FASS512: Confidence intervals. Effect sizes. Hypothesis testing.

Professor Patrick Rebuschat, [p.rebuschat@lancaster.ac.uk](mailto:p.rebuschat@lancaster.ac.uk)

Please work through the following handout at your own pace.

As in the previous handouts, please type the commands in your computer. That is, don’t just read the commands on the paper, please type every single one of them.

Before running the commands, think about what you expect to happen. If you are able to do this, that’s a good sign that you are starting to understand the R language. ☺

This handout assumes that you have completed all previous handouts. If you haven’t, please do this before working on the following handout. Handouts are available on [Moodle](https://modules.lancaster.ac.uk/course/view.php?id=37879).

**References for this handout**

Many of the examples and data files from our class come from these excellent textbooks:

* Andrews, M. (2021). *Doing data science in R*. Sage.
* Brown, D. S. (2021). *Statistics and data visualization using R. The art and practice of data analysis*. Sage.
* Cumming, G. & Calin-Jaegeman, R. J. (2017). *Introduction to the New Statistics: Estimation, Open Science, and Beyond*. Routledge.
* Crawley, M. J. (2013). *The R book*. Wiley.
* Fogarty, B. J. (2019). *Quantitative social science data with R*. Sage.
* Winter, B. (2019). *Statistics for linguists. An introduction using R*. Routledge.

Are you ready? Then let’s start on the next page! ☟

Task 1: *Confidence intervals*

Let’s practice calculating 95% confidence intervals in R. Below, we will look at two days of doing this but, as is so often the case, there many ways of getting confidence intervals in R. ☺

We need to install the tidyverse package.

library(tidyverse)

Remember we assume μ and σ are unknown as this is the most common situation for us. We know the sample statistics, Mean and SD, but not the corresponding population parameters. For this reason, we will use t-distribution rather than z-distribution when computing the confidence intervals.

In the example, we will use (again) the existing, built-in data set called nlschools. The data set contains data from eighth-grade pupils in the Netherlands (hence, NL schools). The sample features data from 2,287 pupils (aged about 11, grade 8) in 132 classes in 131 schools in the Netherlands. There are 2,287 rows (one per participant). Columns include Lang (language test score), IQ (verbal IQ), GS (class size), and SES (social-economic status of pupil's family).

You can inspect the entire data set by typing the following command.

View(MASS::nlschools)

In this task, let’s calculate the **95% CIs for the mean IQ score**.

We place the IQ scores in the data set in a new object, all\_IQ\_scores, and the calculate the mean.

all\_IQ\_scores <- MASS::nlschools$IQ

mean(all\_IQ\_scores)

[1] 11.83406

The mean IQ score in our sample (with over 2,000 pupils) is 11.83. This is our point estimate for the population parameter μ. We now want to calculate the interval estimate, the 95% CI.

Here is one way of doing it.

*Example 1*

To calculate the 95% CIs for the IQ variable, we need to know the following:

* sample mean M
* critical t-value for 95% confidence and for our sample size (corresponds to degrees of freedom, N-1)
* standard error SE
* standard deviation SD
* sample size N

Calculating the sample mean M.

mean <- mean(all\_IQ\_scores) # M = 11.83406

Calculating the sample standard deviation SD.

sd <- sd(all\_IQ\_scores) # SD = 2.06889

We can find out the sample size by using the length() function.

sample\_size <- length(all\_IQ\_scores) # N = 2,287

To calculate the standard error SE we divided the standard deviation by the square root of the sample size.

standard\_error <- (sd / sqrt(sample\_size)) # SE = 0.04326177

Now we can find out the critical t-value for 95% confidence, bearing in mind the sample size 2,287.

standard\_error

[1] 0.04326177

Finally, we also need to determine the critical value for 95% confidence from the t-distribution. The qt() function does this for us. The first argument is the probability .05, which we here divide by two as we are interested in scores in two tails, the left and the right one. The second argument is the degrees of freedom (2,286). Remember: The degrees of freedom is N – 1, so 2,287 – 1 = 2,286. Also remember that the t-value can be positive or negative, depending on the tail in question. Below, we add the abs() function, which ensures that we get a positive value.

t\_value <- abs(qt(p = .05 / 2, df = 2286))

t\_value <- abs(qt(.05 / 2, 2286))

t\_value

[1] 1.961002

We’re finally ready to calculate the 95% CIs for the IQ scores!

