University of Toronto Scarborough
Department of Computer and Mathematical Sciences
STAC32 (K. Butler), Midterm Exam
November 4, 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.

The University of Toronto's Code of Behaviour on Academic Matters applies to all University of Toronto Scarborough students. The Code prohibits all forms of academic dishonesty including, but not limited to, cheating, plagiarism, and the use of unauthorized aids. Students violating the Code may be subject to penalties up to and including suspension or expulsion from the University.

Reading data files

The following questions are about reading data files into R. Here, and elsewhere on this exam where I ask for code, you *do not* need to show the output that the code will produce; full credit can be obtained by giving correct code.

(1) (3 points) Read Scenario A in Figure 2. What R code would read these data into an R dataframe, and display at least some of that dataframe?

The data values are separated by single semi-colons, so this calls for `read_delim`. You can assume that the `tidyverse` is loaded (see exam instructions):

```
plans <- read_delim("college-plans.txt", ";")
```

```
Rows: 16 Columns: 4
-- Column specification --------------------------------------------------------
Delimiter: ";"
chr (3): social_stratum, encouragement, college_plans
dbl (1): frequency

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.
```

```
plans
```

```
# A tibble: 16 x 4
   social_stratum encouragement college_plans frequency
   <chr>          <chr>         <chr>             <dbl>
 1 lower          low           no                  749
 2 lower          low           yes                  35
 3 lower          high          no                  233
 4 lower          high          yes                 133
 5 lowermiddle    low           no                  627
 6 lowermiddle    low           yes                  38
 7 lowermiddle    high          no                  330
 8 lowermiddle    low           no                  303
 9 uppermiddle    low           no                  627
10 uppermiddle    low           yes                  38
11 uppermiddle    high          no                  374
12 uppermiddle    high          yes                 467
```

```
13 higher         low           no                  153
14 higher         low           yes                  26
15 higher         high          no                  266
16 higher         high          yes                 800
```

In my solutions, I display the answer, to show that my code does indeed work.

Two points for the first line, and one for the second line (or something equivalent to it that will display at least some of the dataframe, like `View(plans)`). Minus a half point for giving your dataframe a name that does not say something about what is in it (thus `plans` or `college` or `students` or something like that is fine, but `data` or `mydata` or equivalent is not).

The column names might be two words separated by an underscore, but the thing separating the words is not a semicolon, so the pairs of words will be kept together.

(2) (2 points) Read Scenario B in Figure 4. What R code would read these data into an R dataframe, and display at least some of that dataframe?

The thought process needs to be that the data values in the file are separated by whitespace (actually spaces), *and* that there are varying numbers of them each time (sometimes only one, sometimes several). Thus `read_delim` will not work, and we need to use `read_table`:

```
dogs <- read_table("dogs2.txt")
```

```
-- Column specification -------------------------------------------------------
cols(
  Drug = col_character(),
  lh0 = col_double(),
  lh1 = col_double(),
  lh3 = col_double(),
  lh5 = col_double()
)
```

```
dogs
```

```
# A tibble: 8 x 5
  Drug           lh0   lh1   lh3   lh5
  <chr>        <dbl> <dbl> <dbl> <dbl>
1 Morphine     -3.22 -1.61 -2.3  -2.53
2 Morphine     -3.91 -2.81 -3.91 -3.91
3 Morphine     -2.66  0.34 -0.73 -1.43
4 Morphine     -1.77 -0.56 -1.05 -1.43
5 Trimethaphan -3.51 -0.48 -1.17 -1.51
6 Trimethaphan -3.51  0.05 -0.31 -0.51
7 Trimethaphan -2.66 -0.19  0.07 -0.22
8 Trimethaphan -2.41  1.14  0.72  0.21
```

Despite the messy layout of the file (and even though the columns were not lined up), the data were read in properly. (The logic is: `read_table` works whenever the columns are separated by varying amounts of whitespace; this includes times where the columns are lined up, as in the lecture example, but also includes messier things like this one, as long as the thing separating the data values is whitespace and not something else.)

This one is likely to be two points (if you had `read_table`) or nothing, as long as you get the filename right. There is no credit for `read.table`, which was not taught in this course. Make sure your underscore *looks* like an underscore.

Grading note: if you forgot to display the dataframe in the previous question *and* you forgot here as well, you should not get penalized again. If you remembered to display it in the previous question but forgot to do so here, it's only minus 0.5 (the point of this question was to recognize that you needed `read_table`).

Extra: these data came from D29, and you will see them again there. What makes the analysis difficult is that we have four measurements at different times for the *same* dog, rather than the measurements at different times coming from *different* dogs, as you would expect in an ANOVA situation. This is more like matched pairs (where you would have two measurements, paired up, on the same individual); here, we have four measurements on each dog, and the appropriate technique is called "repeated measures".

Also, the data as used in D29 are in aligned columns, but I deliberately messed them up to make you think about what you would need to do.

(3) (3 points) Read Scenario C in Figure 6. Describe the process you would use to get the data into a dataframe called `animals` in R Studio on `r.datatools.utoronto.ca`. Give enough detail to allow your reader to reproduce your process and get the same results you did. (You may assume that R Studio on `r.datatools` is currently open in a tab in your web browser.) Your answer will probably need to contain words *and* code.

What you need to describe is how to get the spreadsheet file from where it is now (an attachment in your email) to R Studio on `r.datatools`, and, having done that, how to read it in. That is, some steps like these (one point each):

- save the email attachment to a file on your computer (you can name the folder if you like)
- in the R Studio files pane (bottom right), click on Upload, select the file on your computer, and click OK to upload it. It is probably smart also to check the list of Files to see that it uploaded (that you have a file called `animals.xlsx` in there), but this is not obligatory.
- in R Studio, run this code:

```
animals <- read_excel("animals.xlsx", sheet = "Sheet1")
```

You don't need a `library(readxl)` (see Figure 1), but no harm if you put it in.

You need to express that *you* saved the email attachment somewhere (that you will be able to find it from), otherwise it is not clear enough where your reader will need to upload it from.

Diabetes

The dataframe shown in Figure 7 contains five numerical measurements made on 145 non-obese adult patients classified into three groups. The dataframe is called `diabetes`.

The three primary variables are glucose intolerance (`glucose`), insulin response to oral glucose (`insulin`) and insulin resistance (`sspg`). Two additional variables, the relative weight (`rw`) and fasting plasma glucose (`fpg`), are also included.

The column `group` contains the classifications of the subjects into three groups, obtained by current medical criteria: "normal", "chemical diabetic", and "overt diabetic".

(4) (2 points) What would be an appropriate graph of the insulin resistance and group variables (one graph showing both variables)? Explain briefly why your graph is an appropriate choice.

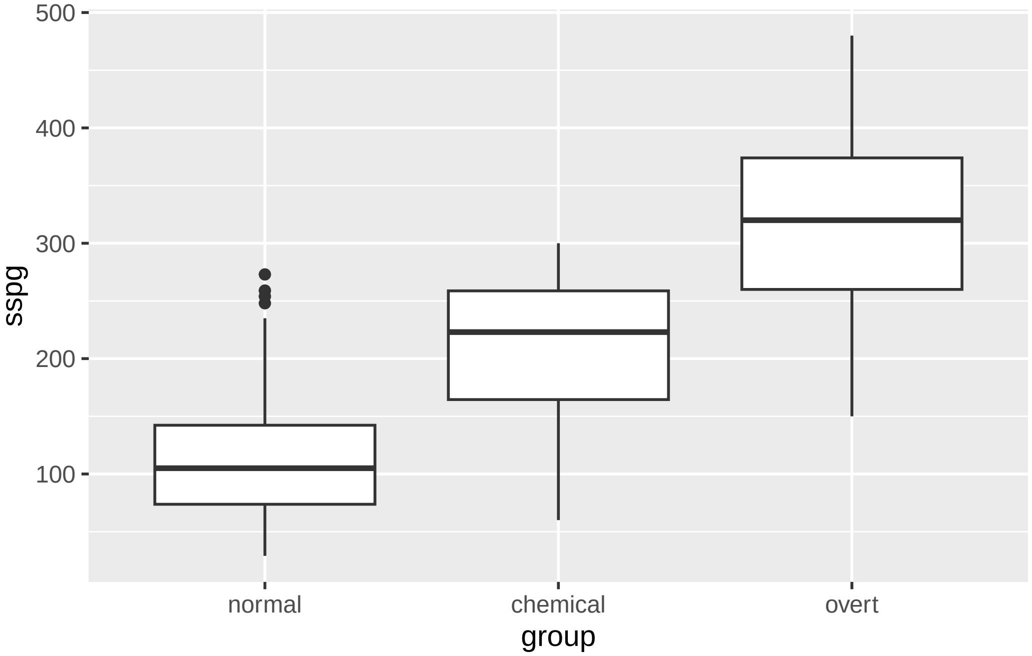
A boxplot. These two variables are quantitative (insulin resistance) and categorical (group), so this would be an appropriate graph that shows the two of them together. (Equivalently, say that there are three groups and we want a graph that will show the insulin resistance for each group so that we can compare them.)

Make sure you say *which* variable is quantitative and which one is categorical, or else it looks as if you don't know.

(5) (3 points) What code would draw the graph that you named in the previous part?

Bear in mind that the insulin *resistance* is in the column `sspg`, so you need that column and not `insulin` (which is insulin *response*: read carefully!).

```r
ggplot(diabetes, aes(x = group, y = sspg)) + geom_boxplot()
```



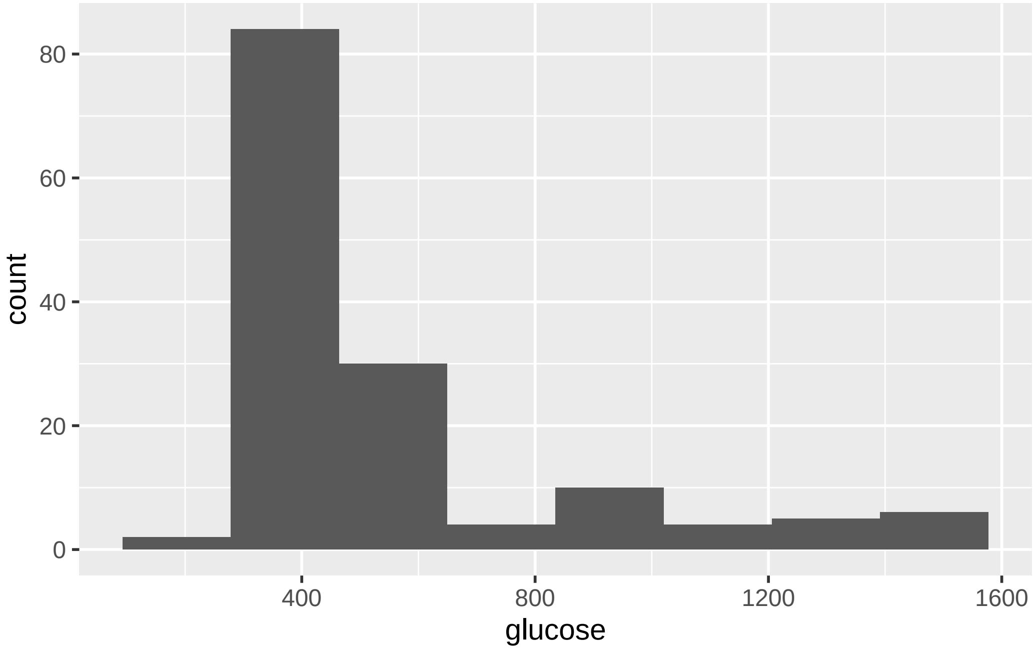
The standard form of the boxplot has the boxes up and down the page, so this is what you should draw unless you say otherwise; for example saying "I prefer the boxes on a boxplot to go across the page" would justify switching the `x` and `y` around, but you need to demonstrate some kind of awareness of the roles of `x` and `y` in this graph.

