

Learning Journal

Kylie Reynolds
Semester 2, 2019



Contents

1	Learning journal: General Info	5
2	GitHub Commit	6
3	Errors	8
3.1	Data Carpentry	8
3.1.1	Data Organization in Spreadsheets for Social Scientist	8
3.1.2	Unix Shell	8
3.1.3	OpenRefine for Social Science Data	8
3.1.4	R for Social Scientists	8
3.2	Overleaf - LaTeX Editor	8
3.3	Proof of Concept	8
4	Data Carpentry - Data Organization in Spreadsheets for Social Scientists	9
4.0.1	Introduction	9
4.0.2	Formatting data tables in Spreadsheets	10
4.0.3	Formatting problems	14
4.0.4	Dates as data	15
4.0.5	Quality assurance	16
4.0.6	Exporting data	19
5	Data carpentry - The Unix Shell	20
5.1	Introducing the Shell	20
5.1.1	The Shell	20
5.1.2	Flexibility and automation	21
5.2	Navigating Files and Directories	21
5.2.1	Getting Help	23
5.2.2	Shortcuts	26
5.3	Working with files and directories	27
5.3.1	Creating directories (folders):	27
5.3.2	What's in a name?	29
5.3.3	Moving files and directories:	29
5.3.4	Moving to the current folder	29
5.3.5	Copying files and directories	30
5.3.6	Renaming files exercise:	30
5.3.7	Removing files and directories	30
5.3.8	Operations with multiple files and directories	31
5.3.9	Using wildcards for accessing multiple files at once	31
5.3.10	Organising Directories and Files	32
5.4	Pipes and Filters	32
5.4.1	Sort -n	33
5.4.2	what does » mean	33
5.4.3	The pipeline	34
5.5	Loops	34
5.5.1	Follow the prompt	34
5.5.2	Same symbols different meanings	35
5.5.3	Spaces in Names	36

5.6	Shell Scripts	38
5.6.1	Text vs. Whatever	39
5.6.2	Double-Quotes Around Arguments	40
5.6.3	List Unique Species exercise	42
5.6.4	Why Record Commands in the History Before Running Them?	42
5.6.5	Variables in Shell Scripts exercise	42
5.6.6	Find the Longest File With a Given Extension exercise	42
5.6.7	Script Reading Comprehension exercise	43
5.6.8	Debugging Scripts exercise	43
5.7	Finding Things	44
5.7.1	Forever, or Five Years	44
5.7.2	Using grep exercise	46
5.7.3	Wildcards	46
5.7.4	Tracking a Species exercise	47
5.7.5	Little Women exercise	48
5.7.6	Listing vs. Finding	48
5.7.7	Matching and Subtracting exercise	49
5.7.8	Binary Files	49
5.7.9	find Pipeline Reading Comprehension exercise	50
5.7.10	Finding Files With Different Properties exercise	50
6	Data carpentry - OpenRefine for Social Science Data	51
6.1	Introduction	51
6.1.1	Motivations for the OpenRefine Lesson	51
6.1.2	Features	51
6.2	Working with OpenRefine	51
6.2.1	Creating a new OpenRefine project	52
6.2.2	Using Facets	52
6.2.3	Exercise	53
6.2.4	More on Facets	53
6.2.5	Using clustering to detect possible typing errors	54
6.2.6	Different clustering algorithms	55
6.2.7	Transforming data	55
6.2.8	Using undo and redo	57
6.2.9	Trim Leading and Trailing Whitespace	57
6.3	Filtering and Sorting with OpenRefine	58
6.3.1	Filtering	58
6.4	Excluding entries	59
6.4.1	Sort	59
6.4.2	Sorting by multiple columns	59
6.5	Examining Numbers in OpenRefine	60
6.5.1	Numbers	60
6.5.2	Numeric facet	61
6.6	Using scripts	61
6.6.1	How OpenRefine records what you have done	61
6.6.2	Saving your work as a script	61
6.6.3	Importing a script to use against another dataset	62
6.7	Exporting and Saving Data from OpenRefine	62

6.7.1	Saving and Exporting a Project	62
6.7.2	Exporting Cleaned Data	62
6.8	Other Resources in OpenRefine	62
7	Overleaf	63
8	Proof of Concept	76
8.1	Elaboration stage	76
8.1.1	Applying for an API from Twitter	76

1 Learning journal: General Info

12/08/2019 - 11:31am

Instructions for this week

Before class next week, in your learning journal, finish reading to “Formatting Problems” and document the two exercises we have (hopefully) done today in your GitHub repository.

Problem: not sure how to do this or what I am meant to be documenting?? Try and find out through looking through slack comments/questions or re-listening to echo recording, if no answer found, ask on Slack.

Solution: Notes from Brian on Slack for journal entry (09/08/2019, 7:57pm):

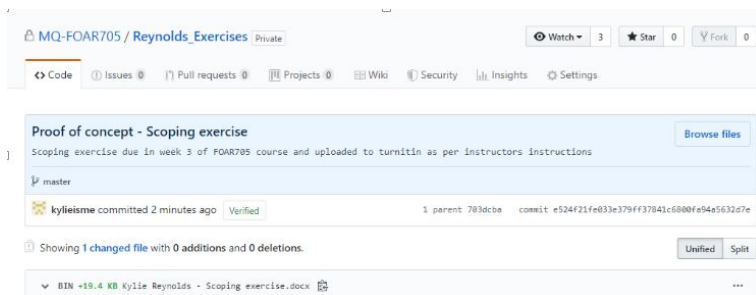
All technical work outside of class should be recorded in a laboratory notebook (document kept with the code or on Cloudstor) which documents the intention of the action, the specifics of the action taken, and the results, along with any marginal notes for improvement or updating your mental model of what should have had happened.

This documentation includes: an answer to the specific objective: "What do you intend to be the result of the action you are about to take", the action to be taken containing timestamp, and commands or actions performed, and the result, documenting what happened, success or failure in relation to the objective, and error states. Documentation of errors (each in its own entry) and their remediation are strongly encouraged.

2 GitHub Commit

16 August 2019

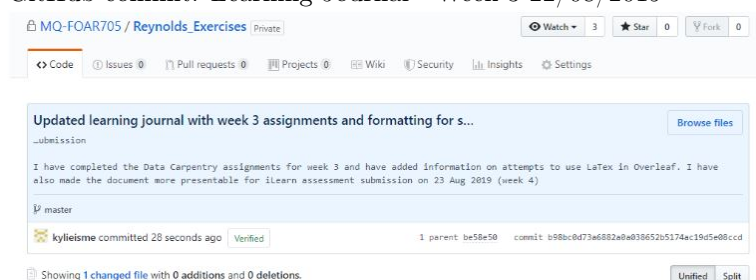
- Logged in to GitHub - MQ-FOAR705/Reynolds_Exercises
- Clicked on Upload files
- Uploaded the document from my laptop file
- In Heading text bar for commit comments, I entered the file name which was “Proof of concept - scoping exercise”
- I added a comment to describe the document (see below)
- this was successfully completed with no errors (as far as I know).



GitHub commit: Learning journal



GitHub commit: Learning Journal - Week 3 22/08/2019

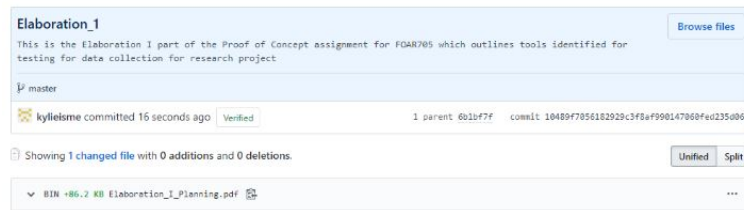


GitHub commit: Proof of Concept Scoping exercise II 24/08/2019 8:06pm



GitHub commit: Elaboration I 04/09/2019 7:54pm

I forgot to commit my PoC Elaboration I document the other day so I am doing it now since I just

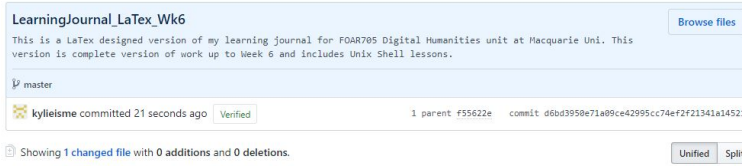


remembered. See below.

GitHub commit: Elaboration II 06/09/2019 10:06pm



GitHub commit: Learning Journal - Unix Shell 08/09/2019



3 Errors

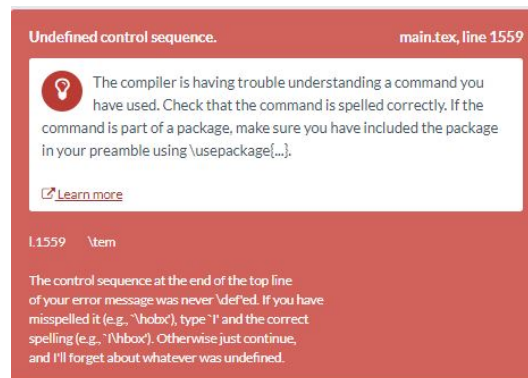
3.1 Data Carpentry

3.1.1 Data Organization in Spreadsheets for Social Scientist

3.1.2 Unix Shell

Start 14/09/2019 2:33pm

- An error has arisen - I believe it was generated by the code `$()` and round brackets directly after. See error message below:



- I tried putting `\` in front of the brackets but the error message remained.
- I noticed that below the line where `$()` was typed there was an empty `\item` command so I deleted that and the error disappeared.
- problem solved!

3.1.3 OpenRefine for Social Science Data

3.1.4 R for Social Scientists

3.2 Overleaf - LaTeX Editor

3.3 Proof of Concept

4 Data Carpentry - Data Organization in Spreadsheets for Social Scientists

14/08/2019 8:30pm

Objective: to read and complete exercises in DC Data Organization in Spreadsheets for Social Scientists (<https://datacarpentry.org/spreadsheets-socialsci/>)

4.0.1 Introduction

(<https://datacarpentry.org/spreadsheets-socialsci/00-intro/index.html>). Three spreadsheets were required for this and subsequent lessons in this section of the course. The spreadsheets were downloaded in the first week under instruction from presenters in the of the Macquarie University (MQ) FOAR705 class. Therefore, I did not need to download the spreadsheets for the lesson today. The spreadsheets are downloadable from:

<https://datacarpentry.org/spreadsheets-socialsci/setup.html>

- The three spreadsheets that were downloaded are:
 - SAFI_clean.csv
 - SAFI_messy.xlsx
 - SAFI_dates.xlsx
- they were saved in the “Data Carpentries for Social Scientists/Spreadsheets” folder on my laptop desktop.
- This task was straight forward and the instructions in this lesson were easy to follow and to execute.
- I did not have any problems or come across any errors.

Reflection: Maybe it would have been best practice to download the files again today as they may have been updated since last week.

Solution: I re-downloaded the spreadsheets and replaced the old ones in the same folder.

The introduction outlined what will be covered in this and subsequent sections of the Data Organization in Spreadsheets for Social Scientists lesson. The main objective of this lesson is to teach us best practice in the organization of data and data entry practices, and the critical importance of the data we collect for our research projects.

- To complete tasks for this lesson, a spreadsheet software is needed. The program that I will be using is excel.
- This section of the lesson lists some of the problems that can arise with spreadsheets. It spells out the usefulness of spreadsheets for data entry purposes, table and figure creation, statistics etc. However, it also warns of some of the dangers associated with these types of processes which could cause problems with research data analysis further on, some of these are:
 - White space

- Highlighting of cells
- Merging of cells
- Applying calculations

Reflection: I often do all of the things listed above, and now that I understand that this can cause big problems, I will be more aware and mindful of when and how I use these features of the spreadsheet software.

Solution: Learning more about good data processing and organisational skills by continuing through the next lessons will help. And discontinuing bad practices and habits which may compromise my research data.

Exercises:

- How many people have used spreadsheets in their research?
 - I assume most researchers have used spreadsheets in their research to organise and analyse data. I have used spreadsheets to record data for research projects in uni and at work.
- How many people have accidentally done something that made them frustrated or sad?
 - I have accidentally deleted data from an original data source, and I have also organised my data in sloppy ways (as I have learnt today) which has caused problems for me and has most likely caused problems for my co-workers in the analysis stages.

4.0.2 Formatting data tables in Spreadsheets

15/08/2019 8:30 pm - 9:45 pm

Objective: To successfully complete the Data Carpentry for social scientists: Formatting data tables in Spreadsheets module

<https://datacarpentry.org/spreadsheets-socialsci/01-format-data/index.html>

This module is teaching us how to avoid making mistakes when working with data and how to avoid creating challenges when formatting of spreadsheets for data collection, storage and analysis.

Common mistakes are:

- adding contextual elements to the data spreadsheet - e.g. margin notes, the layout/design of the spreadsheet, and using fields to convey meaning to contextualise data (e.g. notes, colours etc., I think)
 - may confuse the computer and mess with the validity of calculations and analysis of data

Important:

- good formatting of spreadsheets from the very beginning
- Data organization is the foundation of your research project
- Data entry/analysis tip - automate conversion of files

To keep track of analyses:

- Keep the original data set and create a new file for any changes/cleaning up of data/analysis
- Track data clean-up/analysis/ step-by-step (experiment note stylz) in a new file or tab (to keep it together)

Structuring data - cardinal rules of using spreadsheet programs for data

1. Put all your variables in columns
2. Put each observation in its own row.
3. Don't combine multiple pieces of information in one cell.
4. Leave the raw data raw - don't change it!
5. Export the cleaned data to a text-based format like CSV (comma-separated values) format.
 - QUESTION: how do you do that??

Exercises:

Take a messy version of the SAFI data and describe how we would clean it up.

1. Download the messy data

- Messy data downloaded previously (see notes and links for journal entry 14/08/2019)
- original files saved on laptop desktop Data Carpentry for Social Scientists, Spreadsheets file.

2. Open the data in a spreadsheet program.

- I went onto the laptop desktop, double clicked on Data Carpentry for Social Scientists folder, and then Spreadsheets folder, and double clicked on the messy excel file
- The file auto opened in Excel.
- I clicked on enable editing button at the top of the page to access the file
- as this is my raw data file, I then clicked on “File” and then “Save as” and created a copy which I named - SAFI_messy_cleaned_15082019 in the same folder as the original
- I moved the raw data SAFI_messy file into a newly created folder in the Spreadsheets folder, which I named SAFI_RAW_data for safe keeping

3. Notice that there are two tabs. Two researchers conducted the interviews, one in Mozambique and the other in Tanzania. They both structured their data tables in a different way. Now, you’re the person in charge of this project and you want to be able to start analyzing the data.

- I have surveyed both tabs to look at the data and analyse what I am working with
- My first thoughts are that it would be impossible to analyse the data in this format and that I would need to merge the data and organise it in a way that which makes sense and according to the good data organization principles learnt in this module and in the introduction module

4. Identify what is wrong with this spreadsheet. Discuss the steps you would need to take to clean up the two tabs, and to put them all together in one spreadsheet.

