

Paired-samples t-test Tutorial

A paired-samples t-test is used when you collect numerical data from the same individual twice and you are using that data to determine if change has occurred. For example, you give a test to your students, you do the intervention or treatment, and then you test them again to determine if the treatment was effective. The key thing is you know each individual student's pre-treatment and post-treatment score.

Example. We give the students an algebra quiz (the “before” test scores, see below). The next class we give the students a quick review focusing on the most common mistakes they made (the “treatment”). Then, we re-test them with a similar algebra quiz (the “after” test scores). We want to determine if this treatment is effective, i.e., does it raise quiz scores?

Here are the students' scores (see below) as they appear in a spreadsheet. Note “abs” means the student was absent and “late” means the student arrived too late to take the quiz. The yellow is just to highlight the “abs” and “late” entries.

	A	B	C	D
1	StudentID	before	after	
2	1	abs	80.9	
3	2	77.4	abs	
4	3	late	abs	
5	4	80.5	90	
6	5	abs	abs	
7	6	79.1	78.5	
8	7	76.7	90.8	
9	8	76.6	84.4	
10	9	80.1	93.3	
11	10	78	84.8	
12	11	78.8	87.5	
13	12	81.4	85.6	
14	13	82.4	91.2	
15	14	71.2	84.9	
16	15	80.3	84.6	
17	16	80.8	86.8	
18	17	83.4	83.9	
19	18	79.7	82	
20	19	76.7	91.8	
21	20	80.3	84.8	
22	21	77.2	89.5	
23	22	76.7	94.4	
24	23	79.2	91	
25	24	82.1	88.1	
26	25	83.9	83	
27	26	84.6	82.9	
28	27	79.4	85.5	
29	28	80.4	abs	
30	29	84.6	83.9	
31	30	78.1	86.3	
32	original data			

Statistical Goal. We want to know if this data provides **statistically significant evidence**¹ that the treatment, on

¹ **Understanding statistical significance and p-values: an example.** Suppose you want to know who will win the next election.

Case 1. You ask three people how they will vote and two say “Democrat”. This would not be considered statistically significant evidence (exact binomial test, p-value $p = 0.5 > 0.05$) that the Democrats will win because the p-value > 0.05 . Explanation: even though 2 out of 3 favored the Democrats, the sample size was too small. Hence, the p-value ended up being too big.

Case 2. You ask 100 people and 51 of them say they'll vote “Democrat”. This would not be considered statistically significant evidence (exact binomial test, p-value $p = 0.4602 > 0.05$) that the Democrats will win because the p-value > 0.05 . Explanation: even though the sample size is not small, only 51% favored the Democrats, 51% is too close to 50%. Hence, the p-value ended up being too big.

Case 3. You ask 100 people and 59 of them say they'll vote “Democrat”. This would be considered statistically significant evidence (exact binomial test, p-value $p = 0.0443 < 0.05$) that the Democrats will win because the p-value < 0.05 . Explanation: the sample size is large enough and 59% is not too close to 50%. Hence, the p-value ended up being small enough.

Note. Just because you have statistically significant evidence that some claim is true (e.g., that the democrats will win), it doesn't mean the claim is true. For example, your data might not be representative of the population from which it is drawn; or your respondents might be lying; or your data might be corrupted due to faulty equipment, or your data isn't measuring what you think it is, etc.

The **p-value** is a mathematical measure of how “gullible” we need to be in order to accept a given claim as true based upon the given data. So, the smaller the p-value, the less gullible we'd need to be, and less gullible is good. The cut-off for statistical significance is a p-value of .05. **If the p-value $< .05$ we can say we have statistical significance.** On a practical level, the p-value is the first hurdle your study needs to clear. It tells your readers that if they are to dispute your results, they shouldn't say things like your sample size was too small, but rather they should attack your selection process as being biased or not-random, or that your survey was poorly worded, or that your respondents aren't answering honestly, and so on.

average, will lead to a student getting a higher quiz score when re-tested. The key thing is that we are going to make an **inference** about students in general based upon how our students did.

Analyzing the Data in R

The work flow for statistical analysis is like this. You keep your data in a data file, which for us will be a spreadsheet, e.g., Excel² or OpenOffice³, which is free. When you want to analyze your data, you import it into the free, open-source statistical computing environment R⁴ and use an R script, see below, to analyze it. R will not modify the data file, unless you explicitly tell it to.