If you had a different answer to the previous question, then give code to draw the graph(s) you named there. The grader is looking for consistency with the previous question: that is to say, if your answer here would be correct if your previous answer had been correct *and* you have not made this question easier than it should be, you can get full marks here.

(6) (3 points) A graph is shown in Figure 8. What code was used to draw this graph?

This code, precisely:

```
ggplot(diabetes, aes(x = glucose)) + geom_histogram(bins = 8)
```



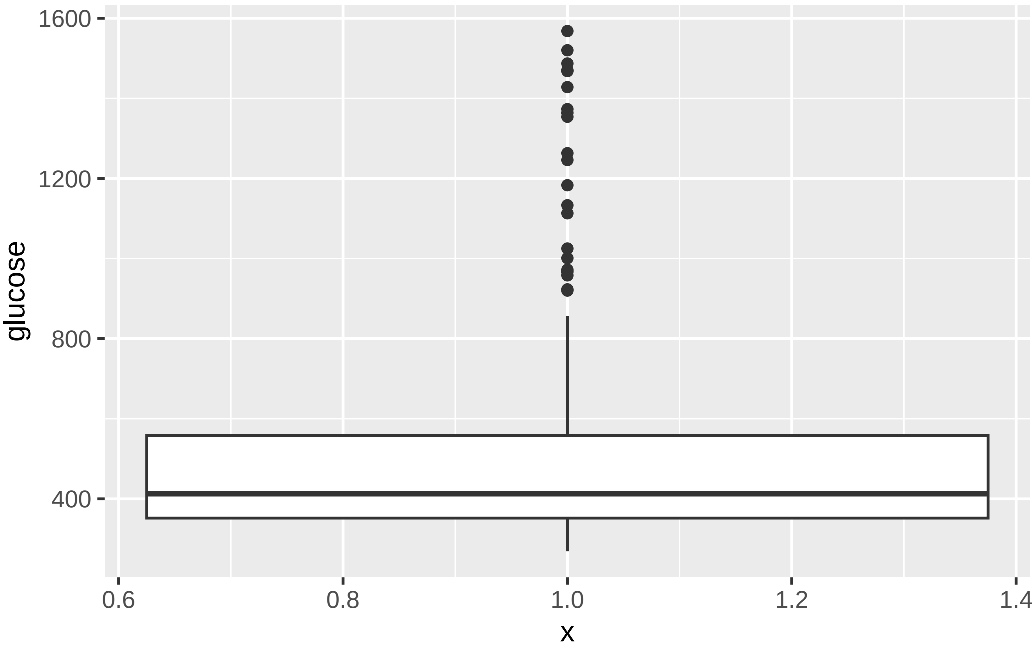
The graph is evidently a histogram of the quantitative variable `glucose` in the dataframe `diabetes`; remember that a histogram always needs a number of bins, so you should count up how many bins appear in my histogram (8).

Two points if you have a histogram with absent or obviously wrong number of bins. 2.5 if you seem to have miscounted the number of bins (if you are off by one). One if you are missing something substantial but you are obviously on the way to making a histogram (for example, you have the `x` variable wrong).

(7) (2 points) Sketch a one-sample boxplot of the data shown in Figure 8.

I, of course, am in the privileged position of being able to actually draw it:

```
ggplot(diabetes, aes(x = 1, y = glucose)) + geom_boxplot()
```



The features I am looking for include

- a vertical scale that puts the median somewhere near 400 (we know there are 145 observations altogether and over half of them are in that tall bar on the histogram)
- a relatively small box because most of the observations are between about 300 and 600. I'm not particular about where you think the quartiles are.
- lots of upper-tail outliers (boxplot-outliers) because the IQR will be small. I said "sketch", so some kind of artistic expression of the many outliers is fine (even the word "many"). Remember that anything more than 1.5 times the IQR above Q3 is an outlier on a boxplot, and the IQR will be pretty small. Thus the upper whisker cannot be any longer than 1.5 times the IQR.

A rather confusing thing is that there are no actual real "outliers" on the histogram: that is, no points very different from the rest of the distribution. You might be tempted to draw a long upper whisker and leave it at that, but you need to remember how an outlier is defined *on a boxplot* to get the boxplot right.

One point for a boxplot with a long upper whisker and no outliers; at the grader's discretion, 1.5 if the upper whisker is too long or the outliers don't go up far enough; in addition, minus a half point if the vertical scale is missing or obviously in the wrong place.

I tried to give you enough space on the page for a decent-sized sketch; if yours seems to come out bigger, you can try to scale it somehow, or you can use the blank last page of the exam.

(8) (3 points) Another graph is shown in Figure 9. What kind of graph is it? What are two distinct things that you learn from this graph?

This is a bar chart (one easy point).

As to what you learn, I would like to see something that compares the bars (and what it means), and something about the height of one of the bars, and what that means. For example, the other two points could be for something like these:

- There are more patients in the `normal` group than in either of the other two groups (because its bar is taller)
- There are about 75 patients in the `normal` group (or about 37 in the `chemical` group or about 32 in the `overt` group).

You might also reason that there are more patients in the `normal` group than in the other two groups *put together*, perhaps because if you stack the `overt` bar on top of the `chemical` bar it's still not as tall as the `normal` bar, or perhaps because there are about 75 `normal` patients and only 145 in total, so the others add up to about 70.

Other reasonable conclusions, in the grader's opinion, are fine, but giving two (or three) heights of bars is only *one* distinct thing, so it is only one point out of two.

Extra: make sure you can tell the difference between the histogram in Figure 8 and the bar chart in Figure 9, the key thing being what kind of variable is on the x-axis. A histogram has a quantitative variable on the x-axis, and any possible value of the quantitative variable must be able to appear on the graph, so the bins *join up* on Figure 8 (and other histograms). A bar chart has a *categorical* variable on the x-axis; the only possible values of it are one of a list of categories (the three `groups` in Figure 9). A person cannot be "in between" normal and chemical, so the bars have gaps between them.

Diabetes revisited

The following questions also use the dataframe diabetes, some rows of which are shown in Figure 7. This contains five numerical measurements made on 145 non-obese adult patients classified into three groups.

(9) (3 points) What code would find the number of observations, the median, and the interquartile range of values of fpg for each group?

This is a standard group-by and summarize:

```
diabetes %>%
  group_by(group) %>%
  summarize(n = n(), med_fpg = median(fpg), iqr_fpg = IQR(fpg))
```

```
# A tibble: 3 x 4
  group        n med_fpg iqr_fpg
  <fct>    <int>   <dbl>   <dbl>
1 normal      76    90      11
2 chemical    36    99.5    15.2
3 overt       33   203     131
```

Make an effort to give your summaries reasonable names, and don't forget that IQR is uppercase for the function that calculates interquartile ranges.

Expect to lose a point if something important is missing (the group_by or one of the three things in summarize) and a half point for a small error like using iqr instead of IQR. What the grader is shooting for is a mark that reflects how much of the answer you successfully got.

(10) (2 points) What code would display (all the rows of) all the columns that have the letter "g" in their names somewhere, either uppercase or lowercase, without naming or numbering any columns?

This is select with a select-helper. The right one this time is contains:

```
diabetes %>%
  select(contains("g"))
```

```
# A tibble: 145 x 4
     fpg glucose  sspg group
   <int>   <int> <int> <fct>
 1    80     356    55 normal
 2    97     289    76 normal
 3   105     319   105 normal
 4    90     356   108 normal
 5    90     323   143 normal
 6    86     381   165 normal
 7   100     350   119 normal
 8    85     301   105 normal
 9    97     379    98 normal
10    97     296    94 normal
# i 135 more rows
```

Remember that the select-helpers match both uppercase and lowercase, so you don't need to say anything special about case to answer the question. If you say something like `ignore.case = TRUE`, you reveal that you don't know that this is the default and doesn't need to be specified (so minus a half point).

This actually also works, and thus is full points:

```
diabetes %>%
  select(matches("g"))
```

```
# A tibble: 145 x 4
     fpg glucose  sspg group
   <int>   <int> <int> <fct>
 1    80     356    55 normal
 2    97     289    76 normal
 3   105     319   105 normal
 4    90     356   108 normal
 5    90     323   143 normal
 6    86     381   165 normal
 7   100     350   119 normal
 8    85     301   105 normal
 9    97     379    98 normal
10    97     296    94 normal
# i 135 more rows
```

The thing inside `matches` is actually a regular expression that uses none of the special regular expression stuff we saw in lecture like `.` or `^` or `$` because the match with the letter `g` can be anywhere in the column name (not only at the end like the `.t$` we saw in lecture).

Roughly speaking, one point each for recognizing that you need `select` and either `contains` or `matches`, and then minus a half point for small errors.

(11) (2 points) What code did I use to make the display in Figure 7?

This is 20 randomly chosen rows, and the way you randomly choose rows is with `slice_sample`:

```
diabetes %>%
  slice_sample(n = 20)
```

```
# A tibble: 20 x 6
      rw   fpg glucose insulin  sspg group
   <dbl> <int>   <int>   <int> <int> <fct>
 1  1.08   105     527     480   233 chemical
 2  0.89    85     373     174    78 normal
 3  0.74   346    1568      15   253 overt
 4  0.71    75     352     169    32 normal
 5  0.81    80     356     124    55 normal
 6  0.76    90     353     263   165 normal
 7  0.95    95     347     184    91 normal
 8  0.99    97     379     142    98 normal
 9  0.9    213    1025      29   209 overt
10  1.07   124     538     460   320 overt
11  0.91   180     923      77   150 overt
12  1.02    88     439     208   244 chemical
13  0.98   130     670      44   167 overt
14  0.74    93     318      73    42 normal
15  0.87    94     313     200   233 normal
16  0.95    96     356     112    73 normal
17  0.83    86     319     144   138 normal
18  0.91   103     537     622   264 chemical
19  0.91   100     350     221   119 normal
20  1.19    85     425     143   204 chemical
```

The actual rows displayed this time are different from the ones in Figure 7, because different random numbers are used, but that is of no concern to you: the way to display the dataframe according to the specifications in the caption to the Figure is as above. The `n = 20` gives you the 20 rows.

The points here are one for `slice_sample` and one for `n`. Minus a half point per small error apart from that.

(12) (2 points) For only the patients whose `glucose` value is more than 1000, what code will display all their information?

This is selecting rows (patients) that satisfy a condition, so `filter` is needed:

```
diabetes %>%
  filter(glucose > 1000)
```

```
# A tibble: 16 x 6
      rw   fpg glucose insulin  sspg group
   <dbl> <int>   <int>   <int> <int> <fct>
 1  0.92   300    1468      28   455 overt
 2  0.86   303    1487      23   327 overt
 3  0.83   280    1470      54   382 overt
 4  0.85   216    1113      81   378 overt
 5  0.92   303    1364      42   346 overt
 6  0.86   275    1373      45   300 overt
 7  0.9    260    1133     118   300 overt
 8  1.16   233    1183      73   458 overt
 9  0.93   213    1001      42   297 overt
10  0.85   330    1520      13   303 overt
11  1.06   339    1354      10   450 overt
12  1.03   265    1263      83   413 overt
13  1.05   353    1428      41   480 overt
14  0.9    213    1025      29   209 overt
15  1.11   328    1246     124   442 overt
16  0.74   346    1568      15   253 overt
```

There is no `select` needed because "all their information" here means "all the columns" for those patients. One point each for the `filter` and the right logical condition.

