Chapter 2 Lab + Homework Cheat Sheet

**Subsetting:** a way to create a smaller dataset with only the specific criteria you are looking for.

subset(*dataset*, *logical operator*)

Logical operator options:

* Greater than >
* Less than <
* Equal to == (TWO! TWO!)
* Or |
* And &

Save the data:

groupname = subset(master, group == “yay”) ##just an example

**Tapply:** calculates functions on specific groups in the dataset. Tapply is used when you have factor variables.

tapply(*dataset$column*, *dataset$column*, *function name*)

* The first column should be the variable you want to calculate the descriptives on (like the dependent or continuous variable)
* The second column is the factor variable that you would like to split the dataset on (like the independent or categorical variable).
* Functions examples: mean, sd, var … but do not include ().

More than one factor:

* Discuss levels versus conditions

|  |  |  |
| --- | --- | --- |
|  | Left | Right |
| Male | Male, Left | Male, Right |
| Female | Female, Left | Female, Right |

* tapply(master$dv, list(master$IV1, master$IV2), mean)

**Standard Error and Length function:**

Formula for SE = SD / sqrt(N) – but we want to find N dynamically.

length(*dataset$column*) tells you the number of rows / items in that column. Careful, length(*dataset*) is also a valid function, but that gives you the number of columns.

You can use the length function in tapply – tapply(master$dv, master$IV1, length)

To get standard error, you can use the combination of everything we’ve done so far, but let’s be fancy:

M = tapply(master$dv, master$IV, mean)

STDEV = tapply(master$dv, master$IV, sd) ##do not save this as SD, trust me

N = tapply(master$dv, master$IV, length) ##only with no missing data

SE = STDEV / sqrt(N)

When you save things, it will not print them out, but you can tell it to print all at once, which moderately nicely formats them for you:

M; STDEV; N; SE

Why save? We are going to use these saved outputs later! For example, if you want to get **degrees of freedom**:

N – 1

**Confidence intervals:** We are going to calculate these based on the Z distribution:

M + Zcutoff \* (SE)

M – Zcutoff \* (SE)

Zcutoff is 1.96 for 95% confidence interval, while 2.58 is for a 99% confidence interval.

**Effect size:** how big the difference is between groups. In this example, we are going to use independent *t* formula, but we will cover more later on how to know which one to use.

FIRST TIME INSTALLING MOTE ONLY:

* Install the devtools library.
* In the command line (not your Rmd or if you do, # it out): devtools::install\_github(“doomlab/MOTE”)
* Install MBESS package

Now, MOTE will be installed like a regular package, which you can load:

library(MOTE)

d.ind.t(*mean1, mean2, sd1, sd2, n1, n2*, a = .05)

Since we saved all of these numbers earlier, we can do something like this:

d.ind.t(M[1], M[2], STDEV[1], STDEV[2], N[1], N[2], a = .05)

**Power:** Using the *pwr* library, we can figure out how many participants we would need to find a significant effect.

pwr.t.test(n = NULL, ##leave this alone to get the sample size

d = *effect size,* ##the effect size from the above calculation

sig.level = .05, ##leave this alone, traditional alpha

power = .80, ##leave this alone, traditional power levels

type = c(“two.sample”), ##this part will change depending on the type of test

alternative = c(“two.sided”)) ##leave this alone for two tailed test

pwr.t.test(n = NULL, d = effect$d, sig.level = .05, power = .80,

type = c("two.sample"), alternative = c("two.sided"))

Notice that I saved the effect size output, and then used it later as effect$d. How can you tell if it should be $ or [ ]? Just try it! Start by using $ and see if anything happens. If not, try [ ].