Assignment 4

Instructions (the same as for Assignment 1): Make an R Notebook and in it answer the two questions below (one Notebook for both questions). When you are done, hand in on Quercus the *output* from Previewing (or Knitting) your Notebook. Do *not* hand in the Notebook itself. You want to show that you can (i) write code that will answer the questions, (ii) run that code and get some sensible output, (iii) write some words that show you know what is going on and that reflect your conclusions about the data. Your goal is to convince the grader that you *understand* what you are doing: not only doing the right thing, but making it clear that you know *why* it's the right thing.

Do *not* expect to get help on this assignment. The purpose of the assignments is for you to see how much *you* have understood. You will find that you also learn something from grappling with the assignments. The time to get help is after you watch the lectures and work through the problems from PASIAS, via tutorial and the discussion board. The only reason to contact the instructor while working on the assignments is to report something missing like a data file that cannot be read.

You have 3 hours to complete this assignment after you start it.

My solutions to this assignment, with extra discussion, will be available after everyone has handed in their assignment.

There is only one question this time, but it is longer. As I write this, the parts continue onto a second page. You will probably need `smmr` as well as the `tidyverse`.

1. Twenty high-school French teachers attended a summer institute to improve their French skills. At the beginning of their session, each teacher took a listening test (to test their understanding of spoken French). After 4 weeks of immersion in French, each teacher took a similar listening test again. (The actual French spoken in the two tests was different, so simply taking the first test should not improve the score in the second one; the tests were otherwise similar.) The maximum score on each test was 36, and a higher score is better. The data are here. (Right-click on the blue text, select "copy link address", and then paste that URL into R Studio.) The data values are separated by *tabs*.

   The data file has three columns:

   - an identification for each teacher
   - the teacher's score in the first test
   - the teacher's score in the second test

   (a) Read in and display (some of) the data.

   Solution:

   Separated by tabs means `read_tsv`:

```
my_url <- "https://raw.githubusercontent.com/nxskok/STAC32/master/frenchtest.txt"
french <- read_tsv(my_url)
```

```
## Parsed with column specification:
## cols(
##   id = col_double(),
##   pre = col_double(),
```

```
##   post = col_double()
## )
```

```
french
```

```
## # A tibble: 20 x 3
##       id   pre  post
##    <dbl> <dbl> <dbl>
##  1     1    32    34
##  2     2    31    31
##  3     3    29    35
##  4     4    10    16
##  5     5    30    33
##  6     6    33    36
##  7     7    22    24
##  8     8    25    28
##  9     9    32    26
## 10    10    20    26
## 11    11    30    36
## 12    12    20    26
## 13    13    24    27
## 14    14    24    24
## 15    15    31    32
## 16    16    30    31
## 17    17    15    15
## 18    18    32    34
## 19    19    23    26
## 20    20    23    26
```

As promised. The score on the first test is called `pre` and on the second is called `post`.

(b) Explain briefly why this is a matched-pairs study.

Solution:

There are two measurements for each teacher, or, the 20 `pre` measurements and the 20 `post` measurements are paired up, namely, the ones that come from the same teacher. Or, if it were two independent samples, our 40 measurements would come from 40 different teachers, but there are only 20 teachers, so the 40 measurements must be paired up.

(c) Run a suitable matched-pairs t-test to see whether the teachers' scores have on average *improved* over the four weeks.

Solution:

Seeing whether the scores have improved implies a *one*-sided test that `post` is bigger than `pre`. There are three ways you might do that, any of which is good. Remember that if you are running a test with `paired = TRUE`, the alternative is relative to the column that is *input first*, not the first one in alphabetical order or anything like that:

(i):

```
with(french, t.test(pre, post, paired = TRUE, alternative = "less"))
```

```
##
##  Paired t-test
##
## data:  pre and post
## t = -3.8649, df = 19, p-value = 0.0005216
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##       -Inf -1.381502
## sample estimates:
## mean of the differences
##                    -2.5
```

(ii):

```
with(french, t.test(post, pre, paired = T, alternative = "greater"))
```

```
##
##  Paired t-test
##
## data:  post and pre
## t = 3.8649, df = 19, p-value = 0.0005216
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  1.381502      Inf
## sample estimates:
## mean of the differences
##                     2.5
```

Your choice between these two might be influenced by whether you think `pre` comes first, or whether you think it's easier to decide how `post` compares to `pre`. It's all down to what seems natural to you.

