University of Toronto Scarborough
Department of Computer and Mathematical Sciences
STAC32 (K. Butler), Final Exam
December 19, 2019

Aids allowed (printed or handwritten): My lecture overheads (slides); Any notes that you have taken in this course; Your marked assignments; My assignment solution; Non-programmable, non-communicating calculator.

This exam has 84 numbered pages of questions. Check to see that you have all the pages. There is an additional empty page that you can use if you need more space for any answers.

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.

Question 1 (14 marks)

"Time-of-day pricing" is a plan by which electricity customers are charged at a higher rate for using electricity at peak hours (hours at which there is a large demand for electricity), and a lower rate for use at off-peak hours. A study was carried out by a large electricity company to measure customer satisfaction with various pricing schemes. The study consisted of two factors: price ratio (how many times more expensive peak-hours electricity is than off-peak electricity; ratios of 2:1, 4:1 and 8:1) and peak period length (6, 9, or 12 hours).

For each combination of price ratio and peak period length, known as a "plan", four customers were randomly selected and charged for electricity use according to that plan for a certain time. At the end of this time period, they were given a questionnaire that assessed satisfaction with the plan they had been on. The questionnaire results were summarized into an overall level of satisfaction, with a minimum (worst) value of 10 and a maximum (best) value of 38. The data are shown in Figure 2.

(a) (4 marks) The data are stored in a file on your SAS Studio called `timeofday.txt`. Give SAS code to read in and display the data set.

My answer: This is entirely straightforward. The data file is delimited by single spaces, so this will work (at least for me; see below):

```
proc import
  datafile="/home/ken/timeofday.txt"
  out=timeofday
  dbms=dlm
  replace;
  getnames=yes;
  delimiter=" ";

proc print;
```

Obs	length	ratio	satisfaction
1	06-hours	2-1	25
2	06-hours	2-1	26
3	06-hours	2-1	28
4	06-hours	2-1	27
5	06-hours	4-1	31
6	06-hours	4-1	26
7	06-hours	4-1	29
8	06-hours	4-1	27
9	06-hours	8-1	24
10	06-hours	8-1	25
11	06-hours	8-1	28
12	06-hours	8-1	26
13	09-hours	2-1	26
14	09-hours	2-1	29
15	09-hours	2-1	27
16	09-hours	2-1	30
17	09-hours	4-1	25
18	09-hours	4-1	30
19	09-hours	4-1	24
20	09-hours	4-1	26
21	09-hours	8-1	33
22	09-hours	8-1	25
23	09-hours	8-1	28
24	09-hours	8-1	27
25	12-hours	2-1	22
26	12-hours	2-1	25
27	12-hours	2-1	20
28	12-hours	2-1	21
29	12-hours	4-1	33
30	12-hours	4-1	25
31	12-hours	4-1	27
32	12-hours	4-1	27
33	12-hours	8-1	30
34	12-hours	8-1	26
35	12-hours	8-1	31
36	12-hours	8-1	27

Minus one per error, to a minimum of one if you got something substantial correct. (If we think it's a small error, you might lose only 0.5.)

- In the `datafile` line, you need to use something that looks like *your* username, not mine. (If you use `ken` and your actual name does not look like Ken, you'll lose a mark because you are copying and not thinking. If your real name is Ken, add something to your username to distinguish it from mine.)
- The data set name on the right side of `out=` can be anything.
- `dbms` must be `dlm`, and you need to have the `delimiter` line as well.
- The `replace`, the `getnames`, and the `delimiter` lines are the only ones in `proc import` that have semicolons on them.
- Last, I asked you to display the data set so you need the `proc print`.

(b) (3 marks) What SAS code would display the mean satisfaction score for each combination of peak period length and price ratio? (It is fine if your output will include other things as well as the mean.)

My answer: My parenthetical remark at the end means that this can be a straight `proc means`. I don't think we've seen one with a `class` line like this, but the clue is that `length` and `ratio` are both categorical, so they *both* go in the `class`:

```
proc means;
class length ratio;
var satisfaction;
```

```
                              The MEANS Procedure

                        Analysis Variable : satisfaction

                   N
length    ratio  Obs   N         Mean      Std Dev      Minimum      Maximum
-----------------------------------------------------------------------------
06-hours  2-1      4   4   26.5000000    1.2909944   25.0000000   28.0000000

          4-1      4   4   28.2500000    2.2173558   26.0000000   31.0000000

          8-1      4   4   25.7500000    1.7078251   24.0000000   28.0000000

09-hours  2-1      4   4   28.0000000    1.8257419   26.0000000   30.0000000

          4-1      4   4   26.2500000    2.6299556   24.0000000   30.0000000

          8-1      4   4   28.2500000    3.4034296   25.0000000   33.0000000

12-hours  2-1      4   4   22.0000000    2.1602469   20.0000000   25.0000000

          4-1      4   4   28.0000000    3.4641016   25.0000000   33.0000000

          8-1      4   4   28.5000000    2.3804761   26.0000000   31.0000000
-----------------------------------------------------------------------------
```

All the lines need semicolons (if you forget *all* of them, you should only be penalized once).

This also works (and is therefore also correct). The next part gets into why this works:

```
proc means;
class length ratio;
```

```
                                  The MEANS Procedure

                            Analysis Variable : satisfaction

                    N
length    ratio   Obs   N          Mean        Std Dev        Minimum        Maximum
--------------------------------------------------------------------------------------
06-hours  2-1       4   4    26.5000000      1.2909944     25.0000000     28.0000000

          4-1       4   4    28.2500000      2.2173558     26.0000000     31.0000000

          8-1       4   4    25.7500000      1.7078251     24.0000000     28.0000000

09-hours  2-1       4   4    28.0000000      1.8257419     26.0000000     30.0000000

          4-1       4   4    26.2500000      2.6299556     24.0000000     30.0000000

          8-1       4   4    28.2500000      3.4034296     25.0000000     33.0000000

12-hours  2-1       4   4    22.0000000      2.1602469     20.0000000     25.0000000

          4-1       4   4    28.0000000      3.4641016     25.0000000     33.0000000

          8-1       4   4    28.5000000      2.3804761     26.0000000     31.0000000
--------------------------------------------------------------------------------------
```

If you want *only* the mean, that goes like this (and is also correct):

```
proc means mean;
  class length ratio;
  var satisfaction;
```

```
                         The MEANS Procedure

                   Analysis Variable : satisfaction

                                    N
              length      ratio    Obs            Mean
              --------------------------------------
              06-hours    2-1        4      26.5000000

                          4-1        4      28.2500000

                          8-1        4      25.7500000

              09-hours    2-1        4      28.0000000

                          4-1        4      26.2500000

                          8-1        4      28.2500000

              12-hours    2-1        4      22.0000000

                          4-1        4      28.0000000

                          8-1        4      28.5000000
              --------------------------------------
```

Once again, without the `var` line also works.

(c) (2 marks) Do you have a `var` line in your code for the previous part? Do you need one? Explain briefly.

My answer: It doesn't matter whether you have one or not, because `proc means` will by default compute means for *all* the quantitative variables. Here, the only one is `satisfaction`, so getting means for "all" of them (that is, the only one) is just fine.

There are no marks for saying whether *you* have one or not; that is to guide your thinking. The two marks are for saying that it doesn't matter whether you have one or not, and for saying why that is. Or, first, for showing that you know what the `var` line is for, and second, for saying what happens if you don't have it. This means knowing what `proc means` will do if you don't give it a `var` line, and also knowing which variables in your data set are quantitative. As I did on the (2019) midterm, I gave the categorical variables values with text in them to make it clearer that they were categorical.

Saying that you *need* the `var` line, in order to say which variable you want means for, is sort-of true, but incomplete (for reasons discussed above). One point.

(d) (3 marks) The electricity company wanted to see how customer satisfaction depended on the *combination* of peak period length and price ratio. To do this, they made the graph shown in Figure 23. (This is at the end of the Figures because it is in colour.) Give the SAS code that was used to make this graph.

My answer: This is a grouped boxplot (as on the midterm, in fact, but that was in R). The code I used is this:

```
proc sgplot;
  vbox satisfaction / category=length group=ratio;
```

with the output shown in the Figure (I actually used this code to generate that output).

You need a `proc sgplot` with a `vbox satisfaction` to get one point. To get two, you need a `category=length`, or a `group=ratio`, or both of those with the category and group switched, or with some other error (eg. forgetting the equals signs). To get all three, you need all of the above. You really ought to have both semicolons as well, but I'm willing to forgive you that if you have everything else correct.

If you have a `category=ratio` and *no* `group`, you only get one (two errors).

You can tell which categorical variable is `category` and which one is `group`: the `category` is the thing on the x-axis, as on an ordinary boxplot, and the `group` determines the legend and the colours. If you only remember how to draw an ordinary boxplot, give code to draw that (with `category=length`) and you'll get two points if you do it right.

If you were drawing this graph for yourself, you might remember that you typically have `category` be the categorical variable with more levels and `group` be the one with fewer (like the sports and gender example in class). For this data set, though, both categorical variables have three levels, and so you could draw them either way around. The point *in this question*, though, is not that; I want you to work out what I did to draw *this* graph, which means knowing how you distinguish between the variable on the x-axis and the one making the colours. For yourself, you might randomly choose one categorical variable for each, and flip them around if you don't like what came out, but I would like your thinking to be a bit clearer than that here.

