# STAD29 / STA 1007 at-home final exam

Due Friday April 17 at 22:00 (10:00pm) on Quercus. Accessability students may hand the exam in later, according to their accommodations. It is your responsibility to submit the exam no later than it is due. Unauthorized late exams may be subject to a penalty.

Answer the four questions below. Hand in your code, output and explanations, by creating an R Notebook and Previewing it to make an HTML file, exactly as you have done for assignments. Hand this in on Quercus, and check, by downloading your file and looking at it, that you have handed in what you intended to hand in.

This exam is open book and open Internet. That means that you may use information from any website, but you may not seek or obtain help from any other person (for example, by asking a question yourself on a forum).

In this exam, "display" means displaying enough of the results to show that you have the right idea. You do not need to display *all* the results. For example, displaying ten lines of a dataframe is enough.

If you have questions, make your best effort to answer them yourself. If you are not clear about what you need to do, make your best assumption about what I am asking you to do, and *state that assumption clearly*. If you get stuck on an early part of a question, and therefore cannot do the later parts, bear in mind that there are no surprises in the early parts of any of the questions, and you may simply need to come back to it later. Get as close as you can to answering the question as asked, or at least getting *an* answer, even if you do it a different way.

If you absolutely must, email me at `ken.butler@utoronto.ca`, but only do so if something is truly impossible, such as a data file that cannot be accessed. If it can be accessed but you cannot read it in, that is *your* problem to solve, and do not expect any help solving it. I will get to emails as soon as I can, but that may not be quickly.

You will need to load `MASS`, `tidyverse`, 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 ones being to prefer `dplyr::select`, the `tidyverse` one, over `MASS::select`, and generally to prefer things from `dplyr` and `tidyr`).

1. In a study on depression, a researcher asked each of 45 patients five questions about their everyday functionality:

   **useful** Have you recently felt that you are playing a useful part in things?

   **contented** Have you recently felt contented with how things are?

   **decisions** Have you recently felt capable of making decisions about things?

   **no_start** Have you recently felt that you were not able to make a start on anything?

   **dread** Have you recently felt yourself dreading everything you have to do?

   The answer to each question was given on a four-point scale, with 1 meaning "definitely no" and 4 meaning "definitely yes". These will be treated as quantitative. The names of the columns in the data set are given in the description above. In addition, each patient was classified as "ill" (depressed) or "well" (not depressed); this is in the column `wellness`. Our aim is to find out whether the responses to the above questions are different for ill and well patients, and, if so, how. There are 45 patients altogether.

   The data are at `http://ritsokiguess.site/STAD29/depression.csv`.

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

**Solution:** Absolutely no surprises here:

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

```
## Parsed with column specification:
## cols(
##   id = col_double(),
##   useful = col_double(),
##   contented = col_double(),
##   decisions = col_double(),
##   no_start = col_double(),
##   dread = col_double(),
##   wellness = col_character()
## )
```

```
depression
```

```
## # A tibble: 45 x 7
##       id useful contented decisions no_start dread wellness
##    <dbl>  <dbl>     <dbl>     <dbl>    <dbl> <dbl> <chr>
##  1     1      2         2         2        3     3 ill
##  2     2      2         2         2        4     3 ill
##  3     3      1         1         2        4     4 ill
##  4     4      2         2         2        4     3 ill
##  5     5      1         1         2        4     3 ill
##  6     6      1         1         2        4     4 ill
##  7     7      2         2         2        3     3 ill
##  8     8      1         1         2        4     3 ill
##  9     9      1         1         2        4     3 ill
## 10    10      2         1         2        4     3 ill
## # ... with 35 more rows
```

Extra: if you are used to questionnaires like this: the last two items were (in my data source) "reverse-coded", with No and Yes reversed. This means that in the original data source a high score was "feeling good" and a low score was "feeling bad". I changed this around to make it clearer that Yes and No were the same way around for all the questions, with the consequence that sometimes a high score goes with being depressed and sometimes a low score.

(b) (3 marks) Run a suitable MANOVA that will address the first part of our aim above. Display the results.

**Solution:**

Make a response, then run it through `manova` or `Manova` as you prefer.

```
response <- with(depression, cbind(useful, contented, decisions,
                                   no_start, dread))
depression.1 <- manova(response~wellness, data=depression)
summary(depression.1)
##           Df  Pillai approx F num Df den Df    Pr(>F)
## wellness   1 0.48468   7.3363      5     39 6.291e-05 ***
## Residuals 43
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

One point a line. Or, if you prefer:

```
response <- with(depression, cbind(useful, contented, decisions,
                                   no_start, dread))
depression.1a <- lm(response~wellness, data=depression)
Manova(depression.1a)
##
## Type II MANOVA Tests: Pillai test statistic
##          Df test stat approx F num Df den Df    Pr(>F)
## wellness  1   0.48468   7.3363      5     39 6.291e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This latter solution requires you to have loaded `car`.

(c) (2 marks) What do you conclude from your MANOVA, in the context of the data?

**Solution:** We reject the null hypothesis (which says that all the variables have the same mean for both the groups) and conclude that the groups differ on one or more of the variables (or some combination of them). Or something equally vague.

What I am looking for is the notion that some or all of the five variables differ for the ill and well patients. "There exist differences" is another way to say that. It's one thing to say that the null hypothesis (all variables have the same means for both the ill and well patients) is rejected, but I'd like you to say something about what that implies for the data.

"Reject the null hypothesis" as a complete answer is not really worth anything, because that doesn't demonstrate any understanding of what MANOVA is for and what you might be able to learn from it.

Extra: having come to this rather vague conclusion, we need to run a discriminant analysis to find out what distinguishes the groups of patients. This we now do. (I was pleased to see that some people recognized that this was therefore a sensible thing to do.)

(d) (3 marks) Run a suitable discriminant analysis and display the results.

**Solution:** Make sure `MASS` is loaded, and:

```
depression.2 <- lda(wellness~useful+contented+decisions+no_start+dread,
                    data=depression)
depression.2
## Call:
```

```
## lda(wellness ~ useful + contented + decisions + no_start + dread,
##     data = depression)
##
## Prior probabilities of groups:
##       ill      well
## 0.3333333 0.6666667
##
## Group means:
##      useful contented decisions no_start    dread
## ill     1.6  1.466667  2.000000 3.666667 3.133333
## well    2.8  2.266667  2.333333 2.400000 2.300000
##
## Coefficients of linear discriminants:
##                  LD1
## useful     0.6463491
## contented  0.3460995
## decisions -0.1384186
## no_start  -0.3450111
## dread     -0.6812305
```

(e) (2 marks) How many linear discriminants do you have? Explain briefly why that is.

**Solution:** There is only one, `LD1`.

This is because there are two groups and five variables, and $\min(2-1, 5) = 1$. More loosely, "because there are two groups".

(f) (2 marks) Which two of your original variables make the biggest contribution to LD1? Explain briefly.

**Solution:** `useful` and `dread`, because their coefficients are farthest away from zero. Or are largest in absolute value, however you want to say it.

I didn't say "largest positive contribution" or anything like that, so include any big-in-size negative ones as well.

If your discriminant analysis came out different, for example if you missed out some variables, then pick the largest two of your coefficients in absolute value.

(g) (2 marks) Based on your answer to the previous part, what would make a patient have a large (and positive) score on LD1?

**Solution:** They would need a *large* score on `useful` (positive coefficient) and a *small* score on `dread` (negative coefficient).