See table below

What is wrong	Steps to be taken
Data separate - needs to be merged	<p>Merging of data:</p> <ul style="list-style-type: none"> • Create new tab • create columns for each variable • each observation must have own line • Carefully and correctly enter the data into the new sheet • Double-check data entry against raw data source <p>Possible problem: not sure how to deal with the Key_IDs as they are, they should be different for each entry and there will be duplicates if using the original data from both locations)</p> <p>Possible solution: (need to find out if it is ok to do this??) renumber the key_ID and make a note in a notes section which will be created in a separate tab to log what has been done, step-by-step</p>
Contextual information and formatting that may cause analysis problems - e.g. White space, colour coding, notes in random places	<ul style="list-style-type: none"> • In new spreadsheet cells only to be used for recording data and column names • no merging of cells • add a column for "notes" - any notes from observation can be put there • Any random notes or colour coding etc. can be logged in a separate notes tab or file • Merged cells and white space for design purposes are a no-no
The title of the study and "2017 Data Collection" Info	This information can go in the Notes tab.
Blank A column	<p><u>Organise</u> and format according to Tidy Data principles:</p> <ol style="list-style-type: none"> 1. Each variable forms a column. 2. Each observation forms a row. 3. Each type of observational unit forms a table (<u>Wickham</u>, 2014, p. 4). <p>The first column should start with Key_ID to identify the observation line and all other columns should continue from there with no missed columns in between</p>

Metadata exercise:

What is not immediately obvious to me about this data?

- I opened the previously downloaded copy of SAFI_clean spreadsheet saved on laptop desktop Data Carpentries for Social Scientists, Spreadsheets folder and noticed that some of the variable column titles are not self-explanatory.
- There are also no notes connected to the document and no references to any other files which could explain the meanings of the variable titles, values, or the questions that were asked which could give me some context information which I would need when thinking about during the analysis of the data.

What questions would I need to know the answers to in order to analyze and interpret this data?

- I would need to know exactly what the column headers stand for
- I would need to know what was observed, how it was observed, the type of data that was required for each response, how data was coded, description of each variable, value meanings, etc.

4.0.3 Formatting problems

16/08/2019 9:30 am - 9:55am

Objective: To successfully complete the Data Carpentry for social scientists: Formatting problems module <https://datacarpentry.org/spreadsheets-socialsci/02-common-mistakes/index.html>

Common spreadsheet formatting problems/solutions:

- Using multiple tables - Don't do it - it confuses the computer, which may not be able to read associations between the tables
- Using multiple tabs - Data split between tabs - don't do it - computer cannot make connections
- Not filling in zeros - empty spaces means no data collected (null) - zero are entered if questions has been answered and the value is zero
- Using problematic null values - Blanks (most applications) and NA (for R) are best to use - check for different software
- Using formatting to convey information - do not highlight or leave blank rows in data - don't use design features in data tables, e.g. merging cells etc.
- Using formatting to make the data sheet look pretty
- Placing comments or units in cells - do not write notes in cells with other units of data - create a new notes field
- Entering more than one piece of information in a cell - each cell should only have on unit of data - create new columns instead
- Using problematic field names - Yes: wall_type OR WallType - No: wall type - no spaces and clear naming instead of abbreviations that may not make sense down the track
- Using special characters in data - Word, formatting and fancy non-standard characters can mess up data - be careful when copying and pasting data and do not use.

Reflection: The common formatting problems that are listed and demonstrated on the SAFI_messy spreadsheet are many. When I picked up problems for the last module exercises, I seemed to have missed quite a few of the mistakes that were.

Solution: create a checklist of problems to be mindful of when formatting spreadsheets for research data and to check when I have been supplied data from another person.

4.0.4 Dates as data

21/08/2019 9:30pm

Objective: To successfully complete the Data Carpentry for social scientists: Dates as data module
<https://datacarpentry.org/spreadsheets-socialsci/03-dates-as-data/index.html>

General notes:

- Single column for dates is not best practice
 - may cause problems for computer
 - allow ambiguity to creep into your data
- functions differ between spreadsheet programs - may not be compatible if exporting

Dates as integers (whole numbers not fractions of):

- Excel counts date from December 31, 1899 and stores numbers as serial numbers (e.g. 41834 same as 07/14/14)
- must check dates for accuracy when exporting data from Excel (this link in Data carpentry lesson doesn't work)
- Formula for adding data in Excel e.g. =B2(cell with date)+90(days) = new date

Regional date formatting:

- different countries format dates differently (e.g. US month/day backwards) - this is why should not put dates in same column
- put in separate columns - 3 pieces of data (month, day, and year)
 - avoids confusion
 - better for comparison

Exercise:

- download spreadsheet - SAFI_dates.xlsx <https://ndownloader.figshare.com/files/11502827> (already completed this task in first week of class) file saved in DataCarpentry/Spreadsheets folder
- I opened the spreadsheet and pressed the enable editing button at the top of the page
- I created three new columns (B, C, D) on the right side of the date column (column A where dates are formatted as 17/11/2016)
- I selected columns B, C, and D, and formatted the columns to numbers
- I named column B interview_day, C interview_month, D interview_year
- In the first row of Column B i entered =DAY(A2) and pressed enter, it showed up as 17.00
- I selected Column B C D and reduced the decimal places to zero

- In the first row of Column C I entered =MONTH(A2) and pressed enter and it showed up as 11
- In the first row of Column D I entered =YEAR(A2) and it showed up as 2016
- Default year exercise - in Cell A2 I changed the date to 17/11 and the year changed to 2019 (default year is the current one)

	A	B	C	D	
1	interview_date	interview_day	interview_month	interview_year	year
2	17/11/2019	17	11	2019	
3	17/11/2016	17	11	2016	
4	16/11/2016	16	11	2016	
5	16/12/2016	16	12	2016	
6	21/11/2016	21	11	2016	
7	21/11/2016	21	11	2016	
8	21/11/2016	21	11	2016	

- Historical data (beware):
 - pre 31 Dec 1899 Excel will leave as is
 - Be very careful when mixing before and after dates

4.0.5 Quality assurance

22/08/2019 8:37pm

Objective: To successfully complete the Data Carpentry for social scientists: Quality assurance module <https://datacarpentry.org/spreadsheets-socialsci/04-quality-assurance/index.html>

Validate data on input:

- one type of data per column
- it is possible to add data restriction for values in columns and cells
- In Excel can apply data validations to cells which can raise errors to alert us and data is not entered
- In Excel, can also add validation criteria to cells with data already entered - data not removed and it is flagged by a triangle in top left of cell
- Excel validation rules available here: <https://support.office.com/en-us/article/Apply-data-validation-to-cells-29FECBCC-D1B9-42C1-9D76-EFF3CE5F7249>

Restricting data to a numeric range

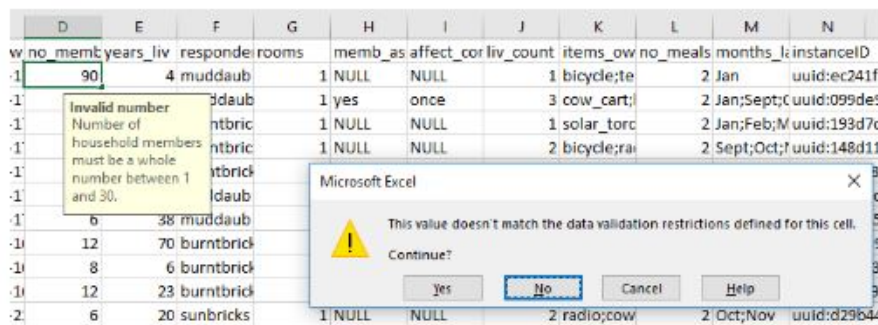
- I opened the clean SAFI file downloaded in previous lesson
- Checked column D as per instructions - numbers in this column should be integers and amount should be limited (in this instance 30 is the limit)
- I selected the Column D (no_members)
- I located the Data tab and selected Data tools and then Data validation and a pop up appeared
- I pressed the drop-down menu arrow on the right of the 1st drop down box
- I selected Whole number and entered Min. 1 and Max 30 as the range
- to test it out I entered 90 into the cell D2 and a pop up populated (see below)

	D	E	F	G	H	I	J	K	L	M	N	O
	no_members	years_liv	response	rooms	memb_as	affect_cor	liv_count	items_ow	no_meals	months_liv	instanceID	
1	90	4	muddaub	1	NULL	NULL	1	bicycle;te	2	Jan	uuid:ec241f2c-06	
1	7	9	muddaub	1	yes	once	3	cow_cart;l	2	Jan;Sept;C	uuid:099de9c9-34	
1	10	15	burntbric	1	NULL	NULL	1	solar_torc	2	Jan;Feb;M	uuid:193d7daf-95	
1	7	6	burntbric	1	NULL	NULL	2	bicycle;rai	2	Sept;Oct;f	uuid:148d1105-7	
1	7	40	burntbrick									B11-9E
1	3	3	muddaub									c91-c8
1	6	38	muddaub									58d-5f
1	12	70	burntbrick									930-7f
1	8	6	burntbrick									3d2-b:
1	12	23	burntbrick									9bc-d:
2	6	20	sunbricks	1	NULL	NULL	2	radio;cow	2	Oct;Nov	uuid:d29b44e3-3	

- I pressed cancel to get rid of the error message
- selected Column D again and went to the data validation tool again
- in the pop up I selected Input message tab - types in the message to instruct user why the error had populated, and the rule applied to guide them in how to enter the correct data (1-30 number range)
- I then selected the Error alert tab and changed the Style to Warning.
- I then tried to re-enter the incorrect number in cell D2 to see if it was working (see below)

Exercise:

- Apply a new data validation rule to a numeric column
- Using the same clean SAFI file, I selected column G (rooms)
- I located the Data tab and selected Data tools and then Data validation and a pop up appeared
- I pressed the drop down menu arrow on the Settings drop down and I selected Whole number and entered Min. 1 and Max 7 (because there is one entry where the number is 8 and I want to test to see if the triangle appears to let me know that the data is incorrect)



- I then selected the Input message tab entered Invalid number in the title and “Number of rooms must be a whole number between 1 and 7.” in the Input message
- I selected the Error Alert tab and changed the Style setting from Stop to Warning and pressed OK
- to test it out I entered 90 into the cell G2 and a pop up populated - the warning populated, and I pressed cancel and the original data remained
- I checked the cell where the number 8 was (G73) however no little triangle populated to tell me there was a validation error (hmmm..... Y?), however, there is an annoying little pink square (see below) which was sitting over the data at the top of the column which I can’t get rid of, is that trying to tell me there is something wrong or is that just what happens when you add a validation rule to the column.

	A	B	C	D	E	F	G	H
58	67	Chirodzo	2016-11-10	5	31	burntbrick	2	no
59	68	Chirodzo	2016-11-10	8	52	burntbrick	1	yes
70	69	Chirodzo	2016-11-10	4	12	muddaub	1	yes
71	70	Chirodzo	2016-11-10	8	25	burntbrick	2	no
72	71	Ruaca	2016-11-10	6	14	burntbrick	1	yes
73	127	Chirodzo	2016-11-10	4	18	burntbrick	8	NULL
74	133	Ruaca	2016-11-20	5	25	burntbrick	2	no

- To test the annoying box theory, I am going to change the 8 to a 3 to see if the box disappears
- The box did not disappear. Which is annoying but handy as I am unlikely to add the incorrect data with it in my face.
- Aha - Just found the tip at the bottom of Data Carpentry lesson that says if it is an existing spreadsheet with data already in it and it breaks the new validation rule it will not be flagged - that answers my question. Note to self: read the whole lesson so you don’t waste time on unnecessary experiments when the answer is already there. Though I am more likely to remember that now, so all is well in the world.

4.0.6 Exporting data

22/08/2019 9:54pm

Objective: To successfully complete the Data Carpentry for social scientists: Exporting data module <https://datacarpentry.org/spreadsheets-socialsci/05-exporting-data/index.html>

- Don't save files in .xls or .xlsx
 - may cause problems in the future or may not be able to open in other software or other versions of Excel
 - Some data repositories may not accept this file format
- Save files in universal, open and static format like TSV or CSV
 - “tab-delimited (tab separated values or TSV) or comma-delimited (comma separated values or CSV”
 - can use nearly any software to open and view
 - importation easy for other uses
 - may give you a warning when saving but that's ok

Exercise:

- Click File and Save As
- in File format section select CSV
- Check save location and click Save
- Success!

Note: check for commas before saving in CSV or enclose data including commas in quotation marks.

5 Data carpentry - The Unix Shell

5.1 Introducing the Shell

28/08/2019 8:30pm

Objective: To successfully complete module 1 of the course with the intention of learning what a shell is and what it does - <http://swcarpentry.github.io/shell-novice/01-intro/index.html>

- GUI - graphical user interface - most common way that general computer users interact with computers - mouse, keyboard etc - good for general and easy stuff - not good for larger more complicated tasks - would take too long
- CLI - command-line interface - faster and good for automation of repetitive tasks -uses new language, combinations of commands and parameters.
- REPL - read-evaluate-print loop - “The heart of a command-line interface” - process:
 1. type command and press Enter
 2. shell reads command
 3. evaluates/executes
 4. output of command printed
 5. loops around and waits for next command

5.1.1 The Shell

- The Boss:
 - tells programs (simple [stand alone tasks AKA commands] and complicated [eg. modelling software]) what to do
 - it doesn’t do the work itself.
- The most popular shell is Bash - “default shell on most modern implementations of Unix [whatever that is? better find out] and in most packages that provide Unix-like tools for Windows”

Activity:

- I opened the GitHub Bash shell that was downloaded in class because I am curious and need to know if the instructions work.
- There was dollar sign the prompt as the instructions indicated - most commonly the prompt is dollar sign but can be different
- the lesson says that the prompt “ls” will list the contents of the current directory that I am in
- it listed a whole bunch of things in different colours - Do the colours mean anything? looks like they could be different types of files, some show what is on my PC desktop and others (different colour) show eg. application data, which seem like background files
- A new prompt is showing at the end, and example of REPL, waiting for a new command from me (the Big Boss, lol).

- The next instruction says if you accidentally type in the wrong command eg. “ks” it will tell you command not found
- I tested this and this was correct
- Question? - so what happens if the command that you incorrectly types was an actual command that might stuff up what you’re doing?
 - Ask in lesson on Friday (turns out I needed to find out earlier see below)

5.1.2 Flexibility and automation

- Script = Sequences of commands
- Shell language:
 - enables us to combine tools to make pipelines to automate and handle large amounts of data
 - makes it easier to interact with remote and super computers
 - is necessary to interact with scientific data, clusters and cloud computing systems

Key Points

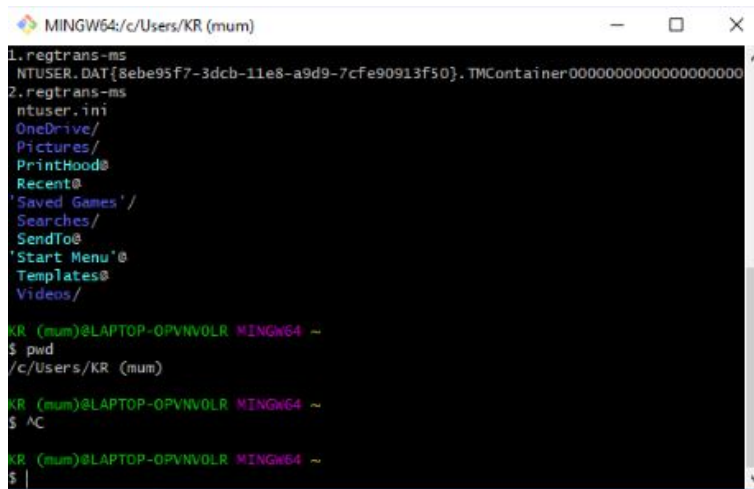
- A shell is a program whose primary purpose is to read commands and run other programs.
- The shell's main advantages are its high action-to-keystroke ratio, its support for automating repetitive tasks, and its capacity to access networked machines.
- The shell's main disadvantages are its primarily textual nature and how cryptic its commands and operation can be.