(iii) working out the differences and testing those (but look ahead in the question to see whether you need the differences for anything else: you do):

```
french %>% mutate(gain = post - pre) -> french1
with(french1, t.test(gain, mu=0, alternative = "greater"))
```

```
##
##  One Sample t-test
##
## data:  gain
## t = 3.8649, df = 19, p-value = 0.0005216
## alternative hypothesis: true mean is greater than 0
## 95 percent confidence interval:
##  1.381502      Inf
## sample estimates:
## mean of x
##       2.5
```

This last is an ordinary one-sample test, which saves you having to learn anything new, but requires you to calculate the differences first. You will need the differences for a plot anyway,

so this may not be as much extra work as it appears. The right thing to do here is to *save* the data frame with the differences in it, so that you don't need to calculate them again later.

A fourth alternative is to calculate the differences as `pre` minus `post`, and then switch the `alternative` around (since if going to the French institute helps, the differences this way will be mostly *negative*):

```
french %>% mutate(gain = pre - post) -> french2
with(french2, t.test(gain, mu=0, alternative = "less"))
```

```
##
##  One Sample t-test
##
## data:  gain
## t = -3.8649, df = 19, p-value = 0.0005216
## alternative hypothesis: true mean is less than 0
## 95 percent confidence interval:
##       -Inf -1.381502
## sample estimates:
## mean of x
##      -2.5
```

(d) What do you conclude from your test, in the context of the data?

Solution:

The P-value of 0.0005 is much less than 0.05, so we reject the null hypothesis that the mean scores before and after are the same, in favour of the alternative that the mean score afterwards is higher. That is to say, the four-week program is helping the teachers improve their understanding of spoken French.

(e) How much is the teachers' listening skill improving, on average? Give a suitable interval to support your answer.

Solution:

A 95% (or other level) confidence interval for the mean difference. A one-sided test doesn't give that, so you need to do the test again without the `alternative` (to make it two-sided), via any of the methods above, such as:

```
with(french, t.test(post, pre, paired = T))
```

```
##
##  Paired t-test
##
## data:  post and pre
## t = 3.8649, df = 19, p-value = 0.001043
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.146117 3.853883
## sample estimates:
## mean of the differences
##                     2.5
```

This says that, with 95% confidence, the mean test score afterwards is between about 1.1 and 3.9 points higher than before. So that's how much listening skill is improving on average. Give the suitably rounded interval; the test scores are whole numbers, and there are 20 differences making up the mean, so one decimal is the most you should give.

If you did it the first way:

```
with(french, t.test(pre, post, paired = T))
```

```
##
##  Paired t-test
##
## data:  pre and post
## t = -3.8649, df = 19, p-value = 0.001043
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.853883 -1.146117
## sample estimates:
## mean of the differences
##                    -2.5
```

you have given yourself a bit of work to do, because this is before minus after, so you have to strip off the minus signs and switch the numbers around. Giving the answer with the minus signs is wrong, because I didn't ask about before minus after. Disentangle it, though, and you're good.

(f) Make a suitable plot to assess any assumptions for this test.

Solution:

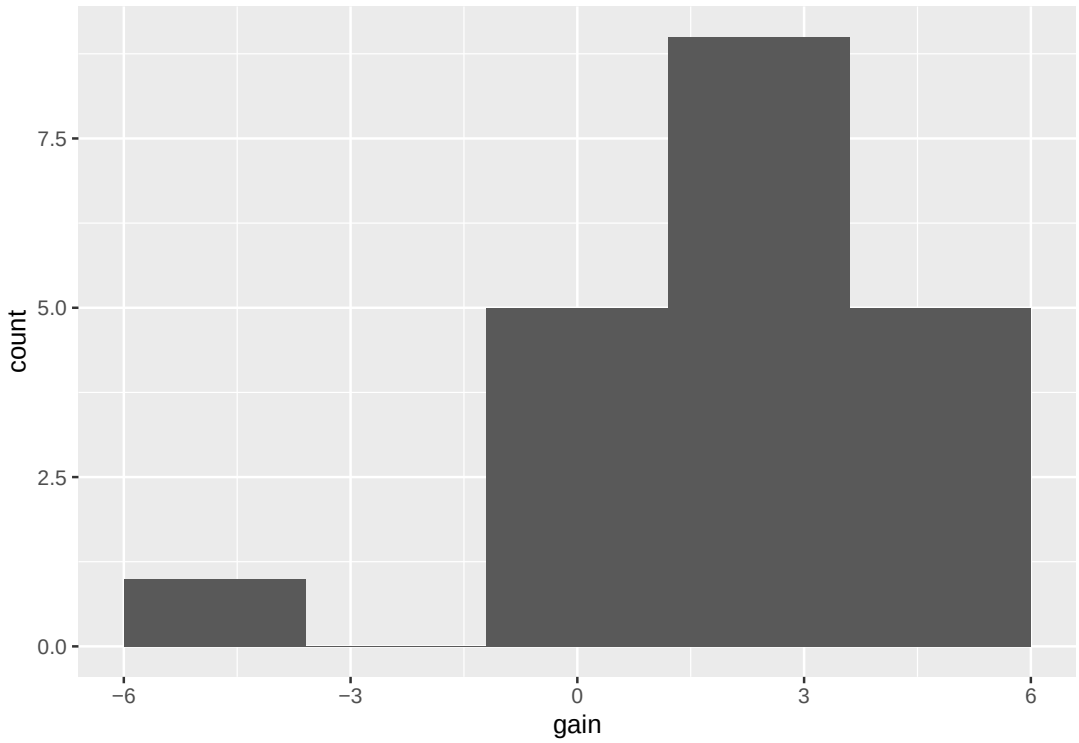
The key assumption here is that the *differences* are approximately normally distributed.

First calculate *and save* the differences (since you will need them later for a sign test; otherwise you would have to find them again). If you found the differences to make your t-test, use the ones you saved there.

```
french %>% mutate(gain = post - pre) -> french1
```

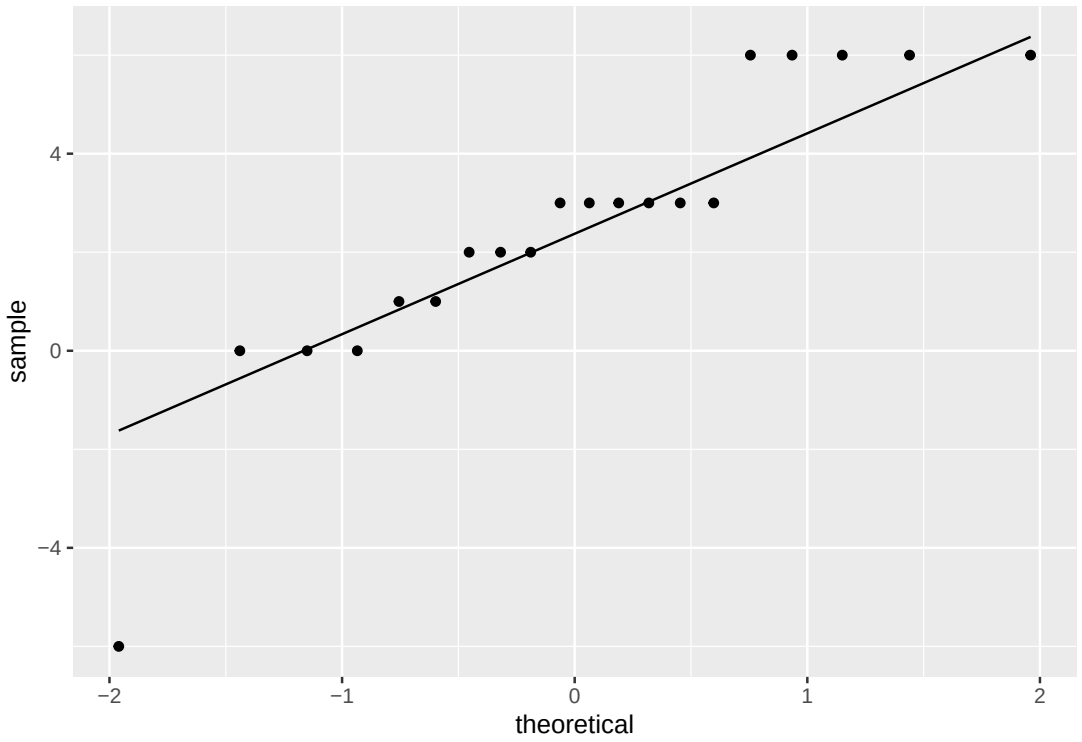
Assess that with a histogram (with suitable number of bins):

```
ggplot(french1, aes(x=gain)) + geom_histogram(bins=6)
```