Putting this in more comprehensible terms (which is a good idea): they would have to agree that they felt useful, and disagree that they are dreading everything that they have to do.

Make sure you mention that `useful` needs to be *large* and `dread` needs to be *small*.

(h) (4 marks) Obtain the discriminant scores for all the patients, and make a data frame containing all of: the original data, the predicted group membership, the posterior probabilities, and the discriminant scores. Display (some of) your results. (If you cannot obtain this data frame, you should still be able to do at least some of the remaining parts, but you will probably find it more difficult.)

**Solution:** Use `predict` to get the discriminant scores (one point), use `cbind` to glue the output from `predict` onto the end of the data (two points), display some of the results (one point):

```
depression.3 <- predict(depression.2)
preds <- cbind(depression, depression.3)
preds
```

```
##    id useful contented decisions no_start dread wellness class posterior.ill
## 1   1      2         2         2        3     3      ill  well  0.4483505551
## 2   2      2         2         2        4     3      ill   ill  0.6192821771
## 3   3      1         1         2        4     4      ill   ill  0.9792121686
## 4   4      2         2         2        4     3      ill   ill  0.6192821771
## 5   5      1         1         2        4     3      ill   ill  0.9228974302
## 6   6      1         1         2        4     4      ill   ill  0.9792121686
## 7   7      2         2         2        3     3      ill  well  0.4483505551
## 8   8      1         1         2        4     3      ill   ill  0.9228974302
## 9   9      1         1         2        4     3      ill   ill  0.9228974302
## 10 10      2         1         2        4     3      ill   ill  0.7654026197
## 11 11      2         2         2        4     3      ill   ill  0.6192821771
## 12 12      2         1         2        4     3      ill   ill  0.7654026197
## 13 13      1         1         2        3     3      ill   ill  0.8567483764
## 14 14      1         1         2        4     3      ill   ill  0.9228974302
## 15 15      3         3         2        2     3      ill  well  0.0522992765
## 16 16      4         3         3        2     3     well  well  0.0194830021
## 17 17      3         3         2        2     2     well  well  0.0138290907
## 18 18      3         2         2        2     3     well  well  0.0996582531
## 19 19      4         2         2        3     3     well  well  0.0569450440
## 20 20      2         3         2        2     2     well  well  0.0489296251
## 21 21      2         2         2        3     2     well  well  0.1711730398
## 22 22      3         2         2        4     2     well  well  0.1012559539
## 23 23      3         3         2        4     2     well  well  0.0531825762
## 24 24      2         2         2        3     3     well  well  0.4483505551
## 25 25      3         1         3        1     1     well  well  0.0093733423
## 26 26      2         2         3        4     3     well   ill  0.6824104443
## 27 27      3         2         2        1     3     well  well  0.0524078667
## 28 28      3         2         2        2     2     well  well  0.0273574806
## 29 29      2         2         2        2     4     well   ill  0.6151048497
## 30 30      3         2         4        2     2     well  well  0.0467845117
## 31 31      3         1         3        4     2     well  well  0.2298870320
## 32 32      1         2         2        4     3     well   ill  0.8564795387
## 33 33      3         3         2        1     2     well  well  0.0069579011
## 34 34      2         3         2        1     2     well  well  0.0250614183
## 35 35      3         3         3        1     2     well  well  0.0091707214
## 36 36      2         1         2        2     2     well  well  0.1714837891
## 37 37      4         4         4        1     1     well  well  0.0004220202
## 38 38      2         1         2        2     2     well  well  0.1714837891
## 39 39      4         1         4        1     1     well  well  0.0033953430
## 40 40      3         3         2        3     2     well  well  0.0272993000
## 41 41      2         2         2        4     3     well   ill  0.6192821771
## 42 42      4         2         2        3     3     well  well  0.0569450440
## 43 43      3         3         2        2     2     well  well  0.0138290907
## 44 44      2         3         2        3     3     well  well  0.2883592663
## 45 45      4         3         1        3     2     well  well  0.0057577476
```

```
##    posterior.well         LD1
## 1      0.55164944 -0.57674594
## 2      0.38071782 -0.92175708
## 3      0.02078783 -2.59543625
## 4      0.38071782 -0.92175708
## 5      0.07710257 -1.91420571
## 6      0.02078783 -2.59543625
## 7      0.55164944 -0.57674594
## 8      0.07710257 -1.91420571
## 9      0.07710257 -1.91420571
## 10     0.23459738 -1.26785658
## 11     0.38071782 -0.92175708
## 12     0.23459738 -1.26785658
## 13     0.14325162 -1.56919457
## 14     0.07710257 -1.91420571
## 15     0.94770072  0.76071383
## 16     0.98051700  1.26864432
## 17     0.98617091  1.44194437
## 18     0.90034175  0.41461433
## 19     0.94305496  0.71595232
## 20     0.95107037  0.79559524
## 21     0.82882696  0.10448460
## 22     0.89874405  0.40582259
## 23     0.94681742  0.75192209
## 24     0.55164944 -0.57674594
## 25     0.99062666  1.63756842
## 26     0.31758956 -1.06017572
## 27     0.94759213  0.75962547
## 28     0.97264252  1.09584487
## 29     0.38489515 -0.91296534
## 30     0.95321549  0.81900760
## 31     0.77011297 -0.07869555
## 32     0.14352046 -1.56810621
## 33     0.99304210  1.78695551
## 34     0.97493858  1.14060638
## 35     0.99082928  1.64853688
## 36     0.82851621  0.10339624
## 37     0.99957798  3.18379741
## 38     0.82851621  0.10339624
## 39     0.99660466  2.14549891
## 40     0.97270070  1.09693323
## 41     0.38071782 -0.92175708
## 42     0.94305496  0.71595232
## 43     0.98617091  1.44194437
## 44     0.71164073 -0.23064644
## 45     0.99424225  1.88170099
```

For me, this is an old-fashioned `data.frame`, so I get it all (whether I want it or not).

For you, scroll across to make sure you have everything. There should be a column called `class`, two columns beginning `posterior`, and a column `LD1` with the scores on the one and only discriminant.