5.2 Navigating Files and Directories

28/08/2019 10:30pm ish

Objective: To successfully complete the Navigating files and directories module and to learn about files and directories, absolute paths and relative paths, how to navigate my computer using the shell, and tab completion.

- File system - the operating system responsible for managing the files and directories (AKA folders)
- Commands to manage file system:
 - create
 - inspect - pwd (print working directory) (where am I?) current working directory
 - rename
 - delete
- Crap! - tried to copy what the response was from Bash and it did a weird command and I am not sure what it means and if it will be bad for my computer - ahh help! see below:



```
MINGW64/c/Users/KR (mum)
1. regtrans-ms
NTUSER.DAT{8e95f7-3dcb-11e8-a9d9-7cfe90913f50}.TMContainer000000000000000000
2. regtrans-ms
ntuser.ini
OneDrive/
Pictures/
PrintHood$
Recent$
'Saved Games' /
Searches/
SendTo$
'Start Menu'$
Templates$
Videos/

KR (mum)@LAPTOP-OPVNV0LR MINGW64 ~
$ pwd
/c/Users/KR (mum)

KR (mum)@LAPTOP-OPVNV0LR MINGW64 ~
$ ^C

KR (mum)@LAPTOP-OPVNV0LR MINGW64 ~
$ |
```

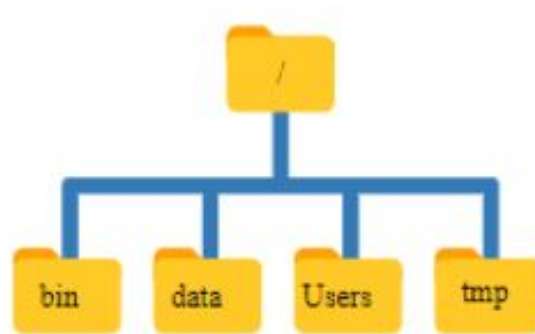
I am too scared to use my computer now that I have made some sort of command (Ĉ) on GitHub Bash, I don't know what I commanded it to do, so I have put a note up on Slack and Googled to see if I can find an answer to what I should do next. Just in case I need to do something to fix things. Will have to finish tomorrow instead. I shut down Bash - nothing seems to be going wrong at the moment - will wait and see tomorrow.

29/08/2019 10:26am

- Outcome of enquiry: Sheri commented on Slack that there doesn't seem to be an imminent problem and Brian confirmed this, this morning. Phew! Back to the exercise then.
- Resolution: Turns out Ctrl C (Ĉ) means clear which is great because I didn't stuff anything up and because I now know how to clear a command. we are going to discuss in class.
- Note to self: don't use GUI shortcuts (eg. Ctrl C to copy) in the Shell, the commands mean something different.

Lesson continued:

- root directory - holds everything else
- leading / command refers to the root directory
- /bin - root directory "/" + "bin" (other directories) - some built in programs are stored here
- /data - root directory "/" + "data" (misc. data files)
- /users - root directory "/" + "users" (user's personal directories)
- /tmp - temporary files
- 2 meaning for /
 - leading / - root directory



useful image to demonstrate above

- if used inside a name it is a separator
- commands, options, arguments and parameters
 - ls - eg. of command - lists (prints) names of files in directory I am in -F - eg. of option [aka Switch or Flag],
 - * starts with “-“ or “- -“ changes behavior of command -F adds a marker to output of file/directory to tell you what type of file it is:
 - / - directory
 - @ - link
 - * - executable
 - / - Arguments - tells command what to operate on (which file/directory)
 - Parameters - options/arguments sometimes referred to as parameters
 - Commands can have more than one option/argument
 - Command doesn't always require argument/option
 - each part of the parameter must have a space in between (ls -F)
 - Capitals are different from lowercase instructions
- I tried the / -F command in Bash and the marker showed up at the end of the file name(/ at end is subdirectory) - eg. Desktop/ and Cookies@
- no classification symbol means - plain files
- I tried typing in ls -F / and a list of files populated which was smaller than the “ls” only command

5.2.1 Getting Help

- ls - -help - more info on how to use help command
- man ls - (man = manual)
- I used the “ls – help” which worked in bringing up the list of options - VERY HANDY

- “man ls” - did not work for me
 - error message reads - “bash: man: command not found”

Help page navigation - helpful image

To navigate through the `man` pages, you may use `↑` and `↓` to move line-by-line, or try `B` and `Spacebar` to skip up and down by a full page. To search for a character or word in the `man` pages, use `/` followed by the character or word you are searching for. Sometimes a search will result in multiple hits. If so, you can move between hits using `N` (for moving forward) and `Shift` + `N` (for moving backward).
To quit the `man` pages, press `Q`.

- conducted a google search for “unix man page” and came up with a user guide - <https://acadix.biz/Unix-guide/unix-guide.pdf> which might come in handy

Exercise: Exploring ls flags:

- what happens when using `ls -l`
 - I am pretty sure that it did the same as `ls -F` - it listed the files/directories with the markers eg. `/`, `@` at the end.
- what happens when using `ls -h` - it did the same thing as `ls -l` and `ls -F`. I cant see a difference
- Problem - the solution says that they should have made the list human readable and showed file sizes but mine did not do that
- Action: follow-up Friday.

omg this is a long lesson - break time 29/08/2019 - 12:32pm

29/08/2019

Exercise: Listing Recursively and By Time

Question: In what order does `ls -R -t` display things?

- I typed in `ls -R` at `$` prompt and a whole bunch of data that means nothing to me started being listed, it is taking a very long time and I think it might be listing every single file on my laptop - it finished after about 3 minutes - the files were listed in Alphabetical order i think
- I typed in `ls -t` at `$` prompt and a short list populated similar to when i typed in `ls` but in a different order - the Unix shell lesson says they are in time of last update order but I am unable to tell if that is true because there are no time stamps in the list
- `$ ls F Desktop` - this worked showing files on my desktop
- `$ ls -F Desktop/data-shell` - this did not show up anything for me (see below)
- why?
- `cd` - change directory


```
KR (mum)@LAPTOP-OPVNVOLR MINGW64 ~
$ ls -F Desktop
'Data Carpentry for Social Scientists/' 'My EndNote Library.xml'
data-shell/ 'PLAYERTEK Sync.lnk'*
data-shell.zip SCHOOL/
desktop.ini 'Spreadsheets - Shortcut.lnk'*
MRES/ STUFF/
'My EndNote Library.txt'

KR (mum)@LAPTOP-OPVNVOLR MINGW64 ~
$ ls -F Desktop/data-shell
data-shell/

KR (mum)@LAPTOP-OPVNVOLR MINGW64 ~
$ |
```

- i cant continue with this exercise because last one didnt work not sure if I should try it until i figure what happened in last step. wait I think it did work it is just there is only one file and doesn't look the example in lesson but that is probably because the sample probably had more files connected to data-shell file.
- \$ cd Desktop - shows that we are in the Desktop (in yellow) directory on line above prompt
- \$ cd data-shell - shows that we are in the Desktop/data-shell (in yellow) directory on line above prompt
- \$ cd data - PROBLEM - says that no such file or directory exists (see image below)

```
KR (mum)@LAPTOP-OPVNVOLR MINGW64 ~
$ cd Desktop

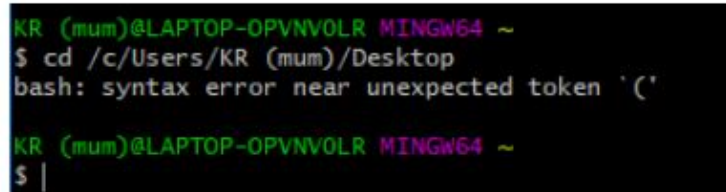
KR (mum)@LAPTOP-OPVNVOLR MINGW64 ~/Desktop
$ cd data-shell

KR (mum)@LAPTOP-OPVNVOLR MINGW64 ~/Desktop/data-shell
$ cd data
bash: cd: data: No such file or directory

KR (mum)@LAPTOP-OPVNVOLR MINGW64 ~/Desktop/data-shell
$ |
```

- typed in pwd after prompt - shows that I am still in Desktop/data-shell directory
- Problem: why? this is different from what the lesson says I am going to go back to the Unix shell set up page to make sure I downloaded file properly.
- Solution: Instead of going back to set up page I first checked the file on my desktop, turns out I have an additional data-shell directory within my data-shell directory so I had to do an extra cd data-shell and then cd data
- ls -F - shows files as they are meant to be showing - SUCCESS!
- To go backwards up the directory tree type cd .. - Success apparently a special directory is missing (how would you know that if you didn't know what files are in the directory) - to get to special directory use show all -a

- \$ ls -F -a - shows special hidden files (which begin with . or .. - “The prefix “.” is used to prevent these configuration files from cluttering the terminal when a standard ls command is used”).)
- Orthogonality -
- tried “cd” on its own and it took me back to my user directory (the home directory)
- to go to specific path (absolute path) enter in full directory name from root directory eg. /blah/blah2/blah3
- I entered pwd to find out where I was and it showed up as - /c/Users/KR (mum)/Desktop but it showed an error (see below)



```

KR (mum)@LAPTOP-OPVNVOLR MINGW64 ~
$ cd /c/Users/KR (mum)/Desktop
bash: syntax error near unexpected token `('
KR (mum)@LAPTOP-OPVNVOLR MINGW64 ~
$ |

```

- Problem: Not doing what lesson says it should - possible cause the brackets in my KR (mum) directory - does it not like brackets in directory names?? need to find out and change this somehow on my laptop files
- Solution: unsure for now. Slack??

5.2.2 Shortcuts

- (tilde) - at beginning of the path it means the current user’s home directory
- cd - dash after cd takes you to back to last directory - different from cd .. which takes you up the directory tree.

Exercise: Absolute vs Relative Paths

Q1. Starting from /Users/amanda/data, which of the following commands could Amanda use to navigate to her home directory, which is /Users/amanda?

- Answer: cd .. CORRECT

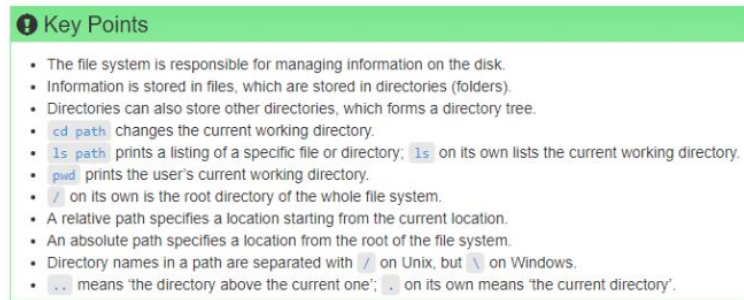
Q2. Using the file system diagram below, if pwd displays /Users/thing, what will ls - F ../backup display?

- Answer: ../backup: No such file or directory INCORRECT
- Correct answer: Yes: ../backup/ refers to /Users/backup/.

Q3. Using the filesystem diagram below, if pwd displays /Users/backup, and -r tells ls to display things in reverse order, what command(s) will result in the following output:

- pnas_sub/ pnas_final/ original/

- Answer: `ls -r -F /Users/backup` CORRECT
- tab completion - To auto populate file/directory names
- handy image to remember shortcuts and commands



Finish - 29/08/2019 2:57pm

5.3 Working with files and directories

01/09/2019 11:20am

Objective: To successfully complete the Working with files and directories module and to learn about how to create a directory hierarchy, files in that hierarchy using an editor or by copying and renaming existing files, and to delete, copy and move files and/or directories.

5.3.1 Creating directories (folders):

- see where we are
 - For this lesson I was instructed to go to the Data-shell directory on my desktop in shell (Bash)
 - I entered `cd Desktop` and Enter to change directories from user directory
 - I entered `cd data-shell` and Enter to go to Data-shell directory
 - type "`ls -F`" so I can see what is in the directory
- create a directory called thesis
 - `mkdir` (make directory) `thesis` (relative path - no `/` before so directory is located in current directory)
 - typed `ls -F` to check that directory was created
 - and `thesis/` shows up in directory and checked windows file explorer and thesis folder is there also
 - Success!
- Note: Good naming conventions for files

- NO spaces, NO dashes at beginning or end (saw end bit on a different site) because computer read - (dash) as a command for options.
- DO use letters, numbers, dash (surrounded by other text), fullstops, and underscores (_).
- if referring to filenames with spaces surround with quotation marks eg. “KR MUM” directory on my laptop.
- Create a text file in thesis directory
 - changed into thesis directory by typing `cd thesis`
 - typed in nano `draft.txt` as instructed - on windows can use notepad as well
 - had to try it so typed `^X` as is instructed at the bottom of the window.
 - it just typed as text though - will have to ask in Slack if I dont find out in modules - nevermind just found out `^` = `Ctrl` on keyboard
 - it populated what looks to be a window to type text (see below)



- I wrote out some text and pressed `Ctrl` and `O` as instructed and it populated a File name to write to option which was the `draft.txt` file that we created earlier
- I pressed `Enter`
- It populated a note at the bottom telling me that it had written two lines - that mean saved
- I pressed `Ctrl X` as instructed, however, it didnt send me back to the shell as it said that it would, instead it asked about a buffer, options were Yes No or Cancel.
- I said No and it took me back to the shell.
- I wrote `ls` to see if the file was in the directory and it was
- Success!

Exercise: Creating files in a different way

- typed command `touch my_file.txt` as instructed
- the files are listed with what looks like a character count
- Question: When might you want to create a file this way?
 - Answer: to see which files are the most recent and that have content ??
 - Solution: Some programs do not generate output files themselves, but instead require that empty files have already been generated. When the program is run, it searches for an existing file to populate with its output. The `touch` command allows you to efficiently generate a blank text file to be used by such programs.
- typed `cd -` before ending session - having a break

01/09/2019 8:37pm

5.3.2 What's in a name?

filename extension - eg. `.txt`, `.pdf`, `.png` etc. - to help identify the type of file

5.3.3 Moving files and directories:

- went to data-shell direcoty by typing in `cd /Desktop/data-shell`
- to rename file `mv thesis/draft.txt thesis/quotes.txt` rename - `mv` means move but because it is in the same file it just renames
- I pressed enter and it went to prompt
- I typed in `ls thesis` and it listed the `quotes.txt` file
- however there was also another file in there `my_file.txt` also and I dont know where that came from??

5.3.4 Moving to the current folder

- I was going to attempt this exercise, however, I could not find any of these directory or file names in the directory we downloaded for these exercises so I am assuming that it is a test question of what to do.
- It asks how to move the files to another directory and to swap the file names over
- the question wasnt clear to me and the instructions were a little confusing so I went to the solution which is reminding me that you can use `..` (parent directory) to go back a level in the directory heirarchy and `.` for the current directory. In the answer it says to move the files to the current directory which was the raw directory however in the solution it wants the files in the analyzed directory as different names
- I don't understand what it is trying to tell me to do - it is very unclear

5.3.5 Copying files and directories

- cp - copy file
 - this exercise doesn't give instructions about the directory we are supposed to be starting from - I am making an assumption that it is the one we were in from the last exercise
 - this exercise did not work for me as the command "cp quotes.txt thesis/quotations.txt" did not work for me
 - Copying the directory to thesis_backup worked by following the commands suggested from the data-shell directory. the command was - \$ cp -r thesis thesis_backup
 - I checked to see if the last step worked by typing \$ ls thesis thesis_backup as suggested and this worked The file had been created

5.3.6 Renaming files exercise:

- Q: 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: statistics.txt
- After creating and saving this file you realize you misspelled the filename! You want to correct the mistake, which of the following commands could you use to do so?
 - cp statistics.txt statistics.txt
 - mv statistics.txt statistics.txt
 - mv statistics.txt
 - cp statistics.txt
- A: 2 mv statistics.txt statistics.tx
- Result: Yes, this would work to rename the file. Success!