or, better, a normal quantile plot (since the normality is our immediate concern):

```
ggplot(french1, aes(sample=gain)) + stat_qq() + stat_qq_line()
```



(note that the horizontal lines of points are because the test scores were whole numbers, therefore the differences between them are whole numbers also, and some of the teachers had the same difference in scores as others.)

(g) Do you trust the result of your matched-pairs t-test? Explain briefly.

Solution:

There are about three considerations here:

- the plot shows an outlier at the low end, but no other real problems.
- the sample size is 20, so we should get some help from the Central Limit Theorem.
- the P-value was really small.

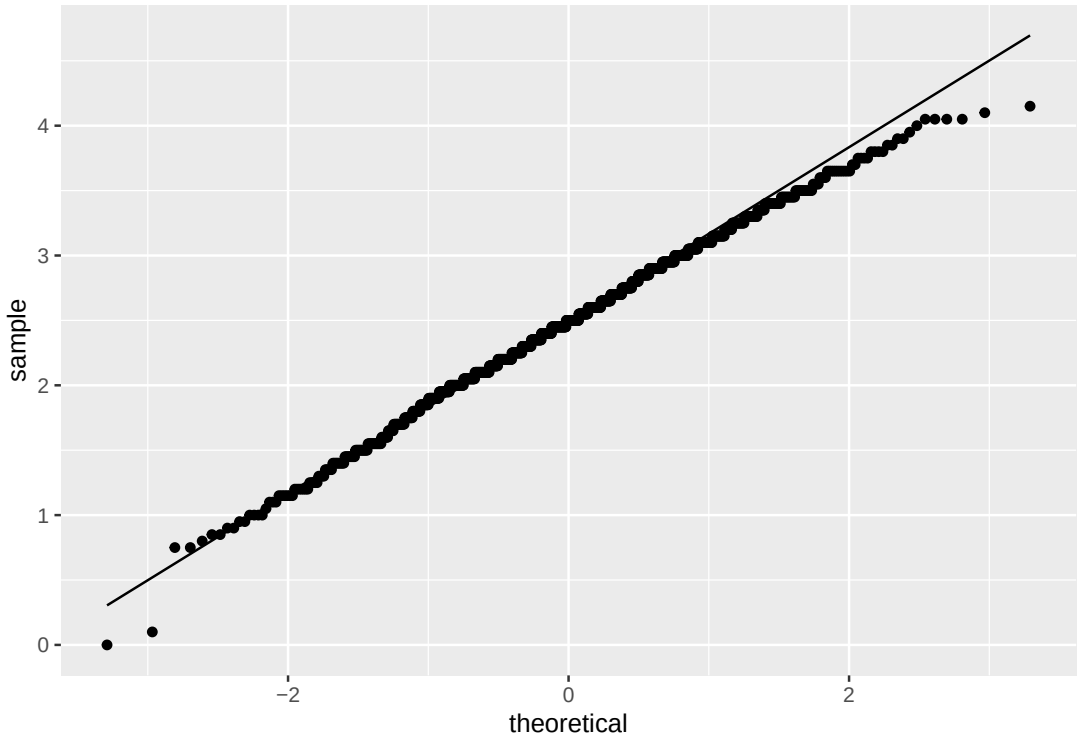
I expect you to mention the first two of those. Make a call about whether you think that outlier is too much of a problem, given the sample size. You could, I think, go either way with this one.

