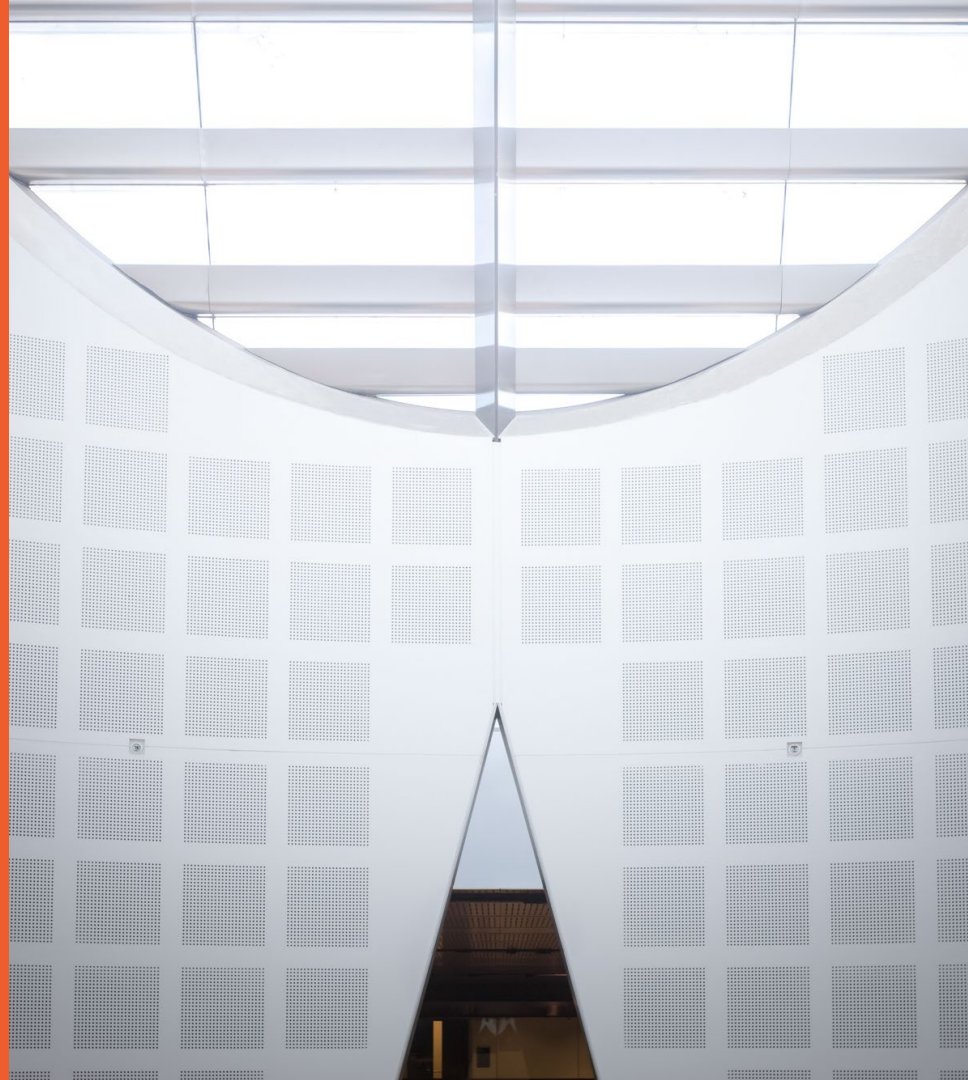
DATA2901: Data Science, Big Data and Data Diversity (adv)