Exercise - Moving and copying

Answer is 4 - proteins-saved.dat bumper got it wrong it was the directory name recombine

5.3.7 Removing files and directories

- rm - to remove files
- I removed the quotes.txt from the directory by typing in rm quotes.txt
- I then tested to see if it was gone by listing the files in the directory by typing ls
- the quotes.txt file was gone
- Note: deleted files cannot be recovered - there is no trash/recycle bin

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

- Answer: It says no such file/directory exists

- this is not what was supposed to happen - it says that using -i is supposed to make it prompt you before going ahead and deleting.
- I am not sure why my file has disappeared.
- I made sure I was in the right directory which I wasn't
- I changed directory
- Did the command again and it worked
- I tried to delete the thesis file by typing `rm thesis` but it did not work
- I tried `rm -r (recursive) thesis`
- I typed `ls` to list files and it worked

5.3.8 Operations with multiple files and directories

Note: copy or move several files at once - done by providing a list of individual filenames, or specifying a naming pattern using wildcards. eg.

- I entered `cd data` to change directory
- I typed `mkdir backup` as directed to create a new directory
- It went to prompt which means everything went ok
- I typed `cp amino-acids.txt animals.txt backup/`
- the outcome was that two new files were created in the new backup directory

what does `cp` do when given 3+ files names?

- I typed `ls -F` as directed and the tow files as listed above showed up
- `cp amino-acids.txt animals.txt morse.txt` and pressed enter
- it came back with an error saying that `morse.txt` is not a directory

5.3.9 Using wildcards for accessing multiple files at once

- `*` - wildcard - matches 0 or more characters
- `?` - wildcard - matches one character
- When the shell sees a wildcard, it expands the wildcard to create a list of matching filenames before running the command

List filenames matching a pattern:

- Answer: 3 correct
- Problem: I am starting to forget what some of the commands do
- Solution: I have started a new spreadsheet in my One Drive so that I can keep track of the code and of what they do.

5.3.10 Organising Directories and Files

Exercise: move two files from one file to another

- Answer: `mv *.dat analyzed/`
- Solution: answer is correct!

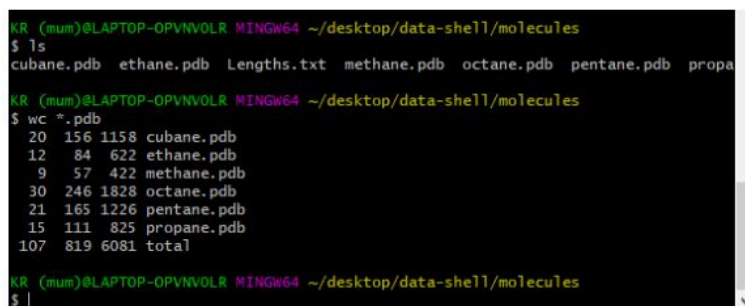
5.4 Pipes and Filters

Objective: to successfully complete the Unix Shell: Pipes and Filters module of this lesson. See here: <http://swcarpentry.github.io/shell-novice/04-pipefilter/index.html>

In this module I will learn to redirect a command's output to a file, process a file instead of keyboard input using redirection, construct command pipelines with two or more stages, explain what usually happens if a program or pipeline isn't given any input to process, explain Unix's 'small pieces, loosely joined' philosophy.

WC - Counts the number of lines, words, and characters in a file

- I went into the molecules file and typed `wc *.pdb` to see the word count of all the files with `pdb` as the file extension and it listed all of the files in the directory (they all ended with `.pdb`) with their counts as stated above.



```
RR (mum)@LAPTOP-OPVNV0LR MINGW64 ~/desktop/data-shell/molecules
$ ls
cubane.pdb  ethane.pdb  Lengths.txt  methane.pdb  octane.pdb  pentane.pdb  propa
RR (mum)@LAPTOP-OPVNV0LR MINGW64 ~/desktop/data-shell/molecules
$ wc *.pdb
 20  156 1158 cubane.pdb
 12   84  622 ethane.pdb
  9   57  422 methane.pdb
 30  246 1828 octane.pdb
 21  165 1226 pentane.pdb
 15  111  825 propane.pdb
107  819 6081 total
RR (mum)@LAPTOP-OPVNV0LR MINGW64 ~/desktop/data-shell/molecules
$ |
```

- I typed `wc -l` as directed and it was supposed to list the line numbers for the files but this did not happen and it did not go to the prompt. I am not sure what happened.
- I realised that I had forgotten the file extension and pressed `Ctrl C` to escape as we were instructed in our last lesson at uni.
- Turns out this mistake is the next exercise so bonus I just saved some time. Yay!
- I will now do the correct command and list the items with their line numbers by typing
- `wc -l *.pdb`
- and success!
- replace `wc -l` with `wc -w` for number of words and `wc -c` for characters

- `$ wc -l *.pdb > lengths.txt` - this command creates a txt file in the molecules directory -
- after typing the command I checked to see if it had been created by typing `ls` to list the files and
- it was there.
- to see what is in the file I typed `cat lengths.txt`
- It showed the contents
- I typed this in and pressed enter and it showed me the page
- the lesson indicated that I could type `b` to go back but this did not work for me and it came up with the `^` symbol instead - I am not sure why
- I pressed `q` for quit - this worked
- Reflection - I am assuming `b` did not work because there were no other pages

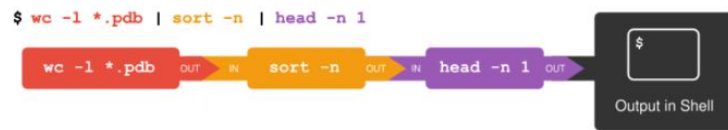
5.4.1 Sort -n

- Sort - sorts - `n` (numerically) doesn't work on its own have to specify file
- it shows up in numerical order
- to put sorted data in a file I typed `sort -n lengths.txt > sorted-lengths.txt`
- I typed `ls` to see if it had worked
- then typed `head -n 1 sorted-lengths.txt` to see the first line in the file
- to see if changing the number showed 2 lines I typed in `head -n 2 sorted-lengths.txt`
- and it did show the top 2 lines in numerical order
- I will try 4 to see if it keeps working with higher numbers
- it did.

Note: do not redirect files to the same file name - could delete the original file

5.4.2 what does `»` mean

- Echo - command - prints the text you type after it on the screen
- testing echo and operators `>` and `»`:
 - Echo hello `>` testfile01.txt - it adds two hello's if you do the command twice
 - Echo hello `»` testfile02.txt
 - the difference between the operators - one `>` there was only one hello first hello is overwritten by the 2nd command - two `»` there were two hellos in the file it adds more info, it appends the string to the file.



5.4.3 The pipeline

- shift + backslash = | (the pipe)
- The pipe filters through one command to the next (see image below)
- Piping commands together exercise - answer: 4 - Correct
- -d - is typed before the character which will be the delimiter (eg. -d , - the comma is now the delimiter, I think meaning whatever comes before it, is cut out of the equation) ?? not sure though
- uniq - filters out adjacent matching lines in a file

5.5 Loops

Objective: to successfully complete the Unix Shell: Loops module. See here:

<http://swcarpentry.github.io/shell-novice/05-loop/index.html>

In this module I will learn to write a loop, learn about variables within a loop, identify executed commands, and re-run executed commands.

5.5.1 Follow the prompt

Loops are a “programming construct which allow us to repeat a command or set of commands for each item in a list”

Example of a loop:

```
for thing in list_of_things
do
    operation_using $thing    # Indentation within the Loop is not required, but aids legibility
done
```

```
$ for filename in basilisk.dat minotaur.dat unicorn.dat
> do
>     head -n 2 $filename | tail -n 1
> done
```

```
CLASSIFICATION: basiliscus vulgaris
CLASSIFICATION: bos hominus
CLASSIFICATION: equus monoceros
```

- in creatures file to find the classification for each species (located on second line of each file) could execute `head -n 2 | tail -n 1` for each individual file
- instead better to create a loop
- I typed the command as per image above and it printed the list below it successfully
- prompt `$` changes to `>` and back to `$`
- the command `for` means repeat command
- `;` (semi-colon) can be used to separate commands on a single line
- the second `$` marks the variable in the list - tells the shell interpreter to treat the variable as a variable name and substitute its value in its place, rather than treat it as text or an external command

5.5.2 Same symbols different meanings

- `>` - overwrites OR can also be used as a prompt in a loop
- `$` - shell prompt OR can also ask shell to get value of a variable
- if they show up on screen they are prompts
- if you type yourself they are instructions to `>` redirect output or `$` get value/variable
- `$filename`, `$filename same`, `$filename different`

Variables in Loops - Exercise

- `$` for datafile in *.pdb
 - `> do`
 - `> ls *.pdb`
 - `> done`
 gives the same output on each loop - matches all files ending in .pdb and then lists (ls) them
- `$` for datafile in *.pdb
 - `> do`
 - `> ls $datafile`
 - `> done`
 lists a different file on each loop. The value of the datafile variable (`$datafile`) is listed (ls)

Limiting Sets of Files - Exercise

- `$` for filename in c*
 - `> do`
 - `> ls $filename`
 - `> done` Answer: 4 Only cubane.pdb is listed - Correct
- `$` for filename in *c*
 - `> do`
 - `> ls $filename`
 - `> done` Answer: 4 The files cubane.pdb and octane.pdb will be listed - Correct

Saving to a File in a Loop - Part One - Exercise

- \$ for alkanes in *.pdb
 - > do
 - > echo \$alkanes
 - > cat \$alkanes > alkanes.pdb
 - > done

Answer: none of the above - incorrect

Correct answer: The text from each file in turn gets written to the alkanes.pdb file. However, the file gets overwritten on each loop, so the final content of alkanes.pdb is the text from the propane.pdb file.

Saving to a File in a Loop - Part Two - Exercise

- \$ for datafile in *.pdb
 - > do
 - > cat \$datafile » all.pdb
 - > done

Answer: 3 All of the text from cubane.pdb, ethane.pdb, methane.pdb, octane.pdb, pentane.pdb and propane.pdb would be concatenated and saved to a file called all.pdb - Correct

more complicated loop:

- \$ for filename in *.dat
 - > do
 - > echo \$filename
 - > head -n 100 \$filename | tail -n 20
 - > done
- *.dat - all files with extension .dat listed for processing
- loop body - echo (print) \$filename (prints the name of files)
- the head and tail part are a bit confusing - apparently it selects the 81-100 line of each file to print - not sure.

5.5.3 Spaces in Names

If an element eg. file name has a space in it we need to put quotation marks around it. eg. Kylie Reynolds.pdf would be "Kylie Reynolds.pdf" - this is because spaces are used to separate commands.

Try to avoid using spaces in filenames as it can cause problems'

```
$ for filename in "red dragon.dat" "purple unicorn.dat"
> do
>   head -n 100 "$filename" | tail -n 20
> done
```

```

KR (mum)@LAPTOP-OPVNVOLR MINGW64 ~/desktop/data-shell/molecules
$ for filename in red dragon.dat purple unicorn.dat
> do
> head -n 100 $filename | tail -n 20
> done
head: cannot open 'red' for reading: No such file or directory
head: cannot open 'dragon.dat' for reading: No such file or directory
head: cannot open 'purple' for reading: No such file or directory
head: cannot open 'unicorn.dat' for reading: No such file or directory

```

- I removed the quotes around \$filename in the loop above to see the effect - see below:

Echo (prints) good for checking what is happening in the loop Can put whole loop on one line by separating with semi-colons (;) between the actions

Beginning and End

- Ctrl a - moves to beginning of line
- Ctrl e - moves to end

Those Who Know History Can Choose to Repeat It

- the history command to get a list of the last few hundred commands that have been executed - eg. - \$ history | tail -n 5 - to look up last five commands
- to repeat a command you can type in !(and number of the command line on the list)

Other History Commands

- Ctrl R - 'reverse-i-search' - history search mode - most recent command in your history that matches the text you enter next
- !! - retrieves the last command entered - preceding command
- !! retrieves the immediately preceding command
- less !\$ to look at the file

Doing a Dry Run - Exercise

- to preview the commands - Echo
- Answer: Version 1 - incorrect - version 2

Key Points

- A `for` loop repeats commands once for every thing in a list.
- Every `for` loop needs a variable to refer to the thing it is currently operating on.
- Use `$name` to expand a variable (i.e., get its value). `${name}` can also be used.
- Do not use spaces, quotes, or wildcard characters such as `"` or `?` in filenames, as it complicates variable expansion.
- Give files consistent names that are easy to match with wildcard patterns to make it easy to select them for looping.
- Use the up-arrow key to scroll up through previous commands to edit and repeat them.
- Use `Ctrl-R` to search through the previously entered commands.
- Use `history` to display recent commands, and `!number` to repeat a command by number.

There are two more modules to go to finish this unit. I have been unable to complete them in time for handing in of Learning Journal for Unix Shell submission. I will be catching up on these modules this week. Each module is taking me hours to get through and understand. I may need to catch up on break with all of my work.

