Lab 8: One-sample and Paired Samples *t*-tests

Table of Contents

Unquote the following install line to install the papaja package, which is used for preparing APA style reports.

# install.packages("papaja")   
library(papaja) # for reporting results

Today’s lab will guide you through the process of conducting a [One-Sample *t*-test](#one) and a [Paired Samples *t*-test](#paired).

To quickly navigate to the desired section, click one of the following links:

1. [One-Sample *t*-test](#one)
2. [Paired Samples *t*-test](#paired)
3. [Minihacks](#minihacks)

As always, there are [minihacks](#minihacks) to test your knowledge.

# One-Sample *t*-test

## The Data

In our first example, we will be analyzing data from Fox and Guyer’s (1978) anonymity and cooperation study. The data is included in the {carData} package and you can see information about the data set using ?Guyer. Twenty groups of four participants each played 30 trials of the the prisoner’s dilemma game. The number of cooperative choices (cooperation) made by each group were scored out of 120 (i.e., cooperative choices made by 4 participants over 30 trials).

Run the following code to load the data into your global environment.

#install.packages('carData') #if you need to install this package  
# load data   
library(carData) # includes the Guyer data set  
data <- Guyer

## What is a One-Sample *t*-test?

A One-Sample *t*-test tests whether an obtained sample mean came from a population with a known mean. Let’s consider the cooperation data form Guyer. We might be interested in whether people cooperate more or less than 50% of the time. If people cooperate 50% of the time, then out of the 120 trials, we would expect people to have chosen to cooperate 60 times. Let’s compare our actual results (our sample mean) to the mean we would expect if our sample was obtained from a population with a mean of 60.

**Question:** How does this differ from a Z-test?

**Question:** What are our null and alternative hypotheses?

**Question:** What are our assumptions?

**Question:** What kind of sampling distribution do we have? What does it represent?

**Question:** How do we calculate our test statistic? What does it represent?

## A One-Sample *t*-test

### The null and alternative hypotheses

The null hypothesis states that..

The alternative hypothesis states that..

### Assumptions

The assumptions include..

1. *Normality*: Assume that the population distribution is normally distributed.
2. *Independence*: Assume that the observations are independent of one another.

### Sampling Distribution of Means

For a one-sample t-test, the sampling distribution is a *t* distribution. It represents all the possible sample means we could expect to obtain if we randomly obtained a sample of size *n* from the population that is described by the null hypothesis.

### The One-Sample t-Statistic

The t-statistic is calculated by calculating the difference between the sample mean and the population mean predicted by the null hypothesis, divided by the estimated standard error. In other words, this statistic tells us how far away our sample mean is from the mean of our sampling distribution (which describes what we expect to find *if the null hypothesis is true*), in standard error units.

Use R to calculate the t-statistic for this example “from scratch”:

#your code here:

## Conducting the One-Sample t-Test in R

There are two useful functions for conducting a one-sample *t*-test in R. The first, is called t.test() and it is automatically loaded as part of the {stats} package when you first open R. To run a one-sample *t*-test using t.test(), you provide the function the column of the data you are interested in (e.g., x = data$cooperation) and the mean value you want to compare the data against (e.g., mu = 60).

t.test(x = data$cooperation, mu = 60)

##   
## One Sample t-test  
##   
## data: data$cooperation  
## t = -3.6624, df = 19, p-value = 0.001656  
## alternative hypothesis: true mean is not equal to 60  
## 95 percent confidence interval:  
## 41.61352 54.98648  
## sample estimates:  
## mean of x   
## 48.3

As part of the output, you are provided the mean of cooperation, the *t*-statistic, the degrees of freedom, the *p*-value, and the 95% confidence interval. The values outputted by t.test() should be exactly the same as the values we calculated. Unfortunately, we did not get a measure of the effect size.

The oneSampleTTest() function from the the {lsr} package includes Cohen’s *d* automatically, but the downside is that you have to load the package separately.

oneSampleTTest(x = data$cooperation, mu = 60)

##   
## One sample t-test   
##   
## Data variable: data$cooperation   
##   
## Descriptive statistics:   
## cooperation  
## mean 48.300  
## std dev. 14.287  
##   
## Hypotheses:   
## null: population mean equals 60   
## alternative: population mean not equal to 60   
##   
## Test results:   
## t-statistic: -3.662   
## degrees of freedom: 19   
## p-value: 0.002   
##   
## Other information:   
## two-sided 95% confidence interval: [41.614, 54.986]   
## estimated effect size (Cohen's d): 0.819

As you can see from the output, oneSampleTTest() provides you all of the information that t.test() did, but it also includes Cohen’s *d*.

