

Assignment 4

Instructions: Make an R Notebook and in it answer the questions below. When you are done, hand in on Quercus the *output* from Previewing (or Knitting) your Notebook, probably an **html** or **pdf** file. 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, that is *before* you start work on the assignment. The only reasons to contact the instructor while working on an assignment are to report (i) something missing like a data file that cannot possibly be read, (ii) something *beyond your control* that makes it impossible to finish the assignment in time after you have started it.

There is a time limit on this assignment (you will see Quercus counting down the time remaining).

1. Going to the dentist is scary for a lot of people. One way in which this might show up is that people might have higher blood pressure on average before their dentist's appointment than an hour after the appointment is done. Ten randomly-chosen individuals have their (systolic¹) blood pressure measured while they are in a dentist's waiting room, and then again one hour after their appointment is finished.

The data are in http://ritsokiguess.site/STAC32/blood_pressure1.csv.

- (a) Read in and display the data.

Solution:

```
my_url <- "http://ritsokiguess.site/STAC32/blood_pressure1.csv"
blood_pressure <- read_csv(my_url)
```

```
##
## -- Column specification -----
## cols(
##   person = col_character(),
##   before = col_double(),
##   after = col_double()
## )
```

```
blood_pressure
```

```
## # A tibble: 10 x 3
##   person before after
##   <chr>   <dbl> <dbl>
## 1 p1      132    118
## 2 p2      135    137
## 3 p3      149    140
```

```
## 4 p4      133  139
## 5 p5      119  107
## 6 p6      121  116
## 7 p7      128  122
## 8 p8      132  124
## 9 p9      119  115
## 10 p10    110  103
```

(b) What kind of experimental design is this? How do you know? Explain briefly.

Solution:

This is a matched pairs design. We know this because we have two measurements on each person, or the same people were measured before and after seeing the dentist. (The thing that it is *not* is one group of people measured before seeing the dentist, and a *different* group of people measured afterwards, so a two-sample test is *not* the right thing.)

(c) Run a suitable t -test on these data. What do you conclude, in the context of the data?

Solution:

A matched-pairs t -test, then. Remember, we want to see whether blood pressure is *lower* afterwards (that is, before is *greater* than after), so this needs to be one-sided:

```
with(blood_pressure, t.test(before, after, alternative = "greater", paired = TRUE))
```

```
##
## Paired t-test
##
## data: before and after
## t = 2.9945, df = 9, p-value = 0.007545
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  2.210659      Inf
## sample estimates:
## mean of the differences
##                5.7
```

There are some variations possible here: `before` and `after` could be switched (in which case `alternative` must be reversed also).

Or, you can do a one-sample t on the differences, with the right `alternative` corresponding to the way you took differences. If you are looking ahead, you might realize that working out the differences *now* and adding them to the dataframe will be a good idea:

```
blood_pressure %>%
  mutate(difference = before - after) -> blood_pressure
blood_pressure

## # A tibble: 10 x 4
##   person before after difference
##   <chr>   <dbl> <dbl>     <dbl>
## 1 p1      132  118         14
```

```
## 2 p2      135  137      -2
## 3 p3      149  140       9
## 4 p4      133  139      -6
## 5 p5      119  107      12
## 6 p6      121  116       5
## 7 p7      128  122       6
## 8 p8      132  124       8
## 9 p9      119  115       4
## 10 p10     110  103       7
```

I took the differences this way around since I was expecting, if anything, the before numbers to be bigger than the after ones. And then:

```
with(blood_pressure, t.test(difference, mu = 0, alternative = "greater"))
```

```
##
## One Sample t-test
##
## data:  difference
## t = 2.9945, df = 9, p-value = 0.007545
## alternative hypothesis: true mean is greater than 0
## 95 percent confidence interval:
##  2.210659      Inf
## sample estimates:
## mean of x
##      5.7
```

If you did the differences the other way around, your `alternative` will need to be the other way around also.

