Notes 4/22/19

R Packages

* File structure
* Coding/ programming
  + Functions
  + Tests
* Probability

Anatomy (File Structure) of an R Package

* Minimum & mandatory file structure
* Ex: Package “Pkg”

Pkg/

R/ ***(R scripts for functions, use Roxygen comments to document things)***

Function1.R

Function2.R

* + - * Make sure to do:
        + #’ @title / #’ @description / #’ @param / #’ @return
        + Special roxygen for packages:

#’ @export 🡨 will export this function to NAMESPACE

#’ @examples

#’ *example 1 goes here*

#’ *example 2 goes here*

Man/ ***(Files for manual-technical documentation)***

Documentation.Rd

* + - * Uses a TeX language dialect for the text files

DESCRIPTION ***(plain text file, no .txt, must be all upper case)***

* + - * Metadata of your package (business card of the package)

NAMESPACE ***(plain text file, no .txt, must be all upper case)***

* + - * Lists the imports & exports directives, list which functions are exported to the end user and which are imported from other packages (dependencies)
      * How does R know which function to use if your function is the same name as another function?
        + A <- pkg::function() in the NAMESPACE file
        + Only thing in NAMESPACE is *exportPattern(“^[[:alpha:]]+”)*

.Rbuildignore

Pkg.Rproj 🡨 these two parts are optional, not totally necessary

To make the package…

Go to R Studio, create new *project* not a new *file*

Existing directory or new directory

Select the project option R package

Give it a name

Devtools Package

* Need to remove NAMESPACE and let devtools make it
* Need to remove all the default R Studio files
* Run devtools::document()
  + Generates the documentation of the package
* Run devtools::check\_man()
  + Want to see no issues detected
* Run devtools::test()
  + Check the tests
* Run devtools::build()
  + Build the package