STAC33 Assignment 2

Due Tuesday January 28 at 11:59pm

Hand in your answers to Questions 2 and 4. Questions 1 and 3 contain suggested problems from PASIAS to work through. They may contain hints for the questions to hand in.

The assignment is due on the date shown above. An assignment handed in after the deadline is late, and may or may not be accepted (see course outline). My solutions to the assignment questions will be available when everyone has handed in their assignment.

You are reminded that work handed in with your name on it must be *entirely your own work*.

Assignments are to be handed in on Quercus. See `https://www.utsc.utoronto.ca/~butler/c32/quercus1.nb.html` for instructions on handing in assignments in Quercus. Markers' comments and grades will be available there as well.

You will probably want to begin with this:

```
library(tidyverse)
```

1. Work through Chapter 6 of PASIAS.
2. Children in a psychology study were asked to solve some puzzles. The children were then given some feedback on their performance, and after that the children were asked to rate how lucky they had been at solving the puzzles, on a scale from 1 ("very lucky") to 10 ("very unlucky"). The scores for the 60 children in the study are in `http://www.utsc.utoronto.ca/~butler/assgt_data/feellucky.csv`, as a `.csv` file.
   (a) (2 marks) Read in and display (some of) the data. Does it look as if you have the right thing? Explain briefly.

   Solution: This is the easy version of reading in from a file:

```
my_url <- "http://www.utsc.utoronto.ca/~butler/assgt_data/feellucky.csv"
lucky <- read_csv(my_url)
```

```
## Parsed with column specification:
## cols(
##   luck = col_double()
## )
```

```
lucky
```

```
## # A tibble: 60 x 1
##     luck
##    <dbl>
##  1     1
##  2    10
##  3     1
##  4    10
##  5     1
##  6     1
##  7    10
##  8     5
##  9     1
## 10     1
## # ... with 50 more rows
```

There is a column called `luck`, that looks like the children's ratings of their own luck, between 1 and 10. There are 60 rows, which matches how many children there were in the study.

Extra: many of the scores are not just between 1 and 10, but actually *are* 1 and 10, which ought to be making you suspicious if you are thinking about normal distributions.

(b) (2 marks) Obtain a 99% confidence interval for the mean luck score.

Solution:

99% confidence is not the default, so you'll need to figure out how to change the confidence level from 95%. It goes like this:

```
with(lucky, t.test(luck, conf.level=0.99))
```

```
##
##  One Sample t-test
##
## data:  luck
## t = 12.116, df = 59, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 99 percent confidence interval:
##  4.603861 7.196139
## sample estimates:
## mean of x
##       5.9
```

4.6 to 7.2 points, *which you need to say.* You should also round off to some sensible number of decimal places; with a not-enormous data set, one more decimal place than the scale of the

data is sensible; here, the children recorded their level of luck as whole numbers (look back at the data), so one decimal place is OK.

(c) (3 marks) The middle of the luck scale is 5.5 points (halfway between 1 and 10). Is there any evidence that all children, if they were to do this study, would rate themselves as luckier than average? What do you conclude, in the context of the data?

Solution: This is a hypothesis test, testing the null hypothesis that the population mean is 5.5, against the alternative that the population mean is less than 5.5. (Think about what you are trying to prove; you are trying to demonstrate that children rate themselves as having better than average luck in this study, and 1 is the "lucky" end of the scale.) Thus:[1]

```
with(lucky, t.test(luck, mu=5.5, alternative="less"))
```

```
##
##  One Sample t-test
##
## data:  luck
## t = 0.82144, df = 59, p-value = 0.7926
## alternative hypothesis: true mean is less than 5.5
## 95 percent confidence interval:
##      -Inf 6.713736
## sample estimates:
## mean of x
##       5.9
```

The P-value, 0.7926, is not less than 0.05, so there is no evidence against the null hypothesis, and we cannot reject a population mean of 5.5: that is, children, if they were to do what was done in this study, do not rate themselves as having better than average luck. (You have two choices: reject the null, or fail to reject the null. This one is the latter.)