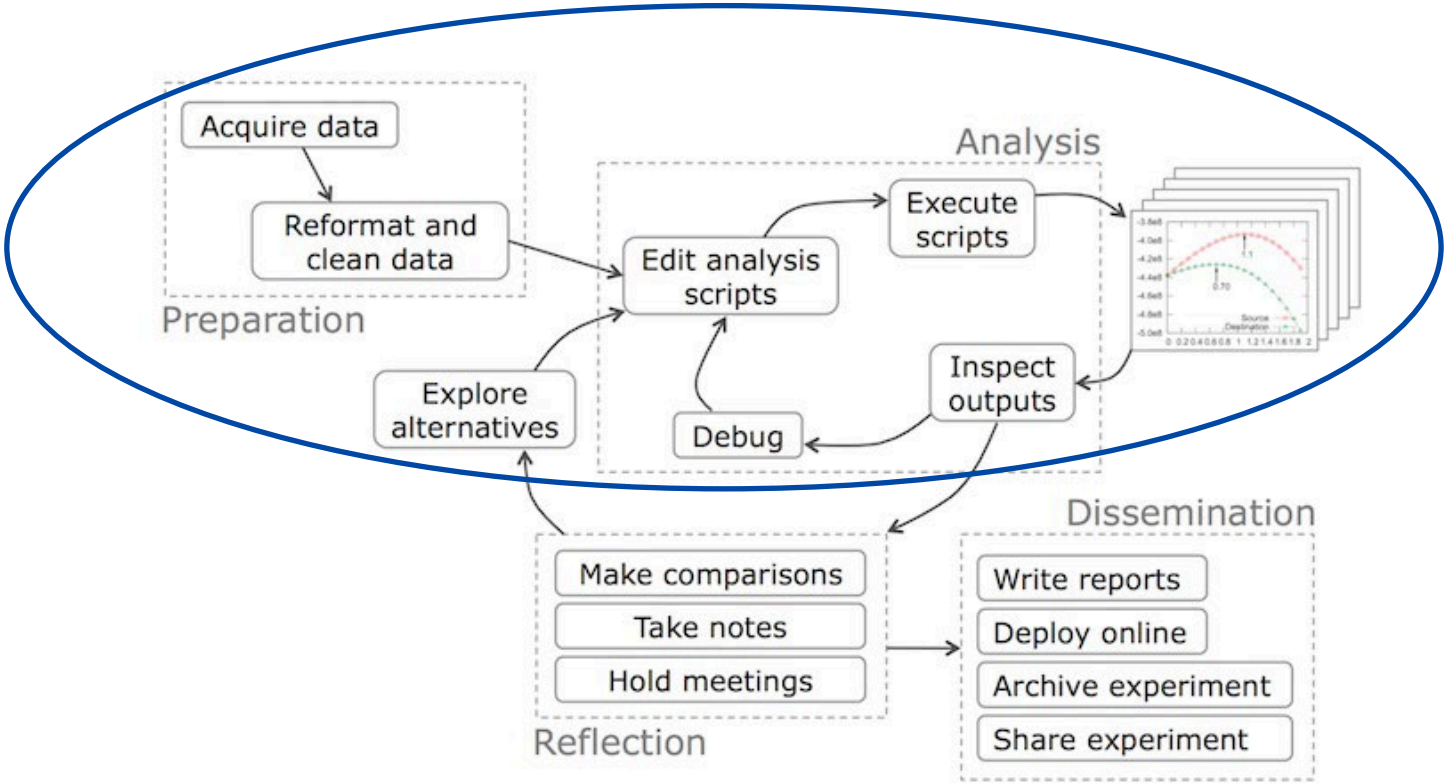
Introduction to Unix Tools

Dr Ali Anaissi
School of Computer Science

THE UNIVERSITY OF SYDNEY



Exploratory Analysis Workflow



Exploratory Data Analysis with Unix

Unix…

- Unix is a family of multi-user, multi-tasking operating systems
  - Development started in the 1970s at Bell Labs
  - AT & T then licensed Unix to outside partners
  - Various commercialization attempts (some very successful)
    - Sun **Solaris**, IBM AIX, Microsoft Xenix, HP-UX
    - Berkeley **BSD Unix**
- **Linux** started 1991 as separate Unix-like project by Linus Torvalds
  - Free Software Foundation keeps referring to it as GNU/Linux
- Apple macOS is based on a Mach kernel with additional layers and tools derived from **BSD Unix**

Things to keep in mind

- Unix command line tools are **case sensitive**

- Unix commands are **keyboard-centric**; every character counts
  - ls instead of "list" or "dir"
  - chmod instead of "change_mode"

- Unix **seldomly asks** before executing something
  - be very careful when issuing modification or deletion commands!

