STAD29 / STA 1007 assignment 7

Due Tuesday Mar 17 at 11:59pm on Quercus

You will need to load `tidyverse`, `MASS`, and `car`. If you want to avoid problems, load `MASS` first, or, load `conflicted` as well and deal with any conflicts as they occur (the important one being to prefer `dplyr::select`, the `tidyverse` one, over `MASS::select`).

Hand in problems 2 and 4.

1. Work through Chapter 24 of PASIAS.

2. One of the ways to measure the effectiveness of a drug is to measure its concentration in the bloodstream at different times after it is taken. A small study was designed to compare the effectiveness of two different forms of the same drug: a tablet and a capsule. Ten subjects were used and were randomly assigned to the `form` of drug that they would receive (each subject received only one form, either the tablet or the capsule, not both). Each subject was measured at five times after receiving the drug in their assigned form, at 0.5, 1, 2, 3, and 4 hours. At each time, a blood sample was taken, and the concentration of the drug in the subject's bloodstream was measured. The data are in `http://ritsokiguess.site/STAD29/bloodstream.csv`.

   (a) (2 marks) Read in and display (some of) the data. Is the data frame in long or wide format? Explain briefly. (If you prefer, talk about whether the data frame is "tidy" or "untidy".)

Solution: One point for reading and displaying, since that's *very* familiar by now:

```
my_url <- "http://ritsokiguess.site/STAD29/bloodstream.csv"
bloodstream <- read_csv(my_url)
```

```
## Parsed with column specification:
## cols(
##  subject = col_character(),
##  form = col_character(),
##  t0.5 = col_double(),
##  t1 = col_double(),
##  t2 = col_double(),
##  t3 = col_double(),
##  t4 = col_double()
## )
```

```
bloodstream
```

```
## # A tibble: 10 x 7
##    subject form     t0.5    t1    t2    t3    t4
##    <chr>   <chr>   <dbl> <dbl> <dbl> <dbl> <dbl>
##  1 T1      tablet     50    75   120    60    30
##  2 T2      tablet     40    80   135    70    40
##  3 T3      tablet     55    75   125    85    50
##  4 T4      tablet     70    85   140    90    40
##  5 T5      tablet     60    90   150    95    50
##  6 C1      capsule    30    55    80   130    65
```

```
##  7 C2      capsule    25    50    75   125    60
##  8 C3      capsule    35    65    85   140    85
##  9 C4      capsule    45    70    90   145    80
## 10 C5      capsule    50    75    95   160    90
```

This is wide format, because all the observations for each subject (at different times) are in one row. Or you can think of this as untidy and you would rather have each observation (one drug concentration at one time) in a row by itself to make it tidy.

Extra: the reason for the names `pivot_longer` and `pivot_wider` is that Hadley Wickham would rather describe data as "longer" or "wider" rather than absolutely "long" or "wide"; you can often make eg. "wide" data wider still, for example. So I gave you the opportunity to call it "tidy" or "untidy" instead.

(b) (4 marks) Run a repeated-measures ANOVA to see whether concentration (as measured in the columns `t0.5` through `t4`) depends on `form`, time or the combination of both. Remember the steps: create a response variable, run `lm`, create the within-subjects structure, run `Manova` from `car`.