Place markers for modules to be done are below:

5.6 Shell Scripts

09/09/2019 9:00pm

Objective: To successfully complete the Shell script module, where I will learn to write a shell script for fixed set of files and set of files of defined by user on command line, run a shell script from a command line, create a pipeline that includes shell scripts.

shell script - a bunch of commands saved in a file (small programs)

- to create a new file in molecules directory I first went to the changed directory and then typed in nano middle.sh and pressed enter
- this took me to a blank screen, the text editor, where I can write the text for the document.
- I wrote out "head -n 15 octane.pdb | tail -n 5" as instructed
- I pressed Ctrl O to save the text and the file name middle.sh showed up and I pressed enter to keep that file name
- I then pressed Ctrl X to exit and to go back to the shell prompt
- The text entered in the file is not being run as we will be running it as part of bunch of commands, I think.
- I typed "ls" after the prompt to check if the file had been created and it was there
- to run the shell (bunch of commands) that the file contains we type in - bash filename.sh
- the information printed on the screen according to the pipeline commands that had been entered in the middle.sh file.

```
MINGW64; c:/Users/KR (mum)/desktop/data-shell/molecules
$ cd molecules
KR (mum)@LAPTOP-OPVNV0LR MINGW64 ~/desktop/data-shell/molecules
$ nano middle.sh
KR (mum)@LAPTOP-OPVNV0LR MINGW64 ~/desktop/data-shell/molecules
$ ls
cubane.pdb  methane.pdb  pentane.pdb  testfile01.txt
ethane.pdb  middle.sh    propane.pdb  testfile02.txt
Lengths.txt octane.pdb   sorted-lengths.txt testfile03.txt
KR (mum)@LAPTOP-OPVNV0LR MINGW64 ~/desktop/data-shell/molecules
$ bash middle.sh
ATOM   9  H      1      -4.502   0.681   0.785   1.00   0.00
ATOM  10  H      1      -5.254  -0.243  -0.537   1.00   0.00
ATOM  11  H      1      -4.357   1.252  -0.895   1.00   0.00
ATOM  12  H      1      -3.009  -0.741  -1.467   1.00   0.00
ATOM  13  H      1      -3.172  -1.337   0.206   1.00   0.00
KR (mum)@LAPTOP-OPVNV0LR MINGW64 ~/desktop/data-shell/molecules
$
KR (mum)@LAPTOP-OPVNV0LR MINGW64 ~/desktop/data-shell/molecules
$ |
```

5.6.1 Text vs. Whatever

- Use a plain text editor or save as plain text
- to edit file created above, middle.sh
- I typed in nano middle.sh as instructed
- I then replaced the text with "head -n 15 "\$1" | tail -n 5" as instructed
- Saved by pressing Ctrl O, Enter for filename and Ctrl X to close the file
- I then typed in bash middle.sh octane.pdb and pressed enter but it came up with an error
- i double checked that there were no errors in the command and realised that I had put a space after octane before putting in the file extension
- I re-entered the command and pressed enter
- and the correct information was generated.

```
KR (mum)@LAPTOP-OPVNV0LR MINGW64 ~/desktop/data-shell/molecules
$ nano middle.sh
KR (mum)@LAPTOP-OPVNV0LR MINGW64 ~/desktop/data-shell/molecules
$ bash middle.sh octane. pdb
head: cannot open 'octane.' for reading: No such file or directory
KR (mum)@LAPTOP-OPVNV0LR MINGW64 ~/desktop/data-shell/molecules
$ bash middle.sh octane.pdb
ATOM   9  H      1      -4.502   0.681   0.785   1.00   0.00
ATOM  10  H      1      -5.254  -0.243  -0.537   1.00   0.00
ATOM  11  H      1      -4.357   1.252  -0.895   1.00   0.00
ATOM  12  H      1      -3.009  -0.741  -1.467   1.00   0.00
ATOM  13  H      1      -3.172  -1.337   0.206   1.00   0.00
KR (mum)@LAPTOP-OPVNV0LR MINGW64 ~/desktop/data-shell/molecules
$ |
```

10/09/2019 8:51pm

5.6.2 Double-Quotes Around Arguments

- This lesson says that using double quotation marks around the \$1 because the file name may contain spaces and this will cancel that out.
- in this lesson we are using \$2 and \$3 variables to pass to head and tail
- I opened the file middle.sh to edit it by typing "nano middle.sh"
- I then replaced the text that was there with "head "\$2" "\$1" | tail "\$3"
- I saved the file (Ctrl O) and exited (Ctrl X)
- and then ran the command by typing "bash middle.sh pentane.pdb 15 5" and a list of lines from the file printed out in the shell
- I typed the command "bash middle.sh pentane.pdb 20 15" and it printed out some other lines
- Reflection: I am failing to understand how you know which lines you would need printed out in a random document and how this is going to help me down the track. I don't seem to understand the point in these exercises. I hope it becomes clearer at some stage of this course. I also hope it is useful at some stage. I am trusting that it will all become clear.
- we can make comments by entering # at the beginning
- The comments do not make the purpose of these exercises clearer to me but I am going to continue on.
- I typed "\$ nano middle.sh" and entered the # comments in the file
- I saved and exited
- The # comments lesson is handy to know for later on so this is good
- it is good for helping other people and myself later on understand the script - also important to note is to change the comments when updating script so that it still makes sense
- now we are going to process many files in a pipeline - all files ending .pdb by typing "wc -l *.pdb | sort -n"
- it came back with an error, see below:


```

KR (num)@LAPTOP-OPVNVOLR MINGW64 ~/desktop/data-shell/molecules
$ bash middle.sh pentane.pdb 20 5
ATOM      14  H          1      -1.259      1.430      0.112      1.00      0.00
ATOM      15  H          1      -2.608     -0.407      1.130      1.00      0.00
ATOM      16  H          1      -2.540     -1.303     -0.404      1.00      0.00
ATOM      17  H          1      -3.393      0.254     -0.321      1.00      0.00
TER         18          1
KR (num)@LAPTOP-OPVNVOLR MINGW64 ~/desktop/data-shell/molecules
$ nano middle.sh
KR (num)@LAPTOP-OPVNVOLR MINGW64 ~/desktop/data-shell/molecules
$ wc -l *.pdb | sort -n
wc: unknown option -- 1
Try 'wc --help' for more information.
KR (num)@LAPTOP-OPVNVOLR MINGW64 ~/desktop/data-shell/molecules
$ |

```

- the shell says to type "wc --help" for more info so I did
- It printed text to tell me about wc command however, I still don't know what went wrong. See handy info for "wc" command below

```

KR (num)@LAPTOP-OPVNVOLR MINGW64 ~/desktop/data-shell/molecules
$ wc --help
Usage: wc [OPTION]... [FILE]...
or: wc [OPTION]... --files0-from=F
Print newline, word, and byte counts for each FILE, and a total line if
more than one FILE is specified. A word is a non-zero-length sequence of
characters delimited by white space.

With no FILE, or when FILE is -, read standard input.

The options below may be used to select which counts are printed, always in
the following order: newline, word, character, byte, maximum line length.
-c, --bytes      print the byte counts
-m, --chars      print the character counts
-l, --lines      print the newline counts
--files0-from=F  read input from the files specified by
                  NUL-terminated names in file F;
                  If F is - then read names from standard input
-L, --max-line-length  print the maximum display width
-w, --words      print the word counts
--help          display this help and exit
--version       output version information and exit

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Full documentation <https://www.gnu.org/software/coreutils/wc>
or available locally via: info '(coreutils) wc invocation'
KR (num)@LAPTOP-OPVNVOLR MINGW64 ~/desktop/data-shell/molecules
$ |

```

- I think the error may have populated because it didn't say which directory I was supposed to be in and the one that I am in from the previous part of the exercise may not contain any .pdb files. I will check by listing the files and checking
- turns out that isnt the case I will need to find out what went wrong at some other stage as it did not work
- just figured out that the l is a letter not the number 1. Woops!
- trying again with the correct script and success!
- "\$@" - to do all files - can't use \$1 or \$2 if we dont know how many files there are
- creating a new file in text editor -
- I typed - nano sorted.sh
- filled in the text with comments describing the script and I typed in


```

# Sort files by their length.
# Usage: bash sorted.sh one_or_more_filenames
wc -l "$@" | sort -n

```

- I then ran the command
\$ bash sorted.sh *.pdb ../creatures/*.dat
- this listed a few lines but says that there were 596 total

Finally finished that section 10.15pm Reflection: tonight I planned to try and finish this module and the next however this section of the module took over an hour to complete and there are still 7 subsection in this module and then however many in the final Unix module. I am going to try and get through another one and I am hoping that it is shorter and doesn't take so long. I am wondering whether or not it takes everyone this long or just me. I'm going to start keeping a record of start and finish times, not the short excercises though.

5.6.3 List Unique Species exercise

I am unable to do this exercise as I cannot remember what -d means and I dont have time to go searching for it - I will have to come back to this one on my holidays - AKA, catch up time.

- The next section tells me to enter the following commands \$ history | tail -n 5 > redo-figure-3.sh - handy for saving and creating something over again later on
- I typed it in and it is now saved in the file redo-figure-3
- I checked the directory by writing the ls command and it was there
- I also checked in the text editor by writing nano redo-figure-3.sh, however the script is not printed in there.
- does this mean that it wasn't done correctly or am I jumping ahead of the exercise
- looks like I haven't this lesson is making no sense to me, maybe there was supposed to have been something done in another lesson. Or this lesson has skipped something. or maybe it was just an example of something - this is unclear.
- moving on again

5.6.4 Why Record Commands in the History Before Running Them?

- I didnt know why
- in case it creashes - to look back and see why - there would be no record of it otherwise

finished 10/09/2019 10:45pm

5.6.5 Variables in Shell Scripts exercise

Answer: 4 an error Solution: 2 incorrect

5.6.6 Find the Longest File With a Given Extension exercise

I cant remember how to do the coding for the excercise and I don't have time to go back through the many pages of notes to find out how - I will need to come back to this exercise.<https://www.overleaf.com/project/5d52a1e0a7824073968b36bc>

5.6.7 Script Reading Comprehension exercise

For this question, consider the data-shell/molecules directory once again. This contains a number of .pdb files in addition to any other files you may have created. Explain what each of the following three scripts would do when run as `bash script1.sh *.pdb`, `bash script2.sh *.pdb`, and `bash script3.sh *.pdb` respectively.

Finish 10/09/2019 11:15pm

Start 12/09/2019 9:40am

```
# Script 1
echo *.*
```

Answer: this would print out a list of all files in the directory

Solution: "Script 1 would print out a list of all files containing a dot in their name. The arguments passed to the script are not actually used anywhere in the script." - not sure if my answer was correct

```
# Script 2 for filename in $1 $2 $3 do cat $filename done
```

Answer: prints a specific set of lines from each file in the directory

Solution: "Script 2 would print the contents of the first 3 files with a .pdb file extension. \$1, \$2, and \$3 refer to the first, second, and third argument respectively." - I don't think I got this one correct.

```
# Script 3
echo $@.pdb
```

Answer: prints a list of all files in the directory which use the file extension .pdb

Solution: "Script 3 would print all the arguments to the script (i.e. all the .pdb files), followed by .pdb. \$@ refers to all the arguments given to a shell script." - i think I got this one right but not sure

5.6.8 Debugging Scripts exercise

Suppose you have saved the following script in a file called do-errors.sh in Nelle's north-pacific-gyre/2012-07-03 directory:

```
# Calculate stats for data files.
for datafile in "$@"
do
echo $datafile
bash goostats $datafile stats-$datafile
done
```

When you run it:

```
$ bash do-errors.sh NENE*[AB].txt
```

the output is blank. To figure out why, re-run the script using the -x option:

```
bash -x do-errors.sh NENE*[AB].txt
```

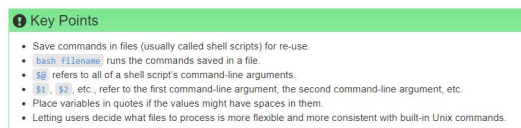
What is the output showing you? Which line is responsible for the error?

Answer: there is a space in \$datafile stats

Solution: "The -x option causes bash to run in debug mode. This prints out each command as it is run, which will help you to locate errors. In this example, we can see that echo isn't printing anything. We have made a typo in the loop variable name, and the variable datfile doesn't exist, hence returning an empty string." - I was half right

Reflection on exercises: I need to go back and find out what I didn't know. I will do this in the holidays.

Handy key points image:



Finish: 12/09/2019 10:15pm

5.7 Finding Things

Start: 12/09/2019 11:30am

Objective to successfully complete the Finding things module of the Unix shell course. In this module I will learn how to find files and directories, and how to find text lines from files that match simple patterns, use output from a command as the command-line argument(s) for another command, and to understand the difference between 'text' and 'binary' files.

- grep - grep means 'global/regular expression/print':
 - it finds and prints lines in files that match a pattern
 - very useful command-line program
 - common sequence of operations in early Unix text editors
- new exercise to show contents of a file
- I typed - cd desktop/data-shell/writing - to go to the correct directory
- I typed - cat haiku.txt - to see the contents of that file
- the contents of the file, which were 3 haiku poems. They were printed on the screen

5.7.1 Forever, or Five Years

- Finding lines that include the word "not":
 - I typed - grep not haiku.txt
 - there were three lines printed on the screen from the file
 - Reflection: the command grep specified what we wanted to find, not was the word we were looking for, and the filename was where we wanted to look

- Note: grep looks for a pattern in a case sensitive way
- to search for the pattern "The" in the same file we would write - grep The haiku.txt
- two lines were printed that both contained "The", which included a word that started with "The", which was "Thesis".
- to limit the search term put -w (will limit matches to word boundaries) directly after grep, and then the rest of the command - grep -w The haiku.txt item one line shows up that contains the word "The".
- Word boundary - includes the start and end of a line, so not just letters surrounded by spaces.
- Sometimes we don't want to search for a single word, but a phrase
 - put the phrase in quotes
 - eg. grep -w "is not" haiku.txt
 - one line showed up that contained the phrase "is not"
 - success!
- -n - numbers the lines that match
 - i typed grep -n "it" haiku.txt
 - 3 lines showed up that contained the word "it" and in front of the lines was a number (5,9 and 10).
- combine options
 - to find the lines that contain the word 'the' we can use -w and -n
 - I typed the command: grep -n -w "the" haiku.txt
 - and two lines showed up (2 and 6) and both contained the word 'the'.
- change the search behavior of 'grep' with respect to case sensitivity.
- use the option -i to make our search case-insensitive:
 - i typed the commands: grep -n -w -i "the" haiku.txt
 - it printed out 3 lines (1, 2 and 6) that contained the word 'the'. In some of the lines 'the' started with a capital letter
 - success!

Finish: 12/09/2019 12:35pm

Start: 12/09/2019 3:51pm

- -v - invert search, does not contain the word identified
- I typed the commands: grep -n -w -v "the" haiku.txt
- this brought up a bunch of lines, including the line numbers. All of the lines that were printed did not contain the word "the". Blank lines were also printed.

To find out what else 'grep' can do we can type in: grep --help - this brings up a long list of different options that can be used and tells you what the options can do. See very handy images below:

```

RR (mm)@LAPTOP-OPVNVOLR MINGW64 ~/desktop/data-shell/writing
$ grep --help
Usage: grep [OPTION]... PATTERN [FILE]...
Search for PATTERN in each FILE.
Example: grep -i 'hello world' menu.h main.c

Pattern selection and interpretation:
-E, --extended-regexp  PATTERN is an extended regular expression
-F, --fixed-strings    PATTERN is a set of newline-separated strings
-G, --basic-regexp    PATTERN is a basic regular expression (default)
-P, --perl-regexp     PATTERN is a Perl regular expression
-e, --regexp=PATTERN  use PATTERN for matching
-f, --file=FILE       obtain PATTERN from FILE
-i, --ignore-case     ignore case distinctions
-w, --word-regexp     force PATTERN to match only whole words
-x, --line-regexp     force PATTERN to match only whole lines
-z, --null-data       a data line ends in 0 byte, not newline

Miscellaneous:
-s, --no-messages    suppress error messages
-v, --invert-match    select non-matching lines
-V, --version        display version information and exit
-h, --help           display this help text and exit

```

```

Output control:
-m, --max-count=NUM  stop after NUM selected lines
-b, --byte-offset    print the byte offset with output lines
-n, --line-number    print line number with output lines
--line-buffered      flush output on every line
-H, --with-filename  print file name with output lines
-h, --no-filename    suppress the file name prefix on output
--label=LABEL        use LABEL as the standard input file name prefix
-o, --only-matching  show only the part of a line matching PATTERN
-q, --quiet, --silent suppress all normal output
--binary-files=TYPE  assume that binary files are TYPE;
                     TYPE is 'binary', 'text', or 'without-match'
                     equivalent to --binary-files=text
-I, --text           equivalent to --binary-files=without-match
-d, --directories=ACTION how to handle directories;
                     ACTION is 'read', 'recurse', or 'skip'
-D, --devices=ACTION  how to handle devices, FIFOs and sockets;
                     ACTION is 'read' or 'skip'
-r, --recursive      like --directories=recurse
-R, --dereference-recursive likewise, but follow all symlinks
--include=FILE_PATTERN search only files that match FILE_PATTERN
--exclude=FILE_PATTERN skip files and directories matching FILE_PATTERN
--exclude-from=FILE    skip files matching any file pattern from FILE
--exclude-dir=PATTERN  directories that match PATTERN will be skipped.
-l, --files-without-match print only names of FILES with no selected lines
-l, --files-with-matches print only names of FILES with selected lines
-c, --count          print only a count of selected lines per FILE
-T, --initial-tab     make tabs line up (if needed)
-z, --null           print 0 byte after FILE name

Context control:
-B, --before-context=NUM print NUM lines of leading context
-A, --after-context=NUM  print NUM lines of trailing context
-C, --context=NUM       print NUM lines of output context
-UM, --color[=WHEN], --colour[=WHEN] use markers to highlight the matching strings;
                                     WHEN is 'always', 'never', or 'auto'
-U, --binary           do not strip CR characters at EOL (MSDOS/Windows)

When FILE is '-', read standard input. With no FILE, read '.' if
recursive, '-' otherwise. With fewer than two FILES, assume -h.
Exit status is 0 if any line is selected, 1 otherwise;
if any error occurs and -q is not given, the exit status is 2.

Report bugs to: bug-grep@gnu.org
GNU grep home page: <http://www.gnu.org/software/grep/>
General help using GNU software: <http://www.gnu.org/gethelp/>

```

5.7.2 Using grep exercise

Which command would result in the following output:

and the presence of absence:

```
grep "of" haiku.txt
```

```
grep -E "of" haiku.txt
```

```
grep -w "of" haiku.txt
```

```
grep -i "of" haiku.txt
```

Answer: `grep -w "of" haiku.txt`

Solution: Correct!