## Interpretation and Write-Up

You can use the ‘apa\_print()’ function from the papaja package to efficiently write-up your results in apa format.

#First I'll run my t-test again, but this time I'll save it to a variable:  
  
t\_output <- t.test(x = data$cooperation, mu = 60)  
  
#Let's look at our output options  
papaja::apa\_print(t\_output)

## $estimate  
## [1] "$M = 48.30$, 95\\% CI $[41.61, 54.99]$"  
##   
## $statistic  
## [1] "$t(19) = -3.66$, $p = .002$"  
##   
## $full\_result  
## [1] "$M = 48.30$, 95\\% CI $[41.61, 54.99]$, $t(19) = -3.66$, $p = .002$"  
##   
## $table  
## A data.frame with 5 labelled columns:  
##   
## estimate conf.int statistic df p.value  
## 1 48.30 [41.61, 54.99] -3.66 19 .002  
##   
## estimate : $M$   
## conf.int : 95\\% CI   
## statistic: $t$   
## df : $\\mathit{df}$   
## p.value : $p$   
## attr(,"class")  
## [1] "apa\_results" "list"

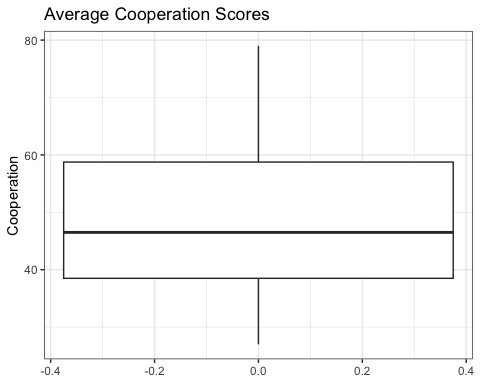
Depending on what object you give to apa\_print, it may give you different commonly reported statistics from that test. You can access these with a ‘$’. Here is an example of how I might report this:

“Average cooperation (, 95% CI ) was significantly less than 60, , .

## Visualization

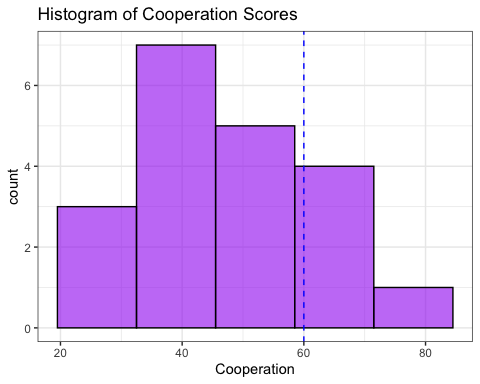
To plot this, we’re just plotting one variable. You already know how to do this! Here is one option. I’ve plotted a boxplot below.

ggplot(data = data, aes(y = cooperation)) +  
 geom\_boxplot() +  
 theme\_bw() +  
 labs(title = "Average Cooperation Scores",  
 y = "Cooperation")



Or I might want to plot a histogram. I used the additional geom, ‘geom\_vline’ to plot a line representing the theorized population mean.

ggplot(data = data) +  
 geom\_histogram(aes(x = cooperation), fill = "purple", color = "black", alpha = .6, bins = 5) +  
 geom\_vline(xintercept = 60, color = "blue", linetype = "dashed") +  
 theme\_bw() +  
 labs(title = "Histogram of Cooperation Scores",  
 x = "Cooperation")



# Paired-Sample *t*-test

To illustrate how paired-samples *t*-tests work, we are going to walk through [an example from your textbook](https://learningstatisticswithr-bookdown.netlify.com/ttest.html#pairedsamplesttest). In this example, the data comes from Dr. Chico’s introductory statistics class. Students in the class take two tests over the course of the semester. Dr. Chico gives notoriously difficult exams with the intention of motivating her students to work hard in the class and thus learn as much as possible. Dr. Chico’s theory is that the first test will serve as a “wake up call” for her students, such that when they realize how difficult the class actually is they will be motivated to study harder and earn a higher grade on the second test than they got on the first test.