Do not put the 1000 in quotes. It is a number, and quotes are for text. What will actually happen if you do?

```
diabetes %>%
  filter(glucose > "1000")
```

```
# A tibble: 145 x 6
      rw   fpg glucose insulin  sspg group
   <dbl> <int>   <int>   <int> <int> <fct>
 1  0.81    80     356     124    55 normal
 2  0.95    97     289     117    76 normal
 3  0.94   105     319     143   105 normal
 4  1.04    90     356     199   108 normal
 5  1       90     323     240   143 normal
 6  0.76    86     381     157   165 normal
 7  0.91   100     350     221   119 normal
 8  1.1     85     301     186   105 normal
 9  0.99    97     379     142    98 normal
10  0.78    97     296     131    94 normal
# i 135 more rows
```

There are now 145 rows (that is, all of them) instead of only 16. What happens if you do it this way is that the values in `glucose` are converted to *text*, in order to compare them with the 1000-as-text, and text values are compared in alphabetical order[1] by comparing the first character in `glucose` with the first character of `1000` (1), and if that is the same, comparing the second character, and so on. Hence 356 as text is "greater" than 1000 because 3 is greater than 1, and 102 is "greater" than 1000 because the first two characters are the same and, after that, 2 is greater than 0.

(13) (3 points) What code will display the `sspg` values for patients in the `overt` group that have the largest 8 values of `insulin` (but not display these for patients in any other group, or display any other variables)?

There are some things to consider here, one point each:

- choose the patients in the `overt` group only (`filter`)
- find the largest 8 values of `insulin` for those patients (`slice_max` or `arrange` followed by `slice`)
- display only the values of `sspg` for those patients (`select`).

[1] Strictly, in order of ASCII code, one character at a time.

The steps need to be in that order: the filtering has to happen before you find the largest 8 values (otherwise you are finding the 8 largest values of the wrong thing, such as the 8 largest `insulin` values for *all* the data, not just the `overt` group), and the `select` has to happen *after* you have found those 8 largest values (because the thing you are displaying is not the thing you are finding the 8 largest values of):

```
diabetes %>%
  filter(group == "overt") %>%
  slice_max(insulin, n = 8) %>%
  select(sspg)
```

```
# A tibble: 8 x 1
   sspg
  <int>
1   320
2   220
3   279
4   209
5   357
6   310
7   324
8   351
```

Again, minus a half point for each small error, and, here, minus one for getting the steps out of order. I'm willing to tolerate displaying the `group` column as well (which you might want to do in order to check your work), as long as you `select` something at the end (or are very careful about what you `select` if you do it earlier).

I just thought of another way that might work:

```
diabetes %>%
  group_by(group) %>%
  slice_max(insulin, n = 8) %>%
  filter(group == "overt") %>%
  select(sspg)
```

```
Adding missing grouping variables: `group`
```

```
# A tibble: 8 x 2
  group  sspg
  <fct> <int>
1 overt   320
2 overt   220
3 overt   279
4 overt   209
5 overt   357
6 overt   310
7 overt   324
8 overt   351
```

What this does is to group the patients by `group`, then find the largest 8 values of `insulin` *in each group*, then get only the `overt` patients, and finally display the column we want. (These last two steps can be the other way around). It actually doesn't quite answer the question because the grouping variable is displayed as well, but I don't expect you to know that this will happen, so if you do it this way, I am also happy.

In principle, you get full credit for something that will work, but don't expect the grader to spend more than a few seconds trying to disentangle your code if it is something different from these. If it is not obvious that it will work, you may lose out on some points that you might otherwise have had.

Heights of males

Figure 10 shows the heights of a sample of 100 males, measured in inches. Use these data to answer the following questions.

(14) (2 points) A histogram of the heights is shown in Figure 11. Based on this histogram and anything else you have learned about these data so far, explain briefly why it reasonable to use t procedures (test or confidence interval) to make inferences about the population mean height of males.

The usual two issues, one point for addressing each:

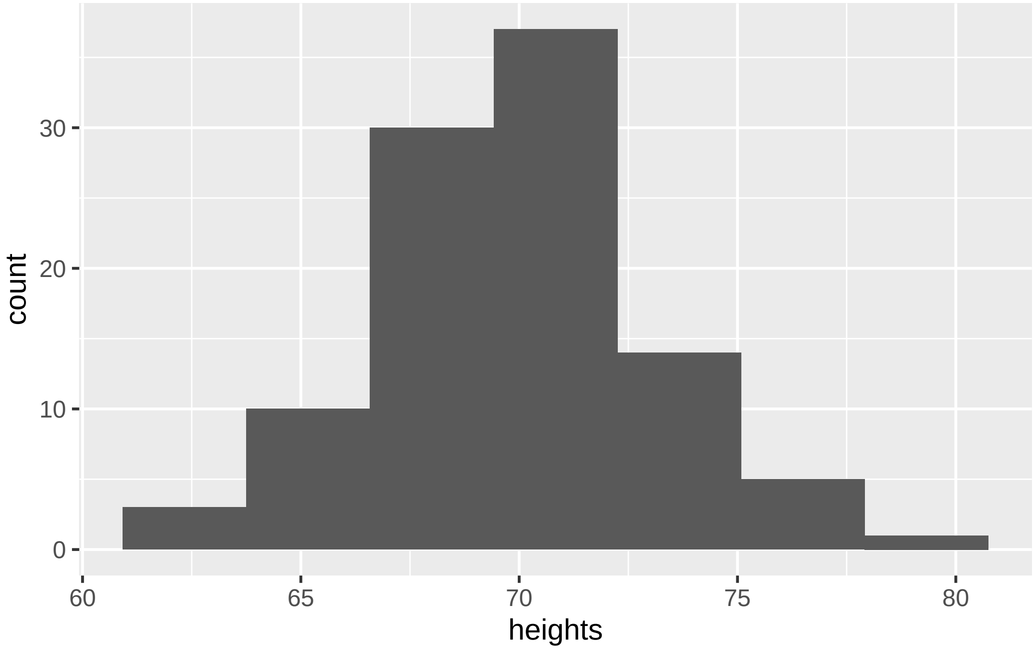
- normality: the distribution looks slightly right-skewed, since the right tail is longer than the left one. You need to capture the non-symmetry somehow; saying that there are too few observations in the lower tail would also do it.
- the sample size of $n = 100$ is large in Central Limit Theorem terms.

Hence, whatever non-normality you saw will be easily taken care of by the large sample, and thus we should have no problems using t-procedures.

The wording of the question says that it is not correct to look at the bootstrap sampling distribution of the sample mean (yet: that is coming later).

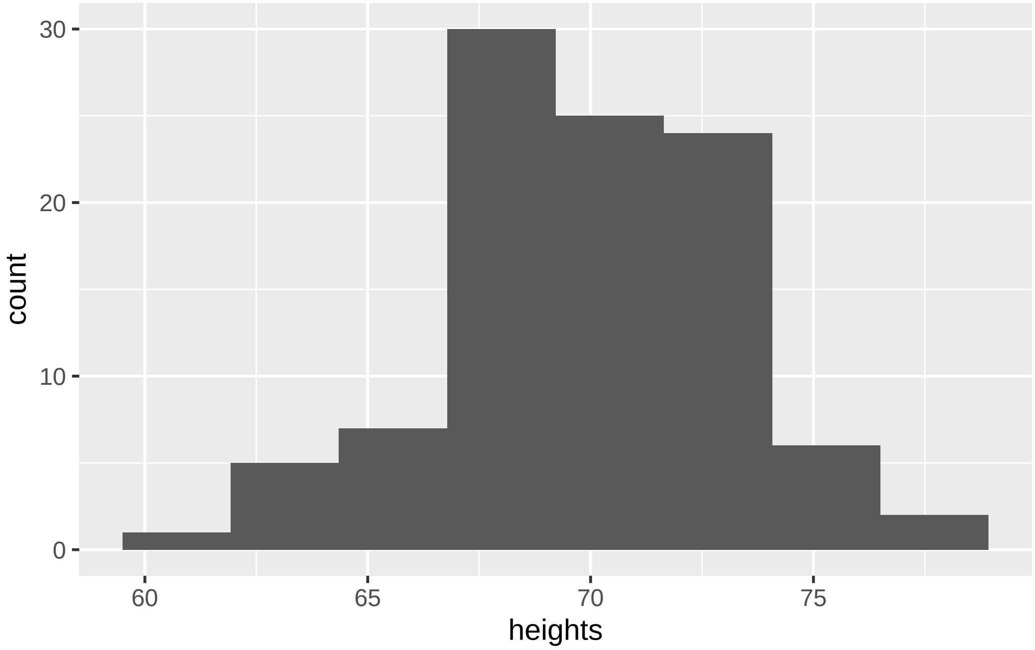
Extra: the appearance of the histogram seems to depend substantially on the number of bins. Here is 7:

```
ggplot(male_heights, aes(x = heights)) + geom_histogram(bins = 7)
```



which looks pretty bell-shaped, and here is 8:

```
ggplot(male_heights, aes(x = heights)) + geom_histogram(bins = 8)
```

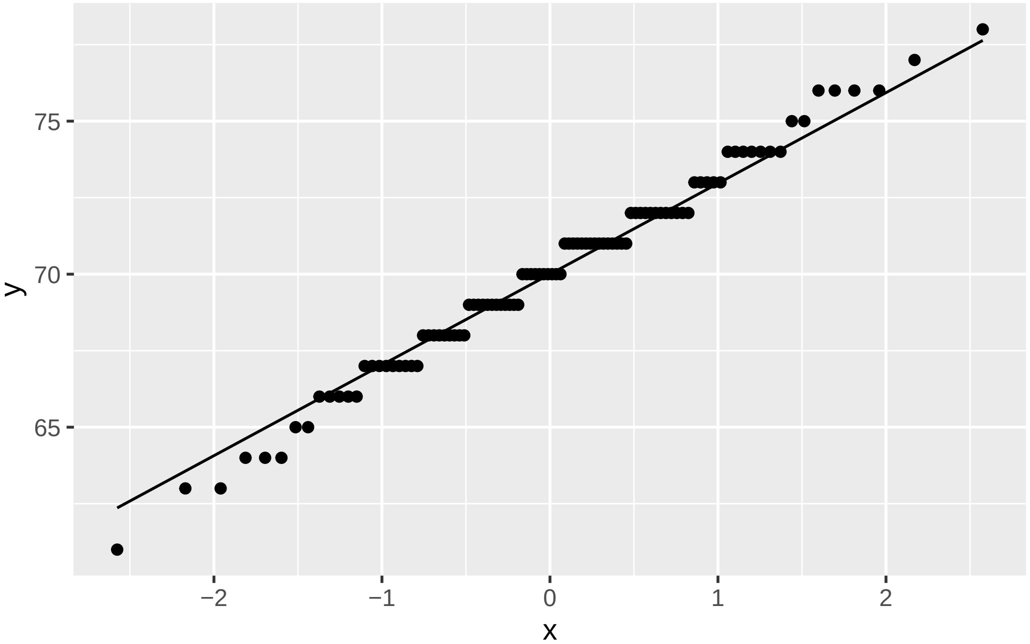


which appears to be more or less symmetric but to have a “flat top” rather than the nice peak that you would expect a normal distribution to have. I didn’t want to give you this one to interpret, and the 7-bin one looks close to normal already, so you got the 9-bin histogram.

I suspect the reason for the number of bins being important is that the heights were measured to the nearest inch, and with this kind of number of bins, the bins might contain different numbers of whole-number inches. For example, on the 8-bin histogram, the tallest bin appears to contain 67, 68, and 69 inches (three different heights), and the short bin next to it seems to contain only 65 and 66 inches (only two different heights).