The P-value (either way) is 0.008,² so we have evidence that the mean blood pressure before is greater than the mean blood pressure after.

- (d) Run a suitable sign test on these data. What do you conclude, in the context of the data?

Solution:

A sign test on the differences. By this point, you will realize that you will need to have obtained the differences. Get them here if you did not already get them:

```
sign_test(blood_pressure, difference, 0)
```

```
## $above_below
## below above
##      2      8
##
## $p_values
##   alternative   p_value
## 1      lower 0.9892578
## 2      upper 0.0546875
## 3 two-sided 0.1093750
```

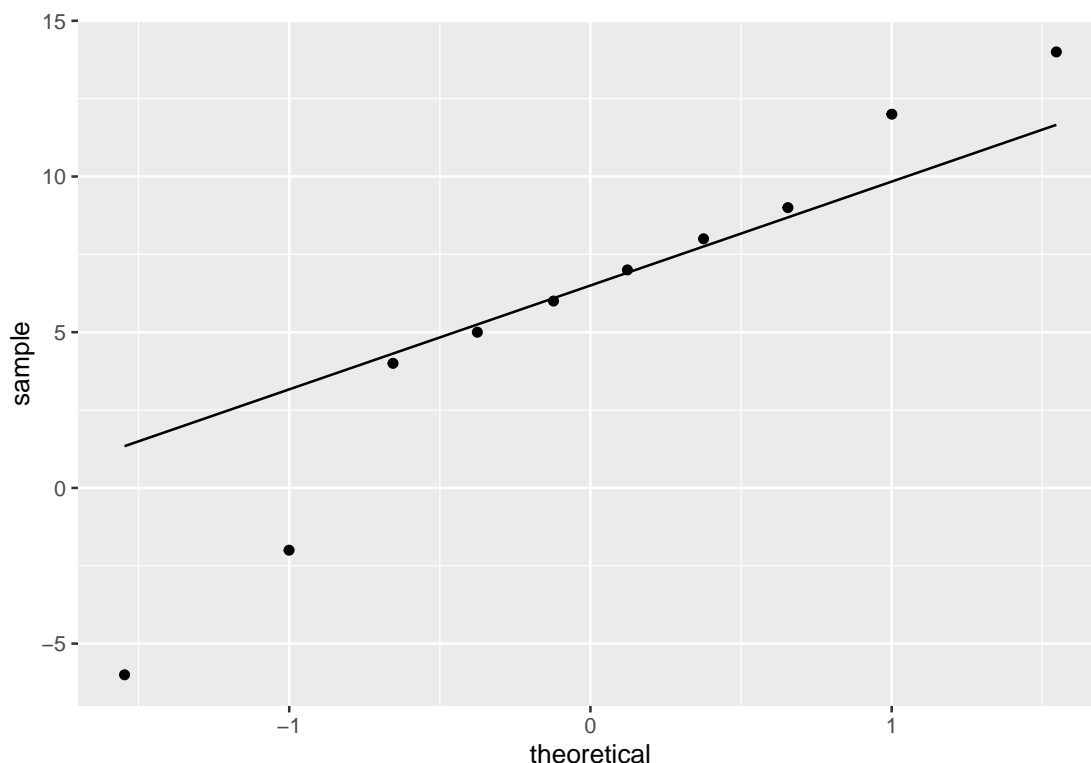
This one gives us all three P-values. The way around I found the differences, the one we want is “upper”, 0.055. There is not quite evidence that median blood pressure before is higher.

- (e) Draw a suitable normal quantile plot of these data, one that will enable you to decide between the tests you ran in the previous two parts.

Solution:

The *differences* are supposed to be approximately normal if a matched-pairs *t*-test is the thing:

```
ggplot(blood_pressure, aes(sample=difference)) + stat_qq() + stat_qq_line()
```



- (f) Discuss briefly which of your two tests is the more appropriate one to run.

