University of Toronto Scarborough
Department of Computer and Mathematical Sciences
STAD29 / STA 1007 (K. Butler), Midterm Exam
March 2, 2019

Aids allowed:

- 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 28 numbered pages of questions. Check to see that you have all the pages.
In addition, you should have an additional booklet of output to refer to during the exam. Contact an invigilator if you do not have this.
Answer each question in the space provided (under the question).
The maximum marks available for each part of each question are shown next to the question part.
You may assume throughout this exam that the code shown in Figure 1 of the booklet of code and output 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.

In my Crowdmark grading, I use the shaded (blue) rectangle to denote something wrong (in your answer) or something missing (in the question).

Question 1 (7 marks)

An office equipment repair company services photocopiers in large businesses. Data are collected from a number of service calls, as shown in Figure 2. The column `machines` shows the number of photocopiers needing service on a service call, and the column `minutes` shows the total number of minutes the technician spends on site to service all the photocopiers that need service on that visit.

(a) (2 marks) A plot of the dataset is shown in Figure 3. Describe briefly what you conclude from this plot.

My answer: There is a strong upward linear trend. As the number of machines increases, the number of minutes it takes to service them increases. The relationship is not obviously curved. There is not much scatter.

At a minimum: there is an upward trend. As the number of machines increases, the number of minutes it takes to service them increases. (Something about the graph, and something about what that says about the data. Guideline: a point for each of those.)

Extra: I'm not sure I believe that these are real data. I would have expected that occasionally a photocopier would have the kind of problem that takes a long time to figure out, and so the pattern would be (at my guess) a lot less clear than this. But these are the data I have (they came from a textbook).

(b) (2 marks) A regression model is fitted, predicting the number of minutes spent on site by the technician from the number of machines needing service. The output is shown in Figure 4. Quote *one* number from this output that is consistent with what you saw in the plot, and explain briefly how it is consistent.

My answer: I reckon you can pick any of: the slope, the P-value for the slope, the R-squared for the regression, the P-value for the whole regression. So one of these:

- The slope is 14.74, clearly positive, and consistent with the upward trend on the scatterplot.
- The P-value for the slope is 4.1×10^{-15}, very small. The null hypothesis being tested is that the slope is 0, which is clearly rejected; there really is a trend.
- R-squared is 0.98, very high, so that the fit of a straight line to the data is very good.
- The P-value for the regression as a whole is also 4.1×10^{-15}. This means that something (ie. number of machines) helps to predict the total number of minutes required for service.

(c) (3 marks) The repair company receives a call from a business with 6 photocopiers needing service. Use the appropriate part of Figure 5 to obtain an interval estimate of how long the technician will be on site. Justify your choice briefly.

My answer: This is talking about *one* business, not the average time spent on site for *all* businesses that have 6 machines needing repair. It therefore requires a prediction interval, the

second one. One point for choosing a prediction interval, one for explaining why (or why not the other one).

Thus the technician can expect to be on site between 76 and 96 minutes (you need to supply units). One more point.

Question 2 (16 marks)

In some places, you pay a deposit when you buy a soft-drink bottle, say 10 cents, and then you can return the empty bottle to the store where you bought it and get the deposit back. This is environmentally friendly, because the soft-drink manufacturer can get the bottles back from the store, wash them and re-use them. (This is usually done with glass bottles rather than plastic ones.)

A convenience store chain ran an experiment to see whether soft drink bottles are more likely to be returned if the deposit is higher. 3,000 one-litre bottles of a certain soft drink were individually bar-coded and randomly assigned to one of six different deposits ranging from 2 cents to 30 cents, labelled so that a customer could see what the deposit was after they bought the bottle. Records were kept of whether or not each bottle was returned to the store where it was bought, and what the deposit was on that bottle. The dataset is shown in Figure 6, with each row containing a summary of how many bottles were sold at each deposit level, and how many were returned.

(a) (2 marks) Why is logistic regression a sensible idea here?

My answer: The response variable, whether or not a bottle is returned, is a categorical variable (with two levels, returned or not). Extra: we are modelling the probability that a bottle will be returned as it depends on the deposit cost of the bottle.)

The way I asked the question, "the response variable is categorical" is enough. If I had added "explain briefly", I would have expected to see how you know that the response variable is categorical, eg. by naming its two levels "returned" and "not returned". In retrospect, it would have been better for me to ask for it, but I can only expect you to answer the question I asked!

Make sure you're able to distinguish between a categorical *variable* and its *levels*, the different categories it can be.

(b) (3 marks) A logistic regression was run in Figure 7, to predict the probability of a bottle being returned, as it depends on the deposit cost. The output is shown. A variable `y` was used. Give the R code that was used to create `y`.

My answer: Some thinking first: there are only six rows of the data frame and 3,000 bottles, so each row represents more than one individual bottle (actually 500 of them). So we need a two-column response, with the first column being "successes" (the number of bottles returned) and the second column being "failures" (the number not returned).

A second piece of thinking: we don't have the number of bottles not returned at each deposit level, but we have the totals, so we can work it out.

This is what I actually did:

```
bottles %>% mutate(unreturned=sold-returned) %>%
    select(returned, unreturned) %>% as.matrix() -> y
y
##      returned unreturned
## [1,]       72        428
## [2,]      103        397
## [3,]      170        330
## [4,]      296        204
## [5,]      406         94
## [6,]      449         51
```

I'm guessing that you probably did something more like this:

```
bottles %>% mutate(unreturned=sold-returned) -> bottles2
y=with(bottles2, cbind(returned, unreturned))
y
```

```
##      returned unreturned
## [1,]       72        428
## [2,]      103        397
## [3,]      170        330
## [4,]      296        204
## [5,]      406         94
## [6,]      449         51
```

or possibly this for the last step:

```
y=cbind(bottles2$returned, bottles2$unreturned)
y
```

```
##      [,1] [,2]
## [1,]   72  428
## [2,]  103  397
## [3,]  170  330
## [4,]  296  204
## [5,]  406   94
## [6,]  449   51
```

It doesn't matter whether the columns have names or not (or if they do, what those names are).

