

# HW 6: t-tests and effect sizes

*partner names*

*type date*

## Part A

### General Questions

1. **QUESTION:** Describe the purpose of a one-sample  $t$ -test.

*ANSWER:* The one-sample  $t$ -test is used to test whether a sample mean is equivalent to (or different from) a known population mean when the population standard deviation is not known. The dependent variable must be interval or ratio in scale and the distribution should be approximately normal in shape.

2. **QUESTION:** Describe the purpose of a paired-samples  $t$ -test.

*ANSWER:* The paired-samples  $t$ -test for is used to test whether the mean difference score for variable pairs is equivalent to (or different from) 0. The dependent variables must be interval or ratio in scale and the distribution of differences should be approximately normal in shape.

## Part B

### Before you begin

This homework exercise involves having you answer some questions, write some code, and create a nice HTML file with your results. When asked different questions, simply either type your coded or written responses after the ANSWER message. When asked to write code to complete sections, type your code in the empty code blocks that follow the ANSWER message (between the back ticks). After adding that code, you must make sure that it will execute. So remember to read the content and run each line of your code as you write it so that you know it executes correctly. If your code does not execute, then RMarkdown won't know what you are telling it to do and your HTML file will not be produced. Also, don't create your HTML file until you finish and know that all of your code works correctly.

### 1. Installing and using libraries in RStudio

1.1. Use the RStudio interface to install packages/libraries. Go to the Tools option and select Install Packages. Type the package name(s) correctly using the proper letter casing. Also, make sure that you *check the box to Install Dependencies*. Do not install with code.

- Mac users, most of you should already have XQuartz installed from the first R class, but please make sure that you have it downloaded and installed in your Applications folder. See the original download instruction file if you need help.

## 1.2. Key functions used for this assignment:

- `as.data.frame()` for combining vectors into a data frame; built-in
- `histogram()` for histograms; lattice library
- `lillie.test()` for testing normality; nortest library
- `options()` for changing the scientific notation display (e.g., `scipen = 999`)
- `qqnorm()` and `qqline()` for examining normality; built-in library
- `shapiro.test()` for testing normality; built-in stats library
- `t.test()` for calculating t-tests; built-in stats library

## 2. Loading libraries

If you know what libraries you will use for your code, you can load them now. Use the `library()` function to load the following libraries: lattice, nortest.

ANSWER:

```
library(lattice)
library(nortest)
```

Change from scientific notation if you want.

```
options(scipen=999) #options(scipen=0)
```

## 1. Overview of the one-sample *t*-test

The procedures for testing hypotheses described up to this point comparing a group of scores to a known population (e.g., z-test). For many research questions, however, you do not have without any direct information about populations you wish to study. In such cases, you can estimate population parameters using sample statistics. These kinds of research situations are very common in behavioral and social science research, where usually the only information available is from the samples. You want to compare the mean of the scores in a sample to a population for which the mean is known but the variance is unknown. Hypothesis testing in this situation is called a one-sample *t*-test.

The *one-sample t-test*, like the z-test, the one-sample *t*-test compares the mean of a single sample to a hypothesized population value (e.g.,  $\mu$ ). Unlike the other *t*-tests (e.g., independent and dependent), this test analyzes data from only one dependent variable.

For example: To test this we are going to create a data set of 20 first year resident female doctors resting systolic blood pressure drawn at random from one area,\*\*

If you don't already have data in a data frame, you can combine values into vector variables and then combine them into a data frame object using the built-in `data.frame()` function. We will create two vectors by combining values using the `c()` function to combine them.

```

Id <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20)
BP <- c(128, 127, 124, 126, 144, 142, 133, 140, 132, 131, 130, 132, 149, 122, 139, 119, 136, 129, 126, 128)

BLOOD <- data.frame(Id, BP) # pass the two vectors into the function to create a data frame object.

# Look at the data frame
str(BLOOD)

```

```

## 'data.frame':    20 obs. of  2 variables:
##  $ Id: num  1 2 3 4 5 6 7 8 9 10 ...
##  $ BP: num 128 127 124 126 144 142 133 140 132 131 ...