Solution:

Make a call about whether the differences are normal enough. You have a couple of angles you can take:

- the lowest two values are too low, so we have two outliers at the low end
- the lowest *and* highest values are too extreme, so that we have a long-tailed distribution

Either of these would suggest a non-normal distribution, which I think you have to conclude from this plot.

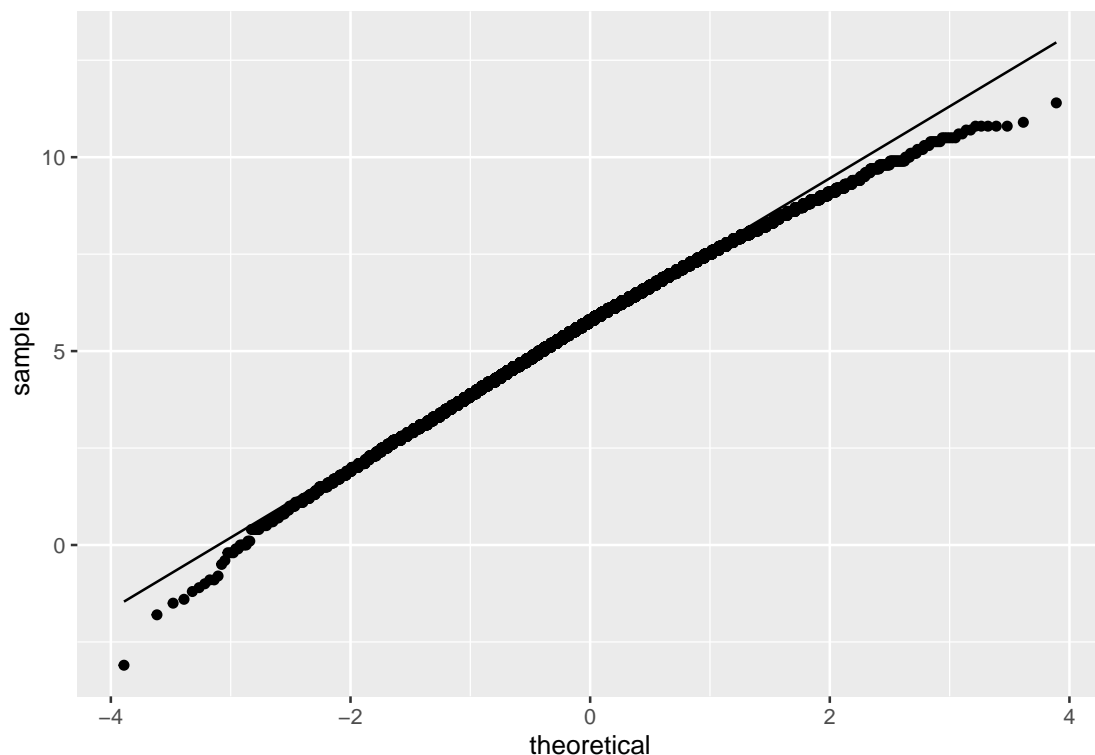
The best answer also considers the sample size: there are only 10 differences, a small sample size, and so we will not get much help from the Central Limit Theorem (the sample size is likely not enough³ to overcome those two outliers or the long tails). Thus, we should *not* trust the *t*-test and should prefer the sign test.

Extra: you might be disappointed to go through this and come to the conclusion that there was not a decrease in blood pressure between before and after.

What has happened, I think, is that we have only a small sample (10 people), and having 8 positive differences and 2 negative ones is not quite unbalanced enough (with such a small sample) to rule out chance: that is to say, a median difference of zero. The t -test accounted for the size of the differences, and if you believed the normality was satisfactory, you could demonstrate a difference between before and after. But if you didn't like the normality, you were out of luck: the only test you have is an apparently not very powerful one.