Finish: 4:15pm

5.7.3 Wildcards

Start: 12/09/2019 7:30pm

Link for more complicated - regular expression (the re in grep):

<https://v4.software-carpentry.org/regexp/index.html>

Regular expression:

- "A regular expression is a pattern that matches sets of related character strings. While there are patterns that regular expressions cannot match, they are the power tool most programmers turn to when they need to extract information from legacy text files.
 - Regular expressions are written as character strings (which makes the notation somewhat clumsy).
 - Alphanumeric characters match themselves.
 - Use *, +, and ? for repetition.
 - Use character sets, character set shortcuts, and | to match alternatives.
 - Use parentheses to group things and to extract information from matches.
 - Use the regular expression library to find all matches, replace strings, and perform other operations". (retrieved from: <https://v4.software-carpentry.org/regexp/index.html>)

Example:

- find lines that have an 'o' in the second position
- typed: `grep -E '^o' haiku.txt`
- three lines of text printed on the screen
- all of the lines contained o's in the second position.
- Note:
 - the -E option and by putting the pattern in quotes, prevents the shell from trying to interpret it - which is what we want the ^ - anchors the match to the beginning of line
 - The . matches a single character
- so ^.o - is telling the shell from the beginning of the line skip one letter then look for an o.

Finish: 12/09/2019 9:00pm (this includes time spent in Overleaf section with LaTeX issues)

5.7.4 Tracking a Species exercise

Start: 12/09/2019 9:53pm

Put in the right order:

```
cut -d : -f 2
>
|
grep -w $1 -r $2
|
$1.txt
cut -d , -f 1,3
```

Answer: I don't know but will have a try, actually no I won't because I have absolutely no idea.

Solution:

```
grep -w $1 -r $2 | cut -d : -f 2 | cut -d , -f 1,3 > $1.txt
```

5.7.5 Little Women exercise

This exercise asks me to write a loop command to find out which sister in the book little women has the most mentions. I have absolutely no idea so am going to look at the solution and then write the script to see what happens - hopefully I will learn something

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

This did a wordcount for each of the sisters name throughout the book. Very handy and something that I should learn. I will need to revise.

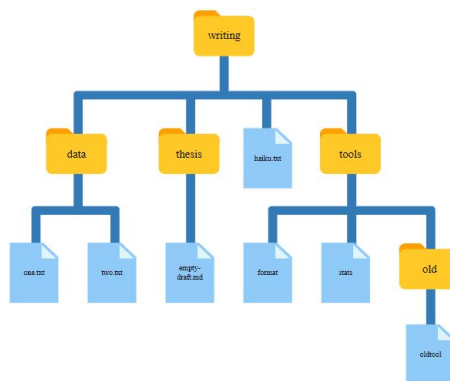
Note:

- **grep** - finds lines in files
- **find** - finds files

Finish: 12/09/2019 10:33pm - Bed time!

5.7.6 Listing vs. Finding

Start: 14/09/2019 8:57am



- Grep finds lines in files
- Find find files (directory tree above used for examples)
- type command 'find .' after the prompt

- this lists all of the files that are underneath the directory that we are in (data-shell/writing).
- the '.' means current working directory
- find options to filter the output
 - -type d - things that are directories
 - -type f - listing of all the files
 - -name *.txt - by name
- combining tools
 - put the find command inside \$() (the shell executes what is in the brackets first) - (Side Note: see error in above Error section)
 - eg. \$ wc -l \$(find . -name '*.txt') - (Note: this is the expanded command that is running with example - \$ wc -l ./data/one.txt ./data/LittleWomen.txt ./data/two.txt ./haiku.txt)
 - this command list all files in the working directory with the line count for each file at the beginning of the line
- find and grep can be used together
 - The first finds files that match a pattern; the second looks for lines inside those files
 - Eg. \$ grep "FE" \$(find .. -name '*.pdb')
 - this did not work for me, see error below (see Error section for reflection/solution):
 - * The error seems to be created by incorrect input, specifically the second \$ was missing. once the commands were entered correctly it worked successfully.

```

KR (mum)@LAPTOP-OPVNVOLR MINGW64 ~/desktop/data-shell/writing
$ grep "FE" $(find .. -name '*.pdb')
bash: syntax error near unexpected token `('
KR (mum)@LAPTOP-OPVNVOLR MINGW64 ~/desktop/data-shell/writing
$ |

```

Finish: Start: 14/09/2019 10:13am

5.7.7 Matching and Subtracting exercise

-v to 'grep' inverts pattern matching - only lines which do not match the pattern are printed

Answer: 4. None of the above - incorrect Solution: 1

5.7.8 Binary Files

Finding data stored as images, in a database, or other format Might be better to use a more modern programming language for this.

5.7.9 find Pipeline Reading Comprehension exercise

- Question: Write a short explanatory comment for the following shell script:

```
wc -l $(find . -name '*.dat') | sort -n
```
- Answer: find all files in the working directory ending with the extension .dat and show the amount of lines in the file, and sort this list by number.
- Solution:
 - Find all files with a .dat extension recursively from the current directory
 - Count the number of lines each of these files contains
 - Sort the output from step 2. numerically

5.7.10 Finding Files With Different Properties exercise

- criteria (AKA tests) - to locate files with specific attributes, such as creation time, size, permissions, or ownership
- man find - to explore these
- this did not work for me - TO DO - FIND OUT WHAT THE CRITERIA ARE

Keypoints:

Key Points

- `find` finds files with specific properties that match patterns.
- `grep` selects lines in files that match patterns.
- `--help` is an option supported by many bash commands, and programs that can be run from within Bash, to display more information on how to use these commands or programs.
- `man <command>` displays the manual page for a given command.
- `${<command>}` inserts a command's output in place.

THE END!!!!!!

6 Data carpentry - OpenRefine for Social Science Data

6.1 Introduction

6.1.1 Motivations for the OpenRefine Lesson

What is OpenRefine useful for?

- set of tools - to identify and amend the messy data.
- track changes to dataset
- actions are reversible
- actions are repeatable
- data cleansing is quick
- powerful - makes complex tasks easy

6.1.2 Features

- GitHub open source - <https://github.com/OpenRefine/OpenRefine>
- help community
- works with large datasets (100K)

Key Points

- OpenRefine is a powerful, free and open source tool that can be used for data cleaning.
- OpenRefine will automatically track any steps allowing you to backtrack as needed and providing a record of all work done

6.2 Working with OpenRefine

Installation of the software was completed in FOAR705 class on campus. I saved a shortcut to open the program on my laptop desktop, otherwise the file can be found in my desktop directory: MRES/2019/Semester2/FOAR705_Digital_Humanities. The files that will be being used for the OpenRefine lessons can be found on my desktop.

Objective: To successfully complete the module stated above

(<https://datacarpentry.org/openrefine-socialsci/02-working-with-openrefine/index.html>), which will teach me to:

- "Create an OpenRefine project from CSV file.
- Understand potential problems with file headers.
- Use facets to summarize data from a column.
- Use clustering to detect possible typing errors.
- Understand that there are different clustering algorithms which might give different results.
- Employ drop-downs to remove white spaces from cells.
- Manipulate data using previous steps with undo/redo".

6.2.1 Creating a new OpenRefine project

In the left margin there are three options to start with OpenRefine. They are:

1. Create Project,
 2. Open Project, or
 3. Import Project
- I clicked the Create Project button and selected Get data from this computer
 - I pressed the Choose Files button mid-left of the screen and then opened the SAFI_messy_openrefine.csv from the Data Carpentry directory on my desktop.
 - The file opened in the top half of the screen and a tick box options screen opened up on the bottom half of the screen. Some of the boxes were already ticked and some were empty
 - everything looked ok so I clicked Create Project button in the top right hand corner

6.2.2 Using Facets

- Facets will help you get an overview of the data and can help with creating consistency across the dataset by filtering the data down to a subset where changes can be made in bulk.
- Text facets:
 - groups all the identical text values in a column
 - lists each value with the number of records
 - info appears in the left panel

Using facets to find possible errors in data entry:

- I used the scroll bar at the bottom of the screen to move to the "village" column of the data sheet.
- I clicked on the arrow drop down at the top of the column and selected Facet and then Text facet from the menus
- a list populated on the panel on the left hand side of the page which had a list of the different values that are in the villages column, some of which were very similar, and should have, most likely, been the same.
- when I selected the count option in the panel it ordered the values with the most entries from highest to lowest - this would be a good way to see, if there had been misspellings in some entries which would be the correct one to use.
- Note if I place my mouse over one of the values, on the right, an edit option appears
- the edit button can be used to fix errors immediately
- OpenRefine will prompt you to fix other errors with the same value.

6.2.3 Exercise

Question 1: Using faceting, find out how many different interview_date values there are in the survey results.

Answer: There were 19 different values - I used the text facet option

Question 2: Is the column formatted as Text or Date?

Answer: I used the text facet option - the number and timeline facet options did not work.

To convert interview_date column to use timeline display:

- choose Edit cells
- select Common transforms
- and choose To date

Question 3: During what period were most of the interviews collected?

Answer: 2016-11-16

6.2.4 More on Facets

Other types of facets:

- Numeric facets:
 - display graphs instead of lists of values
 - ‘drag and drop’ controls - to set a start and end range
- Timeline facets (for dates)
- Custom facets:
 - range of different types of facets
 - Word facet - breaks down text into words
 - Duplicates facet - binary results - ‘true’ or ‘false’.
 - Text length facet - creates a numeric facet based on the length (number of characters) of the text in each row for the selected column - useful for spotting incorrect or unusual data
 - Facet by blank - a binary facet of ‘true’ or ‘false’. Rows appear in the ‘true’ facet if they have no data present - useful for finding rows missing data.
- Scatterplot facets - display graphs instead of lists of values

6.2.5 Using clustering to detect possible typing errors

clustering:

- means - used to find groups of different values that are similar "representations of the same thing".
- a very powerful tool for cleaning datasets typing errors e.g. misspelled/mistyped entries.
- I went back to the village column, clicked on the arrow drop down menu and selected Facets then Text facet
- in the panel on the left 8 values populated
- I clicked on teh Cluster button at the top right of the panel
- this opened a new window (see below)

Cluster & Edit column "village"

This feature helps you find groups of different cell values that might be alternative representations of the same thing. For example, the two strings "New York" and "new york" are very likely to refer to the same concept and just have capitalization differences, and "Gödel" and "Godel" probably refer to the same person. [Find out more...](#)

Method key collision Keying Function fingerprint 8 clusters found

Cluster Size	Row Count	Values in Cluster	Merge?	New Cell Value
1	1	• Chiridozo (1 rows)	<input type="checkbox"/>	<input type="text" value="Chiridozo"/>
1	3	• Ruaca-Nhamuenda (3 rows)	<input type="checkbox"/>	<input type="text" value="Ruaca-Nhamuenda"/>
1	43	• Ruaca (43 rows)	<input type="checkbox"/>	<input type="text" value="Ruaca"/>
1	2	• Ruca (2 rows)	<input type="checkbox"/>	<input type="text" value="Ruca"/>
1	1	• 49 (1 rows)	<input type="checkbox"/>	<input type="text" value="49"/>
1	1	• Ruaca - Nhamuenda (1 rows)	<input type="checkbox"/>	<input type="text" value="Ruaca - Nhamuenda"/>
1	37	• Chirodzo (37 rows)	<input type="checkbox"/>	<input type="text" value="Chirodzo"/>
1	43	• God (43 rows)	<input type="checkbox"/>	<input type="text" value="God"/>

Rows in Cluster

Average Length of Choices

Select All Unselect All Export Clusters Merge Selected & Re-Cluster Merge Selected & Close Close

- choosing the method resulted in no clusters found
- Results for testing of Keying function:
 - Fingerprint - random single values
 - n-gram fingerprint - groups together with like values
 - megaphone3 - groups together with like values (seems more accurate)
 - cologne- phonetic - looks the same/similar to megaphone3 - differe

- the rest are similar to the others - I would suggest playing around until you get the best fit
- I decided on metphone3 from the Keying function drop-down
- I selected the two top selections which seemed to be matching the correct and misspelled values of the village names
- I then chose the Merge selected & Re-cluster button from the bottom right of the window.
- the two that I selected were now merged and the remaining values were re-ordered
- there were still two values which seemed to be the same village although one was misspelled, these did not show up together as an option, I tried to select them both and merge but this did not work
- I closed the window and went back to the facet panel on the left of the page.
- I hovered over the name that I thought was incorrect (least amount of entries) and clicked the edit option on the right
- I then changed some of the values to the correct village names and they merged themselves and a pop-up notification told me that the spreadsheet data had also been corrected.

6.2.6 Different clustering algorithms

Handy link to more info on clustering:

<https://github.com/OpenRefine/OpenRefine/wiki/Clustering-In-Depth> handy info from site:

- clustering in OpenRefine works only at the syntactic level (the character composition of the cell value) - useful to spot errors, typos, and inconsistencies, but not enough to perform "effective semantically-aware reconciliation".
- OpenRefine can use "external semantically-aware reconciliation services (such as Wikidata's) to compensate" - see more here:
<https://github.com/OpenRefine/OpenRefine/wiki/Reconciliation>

6.2.7 Transforming data

- In the items_owned column - is a list
- list values show up in square brackets and in single quotation marks
- to split the list we have to remove the brackets and quotation marks
- I clicked the arrow drop-down menu on the items_owned column
- I chose Edit cells and then Transform... options
- a pop-up window opened (see below)

Custom text transform on column items_owned

Expression Language General Refine Expression Language (GREL)

`value` No syntax error.

Preview History Starred Help

row	value	value
1.	['bicycle'; 'television'; 'solar_panel'; 'table']	['bicycle'; 'television'; 'solar_panel'; 'table']
2.	['cow_cart'; 'bicycle'; 'radio'; 'cow_plough'; 'solar_panel'; 'solar_torch'; 'table'; 'mobile_phone']	['cow_cart'; 'bicycle'; 'radio'; 'cow_plough'; 'solar_panel'; 'solar_torch'; 'table'; 'mobile_phone']
3.	['solar_torch']	['solar_torch']
4.	['bicycle'; 'radio'; 'cow_plough'; 'solar_panel'; 'mobile_phone']	['bicycle'; 'radio'; 'cow_plough'; 'solar_panel'; 'mobile_phone']
5.	['motorcycle'; 'radio'; 'cow_plough'; 'mobile_phone']	['motorcycle'; 'radio'; 'cow_plough'; 'mobile_phone']
6.	NULL	NULL

On error ☒ keep original ☐ set to blank ☐ store error ☐ Re-transform up to times until no change

- the language used in this window is called GREL which stands for General Refine Expression Language
- to remove all of the square brackets type the text:

Custom text transform on column items_owned

Expression Language General Refine Expression Language (GREL)

`value.replace("[", "")` No syntax error.