```

BLOOD

```

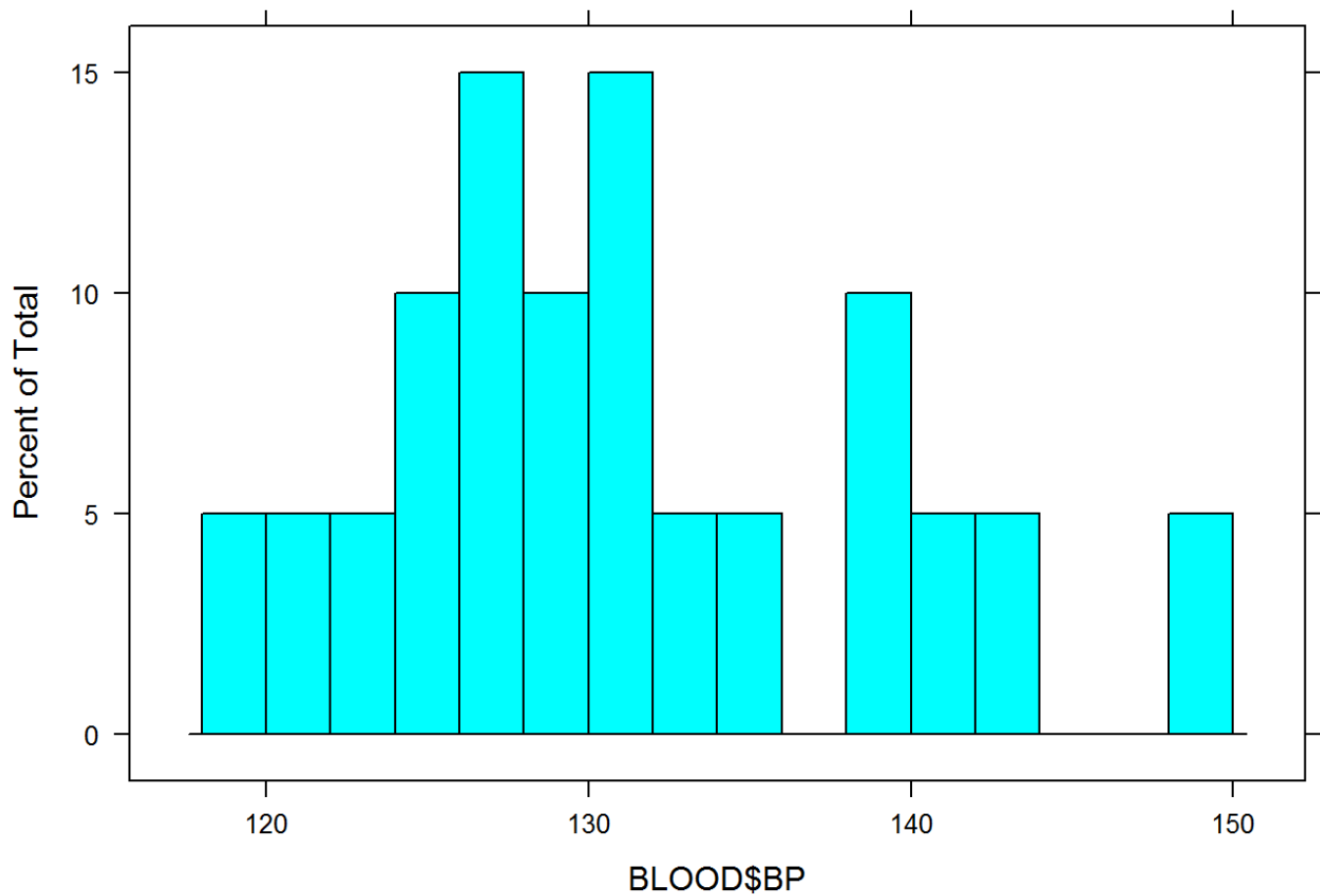
##      Id  BP
## 1     1 128
## 2     2 127
## 3     3 124
## 4     4 126
## 5     5 144
## 6     6 142
## 7     7 133
## 8     8 140
## 9     9 132
## 10    10 131
## 11    11 130
## 12    12 132
## 13    13 149
## 14    14 122
## 15    15 139
## 16    16 119
## 17    17 136
## 18    18 129
## 19    19 126
## 20    20 128

```

## Examining Data and Assumptions of $t$ -tests

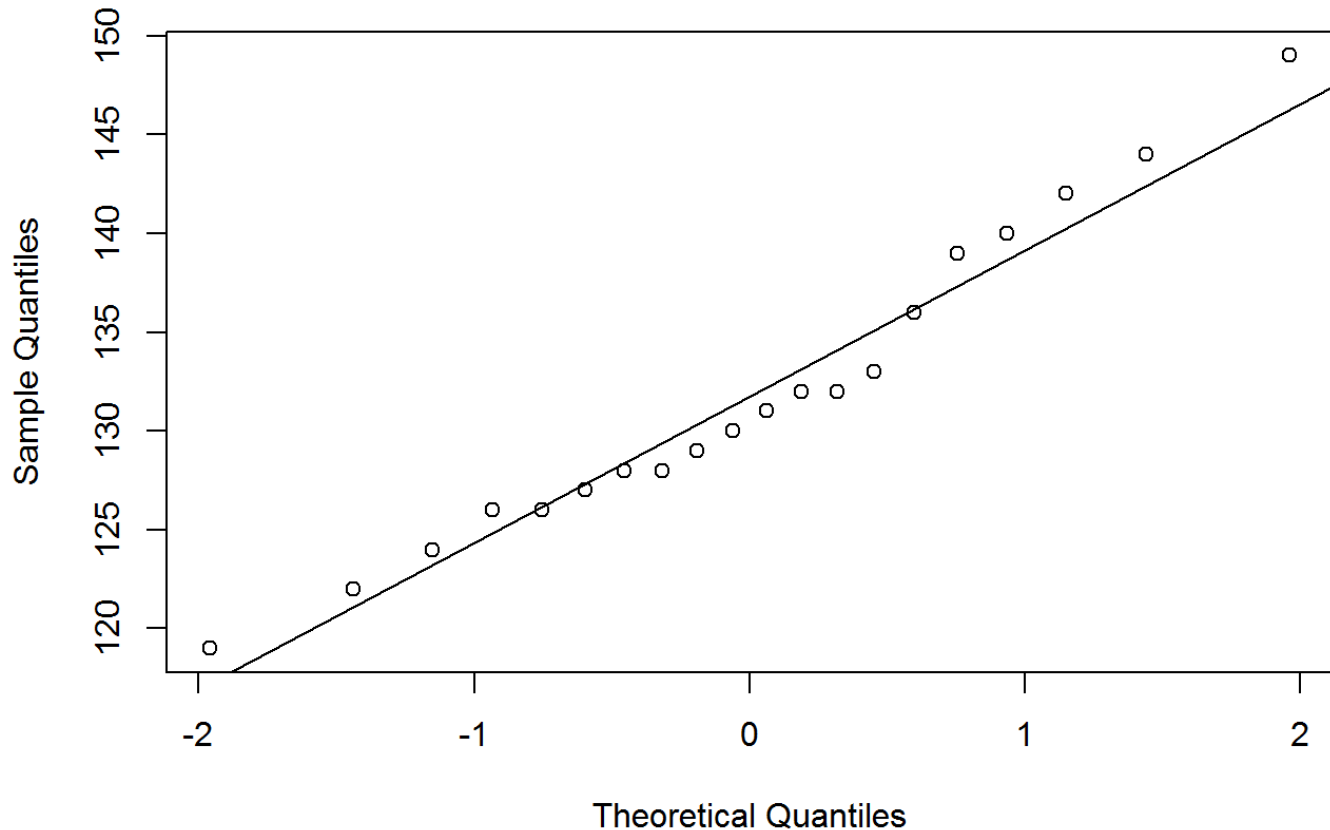
1.1. After determining that your data are either interval or ratio in scale, you should examine whether normality of our data is a reasonable assumption. In the process, you can look for outliers in the data, which you should normally remove, but you don't have to here.

```
histogram(~BLOOD$BP, breaks = 20) # use the breaks argument to manipulate how many breaks
to put in your histogram so that you can see the distribution better than the default op
tion
```



```
# examine visually; normal fit using quantile-quantile plots
qqnorm(BLOOD$BP); qqline(BLOOD$BP)
```