If you wanted to, you could bootstrap the sampling distribution of the sample mean and see how normal it looks:

```
rerun(10000, sample(blood_pressure$difference, replace = TRUE)) %>%  
  map_dbl(~mean(.)) %>%  
  enframe() %>%  
  ggplot(aes(sample = value)) + stat_qq() + stat_qq_line()
```



(Code note: you can do anything with the result of a simulation, and you can use anything that might need to be normal as input to a normal quantile plot. Now that we have the normal quantile plot as a tool, we can use it wherever it might be helpful.)

This is actually not nearly as bad as I was expecting. Even a sample size of 10 is providing some help. The bootstrapped sampling distribution is somewhat left-skewed, which is not a surprise given the two low outliers. However, it is rather close to normal, suggesting that the t -test is not as bad as we thought.

(I did 10,000 simulations because I was having trouble seeing how non-normal it was. With this many, I can be pretty sure that this distribution is somewhat left-skewed.)

There is a second question on the next page.

2. Take your right hand, and stretch the fingers out as far as you can. The distance between the tip of your thumb and the tip of your little (pinky) finger is your handspan. The students in a Statistics class at Penn State measured their handspans and also whether they identified as male or female. The data are at <http://ritsokiguess.site/STAC32/handspan.txt>, with handspans measured in inches. We want to see whether male students have a larger mean handspan than female students.

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

Solution:

Delimited by a single space, so:

```
my_url <- "http://ritsokiguess.site/STAC32/handspan.txt"
span <- read_delim(my_url, " ")
```

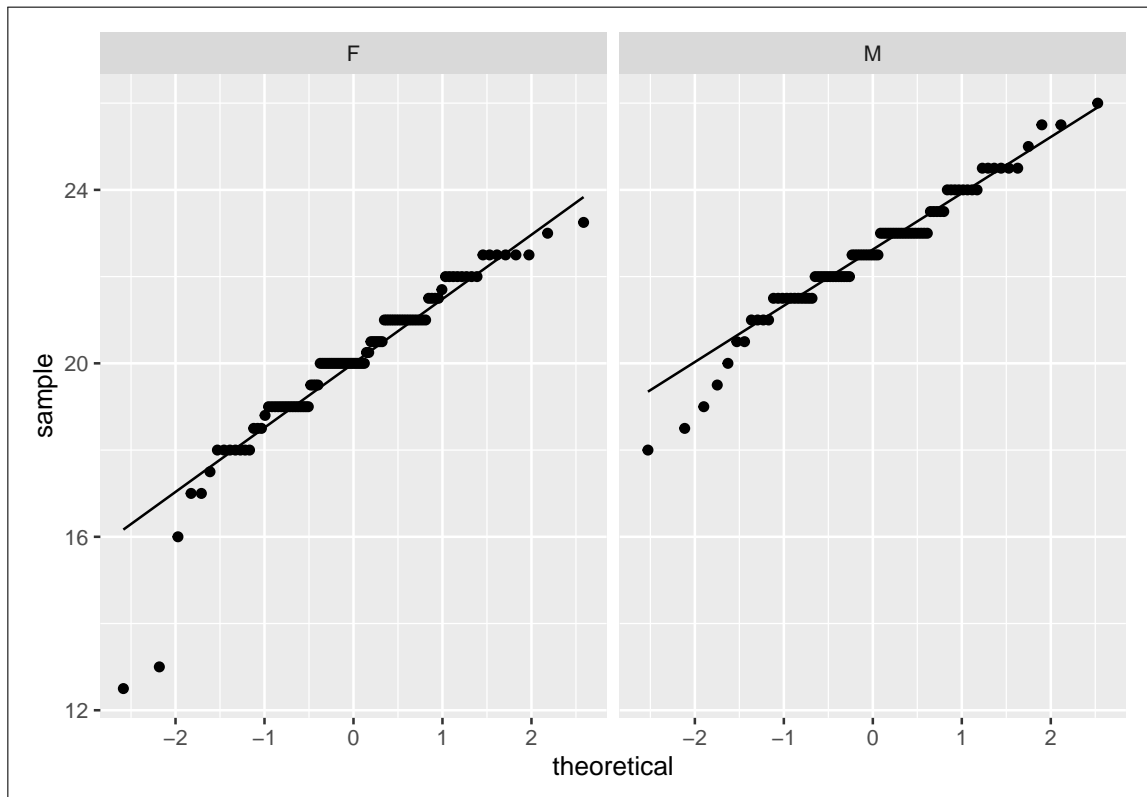
```
##
## -- Column specification -----
## cols(
##   sex = col_character(),
##   handspan = col_double()
## )
span

## # A tibble: 190 x 2
##   sex    handspan
##   <chr>    <dbl>
## 1 M        21.5
## 2 M        22.5
## 3 M        23.5
## 4 F         20
## 5 F         19
## 6 F        20.5
## 7 F        20.5
## 8 F        20.2
## 9 M         23
## 10 M       24.5
## # ... with 180 more rows
```