`y` *must* be an R `matrix` in the end, so you need to explicitly create it with `as.matrix` (my first way), or implicitly create it by feeding `cbind` two vectors (my second and third ways). Thus, for example, this will not work:

```
y=with(bottles2, bind_cols(c1=returned, c2=unreturned))
y
```

```
## # A tibble: 6 x 2
##      c1    c2
##   <dbl> <dbl>
## 1    72   428
## 2   103   397
## 3   170   330
## 4   296   204
## 5   406    94
## 6   449    51
```

```
class(y)
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

`tidyverse` stuff like this is nice generally, but it produces *data frames*, which is not what we want here.

If you find a way to do it that will work, I'm good. As long as I can follow through your process and it will do the right thing, I'm happy.

Grading guideline: one point for calculating the number of bottles *not* returned, two marks for making a two-column response of some reasonably plausible kind. (Quite a lot of people went straight for the two-column response, combining `returned` and `sold`. Two points for this, since it shows the right idea, but for three points you need to combine successes (returned) with *failures*, not-returned, not with the total.

(c) (2 marks) Does the size of deposit have a significant effect on whether or not a bottle is returned?

My answer: This is exactly what the (very small) P-value of 2×10^{-16} on the `deposit` line tells us: that trend of more bottles being returned when the deposit is higher is definitely not chance. The slope is definitely not zero.

I said "significant" because I want you to do a test and get a P-value.

(d) (2 marks) Why does it make sense that the Estimate for `deposit` in Figure 7 is positive? Explain briefly.

My answer: The estimate is 0.136.

This says that as the deposit increases, the probability of the bottle being returned also increases. This makes sense for a couple of reasons, either of which I'm happy with:

- from a practical point of view, a larger deposit means that you get more money back when you return the bottle, which makes it worth doing (particularly if you're returning several bottles).
- if you look back at the data in Figure 6, the number (and thus proportion) of bottles returned goes up as the deposit goes up, and so we'd expect the model to say the same thing.

In this part, we are *not* looking at the P-value (we did that in the previous part). Also, I don't need a precise interpretation of the slope itself; that would be that as the deposit goes up by one cent, the log-odds of the bottle being returned goes up by 0.136, which is hard to make sense of anyway. Beware of trying to interpret the slope as a change in *probability*; if it were that, you could make the probability get larger than 1 by making the deposit large enough!

(e) (1 mark) In Figure 8, I obtain some predictions. What does the `type="response"` do? Explain (very) briefly.

My answer: It obtains predicted probabilities (of a bottle being returned) for each deposit level (in the original data). If you leave it out, you get predicted log-odds.

I don't need all of that for just one point. "It gets predicted probabilities" is enough. I do need to see "probability" in there somewhere, though.

(f) (4 marks) Give R code to make a graph showing the predicted probabilities of a bottle being returned, joined by lines, along with the proportions in the data of bottles that were returned at each deposit level, shown as points. Your graph should look something like Figure 9. Hint: you have a calculation to do before you draw the graph. You may use any of the variables that I obtained in Figures 6 through 8 to make your graph.

My answer: The calculation to be done first is to calculate the observed proportions in the original data set:

```
bottles %>% mutate(proportion=returned/sold) -> bottles_prop
bottles_prop
```

```
## # A tibble: 6 x 4
##   deposit  sold returned proportion
##     <dbl> <dbl>    <dbl>      <dbl>
## 1       2   500       72      0.144
## 2       5   500      103      0.206
## 3      10   500      170      0.34
## 4      20   500      296      0.592
```

```
## 5      25   500      406      0.812
## 6      30   500      449      0.898
```

One point for this.

Now we need to use both this data set and the one I called `preds`. There are two strategies to choose from here:

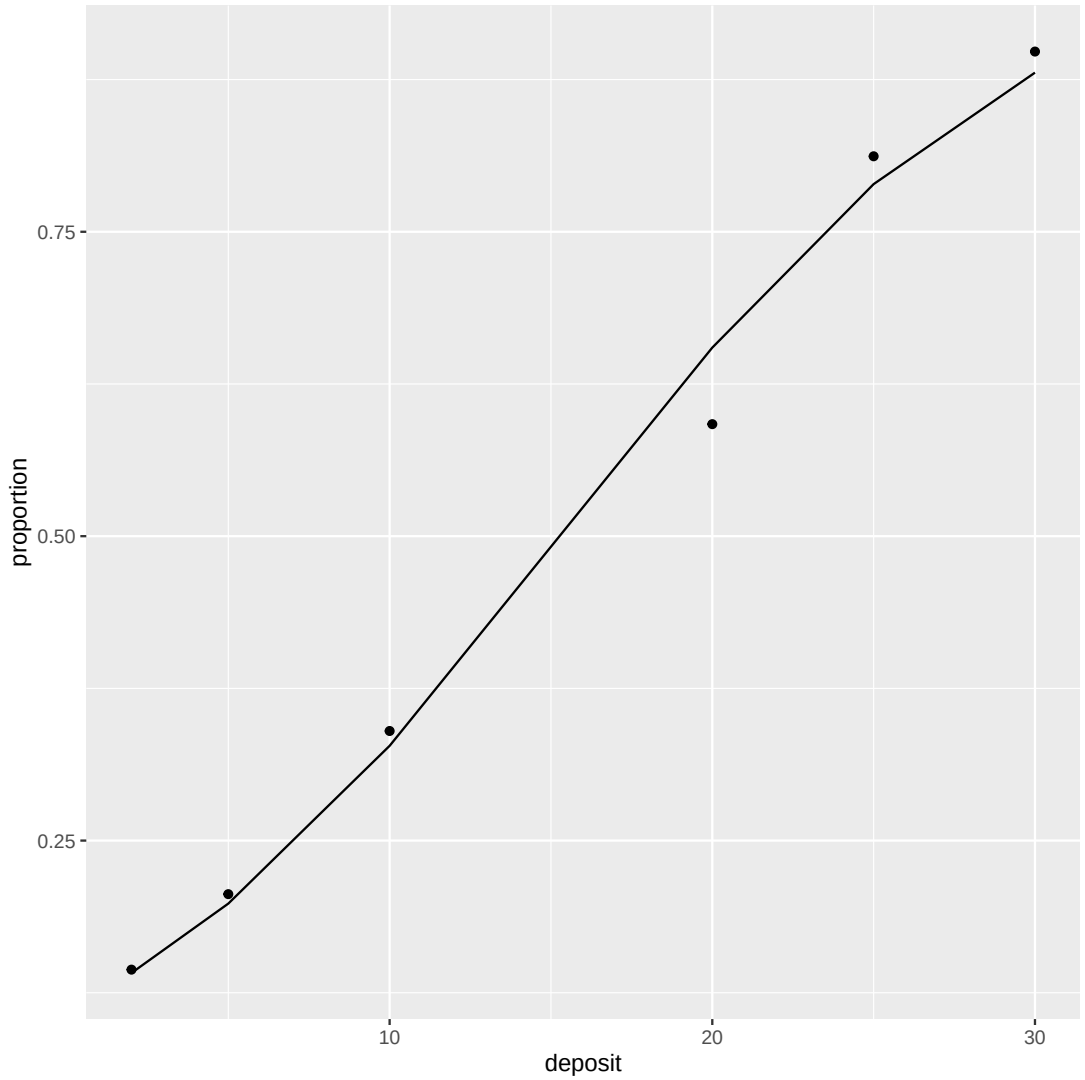
- use `ggplot` and get the data for the plot from two different data frames

- Make a combined data frame first and then use that for the plot.

I'm good with either of these, if you do your way right.

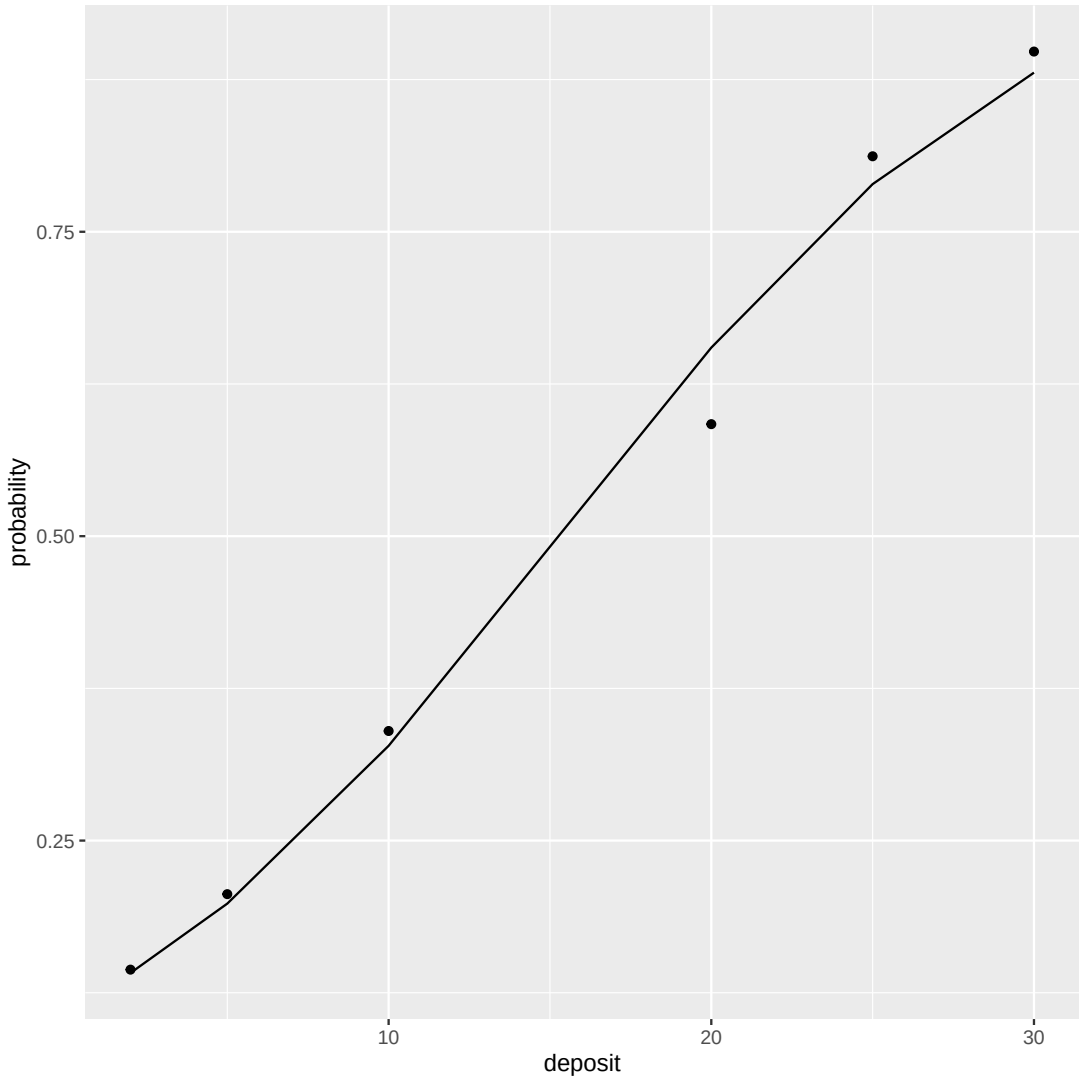
Using two separate data frames in one plot: that goes one of these two ways, depending on whether you use the predictions as the base or whether you use the original data with the proportions as the base:

```
ggplot(bottles_prop, aes(x=deposit, y=proportion))+
    geom_point()+
    geom_line(data=preds, aes(y=probability))
```



Or you can do it the other way around, starting with the predictions:

```
ggplot(preds, aes(x=deposit, y=probability))+
    geom_line()+
    geom_point(data=bottles_prop, aes(y=proportion))
```



The y-axis is labelled according to the first thing you plot as a y, so it makes a difference for the plot which way around you do it, but I have no preferences here.

The other strategy is to combine the two data frames first. Since the deposit values are in the same order in both, `cbind` or `bind_cols` will do it:

```
all=cbind(bottles_prop, preds)
all
```

```
##   deposit sold returned proportion deposit sold returned probability
## 1       2  500       72      0.144       2  500       72   0.1412601
## 2       5  500      103      0.206       5  500      103   0.1982432
```

```
## 3      10  500      170      0.340      10  500      170   0.3278210
## 4      20  500      296      0.592      20  500      296   0.6548554
## 5      25  500      406      0.812      25  500      406   0.7891326
## 6      30  500      449      0.898      30  500      449   0.8806877
```

or

```
all=bind_cols(bottles_prop, preds)
```

```
## New names:
## * `deposit` -> `deposit...1`
## * `sold` -> `sold...2`
## * `returned` -> `returned...3`
## * `deposit` -> `deposit...5`
## * `sold` -> `sold...6`
## * `returned` -> `returned...7`
```

```
all
```

```
## # A tibble: 6 x 8
##   deposit...1 sold...2 returned...3 proportion deposit...5 sold...6 returned...7
## *       <dbl>    <dbl>        <dbl>      <dbl>       <dbl>    <dbl>        <dbl>
## 1           2      500           72      0.144           2      500           72
## 2           5      500          103      0.206           5      500          103
## 3          10      500          170      0.34           10      500          170
## 4          20      500          296      0.592          20      500          296
## 5          25      500          406      0.812          25      500          406
## 6          30      500          449      0.898          30      500          449
## # ... with 1 more variable: probability <dbl>
```

or

```
bottles_prop %>% bind_cols(preds) -> all
```

```
## New names:
## * `deposit` -> `deposit...1`
## * `sold` -> `sold...2`
## * `returned` -> `returned...3`
## * `deposit` -> `deposit...5`
## * `sold` -> `sold...6`
## * `returned` -> `returned...7`
```

```
all
```

```
## # A tibble: 6 x 8
##   deposit...1 sold...2 returned...3 proportion deposit...5 sold...6 returned...7
## *       <dbl>    <dbl>        <dbl>      <dbl>       <dbl>    <dbl>        <dbl>
## 1           2      500           72      0.144           2      500           72
## 2           5      500          103      0.206           5      500          103
## 3          10      500          170      0.34           10      500          170
## 4          20      500          296      0.592          20      500          296
## 5          25      500          406      0.812          25      500          406
## 6          30      500          449      0.898          30      500          449
## # ... with 1 more variable: probability <dbl>
```

or any of those the other way around, starting with `preds` and combining `bottles_prop` with it.

Or, if you wanted to allow for the possibility that the deposit values could be in a different order in the two data frames and you wanted to make sure they matched, you could do