## Normal Q-Q Plot



1.2. You can also use different statistical tests to examine whether the shape of distribution differs from the shape of a normal distribution. Some tests are included in the `nortest` library. We will use the Shapiro-Wilk test using `shapiro.test()` and the Lilliefors (Kolmogorov-Smirnov) test for normality using the `lillie.test()` function.

```
# Load the nortest library if not loaded
#library(nortest)

# pass the variable of interest into the function
shapiro.test(BLOOD$BP) # The p value is > .05
```

```
##
##  Shapiro-Wilk normality test
##
## data:  BLOOD$BP
## W = 0.96378, p-value = 0.6217
```

```
# pass the variable into the function
lillie.test(BLOOD$BP) # The p value is > .05
```

```
##  
##  Lilliefors (Kolmogorov-Smirnov) normality test  
##  
## data:  BLOOD$BP  
## D = 0.14227, p-value = 0.3589
```

To report the outcome of the Shapiro-Wilk test, the general form is:

S-W(df) = test value,  $p < \text{or} > \alpha$

By substitution and making sure to italicize S-W and  $p$ , we can say:

The distribution of resting systolic blood pressure for the sample of first year resident female doctors did not differ statistically from that of a normal distribution, S-W(19) = .96,  $p > .05$ . Normality is a reasonable assumption.

If your data adhere to a fairly normal distribution, you can compare your sample mean to a population mean using the  $t$ -test. In order to calculate a one-sample  $t$ -test you must obtain the mean and standard deviation of the sample. You could do this manually as you have before, but functions will also accomplish this.

```
BP.mean <- mean(BLOOD$BP)  
BP.mean
```

```
## [1] 131.85
```

```
BP.sd <- sd(BLOOD$BP)  
BP.sd
```

```
## [1] 7.741039
```

Knowing your sample mean and standard deviation you also must know the mean of the population you are testing. From previous large studies of women drawn at random from the healthy general public, a resting systolic blood pressure of 120 mm Hg was predicted as the population mean for the relevant age group.

A data summary: - Sample mean = 131.85 - Population mean = 120 - Sample size  $n = 20$  - Sample sd = 7.741039

Below is the formula to conduct a one-sample  $t$ -test:

$$t = (\text{Sample mean} - \text{population mean}) / (\text{unbiased sample sd} / \sqrt{n})$$

By substitution,

$$t = (131.85 - 120) / (7.741039 / \sqrt{20})$$

You could code your own formula:

```
(BP.mean - 120) / (BP.sd / sqrt(20))
```

```
## [1] 6.845956
```

However, there is an easier way to calculate the  $t$ -value and obtain the corresponding  $p$ -value. The `t.test()` function from the built-in stats library can be used to conduct the one-sample  $t$ -test. Of course, you will need to specify variable and the population mean arguments in the `t.test()` function.

The general formula is: `t.test(X, mu = ?)`

```
t.test(BLOOD$BP, mu = 120)
```

```
##  
## One Sample t-test  
##  
## data: BLOOD$BP  
## t = 6.846, df = 19, p-value = 0.000001561  
## alternative hypothesis: true mean is not equal to 120  
## 95 percent confidence interval:  
## 128.2271 135.4729  
## sample estimates:  
## mean of x  
## 131.85
```

## 2. Reporting the test statistic values

Based on the R output, you will see the variable for which you calculated the sample mean and estimated standard deviation (S), the  $t$ -value based on the calculation, the degrees of freedom for the mean ( $n-1$ ), and the  $p$ -value correspond to finding a difference between the sample and population means if they were supposed to be equal to each other under  $H_0$ .

When you calculate a  $t$ -test and report the results of the statistic, you will need to specify the  $t$ -value and the degrees of freedom. The  $df$  for the one-sample  $t$ -test is  $n - 1$ .

To report the outcome of the test, the general form is:

$t(df) = t\text{-value}, p < \text{or } > \alpha$

By substitution and making sure to italicize  $t$  and  $p$ , we can say:

First year resident female doctors had resting systolic blood pressures that was statically higher than that of the population,  $t(19) = 6.85, p < .05$ .