You can load in the data from this example by running the following code:

chico\_wide <- import("https://raw.githubusercontent.com/uopsych/psy611/master/labs/resources/lab9/data/chico\_wide.csv")  
  
# long format  
chico\_long <- import("https://raw.githubusercontent.com/uopsych/psy611/master/labs/resources/lab9/data/chico\_long.csv")

**Note:** You should now have 2 versions of the same data set loaded into your global environment. The only difference in these versions of the data is their “shape” – one is “wide” and the other is “long”. In the wide form, every row corresponds to a unique *person*; in the long form, every row corresponds to a unique *observation* or *measurement*.

* To get a sense of the differences between these two versions of the data, see below:

chico\_wide

## id grade\_test1 grade\_test2  
## 1 student1 42.9 44.6  
## 2 student2 51.8 54.0  
## 3 student3 71.7 72.3  
## 4 student4 51.6 53.4  
## 5 student5 63.5 63.8  
## 6 student6 58.0 59.3  
## 7 student7 59.8 60.8  
## 8 student8 50.8 51.6  
## 9 student9 62.5 64.3  
## 10 student10 61.9 63.2  
## 11 student11 50.4 51.8  
## 12 student12 52.6 52.2  
## 13 student13 63.0 63.0  
## 14 student14 58.3 60.5  
## 15 student15 53.3 57.1  
## 16 student16 58.7 60.1  
## 17 student17 50.1 51.7  
## 18 student18 64.2 65.6  
## 19 student19 57.4 58.3  
## 20 student20 57.1 60.1

chico\_long

## id time grade  
## 1 student1 test1 42.9  
## 2 student2 test1 51.8  
## 3 student3 test1 71.7  
## 4 student4 test1 51.6  
## 5 student5 test1 63.5  
## 6 student6 test1 58.0  
## 7 student7 test1 59.8  
## 8 student8 test1 50.8  
## 9 student9 test1 62.5  
## 10 student10 test1 61.9  
## 11 student11 test1 50.4  
## 12 student12 test1 52.6  
## 13 student13 test1 63.0  
## 14 student14 test1 58.3  
## 15 student15 test1 53.3  
## 16 student16 test1 58.7  
## 17 student17 test1 50.1  
## 18 student18 test1 64.2  
## 19 student19 test1 57.4  
## 20 student20 test1 57.1  
## 21 student1 test2 44.6  
## 22 student2 test2 54.0  
## 23 student3 test2 72.3  
## 24 student4 test2 53.4  
## 25 student5 test2 63.8  
## 26 student6 test2 59.3  
## 27 student7 test2 60.8  
## 28 student8 test2 51.6  
## 29 student9 test2 64.3  
## 30 student10 test2 63.2  
## 31 student11 test2 51.8  
## 32 student12 test2 52.2  
## 33 student13 test2 63.0  
## 34 student14 test2 60.5  
## 35 student15 test2 57.1  
## 36 student16 test2 60.1  
## 37 student17 test2 51.7  
## 38 student18 test2 65.6  
## 39 student19 test2 58.3  
## 40 student20 test2 60.1

We will work with both versions of this dataset today.

**Question:** What is a paired samples t-test?

**Question:** What are our null and alternative hypotheses?

**Question:** What are our assumptions?

### The null and alternative hypotheses

The null hypothesis states that..

The alternative hypothesis states that..

### Assumptions

The assumptions include..

1. *Normality*: Assume that the population distribution is normally distributed.
2. *Independence*: Assume that the participants are independent of one another.

### Sampling Distribution of Mean Difference Scores

For a paired samples t-test, the sampling distribution is a *t* distribution. It represents all the possible mean difference scores we could expect to obtain if we randomly obtained a sample of size *n* from the population that is described by the null hypothesis.

### The Paired Samples t-Statistic

The t-statistic for the paired samples t-test takes the average difference score (the difference between two pairs of related scores), and divides by the estimated standard error of the sampling distribution.

## Paired Samples t-test

### Option 1: One-sample *t*-test of difference scores

One way to conduct a paired samples t-test in R is to use the t.test() function we used earlier, but the variable you give as an input is the *difference scores* between two conditions of related scores.

The example immediately below uses the t.test() function in the {stats} package to conduct the one-sample *t*-test of the difference scores.