- (b) Make a suitable (faceted) normal quantile plot of the data. (Bear in mind what is supposed to have a normal distribution.)

Solution: Here, we need *each* group to be approximately normal, so make normal quantile plots of handspan, faceted by sex:

```
ggplot(span, aes(sample=handspan)) + stat_qq() + stat_qq_line() +
  facet_wrap(~sex)
```



- (c) Discuss briefly whether you might prefer to use Mood's median test to compare the handspans of the male and female students, compared to a two-sample t -test.

Solution: A two-sample t -test assumes that each of the two samples comes from a (approximately) normal distribution ("the data are normal" is not precise enough). The female values, on the left, definitely have some outliers at the low end (or a long lower tail), so these are definitely not normal. The male values (on the right) are slightly skewed to the left, or there are some mild outliers at the low end, or, if you prefer, these are approximately normal. (You need discussion of each of the males and females, or of why looking at one group is enough.) Because the males are not close enough to normal (or, because neither group is close enough to normal), we would prefer to use Mood's median test. (Say this.) You do yourself a favour by making it clear that you know that *both* groups have to be normal enough; if one is good but the other is not, that is not enough.

The other relevant issue is sample size. The best answer discusses that as well, even though you have a lot to think about already. This data set has 190 observations in it, so the samples must be pretty big:

```
span %>% count(sex)
```

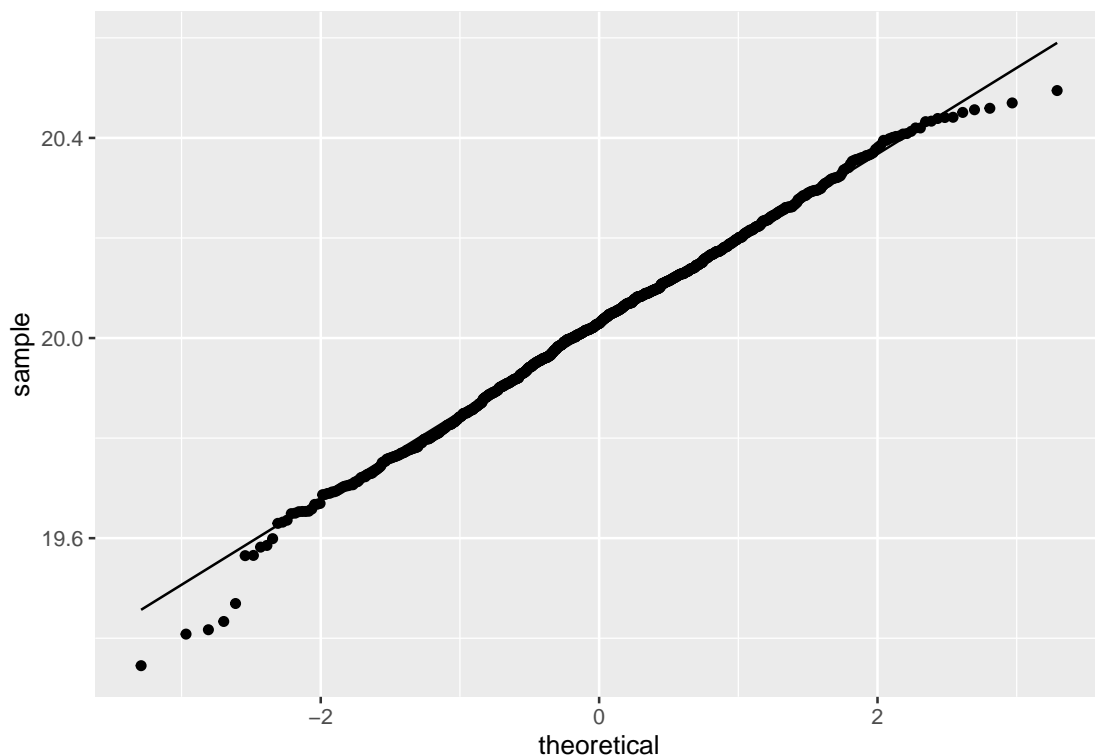
```
## # A tibble: 2 x 2
##   sex      n
## * <chr> <int>
## 1 F      103
## 2 M       87
```