```
bottles_prop %>% left_join(preds) -> all
```

```
## Joining, by = c("deposit", "sold", "returned")
```

```
all
```

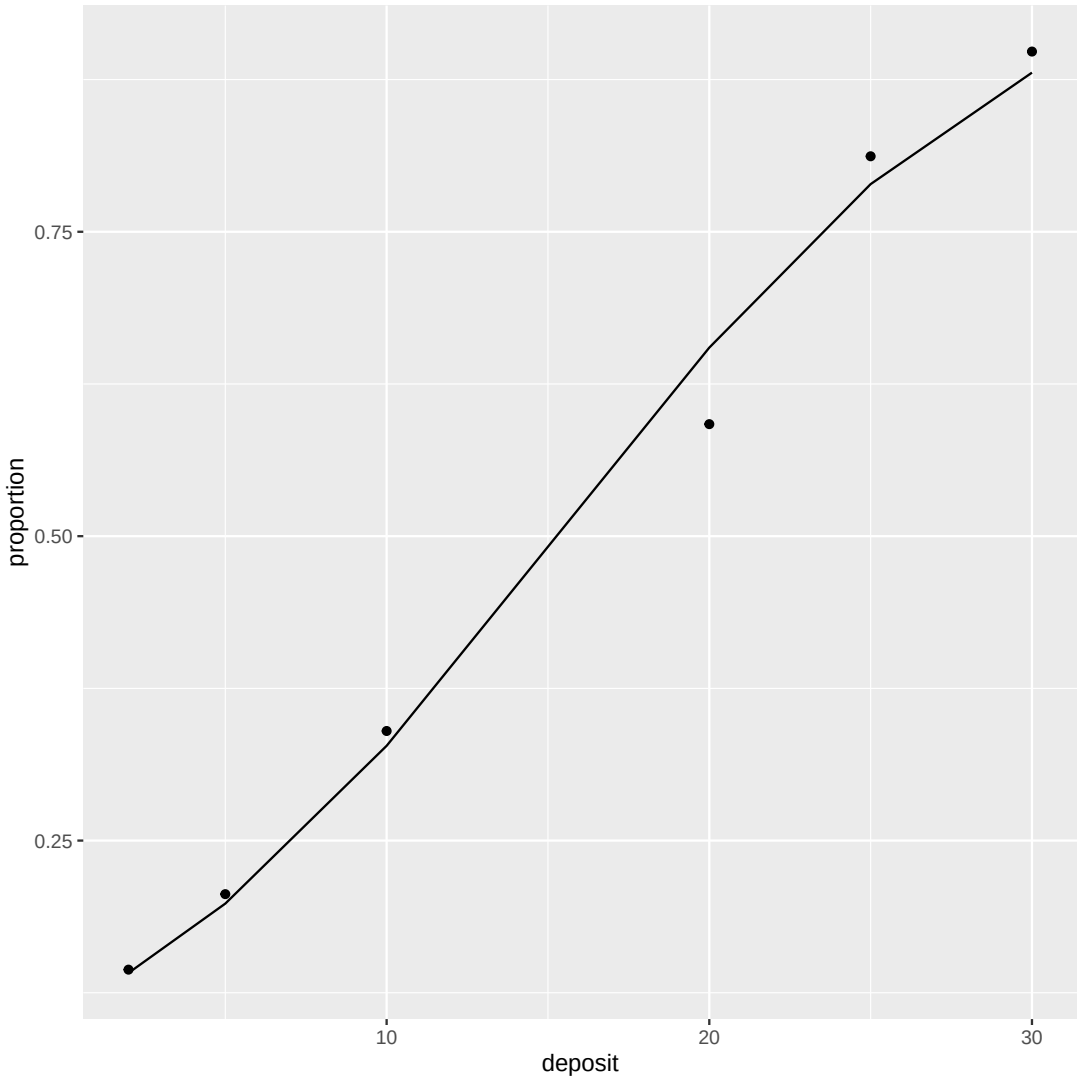
```
## # A tibble: 6 x 5
##   deposit  sold returned proportion probability
##     <dbl> <dbl>    <dbl>      <dbl>       <dbl>
## 1       2   500       72      0.144       0.141
## 2       5   500      103      0.206       0.198
## 3      10   500      170      0.34        0.328
## 4      20   500      296      0.592       0.655
## 5      25   500      406      0.812       0.789
## 6      30   500      449      0.898       0.881
```

or, again, the other way around. This last is the cleanest, because you run the risk of getting repeated column names (when using `cbind`) or having the column names change (when using `bind_cols`). However, for you, any of these is good.

The message "joining" below the `left_join` says that it's looking for deposit-sold-returned rows that are the same in the two data frames, and then combining those with the `probability` that goes with them from `preds`, and the `proportion` that goes with them from `bottle_preds`.

Then the graph, using this data frame `all` that you just created somehow:

```
ggplot(all, aes(x=deposit, y=proportion))+
    geom_point()+
    geom_line(aes(y=probability))
```



or, again, the other way around (plotting probability with lines first, then plotting proportions with points).

Grading: One point for calculating the proportions from the original data. If you create a combined data frame, one point for that and two points for the (easier) graph; if you use two separate data frames, three for the graph, which now has to have a `data=` in it somewhere.

Extra: each of these points is based on the same number of bottles (500), so there is no benefit in using `size` here. When I used that in class, it was because the groups had different numbers of observations in them.

There are a lot of possibilities here, so it looks complicated, but if you have done these things,

in some fashion:

- made a column containing the observed proportions
- plotted both the observed proportions and the predicted probabilities against the `deposit` amount

then I am good with it. Python has a "there is one way to do it" mentality, but R very definitely does not!

(g) (2 marks) The plot that you gave code for in the previous part is shown in Figure 9. What does this plot tell you about how well the logistic regression fits? Explain briefly.

My answer: I would say that it fits very well, because the points are close to the curve. Or, the curve and the data points show the same pattern. Or something like that. If you want to be critical, you could say that fewer bottles with a 20-cent deposit were returned than expected, but I still think you should say that otherwise the fit of the model is good.

Question 3 (15 marks)

Arthritis refers to pain or stiffness in the joints of the human body. A study investigates a new treatment for arthritis. In the study, patients are randomly assigned to the new treatment (labelled `Treated`) or to a placebo (`Placebo`), and the age and sex of each patient is recorded. Some of the data is shown in Figure 10. The column `impr` is a numerical code: 0 is no improvement, 1 is moderate improvement, and 2 is complete recovery from the arthritis. Figure 11 shows the fitting of a model, and some output from that model.

(a) (2 marks) Why did I need to use `factor(impr)` rather than just `impr` in the first line of Figure 11? Explain briefly.

My answer: `impr` is actually a categorical variable rather than a quantitative one (the numbers represent categories), so this needs to be expressed in the model, by turning the numbers into a categorical variable.

"Because `polr` needs a categorical response variable" is only one point by itself. You need to explain how `impr` *is not* currently a categorical variable but should be.

(b) (2 marks) Why did I use `polr` rather than `multinom`? Explain briefly.

My answer: Because the categories of improvement come in a natural order: no improvement is worse than moderate improvement is worse than complete recovery.

Extra: the variable `impr` is ordinal but not interval, in the jargon, because the gap between 0 and 1 is not necessarily the same as the gap between 1 and 2. (It is not clear whether it is or isn't.) If we were happy that the difference between 0 and 1 was the same as the difference between 0 and 2, we could treat `impr` as quantitative, and use ordinary regression to predict the "improvement score", getting fractional numbers as predictions. But that probably doesn't make any sense here.

(c) (2 marks) Should I remove any explanatory variables from the model? If so, which ones? Explain briefly.

My answer: In the `drop1` output of Figure 11, all three explanatory variables are significant at $\alpha = 0.05$, so I should keep them all. If you prefer, the AIC of "none" is smallest, so the best thing is to drop nothing.

If you want to use an α of 0.01, go ahead, in which case you would remove `age` (first, then re-fit). But removing `age` because its P-value is the largest one, without saying what α you are using, will not work. The idea of backward elimination is that you remove the variable with the highest P-value *if it is not significant*. If everything is significant (the case here at $\alpha = 0.05$), you stop.

(d) (3 marks) In Figure 12 I make some predictions for the two sexes, the two treatments and two representative ages. I first created a data frame containing all the combinations of these, which I called `new`. What code would get predicted probabilities of each improvement category for each of these combinations, using the model that I fitted in Figure 11?

Question 3 continues...

My answer: `predict`, using the fitted model, the data frame of values to predict for, and `type="probs"`, in that order:

```
p=predict(arthritis.1, new, type="probs")
```

This is the code I used (and hid from you). One point each for `predict` with the model, the new data frame, and the `type="probs"`.

This also works: `type="p"`. The reason it does is a thing R does called "partial matching": there are several possible values for `type`, but `probs` is the only one that begins with P, so it assumes you mean `probs` in this case.

I had already created the data frame `new`, so you didn't need to; if you did, and you gave it a different name but used that name in your `predict`, I was fine with that. (I didn't need your `new` to be correct, since I wasn't asking about that.)

Make sure you know why it's `probs` and not `response`: the former is for a `polr` model and the latter is for a two-category regular logistic regression that you fit with `glm` and `family="binomial"`.

(e) (3 marks) From Figure 12, is being older rather than younger associated with a better or worse recovery overall? Explain briefly.

My answer: Pick two rows that differ only in age, but are the same for sex and treatment ("all else equal"). This could be the first and second rows of the table of predictions. (One point for picking a sensible pair of rows to compare.)

Then look to see how the recovery categories compare. You will probably find, as age increases, that the (predicted) probability of no improvement goes down, of complete recovery goes up, and of moderate improvement goes either up or down depending on which pair you are looking at. You need to talk about what the levels of improvement actually *are*, not just the numbers 0, 1, and 2, because the question asks about inference from the actual data. One point.

For the final point, you need to say that overall recovery is *better* for an older person, because an older person is both less likely to show no improvement and more likely to show complete recovery.

I wanted you to look at at least two of the recovery categories and preferably all three, so that you know what is going up, what is going down, and what the overall picture is.

I should have given you a bigger space to write your answers in!

(f) (3 marks) Based on the output in Figures 11 and 12, would you say that the treatment is *helpful*? Explain briefly.

My answer: First step: there is a significant effect of treatment, as shown in the `drop1` table of Figure 11. One point. This significant effect might be positive or negative, though; the `drop1` output doesn't say which way it goes. The treatment, for example, could be significantly *harmful*, and we need to rule that out. (This is important because if the treatment is not significant, any improvement we appear to see by looking at the predicted probabilities is just chance. This is also the reason I referred you to Figure 11 in the question.)

To see whether the treatment effect is positive or negative, follow the same kind of procedure as in the previous part: find two combinations that differ in treatment but not anything else, compare the three predicted probabilities (one point), and say whether that indicates an overall improvement or not (the last point).

For example, comparing females of age 46 on treatment and placebo (rows 1 and 5 of the predictions), the predicted probability of no improvement goes sharply down, of moderate improvement goes slightly up, and of complete recovery goes sharply up. The overall picture from here is that recovery overall is *much* better for the patients who received the treatment than for those who received the placebo: they are much more likely to recover completely and much less likely to show no improvement.

I'm willing to forgive repeats of errors from the previous part, or a less extensive discussion of issues you showed that you clearly understood there. The process here, of comparing the three predicted probabilities and inferring an overall picture about recovery, is the same as the previous part.

Question 4 (16 marks)

A number of patients suffering from some disease are given one of two treatments, labelled A and B. Each of these patients is observed for a time; either the patient dies, or something else happens to them

(they are definitely known to have survived, or their status is unknown, but they were never observed to die). Each patient's age is also recorded. The data are shown in Figure 13, with survival times in months. The event of interest is "death".

(a) (3 marks) Give R code to create a suitable response variable `y` for a Cox proportional hazards model.

My answer: This is the one where I forgot to tell you the name of the data frame, so as long your first input to `with` looks like a data frame of some kind, I'm good with it.

The response variable is the time until death (or end of followup), along with an indication of whether it *was* death or something else ("censored").

Whether or not each patient died is in the column `status`, in particular the value `Died`. The other values, `Survived` and `Unknown`, are both "censored" as far as we are concerned:

```
y=with(patients, Surv(survtime, status=="Died"))
y
##  [1]  1   1   4   5   6+  8   9+  9  12  15+ 22  25+ 37  55  72+
```

Minus one per mistake, to a minimum of one if *something* is correct. (There is the potential for losing two marks on the `status` part if you are not careful.)

The "event" is defined by `status` being `"Died"`, not by something that will evaluate to 1 or TRUE, so you need the part with `Died` in it.

Defining a new status variable also works (but is more work for you):

```
patients %>% mutate(vstat=ifelse(status=="Died", 1, 0)) %>%
    with(., Surv(survtime, vstat)) -> y
```

or something equivalent, and now because my `vstat` evaluates to TRUE if the event happens, just supplying `vstat` as the second input to `Surv` will work.

(b) (2 marks) My response variable is as shown:

```
##  [1]  1   1   4   5   6+  8   9+  9  12  15+ 22  25+ 37  55  72+
```

Why is the fifth value (and some of the others) marked with a `+`? Explain briefly. (The `[1]` is not one of the values.)

My answer: This value is censored (one point); it means that the patient was never observed to die, or didn't die of whatever the disease was until after the study finished, or died of something else (the other point).

If you give a clear explanation without using the word "censored", that is also two points. Using the word "censored" without further explanation is only one, though.

(c) (2 marks) A Cox proportional-hazards model is fitted, with the results shown in Figure 14. Is there a significant effect of age? Using this output, describe the effect of age on survival, justifying your conclusion briefly.

My answer: There is a significant effect of age on survival, with a P-value of 0.0099. (One point.) As to what kind of effect: the positive coefficient of 0.22 says that as age increases, the hazard of death also increases: that is, if you are older, you are more likely to die sooner (from the disease, whatever it is). (Describing the direction of the effect is the second point.)

What is increasing with age is *not* survival time, but the hazard of death. (This is easy to get confused about.) Also, don't try to interpret the number: this is an increase in something like log-hazard in fact. All that we care about is whether it's positive or negative and what that means.

(d) (3 marks) The quartiles of age are 67 and 75. Give code to create a data frame called `new` that contains all combinations of these ages and the two treatments, that is to be used in a moment for obtaining predicted survival probabilities.

My answer:

I changed my mind on this one and made it out of 3 points rather than 2.

This implies that the columns of `new` need to have the same names as the corresponding columns in `patients`, thus:

```
ages=c(67,75)
treatments=c("A","B")
new=crossing(age=ages, treatment=treatments)
```

This gives

```
new
```

```
## # A tibble: 4 x 2
##     age treatment
##   <dbl> <chr>
## 1    67 A
## 2    67 B
## 3    75 A
## 4    75 B
```

Feel free to shortcut this, since the lists of ages and treatments are short. If the code will work, I am happy.

My revised scale for this was: 3 for something equivalent to the above, 2 points for the right thing with a mistake, 1 for something that got part of the way but with more than one mistake. I don't think anybody that wrote something got less than 1.

A common thing was to miss off the quotes around A and B. These refer to the treatments by those names, not variables called A and B. (Levels of a categorical variable rather than the categorical variable itself, again.)

(e) (2 marks) Give R code to obtain predicted survival probabilities, in such a way that we will be able to plot the survival curves using `ggsurvplot`.

My answer: This is `survfit` with *the original data frame on the end*. The model in Figure 14 is called `patients.1`, our data frame of values to predict for is called `new`, and the original data frame was called `patients`, thus:

```
s=survfit(patients.1, new, data=patients)
```

Two points if you get this, one if you have the idea but miss something.

(f) (2 marks) Predicted survival curves are shown in Figure 23. Which of the two treatments is more effective at prolonging life? Explain briefly.

My answer: Compare the two survival curves for the different treatments, for the same age, for example strata 1 and 2, red and green. The red survival curve is more up-and-to-the-right, indicating that patients on treatment A have a higher chance of surviving longer than patients (of the same age) on treatment B.

Two points for naming treatment A for a good reason, one for picking out stratum 1 (or 1 and 3) without naming which treatment it goes with.

Extra: if you were smart in part (c), you cross-checked your answer about ages there with this picture. For example, strata 1 and 3, red and blue, compare age 67 and 75 on the same treatment, and we see that survival for the younger patients is definitely better. I didn't ask about ages here, since the question was already long enough.

(g) (2 marks) How does something in Figure 14 support your conclusion about treatments in the previous part? Explain briefly.

My answer: The coefficient for treatment B is 1.88, strongly positive, meaning that patients on treatment B are more likely to die sooner than those on the baseline treatment A (whose coefficient is zero).

I don't think looking at P-values helps here. The not-quite-significant P-value of 0.052 next to `treatmentB` means that at $\alpha = 0.05$, treatments A and B are not significantly different from each other. This is confusing, so I didn't ask about it; it looks from the plot of survival curves that treatment A is better and the coefficient for treatment B supports that if you look at it properly.

If you got no marks in the previous part, I tried to find you something here if what you said here was consistent with what you said before.

Question 5 (16 marks)

Three treatments for headache, labelled A, B, and C, are being tested. Thirty people, fifteen men and fifteen women, are each randomly assigned to one of the treatments. Each person is instructed to take their assigned treatment when they next get a headache, and to note how many minutes it takes "until the pain subsides".

(a) (2 marks) The data as I originally received it is shown in Figure 15. The blank treatments are repeats of the one above, thus for example five men and five women were assigned to treatment A. Describe briefly how the data are not suitable for analysis in their current form, even after I have fixed up the blank treatments.

My answer: This layout is untidy. The columns Male and Female are levels of a categorical variable called eg. `Gender`, so there needs to be a column called `Gender` and another called `Time` with all the pain relief times in it. Or we need to `pivot_longer` the Male and Female columns to make one column of pain relief times and one column of genders. There are lots of ways to say something sensible, and if you seem to have found one of them, you'll have gotten the points.

Extra: the actual tidying I did was as follows:

```
fname <- "timetorelief.txt"
my_col_names = c("Treatment", "Male", "Female")
read_fwf(fname, fwf_empty(fname, col_names = my_col_names), skip = 1) -> time0
```

```
## Rows:  15 Columns:  3
## -- Column specification --------------------------------------------------------
##
## chr (1):  Treatment
## dbl (2):  Male, Female
##
## 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.
```

```
time0 %>%
  fill(Treatment) %>%
  pivot_longer(-Treatment, names_to = "Gender", values_to = "Time") -> painrelief
painrelief
```

```
## # A tibble: 30 x 3
##    Treatment Gender  Time
##    <chr>     <chr>  <dbl>
##  1 A         Male      22
##  2 A         Female    21
##  3 A         Male      25
##  4 A         Female    19
##  5 A         Male      26
##  6 A         Female    18
##  7 A         Male      27
##  8 A         Female    24
##  9 A         Male      24
## 10 A         Female    25
```

Question 5 continues... This page: 2 marks.

```
## # ... with 20 more rows
```

You are probably unfamiliar with `read_fwf` (I was). The problem with using `read_table`, which is what you would guess, is that there are rows where the treatment is blank, and `read_table` will think that the value in the Male column is the treatment, the Female value is the Male value, and will supply missing for the Female value. It used to work by looking for blank columns all the way down, but it doesn't any more.

`read_fwf` reads files in Fixed-Width Format (hence the name). It seems to require a bit more help to get the file read in. For one thing, it doesn't know about column names, so I have to say before I start what the columns are going to be called (even though they are actually in the file). I also define the file name into a variable, because I am actually going to be using it twice.

`read_fwf` itself has three inputs (as I am using it here):

- the file name
- how to sort out where the columns are (which I come back to shortly)
- an instruction to skip one line, the first line (which in fact has the column names in it)

The `fwf_empty` piece is the instructions for where the columns are, and what they will be called. You have other options, like specifying how many characters wide each column is, but this is the laziest: it says "guess where the columns are, using columns that are blank all the way down as separators". The first input to this is the file name (again), and the second one is where you specify the names that the columns are going to have. There is undoubtedly a way to read them from the file first and supply them back to `fwf_empty`, but I was not clever enough to see it.

The rest of the way is not so bad. By happy coincidence, `read_fwf` replaces any empty values (like the blank treatments after the first) by missing values, and that is exactly what `fill` replaces by the non-missing value above them so that when we pivot longer, we have a complete column of treatments, a complete column of genders, and a complete column of times.

Brief extra extra: you are probably used to seeing blank cells in spreadsheets. When you read those in, having saved the spreadsheet as a csv first, the csv would look like this (for the top of the data we have here):

```
Treatment,Male,Female
A,22,21
,25,19
,26,18
```

and the presence of a comma on the *beginning* of the line indicates that, for the second and third rows of data, that the Treatment for those rows is blank. The commas are how `read_csv` knows where one column ends and the next one begins, and for the data we had, we had no indication like this (which is why `read_table` screwed up).

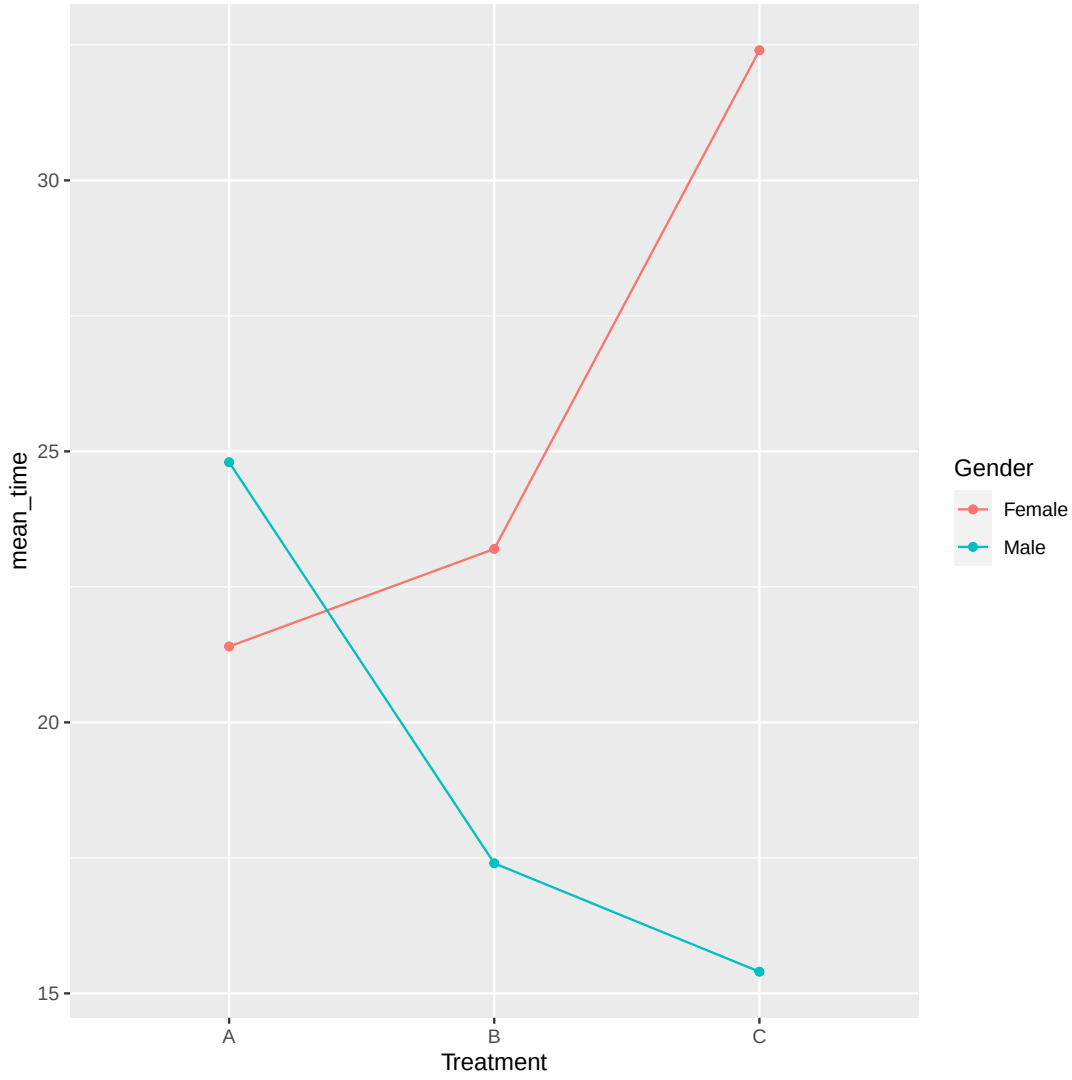
(b) (3 marks) I put the data into a suitable format, and calculated mean pain relief times for each `Treatment-Gender` combination. I saved the group means in a data frame `d`, with the column of means called `mean_time`. In Figure 24 I use `d` to create an interaction plot. Give the R code that was used to create this plot, assuming that you already have the data frame `d`.

My answer:

```
## `summarise()` has grouped output by 'Treatment'.  You can override using the
## `.groups` argument.
```

The bit of code you need is this:

```
ggplot(d, aes(x=Treatment, colour=Gender, group=Gender, y=mean_time)) +
    geom_point()+geom_line()
```

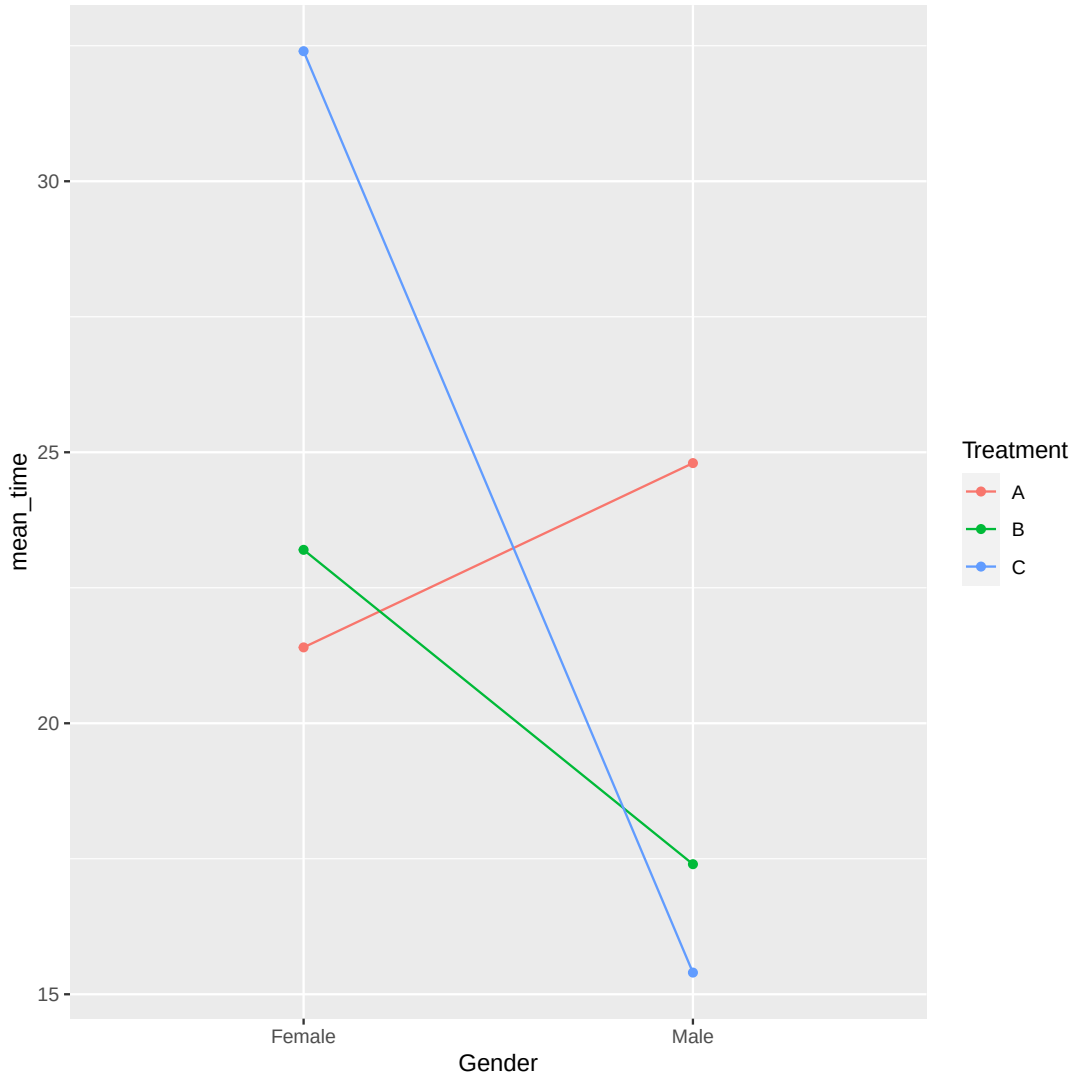


In this one you have to get the `x` and the colour-and-group variables the right way around. On my plot, the `x` is `Treatment`, on the x-axis, and the traces and the legend are `Gender`. `y` is the measured variable, here `mean_time`.

Minus one per error to a minimum of one if you had *something* correct, the usual thing. Some people got two things wrong but this was caused by only one error; those people got two points. If you got one point, it means that you made two (or more) *separate* errors, marked by coloured rectangles around the stuff that wasn't right.

If you get the `x` and the colour-and-group the wrong way around, you'll get not Figure 24 but this:

```
ggplot(d, aes(x=Gender, colour=Treatment, group=Treatment, y=mean_time)) +
    geom_point()+geom_line()
```



which is, you could argue, equally good as an interaction plot, but is not the plot I gave you.

Extra: I got `d` like this:

```
painrelief %>% group_by(Treatment, Gender) %>%
    summarize(mean_time=mean(Time)) -> d
```

```
## `summarise()` has grouped output by 'Treatment'.  You can override using the
## `.groups` argument.
```