Solution: I'm asking you to get as far as displaying the MANOVA; the interpretation comes up next:

```
response <- with(bloodstream, cbind(t0.5, t1, t2, t3, t4))
times <- colnames(response)
times.df <- data.frame(times)
bloodstream.1 <- lm(response~form, data=bloodstream)
Manova(bloodstream.1, idata=times.df, idesign=~times)
```

```
##
## Type II Repeated Measures MANOVA Tests: Pillai test statistic
##             Df test stat approx F num Df den Df    Pr(>F)
## (Intercept)  1   0.98769   641.68      1      8 6.318e-09 ***
## form         1   0.01023     0.08      1      8     0.781
## times        1   0.99413   211.55      4      5 9.217e-06 ***
## form:times   1   0.99312   180.31      4      5 1.370e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

If you have the interpretation either here or in the next part, that's ok.

(c) (2 marks) What does your MANOVA tell you about the data? Explain briefly, in the context of the data.

Solution: Remember the drill: look at the interaction, and if it's significant, interpret it *and stop*.

Here, the interaction is significant. This means that:

- the pattern of drug concentration over time is different for each form (I think this is best, because you really want to compare the forms of treatment over time)
- the effect of time is different for each form
- the effect of form is different for each time

One of those.

Going on to interpret the main effects now is an *error*. What we have to do is something like simple effects of time for each drug, which I will do as an Extra. You might suspect that there

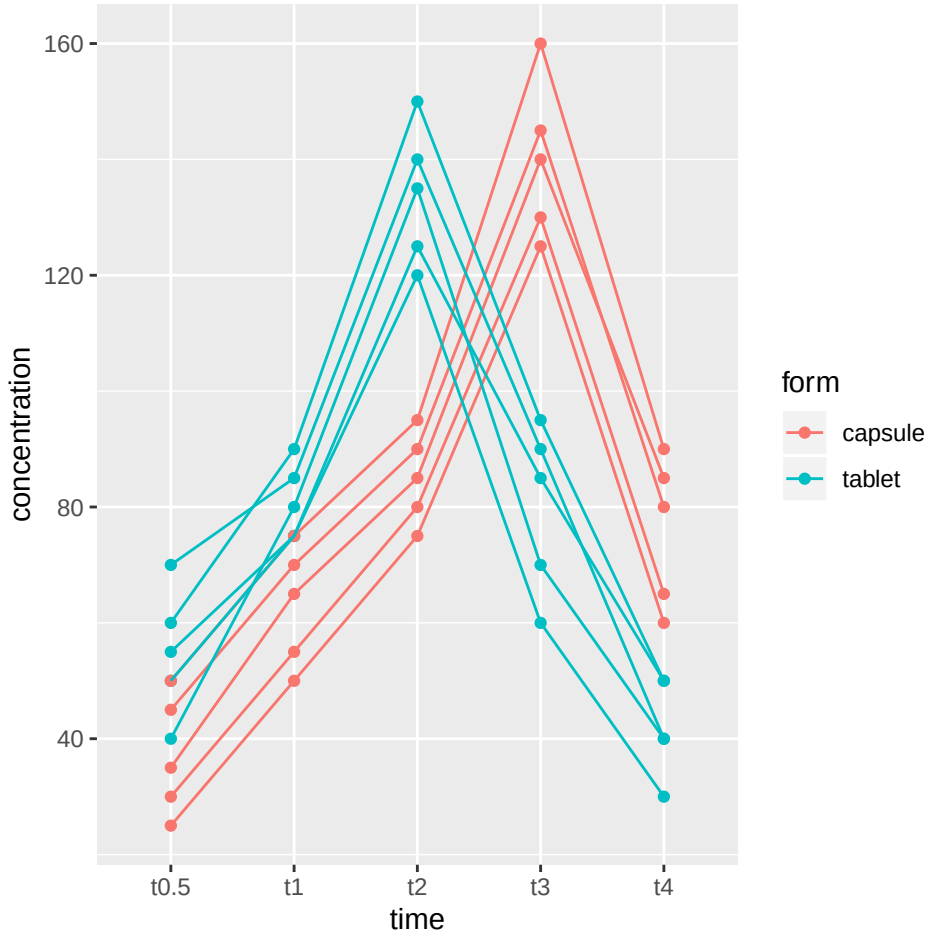
is an effect of time for each drug, but looking at the main effects in the presence of a significant interaction is not the way to do that. (In particular, even though the main effect of `form` is not significant, there *is* an effect of the form of the drug which shows up as a different effect of the pattern over time for each form, rather than a different overall average drug concentration.)

(d) (4 marks) Make a spaghetti plot. That is, plot blood concentration of the drug over time, with the points for each subject being joined by lines, and the lines coloured by the form of the drug that subject received. Do you need wide or long format for your plot?

Solution: The last sentence is a kind of veiled hint that you probably need long format for your plot and you should figure out how to get it. Your thinking ought to be something like "`ggplot` likes having each variable in one column, so I should get that first."

I don't think we need the long format for anything else in this question, so the efficient way to get the graph is to make the long format and pipe it into `ggplot`. However, if you want to make the long format, save it, and then use that in the plot, that's absolutely fine too:

```
bloodstream %>% pivot_longer(starts_with("t"),
                             names_to="time",
                             values_to="concentration") %>%
  ggplot(aes(x=time, y=concentration, colour=form, group=subject)) +
    geom_point() + geom_line()
```



I've split this over several lines so that you can read it, but the formatting doesn't matter. If you saved the output from `pivot_longer`, that data frame should be the first input to `ggplot`.

- Two marks for a suitable `pivot_longer` or `gather`. You can select the columns to gather up in any way that works.
- Two marks for making the plot. Key is to have colour *and* group and for them to be *different*: `group` decides which points are joined by lines, so that needs to be `subject`; `colour` decides what colour the lines will be, so that will be `form`.

You will find this intolerably difficult if you don't `pivot_longer` first. I think it's possible, but it'll be very tedious.

(e) (2 marks) What does your spaghetti plot tell you about the reason for the significance or non-significance of the interaction term? Explain briefly.

Solution: My interaction term was (strongly) significant. I think the reason for that is that the drug concentration peaks at different times; for the tablet, it peaks at 2 hours and then goes down again, and for the capsule, it peaks at 3 hours. This is why the pattern is different over time (and for this kind of study, the patterns are *very* consistent for each form).

Extra: the main effects, though we really shouldn't look at them, are also interpretable in the light of our spaghetti plot:

- the significant time effect is because the drug concentration starts out low, goes up and then comes down again: that is to say, regardless of `form`, the concentration is lowest at the extreme time points and higher some time in the middle.
- the non-significant `form` effect says that, if you average over time, the two forms of the drug are about the same in terms of concentration: they are both between about 20 and about 160.

The issue for the drug company to concern itself with is whether it's important that the peak concentration is achieved more quickly for the tablet than for the capsule. Otherwise, the peak concentration seems to be the same for both forms.

3. Work through Chapter 25 of PASIAS.

4. On a previous assignment, we learned about researchers who are comparing different ways to give technical information about diet. 33 subjects were randomly assigned to one of three groups: technical dietary information from a website; same information from a nurse practitioner; same information from a video. Each subject then made three ratings: difficulty, usefulness, and importance of the information in the presentation.

   The data are in `http://ritsokiguess.site/STAD29/dietary.csv`.

   (a) (1 mark) Again read in and display (some of) the data.

Solution: I am setting up some packages first. I'll need `tidyverse` and `MASS`, and, since I don't want to run into any problems with `select`, I'm going to load `conflicted` as well, and I explain the bottom lines later:

```
library(tidyverse)
library(MASS)
library(conflicted)
conflict_prefer("select", "dplyr")
```

```
## [conflicted] Will prefer dplyr::select over any other package
conflict_prefer("filter", "dplyr")
## [conflicted] Will prefer dplyr::filter over any other package
```

Then, exactly what you did before:

```
my_url <- "http://ritsokiguess.site/STAD29/dietary.csv"
dietary <- read_csv(my_url)
## Parsed with column specification:
## cols(
##  group = col_character(),
##  useful = col_double(),
##  difficulty = col_double(),
##  importance = col_double()
## )
dietary
## # A tibble: 33 x 4
##    group   useful difficulty importance
##    <chr>    <dbl>      <dbl>      <dbl>
##  1 website   19.6       5.15       9.5
##  2 website   15.4       5.75       9.10
##  3 website   22.3       4.35       3.30
##  4 website   24.3       7.55       5
##  5 website   22.5       8.5        6
##  6 website   20.5      10.2        5
##  7 website   14.1       5.95      18.8
##  8 website   13         6.30      16.5
##  9 website   14.1       5.45       8.90
## 10 website   16.7       3.75       6
## # ... with 23 more rows
```

I have again 33 rows (one per person); each person is identified by the group they were in, as well as by the three ratings they gave.

(b) (2 marks) Previously we ran a MANOVA on these data and found a significant result. In this assignment, we aim to find out what the significant result means. To begin, run a suitable discriminant analysis, saving the result. Display the saved result.

Solution: Something like this. My result is called `dietary.3` since I am continuing the numbering from the MANOVA, but you can call it whatever you like. You'll need to run `library(MASS)` first, and you would do well to run it *before* `library(tidyverse)`, or use `library(conflicted)` to deal with any conflicts that arise.

```
dietary.3 <- lda(group~difficulty+importance+useful, data=dietary)
dietary.3
## Call:
## lda(group ~ difficulty + importance + useful, data = dietary)
##
## Prior probabilities of groups:
##     nurse     video   website
## 0.3333333 0.3333333 0.3333333
##
## Group means:
##         difficulty importance   useful
```