I decided to make this plot this way around because it seemed clearer to me to think (next part) about assessing the effect of the price ratio for each peak period length, rather than the other way around.

(e) (2 marks) Look again at Figure 23. The peak period lengths are labelled "06", "09" and "12" so that they come out in a sensible order on the graph. For all the price ratios, the "average" cost of electricity is the same; this means that when the ratio is 8–1, the peak price is highest and the off-peak price is lowest, compared to all the other ratios. On the boxplot boxes, SAS uses the symbol O, X or + (one symbol for each colour of box) to denote the mean of the data in that box.

Describe one thing you can conclude from Figure 23 about average (mean or median) satisfaction as it depends on peak period length and/or price ratio, and explain briefly why your conclusion makes sense, based on what you know or can guess about electricity prices.

My answer: This is very open-ended. The first mark is for correctly concluding something from the graph, and the second is for explaining why it's what you'd expect.

I'm guessing that most people will look at the much lower satisfaction for the 2–1 ratio with a

12-hour peak period. This can be explained by something like: if you have a long peak period, electricity had better be much cheaper during the (short) off-peak period, and a ratio of 2–1 is not enough to keep consumers happy.

Or you can compare the three boxes for the 12-hour peak period and say that it is better to have a higher ratio if the peak period is long, which means that consumers can save a lot of money by shifting consumption to the off-peak period. That is, “if the peak period is long, off-peak electricity had better be cheap”. This applies to the previous one as well.

Or you can look at one of the other peak period lengths and say something like “if the peak period is shorter, the different price ratios are close to equally preferable”, bearing in mind that there are only four observations per box.

If you focus on something like “if the peak period is 6 hours, the preferred price ratio is 4–1”, you might then have a hard time explaining why it’s the middling price ratio that has the highest satisfaction. You could try phrasing it in terms of people who use mainly peak electricity (who would be less satisfied if the price ratio is high), or people who will be able to use off-peak electricity (who will be less satisfied if the price ratio is low, since then they miss out on the chance to use cheap electricity). A 4–1 price ratio is kind of a compromise, looked at this way.

Perhaps the preceding argument works more easily if you look at a 9-hour peak period: the peak users want the ratio to be low, and the off-peak users want the ratio to be high.

Another way to go is to compare the three blue boxes: “if the ratio is low, consumers prefer a shorter peak period”, or compare the three green ones: “if the ratio is high, consumers prefer a longer peak period”. You can argue that these both make sense if customers have some ability to switch usage to off-peak (eg., running laundry at night), but are only motivated to do so if there is a big enough price difference; if there is, they actually prefer the peak period to be longer.

I expect I’ll be fairly relaxed about what I consider to be a reasonable “conclusion from the graph”, the first mark, but I will probably be more stringent about what constitutes an explanation of it making sense. Thus it’s pretty easy to get one, but a fair bit harder to get two.

No points for discussion of shape. With only four observations per box, these are very much the kind of thing you would get if the data really were normal. (An analysis might be a two-way ANOVA, which we have not done in this course, but look at in D29.) There are, in any case, no outliers and not badly unequal spread.

Extra: electricity companies do this because it is more expensive to generate extra electricity at peak times; to do so involves firing up coal-fired power stations, or otherwise using rarely-used generation systems. When demand is generally low, some more electricity can be generated cheaply using systems that are currently running (or, say, renewables). It is thus in the electric company’s interest to get people using less electricity during peak times and more during off-peak times, and they can do it by making off-peak electricity cheaper, maybe much cheaper. (This experiment was designed to answer the “how much cheaper” part.)

Question 2 (15 marks)

Bulimia is an eating disorder. People who suffer from bulimia have an unrealistic body image, and will consume a lot of food at one time, followed by feelings of guilt or shame. Such people often have a fear of being evaluated negatively by others.

25 female students took part in a study. 11 of them suffered from bulimia and the other 14 had "normal" eating habits. Each of the students also completed a questionnaire called FNE which evaluates the "fear of negative evaluation", a higher score on the FNE indicating a greater fear. We are interested in finding out whether people suffering from bulimia tend to have a greater fear of negative evaluation than people who do not.

The data are shown in Figure 3.

(a) (4 marks) A suitable t-test is run, with output shown in Figure 4 and Figure 5. Give the SAS code that produced all of this output.

My answer: The immediate thing to remember is that `proc ttest` produces all the graphs *as well*, so that writing any `proc sgplot` in your code is an error. (I actually *don't know* how to produce exactly these graphs with `proc sgplot`, so that it is very likely that you don't know either.)

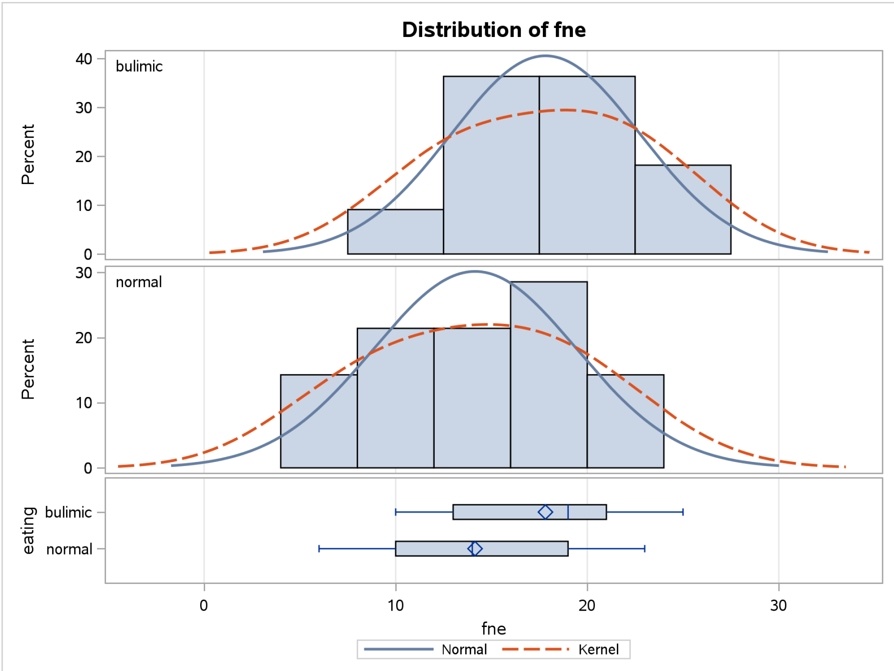
Here is what I did:

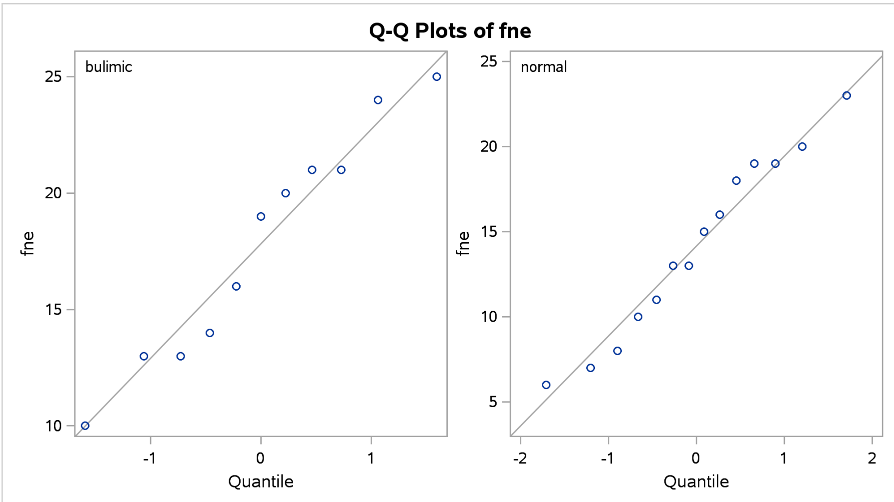
```
proc ttest sides=U;
  class eating;
  var fne;
```

This produces

```
eating          N        Mean     Std Dev     Std Err     Minimum     Maximum

bulimic        11     17.8182      4.9157      1.4821     10.0000     25.0000
normal         14     14.1429      5.2894      1.4137      6.0000     23.0000
Diff (1-2)             3.6753      5.1303      2.0670
   eating        Method               Mean       95% CL Mean        Std Dev

   bulimic                         17.8182     14.5158  21.1206      4.9157
   normal                          14.1429     11.0888  17.1969      5.2894
   Diff (1-2)    Pooled             3.6753      0.1327  Infty        5.1303
   Diff (1-2)    Satterthwaite      3.6753      0.1602  Infty

              eating        Method            95% CL Std Dev

              bulimic                         3.4346   8.6266
              normal                          3.8346   8.5215
              Diff (1-2)    Pooled            3.9873   7.1965
              Diff (1-2)    Satterthwaite
          Method           Variances        DF    t Value    Pr > t

          Pooled           Equal            23       1.78    0.0443
          Satterthwaite    Unequal      22.284       1.79    0.0432
                           Equality of Variances

              Method      Num DF    Den DF    F Value    Pr > F

              Folded F        13        10       1.16    0.8305
```

plus all the graphs:





You need, for the four marks, all of:

- the `sides=U` on the `proc ttest` line. `bulimic` is before `normal` alphabetically, and we're trying to prove that the mean FNE score for bulimic females is higher than for "normal" ones. Hence, a one-sided test with `sides=U`. I expect a lot of people are going to forget this.
- the `class` line with the categorical variable `eating`.
- the `var` line with the quantitative variable `fne`.

