
Echo command not working due to a typo of $datafile as $datfile
Result: Success

Objective: Using grep
Option 3 is correct. The command -w restricts it to whole words, which is why the word “of” is included but not “software”
Result: Success

Objective: Tracking a Species
cut -d : -f 2 | cut -d , -f 1,3 | grep -w $1 -r $2 > $1.txt
Error: I mixed up the order
Solution: grep -w $1 -r $2 | cut -d : -f 2 | cut -d , -f 1,3 > $1.txt
Note: the instructions say to use man grep, but the man command doesn’t work on git bash

Objective: Little Women
I had trouble with this answer. I think I am still a little unclear on loops and should go over the information again. The answer is:

```
for sis in Jo Meg Beth Amy
do
        echo $sis:
        grep -ow $sis LittleWomen.txt | wc -l
done
```

Objective: Matching and Subtracting
Option 1 is correct.
Result: Success

Objective: find Pipeline Reading Comprehension
This script counts the number of lines of all files ending with .dat and lists them numerically.
Result: Success

General

Common Errors

Error: get command doesn't download what you wanted it to
Solution: wget is the command to download from the web

Error: unexpected EOF while looking for matching `"' or line 2: syntax error: unexpected end of file
Solution: Don't copy and paste from learning journal/change the format of the "". In fact if you ever get an error after using a script from the text file just check all the "" are correctly formatted.

Error: sed command not working as expected
Solution: Check that you are using /'s to indicate words and not "'s like you would use with grep.

Error: Certain symbols do not count as text in LaTex
Solution: Most of the time writing \ or \text before the symbol fixed this. For the > symbol you have to write $>$.