# First, construct a new variable of difference scores   
chico\_wide$diff <- chico\_wide$grade\_test2 - chico\_wide$grade\_test1  
  
# Then, pass the difference scores to the t.test function  
t.test(x = chico\_wide$diff, mu = 0)

##   
## One Sample t-test  
##   
## data: chico\_wide$diff  
## t = 6.4754, df = 19, p-value = 0.000003321  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## 0.9508686 1.8591314  
## sample estimates:  
## mean of x   
## 1.405

You could also use the oneSampleTTest() function from the {lsr} package to conduct the one-sample *t*-test of the difference scores.

oneSampleTTest(x = chico\_wide$diff, mu = 0)

##   
## One sample t-test   
##   
## Data variable: chico\_wide$diff   
##   
## Descriptive statistics:   
## diff  
## mean 1.405  
## std dev. 0.970  
##   
## Hypotheses:   
## null: population mean equals 0   
## alternative: population mean not equal to 0   
##   
## Test results:   
## t-statistic: 6.475   
## degrees of freedom: 19   
## p-value: <.001   
##   
## Other information:   
## two-sided 95% confidence interval: [0.951, 1.859]   
## estimated effect size (Cohen's d): 1.448

## Option 2: Paired samples *t*-test

You can also conduct a paired samples t-test using the t.test() function with the raw scores from two related conditions if you set the argument paired = TRUE. The results will be exactly the same as running the one sample *t*-test on the difference scores.

t.test(x = chico\_wide$grade\_test1,  
 y = chico\_wide$grade\_test2,  
 paired = TRUE)

##   
## Paired t-test  
##   
## data: chico\_wide$grade\_test1 and chico\_wide$grade\_test2  
## t = -6.4754, df = 19, p-value = 0.000003321  
## alternative hypothesis: true mean difference is not equal to 0  
## 95 percent confidence interval:  
## -1.8591314 -0.9508686  
## sample estimates:  
## mean difference   
## -1.405

Or you can use the pairedSamplesTTest() function from the {lsr} package.

pairedSamplesTTest(formula = ~ grade\_test2 + grade\_test1, # one-sided formula  
 data = chico\_wide) # wide format

##   
## Paired samples t-test   
##   
## Variables: grade\_test2 , grade\_test1   
##   
## Descriptive statistics:   
## grade\_test2 grade\_test1 difference  
## mean 58.385 56.980 1.405  
## std dev. 6.406 6.616 0.970  
##   
## Hypotheses:   
## null: population means equal for both measurements  
## alternative: different population means for each measurement  
##   
## Test results:   
## t-statistic: 6.475   
## degrees of freedom: 19   
## p-value: <.001   
##   
## Other information:   
## two-sided 95% confidence interval: [0.951, 1.859]   
## estimated effect size (Cohen's d): 1.448

### Long vs. wide format

Note that in the example above, using the pairedSamplesTTest() function, we used the wide format data. When using wide data with pairedSamplesTTest(), you enter a one-sided formula that contains your two repeated measures conditions (e.g. ~ grade\_test2 + gradte\_test1).

The pairedSamplesTTest() function can also be used with long data. In this case, you must use a two-sided formula: outcome ~ group. You also need to specify the name of the ID variable. Note that the grouping variable must also be a factor.

# grouping variable (time) must be a factor  
chico\_long <- chico\_long %>%   
 mutate(time = as.factor(time),   
 id = as.factor(id))  
  
pairedSamplesTTest(formula = grade ~ time, # two-sided formula  
 data = chico\_long, # long format  
 id = "id") # name of the id variable

##   
## Paired samples t-test   
##   
## Outcome variable: grade   
## Grouping variable: time   
## ID variable: id   
##   
## Descriptive statistics:   
## test1 test2 difference  
## mean 56.980 58.385 -1.405  
## std dev. 6.616 6.406 0.970  
##   
## Hypotheses:   
## null: population means equal for both measurements  
## alternative: different population means for each measurement  
##   
## Test results:   
## t-statistic: -6.475   
## degrees of freedom: 19   
## p-value: <.001   
##   
## Other information:   
## two-sided 95% confidence interval: [-1.859, -0.951]   
## estimated effect size (Cohen's d): 1.448

## Interpretation and Write-Up

#First I'll run my t-test again, but this time I'll save it to a variable:  
pairedtoutput <- t.test(x = chico\_wide$grade\_test1,  
 y = chico\_wide$grade\_test2,  
 paired = TRUE)  
  