We will use the following R script to analyze the data from the example and to perform a paired-samples t-test.

```
# tTestPairedSoTL.R
# paired t test for SoTL

# Import Data From Spreadsheet (Step 1)
# -----
# Data: 2 columns of numeric data, blank cells are ok, but no cells with words.
df = read.table(file = "clipboard", sep = "\t", header = TRUE);
df;          # display student data in console.

# Clean data in R (remove the rows with NA's)(Step 2)
# -----
dfc = na.omit(df) # dfc = df with rows with blank cells removed
dfc

# Find the mean and standard deviation of the quiz scores (Step 3)
# -----
mean(dfc$before); # the mean of the before scores
sd(dfc$before);  # the standard deviation of the before scores

mean(dfc$after); # the mean of the after scores
sd(dfc$after);   # the standard deviation of the after scores

# Create histogram of data (Step 4)
# -----
hist(dfc$after - dfc$before,
main = "Histogram of quiz score differences\nPost-treatment minus pre-treatment",
xlab = "Percent",
col = "green" );

# Do the paired t-test (Step 5)
# -----
t.test(dfc$after - dfc$before, alternative = "greater", mu = 0);

# End of Script
#-----
```

See the next pages for how to use the above R script.

²Microsoft Office

³<https://www.openoffice.org/> An excellent, free, open-source version of Microsoft Office.

⁴<https://www.r-project.org/> We will explore and analyze our data using the very powerful, open source statistical software R. R is probably the most widely used statistics software by professional statisticians. You can **download R for free** and find **R tutorials** at: <https://www.r-project.org/>. R is very easy to use. You can easily find free resources online to help you learn R. Also, see the Appendix of these notes for a short introduction to using R.

Step 0. Cleaning the data before exporting it to R

It is important to clean the data before importing it into R. This means deleting the non-numerical information like “abs” or “late.” There are ways to use R to automate data cleaning, however, unless you have a lot of data, it is easier to just do it by hand in Excel or whatever spreadsheet you are using.

before cleaning				after cleaning			
	A	B	C	D	E	F	G
1	StudentID	before	after		StudentID	before	after
2	1	abs	80.9		1		80.9
3	2	77.4	abs		2	77.4	
4	3	late	abs		3		
5	4	80.5	90		4	80.5	90
6	5	abs	abs		5		
7	6	79.1	78.5		6	79.1	78.5
8	7	76.7	90.8		7	76.7	90.8
9	8	76.6	84.4		8	76.6	84.4
10	9	80.1	93.3		9	80.1	93.3
11	10	78	84.8		10	78	84.8
12	11	78.8	87.5		11	78.8	87.5
13	12	81.4	85.6		12	81.4	85.6
14	13	82.4	91.2		13	82.4	91.2
15	14	71.2	84.9		14	71.2	84.9
16	15	80.3	84.6		15	80.3	84.6
17	16	80.8	86.8		16	80.8	86.8
18	17	83.4	83.9		17	83.4	83.9
19	18	79.7	82		18	79.7	82
20	19	76.7	91.8		19	76.7	91.8
21	20	80.3	84.8		20	80.3	84.8
22	21	77.2	89.5		21	77.2	89.5
23	22	76.7	94.4		22	76.7	94.4
24	23	79.2	91		23	79.2	91
25	24	82.1	88.1		24	82.1	88.1
26	25	83.9	83		25	83.9	83
27	26	84.6	82.9		26	84.6	82.9
28	27	79.4	85.5		27	79.4	85.5
29	28	80.4	abs		28	80.4	
30	29	84.6	83.9		29	84.6	83.9
31	30	78.1	86.3		30	78.1	86.3
32	original data				cleaned data - only numeric		

See next page.

Step 1. Import the data into R by copy and paste. Open the `tTestPairedSoTL.R` script file in R. Then, in the spreadsheet copy the entire before and after columns to the “clipboard”. When you copy the two columns, make sure to include the **header row**, i.e., the words “before” and ”after”.

