Practical Significance-Effect Sizes

Neil J. Hatfield

Last Updated: 2023-08-27

Welcome to the latest installment of using R and RStudio. This guide will be looking at statistical inference. As you look through this guide, attempt to calculate the for yourself using the example code. Do this with both the demo data and a new data set (say from HW #1.1).

Getting Started

To get started we need to do two major tasks: load the necessary packages and load our data.

Load Packages and Helper Functions

When you start a new session of R, one of the things that you should do is to load the extra packages that will bring new functions and new capabilities. If you don't load the necessary packages, your code might not work.

For getting effect sizes, we will make use of the `psych` package. In addition, we will also import the ANOVA tools that I have created from my STAT461 GitHub

```
# Load useful packages ----
packages <- c("tidyverse", "psych")
lapply(
  X = packages,
  FUN = library,
  character.only = TRUE,
  quietly = TRUE
)

# Load my helper tools ----
source("https://raw.github.com/neilhatfield/STAT461/master/rScripts/ANOVATools.R")
```

Load Data

For this guide I'm going to use two different data sets: the class demo Oreo data set.

```
# Load Oreo Data ----
oreoData <- read.table(
  file = "https://raw.github.com/neilhatfield/STAT461/master/dataFiles/classDemoOreo.dat",
  header = TRUE,
  sep = ","
)

# Recode Type as a factor
oreoData$Type <- as.factor(oreoData$Type)
```

I've loaded the data and set Type as a factor. We're almost ready to begin.

Practical Significance

When doing inference, there are two kinds of significance that we have to think about: statistical inference and practical significance. Statistical significance deals with whether a model does or does not explain what we've seen in our data; our primary tools are p-values and confidence intervals. For example, statistical significance can tell us whether or not double stuf oreos have twice the mass of créme filling. Practical significance deals with the extent our model explains what we've seen in our data. Here, practical significance can tell us how much more or less créme filling double stuf oreos might have.

To give you another sense of the difference consider the following. Statistical significance can reveal whether a new medication reduces the in-hospital recovery time for a surgery. Practical significance can tell what the reduction in time is so you can decide whether or not the medication is worth the expense.

We will use three of the four types of effect sizes in our course. They are:

- The Difference Family
- Common Language
- Association/Variance Explained Family

For this guide, we will only look at the first two.

The Distance Family

The Distance Family of effect sizes, deals with looking at the distance/difference between two population's location parameters (e.g., Expected Value or Distribution Median). Now you might be thinking "hey, can't I just do the difference in the values of the *SAM* or *Sample Median*?" The answer is not quite. Those differences would describe the differences between your **samples**. Effect sizes (and practical significance in general) deal with extending out to the **population**.

Parametric Case

If you used a parametric shortcut, then you should look at the statistics known as Cohen's D and Hedge's G for effect sizes:

```r
# Getting Cohen's D and Hedge's G ----
effectOut1 <- psych::cohen.d(
  x = oreoData$Filling.Mass,
  group = oreoData$Type
)
```

The `psych::cohen.d` function returns several things that I've stored in the `effectOut1` object (cohen.d, hedges.g, M.dist, r, t, n, p, wt.d, descriptive, se, dict, order, Call). We are only interested in two of these; `effectOut1$cohen.d` and `effectOut1$hedges.g`. While the `hedges.g` is a single number, `cohen.d` consists of three numbers: a lower bound, a point estimate, and an upper bound. While you should look at the bounds, you'll want to report the point estimate.

For my Oreo data, I find that Cohen's D is -13.28 grams (`round(effectOut1$cohen.d[2], digits = 2)`) and Hedge's G is -15.43, -13.06, -10.57 gram (`round(effectOut1$hedges.g, digits = 2)`). Both of these are interpreted as the standardized distance between the location parameters of the Double Stuf Oreo population and Regular Oreo population.

NOTE: you'll want to have a solid understanding of your data set since these values are the result of subtraction and you didn't get to say what the direction was.