The third of my points says that even if the distribution of differences is not normal enough, and so the P-value is off by a bit, it would take a lot to change it enough to stop it being significant. So I don't think we need to worry, for myself.

Extra:

We can assess the t-test by obtaining a bootstrap distribution of the sample mean, by sampling from the differences with replacement:

```
rerun(1000, sample(french1$gain, replace = T)) %>%
  map_dbl(~mean(.)) %>%
  enframe() %>%
  ggplot(aes(sample=value)) + stat_qq() + stat_qq_line()
```



The bootstrapped sampling distribution of the sample mean difference is about as normal as you could reasonably wish for, so there was no need to worry. Only a very few of the most extreme samples were at all off the line.

A histogram would be almost as good, but now that you know about the normal quantile plot, the time to use it is when you are *specifically* interested in normality, as you are here. (If you were interested in shape generally, then a histogram or, if appropriate, a boxplot, would also work.)

The code: the first line takes 1000 bootstrap samples, and the second finds the mean of each one. Instead of saving the sample means, since I was only going to be using them once, I made them into a dataframe, and then made a normal quantile plot of them. The `enframe` creates a dataframe with a column called `value` with the means in it, which I use in the plot.

(h) Run a suitable sign test, and obtain a suitable (95%) confidence interval. Comment briefly on your results.

Solution:

This works with the differences, that you calculated for the plot, so use the data frame that you saved them in:

```
sign_test(french1, gain, 0)
```

```
## $above_below
## below above
##     1    16
##
## $p_values
##   alternative      p_value
## 1       lower 0.9999923706
## 2       upper 0.0001373291
## 3   two-sided 0.0002746582
```

```
ci_median(french1, gain)
```

```
## [1] 1.007812 3.000000
```

The P-value is 0.00014, again very small, saying that the median difference is greater than zero, that is, that the test scores after are greater than the test scores before on average. The confidence interval is from 1 to 3 points, indicating that this is how much test scores are increasing on average.

A technique thing: the first time you are going through this, you probably got to this point and realized that you were calculating the differences for the second (or third) time. This is the place to stop and think that you don't really need to do that, and to go back to the plot you did and *save* the differences after you have calculated them. Then you edit the code here to use the differences you got before and saved. It doesn't matter whether you see this the first time you do it or not, but it does matter that you see it before you hand it in. It's like editing an essay; you need to go back through work that you will be handing in and make sure you did it the best way you could.

(i) Comment briefly on the comparison between your inferences for the mean and the median.

Solution:

The upper-tail P-value is 0.0001, in the same ballpark as the t-test (0.0005). The 95% confidence interval for the median difference is from 1 to 3,[1] again much like the t-interval (1.1 to 3.9).

This suggests that it doesn't matter much which test we do, and therefore that the t-test ought to be better because it uses the data better.[2] This is more evidence that the outlier didn't have that big of an effect.

Extra: choosing a test on the basis of its P-value is *wrong*, because as soon as you introduce a choice on that basis, your P-value looks lower than it should; a P-value is based on you doing one test and only that one test. It is reasonable to note, as I did, that the two P-values are about the same and then choose between the tests on *some other basis*, such as that the t-test uses the data better.[3]

Notes

1. I think I mentioned elsewhere that the P-value of the sign test, as it depends on the null median for a fixed data set, only changes at a data point. Therefore, the ends of a CI for the median must be data points.

2. It uses the actual data values, not just whether each one is positive or negative.

3. If the P-values had come out very different, that would be a sign that it matters which one you use, and you would need to go back and look at your plot to decide. Often, this happens when there is something wrong with the t-test, but not necessarily.