```
## nurse     5.581818   5.109091 15.52727
## video     5.372727   5.636364 15.34545
## website   6.190909   8.681818 18.11818
##
## Coefficients of linear discriminants:
##                     LD1         LD2
## difficulty -0.01019651  0.37731940
## importance  0.29440468 -0.14353490
## useful      0.35110259  0.07901745
##
## Proportion of trace:
##    LD1    LD2
## 0.9942 0.0058
```

(c) (2 marks) Would you prefer to look at one, two or more linear discriminants? Explain briefly.

Solution: The place to look is the Proportion of Trace at the bottom: the first linear discriminant is much much much more important than the second, and there are only two of them, so we should only consider the first one.

(d) (3 marks) For each of your proposed linear discriminants, say whether each of the original variables have a positive, negative or zero effect on it, and what kind of values of those variables would make that discriminant score large and positive.

Solution: I said to only look at LD1. Look in the Coefficients of Linear Discriminants table, and make a call about whether each of those LD1 values is positive, negative or zero. I'm going to call `importance` and `useful` positive and `difficulty` zero. This means that large values of `importance` and `useful` will go with a large LD score.

If you thought it was worth looking at LD2 as well, then you need to make a similar call about its coefficients. I would say that `difficulty` has a positive coefficient and the other two are close to zero (make a call on that yourself), which means that for me a large LD2 score would go with a large value of `difficulty`. For you, maybe you are looking for a *small* value of `importance` as well, since you might call its coefficient negative rather than close to zero.

(e) (3 marks) Obtain predicted group memberships and posterior probabilities. Display at least some of them, side by side with the values they are predictions for. Save the results.

Solution: This means passing your LDA results into `predict`, and then `cbind`ing it to the original data. Save first, and then display. I get the whole thing;[1] you will probably get the first ten rows and the first "few" columns, depending how wide your page is:

```
p <- predict(dietary.3)
p_dietary <- cbind(dietary, p)
p_dietary
```

```
##      group useful difficulty importance   class posterior.nurse posterior.video
## 1  website   19.6       5.15        9.5 website      0.02940150      0.04032966
## 2  website   15.4       5.75        9.1 website      0.27890288      0.33212867
## 3  website   22.3       4.35        3.3 website      0.13269694      0.14655871
## 4  website   24.3       7.55        5.0 website      0.01959194      0.01997433
## 5  website   22.5       8.50        6.0 website      0.03746180      0.03652119
## 6  website   20.5      10.25        5.0 website      0.19887791      0.15810808
```