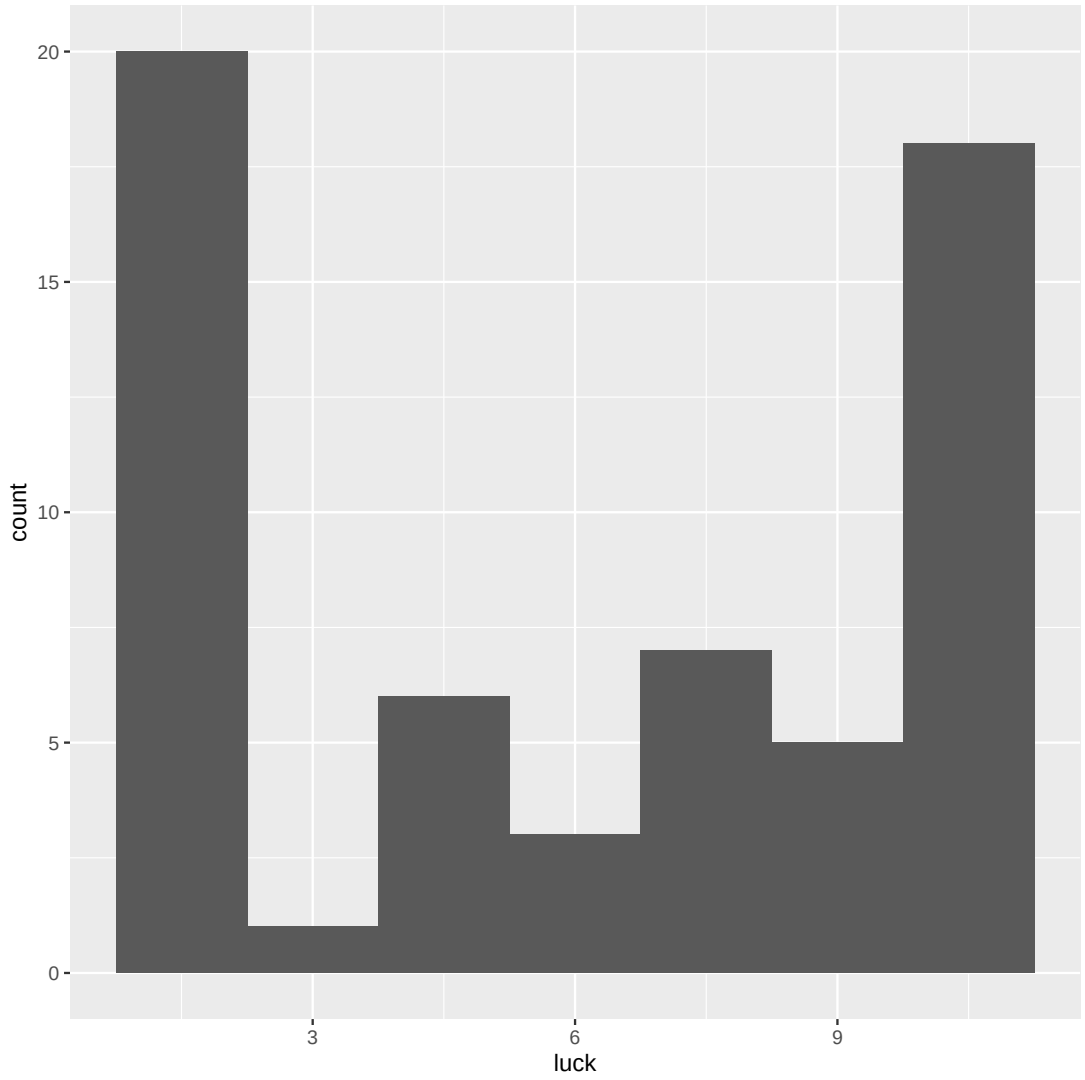
Extra: the sample mean, 5.9, is actually a little bit bigger than 5.5, so we really shouldn't be thinking about rejecting 5.5 in favour of less at all. In this kind of situation, what you can do is to note that you are on the wrong side, say that the P-value is "large", and fail to reject (without even doing any calculations).

As I meant to write the question, with 10 being most lucky and 1 being least lucky, then you have an actual question to answer. With an alternative of "greater", the P-value is 0.2074. What we observed here is the sort of thing that could (easily) have happened by chance if the population mean really were 5.5 (if children rated themselves as having average luck). If you want to prove that the population mean is bigger than 5.5, you need a sample mean that is further above 5.5 than 5.9 is (something like 7 might do it).

(d) (3 marks) Make a suitable graph of the data. Explain briefly why you are doubtful about the test and confidence interval you just did.

Solution: One quantitative variable suggests a histogram. With 60 observations something like 7 bins is OK:

```
ggplot(lucky, aes(x=luck)) + geom_histogram(bins=7)
```