These commands tell us the 95% lower and upper limits.

lower\_limit <- mean - (t\_value \* standard\_error)

upper\_limit <- mean + (t\_value \* standard\_error)

intervals\_95 <- c(lower\_limit, upper\_limit)

intervals\_95

[1] 11.74923 11.91890

So, the mean IQ score is 11.83, and the 95% CI [11.75, 11.92].

Do you think this is a precise interval estimate? If so, why? Compare your thoughts with your neighbour.

*Example 2: Using the ESCI for R tools*

As promied, here is another way of calculating confidence intervals.

Bob Calin-Jageman (Geoff Cumming’s co-author for the Introducing the New Statistics textbook) has developed a very helpful tool in R called “ESCI for R”. You can learn more about this by visiting the [repository on the OSF website](https://osf.io/d89xg/wiki/tools:%20esci%20for%20R/); there are also lots of other useful materials.

First, we need to install ESCI for R.

install.packages("devtools")

devtools::install\_github("rcalinjageman/esci")

library(esci)

In the example, we will focus on calculating confidence intervals for a mean.

We do this by using the estimateMean() function. The first argument specifies the data set, the second argument the column of interest (IQ), and the third argument the confidence level (here, 95%).

estimate <- estimateMean(MASS::nlschools,

IQ,

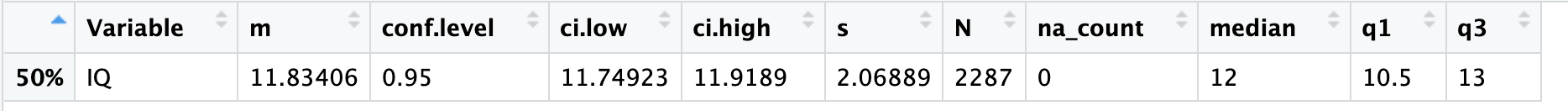
conf.level = .95)

And that’s it.

To make it particularly easy to read off the values, I like to use the View() function with estimate$summary\_data as the argument.

View(estimate$summary\_data)

The output should look like this, see you script editor pane.



Let’s do another example. The data set is from Winter (2019), who in turn took the data from Nettle’s (1999) book Linguistic Diversity. (We have used this a few weeks ago at the very beginning of our R journey.)

library(esci)

First, we load the data set (remember to download from Moodle…) and create the object with the data.

languages <- read.csv('nettle\_1999\_climate.csv')

Let’s have a look at the data.

View(languages)

As we can see, the data set lists 74 countries (Country), and for each country we have information about population size (Population), the area (Area), the mean growing season (MGS, measured in months), and the number of languages spoken (Langs).

Let’s calculate the 95% CIs for number of languages spoken (Langs).

What is the point estimate? What is the 95% confidence interval?

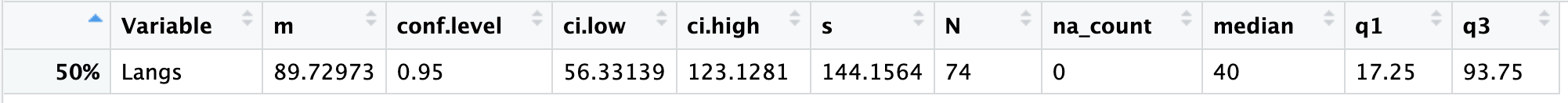
estimate <- estimateMean(languages,

Langs,

conf.level = .95

)

View(estimate$summary\_data)



We see that the mean number of languages is 89.73, 95% CI [56.33, 123.13].

We can also use ggplot to visualize the distribution and the 95% CIs.

plotEstimatedMean(estimate)

Just like in other plots, you can customize the plot with the usual ggplot options (labels, titles, colors, themes, line width and type, etc. etc. etc. etc.).

Below, we first create an object called langs\_CI.

