ST 411/511 Lab 2
Paired and two-sample t-tests

Objectives for this Lab

- Do a paired t-test with R's `t.test()` function.
- Calculate the t-statistic and p-value by hand, with R's assistance.
- Calculate a confidence interval by hand, using R as a calculator.
- Do a two-sample t-test using `t.test()`.
- Read in a data set from a comma-delimited text file.

1. As in Lab 1, start up RStudio. Download and open Lab2.R in RStudio.

2. Load the Sleuth3 and ggplot2 R packages.

```
> library(Sleuth3)
> library(ggplot2)
```

3. Produce a histogram of the differences in the twin study (case0202).

```
> qplot(Unaffected-Affected, data=case0202, geom="histogram")
```

4. Next, we'll do a paired t-test to test if the population of differences in left hippocampus volumes have non-zero mean. A paired t-test is a one-sample t-test on the differences between the responses within pairs.

   (a) Create the differences, stored in a vector called `diffs`. Either of the following commands will accomplish this.

   ```
   > diffs <- case0202$Unaffected-case0202$Affected
   > diffs <- with(case0202, Unaffected-Affected)
   ```

   You can view the differences in the R Console by typing the name of the vector.

   ```
   > diffs
   ```

   If you create a histogram of `diffs`, you'll get the same graph as in 3., except the title and horizontal axis label will be different.

   ```
   > qplot(diffs, geom="histogram")
   ```

   We don't need to tell R the data frame name because `diffs` is a variable, not a column in a data frame.

   (b) Do the t-test.

   ```
   > t.test(diffs)
   ```

   That's it. No need for a `data=` argument. If we omit the `alternative=` argument, we get the *default*, a two-sided t-test.

   (c) Examine the output from `t.test()`. Find the t-statistic, degrees of freedom, p-value, and 95% confidence interval.

(d) There are usually multiple ways to accomplish the same thing in R. Another way to do the paired t-test is to give both groups to `t.test()` but specify you want a paired test.

```
> with(case0202, t.test(Unaffected, Affected, paired=TRUE))
```

With this syntax, R ignores the `data=` argument, so we have to use `with()`.

Except for some slight differences in formatting, the output should be exactly the same as for the previous `t.test()` command. The `paired=TRUE` syntax in the above command is an optional argument. The "default value" of this argument is `FALSE`, so if you give `t.test()` two groups and don't specify `paired=TRUE`, you will get a two-sample t-test.

```
> with(case0202, t.test(Unaffected, Affected))
```

Caution: You get very different results if you do a two-sample t-test. The twin data are *paired* and therefore the two-sample t-test is inappropriate. In Chapter 3, we will focus on the assumptions of t-tests. You may recall that one of the assumptions of the two-sample t-test is that the two samples are *independently* drawn from two populations. Why is this assumption invalid for the twin study?

5. As we encounter more complicated data analyses, it will be useful to be able to calculate quantities "by hand" because R may not calculate the items we need. Here we will calculate the twin study t-statistic by hand, though since R does this automatically, we will only do this calculation once.

(a) Obtain the elements needed to calculate the t-statistic by hand. Refer to the equations on page 35 of the *Sleuth* for $\text{SE}(\overline{Y})$ and the t-ratio. In the twin study, Y refers to the difference in response between the two twins, and $\overline{Y}$ refers to the sample mean of these differences. This sample mean is the "estimate" of the population mean in the t-ratio. The "parameter" is the null hypothesized value for the population mean, and is almost always 0, as it is here (why?). The notation s denotes the sample standard deviation, and n is always the sample size. Below is R code to calculate each of these quantities.

```
> (Ybar <- mean(diffs)) #Calculate the sample mean of the differences
> (s <- sd(diffs))     #Calculate the sample standard deviation
> (n <- length(diffs))         #Find the sample size
```

Embedding each assignment statement in parenthesis tells R to print out the result. You should get $\overline{Y} = 0.1986667$, $s = 0.2382935$, and $n = 15$.

(b) Calculate the standard error of the sample mean $\text{SE}(\overline{Y})$ according to the formula at the top of page 35 of the *Sleuth*.

```
> (se_Ybar <- s/sqrt(n))    #Calculate the SE of the sample mean
```

You should get $\text{SE}(\overline{Y}) = 0.06152713$.

(c) Now calculate the t-ratio according to the formula at the bottom of page 35. Since the null hypothesized value of the population mean is 0, this is simply

$$t\text{-ratio} = \frac{\text{Estimate}}{\text{SE(Estimate)}}$$

which is $\overline{Y}/\text{SE}(\overline{Y})$:

```
> Ybar/se_Ybar
```

Your calculated t-ratio should be the same as in the output of `t.test()` obtained in item 4(b) above.