Minus one per error, including minus one for drawing any graphs. I guess just writing `proc ttest` and getting everything else wrong is one, as long as you don't try to draw any graphs as well.

(b) (2 marks) The statistician involved with this study decided to run a t-test. Why do you think she decided to do this, rather than using some other test, for these data? Explain briefly.

> **My answer:** Look at the two normal quantile plots at the bottom of Figure 5. Both distributions of FNE scores are approximately normal, since they more or less follow the lines. A less precise but still justifiable answer is to look at the histograms at the top of this Figure and say that they have an approximately normal shape (or say that the blue normal curves and the red kernel density curves are pretty close, which you'll have to wave your hands a bit to say).
>
> Equal spreads don't come into *this* part (that will be part of your consideration in the next one). Nothing for that, because you can run a t-test whether or not the spreads are equal.
>
> Also, "because she wanted to compare the group means" is a reason for running a t-test *in general*, not in this particular case. One point. Being an applied statistician means knowing enough theory to know whether it applies to the data in front of you. Knowing the theory is only half (or less) of your job.
>
> Extra: the way I structured this, we are taking the t-test (or one of them) as a given, because I said that this is what the statistician decided to do, rather than giving you a choice. This question is "this is what was done; explain why it was done". If I had given you the choice, you could conceivably have asked for a Mood's median test, which I didn't want to get into, since this was a question about `proc ttest`. I actually think that a t-test is just fine here, because those wiggles in the normal quantile plots are not indicating a failure of normality. They are, if anything, the kind of S-bends that indicate *short* tails, a departure from normality that is not a problem because the mean is still a good measure of centre. A perhaps better indication is that the endmost points are on or very close to the lines.
>
> Extra extra: if I had done the median-and-IQR thing to put the lines on the normal quantile plots, the lines might have been steeper and then the endmost points would have looked further off the line, but they would still have been *less* extreme than the normal, so the mean and t-test would still have been good.

(c) (3 marks) What do you conclude from Figure 4, in the context of the data? Explain briefly. In your answer, you should give a P-value and justify why that P-value is appropriate.

> **My answer:** Your choice here is between the P-values next to Pooled and next to Satterthwaite.
>
> To decide which one, look at the spreads of the distributions of the two groups. You can look at the standard deviations at the top of Figure 4, or you can look at the mini-boxplots (the horizontal ones) under the histograms in Figure 5. For me, the spreads (either way) are almost the same, so I would use the Pooled test. (You have another option, which is to look at the test at the bottom of Figure 4, which tests, and fails to reject, that the two groups have the same variance.) To my mind, this is not the kind of situation where you should be declaring the two groups to have different spreads, because they only differ by a tiny amount that could easily be chance.
>
> So, look at the Pooled P-value of 0.0443. This is less than 0.05, so we reject the null hypothesis that the two groups have the same mean, in favour of a one-sided alternative that the mean FNE score for the bulimic students is higher than for the "normal" ones. There is a clue here that the test is one-sided: the two confidence intervals for the difference in means both go up

to infinity, rather than having a proper upper limit as you might expect. This, in combination with the last sentence of the preamble (the actual question, before part (a)), means that I expect you to make a one-sided conclusion.

Points: one for comparing the spreads and concluding that they are about the same, *for a reason that you give*: that is to say, tell me what you were looking at to conclude that the spreads were about the same. A second for citing the P-value for the pooled t-test and deciding to reject. A third for making a one-sided conclusion in the context of the data. Expect only about one point for making a two-sided conclusion.

Extra: you'll note that *once again* it makes almost no difference whether you do a Pooled or a Satterthwaite test; the P-values are very similar and the conclusion is the same. For the first point, you could also argue that the Satterthwaite (Welch) test is pretty good whether or not the spreads are equal. I'm good with this justification as long as you follow through with 0.0432 as your P-value. I think, with these data, this is the *only* way you can justify using Satterthwaite.

(d) (2 marks) Describe a population to which it would make sense to generalize these results.

My answer: The data came from female students, so you would be entitled to claim something like "all female students" or "all female students at this college". The key thing is to ask yourself what this data set is, or might be, a sample from.

If you can make the case that these students are (like) a random sample of all women, then you could claim "all women" as your population, but you would need to make that case.

I think one mark for each of "female" and "students", or if you can make a different case, two marks for making that case successfully.

(e) (4 marks) The data as it originally came to me is shown in Figure 6. This is a SAS data set, and so `Obs` labels the rows; it is not a real column. Give SAS code that will create a new data set that looks like Figure 3. (It is OK to have some extra missing values in the new data set, and it is OK to have the observations in a different order as long as they are all there.) The data set shown in Figure 6 is called `negevalwide`; your new data set should be called `negevallong`.

My answer: This is like page 186 of the SAS notes, but simpler because there are only two groups. I would be quite happy with the "tedious way", since there is not too much repetition:

```
data negevallong;
  set negevalwide;
  eating='bulimic';
  fne=bulimic;
  output;
  eating='normal';
  fne=normal;
  output;
  keep eating fne;
proc print;
```

You don't need the proc print; you can have it or not. That produces this:

Obs	eating	fne
1	bulimic	21
2	normal	13
3	bulimic	13
4	normal	6
5	bulimic	10
6	normal	16
7	bulimic	20
8	normal	13
9	bulimic	25
10	normal	8
11	bulimic	19
12	normal	19
13	bulimic	16
14	normal	23
15	bulimic	21
16	normal	18
17	bulimic	24
18	normal	11
19	bulimic	13
20	normal	19
21	bulimic	14
22	normal	7
23	bulimic	.
24	normal	10
25	bulimic	.
26	normal	15
27	bulimic	.
28	normal	20

To make it look like Figure 3, the new columns need to have the right names. So I'm looking for:

- the new data set having the right name
- bring in the values from the wide one
- `eating` has the right name
- `fne` has the right name
- two sections in which:
  - `eating` has a text value that is the name of the column you are looking at (in single or double quotes)
  - `fne` has a numeric value in it that is the *value* of the column you are looking at
  - these two values are then `output` to the new data set
- `keep` just the two new columns

I think there are eight things there altogether, including the names of the new columns. According to this, the two sections the same have to be both correct, but the grader may decide to be more generous. I didn't want this part to be worth too much, so 0.5 for each one you get right.

This has the same data values as in Figure 3, except that they are in a different order and there are some missing values near the end, both of which are immaterial differences.

If you want to do it using an array, it goes thus:

```
data negevallong;
  set negevalwide;
  array eat_array [2] bulimic--normal;
  do i=1 to 2;
    fne=eat_array[i];
    eating=vname(eat_array[i]);
    output;
  end;
  keep eating fne;
proc print;
```

You get to choose the name of the array; what I called `eat_array` can have any name.

Strictly speaking there are two dashes between `bulimic` and `normal` (it's only one dash in a thing like `x1-x8`, where you have variables with the same name but different numbers). But if you have one dash that's fine.

Obs	fne	eating
1	21	bulimic
2	13	normal
3	13	bulimic
4	6	normal
5	10	bulimic
6	16	normal
7	20	bulimic
8	13	normal
9	25	bulimic
10	8	normal
11	19	bulimic
12	19	normal
13	16	bulimic
14	23	normal
15	21	bulimic
16	18	normal
17	24	bulimic
18	11	normal
19	13	bulimic
20	19	normal
21	14	bulimic
22	7	normal
23	.	bulimic
24	10	normal
25	.	bulimic
26	15	normal
27	.	bulimic
28	20	normal

The same. Minus 0.5 per error, down to 0.5 if you have something substantial correct (I'm not sure whether there are 8 things any more).

Extra: you can do extra fancy stuff to get the data in the same order as the original and to get rid of the missings, for example:

```
proc sort;
  by eating;
data negevallong2;
  set negevallong;
  if fne ~= .;

proc print;
```

```
Obs    fne    eating

  1     21    bulimic
  2     13    bulimic
  3     10    bulimic
  4     20    bulimic
  5     25    bulimic
  6     19    bulimic
  7     16    bulimic
  8     21    bulimic
  9     24    bulimic
 10     13    bulimic
 11     14    bulimic
 12     13    normal
 13      6    normal
 14     16    normal
 15     13    normal
 16      8    normal
 17     19    normal
 18     23    normal
 19     18    normal
 20     11    normal
 21     19    normal
 22      7    normal
 23     10    normal
 24     15    normal
 25     20    normal
```

If you can do *that*, I'm very impressed! There are several ways to say "not equals" in SAS: `ne`, `~=`, `^=`; any one of those would work. The `ne` comes from Fortran (which I learned, back in the day; you couldn't rely on your input device having symbols, so Fortran, and thus SAS, has EQ, NE, GT, GE, LT, LE whose meanings you can probably guess.)