With these sample sizes, we can expect a lot of help from the central limit theorem. The

apparent outliers in the males won't be a problem, and maybe we could even get away with those outliers in the females.

Extra: you could also think about bootstrapped sampling distributions of the sample mean here. The one we are most concerned about is the females; if it turns out that they are all right, then the males must be all right too, since the plot for them is showing less non-normality (or, without the double negative, is closer to being normal). So let's do the females:

```
span %>%  
  filter(sex == "F") -> females  
rerun(1000, sample(females$handspan, replace = TRUE)) %>%  
  map_dbl(~mean(.)) %>%  
  enframe() %>%  
  ggplot(aes(sample = value)) + stat_qq() + stat_qq_line()
```



My take is that the sampling distribution of the sample mean for the females is normal enough, therefore the one for the males is also normal enough, therefore the two-sample t -test is actually fine.

The reason that this one is close to normal is different from the other one, though. In the other question, we had milder non-normality but a smaller sample; in this one, the data distribution is less normal, but we had a much larger sample size to compensate.

(d) Run Mood's median test. What do you conclude from the test, in the context of the data?

Solution:


```
library(smmr)
median_test(span, handspan, sex)
```

```
## $table
##      above
## group above below
##    F     17    82
##    M     65    11
##
## $test
##      what      value
## 1 statistic 8.06725e+01
## 2          df 1.00000e+00
## 3    P-value 2.66404e-19
```

The P-value of 2.66×10^{-19} is extremely small, so we can conclude that males and females have different median handspans. Remember that we are now comparing medians, and that this test is *two-sided*.

You can stop here, or you can go on and note that most of the males have a handspan bigger than the median, and most of the females have a handspan smaller than the median, so that males have on average a larger handspan. But you have to make the case that males have a larger handspan; you cannot just assert this from the P-value.

A more formal way to do this is to make the same observation as above, then note that this is “on the correct side” (for males to have a larger handspan), and thus that you can halve the P-value, and conclude that males’ handspans are indeed larger in terms of median.

Extra: you are probably expecting a confidence interval now for the difference in medians. I haven’t talked about that in lecture, because the ideas are a bit trickier than they were for the confidence interval for the sign test. The sign test could be used for testing any median, so we could try a bunch of medians and see whether each one was rejected or not. The problem with Mood’s median test is that it only tests that the medians are *the same*. If you could easily test that the difference in medians was 3, say, you would know whether 3 was inside or outside the confidence interval for the difference in medians.