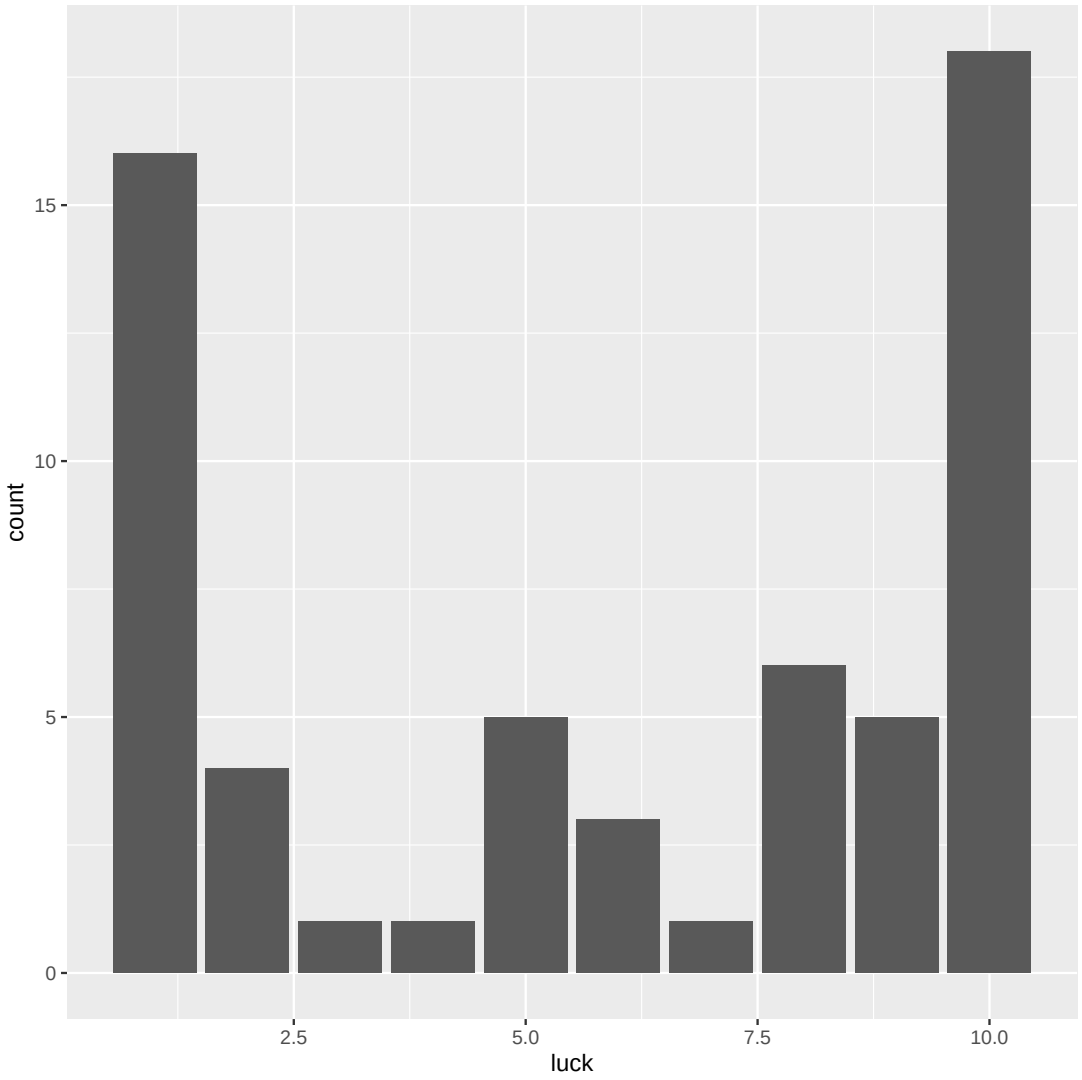


Anything up to about 10 bins is OK, but you *must* specify a number of bins.

This is not anything like "approximately normal", which is what you want. I would call this "bimodal" (two peaks, one each end); you could also reasonably say that it has a lot of outliers. Thus we should not trust either the test or the confidence interval. ("A lot of outliers" is a little odd, here, because the supposed outliers are actually most of the data, and outliers are really a few unusual values, but it gets at the idea I want, so I'm OK with it here.)

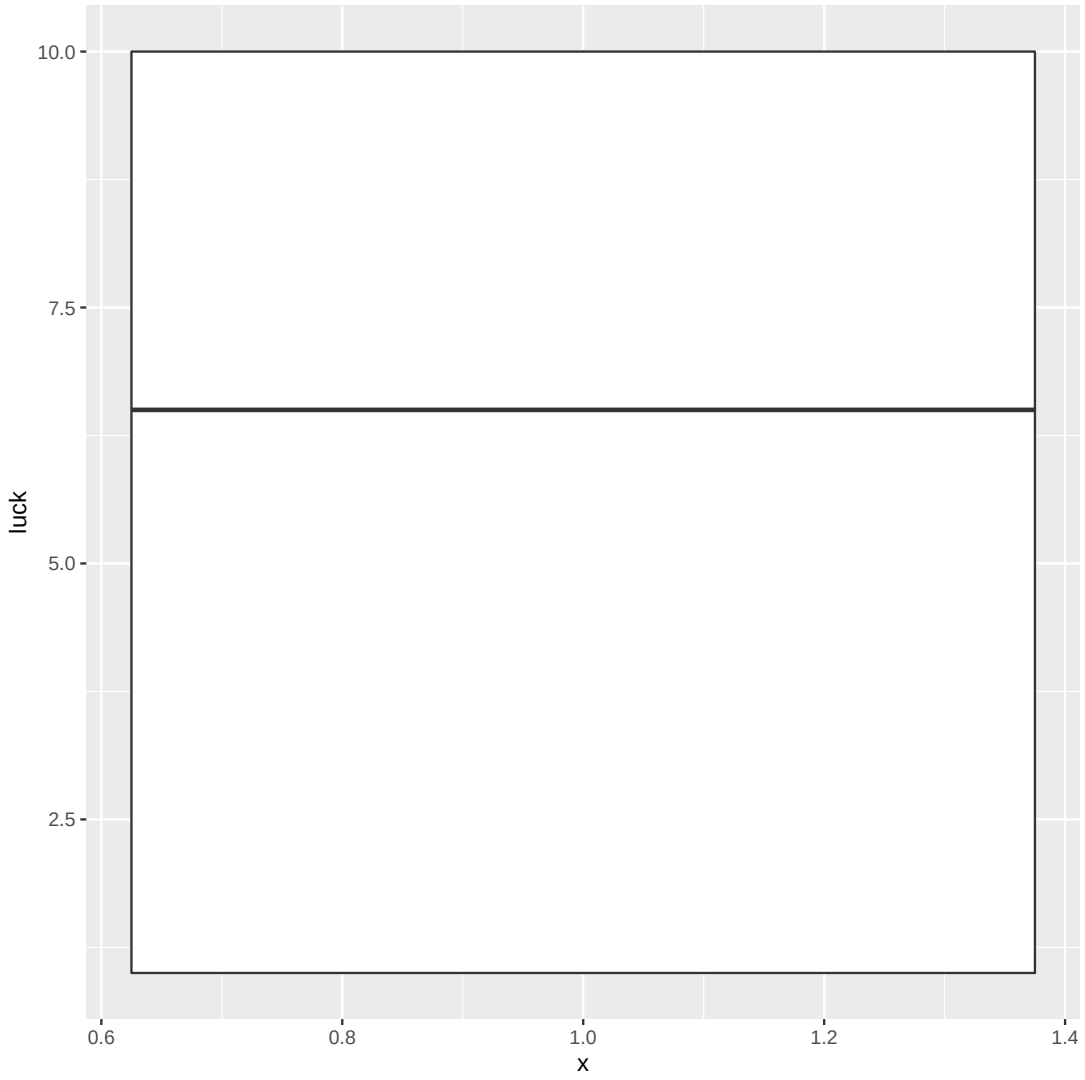
Another way is to note that the luck scores are all integers, and to treat them as categorical. This works here because there are not too many different scores; if there were decimals, say, this would not:

```
ggplot(lucky, aes(x=luck)) + geom_bar()
```