Question 3 (10 marks)

Some people think that a mild amount of stress actually improves performance, but greater amounts of stress can be disruptive. In a study, 28 subjects were each given a task in which they had to keep a pointer on a moving disk. A clock records the time (in seconds) that the pointer is in contact with the disk. A larger value is better. Each subject is randomly assigned to one of four treatments, that relate to who was watching while they were trying to keep the pointer on the disk:

- `none`: no audience
- `experimenter`: experimenter as audience
- `peers`: peers (fellow students) as audience
- `faculty`: senior faculty members as audience

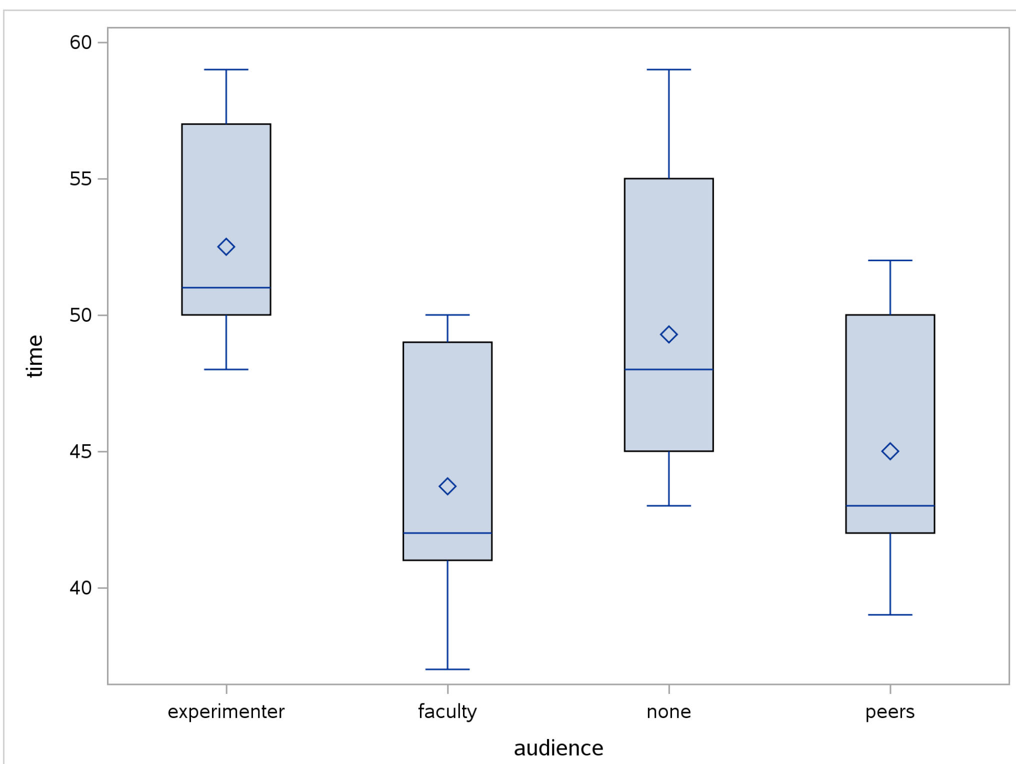
The data are shown in Figure 7. (This is the output from `proc print`.) The data set is called `stress`, which you may assume is the most recently created one.

(a) (3 marks) A boxplot is shown in Figure 8. Give the SAS code that was used to make the boxplot. (It is acceptable if your boxes will be in a different order from mine.)

My answer:

This is what you would probably guess:

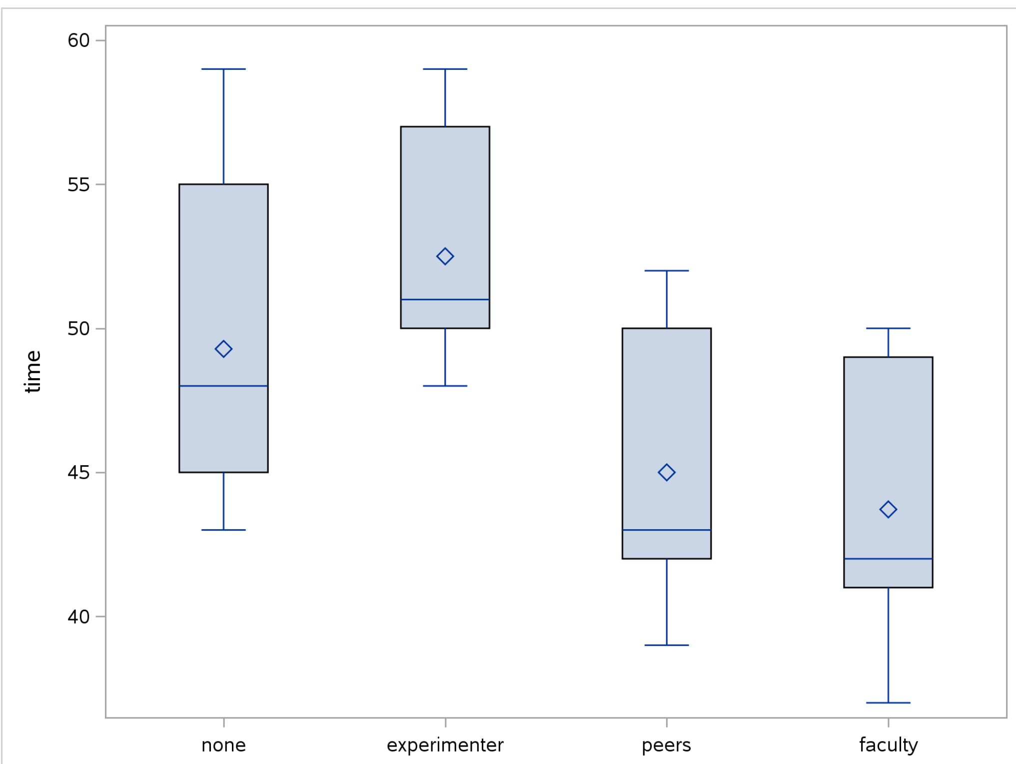
```
proc sgplot;
  vbox time / category=audience;
```



This is actually different from my plot in that the audience categories came out in *alphabetical* order, but since I said that the order of the boxes didn't matter, this is full marks.

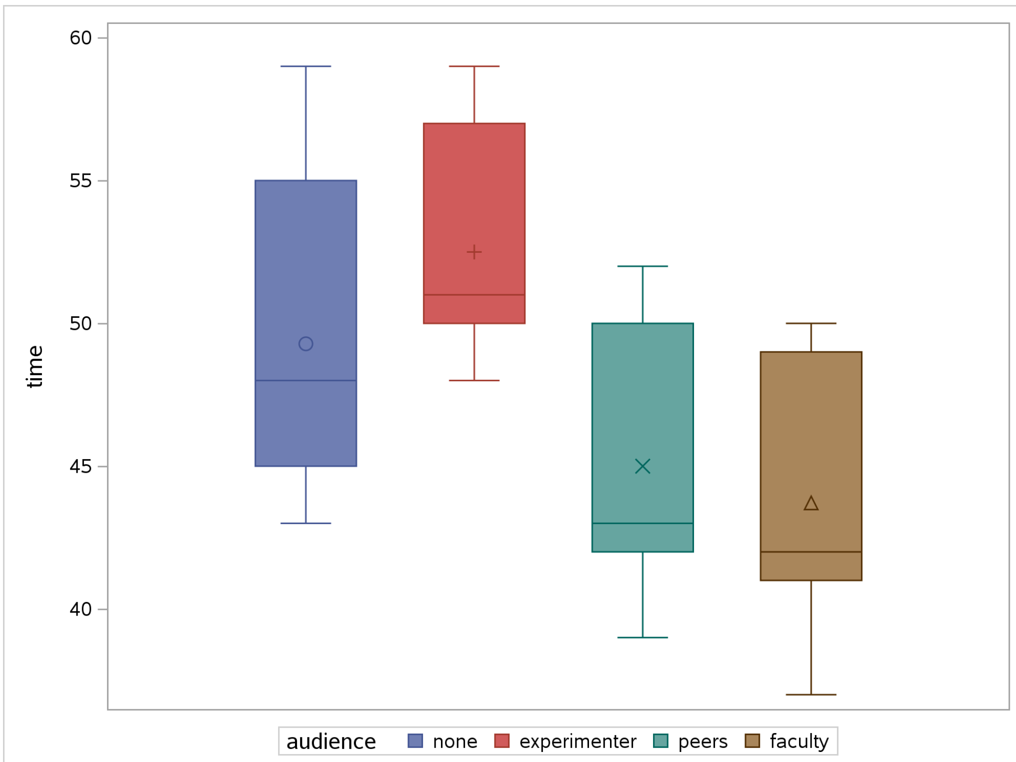
Extra: what I *actually* did was:

```
proc sgplot;
  vbox time / category=audience;
  xaxis display=(nolabel) discreteorder=data;
```



This seems to be the easiest way to do exactly this. (I had to look it up.) A perhaps more straightforward way colours the boxes:

```
proc sgplot;
  vbox time / group=audience grouporder=data;
```



Using `group` instead of `category` produces the colours, and also permits the use of `grouporder` to display the categories in the order that they appear in the data. This is the same idea as for a grouped boxplot, but with no categorical variable playing the role of the `category` (remember that in a grouped boxplot you have both a `category` and a `group`).

(b) (2 marks) Based on what you see in Figure 8, what would be an appropriate test to compare the times for the four groups, testing the null hypothesis that they all have the same mean or median? Explain briefly.