What were the actual sample medians, anyway?

```
span %>% group_by(sex) %>%
  summarize(med = median(handspan))
```

```
## # A tibble: 2 x 2
##   sex    med
## * <chr> <dbl>
## 1 F      20
## 2 M     22.5
```

Here’s an idea: if we shift all the female handspans up by 2.5 inches, the medians would be the same:

```
span %>% mutate(x = ifelse(sex=="F", handspan+2.5, handspan)) -> d
d
```

```
## # A tibble: 190 x 3
##   sex    handspan    x
```

```
##      <chr>      <dbl> <dbl>
##  1 M          21.5  21.5
##  2 M          22.5  22.5
##  3 M          23.5  23.5
##  4 F          20    22.5
##  5 F          19    21.5
##  6 F          20.5  23
##  7 F          20.5  23
##  8 F          20.2  22.8
##  9 M          23    23
## 10 M          24.5  24.5
## # ... with 180 more rows
```

Dataframe `d` has a new column `x` that is the handspan plus 2.5 inches for females, and the unchanged handspan for males. So the median of `x` should be the same for males and females:

```
d %>% group_by(sex) %>%
  summarize(med_x = median(x))
```

```
## # A tibble: 2 x 2
##   sex   med_x
## * <chr> <dbl>
## 1 F     22.5
## 2 M     22.5
```

and also the medians of `x` cannot possibly be significantly different:

```
median_test(d, x, sex)
```

```
## $table
##      above
## group above below
##   F     46    36
##   M     41    35
##
## $test
##      what      value
## 1 statistic 0.07369901
## 2         df 1.00000000
## 3   P-value 0.78602526
```

Quite a lot of the values of `x` are exactly equal to the overall median (and are discarded), so the P-value is not exactly 1 as you would expect. But it is definitely not significant, and so a difference of 2.5 inches smaller for females is going to be in a confidence interval for the difference in medians.

The strategy now is to try shifting the female handspans by different amounts, run Mood's median test for each one, and see which shifts are not rejected. These are the ones for which that difference in medians would be in the confidence interval. Before we get to that, though, I want to simplify the procedure we have, so that it is easier to run it lots of times. First, let's get just the P-value out of the median test:

```
d.1 <- median_test(d, x, sex)
d.1 %>% pluck("test", "value", 3)
```

```
## [1] 0.7860253
```

That's the P-value. `pluck` pulls individual things out of bigger things. The variable I called `d.1` has two things in it. The one called `table` has the numbers of data values above and below the overall median; the one called `test` has the test statistic and P-value in it. `test` is a dataframe; inside *that* is a column called `what` and a column called `value` with the number we want in it, and we want the third thing in that (the other two are the chi-squared test statistic and its degrees of freedom). Hence the `pluck` statement got the right thing.

Let's think strategy: we want to shift the female handspans by a bunch of different amounts, run the test on each one, and get the P-value each time. When you're running a big for-each like this, you want the thing you do each time to be as simple as possible. So let's write a function that takes the shift as input, works out the new `x`, runs the test, and returns the P-value. We have all the ingredients, so it's a matter of putting them together:

```
shift_pval <- function(shift) {  
  span %>% mutate(x = ifelse(sex == "F", handspan + shift, handspan)) -> d  
  d.1 <- median_test(d, x, sex)  
  d.1 %>% pluck("test", "value", 3)  
}
```

In the function, the `shift` is input. The first line computes the handspans shifted by the input amount, whatever it is; the second line runs the median test on the shifted data; the last line pulls out, and returns, the P-value.

I am being a little sloppy here (but R is letting me get away with it): the function is also using a dataframe called `span`, which is the one we read in from the file earlier. That *was not input to the function*, so, if you have experience with other programming languages, you might be wondering whether that is “in the scope” of inside the function: that is, whether R will know about it. R does; anything the function needs that is not part of the input, it will take from your workspace. This is, you might imagine, dangerous; if the input to your function is called, say, `x`, you might easily have an `x` lying around in your workspace from some other analysis that has nothing to do with the `x` you want as the input to your function. The safe way to do it, and what I should have done, is to have `span` be input to my function as well. However, that clutters up the discussion below, so we'll leave things as I did them here.