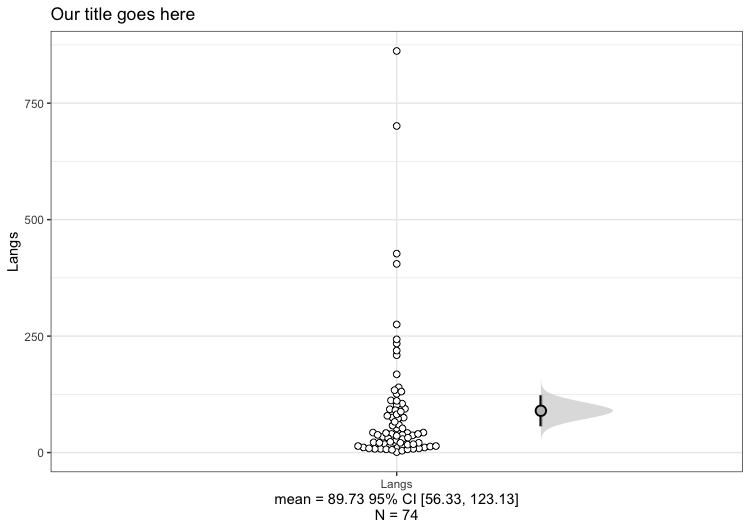
langs\_CI <- plotEstimatedMean(estimate)

Then, we customize the plot by adding the title “Our title goes here”.

langs\_CI <- langs\_CI + ggtitle("Our title goes here")

If you now run the command langs\_CI, you will see your plot. (Note: The comments and arrows won’t be visible, of course, I added them for explanation.)

langs\_CI



The 74 observations, one bubble per country.

The 95% CI. The point estimate (M) is in the center.

The plot below displays the distribution of scores (74 observations, each one transparent bubble) and the 95% CI to the right.

What does the figure tell us about the distribution of Langs, the number of languages spoken across the 74 countries?

Now compare the two data sets.

IQ in Dutch schools: M = 11.83, 95% CI [11.75, 11.92]

Number of languages across 74 countries: M = 89.73, 95% CI [56.33, 123.13].

Observe the point estimates and the interval estimates. What do they tell you about each data set? How confident are you that the population mean μ falls within our interval estimate? How precise are our estimates?

Discuss your answers with your neighbour.Task 2: *Hypothesis tests for continuous variables: two groups*

In this week's session, we start looking at the more formal hypothesis tests for continuous variables. The focus here will not be on the mathematical theory and equations underlying the tests, but rather on their computation using R. The interpretation of these tests, which is what you really need to know about, follows straightforwardly from what we covered today and in the last two sessions.

Let us first load a data set. To illustrate how to different tests, please download and load the language\_scores file from our Moodle site.

language\_scores <- read.table('language\_scores.csv',

header = T,

row.names = 1,

sep = ",")

head(language\_scores)

This table contains three columns, id (student ID), two data variables (exam\_1 and exam\_2, which stand for performance on two language exams), and a grouping variable (study\_level) with two values: 1 and 2. These are merely simulated data for demonstration purposes, but let us assume that 1 = undergraduate and 2 = postgraduate. There are 200 observations (N = 200).

Let’s analyze first the performance of undergraduates and postgraduates on exam 1.

Let’s begin by creating the two subsets of “exam\_1” that correspond to the two levels of study:

exam\_1.ug <- subset(language\_scores, study\_level == 1)$exam\_1

exam\_1.pg <- subset(language\_scores, study\_level == 2)$exam\_1

If you inspect the two new variables, exam\_1.ug and exam\_1.pg, you will see that they only contain the exam 1 scores of UG and PG students, respectively. The rest of the data (e.g., ID, exam 2) has not been included.

We can now proceed to use these new variables in our test commands.

**Choosing the right test**

As discussed in class, selecting an appropriate significance test depends on four basic questions:

1. What kind of data is involved, continuous or categorical?
2. How many groups are being compared?
3. Are the group variances approximately equal?
4. Are the groups independent of one another?

Given the present data set, the answers to questions 1 and 2 are obvious. We know that score is a continuous variable (we will cover categorical tests next week), and we know we are comparing two groups (undergraduates and postgraduates). This leaves questions 3 and 4.

Question 3 (about group variances) strictly refers to the population-level variances of the two groups, not the variances of the actual samples that we have observed. So it is not as easy as just saying that the variance of sample 1 is larger/smaller than the variance of sample 2.

To answer this question, therefore, the most straightforward procedure is to use a rough rule of thumb. We first compute the variances for the two subsamples using the var() function. And then we compute their ratio, with the largest of the two results entered first.

var(exam\_1.ug)

[1] 203.6075

var(exam\_1.pg)

[1] 337.6909

var(exam\_1.pg) / var(exam\_1.ug)

[1] 1.658538

If the resulting ratio is greater than around 2 or 3, then assume “unequal variances”. But since the ratio is less than 2, we can assume equal variances.