It really takes a histogram (or a bar chart) to show this shape; you could in principle also have drawn a one-group boxplot, thus:

```
ggplot(lucky, aes(x=1, y=luck)) + geom_boxplot()
```



but that just makes it look as if there is a lot of variability, without showing the shape. So I think you need a histogram to get full marks on this one. (Boxplots are good at centre and spread and outliers, but not so good at shape generally, particularly for U-shaped distributions like this one.)

Extra: a boxplot doesn't help because there are so many 1's and 10's that these are actually the quartiles (so the boxplot has no whiskers, never mind any outliers).

Extra extra: later in the course, we learn about the sign test, which tests the *median* rather than the mean. If you think (as you possibly do) that the median is a better measure of centre than the mean is for these data, then the sign test would be the way to go. I'll show you how it works, to look forward to later:

```
library(smmr)
sign_test(lucky, luck, 5.5)
```

```
## $above_below
## below above
##    27    33
##
## $p_values
##   alternative  p_value
## 1       lower 0.816853
## 2       upper 0.259479
## 3   two-sided 0.518958
```

The inputs to the `sign_test` function are, in order, the data frame, the column you want to test, and the null median. (The sidedness of the test comes in the output.)

The P-value we want is the one labelled `upper` (since we are testing for *greater* than average luck), 0.2595. This is actually quite similar to the P-value for the t-test. The table above the one with the P-values in it says that 33 of the 60 luck scores were greater than 5.5 and 27 were less. This is pretty close to a 50–50 split, which further supports the idea that the luck scores are not systematically bigger than 5.5; they are about evenly split above and below.

We learn later about getting a confidence interval for the median, which goes like this. I'm doing a 99% interval to be consistent with the one we did before for the mean:

```
ci_median(lucky, luck, conf.level=0.99)
```

```
## [1] 2.005859 8.996094
```

This is a pretty horrific interval! It is much longer than the one for the mean, because it is properly responding to the shape of the distribution, with a lot of high scores and a lot of low ones. (The fact that the confidence intervals for mean and median are so different is a warning that we ought to be careful about which one we use.)

Extra-cubed: I cheated by arranging the question this way. The *right* way to do this kind of work is to draw the graph *first*, and then use the graph to inform your decision about what test to do. Had I done that here, though, you would (rightly) have said that it would make no sense to do anything t-like, and I wanted to have you practice `t.test`. So I asked you to do that first.

3. Work through problems 7.1 through 7.4 in Chapter 7 of PASIAS.

4. What is the effect of sleep deprivation on food intake? To find out, a random sample of 30 men were randomly assigned to one of two groups. In the first group, the men were limited to only 4 hours of sleep on each of two nights; in the second group, the men were allowed to sleep for 8 hours for the same two nights. The day after their two nights of restricted sleep, each man had their food intake (in Kcal) measured.

   The data are in `https://www.utsc.utoronto.ca/~butler/assgt_data/sleep-dep.csv`.

   (a) (3 marks) Read in and display (some of) the data. Do you have what you were expecting to see? Explain briefly.

   Solution: This is (as you can check) a `.csv`, so the usual thing. You can choose a sleep-related name or a food-related name, as you wish. (Note that two of the columns are called `food` and

`sleep`, so avoid those names if you can.):

```
my_url="https://www.utsc.utoronto.ca/~butler/assgt_data/sleep-dep.csv"
sleep_dep=read_csv(my_url)
```

```
## Parsed with column specification:
## cols(
##  subject = col_double(),
##  sleep = col_character(),
##  food = col_double()
## )
```

```
sleep_dep
```