Let's test this on a shift of 2.5 inches, and on the original data (a shift of zero):

```
shift_pval(2.5)
```

```
## [1] 0.7860253
```

```
shift_pval(0)
```

```
## [1] 2.66404e-19
```

Those are the same P-values we got before, so good.

Now, let's get a bunch of shifts, say from 0 to 5 in steps of 0.5:

```
tibble(shift = seq(0, 5, 0.5))
```

```
## # A tibble: 11 x 1  
##   shift  
##   <dbl>  
## 1    0  
## 2  0.5
```

```
## 3 1
## 4 1.5
## 5 2
## 6 2.5
## 7 3
## 8 3.5
## 9 4
## 10 4.5
## 11 5
```

work out the P-value for each one (this is `map_dbl` because each P-value is a single number):

```
tibble(shift = seq(0, 5, 0.5)) %>%
  mutate(p_value = map_dbl(shift, ~shift_pval(.)))
```

```
## # A tibble: 11 x 2
##   shift p_value
##   <dbl>   <dbl>
## 1 0 2.66e-19
## 2 0.5 7.38e-15
## 3 1 6.41e- 9
## 4 1.5 5.46e- 5
## 5 2 5.29e- 2
## 6 2.5 7.86e- 1
## 7 3 5.15e- 3
## 8 3.5 1.13e- 5
## 9 4 1.89e- 9
## 10 4.5 1.84e-14
## 11 5 1.61e-18
```

and finally decide whether each shift is inside or outside the CI (because I am too lazy to figure out the scientific notation):

```
tibble(shift = seq(0, 5, 0.5)) %>%
  mutate(p_value = map_dbl(shift, ~shift_pval(.))) %>%
  mutate(where = ifelse(p_value<0.05, "outside", "inside"))
```

```
## # A tibble: 11 x 3
##   shift p_value where
##   <dbl>   <dbl> <chr>
## 1 0 2.66e-19 outside
## 2 0.5 7.38e-15 outside
## 3 1 6.41e- 9 outside
## 4 1.5 5.46e- 5 outside
## 5 2 5.29e- 2 inside
## 6 2.5 7.86e- 1 inside
## 7 3 5.15e- 3 outside
## 8 3.5 1.13e- 5 outside
## 9 4 1.89e- 9 outside
## 10 4.5 1.84e-14 outside
## 11 5 1.61e-18 outside
```

The confidence interval goes from 2 inches to 2.5 inches on this scale. I checked and it goes up to 3 really, except that 3 itself is outside the interval. So let's call it 2 to 3 inches. This

means that the median female handspan is between 2 and 3 inches *smaller* than the median male handspan, because we had to shift the female handspans up by that much to make them not significantly different.

You, of course, would do just the last pipeline; I showed you the steps so you could see what was going on.

The final observation is that this interval is a long way from containing zero, because the P-value was so tiny.

Let's see how the t -interval looks in comparison (two-sided, because we want the confidence interval):

```
t.test(handspan~sex, data = span)

##
##  Welch Two Sample t-test
##
## data:  handspan by sex
## t = -10.871, df = 187.92, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.001496 -2.079466
## sample estimates:
## mean in group F mean in group M
##      20.01699      22.55747
```

Almost exactly the same (except that F is before M). So it made no difference at all whether we did a t -test or a Mood's median test, as the bootstrapped sampling distribution suggested.

Notes

1. A blood pressure is usually given as two numbers, like "120 over 80". The first number, which is the one shown in our data, is called the systolic blood pressure. It is the pressure in the arteries when the heart is pumping. The second is called the diastolic blood pressure, and it is the pressure in the arteries when the heart is resting.
2. Give the P-value, and round it off to about this accuracy so that your reader can see easily (i) how it compares to 0.05, and (ii) how big it is. More than two decimal places is too many.
3. But see the Extra.