Preview History Starred Help

row	value	value.replace("[", "")
1.	['bicycle'; 'television'; 'solar_panel'; 'table']	'bicycle'; 'television'; 'solar_panel'; 'table'

- and then press OK button

Exercise:

- to remove the single quotation marks I need to repeat this process but replace the square bracket with a single quotation mark and also with the right square bracket. and the spaces.

To see common items owned:

- I went to the drop down arrow at the top of column items_owned and selected Facet
- I then selected Custom text facet...
- In the pop-up window, expression box I typed: `value.split(";")`, and pressed OK button
- in the left hand panel a list of all items populated

Exercise:

Question: Which two items are the most commonly owned? Which are the two least commonly owned?

To find out the answer to these questions I selected the "count" option at the top right of the panel which listed the items from highest to lowest.

Answer:

- most - mobile phone and radio
- least - computer and car

Exercise:

Question: Perform the same clean up steps and customized text faceting for the months_lack_food column. Which month(s) were farmers more likely to lack food?

Answer: November, October and December

Exercise:

Perform the same clean up steps for:

- months_no_water - October and September
- liv_owned - Oxen and Cows
- res_change - NULL and less_work
- no_food_mitigation columns - lab_ex_food and na

Hint: To reuse a GREL command, click the History tab and then click Reuse next to the command you would like to apply to that column.

6.2.8 Using undo and redo

Undo/Redo:

- undo/redo tab is at the top of left facets panel
- click on tab
- Select the step you want to go back to and it automatically shows up in data sheet as undone

6.2.9 Trim Leading and Trailing Whitespace

- spaces are bad
- OpenRefine has a tool to remove spaces
- I created a text facet for the values in column respondent_wall_type
- some of the values had space at the beginning or end of the word which the computer reads as a different value of the same text without spaces
- to delete the white spaces we select (from drop down on column) the Edit cells, then Common transforms, then Trim leading and trailing whitespace
- the list reduces and the values which had white space are now merged with the correct values that didn't contain any

Key Points

- OpenRefine can import a variety of file types.
- OpenRefine can be used to explore data using filters.
- Clustering in OpenRefine can help to identify different values that might mean the same thing.
- OpenRefine can transform the values of a column.

6.3 Filtering and Sorting with OpenRefine

<https://datacarpentry.org/openrefine-socialsci/03-filter-sort/index.html>

6.3.1 Filtering

To filter and work on a subset of the data:

- go to the respondent_roof_type column
- select Text filter
- a text box will open up in the left hand panel with the column name above it
- I typed "mabat" into the text box as instructed, and pressed Enter
- in the column all values beginning with "mabat" showed up - the value showed up "mabatisloping"
- at the top of the data set it shows that there are 58 out of 131 records which match
- just under that it gives you options to view the data:
 - show as rows or record
 - show 5, 10, 25, 50 rows
- I chose to view 50 rows - this shows the first 50 records on the first page
- on the right hand side of the viewing option line you can navigate between pages
- I clicked the Next page option, and the remaining 8 records were shown

Exercise:

Question 1: What roof types are selected by this procedure?

Answer: mabatisloping and mabatipitched

Question 2: How would you restrict this to only one of the roof types?

Answer: I would change the search term to "mabatis" for sloping or "mabatip" for pitched

CORRECT!

6.4 Excluding entries

Narrow the filter"

- include and/or exclude entries
- this can be done by hovering over the name when it shows up in the panel after filtering
- it gives you two option - edit or include
- leave as is to include
- or click on the include button for the ones you want to stay
- I clicked include on the mabatipitched and the include button changed to exclude
- in the column only the mabatipitched values were showing (10 showing)

6.4.1 Sort

- Data can be sorted by the column by choosing Sort on the drop down menu.
- data can be sorted by text, numbers, dates or booleans (TRUE or FALSE values).
- these can be ordered

Exercise:

Question: Sort the data by gps_Altitude. Do you think the first few entries may have incorrect altitudes?

- I found the gps_altitude column
- clicked on the drop down menu and selected Sort
- a new windo opened and I selected Number
- I left the default setting of smallest to largest and pressed OK
- The first few entries that show are coming up as 0 for altitude. I do believe that these could be incorrect

Correct!

If you press sort on the same column again it gives you different options than before. It allows you to modify the original sort, reverse the sort, or remove sort.

6.4.2 Sorting by multiple columns

- dependant on order that columns are selected
- to sort by a particular column tick check box in pop up Sort window which says "sort by this column alone"

Exercise:

In an earlier lesson the value for one of the village was 49. This is wrong - look at the GPS coordinates to decide which village the data was collected from?

- Sort on `gps_Longitude` - largest first.
- Add a sort on `gps_Latitude` as a number with the largest first.
- Using the drop down arrow on the village column, select Edit column then Move column to end - to compare village names with GPS coordinates.
- Scroll through the entries until you find village 49. Can you tell from it's GPS coordinates which village it belong to? Answer: I'm guessing Chirodzo -
- Now sort only by `interview_date` as date. Move the village column to the start of the table. Does the row where village is 49 group with one particular village? Is it the same village as when comparing GPS coordinates? Yes
- I changed the village name by selecting the village drop down, selecting facet, text facet and then by selecting Edit in the facet panel on the left - I changed the name to Chirodzo
- Correct!

6.5 Examining Numbers in OpenRefine

<https://datacarpentry.org/openrefine-socialsci/04-numbers/index.html>

6.5.1 Numbers

We can transform columns to other data types (e.g. number or date):

- by using Edit cells
- selecting Common transforms
- remove text filters from left side panel by clicking on the top left corner X button
- I also removed the sort of the date by clicking sort and remove
- I found the `years_farm` column
- clicked on the drop down arrow
- Selected Edit cells, then Common transforms, and then to Number
- the text changed colour to green and moved the numbers to the right hand side of the column

Exercise:

Transform three more columns, `no_members`, `yrs_liv`, and `buildings_in_compound`, from text to numbers. Can all columns be transformed to numbers? Try it with village for example.

I followed the steps above for the columns above - the columns that contained numbers changed to green and aligned to the right of the column - when I tried on the village column nothing changed

6.5.2 Numeric facet

We can find non-number values or blanks (data entry errors) in a column by using a Numeric facet Exercise:

- I changed the 1st cell in years farm to "abc" and I deleted the number in the 2nd cell
- I then applied the facet by number and a graph populated in the face panel on the left of the screen
- at the bottom of the panel were tick boxes which showed the number of numeric values (129), non-numeric (1) values, blank (1), and error
- I unchecked the numeric box, and the two cells with non-numeric and blank cell were left on the data set screen
- I added the correct data back in by clicking the Edit button on the right hand side of the cell and changing the data
- I then went back to the panel and re-checked the numeric box
- 2 items were now showing as non-numeric
- I then clicked on the drop down menu, selected edit cells, then Common transforms, to number to change the newly edited cells to numbers.

6.6 Using scripts

<https://datacarpentry.org/openrefine-socialsci/05-scripts/index.html>

6.6.1 How OpenRefine records what you have done

OpenRefine saves all changes made to the dataset - which are saved in the JSON (JavaScript Object Notation) format - this can be exported and applied to other data files.

6.6.2 Saving your work as a script

- to save the steps applied to this dataset go to the undo/redo tab on the left hand side panel
- Click the Extract button
- Select the commands that you want to save by checking the tickboxes
- save the copied code into a text document
- save as plain text file (.txt)
- I saved as op1.txt in OpenRefine directory

6.6.3 Importing a script to use against another dataset

- I opened a new project using the messy dataset
- I saved the project
- I clicked on the undo/redo tab at top of left panel
- I copied and pasted the JSON data that I saved earlier into the text box and pressed the Perform operations button at the bottom left of the window.
- the commands were automatically applied
- success!

6.7 Exporting and Saving Data from OpenRefine

<https://datacarpentry.org/openrefine-socialsci/06-saving/index.html>

6.7.1 Saving and Exporting a Project

Saving - OpenRefine does it automatically

Exporting - good for sending people Raw data and the steps used for cleansing

- Click the Export button in top right
- Select Export project
- this opened up a window with options Export to local or Export to Google drive, I selected Local
- the file saved as SAFI_openrefine-csv.openrefine.tar.gz in the download directory on my laptop
- to open the file I downloaded 7-zip
- I extracted the files and it created a new History directory which contained many zipped files

6.7.2 Exporting Cleaned Data

- Click Export in top right
- select the Tab-separated values (tsv) or Comma-separated values (csv) file type
- the file saved in the Download directory - can be opened in a spreadsheet or imported into R or Python

6.8 Other Resources in OpenRefine

See here for additional handy resources:

<https://datacarpentry.org/openrefine-socialsci/07-resources/index.html>

7 Overleaf

13/08/2019 10:00 pm

Create an Overleaf account - <http://overleaf.com/edu/macquarie>

I clicked on the link to the Macquarie University link on Overleaf which was posted on the FOAR705 Slack chat (instant messaging and collaboration tool) that @Brian had put up. It is suggested that assignment and learning journal submissions be in LaTeX format, and I believe this website will allow me to create this type of document. I believe that LaTeX is a typesetting design program but am not sure. ACTION: find out what it does and how to operate or create documents.

I set up my sign in using my Macquarie Uni HDR ID, an email was sent to my email account for verification. Verification was actioned, and this took me to a profile page where I added details of my department (Sociology) and the degree that I am enrolled in (Masters). I noticed that I could link the Overleaf account to my GitHub account which I did, however, I am unsure what this does or how this works. I received an email to notify me that it the GitHub account was successfully linked.

ACTION: I will need to find out about this.

- I found a button on the web page which indicated that I could start a project.
- I clicked on the button where some options came up to set up a blank document, certain other types of documents, or to select a template.
- First, I clicked on the templates link and had a quick look at what was available.
- I then decided to click back and to select a blank document. The blank page came up with some details on it. The details looked like code.
- I clicked on a PDF button on the left side of the page and it showed me a readable version of the text which was coded on the previous link.
- I went back to the code link and didn't know what to do so closed the app.

Reflection: I need to discover how to use Overleaf.

Solution: I looked up "adding text to a document in overleaf" on google and clicked on a video that showed up in the search "LaTeX video tutorial for beginners (video 1) to learn how to use the program [https://www.overleaf.com/learn/latex/LaTeX_video_tutorial_for_beginners_\(video_1\)](https://www.overleaf.com/learn/latex/LaTeX_video_tutorial_for_beginners_(video_1))

14/08/2019 10:25 pm

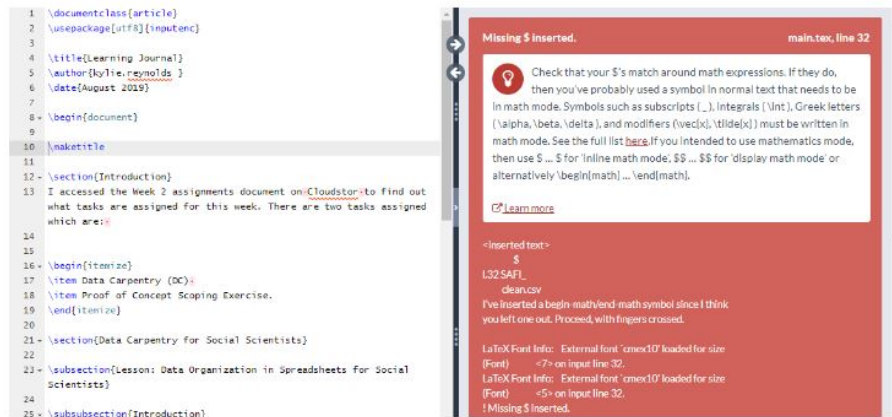
Watched video 1 and 2 about basics and adding text to LaTeX document - need to trial this process out.

Reflection: Note sure if Overleaf is called a program.

Solution: need to find out about the lingo - ask for help in Slack chat, or at consultation on Friday.

16/08/2019 12:00pm

I attempted to create a learning journal in Overleaf, and things seemed to be going well with me copying and pasting from a text document. I then recompiled the document to see what it would look like and I noticed a little note on top of the viewing panel, which I hovered over, and which said that it was a “Logs and output file”. I opened the note by clicking on it and a whole bunch of information that I could not understand showed up.)See below for an example)



This overwhelmed me so I logged off and thought maybe next task I will feel more confident, I do not know how to deal with this information. I will attempt to do the scoping exercise on Overleaf instead.

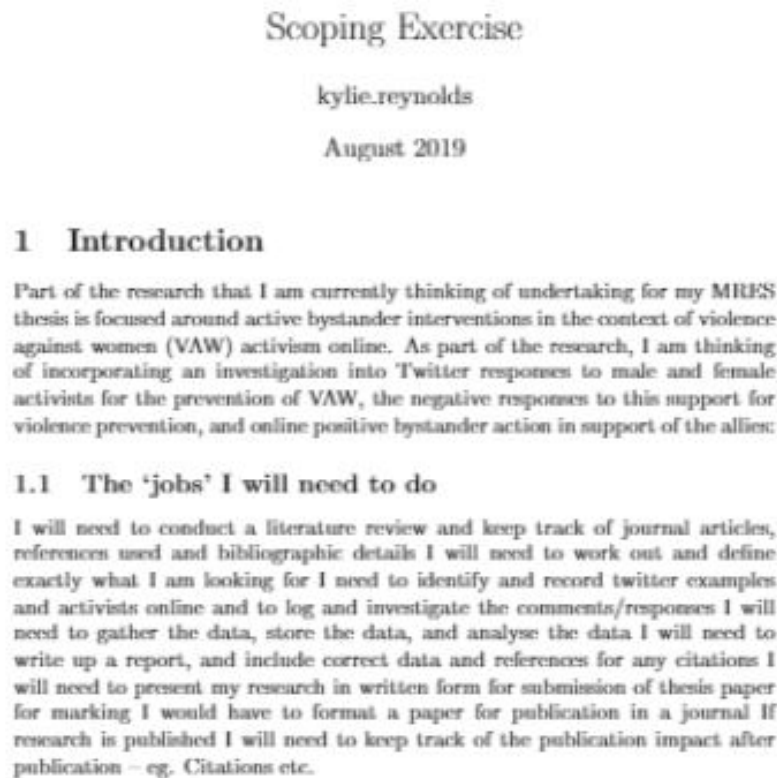
18/08/2019 9:33 am

Objective: to create an Overleaf document for my Proof of concept: Scoping exercise

- I logged on to Overleaf
- Added a new project
- Named it Scoping Exercise and it automatically took me to the document.
- I copy and pasted the Introduction paragraph from my word document into the already set up Introduction section
- I clicked on recompile on the viewing screen and the text seems to have translated properly.
- I then added a subsection and named it “The jobs I will need to do”
- I copy and pasted a list of jobs underneath this heading
- I clicked the recompile button to see on the viewing page

Problem: The text showed up in a paragraph instead of a list. See image below.