I think a normal quantile plot actually shows the situation more clearly:

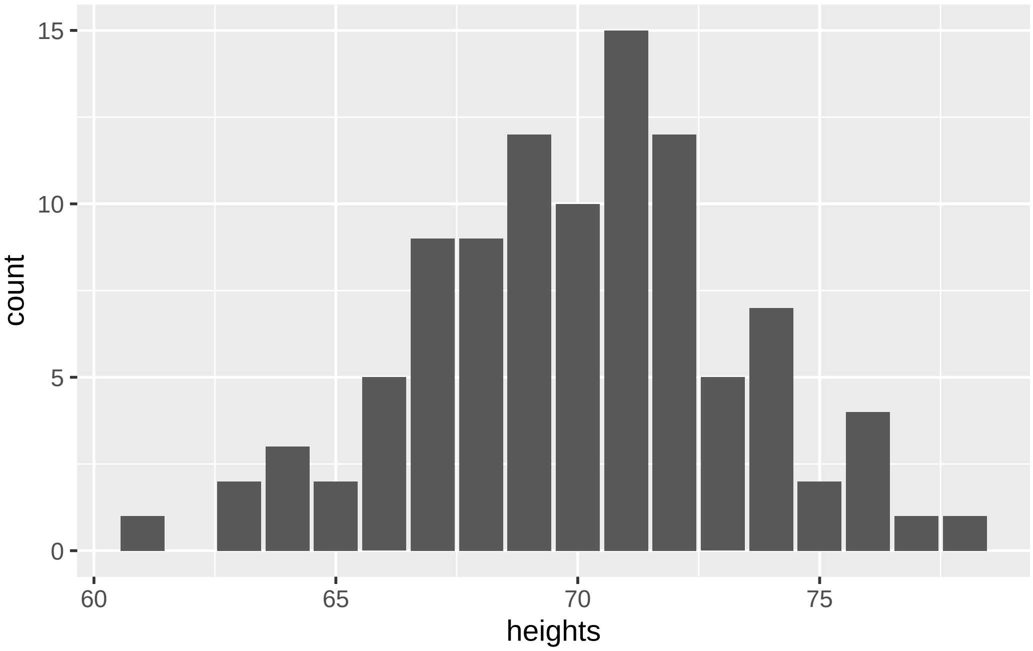
```
ggplot(male_heights, aes(sample = heights)) + stat_qq() + stat_qq_line()
```



The horizontal streaks of points show multiple males with the same height (to the nearest inch), but apart from that, all the points are close to the line, so normality is actually already good, and with a sample size of $n = 100$ we are going to have no problems at all.

Here we have a “discrete” distribution, where only certain values are observable rather than everything over a continuum. If the heights had been measured more accurately (like, say, to the nearest tenth of an inch), it would have made sense to treat them as continuous, where any value is observable. But the heights we have can only be whole numbers, in the same way that if you toss a coin 10 times, you can only get a whole number of heads. In that case, you want a histogram where the bins contain exactly one value each. You can do that most easily by pretending that your discrete height is actually categorical and drawing a bar chart:

```
ggplot(male_heights, aes(x = heights)) + geom_bar()
```

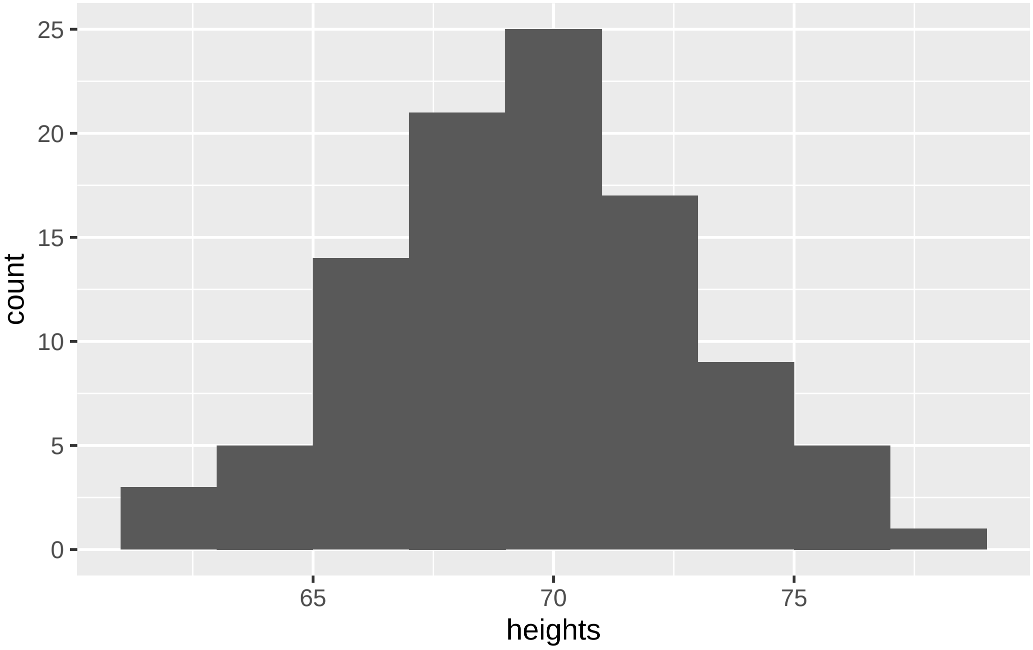


The only problem with this is that there are now too many bins, but if you're willing to look past the irregularities in the bar heights, this is not far from being bell-shaped.

Alternatively, if you're willing to scour the help for `geom_histogram`, you can find out how to put the bins in the right places. I want to have the bins *centred*[2] at a whole number of inches and be two inches wide (based on my experience with the bar chart: I want fewer bins, so I have to make each one wider):

```
ggplot(male_heights, aes(x = heights)) +
  geom_histogram(center = 60, binwidth = 2)
```

[2] The option in `geom_histogram`, curiously enough, seems to require the American spelling.



Now that I have the same numbers of whole-number inches in each bin (two of them), I do indeed have pretty much a bell shape.

Historically, the name "normal distribution" came from "normal variation" in measurements on (healthy) humans, and height is one of the things that typically varies according to a normal distribution.

(15) (4 points) In 2016, the mean height for American males was 69.1 inches. Using Figure 12, what do you conclude? (You may assume that the data in Figure 10 were collected this year.) Make sure you show your thought process clearly.

We are testing a null hypothesis that the population mean height is 69.1 inches against an alternative that it is not equal to this. The P-value is 0.0057, which is smaller than 0.05, so we can reject the null and conclude that the mean is not equal to 69.1: that is, we have evidence that the mean height of American males has changed between 2016 and now.

The grader is looking for (one point each):

- stating or strongly implying that you know what hypotheses you are testing (if it is clear from your conclusion, in the grader's judgment, that you know what hypotheses you are testing, you are good here, but if you mess up the conclusion in a way that indicates that you don't know what hypotheses you are testing, expect to lose the point here as well)

- stating the P-value to about 4 decimals; two or three significant figures is a good rule of thumb for P-values)
- a decision about rejecting the null
- a conclusion in the context of the data.

It may look (for example, from the confidence interval) as if that the mean height has *increased* since 2016, but the only justification for *testing* that is if you had a reason to suspect that *before* looking at the data (in which case you would do a one-sided test). Here, we (apparently) didn't, so we have to test with the hypotheses as given.

All the confidence interval tells you (in this context) is that the P-value of the two-sided test will be less than 0.05, which doesn't tell you anything about the strength of evidence, or what somebody with a different α than you should conclude. If you try to do the test using the confidence interval, you will lose (at least) one point for not stating the P-value accurately.

(16) (3 points) What do you conclude from the output of Figure 13? (The code that produced the output is shown above the output.) Your answer should make it clear that you know what the output is.

This is the bootstrap sampling distribution of the sample mean. It is, according to the plot, very close to a normal distribution, and so provides further support for the decision to do a *t*-test.

One point each for:

- saying that it is the bootstrap sampling distribution of the sample mean
- saying that it is close to normal
- saying that this supports doing a *t*-test (or saying that the earlier decision to do one was actually correct, or it supports your earlier conclusion to do one).

Mentioning the bootstrap part is important, so make sure you do that.

(17) (2 points) Appropriately state a 95% confidence interval for the mean height of all males this year.

This is the interval given in Figure 12. "Appropriately state" means to *round it off* suitably. Figure 10 shows that the heights were given to the nearest inch, so your statement of the confidence interval should include one or at the most two decimal places. Hence the 95% confidence interval is 69.4 to 70.7 inches.

One point only for giving the confidence interval to more decimals than that (that is, for giving three or more decimal places). You need to think of your reader.

Lower back pain

In many industrial settings involving physical labour, lower back pain (LBP) is a serious health problem. A study measured the lateral range of motion (measured in degrees) of workers in a steel factory, some of whom had a history of lower back pain and some of whom did not. The researchers suspected that the workers with no history of lower back pain would have a larger range of motion. Some of the data are shown in Figure 14. There were 28 workers in the no-LBP group and 31 in the LBP group.

(18) (3 points) A boxplot is shown in Figure 15. On the basis of the boxplot, the researchers decided to run a *t*-test to compare the mean lateral range of motion between workers with a history of low back pain and those without. What do you think the researchers' reasoning was?

We need both distributions to be normal enough given their sample sizes. Both distributions are already close to symmetric, with the LBP group having a low outlier that is only just off the end of the whisker. Both sample sizes are around 30, which will offer a lot of help in overcoming this slight non-normality. Thus, both groups are close enough to normal given their sample sizes, and a *t*-test is fine.

Comment on:

- the nature of any non-normality as shown by the boxplot
- the sample sizes
- a conclusion about *both* groups being close enough to normal.

Extra: these are actually made-up data. In my source, only the sample summary statistics were given, so I generated some random normal data from distributions with these sample sizes, means and SDs.

These are the summary statistics:

```
# A tibble: 2 x 4
  Condition Sample.size Sample.mean Sample.SD
  <fct>           <int>       <dbl>     <dbl>
1 No LBP             28        91.5       5.5
2 LBP                31        88.3       7.8
```

and this generates random samples from normal distributions with these means and SDs (note the use of `rowwise`):

```r
lbp0 %>%
  rowwise() %>%
  mutate(Rom = list(rnorm(Sample.size, Sample.mean, Sample.SD))) %>%
  unnest_longer(Rom) %>%
  select(Condition, Rom)
```

```
# A tibble: 59 x 2
   Condition   Rom
   <fct>     <dbl>
 1 No LBP    100.
 2 No LBP     87.4
 3 No LBP     90.0
 4 No LBP     87.7
 5 No LBP     92.7
 6 No LBP     95.4
 7 No LBP     85.6
 8 No LBP     95.9
 9 No LBP     91.5
10 No LBP     97.5
# i 49 more rows
```

We had a question in one of the lectures about how you display all the values in a list-column: the `unnest_longer` is the answer, expanding the columns downwards and duplicating the `Condition` labels as needed.

So these populations actually *are* normal and we shouldn't need any help from the Central Limit Theorem at all. Any non-normality you saw in the boxplot was actually chance, and whatever you saw was certainly going to be taken care of by the sample sizes. (The outlier on the boxplot was a "boxplot-outlier", a value that happened to be smaller than Q1 minus 1.5 times the IQR, but it was not smaller by much, and thus not a "real" outlier, which would be a value clearly smaller than the others.)

(19) (3 points) What code would carry out your preferred *t*-test in this situation? Justify any choices you make.

This is a two-sample *t*-test, since we are comparing the mean `Rom` values between the `LBP` and `No LBP` groups. The choice you need to make is between the Welch and pooled tests. I think you can justify either one, but you need to have a justification of some sort, something along one of these lines:

- the two boxes are of different height, so use Welch

- the two boxes are of similar height, so do a pooled test
- the whiskers in the LBP group are longer, so this group has bigger spread (despite the box), so use Welch
- it is difficult to tell whether the two groups have the same or different spreads, so be safe and use Welch. (If you go this way, you need something beyond "Welch to be safe").

