The time's they are a changin'

- Pandemonium – well nothing new there – look at history!
- Online teaching
  - different & arguably daft, but alternative is dafter
  - But online not really that different…
    - Less interaction with mates and staff, but possibly less distraction
      - And there will be some manner of 2-way interaction via lab-screen sharing etc.
    - But still in front of a machine – no panic, will record brief demos for easy solutions
      - easy to set up basic needs via any machine for this course – will show later
    - Generally this class was led from the front in labs by myself
      - demonstrators were hard to find - previous course was 'different'
    - Will take an easy introduction for the first few weeks, to ease people in...
      - May be a bit slow for some at the start, but walk before run later.
  - But look at the advantages of not coming to college – more time...
    - Travel eliminated… more time to study – right yeah!
    - More time to experiment and be self-taught
      - Can't cover everything and most tech is trivial and tedious rather than mind bending
    - Find a hobby, meaning and purpose in life, as well as just having fun!

Priorities – simple & fault tolerant

- Assessment –
  - 100% years work
  - Avoids over-dependence on one end of year test.
- Lectures & Labs - synchronous intended:
  - interactive at scheduled times,
  - Recorded for later reference, if needed
    - apart from first Mon am to avoid teething problems
    - Canvas registration delays
    - sudden switch off-campus for some with limited internet and no time to fix
- Computer & Tech requirements for all
  - Regardless of internet connectivity
    - Except really poor connectivity may obviously inhibit immediate interaction and possibly in-class tests.
  - On almost all platforms

Assessment

- 100% years work
- Avoids over-dependence on one test.
- replicated in case of poor performance in one
- 2 laboratory assignments, 20 marks each;
  - Problems to take away and have time to solve;
- 2 in-class tests, 30 marks each;
  - Problems to be solved within a short time frame

Lectures & Labs

- synchronous intended:
  - interactive at scheduled times
  - Recorded for later reference, if needed
- Reality is challenging
  - Sticking with online video courses is tough
    - Online courses have quite low completion rates
    - Difficult to maintain concentration
    - Non-interactive – as it says..
    - Absent in person, more likely absent in mind
  - Trawling through video is also tough
  - Video is bandwidth/storage intensive
- As a sensible approach, I hope to provide
  - edited text transcription of main issues, which are easier to search and visually scan,
  - Increased interactivity and engagement : Cycles of short a presentation, followed by a brief demo, followed by an exercise to try it yourself…
  - More like labs, or previous education levels… as material is more practical & technical than theoretical, and more difficult to maintain concentration if not live in person.
- Related free book for fault tolerance, in case of absence, comms. issues

Computer & Tech requirements for all

- Primarily command line interface (CLI)
- Minimal requirements
- Easily supported on almost all platforms,
  - including phones if you wanted
    - With bluetooth keyboard for extra convenience
- good connectivity - connect to CS server
- poor connectivity / speed & convenience
  - use your own machine locally
    - Including rooted phones
      - Not recommended issues with
        - Security & guarantees.

good connectivity - direct to CS server

- Consistency and backup (almost) guaranteed
- Browser based : easiest for now – no install
  - e.g. Chrome extension
    - Recommend Secure shell,
    - developed & used by Google – so should be ok
  - Firefox & others (Chromium based, Firefox, Edge)
    - in a state of flux due to recent codebase & security changes
    - Some tools deprecated, current security status unverified
- Various GUI based apps – similar to FileZilla
- Command line options… the most reliable
  - ssh etc, will cover later

Convenience / Poor connectivity
use your own machine locally

With built in Linux/Unix available

(subject to configuration– more detailed demos to follow)

- Windows Subsystem for Linux (WSL)
  - Available on Windows 10
  - MS new deadly embrace!?
- OS X is built on Unix BSD, so just launch Terminal
- Chromebook (Chrome OS), again launch Terminal
- And if using Linux… already there!
  - Just launch Konsole or Ctrl+Alt+T shortcut

Or virtual machine on older Win… may cover later (time)

- e.g. Virtual Box … and install OS of choice

No pain – no gain!

- Why use this archaic, obtuse interface?
  - GUI (Graphical User Interface) is
    - a menu for mice – or other pointing device!
    - Convenient : - no learning curve, drop down menu
    - But slow for big jobs – e.g. global filesystem edits – click until you're sick!,
      - can't be programmed easily (but changing)
      - Limited by the speed of a mouse (whoever is driving it!)
      - May cause RSI (Repetitive Strain Injury) – many switch hands
  - CLI (Command Line Interface) is
    - Programmable - !!!
    - Not really convenient for once off
    - But fast for big jobs – global filesystem edits ~ one liner!
      - can't be programmed easily (but changing)
      - Limited by speed of machine, not speed of user...
      - May cause FITs (Fuming Insane Tantrums) until tamed
    - Battle tested, robust & reliable…
      - GUI's are often just wrappers / interface for underlying CLI utilities
      - Glitches with GUI's – often fail, even in simple file finds, whereas CLI just works!

Course Progression

- Text streams – regex (but not done first)
- Shell commands – common useful commands
  - (picked up along the way, and how to find info. On them)
  - Combining/piping them to form powerful one-liner utilities
- Filesystem – make, find and manipulate files
  - Find, locate etc.
- File contents – make, find and manipulate text
  - Basic system commands : grep etc
  - Manual editors : vim etc., but vim is common & worth knowing about.
  - Programmable editors – even used in software dev.
    - Sed, awk etc.,
      - Can be thought of as programming language,
      - Do before bash – to avoid reinventing the wheel.
- Shell scripting : bash all of the above, and more together
  - To monitor systems, solve problems & mess about.

Advantages

- Automation of repetitive system tasks
  - Backups, system monitoring: disk, CPU, RAM, Net, etc.
- Powerful text wrangling
  - Incredibly fast
    - Most are written in C (but can incorporate others)
    - Honed over the years for speed optimising code to suit various buffers and cache levels from CPU to disk
  - Even used for data prep. For text mining apps, incl. Genomics
  - Even 'convert' database formats when nothing else available
- Flexible
  - Can generate & modify to suit requirements
- Runs (on) almost everything everywhere.

Some of the most...

• boring,	• useful,
• terse,	• powerful,
• error prone,	• jam-freeing,
• standard-free,	• flexible
• Incomprehensible syntax, almost	• Incomparably powerful tools

you're ever liable to encounter
Are here, like all of life!
And they're honed for speed, C, buffers...

Whats za matter you, eh!?
...but things are not so bad…!

If you have a problem… and can't avoid it...

1)There might already be a command...

2)...Or a command sequenced you can pipe

3)… or you can add a hack to some of the above

4)… or someone has solved a similar problem

5) … which you can hack

6) … only then - consider developing, which gives skills, insight, opportunities for reuse or even contribution to a code repository.