Finally, question 4 asks whether the two samples are independent.

Sometimes, we sample data from the same participants on more than one occasion in order to look for changes in a particular variable (“before and after” design). For example, we might want to see if a given teaching method is effective in promoting language development. To investigate this, we need to test the all participants in our study at least twice, once before they start the educational intervention (pre-test) and again after they complete the intervention (post-test). In this case, we say that the samples (pre-test scores and post-test scores) are dependent on one another as they come from the same participants. Dependent data is also known as “paired data”.

In the present data set, however, the data (exam 1) clearly comes two different groups (undergraduate and postgraduate students). For this reason, we say they are independent of one another.

The answers to questions 1 to 4 lead us to the following decision: To compare two independent groups with continuous data and approximately equal variances, we use the standard t-test.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Test name | Groups | Variance |
| Parametric | (standard) t-test | Two independent groups | approximately equal variances |
|  | Welch t-test | Two independent groups | unequal variances |
|  | paired t-test | Two dependent groups |  |
|  | (standard) one-way ANOVA | More than two groups | approximately equal variances |
|  | Welch one-way ANOVA | More than two groups | unequal variances |
| Non-parametric | Wilcoxon-Mann-Whitney test | Two independent groups |  |
|  | Wilcoxon signed rank test | Two dependent groups |  |
|  | Kruskal-Wallis test | More than two groups |  |

Five of the tests in the table above are **parametric** tests. This means that they are based on the theoretical assumption that the statistics of interest follow (roughly) a particular probability distribution, such as the normal distribution. As a consequence of the Central Limit Theorem (which we covered in session 7), this is often a fairly safe assumption, though it becomes increasingly less safe as sample sizes get smaller. Thus, if you want to be on the safe side, you can instead use one of a corresponding set of non-parametric tests.

**Non-parametric** tests do not make any assumptions about the distributions that the statistics of interest follow (though they do assume that the distributions for the two groups being compared have roughly the same shape). The relevant non-parametric alternatives can also be found in the table above.

**Tests for two independent groups**

To run these statistical tests, it is easiest to pull out the two groups first using the subset command and give them names (e.g. score.ug and score.pg), like we did above. There are sometimes other ways of specifying the groups inside a command, but these are only straightforward if your original grouping variable contains two values exactly. This may not always be the case: for instance, you may have included a third value to account for missing data, or you may just want to perform a test on two groups selected out of a larger number. Also, the alternative methods are not available for some commands. The method I show here works in all cases and is therefore the simplest "recipe" to learn at first.

*Standard t-test*

So, to calculate a standard t-test, i.e. assuming equal variances, the command is:

t.test(exam\_1.ug, exam\_1.pg, var.equal = T)

Two Sample t-test

data: score.ug and score.pg

t = 13.863, df = 198, p-value < 2.2e-16

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

27.66479 36.84091

sample estimates:

mean of x mean of y

65.45040 33.19754

As discussed in class, the most important thing here is the p-value (highlighted in red). To report this test in a paper, you would also need to include the t-value (highlighted in green) and the df (highlighted in blue).

The **t-value** is simply the test statistic, i.e. the number that comes out of the mathematical equation underlying this test.

The df stands for "**degrees of freedom**" and is defined as the number ‬of independent values in the calculation that are free to vary. The degrees of freedom are of far less practical importance nowadays, because we no longer have to look up t-values in books of printed tables to find out the associated p-value; but they are still theoretically important and must therefore be reported in your results. For a standard t-test, the df is equal to N – 2 (i.e. N minus two, where N is the combined sample size), but, as we’ll see in a moment, the df is adjusted as part of the Welch procedure for unequal variances. (Here, we were testing two groups of 100, so my combined sample size was 200 and the df is therefore 200 minus 2 = 198.) Note that the t-test output in R also gives you a 95% confidence interval (highlighted in purple) for the difference between the means of the two groups. Finally, on the last line of the output, you have (for reference) the observed means of the two samples.

The outputs of most significance tests in R have a very similar format, so, in what follows, I will only comment on any additional features of importance.