Code-wise, some code that:

- matches your decision about whether to use Welch or pooled
- does an appropriate one-sided test (based on the researcher's suspicion). This requires clear thinking: LBP is first alphabetically (clue: the left one on the boxplot), so the alternative needs to say how LBP compares to No LBP *in that order*, that is, `less`:

```
t.test(Rom ~ Condition, data = lbp, alternative = "less")
```

```

    Welch Two Sample t-test

data:  Rom by Condition
t = -3.0308, df = 54.066, p-value = 0.001869
alternative hypothesis: true difference in means between group LBP and group No LBP is less
95 percent confidence interval:
      -Inf -2.191263
sample estimates:
   mean in group LBP mean in group No LBP
            87.30550             92.19869
```

if you preferred Welch, and

```
t.test(Rom ~ Condition, data = lbp, alternative = "less",
       var.equal = TRUE)
```

```

    Two Sample t-test

data:  Rom by Condition
t = -2.9793, df = 57, p-value = 0.00212
alternative hypothesis: true difference in means between group LBP and group No LBP is less
```

```
95 percent confidence interval:
      -Inf -2.147021
sample estimates:
   mean in group LBP mean in group No LBP
            87.30550            92.19869
```

if you preferred the pooled test.

As ever, you won't know what the output will be, so if I ask for code, all I need is the code and not the output.

Two points for code to run a one-sided two-sample *t*-test of the type you expressed a preference for, and one point for a reasonable justification of your choice of Welch or pooled test. Only one point for the code if it does a two-sided test.

Extra: as is often the way, the two versions of two-sample test give very similar P-values (just either side of 0.002) and the same conclusion. So in actual fact it doesn't matter which test you preferred, but in this course you need to have a reason for running one test rather than the other.

(20) (2 points) The P-value for your test is 0.0019. What do you conclude, in the context of the data?

The null hypothesis is that mean range of motion is the same for the people that had a history of low back paint and the people that did not, and the alternative is that the mean range of motion is greater for those that did not. The P-value of 0.0019 is smaller than 0.05, so reject the null hypothesis and conclude that the mean range of motion is greater for those with no history of low back pain.

You need to have a conclusion about low back pain in words, and you need to *compare* the P-value to 0.05 (or suitable value). You should at least make it clear that you know what the hypotheses are, so that you can make an intelligent choice between them based on the P-value.

If your code in the previous question was for a two-sided test, then your conclusion here must be two-sided: that is, you need to conclude that there is *a difference* between the mean range of motion for the two groups. (You need to be consistent with yourself.) If you are paying attention, you will realize that this is not what the researchers were interested in assessing, and you should take the invitation to go back and amend your code and conclusion. If you previously did a (correct) one-sided test and you conclude here "there is a difference", you are making an additional error and should expect to lose some points for it here.

Mystery code

(21) (4 points) Some code and output is shown in Figure 16. What precisely do you conclude from the output?

The answer you need to provide (as briefly as possible) is this:

- The true population is normal with mean 40 and SD 15.
- The sample size is 50.
- The power to (correctly) reject $H_0 : \mu = 45$ in favour of $H_a : \mu \neq 45$ is (estimated as) 0.620, where μ denotes the population mean.

Your answer needs to contain the true population mean and SD, the sample size, and the null and alternative hypotheses being tested. You can say it all in one shot if you are organized enough:

- When the population mean is 40, the population SD is 15, and the sample size is 50, the power to (correctly) reject $H_0 : \mu = 45$ in favour of $H_a : \mu \neq 45$ is (estimated as) 0.620, where μ denotes the population mean.

Marking guideline: one point each for (i) the true distribution, (ii) the sample size, (iii) the hypotheses, (iv) the word "power" (or equivalent) and its value here. You need all four of those, because if any of the first three were different, the power would be different. I would accept "the estimated probability of a type II error is 0.380" for the last one. Thus, saying "the power is 0.620" is only one point by itself, because what matters is the power *under what conditions*, which are the other three things.

The code has some hints:

- it is some kind of simulation (suggested by the first line)
- it is a simulation *of power* (the last line, where we are counting P-values no bigger than 0.05)
- the true distribution and sample size are in `rnorm` on the third line
- the null hypothesis is given by the value of `mu =` on the fourth line, and the lack of an `alternative` means that H_a is two-sided.

There were a lot of good answers to this one.

Planning a comparative experiment

(22) (3 points) Two different heat treatments will be compared in their effects on the strength of steel ingots. (An ingot is defined as “a mass of metal cast into a convenient shape for storage or transportation.”) Eight ingots will be cast using each treatment. In the units used, strength has a standard deviation of 0.5 units for each treatment. We want to know how likely it is that a null hypothesis of equal strengths will be rejected if one of the treatments has a mean strength 0.8 units greater than the other, using an appropriate two-sided *t*-test. What code will calculate this? (You may assume that the distributions of strengths are normal.)

This is a power *calculation* (not estimation), and all the assumptions are met for the use of `power.t.test` (normal distributions, equal SDs, equal sample sizes):

```
power.t.test(n = 8, delta = 0.8, sd = 0.5, type = "two.sample")
```

```

     Two-sample t test power calculation 

              n = 8
          delta = 0.8
             sd = 0.5
      sig.level = 0.05
          power = 0.8447929
    alternative = two.sided

NOTE: n is number in *each* group
```

There is no need to put in an `alternative`, since `two.sided` is the default.

I gave you the true difference in means (“if one of the treatments has a mean strength 0.8 units greater than the other”), which you use for `delta`, and the sample sizes and SDs for the two treatments (the same, as they need to be). Make sure you say that this is a two-sample test using `type`, or else it will do a one-sample test (not appropriate for comparing two treatments with each other).

Expect to lose one point for each thing missing or incorrect, down to a minimum of 1 point if you have *something* correct. The grader has the discretion to deduct half a point for what they consider to be a “small” error. `delta` in this case can be negative, because the test is two-sided. The order of the inputs is not important as long as they have the right names. If you don’t have names, they *must* be in the order `n`, `delta`, `sd`, and it is in that case an error not to name `type` (because that is later in the list of inputs).

(23) (2 points) What *changes* would you make to your code of the previous question to find the required sample size to obtain a power of 0.90?

Two things, a point each:

- remove the `n = 8` (sample size)
- replace it with `power = 0.90` (the power you want to achieve).

The sample size is then the missing thing, so that is what will be calculated:

```
power.t.test(power = 0.90, delta = 0.8, sd = 0.5, type = "two.sample")
```

```

     Two-sample t test power calculation 

              n = 9.283698
          delta = 0.8
             sd = 0.5
      sig.level = 0.05
          power = 0.9
    alternative = two.sided

NOTE: n is number in *each* group
```

It is better (and quicker) to say what *changes* you are making rather than to write the whole thing out again. Also, by writing out the whole thing you are wasting the grader's time because they have to look through everything you wrote to see whether it contains the two things they are looking for (and whether the rest of it is still correct).

Extra: you might be surprised that the required sample size is so small. This is because the strength measurements are not very variable (on this scale): the population SDs are small. Even though the true difference in means is also small, it is in effect size[3] terms very large:

```
0.8 / 0.5
```

```
[1] 1.6
```

[3] Statisticians don't usually talk about effect sizes, but people in the social sciences like them.

(this is Cohen's D for the (pooled) two-sample t-test, for which 0.8 is usually considered large.) Hence we did not need a very large sample size in order to have a good chance of detecting it.

(24) (2 points) The output from your changes in the previous question is shown in Figure 17. What does this output tell you about the total number of observations you need to make?

In order to achieve power 0.90 (to detect a 0.8 unit difference between the two treatments), we need 10 observations on each treatment (rounding up, to get power at least 0.90). That is, we need $10 + 10 = 20$ observations altogether.

A point each for "10 observations" and "for each treatment". Alternatively, if your answer is "20 observations" with a good reason for why it's 20 and not 10, that will also do. An answer of "20 observations" without a reason is only one point (and even that is generous, really).

If you take the 9.28 observations per group, double it first and then round up, you'll get $18.56 \rightarrow 19$ observations altogether, but this is not right, because you cannot share those equally between the two treatments (and `power.t.test` always assumes you have the same number of observations in each group). You might *assert* that having 10 observations in one group and 9 in the other will give you the power you want, but you have no way to be sure about that using `power.t.test`. If you want to demonstrate that group sizes of 10 and 9 will work, you'll have to do a simulation.

Wisconsin Card Sorting Test

The Wisconsin Card Sorting Test is widely used by psychiatrists with patients who have a brain injury or other mental illness. Fifty patients were given this test, with some of the data shown in Figure 18. The dataframe is called `wcst`. A higher `score` is better.

(25) (2 points) A histogram of the scores is shown in Figure 19. The distribution is skewed, but the sample size is fairly large. What would you calculate and make a graph of in order to determine whether the sample size is large enough to enable you to use a t-test? (No explanation needed.)

The bootstrap sampling distribution of the sample mean.

That's all you need, but I think you need all those words (apart possibly from the "the"s). The three key concepts are:

- bootstrap (a half point)
- sampling distribution (a half point)
- of the sample mean (one point).

It is much easier to write those words than to give me code for the corresponding thing here, but if you want to do that and can get it right, go for it (but see below). Likewise, describing what you would do, if correct and complete, is also good even without using all those words.

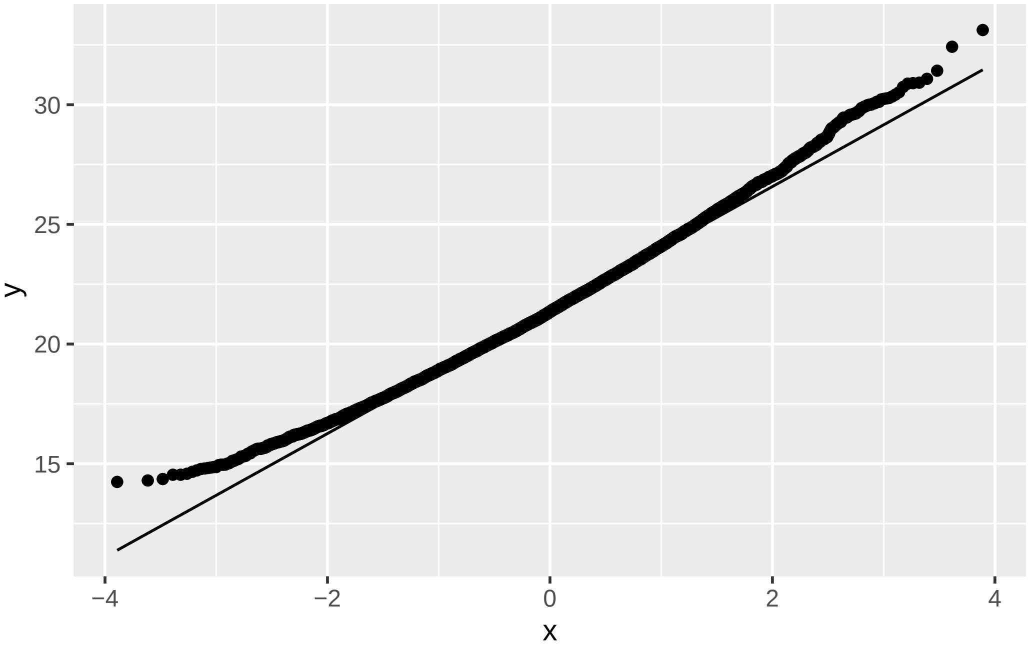
Doing a normal quantile plot or histogram *of the data* is not helpful, because it tells you how normal your data are, and you'd have to make some additional decision about whether the sample size is large enough given whatever non-normality you see. That is to say, a normal quantile plot *of the data* does *not* help you assess whether your sample size is big enough. This kind of answer is worth a half point only.

