University of Toronto Scarborough
Department of Computer and Mathematical Sciences
STAC33 (K. Butler), Midterm Exam
March 2, 2024

Aids allowed (on paper, no computers):

- My lecture overheads (slides)
- Any notes that you have taken in this course
- Your marked assignments
- My assignment solutions
- Non-programmable, non-communicating calculator

This exam has xx numbered pages of questions plus this cover page.
In addition, you have an additional booklet of Figures to refer to during the exam.
The maximum marks available for each part of each question are shown next to the question part.
If you need more space, use the last page of the exam. Anything written on the back of the page will not be graded.
You may assume throughout this exam that the code shown in Figure 1 of the booklet of Figures has already been run.

topics

x- reading a file: performance

x- making a graph: performance

x- numerical summaries performance, sleepstudy

x- choosing things: sleepstudy

x- one sample t: entrance

x - two sample t: bank

x power

x - sign test: bonding

x normal quantile

1. 80 subjects carried out a two-part task. The first part, `What`, could be visual (identify letters) or verbal (identify sentences). The second part, `Report`, was to report the results, which could be visual (pointing at a response) or verbal (speaking a response), independently of the first part. The total `Time` needed to complete the two parts of the subject's task was recorded, in seconds.

   Give the code needed to answer the questions below.

   (a) [3] Some of the data file is shown in Figure 2, stored in a file `perfs.txt` in the same folder as your current R Studio project. Read the data into a dataframe called `performance` and display at least some of that dataframe.

   My answer:

   See that the data values are separated by single semicolons, so this is `read_delim`:

```
performance <- read_delim("perfs.txt", ";")
```

```
Rows: 80 Columns: 3
-- Column specification --------------------------------------------------------
Delimiter: ";"
chr (2): What, Report
dbl (1): Time

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

   and then

```
performance
```

Time	What	Report
28.15	Visual	Visual
10.91	Verbal	Verbal

15.85	Visual	Visual
10.90	Verbal	Verbal
6.91	Visual	Verbal
11.37	Verbal	Verbal
12.64	Visual	Verbal
9.18	Visual	Verbal
8.44	Verbal	Visual
8.33	Verbal	Verbal

Two points for the `read_delim`, one for displaying what you read in. Minus one per error. Any way of displaying it is fine, for example

```
glimpse(performance)
```

```
Rows: 80
Columns: 3
$ Time   <dbl> 28.15, 10.91, 15.85, 10.90, 6.91, 11.37, 12.64, 9.18, 8.44, 8.3~
$ What   <chr> "Visual", "Verbal", "Visual", "Verbal", "Visual", "Verbal", "Vi~
$ Report <chr> "Visual", "Verbal", "Visual", "Verbal", "Verbal", "Verbal", "Ve~
```

as long as it will display some of it so that you would be able to check that the values are reasonable.

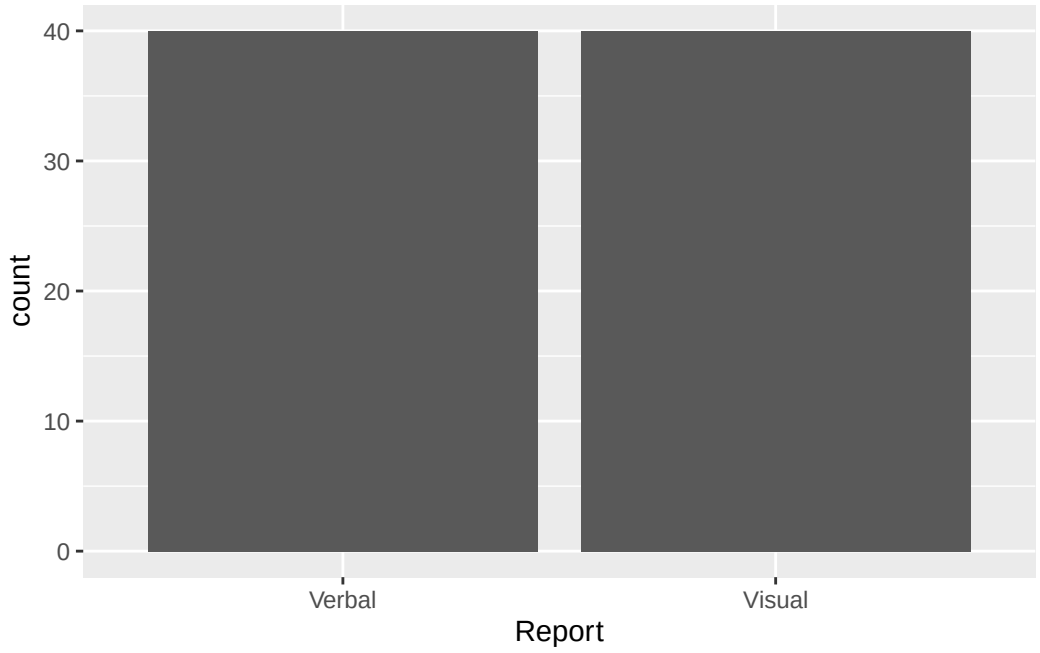
Make sure that your underscore looks like an underscore. I did not teach `read.delim` in this course, or `head`, so don't expect to get credit for those.

(b) [2] Draw a suitable graph of the variable `Report`.

My answer:

This is categorical, so a bar chart:

```
ggplot(performance, aes(x = Report)) + geom_bar()
```



This is actually a rather pointless graph, because all it says is that 40 of the subjects were asked to give each type of report (which is how the study was designed). But you were not to know that.

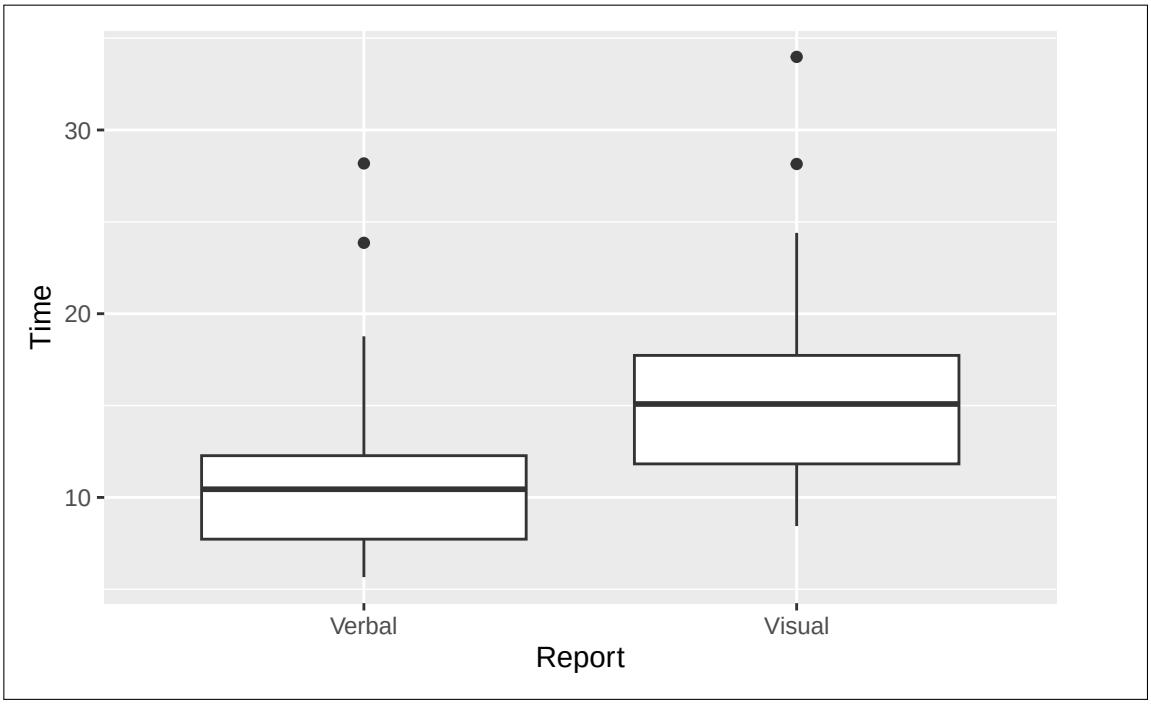
For this part and the rest of the question, minus a point for each error, but one point if something relevant is correct (in the grader's estimation) regardless of the number of errors. The grader is free to decide that there is a lot to get right and to deduct only a half point per error, but that decision is for the grader to make (possibly in consultation with me).

(c) [3] Draw a suitable graph of the variables `Report` and `Time`.

My answer:

One categorical and one quantitative, so a boxplot:

```
ggplot(performance, aes(x = Report, y = Time)) + geom_boxplot()
```

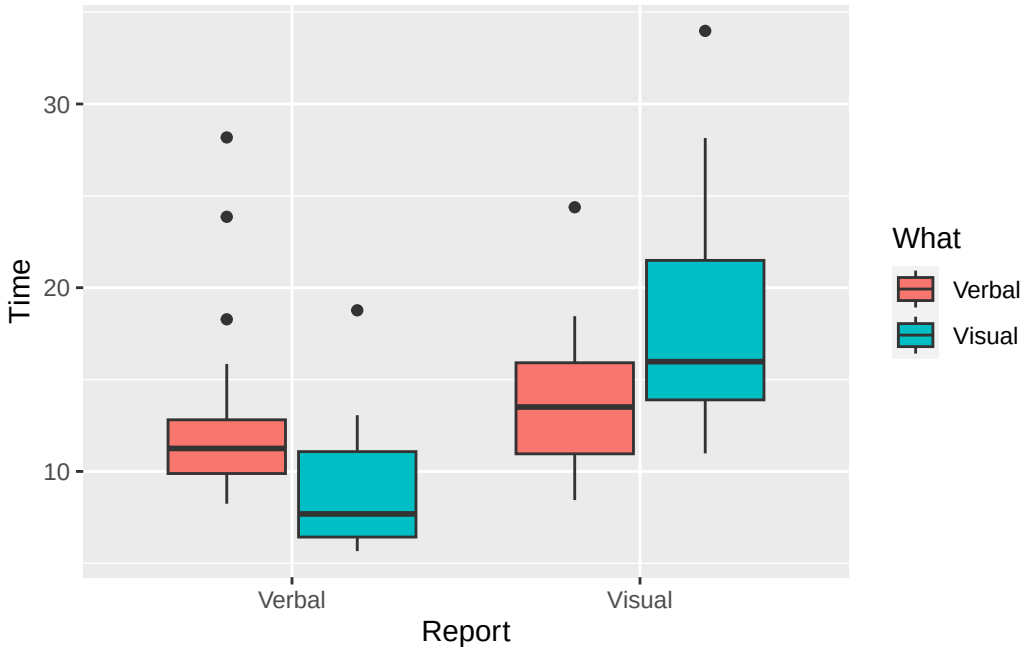


(d) [3] Draw a suitable graph of all three variables. (You have some choices here. It is up to you to make what you think is a sensible choice.)

My answer:

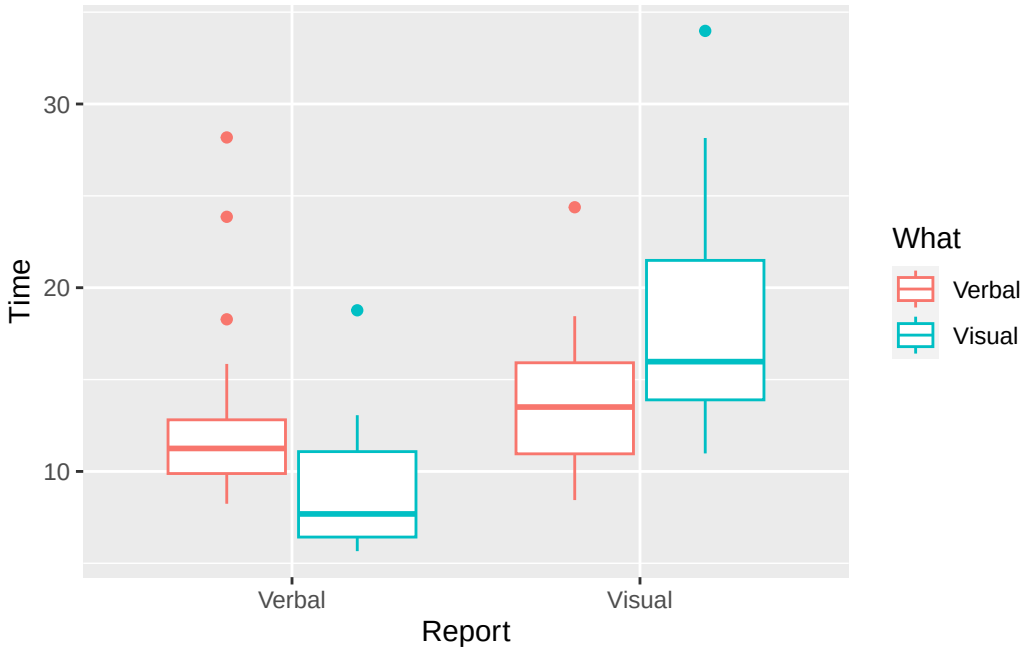
The standard graph with two categorical variables and one quantitative one is a grouped boxplot, something like this:

```
ggplot(performance, aes(x = Report, fill = What, y = Time)) + geom_boxplot()
```



You can switch around `Report` and `What` (I have no particular preference for which is on the x-axis and which is colour). You can also use `colour` rather than `fill`:

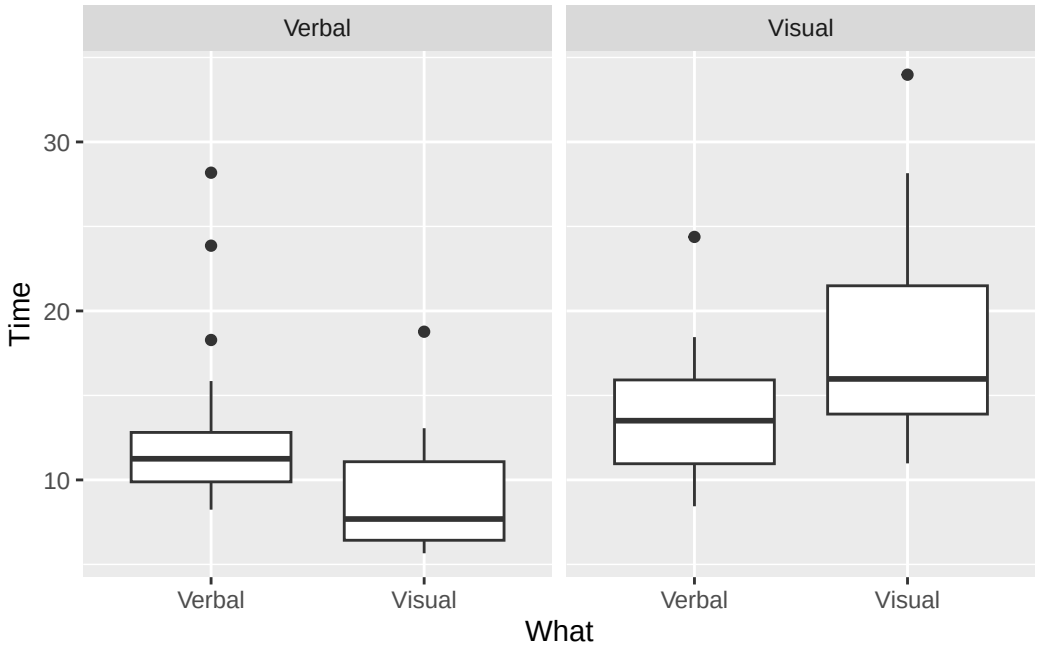
```
ggplot(performance, aes(x = Report, colour = What, y = Time)) + geom_boxplot()
```



This colours the outsides of the boxes rather than the inside, but is also a perfectly acceptable graph.

Another approach you can take is to treat one of the categorical variables as "extra" and use it for facets, with each facet containing an ordinary (not grouped) boxplot, like this:

```
ggplot(performance, aes(x = What, y = Time)) + geom_boxplot() +
  facet_wrap(~ Report)
```

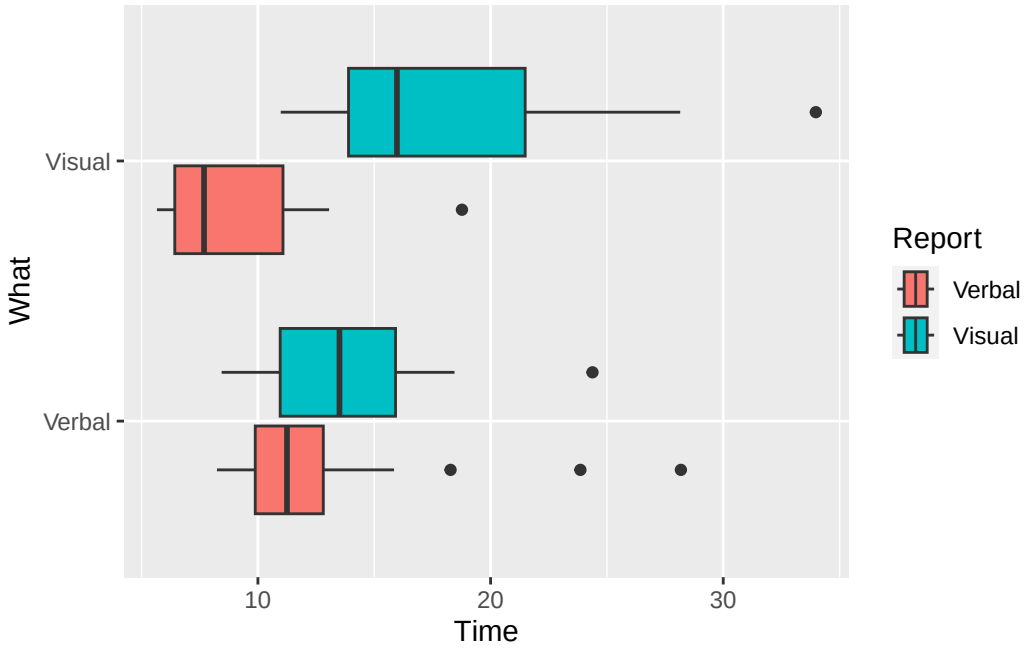


or with `What` and `Report` the other way around (equally good). In summary, one of these:

- a grouped boxplot with `Time` on the y-axis
- facetted boxplots with one of the categorical variables on the x-axis and the other making facets.

This actually does also work:

```
ggplot(performance, aes(y = What, x = Time, fill = Report)) + geom_boxplot()
```



with the boxes going sideways, but is a rather fast way to confuse the grader and risk losing marks that you might have earned. Writing down something that obviously works is a better plan.

Strategy for the grader: go through all the grouped boxplots first, leaving the facetted boxplots aside, then go through the facetted boxplots, then go through anything else. Key on `y = Time` since that is common to all the answers.

(e) [3] Find the number of observations, the mean task completion time, and the standard deviation of task completion time, for the subjects that were asked to make each kind of `Report`.

My answer:

This is a standard group-by and summarize, thus:

```
performance %>%
  group_by(Report) %>%
  summarize(n = n(), mean_time = mean(Time), sd_time = sd(Time))
```

Report	n	mean_time	sd_time
Verbal	40	10.93075	4.723802
Visual	40	15.93075	5.580873

Give the summaries whatever names you like, though it is better if they say something about what they are.

2. A sample of 253 students did skills tests to measure cognitive function, completed a survey that asked many questions about attitudes and habits, and kept a sleep diary to record time and quality of sleep over a two week period. There were many variables measured; some of them, for some of the students, are shown in Figure 3. The dataframe is called `students`. The relevant columns to us are:

   - `AnxietyScore`: Measure of amount of anxiety (a number)
   - `AnxietyStatus`: The amount of anxiety as a category, normal, moderate, or severe
   - `ClassesMissed`: Number of classes missed in a semester
   - `ClassYear` Year in school, 1=first year, up to 4=fourth year
   - `GPA`: Grade point average (0–4 scale)
   - `LarkOwl`: Early riser (gets up early), night owl (goes to bed late), not either of those? Respectively: Lark, Owl, or Neither
   - `WeekdaySleep`: Average hours of sleep on weekdays

   For each of the questions below, give code to obtain the required results.

   (a) [3] Find the median and interquartile range of weekday sleep hours for the students who were lark or owl or neither.

   My answer:

   A group-by and summarize to warm you up. `LarkOwl` distinguishes the larks from the owls (from the students who are neither larks nor owls), and `WeekdaySleep` is the column of weekday sleep hours:

```
students %>% group_by(LarkOwl) %>%
  summarize(median_sleep = median(WeekdaySleep), iqr_sleep = IQR(WeekdaySleep))
```

LarkOwl	median_sleep	iqr_sleep
Lark	7.83	1.47
Neither	7.97	1.15
Owl	7.82	1.53

   Minus a half point per small error, minus a full point for a fundamental error (in the grader's estimation).

   (b) [2] Display all the columns whose names are something followed by the word "score" (which could actually be `score` or `Score`). There are several of these, whose names you do not know. You can display all the rows.

   My answer:

   This is `select` using the select-helper `ends_with`:

```
students %>% select(ends_with("score"))
```

CognitionZscore	DepressionScore	AnxietyScore	StressScore	DASScore
-0.26	4	3	8	15
1.39	1	0	3	4
0.38	18	18	9	45
1.39	1	4	6	11
1.22	7	25	14	46
-0.04	14	8	28	50
0.41	1	0	1	2
-0.59	2	2	3	7
1.03	12	16	20	48
0.72	6	11	31	48

The select-helpers are not case-sensitive, so this will get both `score` and `Score` on the end of the name.

This one you are likely to get or not. Minus a half point for an unnecessary `ignore.case = TRUE` (that is the default).

This, if you can get it right, is also two points:

```
students %>% select(matches("score$"))
```

CognitionZscore	DepressionScore	AnxietyScore	StressScore	DASScore
-0.26	4	3	8	15
1.39	1	0	3	4
0.38	18	18	9	45
1.39	1	4	6	11
1.22	7	25	14	46
-0.04	14	8	28	50
0.41	1	0	1	2
-0.59	2	2	3	7
1.03	12	16	20	48
0.72	6	11	31	48

`matches` requires a regular expression. The `$` is needed to match the end of the column name. This is also not case-sensitive, so will match an uppercase S too. (Outside of R's select-helpers, this regular expression will only match if the S is lowercase.)

(c) [3] For the students that are Owls, how many are there of them, and what is their mean grade point average? Do not display anything for the students who are `Lark` or `Neither`.

My answer:

Use `filter` to grab only the Owls first:

```
students %>%
  filter(LarkOwl == "Owl") %>%
  summarize(n = n(), mean_gpa = mean(GPA))
```

n	mean_gpa
49	3.196735

I guess you can also group-by and summarize, and then use a `filter` to grab only the Owls:

```
students %>%
  group_by(LarkOwl) %>%
  summarize(n = n(), mean_gpa = mean(GPA)) %>%
  filter(LarkOwl == "Owl")
```

LarkOwl	n	mean_gpa
Owl	49	3.196735

You have done some unnecessary calculation then, but I guess I will accept it here. (The purpose of this part was to show that you know about `filter`; thus, only 1.5 points if you don't do a relevant `filter` somewhere.)

(d) [2] Display the students that are either Owls or have an anxiety score above 20, or both.

My answer:

The vertical bar is either-or:

```
students %>% filter(LarkOwl == "Owl" | AnxietyScore > 20)
```

LarkOwl	AnxietyScore
Owl	18
Owl	25
Owl	5
Owl	5
Owl	13
Owl	5
Owl	2
Neither	22
Owl	10
Neither	21
Owl	9

Owl	2
Owl	10
Owl	3
Owl	3
Owl	1
Owl	13
Owl	20
Owl	2
Owl	4
Owl	9
Owl	0
Owl	0
Owl	0
Owl	3
Neither	21
Owl	4
Owl	8
Owl	2
Owl	0
Owl	0
Owl	6
Owl	10
Owl	2
Owl	1
Owl	7
Owl	0
Owl	4
Owl	9
Owl	3
Owl	10
Owl	0
Owl	15
Owl	2
Owl	7
Owl	12
Owl	1
Owl	1
Owl	3
Owl	8
Lark	26
Owl	1
Owl	2

plus all the other columns. You can check from my output that the students that are not Owls are there because their anxiety score is over 20.

(e) [3] Display (only) the class year and GPA for the students that missed 15 or more classes.

My answer:

Do the `filter` first because `ClassesMissed` is not something you're going to display at the end:

```
students %>%
  filter(ClassesMissed >= 15) %>%
  select(ClassYear, GPA)
```

ClassYear	GPA
3	2.80
2	3.07
2	3.12
2	3.25

There are actually only four students who missed that many classes.

(f) [3] How many Owls are there that have at least three early classes?

My answer:

There was a column called `NumEarlyClass` that I forgot to tell you about, that had the number of early classes in it. Since I didn't tell you about it, you couldn't do this part or the next one.

```
students %>%
  filter(LarkOwl == "Owl") %>%
  count(NumEarlyClass >= 3)
```

NumEarlyClass >= 3	n
FALSE	39
TRUE	10

or, better:

```
students %>%
  filter(LarkOwl == "Owl", NumEarlyClass >= 3) %>%
  count()
```

n
10

so the answer is actually 10 (which of course you won't know). The second one is better because it only displays the answer you want, but you might not have used `count()` like this, so I'll accept either, or anything else that will work equivalently.

`count` counts the number of rows. You can either count something that is true or false (and look at the number next to `TRUE`), or you can do everything with `filter` and count how many rows there are in the result. Two `filter`s in a row followed by a `count` is another way to do the last one.

(g) [3] Find the highest eleven anxiety scores for students that have at least 3 early classes.

My answer:

Omitted also.

Use `slice_max` (a step shorter), or sort and then `slice`. You will actually get all the columns, which is fine, or you can `select AnxietyScore` and maybe `NumEarlyClass` as well, which is what I did (surreptitiously):

```
students %>%
  filter(NumEarlyClass >= 3) %>%
  slice_max(AnxietyScore, n = 11)
```

NumEarlyClass	AnxietyScore
5	14
4	13
4	13
3	13
5	13
4	13
5	13
5	12
5	12
5	12
3	11

or

```
students %>%
  filter(NumEarlyClass >= 3) %>%
  arrange(desc(AnxietyScore)) %>%
  slice(1:11)
```

NumEarlyClass	AnxietyScore
5	14
4	13
4	13
3	13
5	13
4	13
5	13
5	12
5	12
5	12
3	11

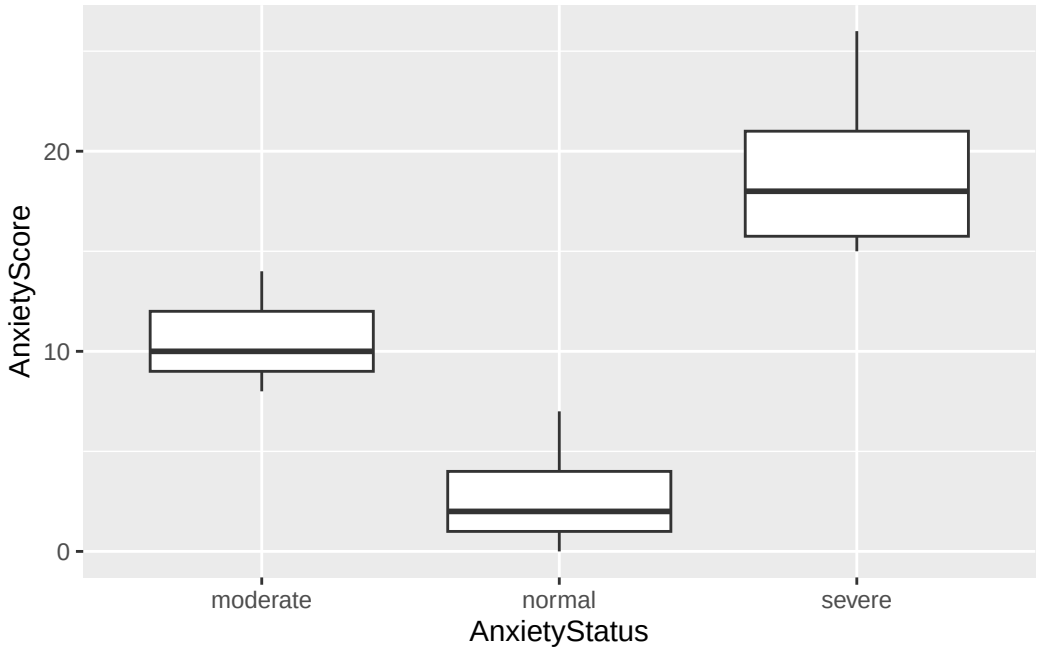
One point for a relevant `filter` and two for `slice_max` done correctly, or in place of the `slice_max`, one for sorting and one for slicing.

(h) [3] Anxiety in this survey was measured as a quantitative score (from a questionnaire), and then turned into a categorical "none", "moderate", or "severe" by converting the score into categories in some consistent way, with a higher anxiety score always becoming one of the later categories. How might you discover, using a graph or a calculation, how this conversion was done? Explain briefly (that is, your answer should have some words of explanation as well as code).

My answer:

This means looking for an association of some kind between `AnxietyScore` and `AnxietyStatus`. You might do this by making a boxplot:

```
ggplot(students, aes(x = AnxietyStatus, y = AnxietyScore)) + geom_boxplot()
```



or by finding the largest and smallest scores within each status category:

```
students %>% group_by(AnxietyStatus) %>%
  summarize(min_score = min(AnxietyScore), max_score = max(AnxietyScore))
```

AnxietyStatus	min_score	max_score
moderate	8	14
normal	0	7
severe	15	26

I have the results here, which you don't.

The table is perhaps clearer: a score up to 7 is normal, 8 through 14 inclusive is moderate, and 15 or above is severe. But you could also read these off from the boxplot.

A third strategy, not as good (as in "as easy to use") but it works, is to sort the anxiety scores, and then display them next to the anxiety status values. This enables you to eyeball where the boundaries are.

One point for saying what your strategy is going to be, and two points for making that strategy work in code. If your strategy won't solve the problem, then a maximum of one point if your code implements your strategy and is of equivalent complexity to this. The point is to find out the connection between anxiety score and anxiety status, so it doesn't help to guess what that connection is and risk being wrong.

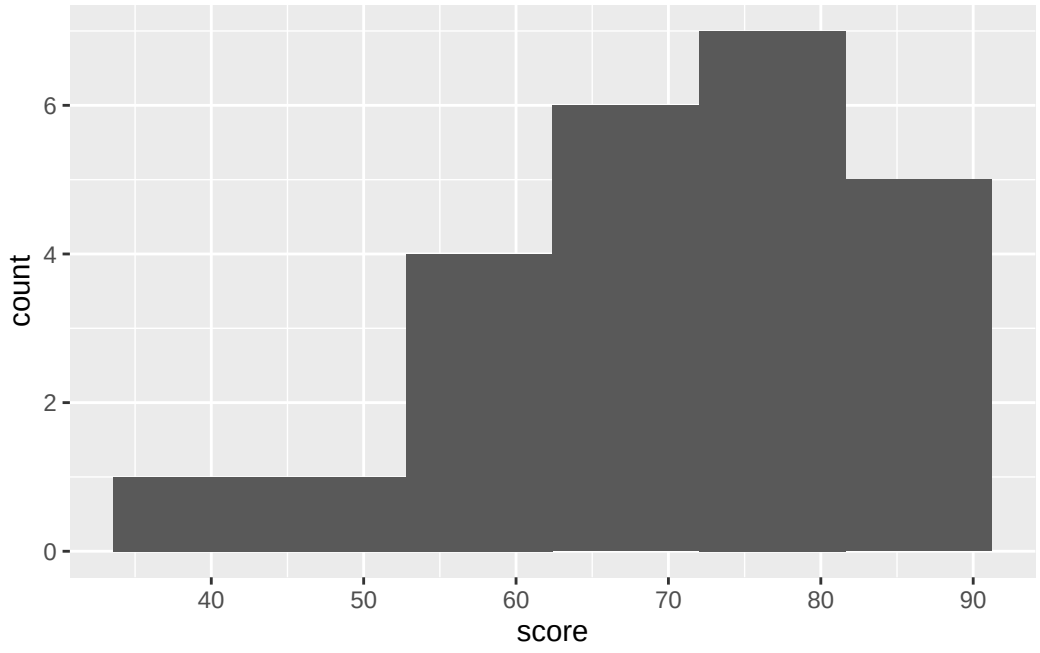
3. 24 high-school students wrote a college entrance exam this year. Some of the data, in dataframe `entrance`, are shown in Figure 4.

   (a) [2] A graph is shown in Figure 5. What code was used to draw this graph?

My answer:

Precisely this:

```
ggplot(entrance, aes(x = score)) + geom_histogram(bins = 6)
```



It's a histogram, with some number of bins. `ggplot`'s histogram bins are always all the same

width (something I would expect you to have noticed in your work with making histograms), so there are actually *two* bins with one observation each on the left side (minus a half point for saying that there are five bins rather than six). Depending on how the printing comes out on the exam (which is always a bit of a lottery), you *might* be able to see that there is a vertical line marking the division between the leftmost two bins, at around a `score` of 43.

(b) [2] Explain briefly why the information you have so far says that you *should* use t-procedures to make inference about the population mean.

My answer:

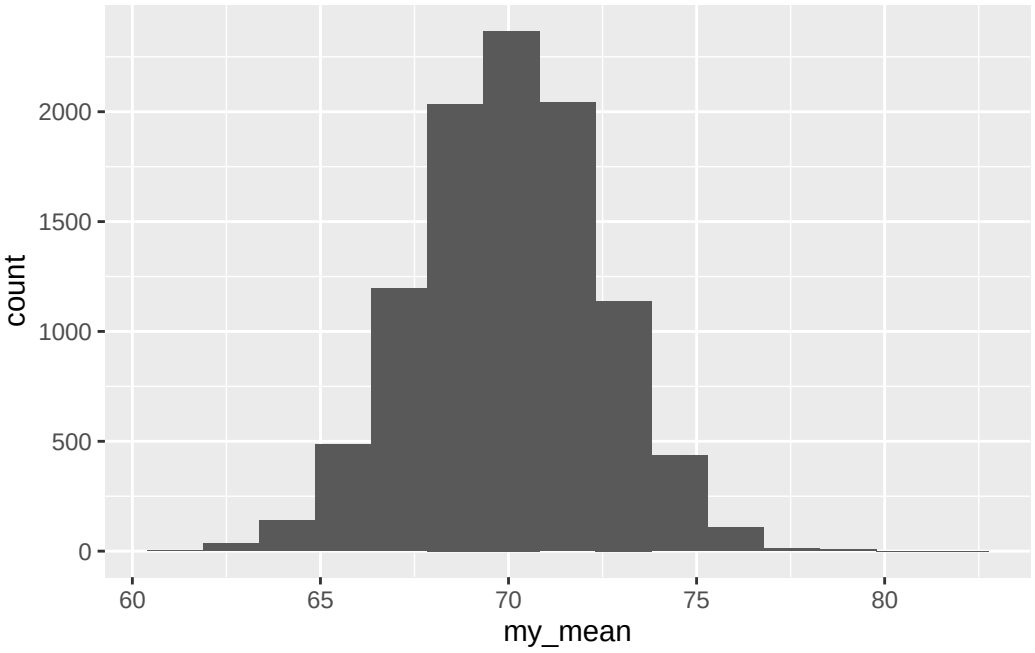
The histogram is somewhat skewed to the left (not really "outliers" on the left, on the evidence of this graph, because those two low values do not seem to be separated from the rest of the distribution). However, the sample size of 24 is large enough to offer us a fair bit of help (via the Central Limit Theorem), and so we can expect this sample size to be large enough to overcome the skewness.

One point each for relevant discussion of (i) the distribution shape, and (ii) the sample size. You could frame it as "the sample size is relatively large, therefore the distribution shape does not matter much", but you need to consider both things somehow.

Note that I told you which argument to make, so you have to come to a place of saying that the sample size *is* big enough.

Extra: the obligatory bootstrap sampling distribution of the sample mean:

```
set.seed(457298)
tibble(sim = 1:10000) %>%
  rowwise() %>%
  mutate(my_sample = list(sample(entrance$score, replace = TRUE))) %>%
  mutate(my_mean = mean(my_sample)) %>%
  ggplot(aes(x = my_mean)) + geom_histogram(bins = 15)
```



which is about as normal as you could wish for.

(c) [2] What code would obtain a 95% confidence interval for the population mean?

My answer:

This is the easiest version: 95% is the default, so this:

```
t.test(entrance$score)
```

```
    One Sample t-test

data:  entrance$score
t = 28.135, df = 23, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 64.89184 75.19150
sample estimates:
mean of x
 70.04167
```

or this:

```
with(entrance, t.test(score))
```

```
    One Sample t-test

data:  score
t = 28.135, df = 23, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 64.89184 75.19150
sample estimates:
mean of x
 70.04167
```

Minus a half point if you put in a `conf.level`: it works, but it is not necessary.

(d) [3] What do you conclude from the output given in Figure 6, in the context of the data? Explain briefly how you get from the information in the output to your conclusion.

My answer:

We are testing the null hypothesis that the population mean score is 72 (in the code) against the alternative that the mean is not 72 (implied by the code and given in the output). The P-value for the test is 0.4395, which is not smaller than 0.05 (or other α), so we cannot reject the null hypothesis, and so there is no evidence that the population mean score (on the entrance exam) is different from 72.

Your answer needs to make it clear than you know how to get from the output to:

- the null and alternative hypotheses
- the P-value
- a conclusion in words about entrance exam scores.

A point for each of those. It is enough if your answer makes it clear that you know what the null and alternative hypotheses are, but it is better if you explicitly state them.

(e) [2] Your friend looks at the output in Figure 6 and says "the sample mean was about 70. Of course this is different from 72." How do you respond to this, as a statistician?

My answer:

Your friend does not understand sampling variability. The logic here is, *if* the population mean is 72, you could very easily get a sample mean like 70, and on that basis, we conclude that the population mean could be 72. Of course, the population mean could easily be something like 70 as well, but we need good evidence to be able to reject a population mean of 72, and we don't have that.

The other way of thinking about this is to look at the 95% confidence interval in the output (which is the same one your code from earlier would have obtained), from about 65 to about 75. On the basis of our sample, any population mean between those two values is consistent with the sample we obtained, so there is no reason to doubt a statement (the null hypothesis) that the population mean is 72.

Somehow you need to get to saying that the sample mean of 70 is entirely consistent with a population mean of 72.

(There was a wide variety of ways in which you tackled this, which is a good thing because you evidently had to *think*, but it also meant that I had to read carefully over what you had written to see whether it built the bridge between what you know as a statistician and what your friend said.)

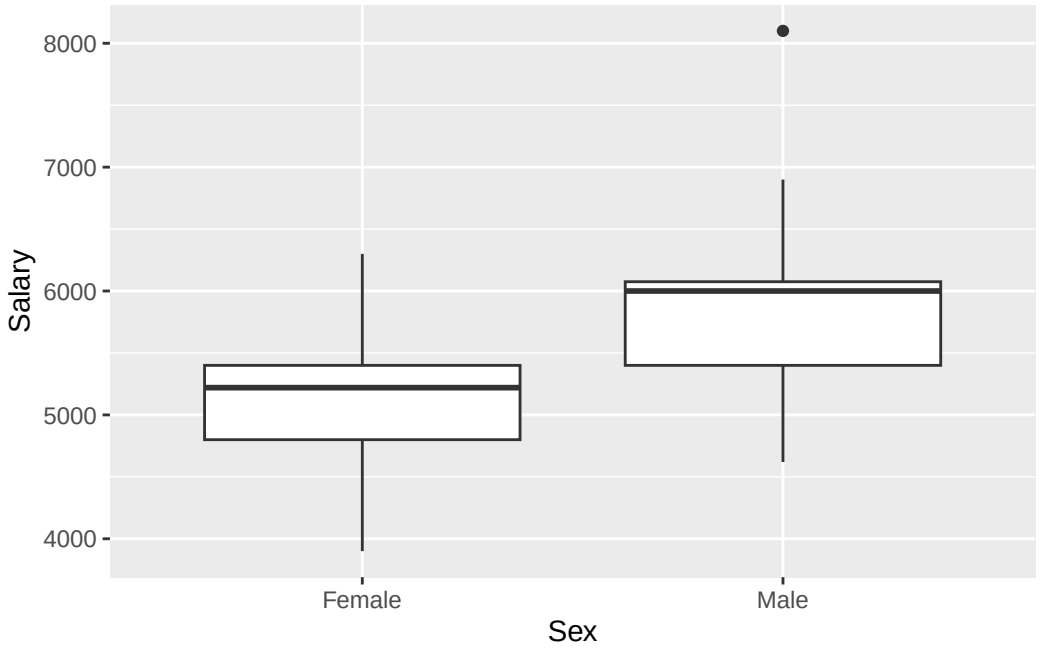
4. A bank hired many skilled entry-level clerical employees between 1969 and 1977. In a random sample of 93 of these employees, 32 of them identified as males, and 61 of them as female. The beginning salary for each employee, in 1970 dollars, was recorded. Some randomly chosen rows of the data are shown in Figure 7. The dataframe is called `bank`.

(a) [2] What code would draw a suitable graph of these data?

My answer:

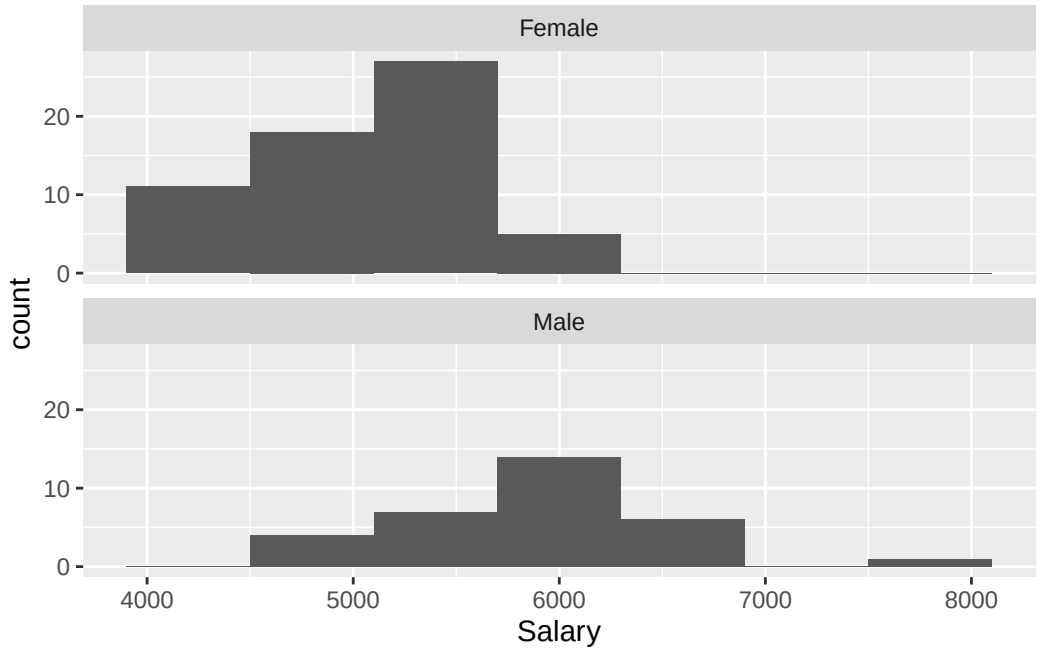
One quantitative variable (`Salary`) and one categorical one (`Sex`), so a boxplot:

```
ggplot(bank, aes(x = Sex, y = Salary)) + geom_boxplot()
```



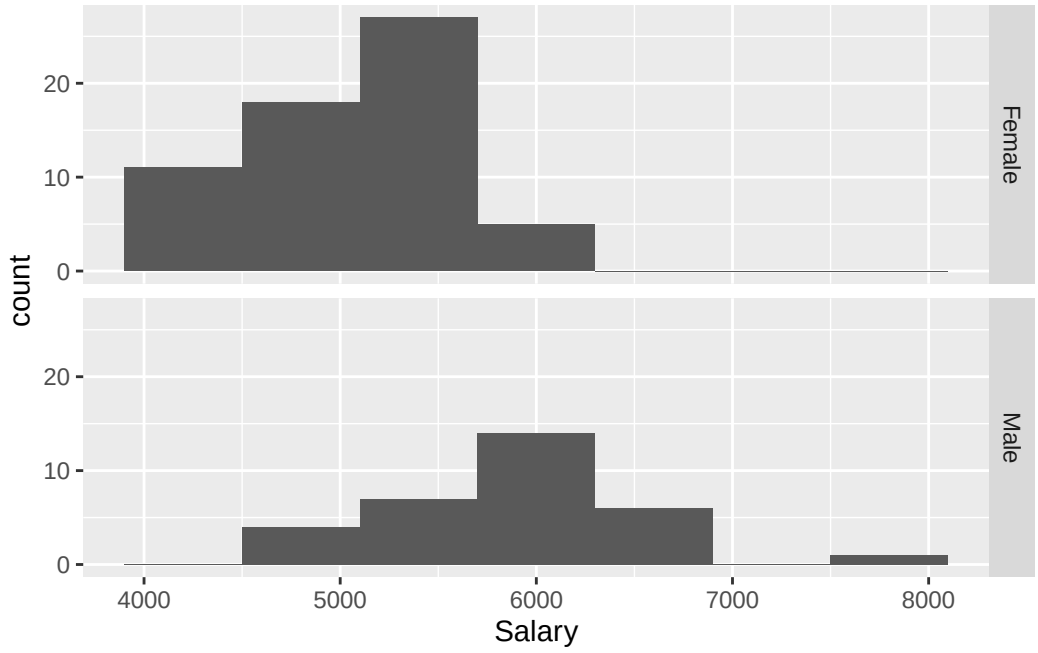
If you wanted to do facetted histograms (above and below), you could also do that (on the grounds that `Sex` is an "extra" categorical variable, and if you just had `Salary`, you would draw a histogram):

```
ggplot(bank, aes(x = Salary)) + geom_histogram(bins = 8) +
  facet_wrap(~ Sex, ncol = 1)
```



The point of drawing a graph here is to *compare* the distributions, so you need your facetted histograms to be above and below. Find a way to make that happen. If you have learned about `facet_grid` (it is probably in PASIAS somewhere), this would be another way to do it:

```
ggplot(bank, aes(x = Salary)) + geom_histogram(bins = 8) +
  facet_grid(Sex ~ .)
```



(b) [3] This dataset was used in a study at a business school of whether women were on average paid less than men for the same job. What code would carry out a suitable t-test to determine whether there is any evidence in favour of this hypothesis? You are not sure whether the spreads of salaries for males and females are equal. Which particular t-test does your code run?

My answer:

This needs to be a two-sample t-test, and because you are not sure about equality of spreads, a Welch test is the right one to run. The hypothesis given is the *alternative* hypothesis (the thing we are looking for evidence for), which is one-sided. The two categories of `Sex` (as shown in Figure 7) are `Female` and `Male` (in alphabetical order), so for our `alternative` we need to say how the first one compares with the second one in that order:

```
t.test(Salary ~ Sex, data = bank, alternative = "less")
```

```
    Welch Two Sample t-test

data:  Salary by Sex
t = -5.83, df = 51.329, p-value = 1.855e-07
alternative hypothesis: true difference in means between group Female and group Male is less t
95 percent confidence interval:
      -Inf -582.9857
sample estimates:
```

```
mean in group Female   mean in group Male
            5138.852             5956.875
```

The model formula (with the squiggle) needs to be first, but the `data` and the `alternative` are both named, so they can be in any order. `var.equal` is by default `FALSE`, so you don't need to have that there. It works with it, but it is not needed, so minus a half point if it is there. If you have `var.equal = TRUE`, you are running a pooled test, and you will lose a whole point. Likewise, minus a point if your code is for a two-sided test.

Two points for the t-test code, one for saying that you are running a Welch test. (I don't need a "why" here because the information given in the question is the why.)

Extra: now that you have seen the boxplot, you see that the two groups have almost the same spread, and so the pooled test would actually be fine:

```
t.test(Salary ~ Sex, data = bank, alternative = "less", var.equal = TRUE)
```

```

    Two Sample t-test

data:  Salary by Sex
t = -6.2926, df = 91, p-value = 5.378e-09
alternative hypothesis: true difference in means between group Female and group Male is less t
95 percent confidence interval:
      -Inf -601.9965
sample estimates:
mean in group Female   mean in group Male
            5138.852             5956.875
```

The P-value is also very small (actually a bit smaller), but the conclusion is the same.

I think this is one of those cases where the pooled test works, and so is a tiny bit more powerful than the Welch test.

(c) [2] The t-test for which you just gave code has a P-value of 1.9×10^{-7}. What do you conclude from this, in the context of the data?

My answer:

The null hypothesis is that mean salaries are equal for males and females; the alternative is that the mean is less for females. The P-value is extremely small, so reject the null in favour of the alternative, and conclude that the mean salary for females is less than for males.

If your code given above was for a two-sided test, then you need to be consistent with yourself and draw a two-sided conclusion here: drawing a one-sided conclusion from a two-sided test is an error. If you draw a two-sided conclusion here, expect the grader to scroll up and see what code you had; likewise, if your conclusion is one-sided but you code is not, that is also an error.

Extra: This is actually pretty good evidence of sex discrimination:

- the salaries we had were a random sample of all salaries (of entry-level employees doing this kind of job at this bank), so this is better than an observational study.
- the males and females here were all hired to do about the same job ("entry-level clerical"), so the salaries should be about the same, but the difference is actually huge (between about $400 and $1200 in 1970 dollars, with 99% confidence).

The way you would have to argue *against* discrimination is that the male and female employees were systematically different when they were hired (unlikely, since these are "entry-level" jobs so something like the amount of prior experience shouldn't matter), or that the *kind* of entry-level jobs at this bank done by males are different from the kind of jobs done by females. This last is more likely, especially in 1970, when a company would have an army of (usually) female staff to type and send letters (or take and distribute meeting minutes) who would not be paid very much (the sort of thing that would be done by email nowadays, by more senior employees themselves).

(d) [2] What code would obtain a 99% confidence interval for the difference in mean salary?

My answer:

```
t.test(Salary ~ Sex, data = bank, conf.level = 0.99)
```

```
    Welch Two Sample t-test

data:  Salary by Sex
t = -5.83, df = 51.329, p-value = 3.71e-07
alternative hypothesis: true difference in means between group Female and group Male is not eq
99 percent confidence interval:
 -1193.3688  -442.6763
sample estimates:
mean in group Female   mean in group Male
            5138.852             5956.875
```

Don't forget the `conf.level`, since it is not the default 95%! Also, remember to remove the `alternative` from your test code, since confidence intervals are by their nature two-sided.

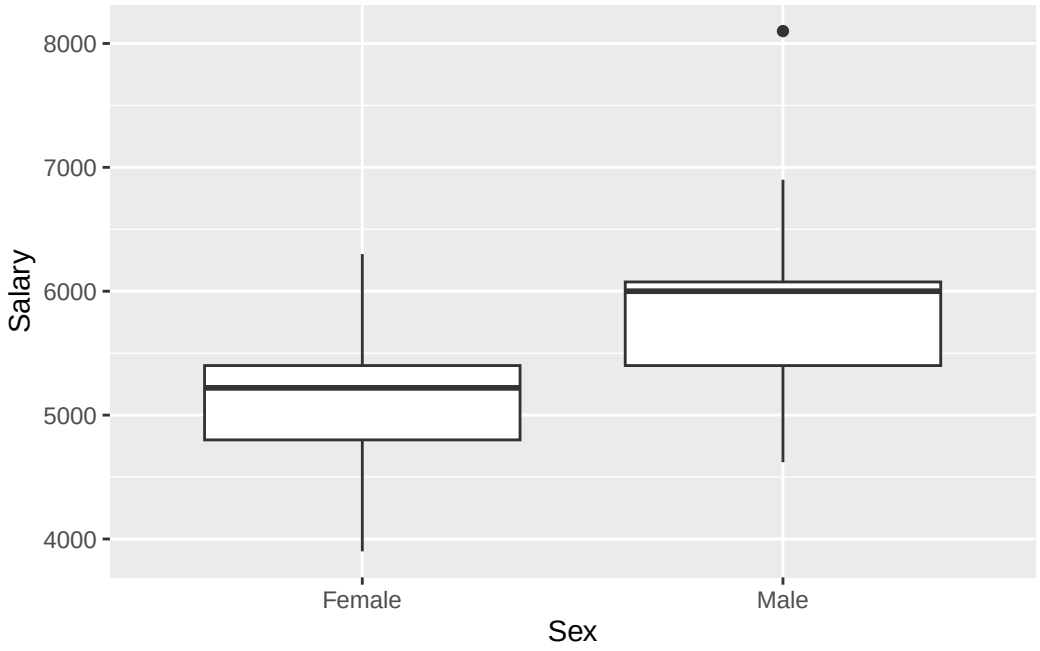
(e) [2] The distribution of salaries for females had one moderate outlier at the upper end of the distribution. Explain briefly why, nonetheless, it may well be reasonable to run a t-test to compare mean salaries.

My answer:

The sample size for females is 61, which is large enough for the Central Limit Theorem to help us (to get a normal sampling distribution of the sample mean for females even in the presence of an outlier). Said differently, the sample size for females is likely to be large enough to overcome the effect of the outlier.

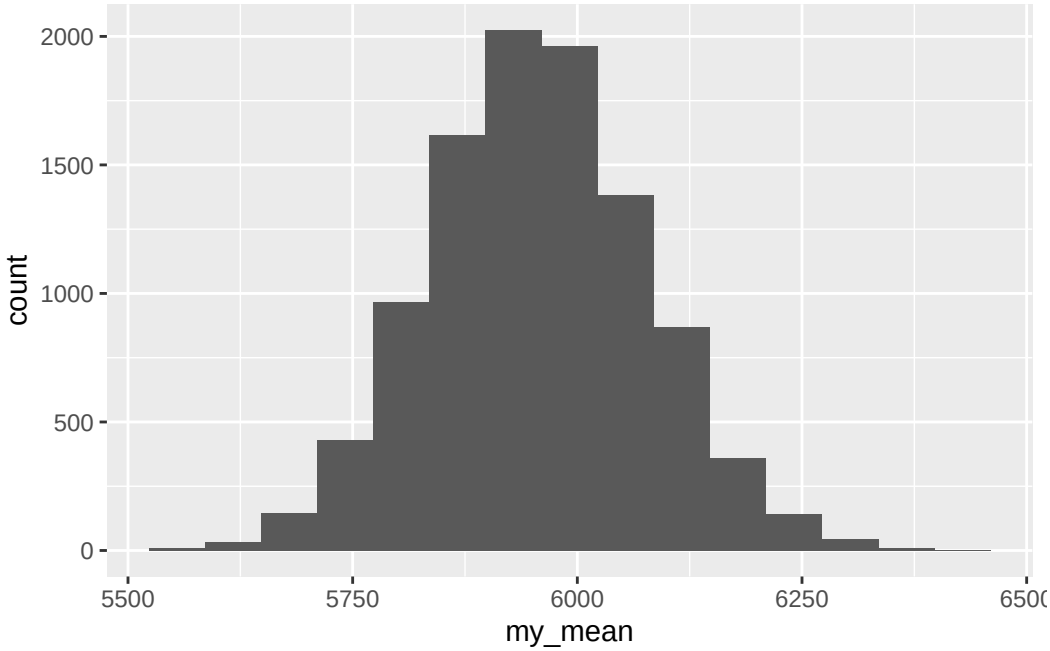
Extra: so I lied: it was actually the *male* distribution that had the upper outlier:

```
ggplot(bank, aes(x = Sex, y = Salary)) + geom_boxplot()
```



I wanted to make it clearer for you, though, so I said that it was the larger sample that had the outlier. A severe outlier might be another story, but a sample size of 61 would certainly be expected to be big enough to deal with a "moderate outlier" like this one. In fact, a sample size of 32 might even be big enough. We can find out about that by obtaining a bootstrap sampling distribution, grabbing just the males' salaries first:

```
set.seed(457298)
bank %>% filter(Sex == "Male") -> males
tibble(sim = 1:10000) %>%
  rowwise() %>%
  mutate(my_sample = list(sample(males$Salary, replace = TRUE))) %>%
  mutate(my_mean = mean(my_sample)) %>%
  ggplot(aes(x = my_mean)) + geom_histogram(bins = 15)
```



This looks close to normal, so it appears that a sample size of 32 would even be big enough, so a sample of size 61 would *certainly* take care of an outlier of that size.

5. A continuous random variable X is said to have a beta distribution if its density function is $Cx^{a-1}(1-x)^{b-1}$ for $0 \leq x \leq 1$, and 0 otherwise, where C is a constant not depending on x. The mean of a beta distribution is $a/(a+b)$. The beta density for $a = 2, b = 6$ is shown in Figure 13. To generate a random sample of size n from a beta distribution with parameters a and b, use the code `rbeta(n, a, b)`.

   (a) [1] Describe the shape of the beta distribution shown in Figure 13.

My answer:

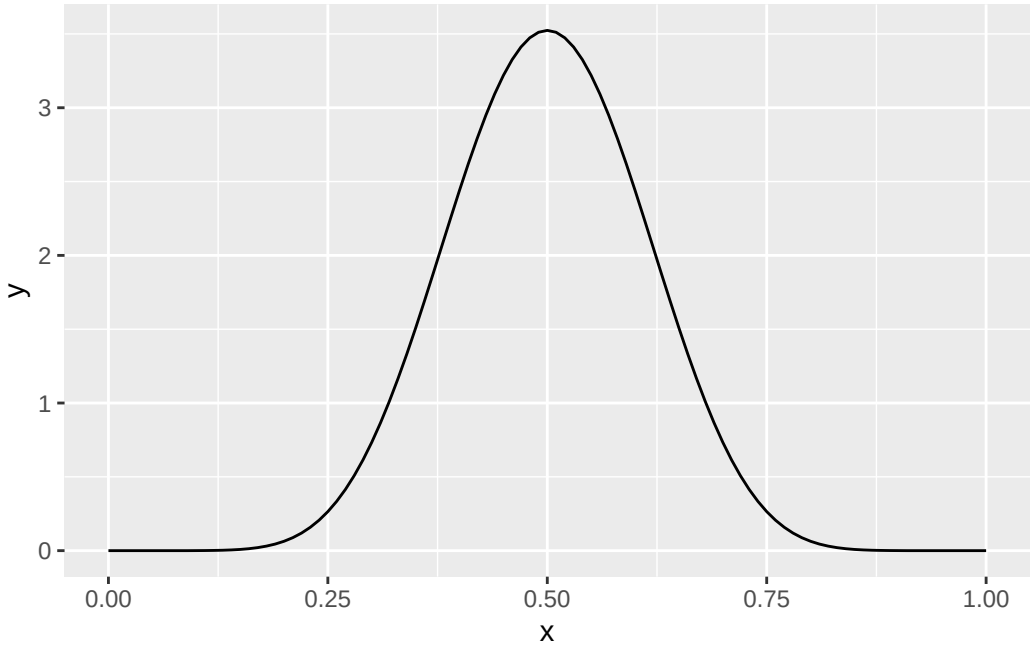
Skewed to the right. A very easy point. Add an adjective like "somewhat" or "moderately" if you like. I wouldn't call this "severe" skewness, but that's a judgement call.

Extra: in a beta distribution, if $a < b$ it is skewed to the right, if $a = b$ it is symmetric, and if $a > b$ it is skewed to the left. The beta distribution with $a = b = 1$ is the uniform distribution,

and with $a = b > 1$ it looks a lot like a normal distribution, but restricted to the interval $[0, 1]$.

Copying and editing my code, here's a beta distribution with parameters 10 and 10:

```
tibble(x = seq(0, 1, 0.01)) %>%
  mutate(y = dbeta(x, 10, 10)) %>%
  ggplot(aes(x = x, y = y)) + geom_line()
```



That looks pretty bell-shaped.

(b) [3] The code in Figure 14 produces the output shown below the code. What precisely does this tell you about the appropriateness of a t-test, in particular a t-test under what circumstances?

My answer:

This is a simulation of the sampling distribution of the sample mean for samples of size 25 from a beta distribution with parameters 2 and 6 (that is, the one shown in Figure 13). The normal quantile plot shows that this is very close to normal (the points are very close to the line), and therefore that a t-test for the mean will be completely appropriate for samples of this size from this distribution. You could say that a sample of size 25 is very much large enough to overcome the skewness in the original distribution. (The earlier one-point part was to point you in the direction of being able to say this.)

This is *not* a bootstrap, since we are not sampling from a sample; we are sampling from a distribution and repeatedly finding the sample mean, and plotting the sampling distribution of the sample mean (as simulated). You can call it a simulation, but not a bootstrap.

I don't want to know what each original line of code does, just what you learn from the whole thing. If you want to go through the code in your head to work out what it's doing, be my guest, but I don't want that to appear in your answer. What I want you to do is to notice that it looks like the code for bootstrapping the sampling distribution of the sample mean (with a normal quantile plot at the end rather than a histogram), but with the key detail that the sampling from the sample has been replaced by sampling from a distribution.

(c) [3] Some more code and output is shown in Figure 15. What precisely do you conclude from this Figure, and how do you know that the t-test used here is appropriate?

My answer:

The output, with counts of the numbers of P-values greater than or less than 0.05, tells you that this is a power by simulation (estimated power). The line with `my_sample` on it says that you are taking a sample of size 40 from a beta distribution with parameters 2 and 6 (and thus mean $2/(2+6) = 0.25$). The line with `t.test` in it below that says that we are testing a mean of 0.3. Thus, the power to reject a mean of 0.3 (against a two-sided alternative) when the mean is actually 0.25 with a sample of size 40 from this beta distribution is 0.574. We know that the t-test is appropriate because in the previous part we found that a sample of size 25 is large enough, and so this larger sample of size 40 is definitely large enough.

(d) [1] Based on the Figures you have seen in this question, what can you say about the sample size required to obtain a power of 0.80 in this situation (without changing anything else)? Explain (very) briefly.

My answer:

The sample size will need to be bigger than 40, because to obtain a bigger power without changing anything else, you will need a larger sample size.

This is meant to be an easy one to finish the question with, but you'll need *some* kind of justification to get the whole point.

Extra: More than that we cannot say. If you were doing this yourself, you might redo the simulation with a sample size like 60, and see how close that brings you to the power you want. Since I have the original code and the ability to copy and paste, I can do that:

```
tibble(sim = 1:1000) %>%
  rowwise() %>%
  mutate(my_sample = list(rbeta(60, 2, 6))) %>%
  mutate(my_t = list(t.test(my_sample, mu = 0.3))) %>%
  mutate(p_value = my_t$p.value) %>%
  count(p_value <= 0.05)
```

p_value <= 0.05	n
FALSE	258
TRUE	742

Not quite there yet. A slightly bigger sample size still is needed. 65? 70? Something like that.

6. Concrete structures are usually strengthened by the use of "reinforced concrete", which has steel bars embedded in it to increase the strength of the concrete. In places that are affected by severe weather, however, there can be problems with corrosion of the steel bars. A study investigated the effect of using plastic bars reinforced with glass fibre (these are called "GFRP rebars") instead of steel ones. The experimenters measured the bond strength of 48 GFRP rebars, in suitable units. Some of the data is shown in Figure 9, in dataframe `bonding`. There is a problem if the average (mean or median, as appropriate) bonding strength is less than 10 in these units, and the experimenters are interested in detecting whether there is evidence of a problem based on these data.

   (a) [2] A one-sample boxplot is shown in Figure 10. From this graph, why do you think the experimenters chose to use a sign test rather than a t-test to detect whether there is a problem?

My answer:

There are two approaches you might take (either is good):

- the distribution is very skewed to the right (and/or has outliers), so the median would be a better summary than the mean, and therefore the sign test (which uses the median) would be better than the t-test (which uses the mean).
- the distribution is very skewed to the right (and/or has outliers). This is very non-normal, and it is not clear whether the sample size of $n = 48$ is large enough to overcome the non-normality and allow us to use a t-test for the mean. (You will have to do some hand-waving about the sample size, but the place you want to get to here is "the sample size is not big enough", so that's the argument you need to make.)

The second approach is the usual one based on the Central Limit Theorem; the first one is the one we used for the IRS data in lecture, where we said that the median was the right summary to use, without even thinking about the sample size. If you're talking about an appropriate summary, the sample size does not matter, but if you're (implicitly or explicitly) trying to use the Central Limit Theorem, you *must* include the sample size in your discussion. I am assuming that your answer is based on the second approach (which needs discussion of the sample size) unless you say that you prefer to test the median rather than the mean, and why. For example, you might say "the mean is distorted by the outliers, while the median is not, so we prefer the sign test, which uses the median". This shows that you understand the issues beyond a rather superficial "the data are not normal, so we cannot use a t-test" which is actually *false* if the sample size is big enough.

Extra: I gave you a one-sample boxplot here instead of a histogram partly for variety, but also partly because the outliers and the long tail showed up really clearly, to guide you towards a good answer.

(b) [3] A sign test for these data, addressing the issues raised in the question, is shown in Figure 11. What do you conclude from this Figure, keeping in mind what the experimenters are hoping to learn from the data? In your answer, state your null and alternative hypotheses, P-value, and conclusion in the context of the data.

My answer:

The null hypothesis is that the population median is equal to 10 (or, perhaps more logically, greater than or equal to 10). The alternative hypothesis is that the population median is (strictly) less than 10. We are looking for evidence of a problem, so whatever corresponds to "a problem" has to be the alternative hypothesis.

The appropriate P-value is the lower-tail one-sided one, 0.001.

The P-value is less than 0.05 (or whatever α you are using), so reject the null hypothesis in favour of the alternative, and conclude that the population median bonding strength is less than 10, and therefore that there is a problem (the GFRP rebars are not strong enough).

Minus a point for anything erroneous or missing. If you have a two-sided alternative hypothesis, as long as you follow through with a two-sided conclusion, you get 1.5 because (i) your alternative is wrong, and (ii) you didn't address the experimenters' concern.

(c) [2] A confidence interval for the median is shown in Figure 12. Explain carefully how this confidence interval is consistent with the output in Figure 11.

My answer:

The word "carefully" in one of my questions means that there are some details that you need to pay attention to. Here these are:

- the confidence interval is a 99% one, not a 95% one
- confidence intervals are two-sided.

So what you have to do is to look at the *two*-sided P-value in Figure 11, which is 0.002, and compare it to the appropriate thing for a 99% CI, which is 0.01 ($1-0.99$). The correct P-value for the comparison, 0.002, is smaller than 0.01, so the null median of 10 should be *outside* the confidence interval, which it is.

A careful answer makes it clear that you are comparing the right P-value (the two-sided one) with the right thing (0.01). An answer like "we rejected the null hypothesis that the median is 10, and 10 is outside the confidence interval" is only one point, because you have not been careful enough: you are actually comparing apples and oranges, because you were doing a one-sided test at (probably) an α that did not correspond with the two-sided confidence interval. Beyond that, a half point more for mentioning the correct P-value for assessing a confidence interval (the two-sided one), and another half point more for mentioning the right thing to compare it with (0.01).

Another way you might approach this is to start with 10 being outside the 99% CI, and hence the P-value of a two-sided test is less than 0.01. Hence, as long as you are on the correct side

(you are), the P-value of the one-sided test we did earlier is less than 0.005, and thus we should have rejected a median of 10 as we did, in favour of our one-sided alternative.

Somehow, build the bridge between having rejected a median of 10 in a one-sided test, and 10 being outside that 99% CI.

The point of a question like this is to show me that you *understand* what you are doing.

Extra 1: you might be interested in how a t test and CI compares here:

```
t.test(bonding$strength, mu = 10, alternative = "less")
```

```
    One Sample t-test

data:  bonding$strength
t = -2.7335, df = 47, p-value = 0.004403
alternative hypothesis: true mean is less than 10
95 percent confidence interval:
     -Inf 9.258243
sample estimates:
mean of x
 8.079167
```

The P-value of about 0.004 is still significant, but it is larger than the one from the sign test. This is interesting: when both tests apply, the t-test will be more powerful than the sign test, so we would expect to see a P-value that is *smaller* than the 0.001 than we got from the sign test, if anything. This suggests that maybe the t-test is not best after all.

To get a confidence interval for the mean, a 99% one to be consistent with the `ci_median` in Figure 12, take out the `alternative` and put in a `conf.level`. Remove the `mu` as well (we are not doing a test now, so it is no longer needed):

```
t.test(bonding$strength, conf.level = 0.99)
```

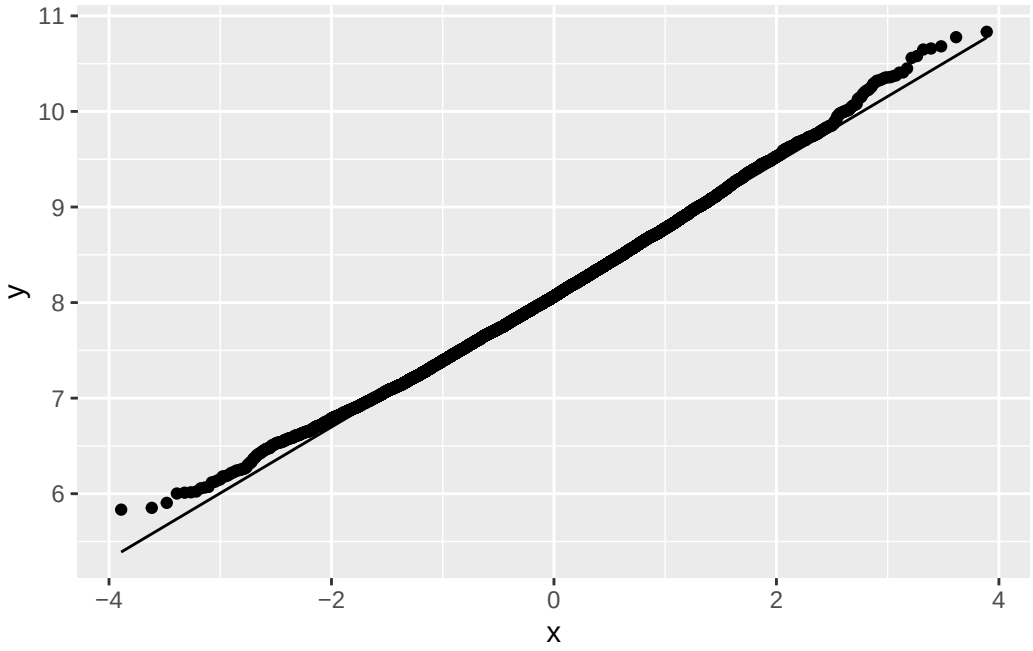
```
    One Sample t-test

data:  bonding$strength
t = 11.497, df = 47, p-value = 2.938e-15
alternative hypothesis: true mean is not equal to 0
99 percent confidence interval:
 6.192734 9.965600
sample estimates:
mean of x
 8.079167
```

This is not dissimilar to the interval for the median, but both the lower and upper limits for the mean are a bit higher than the ones for the median. This is probably because the mean is

getting pulled further upwards than it should by the outliers (and the long right tail). Once again, we should be suspicious about whether this interval is to be trusted.

Extra 2: the inevitable bootstrap sampling distribution of the sample mean. I'm going to go straight to a normal quantile plot (rather than a histogram), so I'm doing 10,000 simulations:

```
set.seed(457299)
tibble(sim = 1:10000) %>%
  rowwise() %>%
  mutate(my_sample = list(sample(bonding$strength, replace = TRUE))) %>%
  mutate(my_mean = mean(my_sample)) %>%
  ggplot(aes(sample = my_mean)) + stat_qq() + stat_qq_line()
```



This is a little bit skewed to the right, although not much, so that if you really want to do inference for the mean, the t-test is actually not so bad. Maybe the stronger argument for using the sign test here is the "summary" one: the data distribution is so skewed that we should be using the median to summarize it, regardless of the sample size.

Use this page if you need more space. Be sure to label any answers here with the question and part they belong to.

Numbered Figures begin here, in with caption and label:

```
library(tidyverse)
library(smmr)
```

Figure 1: Packages

performance

```
Time;What;Report
28.15;Visual;Visual
10.91;Verbal;Verbal
15.85;Visual;Visual
10.9;Verbal;Verbal
6.91;Visual;Verbal
11.37;Verbal;Verbal
12.64;Visual;Verbal
9.18;Visual;Verbal
8.44;Verbal;Visual
```

Figure 2: Performance data file (some)

sleep study

AnxietyScore	AnxietyStatus	ClassesMissed	ClassYear	GPA	LarkOwl	WeekdaySleep
2	normal	4	3	3.90	Neither	8.85
13	moderate	2	1	3.20	Lark	7.20
6	normal	0	2	3.40	Neither	8.60
2	normal	0	3	3.30	Neither	7.97
12	moderate	0	2	3.70	Neither	8.23
0	normal	3	4	3.00	Neither	8.22
10	moderate	4	4	3.25	Neither	8.90
2	normal	0	3	2.90	Neither	7.15
0	normal	0	2	3.20	Neither	7.95
5	normal	20	3	2.80	Owl	7.20
25	severe	4	4	3.20	Owl	6.09
7	normal	0	1	3.90	Neither	6.00
5	normal	0	2	2.70	Neither	10.20
0	normal	0	2	3.60	Neither	5.25
2	normal	0	4	3.79	Lark	7.83

Figure 3: Sleep study data (some)

entrance

score
63
59
75
75
91
77
84
81
55
64

Figure 4: Entrance exam data (10 randomly chosen rows)

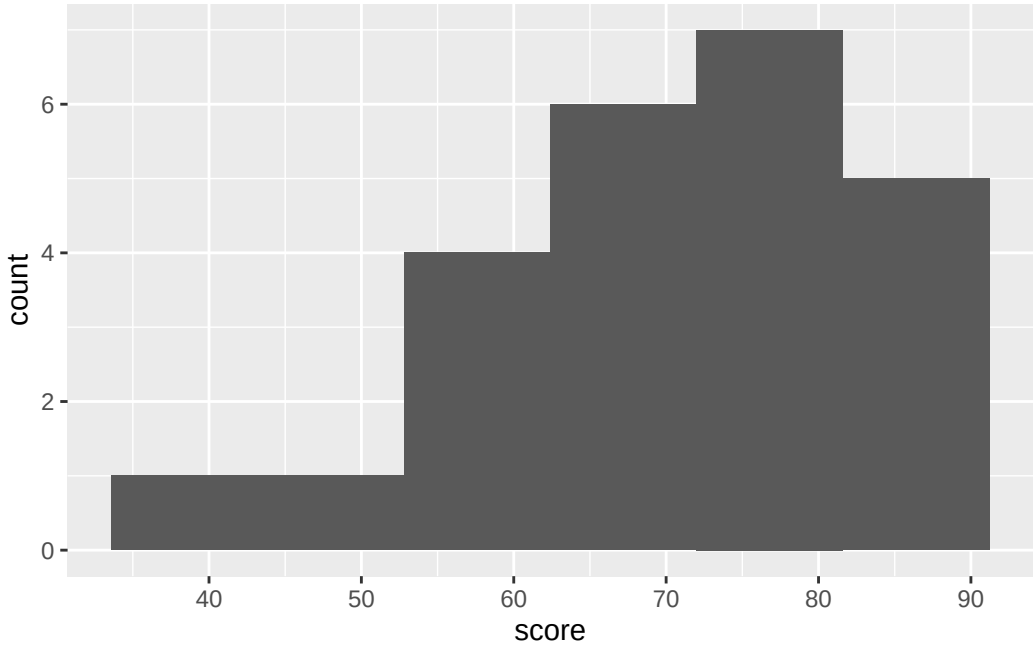


Figure 5: Entrance exam scores graph