*Welch t-test*

To calculate a Welch t-test, i.e. assuming unequal variances, the command is:

t.test(exam\_1.ug, exam\_1.pg, var.equal = F)

Welch Two Sample t-test

data: exam\_1.ug and exam\_1.pg

t = 13.863, df = 186.55, p-value < 2.2e-16

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

27.66306 36.84265

sample estimates:

mean of x mean of y

65.45040 33.19754

Note here the difference in the degrees of freedom (df), which is now smaller than the one we obtained with the standard t-test, and is also no longer a whole number. Adjusting the df (using a formula derived by B. L. Welch and F. E. Satterthwaite) provides a better approximation to the true distribution of the statistic for the “unequal variances” case. In fact, many researchers prefer always to use the Welch t-test without bothering to check the ratio of variances, because it is more robust than the standard t-test. It is also the default case in R, which means that you don’t actually need to include the flag var.equal = F in the command.

*Wilcoxon-Mann-Whitney test*

To calculate a non-parametric Wilcoxon-Mann-Whitney test , the command is as follows. Note that here you have to specify that you want to see a 95% confidence interval, conf.int = 0.95, otherwise this command will only output a p-value.

wilcox.test(exam\_1.ug, exam\_1.pg, conf.int = 0.95)

Wilcoxon rank sum test with

continuity correction

data: exam\_1.ug and exam\_1.pg

W = 9144, p-value < 2.2e-16

alternative hypothesis: true location shift is not equal to 0

95 percent confidence interval:

26.33691 36.87255

sample estimates:

difference in location

31.58026

Note the different terminology from what we had with the t-tests, i.e., “location shift” and “difference in location”. This reminds us that the Wilcoxon-Mann-Whitney test does not test the null hypothesis that the means (or even the medians) of the two populations are equal. It tests, instead, the null hypothesis that the distributions of the two populations coincide exactly, against the alternative that one of the distributions is shifted by a certain amount to the right or left of the other, as illustrated in the diagram below:

There are no degrees of freedom (df) to report with the Wilcoxon-Mann-Whitney test.

**Tests for paired data**

In our data set, we have scores for two exams from the same participants. We can use scores on exam\_1 and exam\_2 to try out our paired-samples t-tests.

First, let’s create two new objects, one for all scores in exam 1, the other for all scores in exam 2. The scores of undergraduates and postgraduates are now combined.

exam\_1 <- subset(language\_scores)$exam\_1

exam\_2 <- subset(language\_scores)$exam\_2

*Paired t-test*

To calculate a paired t-test, then, the command is:

t.test(exam\_1, exam\_2, paired = T)

Paired t-test

data: exam\_1 and exam\_2

t = 0.12796, df = 199, p-value = 0.8983

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

-4.630673 5.273323

sample estimates:

mean of the differences

0.3213252

Note that the statistic at the bottom (and, by extension, the 95% CI) relate to the mean of the differences between the pairs of observations, and not the difference between the means of the two sets of figures taken collectively. These are not the same thing.

*Wilcoxon signed rank test*

To calculate a Wilcoxon signed rank test, the command is:

wilcox.test(

exam\_1,

exam\_2,

paired = T,

correct = F,

conf.int = 0.95

)

Wilcoxon signed rank test

data: exam\_1 and exam\_2

V = 10246, p-value = 0.811

alternative hypothesis: true location shift is not equal to 0

95 percent confidence interval:

-4.577411 5.957200

sample estimates:

(pseudo)median

0.604464

The statistic used here – the “(pseudo-)median” (and its accompanying 95% CI) – is somewhat analogous to the “mean of the differences” in the case of the paired t-test. It is the median value of the pairwise differences, not the difference between the medians of the two sets of figures.

**Testing directional versus non-directional hypotheses**

All of the commands presented so far are tests of non-directional hypotheses (commonly also known as “two-tailed” tests). This is by far the most common procedure used in everyday research practice.

To perform a test of a directional (or “one-tailed”) hypothesis, you would need to include one of the following additional statements inside the brackets of the relevant test command: alternative = "greater" and alternative = "less".

Consider the following example.

t.test(exam\_1.ug,

exam\_1.pg,

var.equal = T,

alternative = "less")

Two Sample t-test

data: exam\_1.ug and exam\_1.pg

t = 13.863, df = 198, p-value = 1

alternative hypothesis: true difference in means is less than 0

95 percent confidence interval:

-Inf 36.09773

sample estimates:

mean of x mean of y

65.45040 33.19754

The statement alternative = "less" tests a working hypothesis that “the difference in means is less than zero” against a null hypothesis that “the difference is equal to zero”. Similarly, alternative = "greater" tests a working hypothesis that “the difference in means is greater than zero” against a null hypothesis that “the difference is equal to zero”.