If you used the  $p$ -value to make decisions about  $H_0$ , you could interpret the data as evidence for the sample (its mean) being different from the population mean. In other words, you have evidence that your sample does not come from a population with a mean of 120; your sample likely comes from a population with a mean that is statistically greater than 120 (perhaps closer to 131.85).

You don't always need to compare a sample mean to a population mean. Perhaps you want to compare your sample mean to some hypothetical value. Some other uses of a one-sample *t*-test include:

1. testing a sample against a pre-defined value,
2. testing a sample against an expected value,
3. testing a sample against common sense or expectations,
4. testing the results of a replicated experiment against the results of an original study

### 3. Estimating the variance and standard deviation of the population

This section is mostly information so that you see what R is doing with your data. If you do not know the variance or standard deviation of the population of individuals, you can estimate them from what you do know-the scores of the people in your sample. In the logic of hypothesis testing, the sample of experimental units (e.g., people) you study is assumed to be a random sample taken from a particular population with a particular mean and variance. The variance of this sample ought to reflect the variance of that population. If the scores in the population have a lot of variation, then the scores in a sample randomly selected from that population should also have a lot of variation. Yet, the variance of your sample will generally be slightly smaller than the population. Thus the variance of your sample is referred to as a *biased estimate* of the population variance.

To obtain an *unbiased estimate* of variance or standard deviation for your population, you can use the built-in `var()` and `sd()` functions which return the *unbiased estimates*. Many other statistics software will always provide unbiased estimates.

```
# Calculate the standard deviation
BP.dev <- BLOOD$BP - mean(BLOOD$BP) # Create deviation scores for each value from the sample mean.
BP.dev # notice some deviation scores are positive or negative based on whether the scores are above or below the sample mean
```

```
## [1] -3.85 -4.85 -7.85 -5.85 12.15 10.15 1.15 8.15 0.15 -0.85
## [11] -1.85 0.15 17.15 -9.85 7.15 -12.85 4.15 -2.85 -5.85 -3.85
```

```
# Square those deviation scores
BP.squared.dev <- BP.dev^2
BP.squared.dev # Now these are all squared values
```

```
## [1] 14.8225 23.5225 61.6225 34.2225 147.6225 103.0225 1.3225
## [8] 66.4225 0.0225 0.7225 3.4225 0.0225 294.1225 97.0225
## [15] 51.1225 165.1225 17.2225 8.1225 34.2225 14.8225
```



```
# sum the squared deviations
BP.SS <- sum(BP.squared.dev)
BP.SS
```

```
## [1] 1138.55
```

```
# biased variance; SS/n
BP.var.biased <- BP.SS/length(BLOOD$BP)
BP.var.biased
```

```
## [1] 56.9275
```

```
# unbiased variance; SS/(n-1)
BP.var.unbiased <- BP.SS/(length(BLOOD$BP) - 1)
BP.var.unbiased
```

```
## [1] 59.92368
```

```
# biased sd
BP.sd.biased <- sqrt(BP.var.biased)
BP.sd.biased
```

```
## [1] 7.545031
```

```
# unbiased sd
BP.sd.unbiased <- sqrt(BP.var.unbiased)
BP.sd.unbiased
```

```
## [1] 7.741039
```

```
# unbiased estimates using the functions
var(BLOOD$BP)
```

```
## [1] 59.92368
```

```
sd(BLOOD$BP)
```

```
## [1] 7.741039
```

Notice how the biased estimate is slightly smaller than the unbiased estimate. This indicates that the biased estimate is stating that the variability in the population of scores is smaller than it actually is. Also, the size of the bias will decrease as sample sizes increase because larger samples will be less biased in general so the correction factor using df (degrees of freedom) is used.

Once you have figured the estimated population variance, you can calculate the standard deviation of the comparison distribution. As always, when you have a sample of more than one, the comparison distribution is a distribution of means (sampling distribution). And the variance of a distribution of means is the variance of the population of individuals divided by the sample size. You have just estimated the variance of the population. Thus, you can estimate the variance of the distribution of means by dividing the estimated population variance by the sample size. The standard deviation of the distribution of means is the square root of its variance.

## 4. Degrees of Freedom

The concept of degrees of freedom is central to the principle of estimating statistics of populations from samples of them. “Degrees of freedom” is commonly abbreviated to df. In short, think of df as a mathematical restriction that we need to put in place when we calculate an estimate one statistic from an estimate of another. The idea is that, when figuring the variance, you first have to know the mean. If you know the mean and all but one of the scores in the sample, you can figure out the one you do not know with a little arithmetic. Thus, once you know the mean, one of the scores in the sample is not free to have any possible value. So in this kind of situation the df is the number of scores minus 1. In terms of a formula,

$$df = N - 1$$

In our example we have 19 degrees of freedom:

```
df = 20 - 1 #Because our n is 20
df
```

```
## [1] 19
```

```
# or use the length() function
df = length(BLOOD$BP) - 1
df
```

```
## [1] 19
```

## 2. t-test for paired sample means

2.1. The situation you just learned about (the *t*-test for a single sample) is for when you know a population’s mean but not its variance and you also have a single sample of scores. It turns out that in most research you do not even know the population’s mean; plus, in most research situations, you usually have not one set but two sets of scores. These two things, not knowing the population mean and

having two sets of scores, almost always go together. This situation is very common. This kind of research situation is called a repeated-measures design (also known as a within-subjects design). A common example is when you measure the same people before and after some social or psychological intervention. This specific kind of repeated-measures situation is called a *before-after design*.