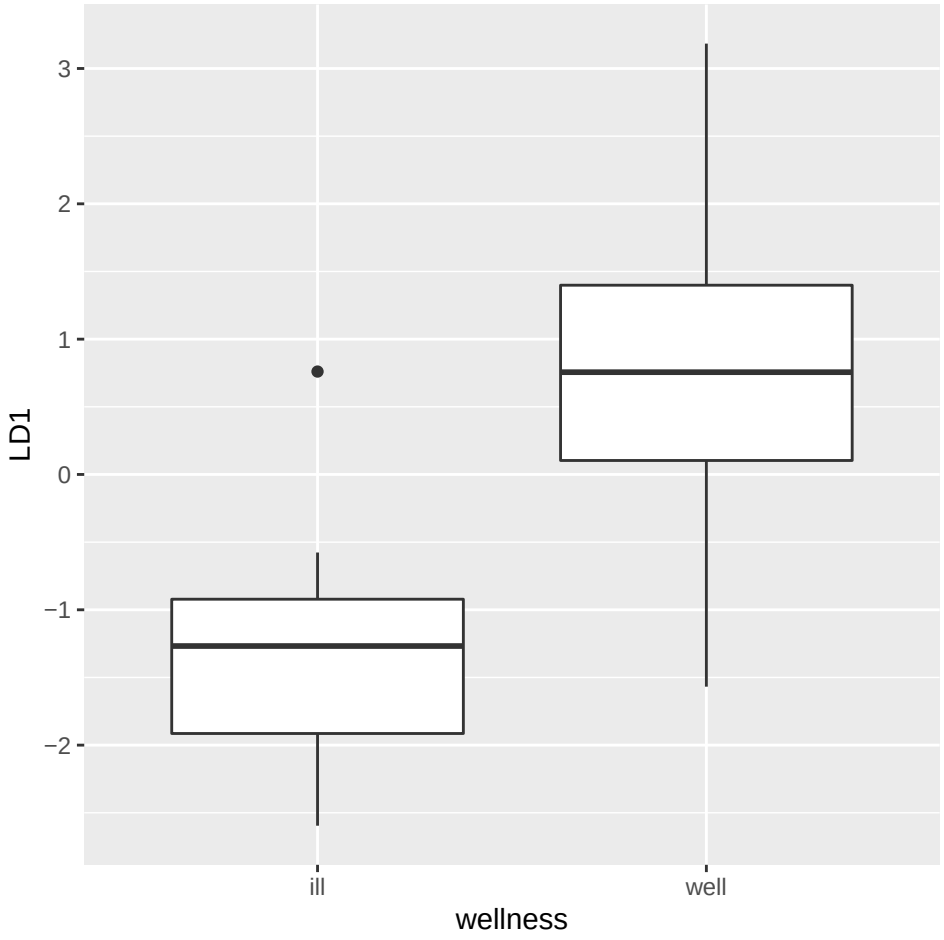
This we use the rest of the way. It is much easier to have everything in one place.

I was at least somewhat relaxed about you missing eg. `class` if you showed later that you knew where to get it from.

(i) (3 marks) Make a suitable plot of the discriminant scores and the original groups.

**Solution:** There is only one discriminant score LD1 (quantitative) and one grouping categorical variable (`wellness`), so a boxplot will display them:

```
ggplot(preds, aes(y=LD1, x=wellness)) + geom_boxplot()
```




(j) (2 marks) Using your plot of the previous part, describe how the groups appear to differ on the discriminant score(s).

**Solution:** There is only one discriminant score LD1, and it is generally higher for the patients that are well, compared to the ones that are ill.

I want a comparison of the two groups (which one is higher). This is not a time to lead off with the Ill group having an outlier. (This is a patient that will be misclassified later, but that's not our concern here.)

(k) (2 marks) Thinking back to what values of the original variables would make the score on LD1 high, does it make practical sense that the groups of patients would compare in the way that they do (as you found in the previous part)? Explain briefly.

**Solution:** "Practical sense" means "based on what you know or can guess about depression". A patient with a high LD1 will tend to answer Yes on `useful` and No on `dread`: that is, they will tend to feel useful, but they will not tend to dread everything. These answers are characteristic of someone who is not depressed, someone with a positive outlook on life. Thus, it makes sense that the Well patients should have high LD1 scores and the Ill patients should have lower ones. (Think, for example, of good and bad mental health, and what that would look like. But make sure you say *something* about that, since that was rather the point.)

Your answer to this will depend on what you said before about things important to LD1. As long as you are consistent, you will be good.

Recall what a high LD1 score means in terms of the original variables, and connect those kinds of answers to what answers you imagine a depressed or a not-depressed person would give. Remember, your job as a statistician is to help the researcher see what the output means *in a way that they can understand*. If you cannot do that, you will not be very helpful as a statistician.

(l) (2 marks) Obtain and display a table showing how many patients are correctly and incorrectly classified.

**Solution:** A couple of ways:

```
with(preds, table(wellness, class))
```

```
##         class
## wellness ill well
##     ill   12    3
##     well   4   26
```

or, more Tidyverse:

```
preds %>% count(wellness, class)
```

```
##   wellness class  n
## 1      ill   ill 12
## 2      ill  well  3
## 3     well   ill  4
## 4     well  well 26
```

or even

```
preds %>% count(wellness, class) %>%
    pivot_wider(names_from=class, values_from=n)
```

```
## # A tibble: 2 x 3
##   wellness   ill  well
##   <chr>    <int> <int>
## 1 ill         12     3
## 2 well         4    26
```

to make it look more like the `table` output.

Or, you can read this as looking for counts of correctly and incorrectly classified and go straight for that:

```
preds %>% count(wellness != class)
```

```
##   wellness != class  n
## 1             FALSE 38
## 2              TRUE  7
```

Either is a reasonable reading of the question. Seven misclassified people, either way.

(m) (2 marks) Would you say that the groups of Well and Ill patients are easy or difficult to distinguish? Explain briefly.

**Solution:** As ever, the thought process is what counts.

My take is that the number of misclassified patients $4 + 3 = 7$ is small compared to the total number of patients (45). Therefore I would say that the groups are fairly easy to distinguish.

Make a call about whether most of the patients are gotten right or too many of them are gotten wrong, and say what that implies about the groups being distinguishable or not.

If you prefer, go back to your boxplot and describe the well and ill patients as distinguishable (or not) based on that.

(n) (3 marks) Display the actual group, predicted group, and the posterior probabilities for the patient with id 7. Two subparts: (i) was this patient misclassified? (ii) was this a clear-cut or close decision? Explain briefly for each subpart.

**Solution:** One point for each of the display plus (i) and (ii):

```
preds %>% filter(id==7) %>%
    select(wellness, class, starts_with("posterior"))
```

```
##   wellness class posterior.ill posterior.well
## 7      ill  well     0.4483506      0.5516494
```

This patient was misclassified because they were actually Ill but were predicted to be Well. The posterior probabilities, though, are close, 0.45 and 0.55, so the decision about which group to classify this patient in was a close one.

Extra: we learned that what distinguishes the groups was being high or low on `useful` and low or high on `dread`. Let's find out where this patient is on those:

```
preds %>% filter(id==7) %>%
    select(useful, dread)
```

```
##   useful dread
## 7      2     3
```

and compare them with the means:

```
preds %>% select(useful, dread) %>%
    summarize_all(~mean(.))
```

```
##   useful    dread
## 1    2.4 2.577778
```

This patient was middling on `useful`, and a little high on `dread`, but there were other `well` patients who scored this high on `dread`, so it was not obvious which group this patient belongs in.

If you look at the other patients, there are some that are very obviously and correctly `well`, such as patient 37, with high scores on the first three questions and low ones on the last two. Also, there are obviously `ill` patients, such as the one with id 14, whose responses are the mirror images of those for patient 37.

2. Some students were asked to document their daily caloric intake once a month for six months. Students were divided into three groups with each receiving instruction in nutrition education using one of three curricula, labelled A through C. We are interested in how caloric intake changes over time, and whether that is different for the different curricula. The data are in `http://ritsokiguess.site/STAD29/nutrition.csv`.

(a) (2 marks) Read in and display (some of) the data. Is this wide or long format? Explain briefly.

**Solution:** One point for the reading in and displaying:

```
my_url <- "http://ritsokiguess.site/STAD29/nutrition.csv"
```

```
nutrition <- read_csv(my_url)
## Parsed with column specification:
## cols(
##  Instruction = col_character(),
##  Student = col_character(),
##  Month = col_character(),
##  Calories = col_double()
## )
nutrition
## # A tibble: 72 x 4
##    Instruction  Student Month Calories
##    <chr>        <chr>   <chr>    <dbl>
##  1 Curriculum A a       M1        2000
##  2 Curriculum A a       M2        1978
##  3 Curriculum A a       M3        1962
##  4 Curriculum A a       M4        1873
##  5 Curriculum A a       M5        1782
##  6 Curriculum A a       M6        1737
##  7 Curriculum A b       M1        1900
##  8 Curriculum A b       M2        1826
##  9 Curriculum A b       M3        1782
## 10 Curriculum A b       M4        1718
## # ... with 62 more rows
```

This is long format. This is because there is one column of nutrition intake values, and a second column showing what number of months each one belongs to. Or, there are several rows for each student, one for each month. Or, wider format would have all the observations for one student in one row. Or something like that. One point for saying this is long format *with a good reason.*

It also works to say that there is only one quantitative column, since if it were wide format there would be more than one.