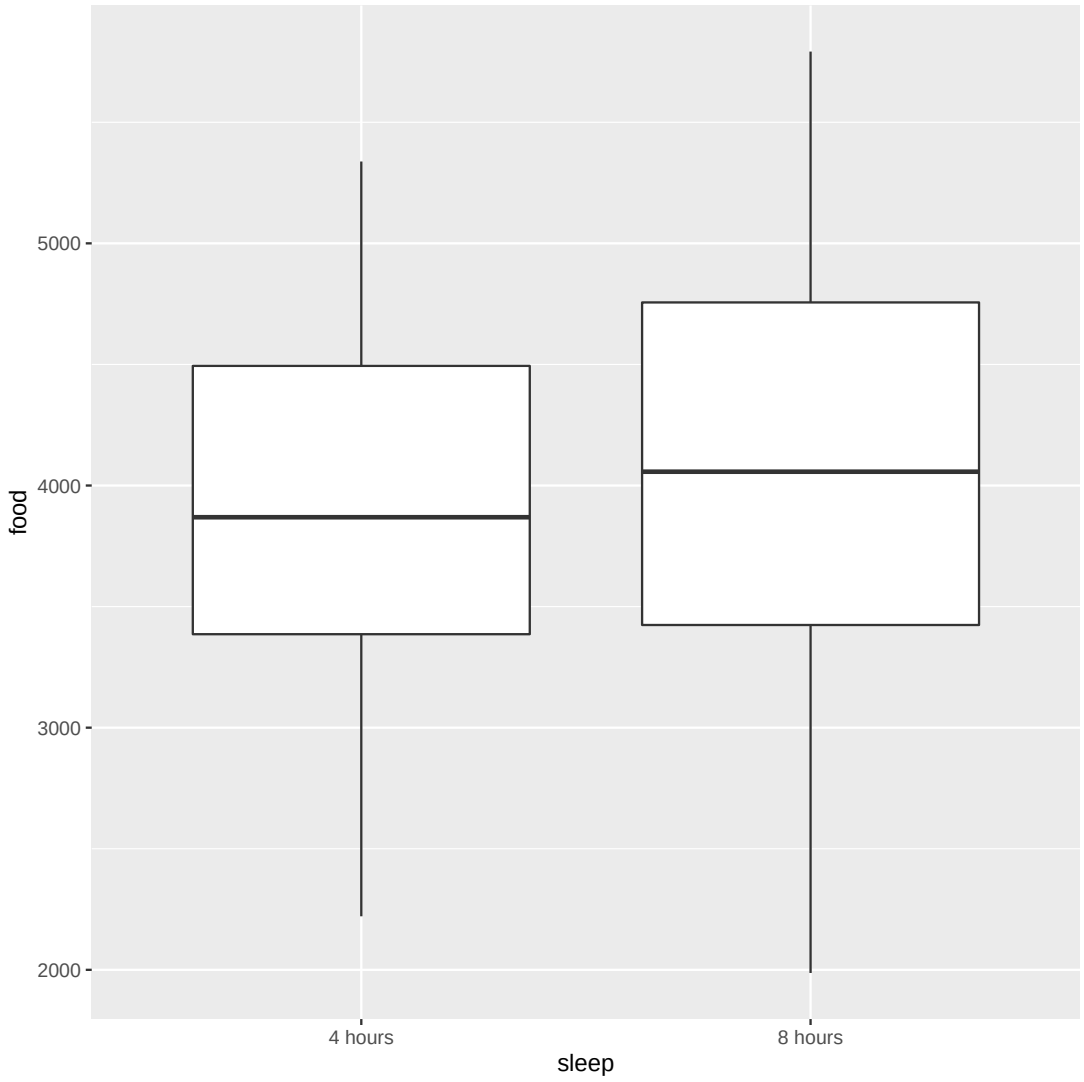
```
## # A tibble: 30 x 3
##    subject sleep    food
##      <dbl> <chr>   <dbl>
##  1       1 4 hours  3585
##  2       2 8 hours  3319
##  3       3 4 hours  3068
##  4       4 4 hours  3187
##  5       5 4 hours  5338
##  6       6 8 hours  4965
##  7       7 4 hours  3869
##  8       8 8 hours  4100
##  9       9 8 hours  3653
## 10      10 4 hours  3099
## # ... with 20 more rows
```

I have a column containing subject numbers, a column of which sleep group each man was in, and a column of their food intake. This is what I was expecting to see.

(b) (4 marks) Make a suitable graph (noting that the subject numbers are an identification variable only and don't need to appear on the graph). Compare the centres and spreads of the two distributions, and note any interesting features.

