**VBGR2**

*Rationale*

The existing VBGR works for many students but in use we have found:

* Some students lack basic computer skills and so find the pace of the material too fast
* It includes little repetition/repeated practice, and
* Students find it hard to apply/generalise the skills they gain

*Pre-requisites*

Working knowledge of a desktop operating system, use of a browser/email.

Some students will come to the materials having previously studied our stage 1 R material. For these students VBGR2 will provide:

* Revision for core concepts (e.g. use of projects, files and paths)
* Introduction/extension of core programming concepts (assignment/comparison/map)
* Chance to apply/generalise skills with new data

*Differentiation of teaching*

Some students will find these material easy, either because they have prior experience of programming, or through stage 1 materials. For these students

* Encourage repeated practice. E.g. use logical operators in several contexts. Use filter, group\_by and summarise on multiple different datasets
* we will provide extension activities to ensure the workshops have value
* self-test quizzes will be used to enable students to skip some sections of worksheets

*Teaching Delivery*

* *Supported self-paced workshops*
* *Self-test quiz material*
* *Narrated (think aloud) screen captures demonstrating all key skills*

Expand on:

* File handling/computer skills? Uploading and loading data from multiple sources (e.g. via a URL, uploading file to RStudio)
* Programming basics: assignment/comparison, types of variables, more practice of mutate. More plotting.

Write extension exercises to avoid case that students who do remember getting bored/frustrated. Introduce different plot types, Rmarkdown?

Use webex to allow more self-testing

lifesavr

lifesavr is for enlightenment in statistics and visualisation in R

Session outlines

\* indicates new content

† indicates extension activity

1. *BeginneRs*
   * Creating a project
   * Rmarkdown
     1. Creating Rmd file
     2. Chunks contain R instructions (example of simple calculations)
     3. Text in between chunks can describe
   * Exploring data
     1. Loading package in Rmd chunk at the top of document
     2. Loading data
     3. Inspecting data
     4. Calculate mean
     5. Deal with missing data
   * File handling and uploads\*
   * Understanding files and directories. Understand the working directory. Motivate R projects as a way of avoiding this\*
   * Explain files/directories/file extensions. Concept of a default mapping for a file type in Windows. What is plain text?
   * Programming basics: assignment and comparison; using functions.
   * RMarkdown? †
2. *Group differences*
   * Current content but with new examples (i.e. t test)
   * Density plot
   * Boxplot \*
   * Filtering data
   * Expand on operators (& and |)\*
   * Effect sizes shown as density plots (make it a quiz\*)
3. *Evidence 1 - group differences*
   * Bayesian T test again — this time explain how it is the comparison against intercept only model (useful for later)\*
4. *Evidence 2 – correlations*
   * Scatter plot
   * Effect sizes shown in scatter plots (make it a quiz\* and/or use <http://guessthecorrelation.com> )
   * Bayesian evidence for a correlation
   * Colouring a scatterplot\*†
   * Smoothed line plot\*†
5. *Integration session*

*PAULS IDEA OF linking everyting in a single workshop with a research question*

* + Really understanding the difference between variables (Environment) and files\*
  + Hammer home variable assignment and ‘state’
  + Learn some of the keyboard shortcuts (running code with control+enter for pipe and assignment as a minimum). Also learn where the backtick and tilde keys are.
  + Organising code for reproducibility: sequential nature of processing (libraries at the top). Getting a script to run ‘in one’ from top to bottom\* See notes below.
  + Exporting figures as png/pdf with ggsave\*
  + Organising code, making comments with #, adding line breaks, minimising line length\*
  + Downloading/backing up R script files (re-uploading to the DLE or psychel)\*
* Explain files/directories and file extensions + default mappings for file types on Windows. Explain that .R files are just special .txt files.
* Practice uploading data and downloading R script.

*Other less important content details/ideas*

* Activity on commenting of code… in one of the final sessions ask them to tidy up and comment a messy R script e.g. would need to add line breaks, put library(x) at the top of the file etc. Could develop tool to automark this with regex or parser?
* Exercise to break up a pipeline into multiple stages storing state at each point? Then recombine to make it into a single pipeline
* Give them a longish pipeline of code and ask them questions about values at various stages in the pipeline — force students to run segments of the code.
* Some sort of activity to demonstate that Envrionment is ephemeral? Reboot the Rstudio server mid session!!