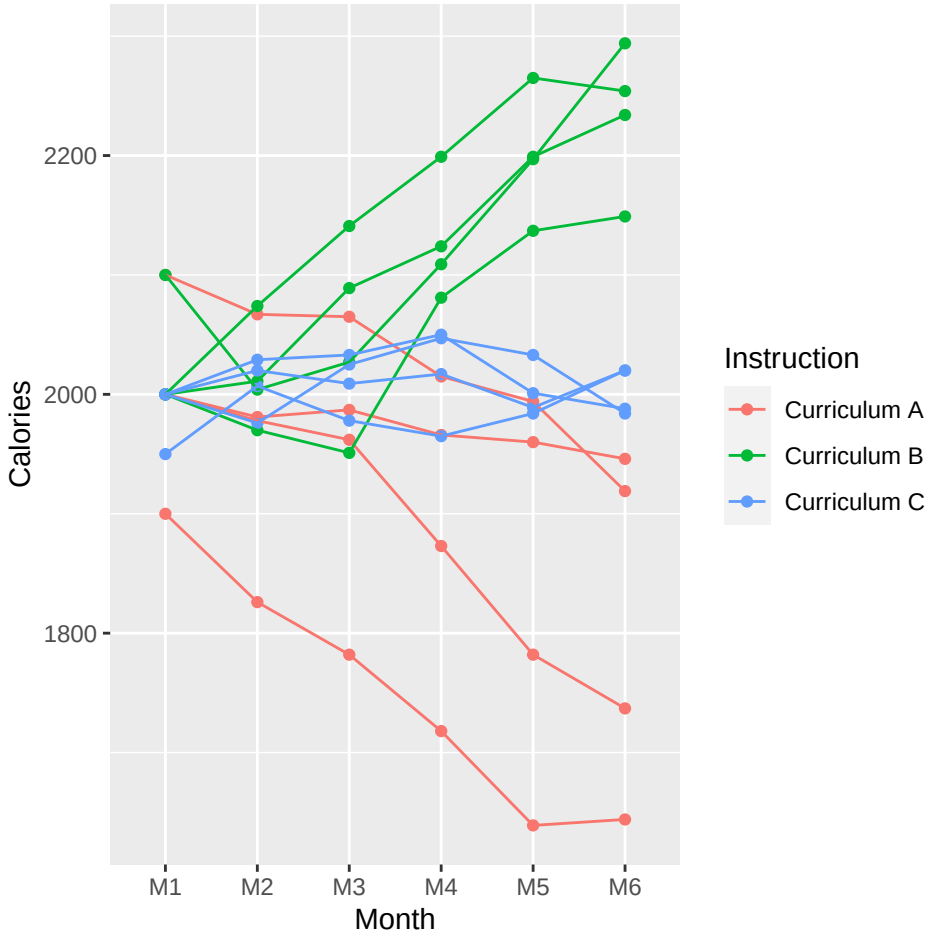
There are lots of other possibilities that work. Find one.

(b) (4 marks) Make a spaghetti plot of these data.

**Solution:** The data frame is already in long format, so there is no need for pivot-anything. The pieces to keep in mind are that the spaghetti plot is a plot of the response variable against time, with the points for each *individual* connected by lines, and the lines for each *treatment* (that is, curriculum) the same colour.

Putting that together gives:

```
ggplot(nutrition, aes(x=Month, y=Calories,
                      colour=Instruction, group=Student)) +
    geom_point() + geom_line()
```




The key is the `colour` and the `group`, and that they are different.

Be sure to know the difference between a spaghetti plot (one trace per subject) and an interaction plot (one trace per treatment, using the treatment mean at each time). You could actually use either here, but I wanted to make sure you knew which was which.

(c) (2 marks) What does your plot suggest about whether there will be a significant interaction between curriculum and time? Explain briefly.

**Solution:** For curriculum A, caloric intake goes down over time, for curriculum B it goes up over time, and for curriculum C it is constant over time (more or less). This says that the patterns over time are different for the different values of `Instruction`; that is, that there will be a significant interaction between `Instruction` and `Months`.

(d) (3 marks) Create a data frame that is in suitable format for a repeated-measures ANOVA, as in the lecture notes. Save and display your new data frame.

**Solution:** To use `Manova` we will need wide format: that is to say, one column with each month's caloric intake in it, and hence one row for each student. This is `pivot_wider` (or, I guess, `spread`):

```
nutrition %>% pivot_wider(names_from = Month,
```

```
                        values_from = Calories) -> nutrition_wide
nutrition_wide
## # A tibble: 12 x 8
##    Instruction  Student    M1    M2    M3    M4    M5    M6
##    <chr>        <chr>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1 Curriculum A a        2000  1978  1962  1873  1782  1737
##  2 Curriculum A b        1900  1826  1782  1718  1639  1644
##  3 Curriculum A c        2100  2067  2065  2015  1994  1919
##  4 Curriculum A d        2000  1981  1987  1966  1960  1946
##  5 Curriculum B e        2100  2004  2027  2109  2197  2294
##  6 Curriculum B f        2000  2011  2089  2124  2199  2234
##  7 Curriculum B g        2000  2074  2141  2199  2265  2254
##  8 Curriculum B h        2000  1970  1951  2081  2137  2149
##  9 Curriculum C i        1950  2007  1978  1965  1984  2020
## 10 Curriculum C j        2000  2029  2033  2050  2001  1988
## 11 Curriculum C k        2000  1976  2025  2047  2033  1984
## 12 Curriculum C l        2000  2020  2009  2017  1989  2020
```

(e) (4 marks) Run a suitable repeated-measures ANOVA and display your results.

**Solution:** Make a response variable. You'll have to type out the names of all 6 columns:

```
response <- with(nutrition_wide, cbind(M1, M2, M3, M4, M5, M6))
response
##         M1   M2   M3   M4   M5   M6
##  [1,] 2000 1978 1962 1873 1782 1737
##  [2,] 1900 1826 1782 1718 1639 1644
##  [3,] 2100 2067 2065 2015 1994 1919
##  [4,] 2000 1981 1987 1966 1960 1946
##  [5,] 2100 2004 2027 2109 2197 2294
##  [6,] 2000 2011 2089 2124 2199 2234
##  [7,] 2000 2074 2141 2199 2265 2254
##  [8,] 2000 1970 1951 2081 2137 2149
##  [9,] 1950 2007 1978 1965 1984 2020
## [10,] 2000 2029 2033 2050 2001 1988
## [11,] 2000 1976 2025 2047 2033 1984
## [12,] 2000 2020 2009 2017 1989 2020
```

Obtain the within-student structure:

```
times <- colnames(response)
times.df <- data.frame(times)
```

Run the model as an `lm`:

```
nutrition.1 <- lm(response~Instruction, data=nutrition_wide)
```