```
with(entrance, t.test(score, mu = 72))
```

```
    One Sample t-test

data:  score
t = -0.78665, df = 23, p-value = 0.4395
alternative hypothesis: true mean is not equal to 72
95 percent confidence interval:
 64.89184 75.19150
sample estimates:
mean of x
 70.04167
```

Figure 6: Entrance exam scores hypothesis test

bank

Salary	Sex
5040	Male
4800	Female
5400	Female
4980	Female
5640	Female
5280	Female
5100	Female
5400	Female
6000	Male
6000	Male
6300	Male
5280	Female
5580	Female
5400	Male
5400	Female

Figure 7: Bank salary data (some randomly chosen rows)

```
    Welch Two Sample t-test

data:  Salary by Sex
t = -5.83, df = 51.329, p-value = 1.855e-07
alternative hypothesis: true difference in means between group Female and group Male is less than 0
95 percent confidence interval:
      -Inf -582.9857
sample estimates:
mean in group Female   mean in group Male
            5138.852             5956.875
```

Figure 8: Bank salary t-test output

bonding

	strength
35	14.2
44	3.8
1	11.5
7	6.6
27	20.6
46	3.6
39	5.1
26	3.4
33	8.2
23	3.9
25	3.6
48	3.6
17	4.9
41	5.2
9	13.4

Figure 9: Bonding strength data (some)

```
ggplot(bonding, aes(x = 1, y = strength)) + geom_boxplot()
```

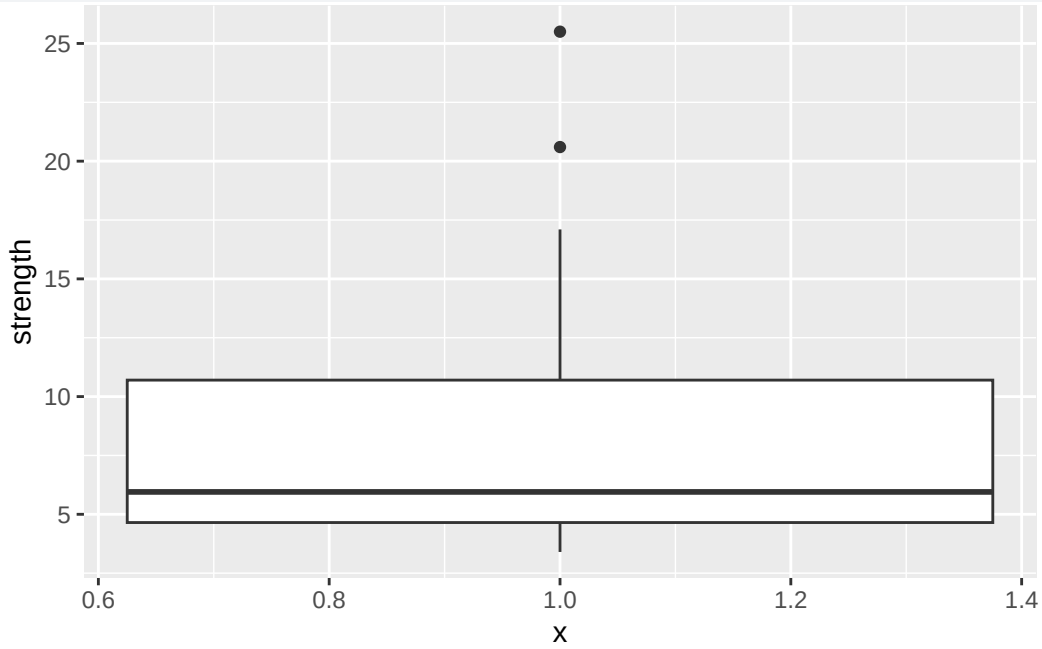


Figure 10: Bonding strength one-sample boxplot

```
sign_test(bonding, strength, 10)
```

```
$above_below
below above
   35    13

$p_values
  alternative     p_value
1       lower 0.001044054
2       upper 0.999641365
3   two-sided 0.002088107
```

Figure 11: Bonding strength sign test

```
ci_median(bonding, strength, conf.level = 0.99)
```

```
[1] 5.000098 9.296399
```

Figure 12: Confidence interval for median bonding strength

beta

```
tibble(x = seq(0, 1, 0.01)) %>%
  mutate(y = dbeta(x, 2, 6)) %>%
  ggplot(aes(x = x, y = y)) + geom_line()
```

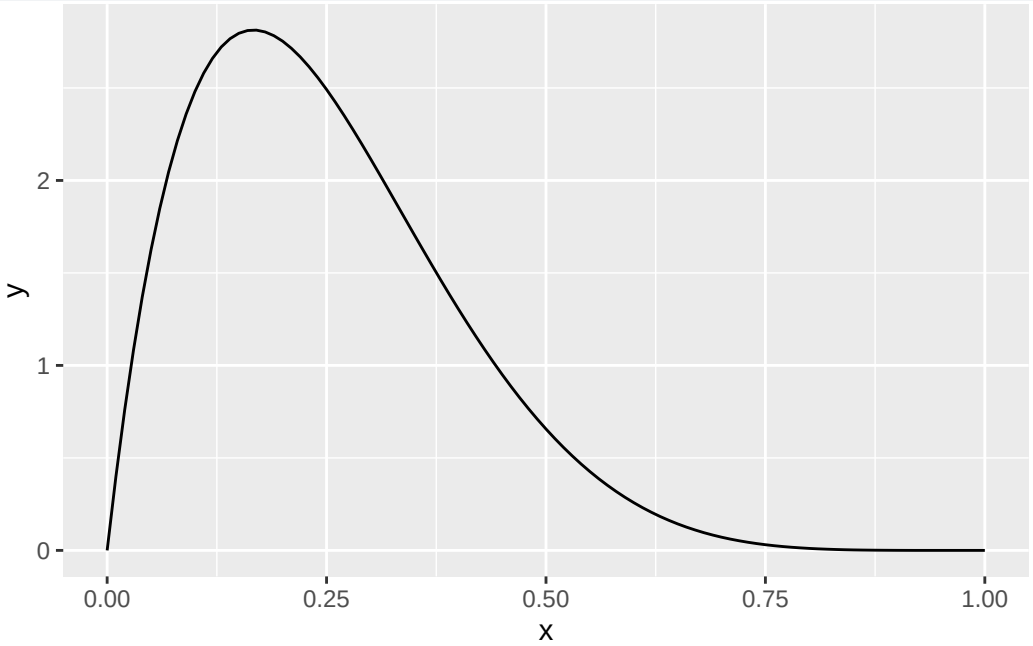


Figure 13: Beta density for $a = 2, b = 6$

```r
tibble(sim = 1:10000) %>%
  rowwise() %>%
  mutate(my_sample = list(rbeta(20, 2, 6))) %>%
  mutate(my_mean = mean(my_sample)) %>%
  ggplot(aes(sample = my_mean)) + stat_qq() + stat_qq_line()
```

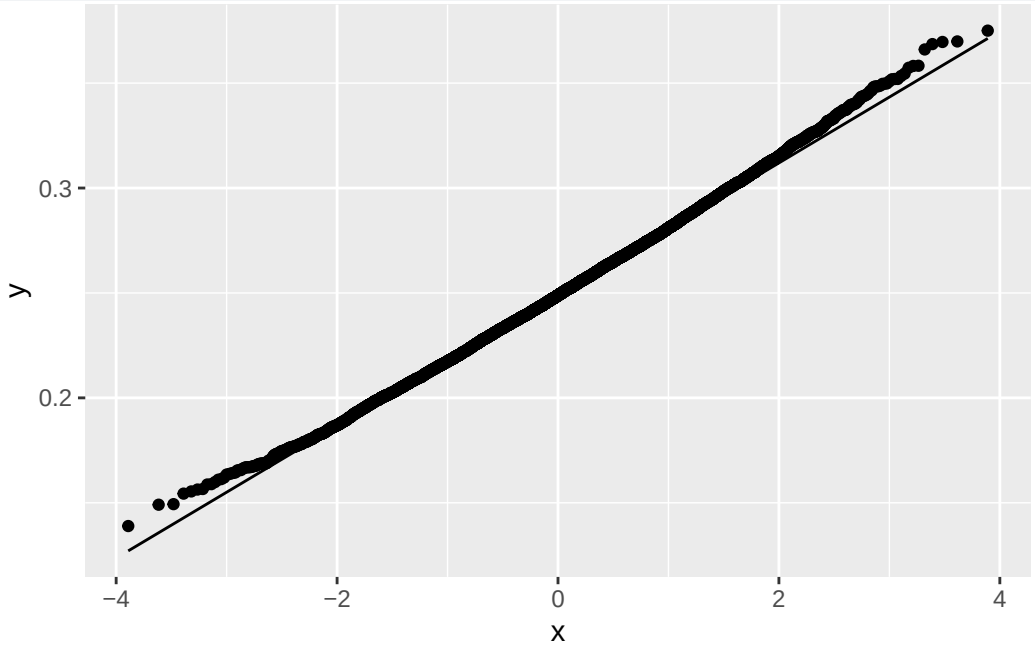


Figure 14: Beta code and output 1

```r
tibble(sim = 1:1000) %>%
  rowwise() %>%
  mutate(my_sample = list(rbeta(40, 2, 6))) %>%
  mutate(my_t = list(t.test(my_sample, mu = 0.3))) %>%
  mutate(p_value = my_t$p.value) %>%
  count(p_value <= 0.05)
```

p_value <= 0.05	n
FALSE	426
TRUE	574

Figure 15: Beta code and output 2