#Let's look at our output options  
apa\_print(pairedtoutput)

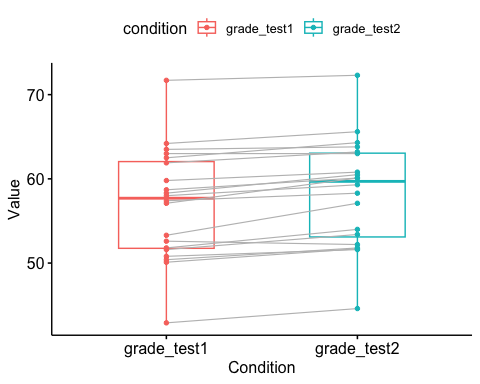
## $estimate  
## [1] "$M\_D = -1.40$, 95\\% CI $[-1.86, -0.95]$"  
##   
## $statistic  
## [1] "$t(19) = -6.48$, $p < .001$"  
##   
## $full\_result  
## [1] "$M\_D = -1.40$, 95\\% CI $[-1.86, -0.95]$, $t(19) = -6.48$, $p < .001$"  
##   
## $table  
## A data.frame with 5 labelled columns:  
##   
## estimate conf.int statistic df p.value  
## 1 -1.40 [-1.86, -0.95] -6.48 19 < .001  
##   
## estimate : $M\_D$   
## conf.int : 95\\% CI   
## statistic: $t$   
## df : $\\mathit{df}$   
## p.value : $p$   
## attr(,"class")  
## [1] "apa\_results" "list"

* Practice using the output from ‘apa\_print’ to write your results here:

## Plotting paired-samples data

When plotting paired samples data, we want some way to clearly represent the repeated measures structure of the data. One way to do this is to draw a line between each pair of data points. This can be done with the ggpaired() function from {ggpubr}.

# wide format  
ggpaired(chico\_wide,   
 cond1 = "grade\_test1", # first condition  
 cond2 = "grade\_test2", # second condition  
 color = "condition",   
 line.color = "gray",   
 line.size = 0.4)



* If you want to make the plot with the long format of the data, you can do the following:

# long format  
ggpaired(chico\_long,   
 x = "time", # grouping variable  
 y = "grade", # outcome variable  
 color = "time",   
 line.color = "gray",   
 line.size = 0.4)

# Minihacks

## Minihack 1:

You are developing a scale to quantify how delicious a food is. You design it such that 0 is as delicious as cement, 25 is delicious as an olive, and 50 is delicious as a strawberry. You aren’t sure exactly where pancakes should fall, but you’re interested in whether they differ from the deliciousness of strawberries. You order takeout pancakes from 17 local diners and score each.

Generate the data by running the the following code:

# set seed for reproducibility  
set.seed(42)  
  
# load data  
pancakes <- data.frame("id" = 1:17,  
 "Delicious" = rnorm(17, 55, 10))

1. Run an appropriate test based on your research question.

#Your code here

1. Report and interpret your results by filling in the embedded r code below.

We found that the average deliciousness of pancakes () was significantly higher compared to strawberries.

1. Plot a histogram of your data.

#Your code here

## Minihack 2:

A clinical psychologist wants to know whether a new cognitive-behavioral therapy (CBT) program helps alleviate anxiety. He enrolls 12 individuals diagnosed with an anxiety disorder in a 6-week CBT program. Participants are given an Anxiety Scale before they begin and after they complete treatment.

Import the data by running the following code:

cbt\_data <- import("https://raw.githubusercontent.com/uopsych/psy611/master/labs/resources/lab9/data/cbt\_data.csv")

1. Run a paired-samples *t*-test to determine whether participants’ anxiety scores changed from before to after the CBT treatment.

#Your code here

1. Verify your results using a one sample t-test on difference scores

#Your code here

1. Report and interpret your results using r code embedded in text.
2. Obtain a table of descriptive statistics.

#Your code here

1. Plot the data using ggpaired().

#Your code here  
cbt\_data

## id before after  
## 1 1 10 5  
## 2 2 12 6  
## 3 3 8 7  
## 4 4 12 8  
## 5 5 15 10  
## 6 6 18 11  
## 7 7 6 3  
## 8 8 5 4  
## 9 9 6 6  
## 10 10 10 7  
## 11 11 12 8  
## 12 12 15 9

1. **BONUS:** Recreate this same type of plot without using ggpaired

#Your code here