My answer: Another one of those where you can justify just about anything. If you think all four groups are more or less normal with more or less the same spreads, you can go with regular ANOVA. I think this is defensible since there are no outliers and not much skewness. If you think the groups are normal enough but the spreads are not close enough to equal, then the Welch ANOVA is the way to go. If you are not happy with the normality, for example you think the `faculty` group is too skewed left, go with Mood's median test. (An additional indication for this is that all four of the means are bigger than their corresponding medians, so there could be some right-skewness that is not showing up in the boxplots.)

The two points here are for the reasoning. You can reasonably propose any of the three possible tests (and by "test" I mean something that produces a P-value); as long as the reason you state is at least reasonably supported by the data and supports the test that you propose, I am good.

You might have wanted to see normal quantile plots by group, which `proc anova` would not produce. It seems to be hard to get these; you have to work out the "expected" normal values yourself and then use `scatter` in `proc sgplot` to plot them. You can do many things in SAS, but some of them seem to be very difficult.

(c) (3 marks) Give SAS code to run your preferred test from the previous part. You do not need to give code for any followup tests you might run if your preferred test gives a significant result.

My answer: I just want code for the equivalent of the ANOVA F test, depending on what you thought the right test was. (There are thus three possible right answers, depending on what you said in the previous part.)

If you thought regular ANOVA was good, then:

```
proc anova;
  class audience;
  model time=audience;
```

```
                          The ANOVA Procedure

                        Class Level Information

          Class         Levels    Values

          audience           4    experimenter faculty none peers
                 Number of Observations Read          28
                 Number of Observations Used          28
                          The ANOVA Procedure

                       Dependent Variable: time

                                      Sum of
Source                     DF        Squares    Mean Square   F Value   Pr > F

Model                       3    340.9553571    113.6517857      4.77   0.0096

Error                      24    572.3571429     23.8482143

Corrected Total            27    913.3125000
             R-Square     Coeff Var      Root MSE     time Mean

             0.373317      10.25399      4.883463      47.62500
```

Source	DF	Anova SS	Mean Square	F Value	Pr > F
audience	3	340.9553571	113.6517857	4.77	0.0096

If you thought Welch, then you have a little more work to do:

```
proc anova;
  class audience;
  model time=audience;
  means audience / hovtest=levene welch;
```

```
                          The ANOVA Procedure

                        Class Level Information

           Class         Levels    Values

           audience           4    experimenter faculty none peers
                   Number of Observations Read          28
                   Number of Observations Used          28
                          The ANOVA Procedure

                       Dependent Variable: time

                                   Sum of
Source                     DF        Squares    Mean Square   F Value   Pr > F

Model                       3    340.9553571    113.6517857      4.77   0.0096

Error                      24    572.3571429     23.8482143

Corrected Total            27    913.3125000
              R-Square     Coeff Var      Root MSE     time Mean

              0.373317      10.25399      4.883463      47.62500
Source                     DF       Anova SS    Mean Square   F Value   Pr > F

audience                    3    340.9553571    113.6517857      4.77   0.0096
                          The ANOVA Procedure

               Levene's Test for Homogeneity of time Variance
                ANOVA of Squared Deviations from Group Means

                            Sum of        Mean
       Source          DF     Squares      Square    F Value    Pr > F

       audience         3       894.3       298.1       0.63    0.6014
       Error           24     11316.0       471.5
                        Welch's ANOVA for time

                  Source            DF    F Value    Pr > F

                  audience      3.0000       5.37    0.0123
                  Error        13.2307
```

```
Level of                  -------------time------------
audience        N              Mean          Std Dev

experimenter    7        52.5000000       3.98956973
faculty         7        43.7142857       4.75093976
none            7        49.2857143       5.85133277
peers           7        45.0000000       4.76095229
```

If you thought Mood's median test, then this (which is exactly as it would be if there were two groups):

```
proc npar1way median;
  class audience;
  var time;
```

```
                          The NPAR1WAY Procedure

      Median Scores (Number of Points Above Median) for Variable time
                     Classified by Variable audience

                              Sum of      Expected       Std Dev          Mean
audience            N         Scores      Under H0      Under H0         Score

none                7           3.50          3.50      1.124228      0.500000
experimenter        7           6.50          3.50      1.124228      0.928571
peers               7           2.00          3.50      1.124228      0.285714
faculty             7           2.00          3.50      1.124228      0.285714

                   Average scores were used for ties.
                       Median One-Way Analysis

                       Chi-Square        8.0110
                       DF                     3
                       Pr > Chi-Square   0.0458
```

The key thing is to be consistent with yourself: that is, to give code that will run the test you said was best in the previous part. Being inconsistent with yourself will cost you marks here. Being consistent with even some crazy suggestion for a test can get you full marks here, if the code you need to write is of equivalent complexity to these.

There is a small exam-strategy thing here, which is that you can make life easier for yourself by trying to justify a test that is easier to write code for (ie. not Welch).

Extra: the three P-values are all fairly similar, and, as usual, the regular ANOVA and Welch ANOVA P-values are very close (just either side of 0.01). So it doesn't really matter which test you do. The fact that Mood's median test gives a slightly larger P-value is an indication that one of the ANOVAs is going to be more powerful; Levene's test, if you want to look at it, says that there is no evidence of a difference in spreads among the four treatment groups. So I think regular ANOVA is perfectly good.

I don't need any code for Tukey or Games-Howell or pairwise median tests (if you can even figure out how to do that). If you have some, it will be ignored. So save yourself some time and don't even try to write that down.