	E	F	G
	StudentID	before	after
	1		80.9
	2	77.4	
	3		
	4	80.5	90
	5		
	6	79.1	78.5
	7	76.7	90.8
	8	76.6	84.4
	9	80.1	93.3
	10	78	84.8
	11	78.8	87.5
	12	81.4	85.6
	13	82.4	91.2
	14	71.2	84.9
	15	80.3	84.6
	16	80.8	86.8
	17	83.4	83.9
	18	79.7	82
	19	76.7	91.8
	20	80.3	84.8
	21	77.2	89.5
	22	76.7	94.4
	23	79.2	91
	24	82.1	88.1
	25	83.9	83
	26	84.6	82.9
	27	79.4	85.5
	28	80.4	
	29	84.6	83.9
	30	78.1	86.3

cleaned data - only numeric

Then, back in R, in the `tTestPairedSoTL.R` script, run the following lines of code:

```
# Import Data From Spreadsheet (Step 1)
df = read.table(file = "clipboard", sep = "\t", header = TRUE);
df; # if you did this right, you will see the students' data in the R console.
```

Step 2. Clean data in R: remove the rows with NA’s. We need to remove the students (i.e, the rows) who didn’t take both the before and the after quiz. The command `na.omit(df)` produces a data frame the same as `df`, except the rows that have NA’s in them are removed. NA = not available.

```
# Clean data in R (remove the rows with NA's)(Step 2)
# -----
dfc = na.omit(df) # dfc = df with rows with blank cells removed
dfc
```

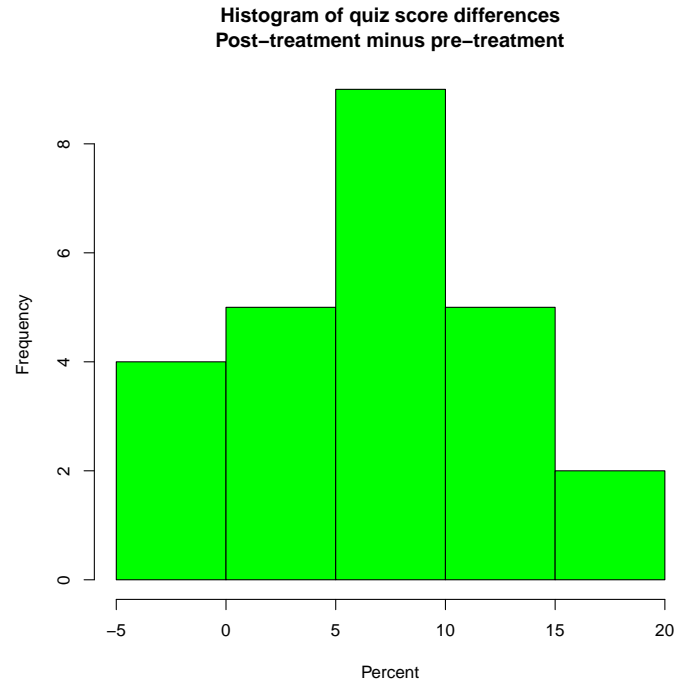
Step 3. Find the mean and standard deviation of quiz scores. We will need this information when we report our results.

```
# Find the mean and standard deviation of the quiz scores (Step 3)
# -----
mean(dfc$before); # the mean of the before scores
sd(dfc$before); # the standard deviation of the before scores

mean(dfc$after); # the mean of the after scores
sd(dfc$after); # the standard deviation of the after scores
```

Step 4. Create a histogram of the quiz score differences. If the treatment is effective most of the histogram will be to the right of 0. The histogram gives a visual representation of how the differences in quiz scores (after – before) are distributed. Histograms are an excellent, visual way to present data.

```
# Create histogram of data (Step 4)
# -----
hist(dfc$after - dfc$before,
     main = "Histogram of quiz score differences\nPost-treatment minus pre-treatment",
     xlab = "Percent",
     col = "green" );
```



Step 5. Do the paired-samples t-test.

```
# Do the paired t-test (Step 5)
# -----
t.test(dfc$after - dfc$before, alternative = "greater", mu = 0);
```

Notes about the above syntax. The R command If we changed the above R command to:

```
t.test(dfc$after - dfc$before, alternative = "greater", mu = 3.1);
```

then R would test the claim that

$$\text{dfc\$after} - \text{dfc\$before} > 3.1,$$

i.e., that the intervention raises each student’s quiz score, on average, more than 3.1 points.

See next page for how to interpret and report the results of the t-test.

Interpreting the t-test results. After we run the t-test in R:

```
# Do the paired t-test (Step 6)
# -----
t.test(dfc$after - dfc$before, alternative = "greater", mu = 0);
```

R outputs the results:

```
One Sample t-test

data:  dfc$after - dfc$before
t = 6.4758, df = 24, p-value = 5.351e-07
alternative hypothesis: true mean is greater than 0
95 percent confidence interval:
 5.230106      Inf
sample estimates:
mean of x
 7.108
```

The most important part of the summary is the p-value. In this example the **p-value = 5.351e-07**, which is less than 0.05. This tells us that our data provides statistically significant evidence that the treatment is effective. Typically, peer reviewed journals will allow us to say we have statistically significant evidence only if the p-value < 0.05.