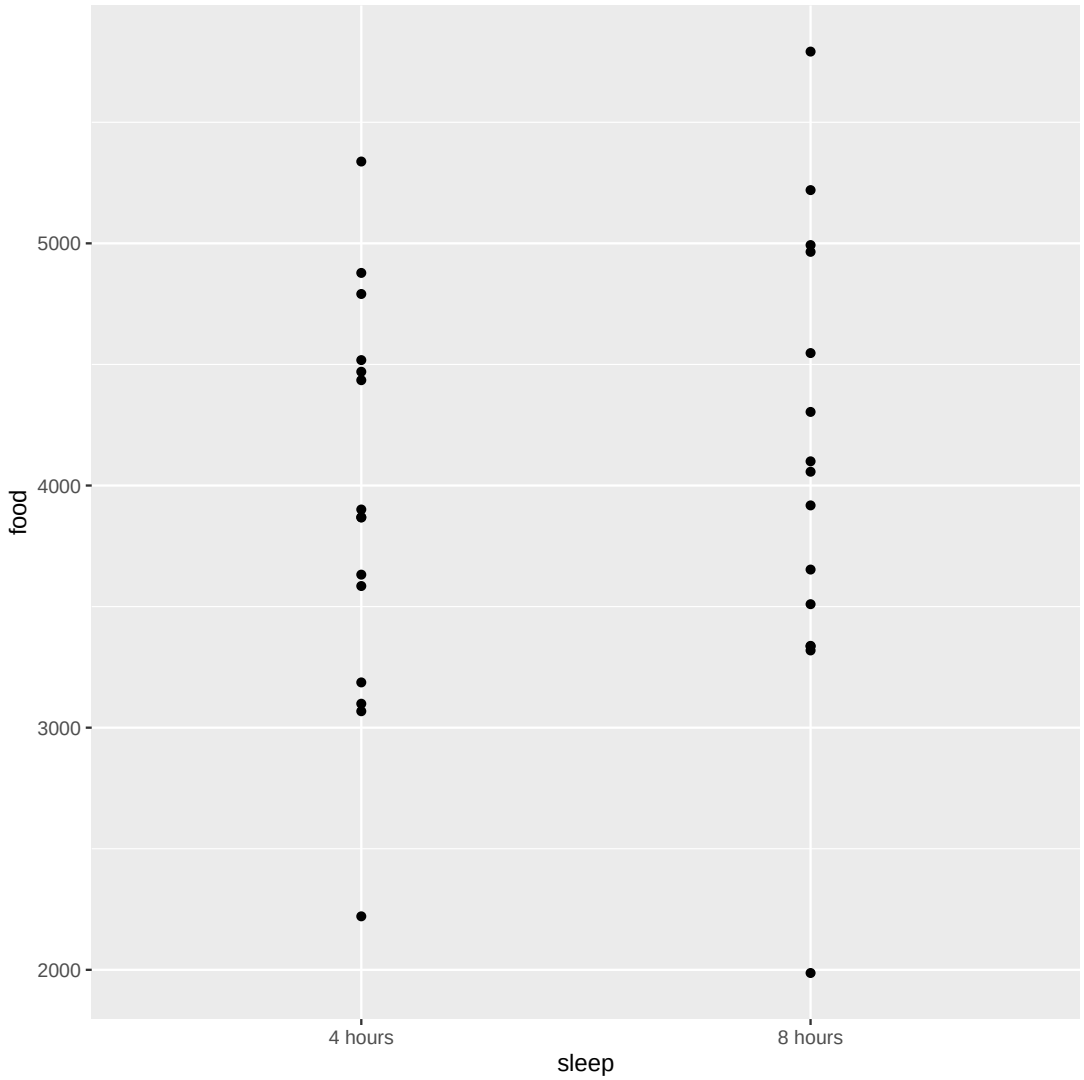
Solution: Two remaining variables, one of which is categorical (hours of sleep) and one quantitative (food intake). This suggests boxplots:

```
ggplot(sleep_dep, aes(x=sleep, y=food)) + geom_boxplot()
```



You might argue that the hours of sleep is also quantitative, and therefore we ought to draw a scatterplot:

```
ggplot(sleep_dep, aes(x=sleep, y=food)) + geom_point()
```



This is actually not quite right because the x-axis on this plot is groups rather than numbers. At least, it's not a scatterplot; it's rather more what you might call a dotplot. The issue is that the number of hours of sleep is indeed a number, but it's being *treated as a category*: we are comparing the two groups defined by the numbers of hours of sleep. By making the hours of sleep get read in as text, I was hoping to guide you towards treating it as categorical.

I guess I can live with the dotplot-like thing as an answer to "suitable graph" (the first two points), but it makes it more difficult for you to get the last two points.

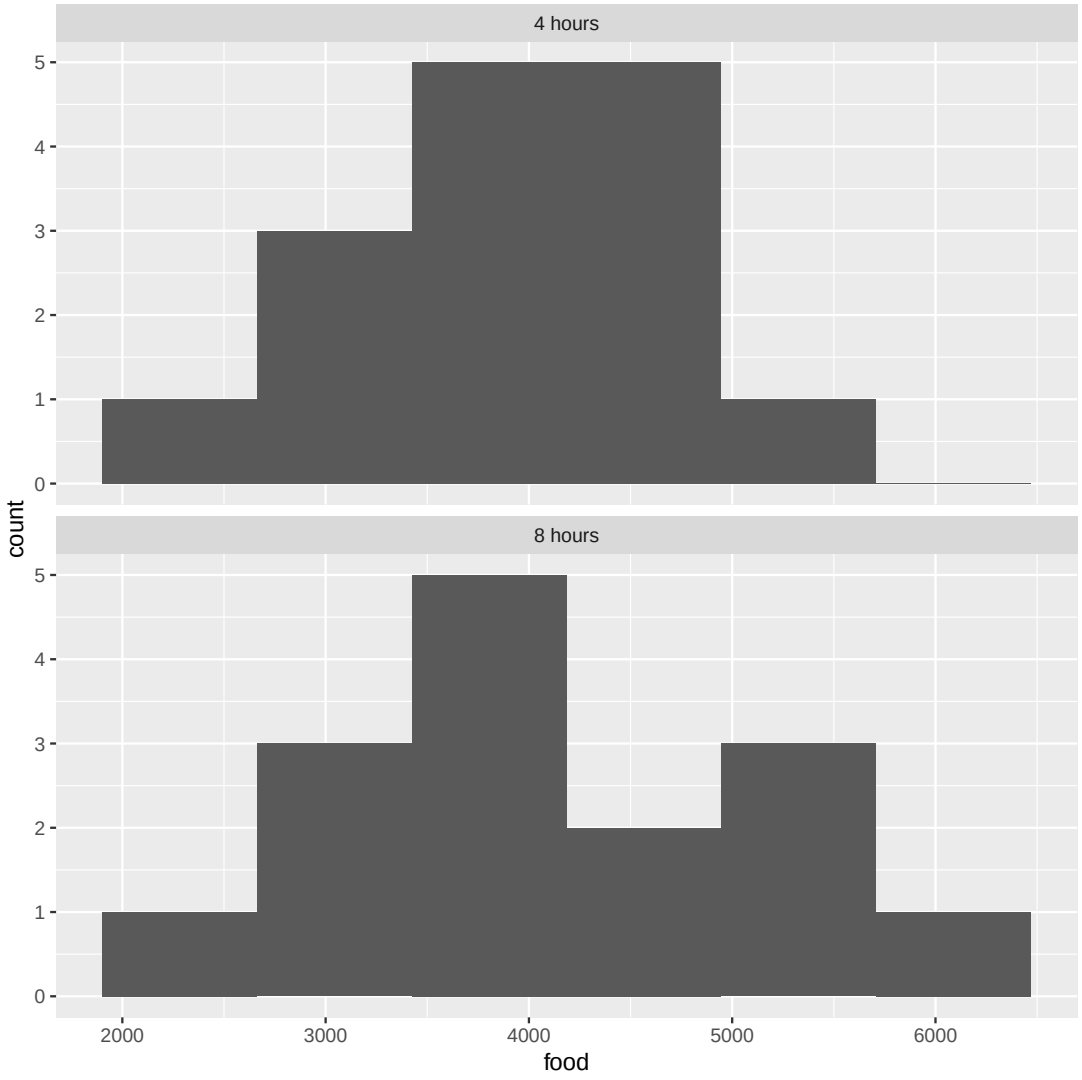
For those, go back to the boxplot: the median of the 8-hours group is slightly higher, and the spread of the 8-hour group is slightly bigger. As for interesting features, this is an invitation to talk about shape and/or outliers; there are no outliers, and the whiskers of both are about the same length above and below, so both groups are more or less symmetric.