This does not work:

```
proc univariate all;
  class audience;
```

```
                     The UNIVARIATE Procedure
                        Variable:  time
                     audience = experimenter

                            Moments

N                           7    Sum Weights                 7
Mean                     52.5    Sum Observations        367.5
Std Deviation      3.98956973    Variance           15.9166667
Skewness           0.90943949    Kurtosis           -0.4902333
Uncorrected SS       19389.25    Corrected SS             95.5
Coeff Variation    7.59918045    Std Error Mean     1.50791562
                   Basic Statistical Measures

          Location                    Variability

     Mean     52.50000     Std Deviation            3.98957
     Median   51.00000     Variance                15.91667
     Mode       .          Range                   11.00000
                           Interquartile Range      7.00000
                              Modes

                          Mode    Count

                            .         .
```

```
                   Basic Confidence Limits Assuming Normality

              Parameter          Estimate     95% Confidence Limits

              Mean               52.50000      48.81026    56.18974
              Std Deviation       3.98957       2.57085     8.78530
              Variance           15.91667       6.60928    77.18143
                             Tests for Location: Mu0=0

                 Test           -Statistic-    -----p Value------

                 Student's t    t  34.81627    Pr > |t|    <.0001
                 Sign           M       3.5    Pr >= |M|   0.0156
                 Signed Rank    S        14    Pr >= |S|   0.0156
                             Location Counts: Mu0=0.00

                             Count                Value

                             Num Obs > Mu0            7
                             Num Obs ^= Mu0           7
                             Num Obs < Mu0            0
                                Tests for Normality

            Test                  --Statistic---    -----p Value------

            Shapiro-Wilk          W     0.883781    Pr < W      0.2438
            Kolmogorov-Smirnov    D     0.264153    Pr > D      0.1431
            Cramer-von Mises      W-Sq  0.086489    Pr > W-Sq   0.1436
            Anderson-Darling      A-Sq  0.462574    Pr > A-Sq   0.1804
                                      Trimmed Means

   Percent    Number                Std Error
   Trimmed   Trimmed       Trimmed     Trimmed     95% Confidence
   in Tail   in Tail          Mean        Mean         Limits                DF

     28.57         2      51.16667     0.755929   47.91417  54.41917          2

                                     Trimmed Means

                         Percent
                         Trimmed      t for H0:
                         in Tail       Mu0=0.00   Pr > |t|

                           28.57       67.68714     0.0002
```

```
                              Winsorized Means

    Percent      Number                 Std Error
 Winsorized  Winsorized  Winsorized  Winsorized    95% Confidence
    in Tail     in Tail        Mean        Mean        Limits              DF

      28.57           2    51.21429    0.857143  47.52630  54.90227         2

                              Winsorized Means

                       Percent
                    Winsorized     t for H0:
                       in Tail      Mu0=0.00    Pr > |t|

                         28.57      59.75000      0.0003
                           Robust Measures of Scale

                                                     Estimate
                  Measure                     Value   of Sigma

                  Interquartile Range      7.000000   5.189106
                  Gini's Mean Difference   4.619048   4.093524
                  MAD                      1.000000   1.482600
                  Sn                       2.385200   2.857470
                  Qn                       4.443800   3.808337
                          Quantiles (Definition 5)

                                             95% Confidence Limits
                Level         Quantile         Assuming Normality

                100% Max            59
                99%                 59     57.74544     73.88697
                95%                 59     55.70919     68.21924
                90%                 59     54.52396     65.28857
                75% Q3              57     52.22007     60.65352
                50% Median          51     48.81026     56.18974
                25% Q1              50     44.34648     52.77993
                10%                 48     39.71143     50.47604
                5%                  48     36.78076     49.29081
                1%                  48     31.11303     47.25456
                0% Min              48

                          Quantiles (Definition 5)

                   95% Confidence Limits    -------Order Statistics-------
    Level             Distribution Free      LCL Rank  UCL Rank    Coverage

    100% Max
    99%                .               .          .         .           .
    95%                .               .          .         .           .
    90%             52.0            59.0          5         7       49.60
    75% Q3          50.5            59.0          3         7       85.36
    50% Median      48.0            59.0          1         7       98.44
```

```
                    Extreme Observations

            ----Lowest----        ----Highest---

            Value      Obs        Value      Obs

             48.0       11         50.5       14
             50.0        8         51.0       10
             50.5       14         52.0        9
             51.0       10         57.0       12
             52.0        9         59.0       13
                       Extreme Values

              ----Lowest----   ----Highest---

               Order   Value    Order   Value

                   1    48.0        3    50.5
                   2    50.0        4    51.0
                   3    50.5        5    52.0
                   4    51.0        6    57.0
                   5    52.0        7    59.0
                      Frequency Counts

                                  Percents
                      Value Count  Cell   Cum

                         48     1  14.3  14.3
                         50     1  14.3  28.6
                         51     1  14.3  42.9
                         51     1  14.3  57.1
                         52     1  14.3  71.4
                         57     1  14.3  85.7
                         59     1  14.3 100.0
                  The UNIVARIATE Procedure
                      Variable:  time
                     audience = faculty

                          Moments

N                           7    Sum Weights                 7
Mean               43.7142857    Sum Observations          306
Std Deviation      4.75093976    Variance           22.5714286
Skewness           0.13108666    Kurtosis            -1.261144
Uncorrected SS          13512    Corrected SS       135.428571
Coeff Variation    10.8681628    Std Error Mean     1.79568644
```

```
                  Basic Statistical Measures

         Location                    Variability

     Mean     43.71429     Std Deviation            4.75094
     Median   42.00000     Variance                22.57143
     Mode     41.00000     Range                   13.00000
                           Interquartile Range      8.00000
                             Modes

                         Mode    Count

                           41        2
          Basic Confidence Limits Assuming Normality

     Parameter          Estimate     95% Confidence Limits

     Mean               43.71429      39.32040    48.10817
     Std Deviation       4.75094       3.06147    10.46188
     Variance           22.57143       9.37262   109.45101
                  Tests for Location: Mu0=0

        Test           -Statistic-    -----p Value------

        Student's t    t  24.34405    Pr > |t|    <.0001
        Sign           M       3.5    Pr >= |M|   0.0156
        Signed Rank    S        14    Pr >= |S|   0.0156
                  Location Counts: Mu0=0.00

                  Count                 Value

                  Num Obs > Mu0             7
                  Num Obs ^= Mu0            7
                  Num Obs < Mu0             0
                     Tests for Normality

   Test                  --Statistic---    -----p Value------

   Shapiro-Wilk          W     0.931244    Pr < W      0.5615
   Kolmogorov-Smirnov    D     0.212316    Pr > D     >0.1500
   Cramer-von Mises      W-Sq  0.050854    Pr > W-Sq  >0.2500
   Anderson-Darling      A-Sq  0.298502    Pr > A-Sq  >0.2500
```

```
                           Trimmed Means

 Percent    Number                 Std Error
 Trimmed   Trimmed       Trimmed     Trimmed     95% Confidence
 in Tail   in Tail          Mean        Mean         Limits              DF

   28.57         2      43.00000    2.563480   31.97024  54.02976         2

                          Trimmed Means

                     Percent
                     Trimmed      t for H0:
                     in Tail       Mu0=0.00   Pr > |t|

                       28.57       16.77407     0.0035
                         Winsorized Means

    Percent      Number               Std Error
 Winsorized  Winsorized  Winsorized  Winsorized    95% Confidence
    in Tail     in Tail        Mean        Mean        Limits            DF

      28.57           2    43.28571    2.906713  30.77914  55.79229       2

                         Winsorized Means

                       Percent
                    Winsorized     t for H0:
                       in Tail      Mu0=0.00    Pr > |t|

                         28.57      14.89164      0.0045
                     Robust Measures of Scale

                                                Estimate
                Measure                  Value   of Sigma

                Interquartile Range   8.000000   5.930407
                Gini's Mean Difference 5.714286  5.064154
                MAD                   4.000000   5.930400
                Sn                    4.770400   5.714939
                Qn                    8.887600   7.616673
```

```
                    Quantiles (Definition 5)

                                     95% Confidence Limits
           Level         Quantile      Assuming Normality

           100% Max            50
           99%                 50     49.96077     69.18275
           95%                 50     47.53591     62.43339
           90%                 50     46.12449     58.94343
           75% Q3              49     43.38093     53.42383
           50% Median          42     39.32040     48.10817
           25% Q1              41     34.00474     44.04764
           10%                 37     28.48514     41.30408
           5%                  37     24.99519     39.89266
           1%                  37     18.24583     37.46780
           0% Min              37

                    Quantiles (Definition 5)

                 95% Confidence Limits    -------Order Statistics-------
Level              Distribution Free      LCL Rank  UCL Rank    Coverage

100% Max
99%                .                .         .         .           .
95%                .                .         .         .           .
90%               46               50         5         7       49.60
75% Q3            41               50         3         7       85.36
50% Median        37               50         1         7       98.44
25% Q1            37               46         1         5       85.36
10%               37               41         1         3       49.60
5%                 .                .         .         .           .
1%                 .                .         .         .           .
0% Min
                        Extreme Observations

                   ----Lowest----        ----Highest---

                   Value       Obs        Value       Obs

                      37        27           41        26
                      41        26           42        23
                      41        22           46        24
                      42        23           49        25
                      46        24           50        28
```

```
                         Extreme Values

         ---------Lowest--------   --------Highest--------

          Order   Value     Freq    Order   Value     Freq

              1      37        1        2      41        2
              2      41        2        3      42        1
              3      42        1        4      46        1
              4      46        1        5      49        1
              5      49        1        6      50        1
                        Frequency Counts

                                     Percents
                        Value Count  Cell   Cum

                           37     1  14.3  14.3
                           41     2  28.6  42.9
                           42     1  14.3  57.1
                           46     1  14.3  71.4
                           49     1  14.3  85.7
                           50     1  14.3 100.0
                    The UNIVARIATE Procedure
                        Variable:  time
                        audience = none

                            Moments

N                          7    Sum Weights                  7
Mean              49.2857143    Sum Observations           345
Std Deviation     5.85133277    Variance            34.2380952
Skewness          0.81034283    Kurtosis            -0.5391761
Uncorrected SS         17209    Corrected SS        205.428571
Coeff Variation   11.8722694    Std Error Mean      2.21159591
                   Basic Statistical Measures

         Location                    Variability

     Mean     49.28571     Std Deviation            5.85133
     Median   48.00000     Variance                34.23810
     Mode     45.00000     Range                   16.00000
                           Interquartile Range     10.00000
                             Modes

                          Mode    Count

                             45        2
```

```
                Basic Confidence Limits Assuming Normality

          Parameter          Estimate     95% Confidence Limits

          Mean               49.28571      43.87413    54.69729
          Std Deviation       5.85133       3.77056    12.88502
          Variance           34.23810      14.21712   166.02378
                         Tests for Location: Mu0=0

             Test           -Statistic-    -----p Value------

             Student's t    t  22.28514    Pr > |t|    <.0001
             Sign           M       3.5    Pr >= |M|   0.0156
             Signed Rank    S        14    Pr >= |S|   0.0156
                         Location Counts: Mu0=0.00

                         Count                Value

                         Num Obs > Mu0            7
                         Num Obs ^= Mu0           7
                         Num Obs < Mu0            0
                            Tests for Normality

        Test                  --Statistic---    -----p Value------

        Shapiro-Wilk          W     0.914391    Pr < W      0.4271
        Kolmogorov-Smirnov    D      0.19662    Pr > D     >0.1500
        Cramer-von Mises      W-Sq  0.052248    Pr > W-Sq  >0.2500
        Anderson-Darling      A-Sq  0.324316    Pr > A-Sq  >0.2500
                                Trimmed Means

 Percent    Number                 Std Error
 Trimmed   Trimmed       Trimmed     Trimmed     95% Confidence
 in Tail   in Tail          Mean        Mean         Limits              DF

   28.57         2      47.66667    2.507133   36.87935  58.45399          2

                               Trimmed Means

                     Percent
                     Trimmed      t for H0:
                     in Tail       Mu0=0.00   Pr > |t|

                       28.57       19.01242     0.0028
```

Winsorized Means

Percent Winsorized in Tail	Number Winsorized in Tail	Winsorized Mean	Std Error Winsorized Mean	95% Confidence Limits		DF
28.57	2	47.57143	2.842821	35.33976	59.80310	2

Winsorized Means

Percent Winsorized in Tail	t for H0: Mu0=0.00	Pr > |t|
28.57	16.73388	0.0036

Robust Measures of Scale

Measure	Value	Estimate of Sigma
Interquartile Range	10.00000	7.413008
Gini's Mean Difference	6.95238	6.161387
MAD	3.00000	4.447800
Sn	5.96300	7.143674
Qn	6.66570	5.712505

Quantiles (Definition 5)

Level	Quantile	95% Confidence Limits Assuming Normality	
100% Max	59		
99%	59	56.97898	80.65307
95%	59	53.99249	72.34045
90%	59	52.25416	68.04217
75% Q3	55	48.87515	61.24414
50% Median	48	43.87413	54.69729
25% Q1	45	37.32729	49.69627
10%	43	30.52925	46.31727
5%	43	26.23097	44.57894
1%	43	17.91836	41.59245
0% Min	43		

Quantiles (Definition 5)