For those who gave me code: you can expect that if I specifically want code, I will ask for it. Otherwise an answer in words is going to be best (which also shows that you know what is going on). If you give me code, it makes me wonder whether you are in fact an automaton that adapts what seems to be the most relevant code from your notes.

No points if you propose a "normal quantile plot" or some other kind of plot without saying what you propose to make a plot of.

Extra: After the fact, you might be curious about whether the sample size *is* normal enough in this case:

```
tibble(sim = 1:10000) %>%
  rowwise() %>%
  mutate(my_sample = list(sample(wcst$score, replace = TRUE))) %>%
  mutate(my_mean = mean(my_sample)) %>%
  ggplot(aes(sample = my_mean)) + stat_qq() + stat_qq_line()
```



I see a definite right-skewed pattern here too, even though the points are close to the line, so my take is that the sample size is not quite large enough. (I would be less concerned if the points were this close to the line without this clear of a pattern.)

(26) (3 points) Based on what the researchers saw in the graph of what you named in the previous part, they decided to run a sign test. The median score in the general population is 25. What code would run a suitable test to see whether the median score in this population is less than 25?

This is a sign test with a null median of 25. The code requires a little care: there is no `alternative` option; instead, the input is only the dataframe, column, and null median:

```
sign_test(wcst, score, 25)
```

```
$above_below
```

```
below above
   35    14

$p_values
  alternative     p_value
1       lower 0.001900827
2       upper 0.999298656
3   two-sided 0.003801654
```

Minus one if you put an `alternative` in, and if you get the inputs out of order. It has to be dataframe, quantitative column, null median.

It cannot be a median test, because that is for two or more samples (there is only one here, not a treatment vs control or anything like that).

In all honesty, this question was a giveaway, because I even told you what test to run!

(27) (2 points) The output from your code in the previous question contains more than one P-value. Which is the correct one to use here? Explain briefly.

Look at a sign test example in your notes to see that the output has three P-values, labelled “lower”, “upper”, and “two-sided”. We are testing the alternative hypothesis that the median is less than 25, so the one we want is the one labelled “lower”.

At a minimum, “the one marked ‘lower’ because we are testing whether the median is *less* than 25”. Say “the one marked ‘lower’ ” and something about why.

Saying “the lower P-value” without putting “lower” in quotes or otherwise indicating that you mean the one *called* lower on the output, is ambiguous, because you could mean the above (which would be correct) or you could mean the smallest of the three P-values (which is wrong in general, though it happens to be correct here). If you couple “the lower P-value” with a good enough explanation of why, you might (at the grader’s discretion) get the points. (I turned out to be pretty generous with this: if it seemed that you knew what you were talking about, you got the two points.) Listing what the three P-values were labelled as (“lower”, “upper”, “two-sided”) was a good way to show that you were looking at the right thing.

One (rather generous) point for saying which P-value to use, and one for saying why. “The lower P-value” by itself is only half a point.

Someone drew what the output would look like (without any numbers, of course), circled “lower”, and wrote next to it “we want”, which I thought was a rather elegant way to demonstrate which P-value we needed.

(28) (3 points) A 99% confidence interval for the population median `score` is shown in Figure 20. What code was used to obtain this confidence interval?

This:

```r
ci_median(wcst, score, conf.level = 0.99)
```

```
[1] 11.00488 22.99854
```

Don't forget the `conf.level` to get a confidence level other than the default 95%, and to express the confidence level as a number less than 1.

Points:

- 1 for using `ci_median`
- 1 for the dataframe and the column in the right order
- 1 for `conf.level` with the right value (something like `conf.level = 99` is a half point).

If you got 26 wrong, I tried to grade this one consistently with that: that is to say, if your answer here was consistent with your answer to 26, even if that was wrong, you'll get some credit here.

(29) (2 points) Let M denote the population median score. Suppose we test $H_0 : M = 20$ against $H_a : M \neq 20$. What does the confidence interval in Figure 20 tell you about the P-value of this test? Explain briefly.

20 is *inside* the confidence interval, so it will *not* be rejected at the corresponding α level, which is $1 - 0.99 = 0.01$. Hence the P-value will be greater than 0.01. (This test is two-sided, unlike the other one we did, so there is no concern about halving P-values.) Don't be tempted to automatically answer "greater than 0.05" here; the P-value is greater than whatever went with the confidence interval you had, which was 99% here, which you needed to realize.

A point each for "greater" and for "0.01", each supported by an adequate reason. No points if you don't say why. It's enough to assert that 0.01 "goes with" a 99% confidence level, but you need to build the link between "inside" and "not rejected" for the other point. "Inside therefore don't reject the null" is one point, but for the second you need to say what you know about the P-value.

To verify:

```r
sign_test(wcst, score, 20)
```

```
$above_below
below above
   32    17

$p_values
  alternative    p_value
1       lower 0.02219208
2       upper 0.98935294
3   two-sided 0.04438416
```

The P-value is 0.044, indeed bigger than 0.01, if not by much. (You might rationalize this by saying that 20 is fairly close to the top end of the interval, and so the P-value for a null median of 20 may not be all that much bigger than 0.01.)

This, by the way, is a good example of why it is a bad idea to use a confidence interval to do a test. *You* might be happy to test $H_0 : M = 20$ against $H_a : M \neq 20$ at $\alpha = 0.01$, but your reader might prefer to use $\alpha = 0.05$ (or maybe even something else). If you get the 99% CI and use that, all it is saying is that the P-value is greater than 0.01, and your reader has on that basis no idea what *they* should be concluding. Do the test, give the P-value (here 0.044), and then your reader has the information they need to decide whether they agree with you. In this case, your $\alpha = 0.05$ reader would reject $M = 20$ in favour of $M \neq 20$, even though you didn't.

Extra: I found out some more about the Wisconsin Card Sorting Test, from https://www.sciencedirect.com/topics/neuroscience/wisconsin-card-sorting-test:

> … four target cards are placed in front of the examinee: one showing one red triangle, one with two green stars, one with three yellow crosses and one with four blue circles. The examinee is then given 128 cards and asked to sort the cards under the target cards according to a set criterion (colour, form or number) and the examiner provides feedback as to whether the decision was right or wrong. For example, the examinee could turn over the first card which had two blue triangles on it, this could be placed under the target card with four blue circles (sorting by colour), under the target card with two green stars (sorting by number) or under the target card with one red triangle (sorting by form). The sorting 'rule' is not made explicit by the examiner, and the examinee learns via the feedback provided after each trial whether they are correct or incorrect. After 10 consecutive correct responses, the rule is changed without the examinee's knowledge, and the examinee must now learn the new

> rule. It is normal practice to score performance in terms of number of correct categories attained (runs of 10 consecutive correct responses) and also in terms of percentage of perseverative errors.

I was in the examinee's shoes one time, and the thing that confused me the most was that the rule *changed* and suddenly, having figured out what I thought was happening, I was getting my card called as "incorrect".[4] What they mean above by "perseverative errors" is continuing to use the old rule after finding that it didn't work any more. You as the examinee have to be able to let go of the old rule and try to discover the current one.

I think the scores in our data are something like the total number of cards correctly classified, allowing for the fact that every time the rule changes, there is a certain amount of trial and error in finding out whether the new rule has to do with colour, shape, or number.

[4] My examiner seemed to take an unreasonable pleasure in saying that my card was "INcorrect".

Comparing lecture sections

A large class has 164 students divided into three lecture sections denoted `a`, `b` and `c`. For each student, their mark `m1` on the midterm exam is recorded, along with their lecture section (`lecture`). Some of the data are shown in Figure 21. The course instructor is interested whether there is any difference in performance among the lecture sections.

(30) (2 points) What two key assumptions are required for the analysis in Figure 22?

This is a standard ANOVA (the output from `oneway.test` and Mood's median test look different, as you need to recognize), so the assumptions we need to make are:

- normally distributed exam marks within each lecture section (specifically, marks within each section that are sufficiently close to normal given the sample size). "Normal data" is not precise enough; you need to specify that you are looking at each group separately.
- equal spreads (strictly: standard deviations) of marks within each lecture section.

The best answer talks about lecture sections and exam marks (not about "groups" and "data"), to show that you understand the specific issues here. I was more relaxed about this here than I might have been; certainly if I had added the words "in the context of the data", I would have expected to see "exam marks" and "lecture sections" in your answer.

You can phrase the normally distributed part as "normal enough given the sample size", or you can talk about sampling distributions (which you would assess using a bootstrap), but if you insist on normal distributions within each group *and* large enough sample sizes, you are showing that you don't understand the issues. If you have normality, the sample size doesn't matter; if you have large enough samples (relative to the non-normality you have), the normality doesn't matter. You need one *or* the other, not both of them.

Extra: to demonstrate what I said above about the outputs looking different:

regular ANOVA:

```
students.1 <- aov(m1 ~ lecture, data = students)
summary(students.1)
```

```
             Df Sum Sq Mean Sq F value Pr(>F)  
lecture       2   1290   645.1   3.484  0.033 *
Residuals   161  29810   185.2                 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Welch ANOVA:

```
oneway.test(m1 ~ lecture, data = students)
```

```

	One-way analysis of means (not assuming equal variances)

data:  m1 and lecture
F = 3.5604, num df = 2.00, denom df = 106.99, p-value = 0.03184
```

Mood's median test

```
median_test(students, m1, lecture)
```

```
$grand_median
[1] 74

$table
     above
group above below
    a    28    29
    b    22    32
    c    31    16