2.2. The hypothesis-testing procedure for the situation in which each person is measured twice (that is, the situation in which we have a repeated-measures design) is called a *t* test for dependent means. The *t*-test for dependent means is also called a paired-sample *t* test, a *t* test for correlated means, a *t*-test for matched samples, and a *t*-test for matched pairs. Each of these names come from the same idea that in this kind of *t* test you are comparing two sets of scores that are related to each other in a direct way, such as each person being tested before and after some procedure.

2.3. You do a *t*-test for dependent means in exactly the same way as a one-sample *t*-test, except that: (a) you calculate and compare the difference of the paired values (e.g., difference scores), and (b) you assume that the population mean (of the difference scores) is 0 under  $H_0$ .

2.4. With a repeated-measures design, your sample includes two scores for each experimental unit (e.g., person) instead of just one. The way you handle this is to make the two scores per person into one score per person! You do this magic by creating difference scores: For each person, you subtract one score from the other. If the difference is before versus after, difference scores are also called change scores.

To show this we are going to read in pre-test and post-test scores from a group of students who ranked their happiness scores (0 least happy to 30 most happy) before and finals week and assign them to a data-frame object named HAPPY.

```
#setwd("C:/Users/gcook/Desktop/Psyc109")
HAPPY <- read.csv("FinalsHappiness.csv")
str(HAPPY)
```

```
## 'data.frame':    20 obs. of  3 variables:
## $ Student: int  1 2 3 4 5 6 7 8 9 10 ...
## $ Pre    : int  18 21 16 22 19 24 17 21 23 18 ...
## $ Post   : int  22 25 17 24 16 29 20 23 19 20 ...
```

```
#HAPPY
```

As you can see we have Student ids, happiness pre-test scores, and happiness post-test scores. Now we need to create the difference scores so we can analyze the data to examine if the difference is close to or far away from 0.

2.5. To conduct a paired-sample *t*-test we will use the built-in `t.test()` function from the `stats` library as well as write out the equation. The two variables are included as separate arguments; the order does not matter. Also, the mean difference because under  $H_0 = 0$ , so there is no need to include the argument to set  $\mu$ . Just make sure to specify that the data are paired by setting the `paired` argument to `TRUE` (e.g., `paired = TRUE`). You did not have to do this for the one-sample *t*-test and by default the test is two-tailed.

The general formula is:

```
t.test(X1, X2, paired = TRUE)
```

```
t.test(HAPPY$Pre, HAPPY$Post, paired = TRUE)
```

```
##  
## Paired t-test  
##  
## data: HAPPY$Pre and HAPPY$Post  
## t = -3.2313, df = 19, p-value = 0.004395  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -3.3778749 -0.7221251  
## sample estimates:  
## mean of the differences  
## -2.05
```

2.6. The  $t$ -value,  $df$ ,  $p$ -value, and mean difference scores are included in the output. If you use the  $p$ -value to help determine the evidence for the difference value being the same as or different from 0, the paired-samples  $t$ -test reveals the true difference in means is not equal to 0. Rather, there is a difference in happiness before and after final exams. Finally, because the sample difference score is estimated from the sample, it is not completely accurate representation of the population difference. A 95% confidence interval provides a range for the actual difference score that you might expect the true difference to be. Notice the the interval does not include a difference of 0, so you can be 95% confident that the real difference score is not 0. Instead, there must be some systematic reason for the difference in happiness scores that is not simply due to random chance. Perhaps students are stressed out before final exams and they relax after them.

To report the outcome of the test, the general form is:

$t(df) = t\text{-value}, p < \text{or} > \alpha$

By substitution and making sure to italicize  $t$  and  $p$ , we can say:

Students' happiness ratings were statistically higher following final exams than they were before those exams,  $t(19) = 3.38, p < .05$ .

## 3. Assumptions for $t$ -tests

3.1. As we have seen, when you are using an estimated population variance, the comparison distribution is a  $t$  distribution rather than a  $z$  distribution (normal).

However, in order to perform a  $t$ -test, there are few assumptions that should be met.

3.2. Assumptions for the one-sample  $t$ -test

- a. The data must be interval/ratio
- b. The data should follow the normal probability distribution
- c. The sample represents a random sample from its population

3.3. Assumptions for the paired-samples  $t$ -test

- a. The data must be interval/ratio
- b. The difference scores for the matched pairs should follow a normal probability distribution.
- c. The sample of pairs represents a random sample from its population

When you have a repeated-measures design with two measurement variables and do not have interval/ratio data or have violated normality, you can use an alternative test called the Wilcoxon Signed-Ranks Test using the `wilcox.test()` function from the built-in stats library. However, if the X1 and X2 values are the same (ties), they are dropped from the analyses.

Similar to the paired-samples *t*-test, the function takes the form:

```
wilcox.test(X1, X2, paired = TRUE)
```

```
wilcox.test(HAPPY$Pre, HAPPY$Post, paired = TRUE)
```

```
## Warning in wilcox.test.default(HAPPY$Pre, HAPPY$Post, paired = TRUE):  
## cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(HAPPY$Pre, HAPPY$Post, paired = TRUE):  
## cannot compute exact p-value with zeroes
```

```
##  
## Wilcoxon signed rank test with continuity correction  
##  
## data: HAPPY$Pre and HAPPY$Post  
## V = 29, p-value = 0.008105  
## alternative hypothesis: true location shift is not equal to 0
```

The Wilcoxon signed-rank test does not calculate the magnitude of the difference between X1 and X2 scores. Rather, it examines the ranking of the variables; how many times  $X1 > X2$  and  $X2 > X1$  independent of the size of the difference. When X1 and X2 are the same (ties), they are dropped from the analysis. The test compares the sum of the ranking order of the difference (e.g., the number of times  $X1 > X2$  vs.  $X2 > X1$ ) and compares that to a distribution that assumes the ranks are the same as expected under  $H_0$ . The output provides a Wilcoxon test value and a *p*-value associated with the likelihood of such differences occurring under  $H_0$ .