Level	95% Confidence Limits Distribution Free		-------Order Statistics------- LCL Rank	UCL Rank	Coverage
100% Max					
99%	.	.	.	.	.
95%	.	.	.	.	.
90%	50	59	5	7	49.60
75% Q3	45	59	3	7	85.36
50% Median	43	59	1	7	98.44

```
                    Extreme Observations

              ----Lowest----        ----Highest---

              Value      Obs        Value      Obs

                 43        3           45        7
                 45        7           48        4
                 45        1           50        2
                 48        4           55        5
                 50        2           59        6
                       Extreme Values

       ---------Lowest--------   --------Highest--------

        Order   Value     Freq    Order   Value     Freq

            1      43        1        2      45        2
            2      45        2        3      48        1
            3      48        1        4      50        1
            4      50        1        5      55        1
            5      55        1        6      59        1
                      Frequency Counts

                                 Percents
                      Value Count  Cell   Cum

                         43     1  14.3  14.3
                         45     2  28.6  42.9
                         48     1  14.3  57.1
                         50     1  14.3  71.4
                         55     1  14.3  85.7
                         59     1  14.3 100.0
                    The UNIVARIATE Procedure
                      Variable:  time
                      audience = peers

                          Moments

N                          7    Sum Weights                 7
Mean                      45    Sum Observations          315
Std Deviation     4.76095229    Variance           22.6666667
Skewness          0.42811504    Kurtosis           -1.2980969
Uncorrected SS         14311    Corrected SS              136
Coeff Variation    10.579894    Std Error Mean     1.79947082
```

```
                    Basic Statistical Measures

          Location                    Variability

      Mean     45.00000     Std Deviation            4.76095
      Median   43.00000     Variance                22.66667
      Mode     42.00000     Range                   13.00000
                            Interquartile Range      8.00000
                              Modes

                          Mode    Count

                            42       2
           Basic Confidence Limits Assuming Normality

      Parameter          Estimate     95% Confidence Limits

      Mean               45.00000      40.59685    49.40315
      Std Deviation       4.76095       3.06793    10.48393
      Variance           22.66667       9.41217   109.91282
                    Tests for Location: Mu0=0

         Test           -Statistic-    -----p Value------

         Student's t    t  25.00735    Pr > |t|    <.0001
         Sign           M       3.5    Pr >= |M|   0.0156
         Signed Rank    S        14    Pr >= |S|   0.0156
                    Location Counts: Mu0=0.00

                    Count                 Value

                    Num Obs > Mu0            7
                    Num Obs ^= Mu0           7
                    Num Obs < Mu0            0
                       Tests for Normality

    Test                  --Statistic---    -----p Value------

    Shapiro-Wilk          W     0.927332    Pr < W      0.5284
    Kolmogorov-Smirnov    D     0.234217    Pr > D     >0.1500
    Cramer-von Mises      W-Sq  0.055523    Pr > W-Sq  >0.2500
    Anderson-Darling      A-Sq  0.314873    Pr > A-Sq  >0.2500
```

```
                              Trimmed Means

 Percent    Number                  Std Error
 Trimmed   Trimmed       Trimmed      Trimmed     95% Confidence
 in Tail   in Tail          Mean         Mean         Limits                DF

   28.57         2      44.00000     2.563480   32.97024  55.02976           2

                             Trimmed Means

                     Percent
                     Trimmed      t for H0:
                     in Tail       Mu0=0.00   Pr > |t|

                       28.57       17.16417     0.0034
                            Winsorized Means

    Percent      Number                Std Error
 Winsorized  Winsorized  Winsorized  Winsorized    95% Confidence
    in Tail     in Tail        Mean        Mean        Limits                DF

      28.57           2    44.28571    2.906713  31.77914  56.79229          2

                            Winsorized Means

                       Percent
                    Winsorized     t for H0:
                       in Tail      Mu0=0.00    Pr > |t|

                         28.57      15.23567      0.0043
                        Robust Measures of Scale

                                                 Estimate
                Measure                   Value   of Sigma

                Interquartile Range    8.000000   5.930407
                Gini's Mean Difference 5.714286   5.064154
                MAD                    4.000000   5.930400
                Sn                     4.770400   5.714939
                Qn                     6.665700   5.712505
```

Quantiles (Definition 5)

Level	Quantile	95% Confidence Limits Assuming Normality	
100% Max	52		
99%	52	51.25965	70.52213
95%	52	48.82968	63.75855
90%	52	47.41529	60.26124
75% Q3	50	44.66595	54.73000
50% Median	43	40.59685	49.40315
25% Q1	42	35.27000	45.33405
10%	39	29.73876	42.58471
5%	39	26.24145	41.17032
1%	39	19.47787	38.74035
0% Min	39		

Quantiles (Definition 5)

Level	95% Confidence Limits Distribution Free		-------Order Statistics------- LCL Rank	UCL Rank	Coverage
100% Max					
99%	.	.	.	.	.
95%	.	.	.	.	.
90%	47	52	5	7	49.60
75% Q3	42	52	3	7	85.36
50% Median	39	52	1	7	98.44
25% Q1	39	47	1	5	85.36
10%	39	42	1	3	49.60
5%	.	.	.	.	.
1%	.	.	.	.	.
0% Min					

Extreme Observations

----Lowest---- Value	Obs	----Highest--- Value	Obs
39	20	42	18
42	18	43	15
42	16	47	17
43	15	50	19
47	17	52	21

```
                         Extreme Values

      ---------Lowest--------   --------Highest--------

      Order   Value     Freq    Order   Value     Freq

          1      39        1        2      42        2
          2      42        2        3      43        1
          3      43        1        4      47        1
          4      47        1        5      50        1
          5      50        1        6      52        1
                        Frequency Counts

                                     Percents
                        Value Count  Cell   Cum

                           39     1  14.3  14.3
                           42     2  28.6  42.9
                           43     1  14.3  57.1
                           47     1  14.3  71.4
                           50     1  14.3  85.7
                           52     1  14.3 100.0
```

(d) (2 marks) At the beginning of the question there was a suggestion that moderate stress levels might be associated with best performance. Is that supported by the data, from the output you have? Explain briefly. (If you cannot tell, describe what you would need to see and why.)

My answer: The best way to approach this is to look back at the boxplot, Figure 8. No audience on the left is the lowest stress, and "faculty" audience on the right is the highest stress. The performance seems to go up and down again, indicating that highest and lowest stress are worst and moderate stress (just the experimenter watching) is best.

You might reasonably say that you need the ANOVA results to see if there are any significant differences at all. I would support that, *if* coupled with a discussion like the above. The point is that a significant F-test doesn't tell you anything about the research hypothesis by itself; for all you know, unless you look at the boxplot, the performance could even be best at high stress levels!

Extra: I didn't give you the ANOVA results, or Tukey, so it's hard to be sure, but looking at the results in these solutions suggests that there are some significant differences *somewhere*, and if they are anywhere, the results appear to be best for moderate stress and worst for high or low stress.

Since I said that regular ANOVA was OK, we can do Tukey, which comes out this way:

```
proc anova;
  class audience;
  model time=audience;
```

```
means audience / tukey;
```

Remember that the categorical variable is called `audience` here, not `group` as in the lecture:

```
                              The ANOVA Procedure

                            Class Level Information

            Class         Levels    Values

            audience           4    experimenter faculty none peers
                    Number of Observations Read          28
                    Number of Observations Used          28
                              The ANOVA Procedure

                          Dependent Variable: time

                                        Sum of
 Source                      DF        Squares    Mean Square   F Value   Pr > F

 Model                        3    340.9553571    113.6517857      4.77   0.0096

 Error                       24    572.3571429     23.8482143

 Corrected Total             27    913.3125000
               R-Square     Coeff Var      Root MSE     time Mean

               0.373317      10.25399      4.883463      47.62500
 Source                      DF       Anova SS    Mean Square   F Value   Pr > F

 audience                     3    340.9553571    113.6517857      4.77   0.0096
                              The ANOVA Procedure

                 Tukey's Studentized Range (HSD) Test for time

NOTE: This test controls the Type I experimentwise error rate, but it generally
                  has a higher Type II error rate than REGWQ.


                  Alpha                                   0.05
                  Error Degrees of Freedom                  24
                  Error Mean Square                   23.84821
                  Critical Value of Studentized Range  3.90126
                  Minimum Significant Difference        7.2009
```

```
          Means with the same letter are not significantly different.


          Tukey Grouping          Mean      N    audience

                       A        52.500      7    experimenter
                       A
                  B    A        49.286      7    none
                  B
                  B             45.000      7    peers
                  B
                  B             43.714      7    faculty
```

This is one of those ambiguous ones: audience `none` is in the middle ground, not significantly worse than `experimenter`, and yet not significantly better than the others. So the results are pointing in the direction of the research hypothesis, but they don't quite get there.

Question 4 (23 marks)

A university professor in California cycles to work. He has three routes that he uses, labelled by the name of the street that most of the route is on. He randomly chooses a route each day, and records the time in seconds that it takes to get from his house to the bike parking at his university. The data, as the professor recorded it, is shown in Figure 9. Note that he did not cycle each route the same number of times. The data has been read into a data frame called `biking`, as shown in the Figure. Give R, that is, Tidyverse code to accomplish the tasks below.

(a) (4 marks) Reorganize the data so that it has one row for each of the 52 completed rides, and columns called `street` and `time` that contain respectively the name of the street on which he cycled and the time in seconds that it took to complete the ride.