When your sample size is small, say $n < 20$, Hedge's G does a better job. However, both of these statistics are upwardly biased; this means that they tend to over-estimate the actual difference.

Nonparametric Case

If you used a Nonparametric method such as the Wilcoxon/Mann-Whitney, Permutation, or Bootstrapping, I would recommend using the Hodges-Lehmann Estimator $\widehat{\Delta}$. You can calculate this by using the `hodgesLehmann` function ***after*** loading my ANOVA Tools. Notice that the function name begins with a period. You will need to have your data set up in two separate vectors/columns (`unstack` is useful here).

```r
# Widen the Data Frame ----
twoColOreo <- unstack(
  x = oreoData,
  form = Filling.Mass ~ Type
)

# Get Hodeges-Lehmann Estimate ----
hlEst <- hodgesLehmann(
  x = twoColOreo$Regular,
  y = twoColOreo$Double.Stuf
)
```

For my data, the Hodges-Lehmann estimate is $\widehat{\Delta} \approx -3.2$ (`round(hlEst, digits = 2)`). This value is the *sample median* créme filling mass difference of all possible pairings of one Regular Oreo and one Double Stuf Oreo.

Common Language

In addition to the Distance Family, you can also use the Common Language effect size, also known as the Probability of Superiority. The way this works is as follows:

Suppose that you find the Probability of Superiority of Double Stuf Oreos' créme filling mass to Regular Oreos is 0.98. This means that if we were to imagine using random sampling to pick one Double Stuf Oreo and one Regular Oreo out of their respective populations, then 98% of the time we do so, the Double Stuf Oreo will have the larger amount of créme filling.

You may get this value by using the `probSup` function from my set of ANOVA Tools. However, you'll need to first calculate Cohen's D and use that value as the input.

```r
# Probability of Superiority ----
## Have to load my helper tools first
ps <- probSup(effectOut1$cohen.d[2])
```

For my data, the Probability of Superiority is 0 (`round(ps, digits = 4)`).

Final Remarks

The above methods work well for two sample situations. Part of what you'll learn in our course are how to get these values and the Variance Explained family within the ANOVA context. So, be on the lookout for these as the semester progresses.

Remember, the best way to improve your skills is by practicing.

- Motor Trend Car Road Tests—access with `data(mtcars)`
- Weight versus Age of Chicks on Different Diets—access with `data(ChickWeight)`
- Effectiveness of Insect Sprays—access with `data(InsectSprays)`
- The Iris data set—access with `data(iris)`
- Palmer Penguin Data—install the `palmerpenguins` package first, then access with `palmerpenguins::penguins`

This concludes this guide to Effect Sizes in R/RStudio.

Code Appendix

```r
# Setting Document Options ----
knitr::opts_chunk$set(
  echo = FALSE,
  warning = FALSE,
  message = FALSE,
  fig.align = "center"
)

# Load useful packages ----
packages <- c("tidyverse", "psych")
lapply(
  X = packages,
  FUN = library,
  character.only = TRUE,
  quietly = TRUE
)

# Load my helper tools ----
source("https://raw.github.com/neilhatfield/STAT461/master/rScripts/ANOVATools.R")


# Load Oreo Data ----
oreoData <- read.table(
  file = "https://raw.github.com/neilhatfield/STAT461/master/dataFiles/classDemoOreo.dat",
  header = TRUE,
  sep = ","
)

# Recode Type as a factor
oreoData$Type <- as.factor(oreoData$Type)

# Getting Cohen's D and Hedge's G ----
effectOut1 <- psych::cohen.d(
  x = oreoData$Filling.Mass,
  group = oreoData$Type
)
# Widen the Data Frame ----
twoColOreo <- unstack(
  x = oreoData,
  form = Filling.Mass ~ Type
)

# Get Hodeges-Lehmann Estimate ----
hlEst <- hodgesLehmann(
  x = twoColOreo$Regular,
  y = twoColOreo$Double.Stuf
)
# Probability of Superiority ----
## Have to load my helper tools first
ps <- probSup(effectOut1$cohen.d[2])
```