1. Intros: Who, what and frequency of R use on a scale of 0 – 5, 0 = never, 5 = you script in your spare time
2. Why R – What do you like about R?
3. Previous topics:
   1. Style, formatting, etc. Plenty of resources, some of which were discussed last time.
   2. Does anyone have anything to add to the good of the order?
4. Projects/sessions – Anna Withington
5. GitHub – Demonstrate usage
   1. Who has a Github account/Who uses a GitHub
   2. GitHub can be used for anything. Works well when collaborating.
   3. Create repo locally, push it up to cloud
   4. Versioning, branching
   5. Pushing/pulling
6. Functions
   1. Functions are good because you can modularize your flow process and make calls to specific tasks without having to recreate the process every time, or can save processes for later use.
   2. When you create a function, does the order of the parameters passed matter?
   3. Does the order matter if you explicitly state the variable name, i.e.,
   4. Data types: 1 / 60 = ???
   5. How do you call a function without loading and “compiling”? Do you have to create a package for this?
   6. What are the expectation for standardization of inputs. E.g., if it just something I use then I can do what I want, but it shouldn’t be something that just I use so what is the best way to require parameters and what is the best way to coerce. E.g., python has “exceptions”, does R have something similar?
7. Ifelse returns vector of same size – is there a function that will (aside from if {} else {})?

GitHub – Kevin B said there is a GitHub endorsed YouTube channel that is very good for learning GitHub

Travis P – GitHub master repo is for development of the scripts, functions and packages until it’s “done”, then subsequent versions are branched.

Functions – Use “=” inside a function, use the assignment “<-“ outside of functions

Use “source()” to call a function within code. Else create a package and load as a library