(d) Calculate the p-value of the two-tailed test. Refer to Display 2.5 on page 37 and the discussion at the bottom of page 36 of the *Sleuth*. The two-sided p-value is the area under the t_{14} distribution to the right of 3.228928 and to the left of -3.228928. The subscript 14 is the appropriate degrees of freedom. Recall for a one-sample t-test, the degrees of freedom are $n - 1$.

R's `pt()` command calculates **p**robabilities from **t** distributions. For example,

```
> pt(3.228928, 14)
```

calculates the area to the left of 3.228928 under the t distribution with 14 degrees of freedom.

We want the area to the *right* of 3.228928, and since the total area is 1, the area to the left is

```
> 1-pt(3.228928, 14)
```

And actually, we need twice this number to account for the area to the left of -3.228928, so calculate the p-value as

```
> 2*(1-pt(3.228928, 14))
```

Compare this to the output from the t-test in item 4(b). When calculating p-values, it's always useful to sketch a picture.

Alternatively, tell `pt()` to calculate the upper (right) tail probability.

```
> 2*pt(3.228928, 14, lower.tail=FALSE)
```

(e) We can also calculate the 95% confidence interval "by hand." Refer to the formula in Display 2.6 on page 38 of the *Sleuth*. We have already calculated $\overline{Y}$ and $\text{SE}(\overline{Y})$, so all we need is t-quantile $t_{n-1}(0.975)$. R will calculate this quantile for you:

```
> qt(.975, 14)
```

You should get 2.144787. For a 95% confidence interval, this number should always be near 2, provided the degree of freedom are not too small.

Calculate the confidence interval:

```
> Ybar - qt(.975, 14)*se_Ybar
> Ybar + qt(.975, 14)*se_Ybar
```

Compare to the output from `t.test` in item 4(b).

6. Do a two-sample t-test to analyze the finch data of Case Study 2.1.1.

   (a) Start by plotting the data.

   ```
   > qplot(factor(Year), Depth, data=case0201, geom="boxplot")
   ```

   R thinks that the data in `Year` are numbers, but we are thinking of Year as a "factor" or grouping variable. If you omit `factor()`, you'll get a warning and a strange-looking plot.

   (b) Think about the data and what we want to test. Are the samples paired, or are they independent?

   As usual, the null hypothesis is "no difference." Is the alternative one-sided or two-sided? Which side?

   (c) Though we plan to do a one-sided t-test, first do a two-sided test. The output will give a two-sided confidence interval, which is what we usually want.

   ```
   > t.test(Depth~Year, data=case0201, var.equal=TRUE)
   ```

   The optional argument `var.equal` has default value `FALSE` which means `t.test` usually does not assume the two populations have equal variance. We will explore this issue in Chapter 3.

   Find the 95% confidence interval in the R output and compare to Display 2.9 on page 43 and to the Statistical Conclusion on pages 29 and 30 of the *Sleuth*.

   (d) To do a one-sided test, we need to determine how R orders the groups. That's apparent from the output of the two-sided test above. The last line of the output gives the group means in R's order. This is usually a sensible ordering–numeric or alphabetical. Here, the 1976 group is first.

   If we let μ_1 denote the population mean depth for 1976 and μ_2 the population mean depth for 1978, then the null hypothesis is $H_0 : \mu_1 - \mu_2 = 0$. What's the alternative hypothesis? This comes from the research question.

   (e) Do the one-sided two-sample t-test.

   ```
   > t.test(Depth~Year, data=case0201, var.equal=TRUE, alternative="less")
   ```

   In the R output, identify the t-statistic and the p-value. How would you report the conclusion of this test?

7. Homework 2 will ask you to work with data from a study comparing direct-touch (DT) to mousing (M) when navigating online maps. These data are contained in the file NavDat.csv, available in Files>Homework on Canvas. There are 36 rows of data, one for each of 36 subjects. The time required to complete a navigation task using each method is recorded in the columns labeled DT and M.

   (a) Download NavDat.csv onto your computer, and note the directory where you put it.

(b) Read NavDat.csv into an R data frame called "HW2Dat." The command below assumes you have the file in your current working directory.

```
> HW2Dat <- read.csv("NavDat.csv")
```

If the file is in a different directory, you will need to include the pathname. For example, if the file was in D:\Stats, the command would be

```
> HW2Dat <- read.csv("D:/Stats/NavDat.csv")
```

Always use a forward slash, even on a Windows system.

(c) Use the `head()` command to view the first few rows of the data frame.

```
> head(HW2Dat)
```