```
## 7  website   14.1       5.95       18.8 website      0.00581090      0.01100582
## 8  website   13.0       6.30       16.5 website      0.04346041      0.06992018
## 9  website   14.1       5.45        8.9   video      0.37155350      0.43531515
## 10 website   16.7       3.75        6.0   video      0.35862026      0.42459062
## 11 website   16.8       5.10        7.4 website      0.28206480      0.32967281
## 12   nurse   17.1       9.00        7.5 website      0.29461066      0.27051210
## 13   nurse   15.7       5.30        8.5 website      0.29194550      0.34901560
## 14   nurse   14.9       9.85        6.0   nurse      0.52583126      0.40508574
## 15   nurse   19.7       3.60        2.9   video      0.35215926      0.38015677
## 16   nurse   17.2       4.05        0.2   nurse      0.52737599      0.45795128
## 17   nurse   16.0       4.40        2.6   nurse      0.50289880      0.47224081
## 18   nurse   12.8       7.15        7.0   nurse      0.50197319      0.46554642
## 19   nurse   13.6       7.25        3.2   nurse      0.56158429      0.43171630
## 20   nurse   14.2       5.30        6.2   video      0.46606927      0.48036622
## 21   nurse   13.1       3.10        5.5   video      0.46255006      0.51885364
## 22   nurse   16.5       2.40        6.6   video      0.31951659      0.42419239
## 23   video   16.0       4.55        2.9   nurse      0.49924028      0.47154659
## 24   video   12.5       2.65        0.7   nurse      0.52857885      0.47055747
## 25   video   18.5       6.50        5.3 website      0.31443138      0.31123069
## 26   video   19.2       4.85        8.3 website      0.06944352      0.09056149
## 27   video   12.0       8.75        9.0   nurse      0.49374878      0.45085724
## 28   video   13.0       5.20       10.3   video      0.35469002      0.44445866
## 29   video   11.9       4.75        8.5   video      0.44531055      0.51226982
## 30   video   12.0       5.85        9.5   video      0.43400138      0.49063034
## 31   video   19.8       2.85        2.3   video      0.37113789      0.40849376
## 32   video   16.5       6.55        3.3   nurse      0.51288997      0.43880323
## 33   video   17.4       6.60        1.9   nurse      0.53095719      0.42858023
##    posterior.website        x.LD1       x.LD2
## 1       0.9302688318  2.044112896 -0.38896379
## 2       0.3889684488  0.445602097 -0.43703162
## 3       0.7207443428  1.174937652  0.41244400
## 4       