It's much harder to make conclusions like these from the dotplot-like thing: you will have to make a guess at the centre of each group, and you will need to try to assess spread without getting taken in by the extreme observations too much. On the dotplot, it looks as if one or

maybe both of the lowest observations in each group is an outlier, also.

Extra: another possible plot here is a facetted histogram. Since our aim is to compare, it's easiest to arrange things so the plots are above and below:

```
ggplot(sleep_dep, aes(x=food)) + geom_histogram(bins=6) +
    facet_wrap(~sleep, ncol=1)
```



The reason for that `ncol=1` is to get the two facets arranged in one column (that is, above and below rather than left and right as they will come out otherwise).

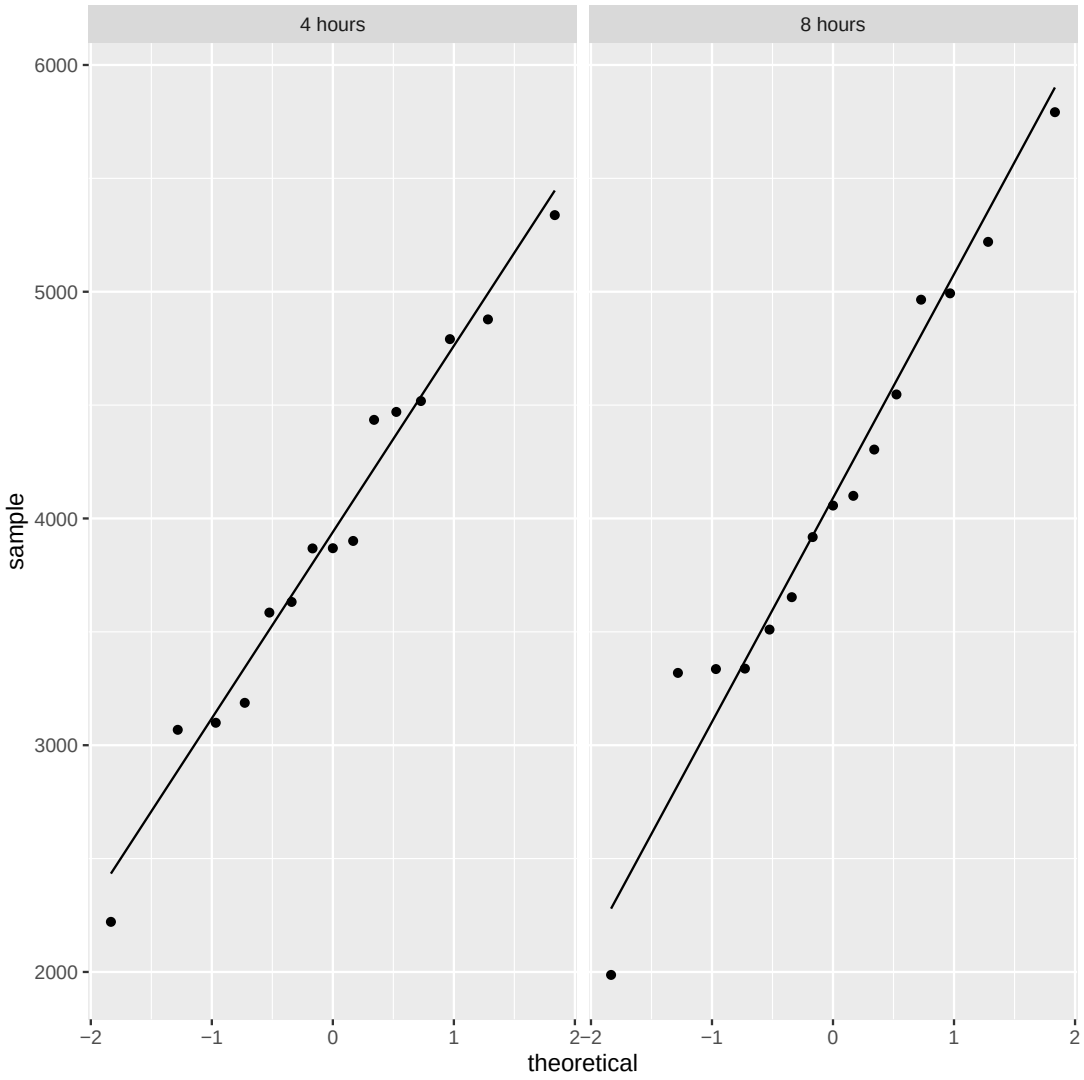
Comparing two histograms, even above and below, is harder than comparing boxplots because you have to *judge* the centres and spreads, rather than having medians and box heights as on a boxplot.