How to report the results of the paired-samples t-test in your article:

A paired-samples t-test was conducted to compare the pre-intervention quiz scores with post-intervention quiz scores. There was a significant difference in the pre-intervention scores ($M = 79.67$, $SD = 3.028$) and post-intervention scores ($M = 86.78$, $SD = 3.882$); $t(24) = 6.4758$, $p < 0.0001$.

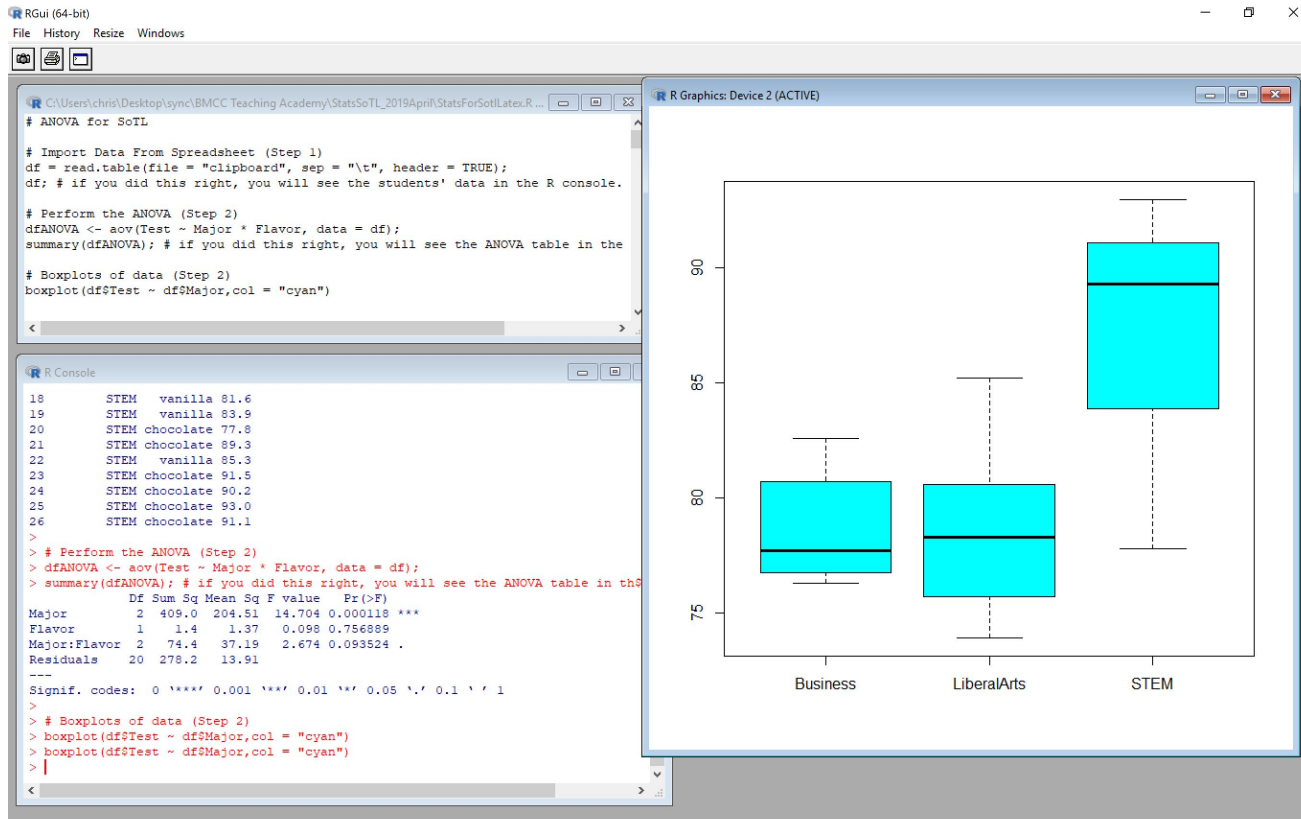
Notes. We report: $t(24) = 6.4758$ because in the summary it says $t = 6.4758$, $df = 24$. If the p-value < .0001 we just report the p-value as $p < 0.0001$

See next page for Appendix: Getting Started with R.