$test
       what      value
1 statistic 6.55956811
2        df 2.00000000
3   P-value 0.03763638
```

Obviously the Mood's median test output looks very different from the others, but the key tells that you have a Welch ANOVA instead of a standard one are the words "not assuming equal variances" across the top, and a fractional denominator degrees of freedom (this test statistic is not exactly F-distributed, so the df are chosen to provide the best approximation).

As it turns out, the P-values for all three tests are about the same. This is usually a sign that the test with the most restrictive assumptions (the regular ANOVA) is all right. (If the P-values had been very different, we would have needed to investigate why.) If the normality is OK and the group spreads are not too unequal, the regular and Welch ANOVAs tend to have very similar P-values, in the same way that the pooled and Welch t-tests do with two groups.

(31) (2 points) What do you conclude from the analysis in Figure 22, assuming that it is an appropriate choice of analysis?

This is the F test of an analysis of variance, so the null hypothesis is that all three lecture sections have the same mean mark. (The alternative is "the null is false".) The P-value is 0.033, which is less than 0.05, so we reject the null and conclude that not all of the lecture sections have the same mean midterm mark, or that at least one of the sections has a different mean midterm mark from the others.

At this point, you *stop*. Saying more is likely to cost you. People were pretty good about doing this.

In this question, I am insisting that you get to a conclusion *in the context of the data*. "Reject the null hypothesis", or even "conclude that the means are not all equal", is just about worthless as a conclusion, in terms of telling the course instructor what their results mean. They are your reader here (or so you need to imagine), and they need to hear something from you about midterm marks and lecture sections. Using a word like "performance" here, or even "score", is all right, because it gets at the idea of what you are comparing among lecture sections, but "means of lecture sections" isn't, because your reader's first reaction is going to be "mean *what*?" For all the reader knows (unless you tell them), you might be comparing mean *heights*, or hours of TV watched last week, or anything.

There is an important piece of English here that you need to have command of: "there is difference" means, if it means anything, that *all* the lecture sections have a different mean midterm mark, which is not what rejecting this null means. "There is *a* difference" or "there is some difference" means that maybe some of them are equal, but there is a difference somewhere, and that *is* what rejecting a null of "all means equal" actually implies. ("There are some differences" squeaks over the line, because it gets at the idea that not all the differences may be significant, but there might be only *one* significant difference in actual fact.)

If you lead your reader through your process to a conclusion that they can use, they are more likely to trust your work.

(32) (1 point) What is the analysis in Figure 23? Explain (very) briefly why it is appropriate to use it here.

It is a Tukey analysis (which the output rather gives away). We should use it because the ANOVA F-test was significant, indicating that there are differences to find between the lecture sections (in terms of midterm mark). Saying clearly enough what the analysis does is also good, but it is quicker to simply name it.

Since this is only one point: “we should use the Tukey analysis shown because the ANOVA F-test was significant” is all I need.

The key point here is that we do the Tukey analysis *because the ANOVA in the previous question was significant*, that is to say, we know that the lecture sections are not all the same and want to know where the differences are. If the ANOVA F-test had not been significant, there would have been no value in doing the Tukey analysis, even if all the other assumptions for ANOVA were correct.

Half a point for saying what the analysis is without getting close enough to saying why we did it.

Observation: there were surprisingly many completely blank answers to this one (or a hurriedly scrawled “ANOVA”). This suggests that a lot of people failed to follow my advice (on the course website) to tackle the easiest questions first. If you had written “Tukey” and nothing else for this question, you would have earned a half point in about five seconds, and I would expect pretty much everybody to have been able to do that.

(33) (3 points) What do you conclude from Figure 23? Your answer should consider which lecture sections have the highest and lowest means.

Only lecture sections C and B differ significantly in mean midterm mark; neither of the other two differences are significant. Two points. Discuss section A somehow, either explicitly (by naming it) or implicitly (by using a word like “only” or something like “neither” as I did). Saying that C and B are significantly different by itself is only 1 out of 2.

By looking at the differences in the `diff` column of the Tukey output, the mean for section C is highest (higher than the other two sections), and the B-A difference indicates that the mean for section B is lower than for section A. In other words, C (highest) is significantly larger than B (lowest), with section A in the middle, significantly different from neither C nor B. One point for a large enough fraction of that.

One point only for saying that C is highest and B is lowest without any discussion of significance. (The implication of the question is that you talk about highest and lowest but possibly also other things, like whatever else you need to say to draw a proper conclusion from the Figure. This is the legalese “including but not limited to”.)

Extra: back in the old days, when we used to do these things by hand, we would write the means down in order and draw lines connecting the ones that were not significantly different:

```
B    A    C
------
     ------
```

B and C are not connected by a line, so they are significantly different, but A is not significantly different from either of the others, so you can see how A occupies that sort of strange middle ground where it is not significantly worse than C and not significantly better than B, even though C and B are significantly different from each other. The resolution to this, insofar as there is one, is that if you had more data (say, these same three lecture sections for more than one year), you would have a better chance of figuring out where lecture section A stands relative to the others.

If you need any more space, use this page, labelling each answer with the question number it belongs to.

Figures

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

Figure 1: Packages

The data file shown in Figure 3 is from a survey of high-school students from different social classes who received high or low parental encouragement to go to college or university, and who said that they did or did not intend to go to college or university. (The last column, `frequency`, is the number of students who fell into that combination of categories.) The data file is called `college-plans.txt`, and is in the folder where you are currently running R.

Figure 2: Scenario A

```
social.stratum;encouragement;college.plans;frequency
lower;low;no;749
lower;low;yes;35
lower;high;no;233
lower;high;yes;133
lowermiddle;low;no;627
lowermiddle;low;yes;38
lowermiddle;high;no;330
lowermiddle;low;no;303
uppermiddle;low;no;627
uppermiddle;low;yes;38
uppermiddle;high;no;374
uppermiddle;high;yes;467
higher;low;no;153
higher;low;yes;26
higher;high;no;266
higher;high;yes;800
```

Figure 3: College plans data set

The file `dogs2.txt`, in the folder where you are currently running R, is shown in Figure 5. The data came from an experiment in which eight dogs were given one of two different drugs, and at times 0, 1, 3, and 5 minutes after the drug was administered, a blood sample was collected and the log of the amount of histamine in the dog's blood was recorded. (A logarithm of an amount can be negative.) Unfortunately, the researcher who collected the data was not very tidy about recording it (although all the values are correct). The text separating the data is spaces (not tabs).

Figure 4: Scenario B

```
Drug    lh0    lh1   lh3     lh5
Morphine      -3.22  -1.61  -2.30  -2.53
Morphine          -3.91  -2.81  -3.91  -3.91
Morphine    -2.66   0.34  -0.73  -1.43
Morphine        -1.77  -0.56 -1.05  -1.43
Trimethaphan        -3.51  -0.48 -1.17  -1.51
Trimethaphan  -3.51   0.05  -0.31  -0.51
Trimethaphan   -2.66  -0.19   0.07 -0.22
Trimethaphan  -2.41   1.14   0.72 0.21
```

Figure 5: Dogs data set

You are working in a lab, and the principal investigator in the lab emails you an Excel spreadsheet called `animals.xlsx` (as an attachment to the email, which you can read but not edit, or save in any other form), containing some animal data that you need to analyze, in `Sheet1`.

Figure 6: Scenario C

```
# A tibble: 20 x 6
      rw   fpg glucose insulin  sspg group
   <dbl> <int>   <int>   <int> <int> <chr>
 1  1.04   203     967     138   351 overt
 2  0.97    86     393     115    85 normal
 3  0.91   100     350     221   119 normal
 4  1.07   104     472     180   239 chemical
 5  0.78    98     321     222    99 normal
 6  1.2     89     472     162   257 chemical
 7  0.99    97     379     142    98 normal
 8  1.2    102     472     297   272 chemical
 9  1.18    96     418     130   153 normal
10  1.16   112     562     139   198 chemical
11  0.76    86     381     157   165 normal
12  0.9    213    1025      29   209 overt
13  1       99     336     143   105 normal
14  1.05    96     456     326   235 chemical
15  1.06   151     854      76   260 overt
16  0.85   216    1113      81   378 overt
17  0.92   303    1364      42   346 overt
18  0.95    96     356     112    73 normal
19  1.05   110     477     124    60 chemical
20  0.76    90     353     263   165 normal
```

Figure 7: Diabetes data (20 randomly chosen rows out of 145 observations)

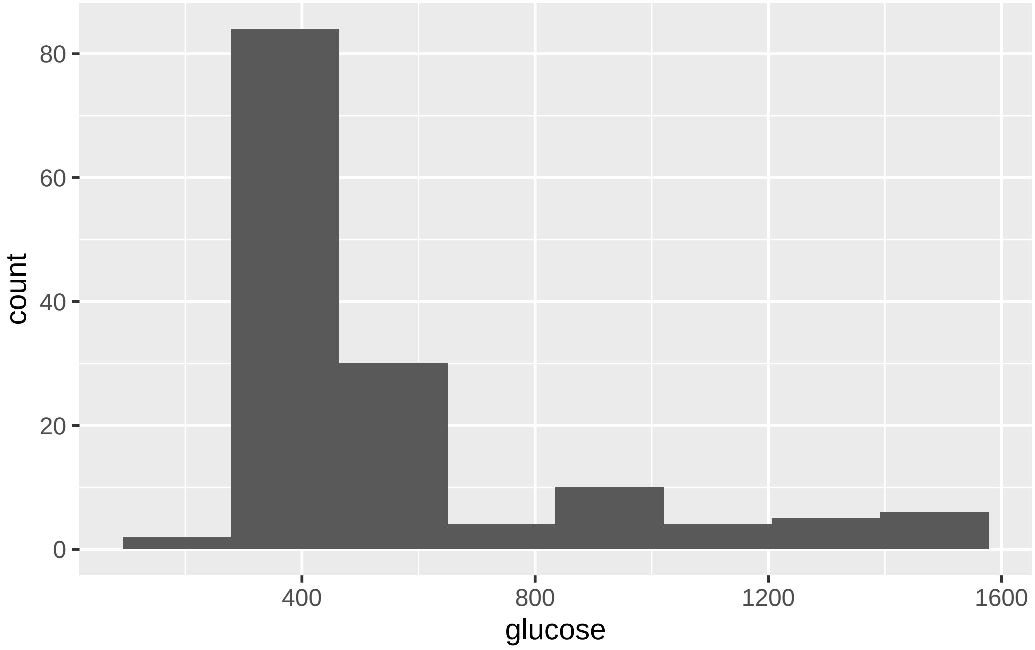


Figure 8: Graph of diabetes data

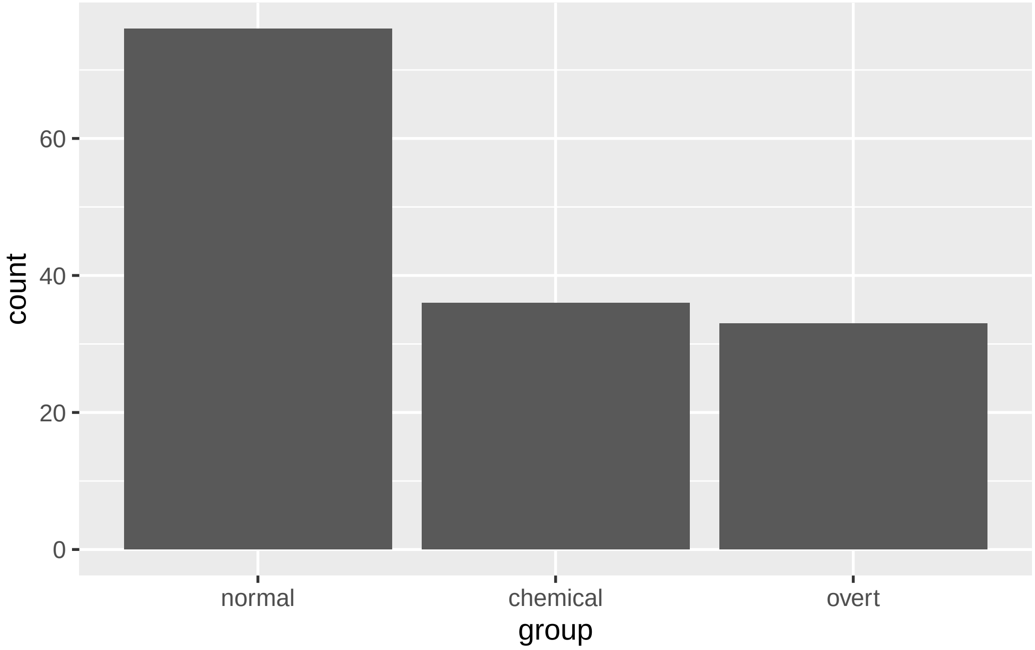


Figure 9: Another graph of diabetes data

