**Data Scientist’s Toolbox**

“Ask yourselves, what problem have you solved, ever, that was worth solving, where you knew all of the given information in advance? Where you didn’t have a surplus of information and have to filter it out, or you didn’t have insufficient information and have to go find some?” – Dan Myer, Mathematics Educator

what do data scientists do?

define the question

define the ideal data set

determine what data you can access

obtain the data

clean the data

exploratory data analysis

statistical prediction/modeling

interpret results

challenge results

synthesize/write up results

create reproducible code

distribute results to other people

how to get help with R

access help file: ?rnorm

search help files: help.search(“rnorm”)

get arguments of function: args(“rnorm”)

type function name without any brackets to get the code that corresponds to that function

how to ask an R question

what steps have you executed to reproduce this problem?

what do you expect the output to be? what do you see instead?

what version of the product are you using?

what operating system?

a note on Googling questions:

Stackoverflow: use the tag “[r]”

R mailing list for software questions

Crossvalidated for general questions

Google: “[data type] data analysis” or “[data type] R package”

“Natural language processing” for data from texts

how to ask a data analysis question

what is the question you’re trying to answer?

what steps or tools did you use to answer it?

what did you expect to see? what do you see instead?

what other solutions have you thought about and tried?

**Command Line Interface**

commands follow the recipe *command flags arguments*

*command* is the CLI command which does a specific task

*flags* are options we give to the command to trigger certain behaviors, preceded by a -

*arguments* can be what the *command* is going to modify, or other options for the *command*

**pwd** tells what directory you’re in

**clear** clears the screen

**ls** lists the files in the current working directory

**cd** allows you to change working directories

**mkdir** creates a new directory

**touch** creates a new file

**cp** copies a file

**rm** removes a file (or directory if you use the –r flag)

**mv** moves or renames a file

**date** gives the date

**echo** echoes out a particular command

**Introduction to Git**

*Version Control* is a system that records changes to a file or set of files over time so that you can recall specific versions later

with Git, everything is stored in local repositories on your computer (called repos)

operated from the command line

**Introduction to GitHub**

allows users to “push” and “pull” their local repositories to and from remote repositories on the web; users repositories are backed up on the GitHub server

provides users with a homepage

social aspect allows users to follow one another and share projects