```
d
```

```
## # A tibble: 6 x 3
## # Groups:   Treatment [3]
##   Treatment Gender mean_time
##   <chr>     <chr>      <dbl>
## 1 A         Female      21.4
## 2 A         Male        24.8
## 3 B         Female      23.2
## 4 B         Male        17.4
## 5 C         Female      32.4
## 6 C         Male        15.4
```

I didn't need to see code for *you* to get this, but if you supplied it and you consistently used it to get your interaction plot (for example, if you called it something other than `d` and used that name in `ggplot`, or if you piped directly into the `ggplot`), I was happy.

(c) (2 marks) What do you conclude from Figure 24? Explain briefly.

My answer: The lines are not anywhere close to parallel, so there is (we would expect) an interaction between gender and treatment.

That's all I was after. If you wanted to say something more detailed about the relative effects of the treatments for males and females, and it was correct, I would take that too.

(d) (3 marks) An analysis of variance is shown in Figure 16. What do you conclude from it, and what does that mean in the context of the data?

My answer: The interaction is (strongly) significant. (One point.) This means that the effect of treatment on pain relief is different for males and females. (Two points.)

Remember to *stop there*: if you try to interpret main effects here, you will lose points, because we have to understand the interaction first before doing anything else.

(e) (2 marks) Some further analysis is shown in Figures 17, 18, and 19. Which of this analysis is appropriate here and which is not? Explain briefly.

My answer: Figure 17 is not appropriate: the interaction is significant, so removing it is a mistake.

Figures 18 and 19 are simple effects of one explanatory variable (`Treatment`) on time to pain relief, for each value of the other one `Gender`. This is an appropriate way to understand interactions.

Eliminate the first Figure, with a reason, and explain why the second and third are appropriate (something brief like "simple effects to understand the interaction" is fine for that). One point for each of those.

(f) (4 marks) What do you conclude from the appropriate analyses out of Figures 17, 18, and 19, in the context of the data? Explain briefly.

My answer: Look at the two simple effects analyses.

For males, there is a significant effect of treatment, and the Tukey shows that `Time` is significantly higher for treatment A than for the other two treatments, which are not significantly different.

For females, there is also a significant effect of treatment, and the Tukey shows this time that treatment C's `Time` is significantly higher than for the other two treatments A and B, which are not significantly different from each other.

Two marks for each of those. In each case I'm looking for an assertion of significant treatment effect, plus a description of what effect the treatment has.

Extra: the fact that the treatments with the highest time are *different* for males and females is what made the interaction significant: you *need* to look at males and females separately to understand what is going on.

Extra extra: the `Time` is time until pain subsides, so a smaller number is better. That means that our recommendations should be treatments B or C for males, treatments A or B for females. (The data doesn't allow us to choose between those; for example, the fact that B's sample mean is slightly higher than A's for females is a statistical fluke. The confidence interval says that the difference could go either way, and thus if we were to do another study, A and B could come out the other way around for females.) I didn't ask the question this way, though, so I didn't need a recommendation of treatment by gender, but there's nothing stopping you providing one if you want. Providing one shows understanding of what's going on, so it will do as a description of the treatment effect in each case. (I didn't care whether you thought that higher or lower was better, as long as you said something about where the significant differences between treatments for males and females (separately) were.)

If you somehow thought that removing the interaction was the way to go, then a sensible conclusion taken from Figure 17 is what you need: a difference between genders, with the time to pain relief being less for males, but no difference among treatments. This is easier than the correct way, though, so a maximum of two points if you say this. Note that taking out the significant interaction obscures the difference in treatments that is actually there (if you look at males and females separately); this way, it just seems as if there is a lot of random variation, but if we do it properly, we can explain a lot of that random variation. Compare the error

mean square (on the `Residuals` line) for the main-effects analysis, 28.70, with the one on the correct analysis with interaction, 9.35. The latter is much smaller, saying that the analysis with interaction is doing a much better job of explaining what is going on.

Question 6 (11 marks)

How moral do people think it is to eat meat? Ninety university professors were asked to rate their answers to this question on a nine-point scale, with 1 being "very morally bad', 5 being neutral and 9 being "very morally good". The professors studied ethics, philosophy but not ethics, and other disciplines. Some of the data are shown in Figure 20. The professors' areas of study are in `discipline`, denoted `Eth` (ethics), `Phil` (philosophy but not ethics), `Other`. The column `discipline` is an R `factor`.

The researchers who collected the data had two research questions in mind:

- Are the Ethics professors and the Philosophy professors different in terms of their attitudes towards eating meat?
- How does the average of the Ethics professors and the Philosophy professors compare with the Other professors?

(a) (2 marks) Explain briefly how contrasts would provide an efficient way of answering the researchers' questions.

My answer: This is the usual thing of wanting to make only certain comparisons (the two listed above) rather than all possible comparisons. By focusing on just those two comparisons, we should be able to get better tests for them (in the sense of giving us the best chance of finding differences if they exist).

(b) (2 marks) Write two contrasts that will represent the researchers' questions of interest, in the same order as above. Call your contrasts `c1` and `c2`. That is, give R code that will create two vectors, one for each contrast, that can be used in an `lm` later to make tests. Note that the `disciplines` are to be taken in alphabetical order.

My answer: This one seemed logically to be out of 2 rather than 3, so I changed it.

Thus the disciplines are `Eth`, `Other` and `Phil` in that order. (One point for saying that, or for implying below that you know that.)

The first contrast is Ethics vs. Philosophy:

```
c1=c(1, 0, -1)
```

One point. Or switch the signs, or use two other numbers of the same size. Make sure the two numbers are of the same size and opposite signs and the middle one is zero.

The second contrast is the average of `Eth` and `Phil` vs. `Other`, so this:

```
c2=c(-1/2, 1, -1/2)
```

or anything that has the first and last numbers equal and the middle one twice as big and the opposite sign. One more point.

If you mess up the alphabetical order, you'll get something for arranging the contrasts in the order you think the disciplines should be. What that something is depends on how easy it is for me to check.

(c) (2 marks) Verify that your two contrasts are orthogonal.

My answer: I have R, so I do:

```
sum(c1*c2)
```

```
## [1] 0
```

Orthogonal, because 0.

You ought to do it with your calculator. My numbers are:

$$(1)(-1/2) + (0)(1) + (-1)(-1/2) = (-1/2) + 0 + (1/2) = 0.$$

If your numbers are different, use what you have, and if you do it right, the calculation will still give you zero. Remember to multiply corresponding entries first and *then* add up. If you add first, you'll *always* get zero, even if they're not orthogonal, because that's how contrasts work.

If you are somehow unable to calculate it, give the same R code that I gave above and say what you would do with the result.

(d) (2 marks) Give R code to set up your two contrasts as contrasts for the model we are about to fit using `lm`. (Two steps. The data frame is called `meat`.)

My answer:

This:

```
m=cbind(c1,c2)
contrasts(meat$discipline)=m
```

One point for each line. The thing inside `contrasts` has to be the grouping variable, *as a*

factor, which is why I told you earlier that it was already turned into a factor.

As the data came to me, it was untidy, with one column for each discipline, so I tidied it with `pivot_longer`. This has an option `names_transform`, which I used to turn the discipline column into a factor so that you wouldn't have any problems later. If the data had already been tidy, `discipline` would have been text as it came in from the file, and I would have had to do something like

```
mutate(discipline=factor(discipline))
```

to make it all work. (I should put one of those on the final!)

(e) (3 marks) What do you conclude from Figure 22, in the context of the data? You may wish to use Figure 21 to give a complete conclusion.

My answer: The first contrast `c1` is not significant (P-value 0.11). This means that there is no significant difference in mean scores between ethics professors and philosophy professors who do not study ethics. One point.

The second contrast `c2` *is* significant (P-value 0.009). This means that the Other professors have a significantly different mean score from the (average of the) Ethics and other-Philosophy professors. One point.

For the last point, look at the table of means in Figure 21 to see that the Other professors have a *higher* mean score than either the Ethics or other-Philosophy professors. This means that the Other professors think that eating meat is less morally bad than either of the other two groups of professors.

Extra: the gaps between `Other` and `Eth` and between `Eth` and `Phil` are about the same. This makes our conclusions a bit surprising: why would `Other` vs. the rest be significant, but `Eth` vs. `Phil` not be? I think the answer is in sample sizes: the second contrast uses all 90 professors, but the first one only uses 60 of them.