If you examine the data, you will see that the Diff variable has only 1 tie (0), 4, instances of  $Pre > Post$  scores and 15 instances of  $Post > Pre$  scores. If using the *p*-value to make decisions about  $Pre = Post$ , you will see that  $p < .05$ , so that can be used as evidence of a difference in happiness before and after final exams using the Wilcoxon signed-Ranks Test for paired samples.

By substitution and making sure to italicize *V* and *p*, we have:

Students' happiness ratings were statistically higher following final exams than they were before those exams,  $V = 29$ ,  $p < .05$ .

## 4. Effect size: Calculating the strength of an effect

The conventions for effect size may already be familiar to you in the form of correlation values. Correlation providing a measure of the strength of relationship between variables. Typically, a larger effect size provides better evidence for the difference between comparison groups.

#### 4.1. Correlation-based effect sizes

One measure of effect size is the Pearson's  $r$  correlation coefficient,  $r$ , which can range from -1.0 to +1.0. Related,  $r$ -squared represented an estimate of the amount of the variance within an experiment that is "accounted for" by the statistical model (e.g., linear regression). This can range from 0 to 1.0. Effect sizes vary from discipline to discipline, so rather than determine was the small or large, just know that larger is generally a better.

Using  $r$ -squared as a measure of effect size for paired samples is a little more challenging. Although the relationship between  $r$  and  $t$  will not be described here, you can estimate  $r$ -squared using the information returned from the `t.test()` function:

$r\text{-squared} = t^2 / (t^2 + df)$

```
HAPPY.t <- t.test(HAPPY$Pre, HAPPY$Post, paired = TRUE)

# Now put those values into a formula
HAPPY.r2 <- HAPPY.t$statistic^2 / ((HAPPY.t$statistic^2) + HAPPY.t$parameter)
HAPPY.r2
```

```
##           t
## 0.3546414
```

```
# if you wanted r rather than r squared, just take the square root.
HAPPY.r <- sqrt(HAPPY.r2)
```

You can see that about 35% of the variability in the data can be accounted for by the statistical model.

#### 4.2. Cohen's $d$ effect size

Another measure of effect size is Cohen's  $d$ . In its general form, this is a measure of the difference between comparison groups as a function of standard deviations. The greater the number of standard deviations that separate the two means, the larger the effect size. Because the measurement is in terms of standard deviations, large difference scores that are accompanied by large standard deviations are not impressive in terms of effect size, whereas large difference scores that are accompanied by small standard deviations can be impressive. Notice that Cohen's  $d$  does not account for sampling distributions based on sample size as does the  $t$ -test, so Cohen's  $d$  is influenced less by fluctuations in sample size. In sum, Cohen's  $d$  provides a measure of the difference in terms of standard deviations. For example, a  $d = 1.0$  simply refers to a difference between the group means is separated by 1 standard deviation, whereas a  $d = .50$  refers to a difference between the group means which is separated by half of a standard deviation. Because it is measured in terms of the number of standard deviations that separate the comparison means, it is not limited to 1.0 like  $r$ -squared is.

A great dynamic illustration of Cohen's  $d$  that I recommend checking out can be found at:

<http://rpsychologist.com/d3/cohend/> (<http://rpsychologist.com/d3/cohend/>) (just use the visual, don't worry about Cohen's U value)

Although classifying values that are continuous places limits on the the size of effect size, Cohen (1988) suggested that effect sizes could be classified as Small,  $d = .2$ , Medium,  $d = .5$ , and Large,  $d = .8$ .

#### 4.3. Calculating measures of effect size

4.3.1. z-test: When the population standard deviation is known we can use Cohen's  $d$  as an estimate of effect size:

$$d = | \bar{x} - \mu / \sigma |$$

4.3.2. One-sample  $t$ -test: When the population variance is not known, Cohen's  $d$  is estimated using the unbiased sample standard deviation:

$$d = | \bar{x} - \mu / \sigma |$$

Using the blood-pressure data, we can calculate Cohen's  $d$  based on the difference between the means as a function of standard deviations.

```
BLOOD.d <- abs((mean(BLOOD$BP) - 120) / sd(BLOOD$BP))  
BLOOD.d
```

```
## [1] 1.530802
```

4.3.3. Paired-Samples  $t$ -test: The estimated effect size for a study using a  $t$ -test for dependent means is the mean of the difference scores divided by the estimated standard deviation of the population of difference scores. Thus, Cohen's  $d$  for a paired-samples  $t$ -test is simply the size of the mean difference as a function of the standard deviation of the difference. If we go back to the pre-test and post-test scores for happiness before and after finals week we can determine the effect size below:

```
HAPPY.d <- abs((mean(HAPPY$Pre) - mean(HAPPY$Post)) / sd(HAPPY$Diff))  
HAPPY.d
```

```
## [1] NA
```

```
# Or if you have a difference score,  
HAPPY.d <- abs((mean(HAPPY$Diff)) / sd(HAPPY$Diff))
```

```
## Warning in mean.default(HAPPY$Diff): argument is not numeric or logical:  
## returning NA
```

```
HAPPY.d
```

```
## [1] NA
```

From that example we have an effect size of about .72 which reveals a medium Cohen's  $d$  effect size which is on the higher end of medium effects. Because classifying the value can lead to confusion, just state the numeric value and don't label the effect as small, medium, or large; instead, let individuals decide for themselves what is small or large.