Appendix

Getting Started with R

This section is quick introduction to R to help you get started. You can download R for free from <https://www.r-project.org/>.



The above shows a screen shot of **R on a Windows PC**. R on an Apple will look slightly different. When you start R the “R Console” will appear (see above, bottom left). You can type commands directly into the R Console, however, it is better to run commands from the “R Editor” (top left). In the R Editor you can create “R scripts” which are just R commands in a text file. When you run R commands from the R Editor you’ll see the output in the “R Console” or in an “R Graphics” window (right).

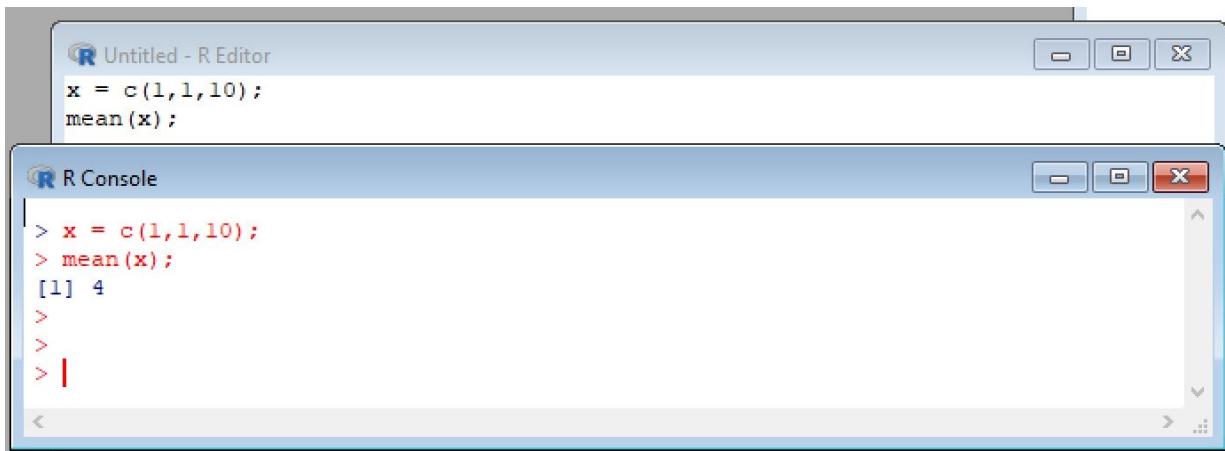
Two minute R Tutorial to Get You Started.

Finding the average of the numbers 1, 1, 10 in R.

1. Open R.
2. Inside of R create the script file by doing the following. Go to the menu, then File, then New script. A blank script will appear in the R Editor. Type, or copy and paste, the following R commands into your R script:

```
x = c(1,1,10);
mean(x);
```

3. Run the script. **On a Windows PC**, to run a line of code in an R script, put your cursor anywhere in that line and do **Ctrl R**. To run a bunch of lines, highlight the lines by doing **Ctrl Shift** and the **Up** or **Down** key, then do **Ctrl R** to run them. **On an Apple Computer**, to run a line of code in an R script, put your cursor anywhere in that line and press **Command Enter** at the same time.
4. The results you should get. After you run this script you should see, in the R console window, the commands from the script displayed together with the number 4, since the mean of 1, 1, 10 is $\frac{1+1+10}{3} = \frac{12}{3} = 4$.

The image shows two overlapping windows from the R environment. The top window, titled "Untitled - R Editor", contains the R code: `x = c(1,1,10);` and `mean(x);`. The bottom window, titled "R Console", shows the execution of this code. The prompt `>` is followed by `x = c(1,1,10);` and `mean(x);`. The output is `[1] 4`, followed by three more `>` prompts, with a vertical cursor on the last one. The console window has a scroll bar on the right side.

```
Untitled - R Editor
x = c(1,1,10);
mean(x);

R Console
> x = c(1,1,10);
> mean(x);
[1] 4
>
>
> |
```

Notes:

1. If you want to save your script file you should save it with the .R extension. Something like `MyScript.R`.
2. To load your R script file into R you need to open R and then from inside of R, go to the menu, File, Open. Double clicking on the R file typically won't load it into R.
3. If you don't know how to do something in R just Google it. R is so widely used that just about any question you might have will have already been asked and answered somewhere online.
4. If you want to put a comment in your R script use the # symbol. Anything on a line after the # symbol is considered by R to be "comment", and won't be compiled.

End of R tutorial.