I would say that on these histograms, the centres are about the same, but the 8-hour plot looks more spread out. On these, both shapes look more or less symmetric and there are no outliers. But this is hard work compared to the boxplots, which are *designed* for assessing centre and

spread and skewness and outliers; that was John Tukey's[2] purpose in coming up with the idea back in the 1950s.

Looking ahead, yet another possibility is facetted normal quantile plots, to assess each group directly for normality:

```
ggplot(sleep_dep, aes(sample=food)) + stat_qq() +
    stat_qq_line() + facet_wrap(~sleep)
```



I think it's OK to have these side by side, since we are mainly comparing each one separately with its line.

Both groups look acceptably normal to me, with the points being close to their lines without any real outliers. It's difficult to assess centre and spread on these; the lines are in similar places on the two graphs, which says the centres are similar (mean or median), and the graphs have about the same slope, which suggests that the spreads are also similar.[3]

An odd remark here: I said that the lowest observation in the 8-hour group looked like an

outlier. But when you look at the (right-hand) normal quantile plot, it's actually about where you'd expect the lowest observation in a normal sample of this size to be (only a little too low). What is happening is that the *second*-lowest observation is too high compared to a normal, which means that there appears to be a big gap between the lowest and second-lowest observations in this group. This is what made the lowest observation look like an outlier on the dotplot: not that it was too low, but its neighbour was too *high*.

(c) (3 marks) Carry out a suitable two-sample t-test. What do you conclude, in the context of the data? (Make sure you justify the particular two-sample test that you do.)

Solution: There is nothing in the question about trying to prove that food intake in a particular group is higher than in the other group, so this is a two-sided test. We are looking for any difference.

Your choices are the Welch and the pooled two-sample tests. To decide between them, look back at the plot you drew and decide whether your spreads are similar or not. If you think they are sufficiently close, go with the pooled test:

```
t.test(food~sleep, data=sleep_dep, var.equal=T)
```

```
##
##  Two Sample t-test
##
## data:  food by sleep
## t = -0.44529, df = 28, p-value = 0.6595
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -813.5121  522.9788
## sample estimates:
## mean in group 4 hours mean in group 8 hours
##              3924.000              4069.267
```

and if you think they are different (or you're not sure and you want to play it safe), go with the Welch test, which is the default:

```
t.test(food~sleep, data=sleep_dep)
```

```
##
##  Welch Two Sample t-test
##
## data:  food by sleep
## t = -0.44529, df = 27.48, p-value = 0.6596
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -814.0826  523.5493
## sample estimates:
## mean in group 4 hours mean in group 8 hours
##              3924.000              4069.267
```

You need to have a reason for doing the test you do. I don't care which test you end up doing, as long as it's consistent with the reason you gave.

Whichever way you go, the P-value is much bigger than 0.05, so there is no way to reject the null hypothesis that the mean food intake is the same, whether a man has 4 or 8 hours of sleep.

Or you can be a bit longer about it: the null hypothesis is that the mean food intake is the same regardless f the amount of sleep, and the alternative hypothesis is that the mean is different between the two amounts of sleep. The P-value is large, and therefore the null hypothesis is not rejected. We therefore conclude that there is no evidence of a difference in food intake between men who sleep 4 hours and men who sleep 8.

The second way is longer, but also makes it absolutely clear that you know what you're doing. (Remember that we are trying to generalize from the 30 men in the sample to "all men", so your conclusion refers to all men: no evidence that the mean food intake for all men who sleep 4 hours and all men who sleep 8 is any different.)

Extra: I am not a fan of doing a formal test for equality of spreads to decide which t-test to do. The research I've seen suggests that a two-step process using something like an F-test or Levene's test to compare the variances of the two groups first, and then using its result to decide which t-test to do, can end up having an actual α that is not 0.05 even if that was what you were aiming for. The rationale is that with small samples, it is hardest to prove that the variances are different (they have to be *very* different), but this is also the case where it matters the most that you do the right t-test. With large samples, a variance test will reject equal variances even if they are almost the same, and so you will end up doing Welch when pooled would have been fine.

Hence, I prefer looking at pictures, and making an informal assessment.

The other thing to keep in mind is that Welch is pretty good if the variances are equal (the pooled test is the absolute best in those circumstances, as you proved if you took STAB57, but the Welch test is pretty good). So always doing the Welch test, as a mindless application of `t.test` would do, is not at all bad. But I need you to have a *reason* for doing whatever it is that you do.