Unix Filesystem

- Hierarchical file system with tree-like directory structure and access rights
  - "/" refers to the root of the Unix file system
  - Everything else is somewhere accessible under the root directory
  - Directory path delimiter: "/"
  - Classical path structure in Linux

    /bin /etc /sbin /usr /var /dev /home

- In Unix, everything is a file
  - e.g. /dev
- Cf. https://upload.wikimedia.org/wikipedia/commons/f/f3/Standard-unix-filesystem-hierarchy.svg

Commands to navigate the Filesystem

```
pwd
cd <name>
    cd
    cd -
    cd /
ls [<name>]
    ls -al
    ls -R
mkdir <name>
mv <old> <new>
cp <source> <destination>
```

- display working directory
- change directory
  - change to home directory
  - change to previous directory
  - change to root directory
- list (current) directory contents
  - list everything with all details
  - list directories recursively
- make new directory
- rename / move a file or directory
- copy a file

Finding files across directories: Find

find *startdir* **–name** *filename* **–print**

- Searches for files by name (or other characteristics) in directory sub-tree
- Displays matching files
- Note: case sensitive
- Many options!
  Can e.g. search by creation date or owner or file flags
  including negation and conjunctive conditions

Basic Unix Utility Commands

man *commandname*

- Read documentation ('man-page') of a given tool
- Also most unix commands support a '**-h**' or '**--help**' option for help

history

- List which commands have been executed before

!*num*

- Repeat command *num* from history

Looking at file content

- **cat** *filename*
  - output a file content in one go
- **more** *filename* or **less** *filename*
  - display (text-)content of a file page-by-page (less is the more powerful command)
- **head** *filename*
  - output the (by default: 10) first lines of a file
- **tail** *filename*
  - output the (by default: 10) last lines of a file
- **wc** *filename*
  - word count of file content; shows: *num_line* *num_words* *num_characters*
- **sort** *filename*
  - output content of file sorted; many options on how to sort

Looking at columns of a (CSV) file

cut –f *fields* **–d ,** *filename*

- output only the given fields of the file, as separated by the delimiter given with option –d (be default, it would assume the tab character)

Example:

```
cut -f 1,4,6-7 -d , programming_experience_survey_2018.csv
```

- Outputs columns 1, 4, 6 and 7 (as separated by comma in the file) from the CSV file programming_experience_survey_2018.csv

Attention: cut is very simplistic; e.g. does not parse quoted strings correctly which contain a delimiter sign…

Output Redirection

- Output of a Unix command is displayed by default on screen
- Can be re-directed into a file
- Those files can be read by the next command then
- Re-direct stdout:

  command > filename
- Re-direct stderr:

  command 2> filename
- Re-direct both to same file:

  command &> filename

Combining Unix Tools: Piping

- Remember: In unix everything is a file
  - In particular the output of a Unix command is a file 'stream'
  - => can hence be directly used as input for a subsequent command
- No need to materialize (store) the output of a command
  - Instead can be 'piped' into a subsequent Unix tool
- | sign

- Example:

  cat *filename* | sort | head -3

Finding data inside a file: grep or egrep

grep *pattern filename*

- Searches for patterns in file contents
- Displays matching lines
- Note: case sensitive

- Egrep variant supports extented regular expressions

Pattern Matching: Regular Expressions

- Sequence of characters that define a search pattern
  - Special characters for wildcards, options, repetitions, conjunctions

- Boolean or (Option): *vertical line* gray|grey
- Grouping with parenthesis: gr(a|e)y
- Wildcard: . matches any 1 char gr.y
- Quantification:
  - ? 0 or 1 (optional) colou?r
  - * 0 or more ab*c
  - + 1 or more ab+c
  - {n} n times
- Alternatives in [...] gr[ae]y
- Negation: ^ gr[^bcdf-z]y

Processing content of files: awk

- A programming language for the special purpose of text processing and data extraction – and its corresponding tool
  - AWK was created at Bell Labs in the 1970s,[3] and its name is derived from the surnames of its authors—Alfred Aho, Peter Weinberger, and Brian Kernighan
- Very powerful pattern matching language, where code blocks can be executed for each match, and data be extracted into variables or send to output
  - Special BEGIN and END 'patterns' to execute at start or end of a file
  - Field separators can be specified

AWK Example: Word Count

```
BEGIN {
      FS="[^a-zA-Z]+"
}
{
       for (i=1; i<=NF; i++)
            words[tolower($i)]++
}
END {
       for (i in words) print i, words[i]
}
```