0.9604337300  2.345001966  1.53389176
## 5       0.9260170104  1.997735576  1.60657888
## 6       0.6430140075  0.983281828  2.25238784
## 7       0.9831832844  2.842854772 -1.85657877
## 8       0.8866194185  1.775942468 -1.48130591
## 9       0.1931313419 -0.066653205 -0.62424305
## 10      0.2167891146  0.009774257 -0.64398943
## 11      0.3882623890  0.443285275 -0.32765551
## 12      0.4348772448  0.538290505  1.15324205
## 13      0.3590388988  0.378878446 -0.49699904
## 14      0.0690829999 -0.684409513  1.51542758
## 15      0.2676839735  0.151957012 -0.01857686
## 16      0.0146727220 -1.525280555  0.34121759
## 17      0.0248603858 -1.243601499  0.03397461
## 18      0.0324803910 -1.099789486  0.18719359
## 19      0.0066994049 -1.938664774  0.83357207
## 20      0.0535645177 -0.824906258 -0.28539492
## 21      0.0185963016 -1.394769800 -1.10194247
## 22      0.2562910124  0.129961539 -1.25529506
## 23      0.0292131319 -1.156809515  0.04751206
```

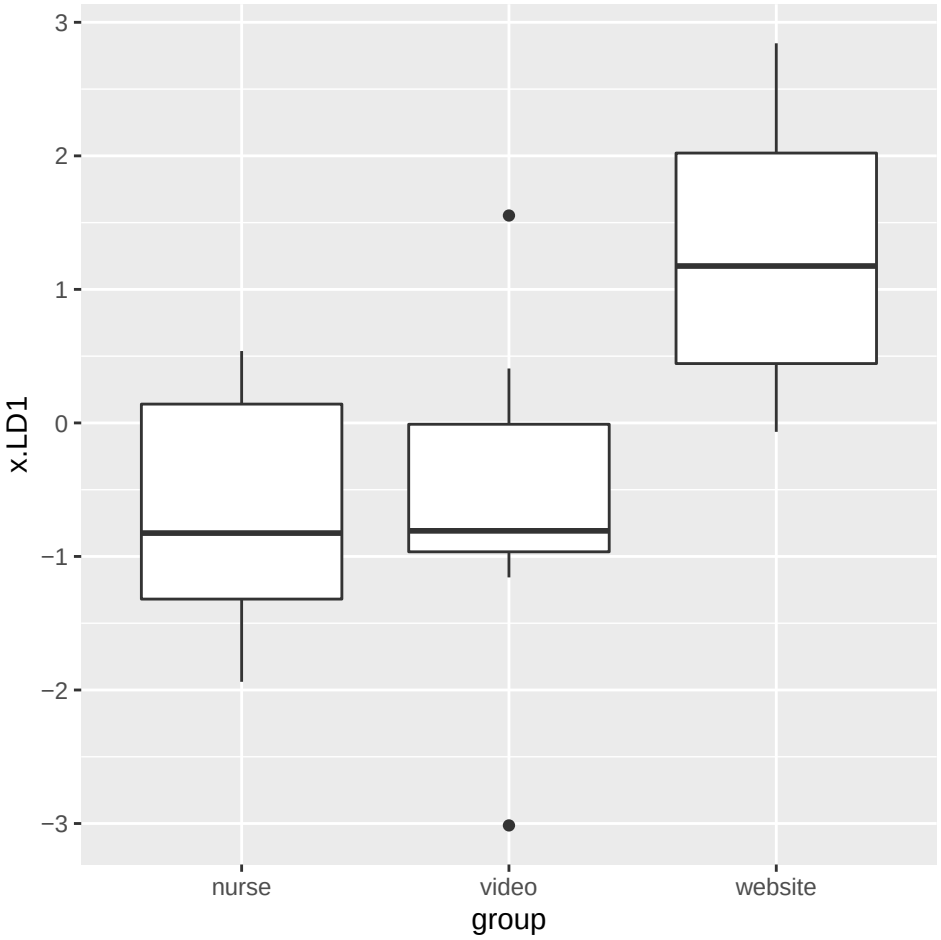
```
## 24      0.0008636734 -3.013985532 -0.63017912
## 25      0.3743379277  0.407635024  0.63634468
## 26      0.8399949897  1.553445389 -0.36152479
## 27      0.0553939860 -0.808176675  0.44062084
## 28      0.2008513201 -0.038150335 -1.00644106
## 29      0.0424196329 -0.949703370 -1.00479110
## 30      0.0753682781 -0.631404456 -0.72537292
## 31      0.2203683594  0.018071266 -0.20754383
## 32      0.0483067932 -0.883889410  0.78424566
## 33      0.0404625803 -0.980573585  1.07517604
```

The first four columns are the original data, and the rest are the predictions: `class` contains the predicted group membership (based on the values of the three quantitative variables), the next three columns are the posterior probabilities of being in each group given the values of the quantitative variables, and the last two columns are the discriminant scores. Probably not all of those will fit on your display.

(f) (3 marks) Make a suitable plot of the discriminant scores against group for your chosen number of discriminants.

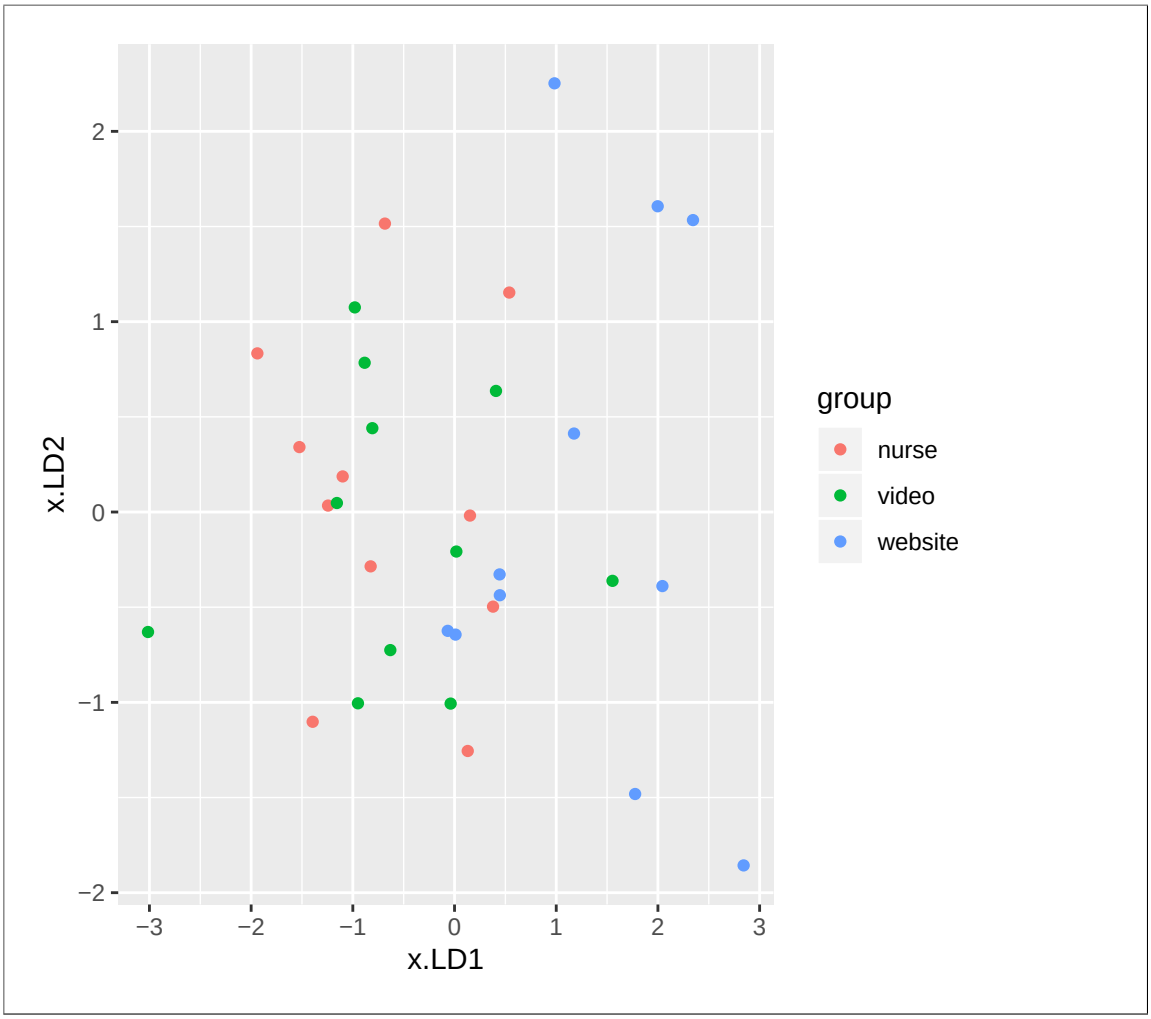
Solution: The kind of plot will depend on how many discriminants you thought were worth looking at. If only one (like me), you have one quantitative discriminant score and one categorical grouping variable (actually called `group`), so side-by-side boxplots is the thing. You'll have to figure out that `x.LD1` is the name the discriminant scores acquired:

```
ggplot(p_dietary, aes(x=group, y=x.LD1)) + geom_boxplot()
```



If you thought you needed two discriminants, you have *two* quantitative discriminant scores to plot, along with one categorical grouping variable, so a scatterplot is the thing, with the groups identified by colour:

```
ggplot(p_dietary, aes(x=x.LD1, y=x.LD2, colour=group)) + geom_point()
```



(g) (2 marks) Comment briefly on how the discriminant scores distinguish the groups, if you think they do.

Solution: On the boxplots, the `website` group typically has the highest LD1 score (comparing, say, the medians), while the other two groups are very similar. There is a certain amount of overlap in the boxplots, so that the LD1 scores do not make a complete distinction of the groups (which suggests that there will be some misclassification later).

If you made a scatterplot, the high LD1 scores usually go with the blue `website` group, on the right. The other two groups, the red and green dots, are all mixed up, so the `nurse` and `video` groups are not distinguishable at all, not in terms of LD1 (which implies not in terms of any of the three variables we measured, but that's coming up). Even though the blue dots are typically on the right, some of them are mixed up with the red and green dots, so, consistently with the boxplots, we'd expect some of the people to be classified into the wrong group.

On the scatterplot, LD1 does at least do *something* to distinguish the groups (the blue dots are mostly on the right), but LD2 does *nothing* to separate the reds, greens and blues; each of those could be at the top or the bottom or in the middle. This is another sign that LD2 is not worth considering at all; the point of the LDs is to separate out the groups, and one that doesn't do this is not worth thinking about.

(h) (2 marks) What do your conclusions above tell you about how the groups are distinguished by their

values on the *original* measured variables, if at all? Explain briefly.

Solution: The `website` group is (for the most part) distinguished by having high (positive) scores on LD1. We said earlier that a high score on LD1 went with large values on `importance` and `useful`, so the reason for the significant MANOVA is that people who got the information from the website found it more important and more useful than the people who got it from the nurse or the video.

Extra: you can now go back to your plot of Assignment 6 and see whether it squares with this. I made a grouped boxplot, and I found that on `difficulty`, all three groups were pretty much the same. But for `importance` (fairly clearly) and `useful` (slightly), the website group had the highest scores on average.

This is why the MANOVA was significant: the `website` group had higher mean scores on `importance` and `useful` compared to the other groups (which were not different from each other).

Extra 2: if we go back to the reason the study was done (in the description on Assignment 6), they were interested in whether distributing the information via the website was as good or better than via the nurse or the video. This was because using the website was cheaper (um, "more cost-effective"). It looks as if the answer to their question is "yes", because people getting the information from the website found it more important and more useful (compared to the people who got it by other means), without finding it any more difficult. So it looks as if the website is a winner.

You might be thinking that this is where the question ought to end, but I wanted to give you some practice with misclassification and posterior probabilities. Coming up.

(i) (2 marks) Make a cross-tabulation of the people actually in each group with the groups they were predicted to be in. To do this, use `table` or `count` as you prefer.

Solution: The actual group is (conveniently) in `group` while the predicted group is in `class`. (The name `class` is always the one `lda` uses for predicted group.)

You might find `count` easier to think about; you need to count up all the combinations of `group` and `class` and see how many observations each combo has:

```
p_dietary %>% count(group, class)
```

```
## # A tibble: 8 x 3
##   group   class       n
##   <chr>   <fct>   <int>
## 1 nurse   nurse       5
## 2 nurse   video       4
## 3 nurse   website     2
## 4 video   nurse       5
## 5 video   video       4
## 6 video   website     2
## 7 website video       2
## 8 website website     9
```

If you like the layout with rows and columns better, you can reshape the above using `pivot_wider`:

```
p_dietary %>% count(group, class) %>%
    pivot_wider(names_from=class, values_from=n, values_fill=list(n=0))
```

```
## # A tibble: 3 x 4
##   group   nurse video website
##   <chr>   <int> <int>   <int>
## 1 nurse       5     4       2
## 2 video       5     4       2
```

```
## 3 website     0     2       9
```

There were no people who actually used the website but were predicted to have seen the nurse. In the wider-format table, this would give a missing value, but I realized that any missing values in the table ought to be zero, so I put the zero in for `values_fill`.

Or you can use `table`. This takes two columns, but it doesn't have a `data=` (like the one-sample `t.test`), so you handle that using `with`:

```
with(p_dietary, table(group, class))
```

```
##          class
## group     nurse video website
##   nurse       5     4       2
##   video       5     4       2
##   website     0     2       9
```

(j) (2 marks) How many people were misclassified: that is, how many people had a different predicted group from their actual group?

Solution: If you have an actual contingency table, total up the values that are off the diagonal: $4 + 2 + 5 + 2 + 0 + 2 = 15$.

If you used `count`, add up the frequencies from the rows where `group` and `class` are different, which gets the same answer.

If you want to be clever, get R to count them. This way is rather slick:

```
p_dietary %>%
    count(group != class)
```

```
## # A tibble: 2 x 2
##   `group != class`     n
##   <lgl>            <int>
## 1 FALSE               18
## 2 TRUE                15
```

The `TRUE` is the number wrongly classified, and the `FALSE` is the number that were correct. You can count anything, not just columns.

Another way is this, based off the original `count`:

```
p_dietary %>%
    count(group, class) %>%
    mutate(wrong=(group != class)) %>%
    filter(wrong) %>%
    summarize(total_wrong=sum(n))
```

```
## # A tibble: 1 x 1
##   total_wrong
##         <int>
## 1          15
```

The logic to this one is to redo the counting, then make a new column that says whether each combo was gotten wrong or not, then grab the rows that were a misclassification, then total up the frequencies of those.

The *misclassification rate* is the number of classifications that were gotten wrong, out of all of them, here:

```
15/33
```

```
## [1] 0.4545455
```

This is rather unimpressively high, but was not helped by two of the groups being indistinguishable from each other. `nurse` and `video` had, on average, the same values for all three variables.

(k) (3 marks) Find a person that was wrongly classified, and display that person's true `group`, predicted `class` and the posterior probabilities for all three groups. It doesn't matter which person you choose. Comment briefly on how close that person was to being predicted correctly.

Solution: The easiest way to tackle this is to display *all* the people that were misclassified, and then pick one:

```
p_dietary %>%
    filter(group != class) %>%
    select(group, class, starts_with("posterior"))
```

```
##      group   class posterior.nurse posterior.video posterior.website
## 1  website   video      0.37155350      0.43531515      0.1931313419
## 2  website   video      0.35862026      0.42459062      0.2167891146
## 3    nurse website      0.29461066      0.27051210      0.4348772448
## 4    nurse website      0.29194550      0.34901560      0.3590388988
## 5    nurse   video      0.35215926      0.38015677      0.2676839735
## 6    nurse   video      0.46606927      0.48036622      0.0535645177
## 7    nurse   video      0.46255006      0.51885364      0.0185963016
## 8    nurse   video      0.31951659      0.42419239      0.2562910124
## 9    video   nurse      0.49924028      0.47154659      0.0292131319
## 10   video   nurse      0.52857885      0.47055747      0.0008636734
## 11   video website      0.31443138      0.31123069      0.3743379277
## 12   video website      0.06944352      0.09056149      0.8399949897
## 13   video   nurse      0.49374878      0.45085724      0.0553939860
## 14   video   nurse      0.51288997      0.43880323      0.0483067932
## 15   video   nurse      0.53095719      0.42858023      0.0404625803
```

The first time I ran this, `conflicted` gave me an error, because I have used `select`, and both the `tidyverse` and `MASS` have a `select`. I got this error:

```
## Error: [conflicted] `select` found in 2 packages.
## Either pick the one you want with `::`
## * MASS::select
## * dplyr::select
## Or declare a preference with `conflict_prefer()`
## * conflict_prefer("select", "MASS")
## * conflict_prefer("select", "dplyr")
```

I figured the `select` I wanted was the one in `dplyr` (part of the `tidyverse`), so I copied the bottom `conflict_prefer` line up to where I loaded all my packages and pasted it in just below `library(conflicted)`. On attempt #2, it noted two copies of `filter`, so I fixed that, and on attempt #3, everything ran.

Pick one of these 15 people. The top two actually used the website, but this is actually the *smallest* posterior probability in each case; the analysis picked `video` because that was (just) the larger of the other two. This is a pretty bad misclassification. On the other hand, if you look at one of the actually-`video` people who was predicted to be in the `nurse` group, the posterior probabilities of `video` and `nurse` are pretty close to each other. These are unlucky misclassifications, you might say.

Notes

[1]This is because using `cbind` to glue a data frame `dietary` to a `matrix` `p` gets you an old-fashioned `data.frame`. In an R Notebook you'll get the first ten rows as usual, but for me I get the whole thing. If it were a `tibble` I'd get the same as you.