Run `Manova` on that:

```
Manova(nutrition.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   0.99897   8730.1      1      9 9.342e-15 ***
## Instruction        2   0.61292      7.1      2      9   0.01397 *
## times              1   0.35894      0.6      5      5   0.73006
## Instruction:times  2   1.40562      2.8     10     12   0.04532 *
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(f) (1 mark) Was your suspicion about the interaction confirmed? (No explanation needed.)

**Solution:** Mine was: I expected an interaction, and it was indeed significant.
For you, an answer of "yes" (or "no" if appropriate) is all you need. If you couldn't get an answer in (e), the right answer here is "I don't know" or "I cannot tell".

3. 30 individuals are measured on two variables `x` and `y`. Our aim is to find clusters of similar individuals, using K-means clustering. The data are in `http://ritsokiguess.site/STAD29/xy.csv`.

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

**Solution:**

```
my_url <- "http://ritsokiguess.site/STAD29/xy.csv"
xy <- read_csv(my_url)
## Parsed with column specification:
## cols(
##   x = col_double(),
##   y = col_double()
## )
xy
## # A tibble: 30 x 2
##        x     y
##    <dbl> <dbl>
##  1    25    79
##  2    34    51
##  3    22    53
##  4    27    78
##  5    33    59
##  6    33    74
##  7    31    73
##  8    22    57
##  9    35    69
## 10    34    75
## # ... with 20 more rows
```

Two variables `x` and `y`.

(b) (3 marks) Obtain a K-means cluster analysis with 5 clusters. Do this by running the analysis 20 times, and taking the best result. Display the output.

**Solution:** The second sentence means to use `nstart`:

```
xy.5 <- kmeans(xy, 5, nstart = 20)
xy.5
## K-means clustering with 5 clusters of sizes 7, 6, 6, 7, 4
##
## Cluster means:
##          x        y
## 1 43.14286 12.28571
```

```
## 2 54.00000 53.00000
## 3 30.83333 74.66667
## 4 51.00000 32.00000
## 5 27.75000 55.00000
##
## Clustering vector:
##  [1] 3 5 5 3 5 3 3 5 3 3 2 4 4 2 2 4 4 2 2 2 4 1 1 1 1 1 4 4 1 1
##
## Within cluster sum of squares by cluster:
## [1] 488.2857 590.0000 146.1667 554.0000 172.7500
##  (between_SS / total_SS =  90.0 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

I used the name `xy.5` because I am making 5 clusters. The name is of course your choice.

If you want to use the pipeline way from class, even though there are here no non-numeric columns to get rid of, that's fine too:

```
xy %>% select_if(is.numeric) %>%
    kmeans(5, nstart = 20) -> xy.5
xy.5
```

```
## K-means clustering with 5 clusters of sizes 6, 4, 6, 7, 7
##
## Cluster means:
##          x        y
## 1 30.83333 74.66667
## 2 27.75000 55.00000
## 3 54.00000 53.00000
## 4 51.00000 32.00000
## 5 43.14286 12.28571
##
## Clustering vector:
##  [1] 1 2 2 1 2 1 1 2 1 1 3 4 4 3 3 4 4 3 3 3 4 5 5 5 5 5 4 4 5 5
##
## Within cluster sum of squares by cluster:
## [1] 146.1667 172.7500 590.0000 554.0000 488.2857
##  (between_SS / total_SS =  90.0 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

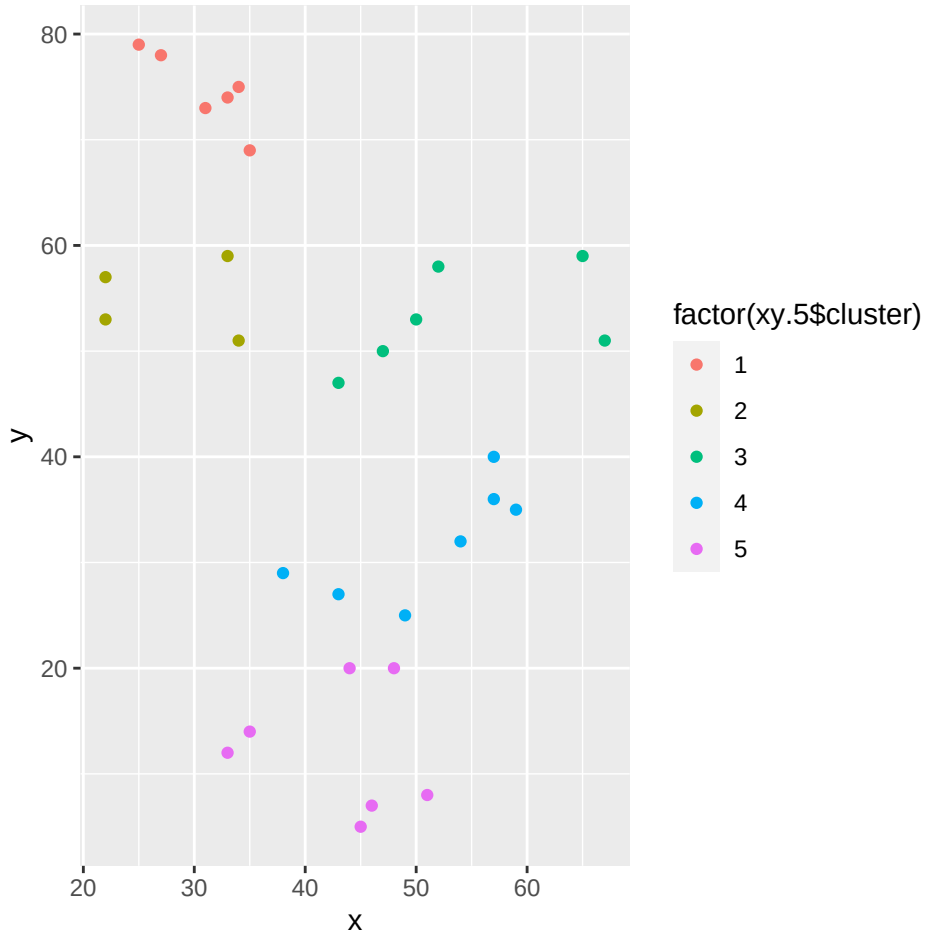
Make it work.

It's up to you whether you think the variables are on the same scales or not. If you think not, `scale` them first. I'm happy either way.

(c) (2 marks) There are only two variables in our data frame, so we can plot the results. Make a suitable plot of your 5-cluster solution.

**Solution:** A scatter plot with the cluster membership indicated by colour. There are a couple of ways to go. Either grab the cluster memberships directly from the output:

```
ggplot(xy, aes(x=x, y=y, colour=factor(xy.5$cluster))) + geom_point()
```



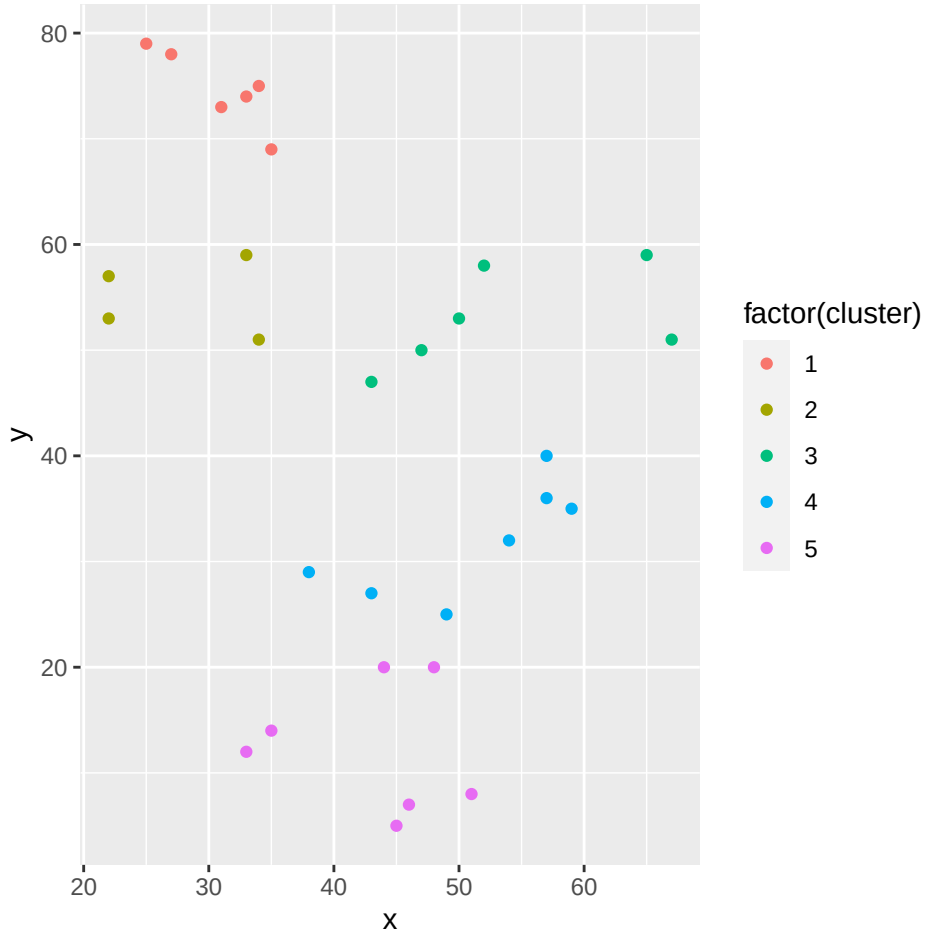

Or make a new data frame and plot that:

```
xy %>% mutate(cluster=xy.5$cluster)
```

```
## # A tibble: 30 x 3
##        x     y cluster
##    <dbl> <dbl>   <int>
##  1    25    79       1
##  2    34    51       2
##  3    22    53       2
##  4    27    78       1
##  5    33    59       2
##  6    33    74       1
##  7    31    73       1
##  8    22    57       2
##  9    35    69       1
## 10    34    75       1
## # ... with 20 more rows
```

or

```
cbind(xy, cluster=xy.5$cluster) %>%
    ggplot(aes(x=x, y=y, colour=factor(cluster))) + geom_point()
```




If your colours are shades of blue, that is because you treated cluster as a number. Wrap it in `factor` to make it categorical, and then you'll get colours.

Your colours don't need to be the same as mine, but the clusters ought to be the same. (If they are not, but you appear to have done the right thing, I am happy.)

(d) (4 marks) Make a scree plot. To do this, first obtain the total within-cluster sum of squares for each number of clusters from 1 to 20, and then plot them. Use the function `ss` from the lecture notes if you wish.

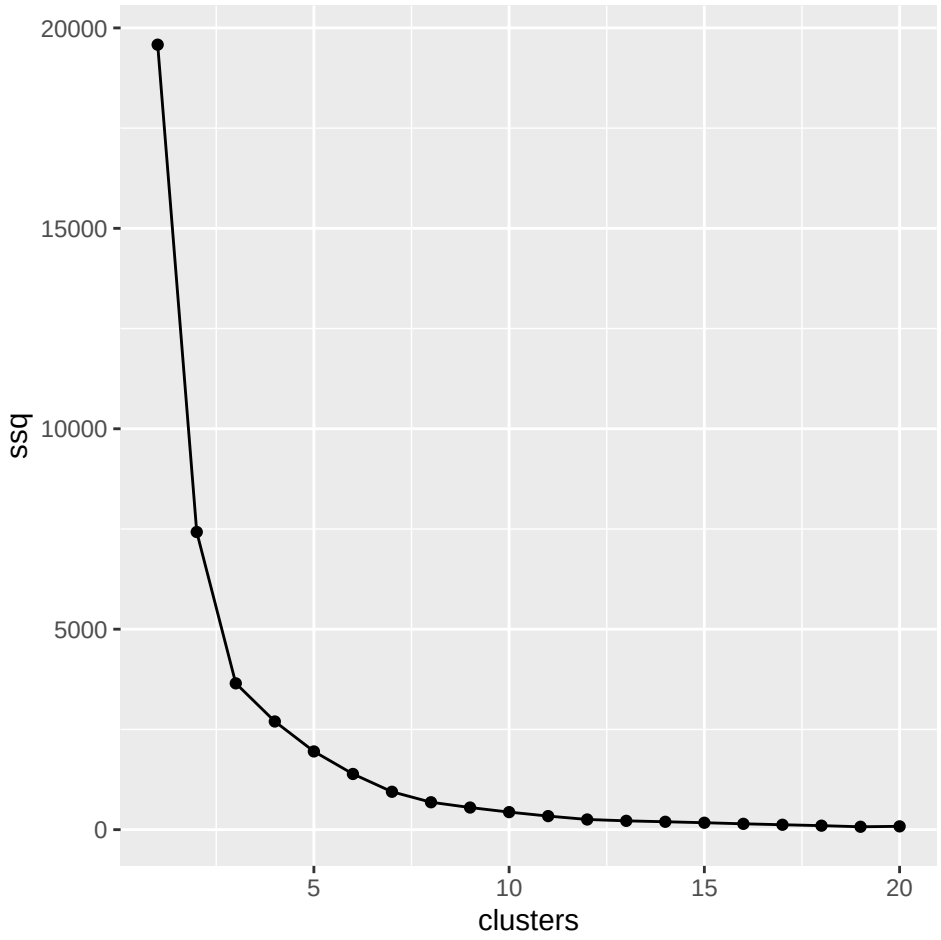
**Solution:** Using the function from the lecture notes:

```
ss <- function(i, d) {
  d %>%
    select_if(is.numeric) %>%
    kmeans(i, nstart = 20) -> km
    km$tot.withinss
}
```

and then:

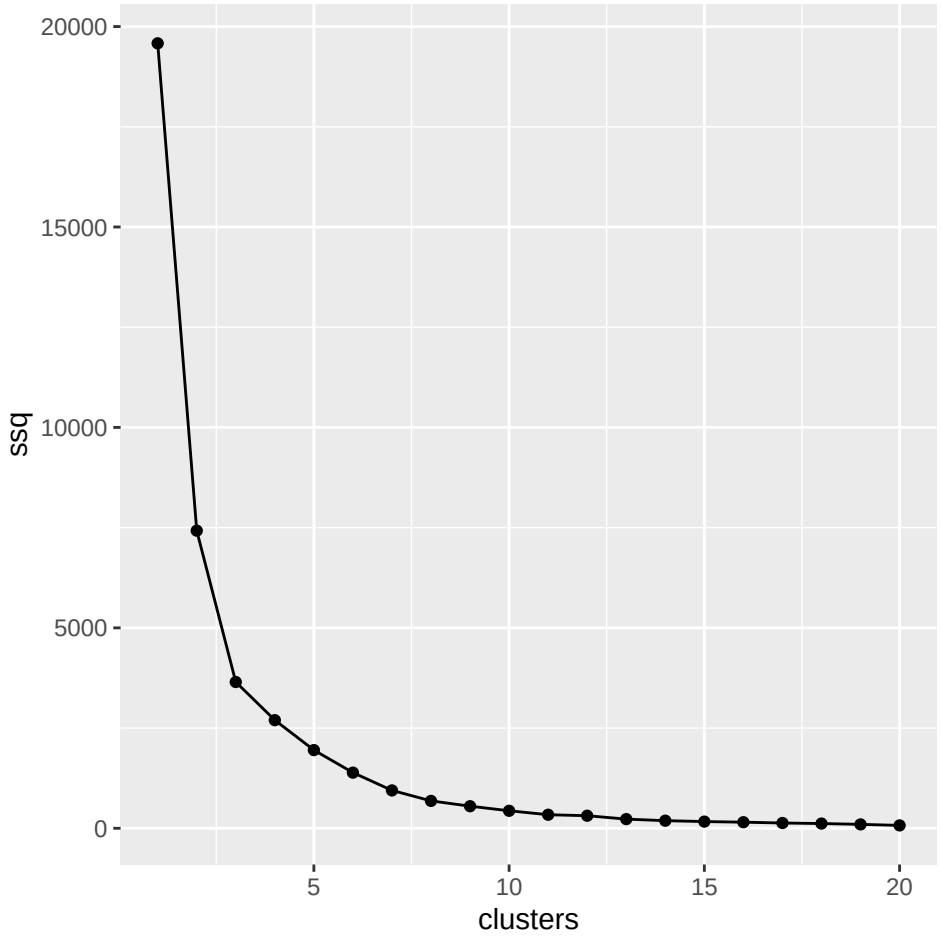
```
tibble(clusters=1:20) %>%
```

```
    mutate(ssq=map_dbl(clusters, ~ss(., xy))) %>%
    ggplot(aes(x=clusters, y=ssq)) + geom_point() + geom_line()
```




This data frame is simple enough, and for that matter the function is simple enough, that you can also do it without defining a function:

```
tibble(clusters=1:20) %>%
    mutate(ssq=map_dbl(clusters,
                       ~kmeans(xy, ., nstart = 20)$tot.withinss)) %>%
    ggplot(aes(x=clusters, y=ssq)) + geom_point() + geom_line()
```




Basically, if you can get the plot by some means of your own, I'm happy.

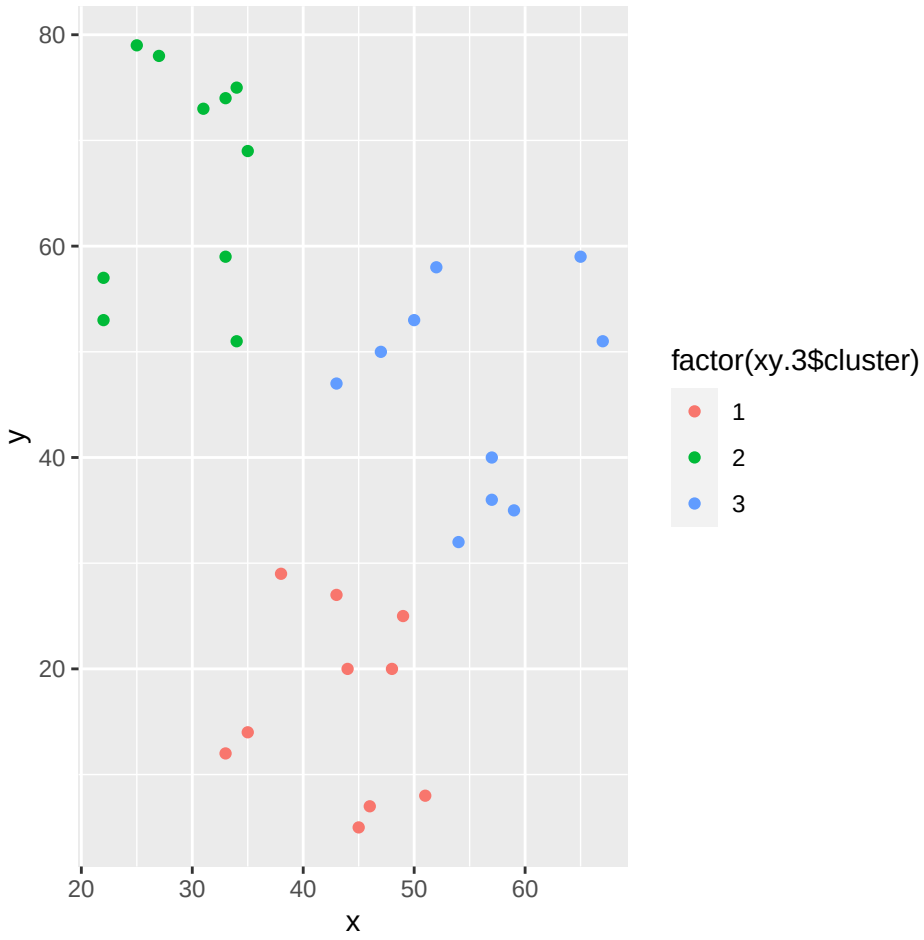
(e) (2 marks) What would be a suitable number of clusters? Explain briefly.

**Solution:** Look for an elbow on the plot. I see a clear one at 3 clusters, and maybe another at 7 or 8. You might see an elbow at 13 clusters, but with 30 observations, this is too many clusters for much insight.

(f) (2 marks) Obtain a K-means cluster analysis with your chosen number of clusters, and make a plot of the data with these clusters displayed. Feel free to re-use your code from earlier parts. (If you were unable to get a scree plot, pick any number of clusters different from 5 for this part.)

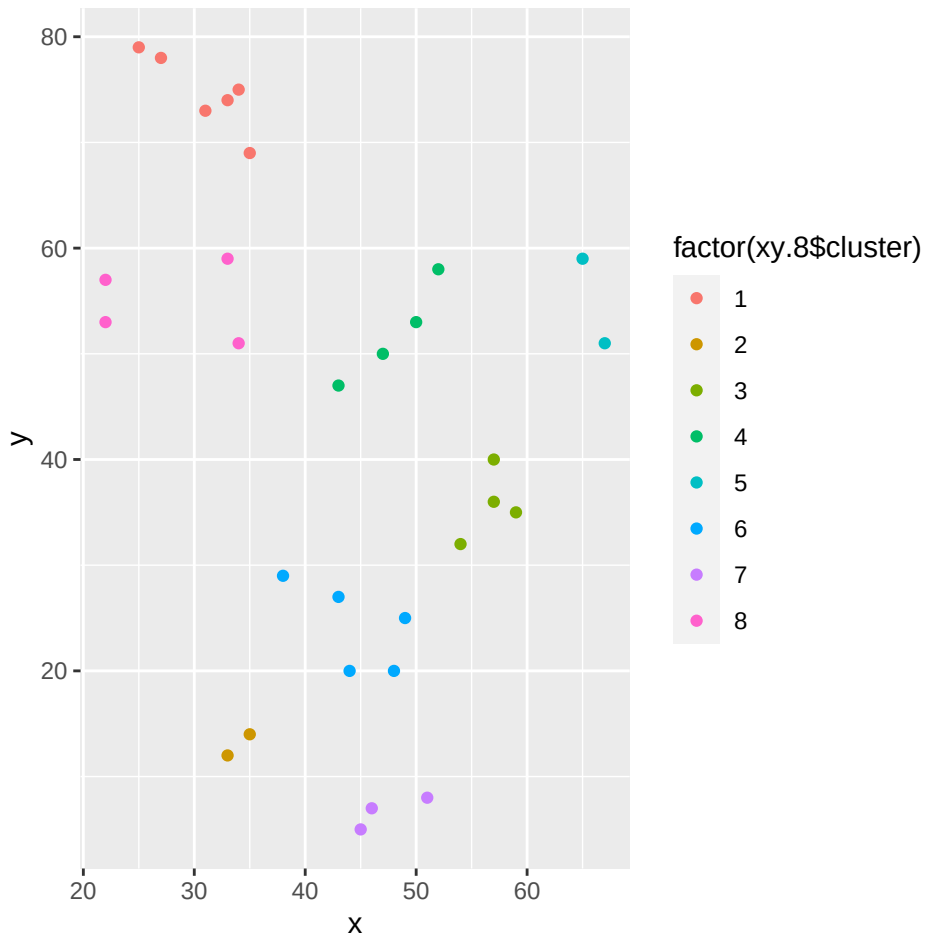
**Solution:** This is mainly copy, paste and edit, and hence only two points. I'm going for just three clusters:

```
xy.3 <- kmeans(xy, 3, nstart = 20)
ggplot(xy, aes(x=x, y=y, colour=factor(xy.3$cluster))) + geom_point()
```




I was going to finish by asking you to say why you thought your preferred number of clusters was better than the 5 you had before, but it's really a matter of opinion. Here is 8:

```
xy.8 <- kmeans(xy, 8, nstart = 20)
ggplot(xy, aes(x=x, y=y, colour=factor(xy.8$cluster))) + geom_point()
```




I'm not sure how these are better than 5 in a way that can be clearly articulated. They are mostly just different. (The two shades of green are different clusters.)

4. 2,121 individuals were studied for four years. Each individual was classified by personality type (I or II: see below), cholesterol level (normal or high), and diastolic blood pressure (normal or high). The main interest of the researchers was to see whether blood pressure was associated with either of the other factors.

   Personality type I is someone who shows signs of stress, worry, and hyperactivity. Personality type II is someone who is normally relaxed, with normal levels of activity.

   The data are in `http://ritsokiguess.site/STAD29/assoc.csv`. There are four columns: the three categorical variables described above, and `frequency`, which is the number of individuals with each combination of levels of the categorical variable.

   (a) (1 mark) Read in and display the data.

**Solution:** There are only eight rows ($2^3$ combinations of levels), so you'll be able to display it all:

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

```
## Parsed with column specification:
## cols(
##  personality = col_character(),
##  cholesterol = col_character(),
##  diastolic = col_character(),
##  frequency = col_double()
## )
assoc
## # A tibble: 8 x 4
##   personality cholesterol diastolic frequency
##   <chr>       <chr>       <chr>         <dbl>
## 1 I           High        High             25
## 2 I           High        Normal          207
## 3 I           Normal      High             79
## 4 I           Normal      Normal          716
## 5 II          High        High             22
## 6 II          High        Normal          186
## 7 II          Normal      High             67
## 8 II          Normal      Normal          819
```

(b) (3 marks) Fit a suitable initial log-linear model, and display the `drop1` output with `test="Chisq"`.

**Solution:** The suitable initial model is the one with the three-way interaction:

```
assoc.1 <- glm(frequency~personality*cholesterol*diastolic,
               data=assoc, family=poisson)
drop1(assoc.1, test="Chisq")
## Single term deletions
##
## Model:
## frequency ~ personality * cholesterol * diastolic
##                                    Df Deviance    AIC     LRT Pr(>Chi)
## <none>                                  0.00000 69.448
## personality:cholesterol:diastolic  1  0.61327 68.062 0.61327   0.4336
```

(c) (4 marks) Remove anything that can be removed, fit a model without what you removed, and repeat these steps as necessary. How do you know when to stop?

**Solution:** The previous `drop1` says that the three-way interaction is not significant and can be removed. Use `update` to remove it, and then run `drop1` again:

```
assoc.2 <- update(assoc.1, .~.-personality:cholesterol:diastolic)
drop1(assoc.2, test="Chisq")
## Single term deletions
##
## Model:
## frequency ~ personality + cholesterol + diastolic + personality:cholesterol +
##     personality:diastolic + cholesterol:diastolic
##                          Df Deviance    AIC    LRT Pr(>Chi)
## <none>                        0.6133 68.062
## personality:cholesterol   1   4.5627 70.011 3.9494  0.04689 *
## personality:diastolic     1   2.9799 68.428 2.3666  0.12395
```

```
## cholesterol:diastolic    1   2.0626 67.511 1.4493  0.22864
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This time the `cholesterol:diastolic` term has the highest P-value, so this is removed:

```
assoc.3 <- update(assoc.2, .~.-cholesterol:diastolic)
drop1(assoc.3, test="Chisq")
## Single term deletions
##
## Model:
## frequency ~ personality + cholesterol + diastolic + personality:cholesterol +
##     personality:diastolic
##                          Df Deviance    AIC    LRT Pr(>Chi)
## <none>                        2.0626 67.511
## personality:cholesterol  1   6.1842 69.632 4.1216  0.04234 *
## personality:diastolic    1   4.6014 68.050 2.5388  0.11108
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Now, the `personality:diastolic` term is not significant, so it is removed:

```
assoc.4 <- update(assoc.3, .~.-personality:diastolic)
drop1(assoc.4, test="Chisq")
## Single term deletions
##
## Model:
## frequency ~ personality + cholesterol + diastolic + personality:cholesterol
##                          Df Deviance     AIC     LRT Pr(>Chi)
## <none>                          4.60   68.05
## diastolic                 1  1651.83 1713.28 1647.23  < 2e-16 ***
## personality:cholesterol  1     8.72   70.17    4.12  0.04234 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

and now everything is significant so we stop.

(d) (2 marks) Is there a significant association between diastolic blood pressure and any of the other variables? Explain briefly.

**Solution:** To see if there is, look for any remaining interaction terms with `diastolic` in them. There are none. This means that there is no significant association between anything else and `diastolic`.

(e) (3 marks) Describe the nature of all the remaining associations. You may wish to obtain frequency tables to help you do this.

**Solution:** There is a significant association between personality type and cholesterol level. (One point.) It's usually easiest to use `xtabs` to investigate what kind of relationship it is:

```
xtabs(frequency~personality+cholesterol, data=assoc)
##             cholesterol
## personality High Normal
##           I   232    795
##           II  208    886
```

People of personality type I are a little more likely to be high cholesterol and people of person-

ality type II are a little more likely to be low cholesterol. Or anything equivalent.

There are almost the same number of people of each personality type, so the comparison works fine with the actual frequencies, but you can make them into proportions if you want:

```
xt <- xtabs(frequency~personality+cholesterol, data=assoc)
prop.table(xt, margin=1)
```

```
##             cholesterol
## personality      High    Normal
##           I  0.2259007 0.7740993
##           II 0.1901280 0.8098720
```

The `margin=1` works out row proportions, appropriate for comparing the number of high and normal cholesterol people for each personality type.

Two more points for some sensible analysis of *how* personality type and cholesterol level are associated.

The `diastolic` "main effect" doesn't tell us much:

```
xtabs(frequency~diastolic, data=assoc)
```

```
## diastolic
##   High Normal
##    193   1928
```

This just says that most people had normal diastolic blood pressure. It doesn't say anything about any associations between `diastolic` and anything else.