## 5. Do it yourself!

One question asked on the survey was the percentage of a bill left as a tip for a server at a restaurant. To help you, the following code will provide you the class data needed to answer the question. You just need to run each line of code so that you will have your data.

```
SURVEY <- read.csv("SurveyQuestions.csv")
SURVEY$Tip <- SURVEY$Q19 # On average, what percentage tip do you leave after a meal at a
restaurant?
SURVEY$SocialMediaPerDay <- SURVEY$Q20 # On average, how many hours a day do you spend on
social media?
ExercisePerWeek <- SURVEY$Q25 # On average, how many hours a week do you exercise?
SURVEY$ExercisePerDay <- ExercisePerWeek/7 # divide by 7 for days
SURVEY$SleepPerNight <- SURVEY$Q26 #On average, how many hours of sleep do you get per ni
ght?
SURVEY <- subset(SURVEY, select = c(ID, Gender, Tip, ExercisePerDay, SocialMediaPerDay))
str(SURVEY)
```

```
## 'data.frame':    75 obs. of  5 variables:
## $ ID             : int  4 5 6 7 8 9 10 11 12 13 ...
## $ Gender          : int  0 1 1 1 1 1 0 0 0 1 ...
## $ Tip             : int  15 10 10 10 15 15 20 15 15 20 ...
## $ ExercisePerDay  : num  0 1.714 0.571 0.857 2.857 ...
## $ SocialMediaPerDay: num  3 10 2 3 2 1 2 0.5 3 1 ...
```

5.1.1. **QUESTION:** For a population of men and women, a very large data set reports that the average tipping percent is 16.08026%. Use the appropriate test function to determine whether the people responding to the class survey they were generous or cheapskates in their tipping behavior relative to those in the population.

**ANSWER:**

```
tipping.t <- t.test(SURVEY$Tip, mu = 16.08026)
tipping.t
```



```
##  
## One Sample t-test  
##  
## data: SURVEY$Tip  
## t = -0.075723, df = 74, p-value = 0.9398  
## alternative hypothesis: true mean is not equal to 16.08026  
## 95 percent confidence interval:  
## 15.34479 16.76187  
## sample estimates:  
## mean of x  
## 16.05333
```

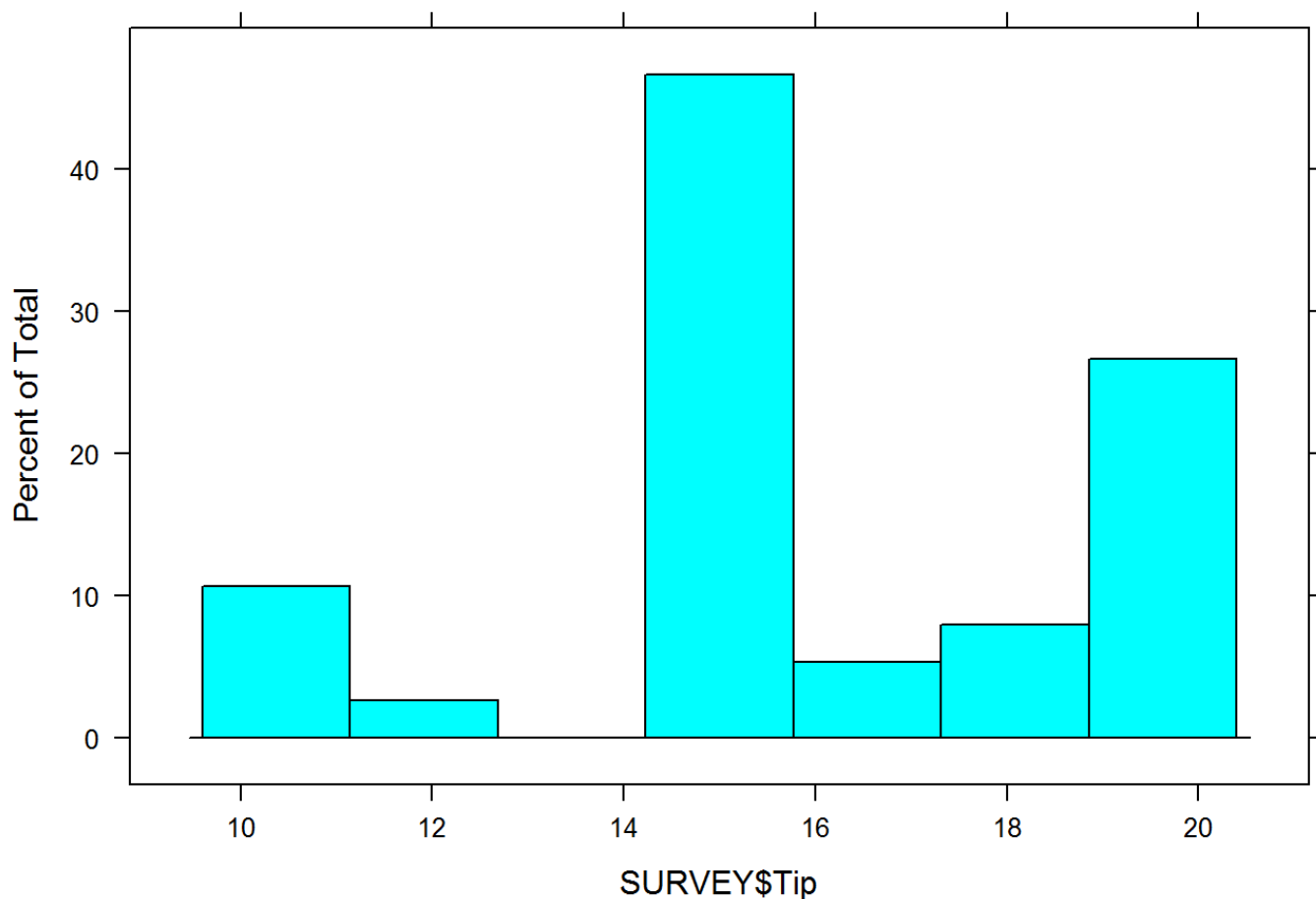
5.1.2 **QUESTION:** Test whether a normal distribution is a reasonable assumption for those data. Test with  $\alpha = .05$ . You can look at a histogram too if you want.

