Burnet Institute

Medical Research. Practical Action.

Coding and Software Club at the Burnet Institute

a Sisyphean story of normalising peer-to-peer learning

I acknowledge the traditional custodians of the land in which my work takes place and pay respect to Elders past and present.

Acknowledgements

- My role is funded by the EC Australia partnership, supported by the Paul Ramsay Foundation
- I work on ACCESS, funded by the Department of Health
- I thank Geoff Chan (Burnet Institute), Mar Quiroga (University of Melbourne), Nick Wong (Monash University), and Merryn Roe (Burnet Institute) for participating in Coding and Software Club from its inception

Outline

- Who – is the Burnet Institute and its Coding and Software Club
- Why – were we motivated to start the Club and keep it going
- What – tools do we use to engage, teach, and learn
- How – has Club impacted people

Who

- Independent medical research institute
- Life sciences (laboratory), public health, and international development
- From a clinical/laboratory beginnings, population health expanded, leading now to large surveillance projects, long-standing cohort studies, and some implementation research (trials)
- Collect, analyse, use and store large amounts of research data
- Dominant software has been Stata, with REDCap now used extensively

Why

- Learning software for cleaning data and analysis seen as an essential skill, but ad-hoc as to how to get the skills:
  - found a buddy, workshop, learnt from a leader, struggle on own
- Few people seen as ‘go to’ for learning Stata (quickly overburdened)
- Very little use of R, MatLab, Python, or Git
- Needed a culture of learning and teaching that was:
  - Teaching how to learn
  - Promoted building skills (as opposed to ‘how do I do this thing now’)
  - Not constrained by research group/department boundaries
  - Broaden the scope of languages so Burnet part of contemporary practice

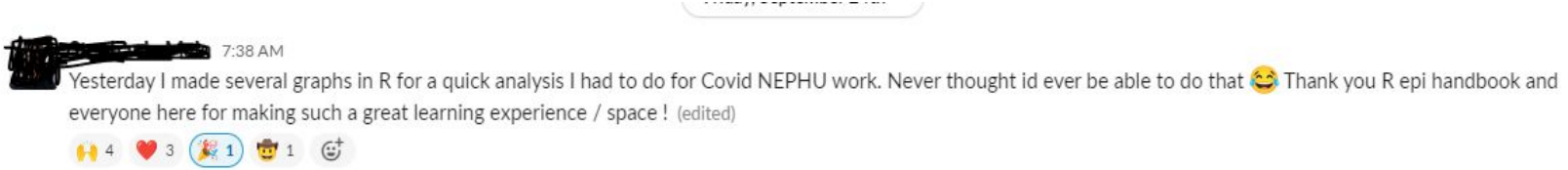
What

- Weekly drop-in session for an hour (now online)
- Started a Slack workspace
- Promoted on intranet, on Slack, through word-of-mouth
- Open to people from other organisations (to leverage off skills and knowledge and build connections)
- Began with a philosophy that Everyone is a teacher, Everyone is a learner
- Tried for more active learning, but unpopular, everyone wants sit back and watch a demonstration
- Attendance variable

What

- Slack took-off with work-from-home
  - 216 members (~ 50 active each week), 2,180 messages (all time leaders has sent ~ 10,000 messages!)
  - Channels: R, Stata, Git, REDCap, lols
- New idea: 'work through a book' on R (most successful strategy to date)
- Share screen on Zoom, work through a section of the book
  - Allows for different learning styles: watch demo, follow along and live code, prepare before and ask questions, present a session
- Offer small group or one to one support to start up and get through the hardest parts of R and what researchers want – immediately see the data in the console, quickly make a plot

How



- Peer-to-peer, free learning (saved people and Burnet money)
- Increased connections (I know them, I see them at Club)
- Created a culture of asking questions (fun, safe Slack workspace)
- Increasing population of R users
  - From one to two R users, we have entire teams and projects that use R
  - Boosted people's skill set (CV)
  - Create flexibility in projects as multilingual people can move between
- Increased interest in data management, coding, version control, analytics

Challenges

- Operating in a prevailing culture of:
  - Use the software I know/we have always used
  - Courses/workshops best way to address professional development needs
- Conveying lessons learnt on what learners want:
  - On-demand, specific to their field of work/task at hand, immediate apply
  - Trusted advice, safe space, permission and time to learn
- Teaching not considered part of people tasks or key performance indicators
- Dealing with reactivity: omics (we need omics skills!, we need data viz!)
- Viewed by some as not a place for beginners
- Is it me: dispersing ownership of the Club to ensure its future

Opportunities

- Building momentum for analytics and people proficient in coding in more than one language
- Speak the Executives language/currency – improve efficiency, meet governance requirements, employer of choice, collaborator of choice
- Students – software agnostic supervisors teach students to use the best software for the task rather than a dogma
- Increasingly complex and international projects force looking for solutions
- Online (Slack, Zoom, You Tube) to address issue of equity access
- Find Indigenous students and give them an opportunity to learn coding

What have I learnt

- Use your values to teach (equity, respect, and democracy of research skills)
- Be a learner if you are going to be a teacher
- Just give learners what they want, how they want it, they are right
- Do not skip the simple stuff with learners – even if you think your starting ‘at the beginning’, you are probably not
- Share; give your time, let others teach, let others take credit, ask for ideas
- Persevere; stand up for teaching and learning best practice and pick a fight about it (but not all the time)
- Permission is rarely needed

Anna Wilkinson
Senior Research Fellow, Burnet Institute, anna.wilkinson@burnet.edu.au