```
# A tibble: 100 x 1
   heights
     <int>
 1      71
 2      67
 3      69
 4      70
 5      68
 6      63
 7      68
 8      72
 9      70
10      70
# i 90 more rows
```

Figure 10: Men's heights (first 10 rows)

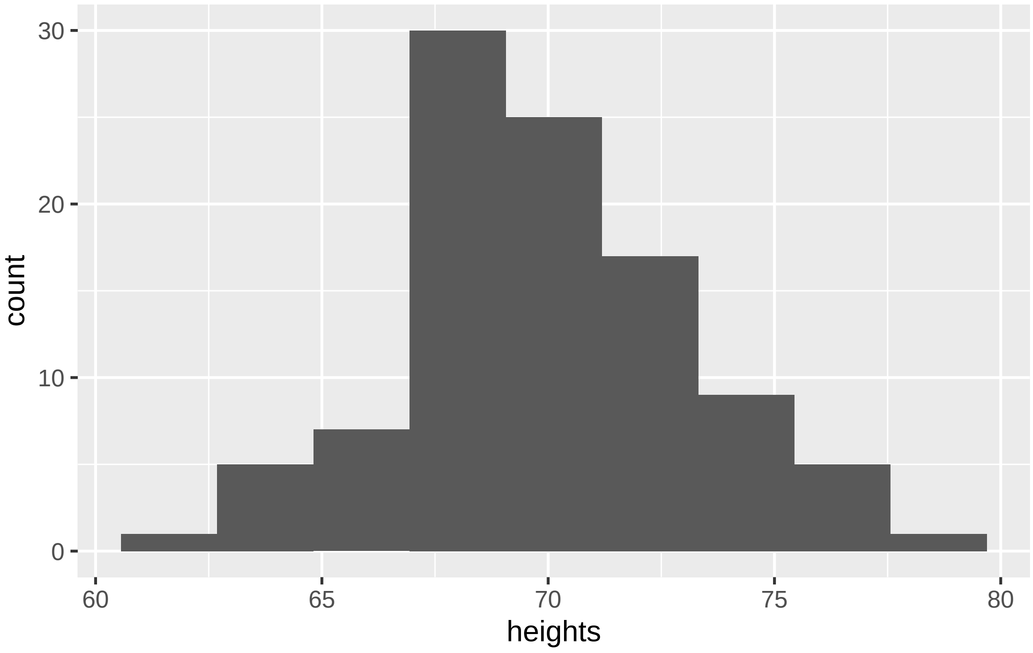


Figure 11: Histogram of heights

```
    One Sample t-test

data:  heights
t = 2.825, df = 99, p-value = 0.005719
alternative hypothesis: true mean is not equal to 69.1
95 percent confidence interval:
 69.37679 70.68321
sample estimates:
mean of x
    70.03
```

Figure 12: Male heights t-test

```
tibble(sim = 1:10000) %>%
  rowwise() %>%
  mutate(my_sample = list(sample(male_heights$heights, replace = TRUE))) %>%
  mutate(my_mean = mean(my_sample)) %>%
  ggplot(aes(sample = my_mean)) + stat_qq() + stat_qq_line()
```

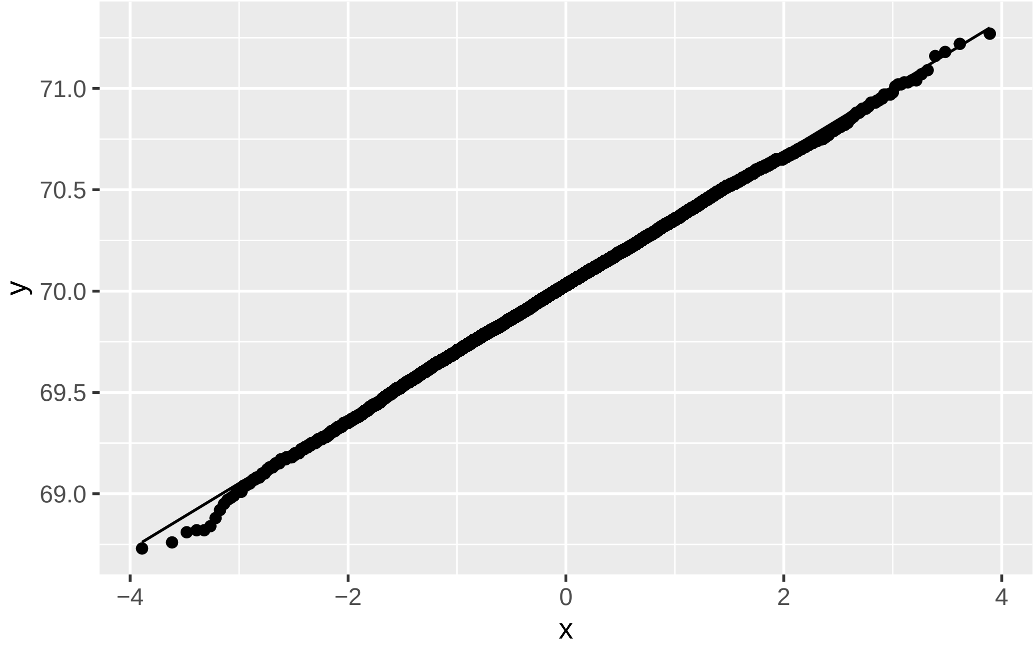


Figure 13: Code and output for heights data

```
# A tibble: 20 x 2
   Condition   Rom
   <chr>     <dbl>
 1 LBP        88.7
 2 No LBP     95.4
 3 No LBP     92.7
 4 No LBP     91.9
 5 No LBP     96.7
 6 LBP        83.3
 7 LBP        72.3
 8 No LBP     94.1
 9 LBP        91.3
10 No LBP     82.4
11 No LBP     90.0
12 No LBP     85.6
13 LBP        89.9
14 No LBP    100.
15 No LBP     97.5
16 No LBP     82.0
17 No LBP     98.5
18 LBP        79.1
19 No LBP     95.3
20 LBP        88.4
```

Figure 14: Lower back pain data (in dataframe `lbp`), 20 randomly chosen rows

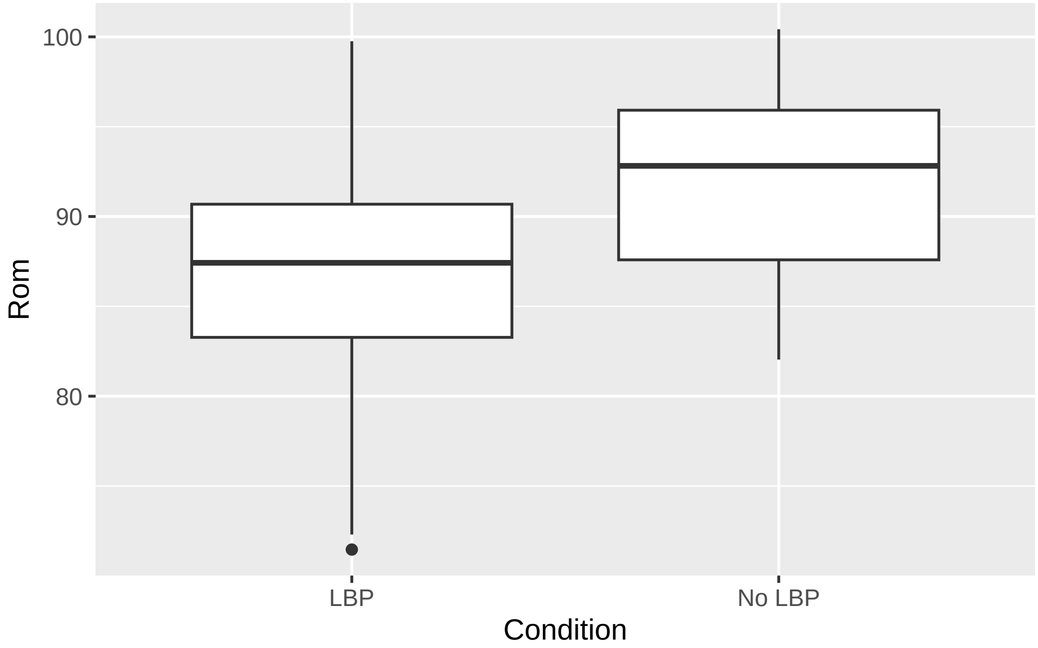


Figure 15: Boxplot of lower back pain data

```
tibble(sim = 1:1000) %>%
  rowwise() %>%
  mutate(my_sample = list(sample(rnorm(50, 40, 15)))) %>%
  mutate(t_test = list(t.test(my_sample, mu = 45))) %>%
  mutate(p_value = t_test$p.value) %>%
  count(p_value <= 0.05)
```

```
# A tibble: 2 x 2
  `p_value <= 0.05`     n
  <lgl>             <int>
1 FALSE               380
2 TRUE                620
```

Figure 16: Mystery code

```
     Two-sample t test power calculation

              n = 9.283698
          delta = 0.8
             sd = 0.5
      sig.level = 0.05
          power = 0.9
    alternative = two.sided

NOTE: n is number in *each* group
```

Figure 17: Output from your previous code

```
# A tibble: 20 x 1
   score
   <int>
 1    40
 2     4
 3    19
 4     8
 5    28
 6    25
 7    19
 8    19
 9    26
10    18
11    78
12    11
13     8
14    65
15    17
16    26
17    11
18    19
19     7
20    94
```

Figure 18: Wisconsin card-sorting test data (20 randomly chosen rows) in dataframe `wcst`

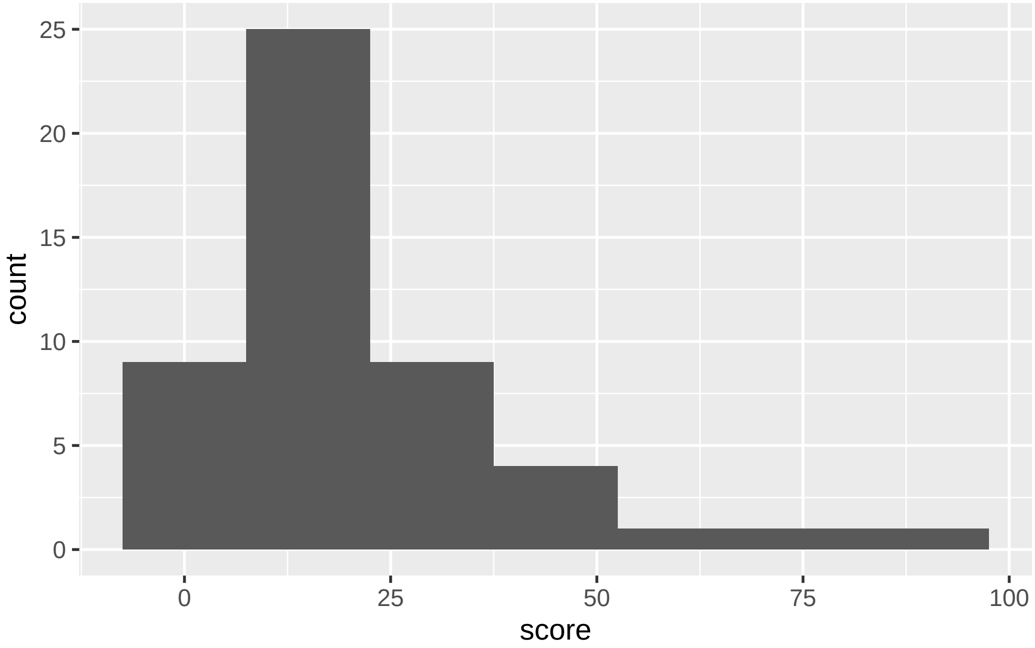


Figure 19: Histogram of Wisconsin card-sorting test data

```
[1] 11.00488 22.99854
```

Figure 20: Confidence interval for median `score`

```
# A tibble: 20 x 2
      m1 lecture
   <dbl> <fct>
 1    89 a
 2    86 b
 3    79 a
 4    83 b
 5    95 c
 6    63 a
 7    48 b
 8    59 c
 9    85 b
10    71 a
11    80 a
12    70 a
13    99 a
14    63 a
15    62 b
16    50 a
17    73 b
18    97 b
19   100 a
20    81 a
```

Figure 21: Lecture section data (20 randomly chosen rows)

```
             Df Sum Sq Mean Sq F value Pr(>F)
lecture       2   1290   645.1   3.484  0.033 *
Residuals   161  29810   185.2
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 22: Lecture section analysis 1

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = m1 ~ lecture, data = students)

$lecture
         diff        lwr       upr     p adj
b-a -3.139812 -9.1981360  2.918512 0.4395988
c-a  3.837728 -2.3413371 10.016793 0.3084418
c-b  6.977540  0.7201228 13.234957 0.0247211
```

Figure 23: Lecture section analysis 2