*ANSWER:*

```
shapiro.test(SURVEY$Tip)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: SURVEY$Tip  
## W = 0.8434, p-value = 0.0000001997
```

```
histogram(~SURVEY$Tip)
```



5.1.3. **QUESTION:** What do you conclude based on your test output? Be clear in your answer.

**ANSWER:** The distribution deviates from that of a normal distribution,  $S-W = .84$ ,  $p < .05$ ; the assumption is not met. Based on this result, the  $t$ -test result may not be stable. Based on the distribution, there may be some outlier to remove, but removing them will likely not change the shape of the distribution because most people tip about 15% and there is not much variability in the data. The  $t$ -test is likely not an appropriate test.

5.2.1. Use the SURVEY data frame and the appropriate test to determine if people report spending equivalent amounts of time per day on social media or exercising. Test with  $\alpha = .05$ .

**ANSWER:**

```
t.test(SURVEY$ExercisePerDay, SURVEY$SocialMediaPerDay, paired = TRUE)
```

```
##
## Paired t-test
##
## data: SURVEY$ExercisePerDay and SURVEY$SocialMediaPerDay
## t = -5.3853, df = 74, p-value = 0.0000008201
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -1.707928 -0.785405
## sample estimates:
## mean of the differences
## -1.246667
```

```
mean(SURVEY$ExercisePerDay); mean(SURVEY$SocialMediaPerDay)
```

```
## [1] 1.113333
```

```
## [1] 2.36
```

5.2.2. **QUESTION:** What do you conclude based on your test output? Be clear in your answer.

*ANSWER:* On average, people spend significantly fewer hours exercising per day ( $M = 2.36$ ) than they do on social media ( $M = 1.11$ ),  $t(74) = 5.39$ ,  $p < .05$ .

5.3.0. A health clinic offers a program to help their clients lose weight. The clinic asks a consumer agency to investigate the effectiveness of the program. The agency takes a sample of 15 people who participate in the program and weighs (in pounds) each individual in the sample before the program begins and again 3 months later. You will need to help determine whether the program is effective.

5.3.1. **QUESTION:** Read in the Weight.csv data file as a data frame object named WEIGHT and look at the data frame's structure. You should see Before and After weights in pounds.

*ANSWER:*

```
WEIGHT <- read.csv("Weight.csv")
str(WEIGHT)
```

```
## 'data.frame': 15 obs. of 3 variables:
## $ Person: int 1 2 3 4 5 6 7 8 9 10 ...
## $ Before: int 210 205 193 182 259 239 164 197 222 211 ...
## $ After : int 197 195 191 174 236 226 157 196 201 196 ...
```

5.3.2. **QUESTION:** Use the appropriate function to analyze the data to answer the questions.

```
t.test(WEIGHT$Before, WEIGHT$After, paired = TRUE)
```

```
##
## Paired t-test
##
## data:  WEIGHT$Before and WEIGHT$After
## t = 6.6897, df = 14, p-value = 0.00001028
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##    7.427994 14.438673
## sample estimates:
## mean of the differences
##                10.93333
```

```
mean(WEIGHT$Before); mean(WEIGHT$After)
```

```
## [1] 207.9333
```

```
## [1] 197
```

5.3.3. Based on the output, answer the following questions.

**QUESTION:** How much weight was lost/gained on average? Be clear in your answer.

**ANSWER:** The program resulted in 10.93 pounds of weight loss.

5.3.4. **QUESTION:** Did the program influence weights more than expected due to chance.

**ANSWER:** The program appeared to work; participants weighed significantly less after the program,  $t(14) = 6.69$ ,  $p < .05$ .

5.3.5. **QUESTION:** Is the difference significantly different from 0 when  $\alpha = .05$ ? Explain how you arrived at your decision.

**ANSWER:**

5.3.6. **QUESTION:** What are the degrees of freedom for the test?

**ANSWER:** 14

5.3.7. **QUESTION:** Based on the test output, determine the  $r$  effect size.

**ANSWER:**

```
# assign the results of the t-test to an object that you can refer to in order to extract
the information needed for calculating the r-squared effect size.
WEIGHT.t <- t.test(WEIGHT$Before, WEIGHT$After, paired = TRUE)

# Now put those values into a formula
WEIGHT.r <- sqrt(WEIGHT.t$statistic^2 / ((WEIGHT.t$statistic^2) + WEIGHT.t$parameter))
WEIGHT.r
```

```
##          t
## 0.8727603
```

# DONE!

E-mail your knit HTML file that you completed with your partner. Make sure to include your names.