Solution: I need to figure out how to change this by learning the command codes for creating a bullet list (see below)



Steps:

- I looked up on Google how to create a bullet list in LaTeX
- One of the links that came up was: <https://latex.org/forum/viewtopic.php?t=12143> which took me to a LaTeX Community forum (saved in bookmarks in Uni/FOAR705) question and answer. The answer included the command codes used to create both numbered and bullet lists, which will come in handy.
- I went to my Scoping exercise project on Overleaf and entered the commands that were presented in the answer referred to above, added a couple of list items and pressed on the recompile button to see if it was successful, and it was. See Figure A below.
- I will now continue to add the rest of the items to the list, and to create the rest of my subsections, paragraphs, and bullet lists.
- This was successful and there were no additional problems
- Once finished I downloaded a PDF version and uploaded to iLearn and Cloudstor.

- Committ to GitHub, See Figure B.

Figure A

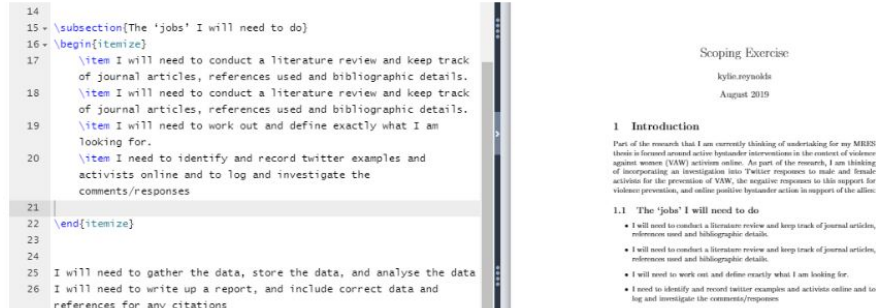
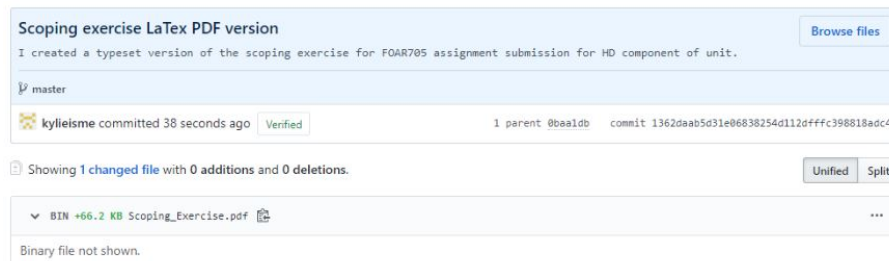


Figure B



21/08/2019 8:09pm

I just had a look at the marks on my scoping exercise I on iLearn and realised that I had doubled up on some bullet points in my LaTeX document. I went back in to edit the document and deleted the duplicate lines. I will need to submit on GitHub and will have to remember to check/proofread my work before handing it in (see below)



- I tried adding the double forward-slash (can't demonstrate as it creates an error in overleaf) to the end of the line before where I wanted the line added and then pressed enter and then recompile and it worked.
- The line was indented so I entered forward-slash noindent (can't demonstrate as it creates an error in overleaf) at the beginning of the paragraph and it got rid of the indent
- Yay!! success.

And now it is time to go to sleep (11:43pm)

22/08/2019 10:35pm

Reflection: Turns out last night's effort was not a success – as self-punishment I have deleted everything and will start again from scratch – this week the journal will have to be submitted in Word format as I don't have time to learn how to do all that I need to do in LaTeX, but, I will not give up!!

24/08/2019 10:49am

Objective: To duplicate my 1st LaTeX scoping exercise and to edit the name of the document to create Scoping Exercise II. What I did:

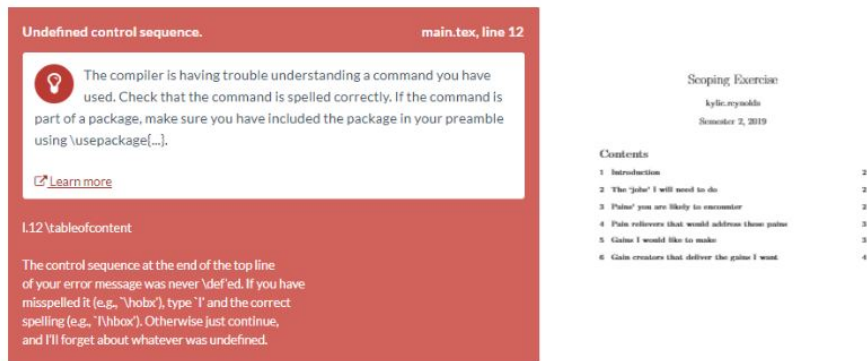
- I logged into my Projects on Overleaf
- I noticed a copy symbol at the end of the individual project rows and I pressed this button
- A new project populated called "Scoping Exercise (Copy)"
- I couldn't see any buttons which indicated that I could rename a document
- I tried checking the tick box next to the copied file and noticed that extra buttons/options showed up on the top right of the page.
- I selected the More drop down option and Rename showed up
- I clicked on rename and renamed the project Scoping Exercise II
- I also renamed the original Scoping Exercise, Scoping Exercise I
- There were no errors throughout this process
- Result: Success

24/08/2019 11:35am

Objective: To add a Contents list to my scoping document What I did:

- searched for Overleaf learn library for Table of contents
- the first link that showed up was "Table of contents - Introduction"
- I clicked on this link

- At the top of the page the code suggested was forward-slash tableofcontents, which seemed straight forward and easy
- I copy and pasted this command after the title area
- I pressed recompile to test it out and it showed up as an error (see below) which was a little confusing.

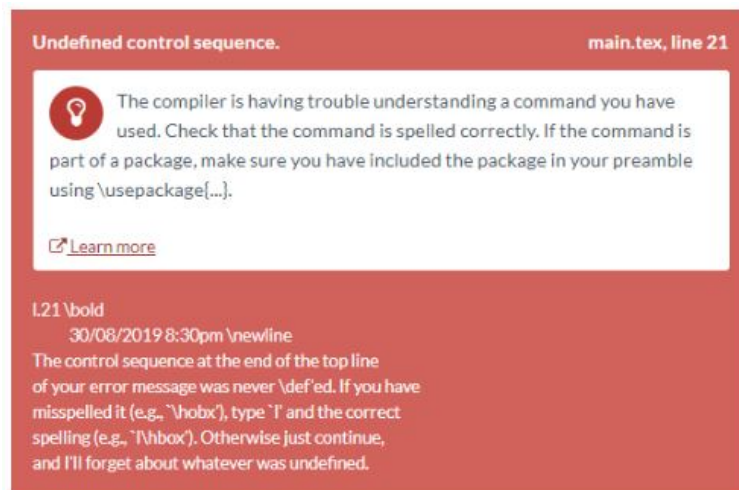


- I checked that what I had copy and pasted was correct
- It was not correct as I had missed the 's' on the end of the command
- I fixed the code and pressed recompile
- and success it worked!
- Reflection: double check when copy and pasting code.

01/09/2018 10:49am

Today I am reattempting to create my learning journal file in LaTeX. I have successfully set up my title page with contents list and have added some Titles and subtitles. I am beginning with the content and want to make the date stamp bold. I tried to guess the code before looking it up.

- the code that I entered before the text that I want bold is forward-slash bold
- I then pressed the Recompile button, however, it showed an error (see below)



I went to the Overleaf help library and typed in “bold” in search bar. A page came up in the search which was Bold, italics and underline which I went to and save in my bookmarks for use at a later time: https://www.overleaf.com/learn/latex/Bold,_italics_and_underlining#Bold_text

- the code to make the date bold is:

`\textbf{30/08/2019 8:30pm}`

- I pressed the Recompile button again and the error was gone and the text was bold. Success!
- Commands for Italics is:

`\textit{add words}`

- Command for Underline is:

`\underline{add words}`

`\emph{add words}`

- Command for Emphasis is:

06/09/2019 8:10pm

LaTeX - Learning journal

I have previously presented my learning journal for assessment in word format as I have used a lot of images throughout to show examples. However, I am now going to attempt to create this document in LaTeX. The first thing that I am going to find out is how to add images.

Adding Images:

- I looked up “adding images” in search bar in Overleaf library and found a page Learn LaTeX in 30 minutes:
https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes#Adding_images. I have saved this in my bookmarks as I believe this page will come in handy throughout this semester.
- I went to section 8 Adding images of this document which says to use the graphics package. See code below:

`\documentclass{article}`

`\usepackage{graphicx}`

`\graphicspath{ {images/} }`

- Which I copied and pasted over what was there previously at the top of the page
- It then said to upload the image that I want
- before I upload the image I will add all the text I want before the first image and format it how I want it to be
- Just added the text and I can't get anything to sit right and everything I try and do makes an error and I don't have time to look up every single thing so I will submit in word again. I hate this program!!!
- What I learnt - nothing!
- Now that I have calmed down. I am just going to try and add a picture since that is what I started out to do.
- I downloaded a random picture to my laptop desktop

- uploaded it to my learning journal overleaf project by

- clicking the upload button
- selecting the file and uploading
- I then typed the command:

```
\includegraphics{nameofimage.jpg}
```



- pressed recompile
- and the image was there
- HOWEVER, it is now showing a horrible little warning symbol. see below.

```
23 \includegraphics{Capture.JPG}
```

- sort of success I guess. Can I just ignore these horrible little annoying symbols if it all looks ok on the page?? I need to find out.
- I think I am getting somewhere now. I have added the command forward-slash, curly brackets, flushleft inside brackets and an end flushleft command at the end of the document, and all of the little annoying warning signs have dissapeared. Yay!
- I have also found out how to get rid of the dollar sign errors by adding a forward-slash sign in front - well it worked for a hyphen so I assuming it will work for other if not I will come to the bridge when it happens
- I will now continue calmly to add in text and images for learning journal and will document any problemos.

Next task: I need to find out how to make my typed commands text so that the computer doesn't recognise as commands instead of text eg forward-slash.

12/09/2019 10:10am

- I have been adding the details of my Unix Shell learnings to the Learning Journal and I have come across an error.
- I believe the error was created because I have missed putting a forward-slash in front of a #, see below.
- I will see if I can find it in the document where I was last adding information to see if I can find the error and will add a forward/in front to see if it fixes the error.
- This fixed the error!


```
You can't use 'macro parameter character #' in vertical      main.tex, line 1579
mode.
l.1579 #
      Calculate stats for data files.
      Sorry, but I'm not programmed to handle this case;
      I'll just pretend that you didn't ask for it.
      If you're in the wrong mode, you might be able to
      return to the right one by typing 'l' or 'l$' or '\par'.

[51][52] (/compile/output.aux)
Here is how much of TeX's memory you used:
2491 strings out of 492990
34412 string characters out of 6132621
100190 words of memory out of 5000000
5975 multiletter control sequences out of 15000+600000
8499 words of font info for 30 fonts, out of 8000000 for 9000
1141 hyphenation exceptions out of 8191
41i,7n,31p,518b,421s stack positions out of 5000i,500n,10000p,200000b,80000s
</usr/local/texlive/2017/texmf-dist/fonts/type
1/public/amsfonts/cm/cmbx10.pfb></usr/local/texlive/2017/texmf-
dist/fonts/type1/public/amsfonts/cm/cmbx12.pfb></usr/local/texlive/2017/texmf-
dist/fonts/type1/public/amsfonts/cm/cmr10.pfb></usr/local/texlive/2017/texmf-
dist/fonts/type1/public/amsfonts/cm/cmr17.pfb></usr/local/texlive/2017/texmf-
dist/fonts/type1/public/amsfonts/cm/cmss10.pfb></usr/local/texlive/2017/texmf-
dist/fonts/type1/public/amsfonts/cm/cmsy10.pfb>
Output written on /compile/output.pdf (52 pages, 1625869 bytes).
```

12/09/2019 8:00pm

Problem: While writing my journal I have come across an issue. I have been trying to write a caret into the text, however, if I put a forward-slash symbol in front of it like the \$, the symbol just disappears and is not shown in the text.

Solution:

- I googled "print a caret symbol in LaTeX" and came up with an answer on Stack Exchange website which demonstrated how, although in a very round about way, to make the symbol stay in text. Here is the link to the page:
<https://tex.stackexchange.com/questions/77646/how-to-typeset-the-symbol-caret-circumflex-hat>.
- I chose to use the simplest suggestion which was forward-slash textsuperscript, curly brackets with \$forward-slashwedge\$ inside (When I say forward-slash I am talking about the symbol because I am unable to type that also.)
- I need to know how to make all of the different symbols print out in text. I am struggling to find anything that makes sense to me.
- \yay I did it!
- I also put a call out on Slack FOAR705 #LaTeX channel and got a suggestion from Brian. Here is the link: https://en.wikibooks.org/wiki/LaTeX/Special_Characters (bookmarked on laptop for other handy symbols that are not shown below)
- see also the handy image below:

Command	Sample	Character
<code>\%</code>	%	%
<code>\\$</code>	\$	\$
<code>\{</code>	{	{
<code>_</code>	—	—
<code>\p</code>	¶	¶
<code>\ddag</code>	n/a	‡
<code>\textbar</code>	n/a	
<code>\textgreater</code>	>	>
<code>\textendash</code>	n/a	—
<code>\texttrademark</code>	n/a	™
<code>\textexclamdown</code>	n/a	¡
<code>a</code>	X ^a	^a
<code>\pounds</code>	n/a	£
<code>\#</code>	#	#
<code>\&</code>	&	&
<code>\}</code>	}	}
<code>\S</code>	§	§
<code>\dag</code>	n/a	†
<code>\textbackslash</code>	n/a	\
<code>\textless</code>	<	<
<code>\textemdash</code>	n/a	—
<code>\textregistered</code>	n/a	®
<code>\textquestiondown</code>	n/a	¿
<code>\textcircled{a}</code>	n/a	Ⓐ
<code>\copyright</code>	n/a	©

Finish: 12/09/2019 8:40pm

Start: 13/09/2019 10:45am

- I just re-read the rubric for HD mark in iLearn and it says that the error section of the learning journal should be step apart from other sections. I have been adding into the particular sections that they are connected to.
- I am now going to setup a new section by adding `\section{Errors}`, and by adding subsection headings for each section throughout this document.
- My next task will be to find all of the errors and to copy and past them into the error section to set them apart. I will also leave them in the section they belong to as it was part of the original flow of the learning process and if I need to follow my directions again I can stop and not make the same mistakes.

- to use my time wisely, while looking through this long document, I will try and fix up any textual and formatting errors throughout.
- **NOTE: TO DO ON HOLIDAYS**
- As I have just myself a task to complete (see line above, NOTE...) I would like to learn how to change the colour of text so that these stand out, so that I can follow up on them down the track.
- I searched Google for LaTeX colour text and an Overleaf page showed up:
https://www.overleaf.com/learn/latex/Using_colours_in_LaTeX#Introduction
- I looked at this page which suggested adding a new package `\usepackage{xcolor}`. I added this to the top of my LaTeX document where the other packages are located.
- The command code for the colour that you want to use seems quite simple and self-explanatory. It is `\color{add colour here}`.
- I am going to make the TO DO text above red so that I remember to do it.
- Success!
- Reflection: I was wondering before I changed the colour if I would need to change it back to black, but it seems as if it changes back to the original colour on the next line.

Finish: 11:20am

8 Proof of Concept

8.1 Elaboration stage

8.1.1 Applying for an API from Twitter

30/08/2019 8:30pm

As part of my proof of concept I am trying to discover how to collect a specific set of data from Twitter. I have been advised that to be able to collect that data, if I am unable to find a publicly available tool that is free, is to create some code to be able to get the data from Twitter directly. To do this I will need and Application Program Interface (API) key from Twitter which I need to apply for. The application process will be documented below: