Notes 4/1/19

Testing Functions

* Can do so using the package “testthat”
  + Core of testthat consists of expectations, to write them use functions such as expect\_equal(), expect\_integer(), or expect\_error()
* Can use toy vectors to test different outcomes and consistency
* The basic functions are expectations in testthat
  + Organizing the expectations (diff tests) into groups of tests (test unit)
    - Do this with test\_that(“some label message”, {

Z <- c(1,2,3,4,5)

Expect\_a(z)

Expect\_b(z)

Expect\_true(z)

} )

# *notice the parentheses is closed at end, not after label*

* + - Use context(“Tests for standardization, etc”) beforehand to contextualize the test\_that() use

Ex: standardizing a variable

* Math way: , where X is an observation and Z is the Mean
* standardize(x, na.rm) where x is a vector of values

Ex: Expectations of length

* expect\_length(x, a)
  + If you get no output 🡪 the length of x is equal to length of a or length(a)
  + Will get an error x is not same length as a

Ex: Expect Error

* expect\_error(some function(values))

The source() function loads R scripts

* allows you to source the code in a given R script
* load() works for a .RData file
* library() works for packages

OVERALL ORDER:

1. source(a.R # the script)
2. context(“…”)
3. test\_that(“…”, {

} )

CAN DO IT ALL WITH: test\_file(“test.R”)

* output:

X | OK F W S | Context

X | 11 | Tests for Standardization

Line of failure report

-----Results-------------------------------------------------------------------

Ok: 11

Failed:

Warnings:

Skipped:

Example Directory:

Project/

Code/

Functions.R

Scripts.R

Tests.R

Data/

Images/

Report/