*A note on scientific notation and funny-looking p-values*

When working with R, as we saw with an example earlier, you may sometimes get a very funny-looking p-value like the one below.

p-value = 2.197e-05

This notation is used by R to report very small numbers with many decimal places. What the above example means is that the number to the left of the letter "e", i.e., 2.197, really begins in the fifth place after a decimal point; in other words, it is preceded by a decimal point and four zeroes. The actual number is therefore 0.00002197, a p-value very much smaller than the usual significance threshold of 0.05.

Task 3: *Effect sizes: Cohen’s d*

Last but not least, let’s look at the standardized effect size Cohen’s d. As mentioned in class, Cohen’s d is often used to measure the strength of a difference between two means. Since it’s a standardized score, we can use it to compare mean differences across many different studies, which is why Cohen’s d is used in meta-analyses.

Remember, Cohen (1988) suggested the following interpretation of effect sizes:

* A value of **0.2** represents a small effect size.
* A value of **0.5** represents a medium effect size.
* A value of **0.8** or above represents a large effect size.

The easiest way to calculate Cohen’s d is perhaps by using the cohen.d() function from the effsize package. Let’s use the language\_scores data set as before.

library(effsize)

In the first example, we will look at performance on exam 1 and exam 2. The mean for exam 1 is 49.32, the mean for exam 2 it’s 49.00, and the mean difference is, of course, only 0.32. Hardly any difference at all. What does Cohen’s d look like?

cohen.d(exam\_1, exam\_2)

Cohen's d

d estimate: 0.01264916 (negligible)

95 percent confidence interval:

lower upper

-0.1839470 0.2092454

As you can see, for Mdiff = 0.32 Cohen’s d is 0.01. Predictably, a “negligible” effect, as R tells us.

Let’s now calculate the effect size for the performance of undergraduates and postgraduates on exam 1. The mean score for undergraduates on exam 1 is 65.45, for postgraduates it’s 33.20, and the mean difference is 32.25. And Cohen’s d?

cohen.d(exam\_1.ug, exam\_1.pg)

Cohen's d

d estimate: 1.96049 (large)

95 percent confidence interval:

lower upper

1.621161 2.299820

As we can see, Cohen’s d is 1.96, a “large” effect. Helpfully, the cohen.d() function also gives us the 95% confidence intervals for the effect size, 95% CI [1.62, 2.30]. That is, we are 95% confident that the population effect size will be between 1.62 (a large effect) and 2.30 (an even large effect). Either way, the effect is large.

Below is another way of calculating Cohen’s d. If we wish to make life more difficult, we can use this way rather than the cohen.d() function from the effsize package. I am adding this example just to illustrate (again) what information is used to compute the effect size.

In the example, we again compare undergraduate and postgraduate performance on exam 1.

As you can see, to compute Cohen’s d, we need the two means and the two standard deviations (to compute the pooled standard deviations). We can input the data as follows.

our\_effect\_size <-

(mean(exam\_1.ug) - mean(exam\_1.pg)) /

sqrt(((sd(exam\_1.ug) ^ 2) + (sd(exam\_1.pg) ^ 2)) / 2)

If we now execute the command, we get the correct Cohen’s d value.

our\_effect\_size

[1] 1.96049

*How to report and interpret this*

The style guides of different journals will give tell you how to report the results of statistical tests. Below is an example based on the widely-used style guide of the American Psychological Association (APA).

“The mean score on exam 1 for undergraduates was 65.45 (SD = 14.27) and for postgraduates 33.20 (SD = 18.38). Cohen’s d is 1.56, 95% CI [1.62, 2.30], which represents a large effect size (Cohen, 1988). Finally, the independent-samples t-test shows this difference is significant, t = 13.863, df = 198, p-value < .001.”

Note: The actual p-value is p = 2.2e-16, i.e. 0.00000000000000022, so that’s definitely below .05 ☺ But as mentioned in class, we should report the exact p-value as long as it is above .01. Once it’s below this value, we can use inequalities, p < .01 and p < .001.

Below is a visualization of effect size 1.96 from the ESCI d picture tool. As you can see, 1.96 is quite a large difference. Note also the N required for sufficient precision. What do you think this means?