My answer: This, for 2019 students, is *exactly* the same as the tidying on Assignment 6, right down to the removal of missing values. I think the best solution, using `pivot_longer`, is:

```
biking %>%
  pivot_longer(everything(), names_to="street", values_to="time",
               values_drop_na = T)
## # A tibble: 52 x 2
##    street  time
##    <chr>  <dbl>
##  1 Oxnard   732
##  2 Rose     869
##  3 Rice     694
##  4 Oxnard   842
##  5 Rose     648
##  6 Rice     629
##  7 Oxnard   736
##  8 Rose    1045
##  9 Rice     863
## 10 Oxnard   732
## # ... with 42 more rows
```

This does everything in one step. Use any way you like to select all the columns, such as `Oxnard:Rice`.

If you couldn't recall that, do the dropping of NA in a second step:

```
biking %>% pivot_longer(everything(), names_to="street",
                        values_to="time") %>%
  drop_na()
## # A tibble: 52 x 2
##    street  time
##    <chr>  <dbl>
##  1 Oxnard   732
##  2 Rose     869
##  3 Rice     694
##  4 Oxnard   842
##  5 Rose     648
##  6 Rice     629
```

```
## 7 Oxnard 736
## 8 Rose 1045
## 9 Rice 863
## 10 Oxnard 732
## # ... with 42 more rows
```

It is also good (in 2019 at least) to use `gather`. This also permits a way of removing the NA in one shot:

```
biking %>% gather(street, time, everything(), na.rm=T)
```

```
## # A tibble: 52 x 2
##    street  time
##    <chr>  <dbl>
##  1 Oxnard   732
##  2 Oxnard   842
##  3 Oxnard   736
##  4 Oxnard   732
##  5 Oxnard   736
##  6 Oxnard   833
##  7 Oxnard   655
##  8 Oxnard   688
##  9 Oxnard   727
## 10 Oxnard   721
## # ... with 42 more rows
```

or, again, do it in a second step:

```
biking %>% gather(street, time, everything()) %>%
  drop_na()
```

```
## # A tibble: 52 x 2
##    street  time
##    <chr>  <dbl>
##  1 Oxnard   732
##  2 Oxnard   842
##  3 Oxnard   736
##  4 Oxnard   732
##  5 Oxnard   736
##  6 Oxnard   833
##  7 Oxnard   655
##  8 Oxnard   688
##  9 Oxnard   727
## 10 Oxnard   721
## # ... with 42 more rows
```

`na.omit()` is a valid alternative to `drop_na` in either case. If you use `drop_na`, or `na.omit`, you need to use it *at the end*, because otherwise you'll drop rows that have missing values *and some good values* in them, in particular the bottom 4 rows.

Grading: 4 if you do everything, including removing the missings. 3 if you successfully reorganize the columns but don't remove the missings. Minus 1 in addition per error in your code, down to a minimum of 1 if you have something substantial correct.

(b) (2 marks) Using the data in Figure 9, display the columns called `Oxnard` and `Rice` (and *not* `Rose`).

My answer: This is `select`:

```
biking %>% select(Oxnard, Rice)
## # A tibble: 19 x 2
##    Oxnard  Rice
##     <dbl> <dbl>
##  1    732   694
##  2    842   629
##  3    736   863
##  4    732   748
##  5    736   767
##  6    833   574
##  7    655   628
##  8    688   637
##  9    727   620
## 10    721   752
## 11    695   608
## 12    707   983
## 13    843   765
## 14    852   666
## 15    789   727
## 16     NA   729
## 17     NA   605
## 18     NA   717
## 19     NA   679
```

You can also do it this way:

```
biking %>% select(c(Oxnard, Rice))
## # A tibble: 19 x 2
##    Oxnard  Rice
##     <dbl> <dbl>
##  1    732   694
##  2    842   629
##  3    736   863
##  4    732   748
##  5    736   767
##  6    833   574
##  7    655   628
##  8    688   637
##  9    727   620
## 10    721   752
## 11    695   608
## 12    707   983
```

```
## 13    843   765
## 14    852   666
## 15    789   727
## 16     NA   729
## 17     NA   605
## 18     NA   717
## 19     NA   679
```

or this way:

```
biking %>% select(-Rose)
## # A tibble: 19 x 2
##    Oxnard  Rice
##     <dbl> <dbl>
##  1    732   694
##  2    842   629
##  3    736   863
##  4    732   748
##  5    736   767
##  6    833   574
##  7    655   628
##  8    688   637
##  9    727   620
## 10    721   752
## 11    695   608
## 12    707   983
## 13    843   765
## 14    852   666
## 15    789   727
## 16     NA   729
## 17     NA   605
## 18     NA   717
## 19     NA   679
```

1 mark if you have the idea but screw something up. For example, `Oxnard:Rice` would display *all three columns* ("Oxnard through Rice").

I inserted the word Tidyverse in the question to dissuade you from doing something like

```
biking[,c(1,3)]
## # A tibble: 19 x 2
##    Oxnard  Rice
##     <dbl> <dbl>
##  1    732   694
##  2    842   629
##  3    736   863
##  4    732   748
##  5    736   767
##  6    833   574
##  7    655   628
```

```
## 8 688 637
## 9 727 620
## 10 721 752
## 11 695 608
## 12 707 983
## 13 843 765
## 14 852 666
## 15 789 727
## 16 NA 729
## 17 NA 605
## 18 NA 717
## 19 NA 679
```

or

```
biking[,-2]
```

```
## # A tibble: 19 x 2
##    Oxnard  Rice
##     <dbl> <dbl>
##  1    732   694
##  2    842   629
##  3    736   863
##  4    732   748
##  5    736   767
##  6    833   574
##  7    655   628
##  8    688   637
##  9    727   620
## 10    721   752
## 11    695   608
## 12    707   983
## 13    843   765
## 14    852   666
## 15    789   727
## 16     NA   729
## 17     NA   605
## 18     NA   717
## 19     NA   679
```

which is one mark if you get it right, because it works, but it's not what I asked for. (I haven't done this in this class, so it's not what I want to test.)

(c) (3 marks) Using the data in Figure 9, display the columns whose names begin with R, *without naming any columns or referring to them by number.*

My answer: This means to use a select-helper, thus:

```
biking %>% select(starts_with("R"))
```

```
## # A tibble: 19 x 2
##     Rose  Rice
##    <dbl> <dbl>
##  1   869   694
##  2   648   629
##  3  1045   863
##  4   674   748
##  5   821   767
##  6   708   574
##  7   840   628
##  8  1029   637
##  9   735   620
## 10   745   752
## 11   794   608
## 12   652   983
## 13   552   765
## 14   732   666
## 15   578   727
## 16   661   729
## 17   657   605
## 18   869   717
## 19    NA   679
```

or anything else less sane that will still work, such as:

```
biking %>% select(ends_with("e"))
```

```
## # A tibble: 19 x 2
##     Rose  Rice
##    <dbl> <dbl>
##  1   869   694
##  2   648   629
##  3  1045   863
##  4   674   748
##  5   821   767
##  6   708   574
##  7   840   628
##  8  1029   637
##  9   735   620
## 10   745   752
## 11   794   608
## 12   652   983
## 13   552   765
## 14   732   666
## 15   578   727
## 16   661   729
## 17   657   605
## 18   869   717
```

```
## 19    NA   679
```

Expect to get 1 mark if you do something that names a column, such as

```
biking %>% select(-Oxnard)

## # A tibble: 19 x 2
##     Rose  Rice
##    <dbl> <dbl>
##  1   869   694
##  2   648   629
##  3  1045   863
##  4   674   748
##  5   821   767
##  6   708   574
##  7   840   628
##  8  1029   637
##  9   735   620
## 10   745   752
## 11   794   608
## 12   652   983
## 13   552   765
## 14   732   666
## 15   578   727
## 16   661   729
## 17   657   605
## 18   869   717
## 19    NA   679
```

Or if you try to do something with base R, such as this (which also fails because it refers to columns by number):

```
biking[,2:3]

## # A tibble: 19 x 2
##     Rose  Rice
##    <dbl> <dbl>
##  1   869   694
##  2   648   629
##  3  1045   863
##  4   674   748
##  5   821   767
##  6   708   574
##  7   840   628
##  8  1029   637
##  9   735   620
## 10   745   752
## 11   794   608
## 12   652   983
## 13   552   765
## 14   732   666
```

```
## 15   578   727
## 16   661   729
## 17   657   605
## 18   869   717
## 19    NA   679
```

I am very unsure about giving you one for this (it really deserves 0.5), because it doesn't get at the *spirit* of the question at all, which was "give me all the columns whose names begin with R, no matter where in the data frame they are". Looking at the data frame, and then picking out the columns you want by number, is really cheating, in the same way that this is:

```
biking %>% select(2:3)
```

```
## # A tibble: 19 x 2
##      Rose  Rice
##     <dbl> <dbl>
##  1   869   694
##  2   648   629
##  3  1045   863
##  4   674   748
##  5   821   767
##  6   708   574
##  7   840   628
##  8  1029   637
##  9   735   620
## 10   745   752
## 11   794   608
## 12   652   983
## 13   552   765
## 14   732   666
## 15   578   727
## 16   661   729
## 17   657   605
## 18   869   717
## 19    NA   679
```

What if you have over a hundred columns, some of whose names begin with R? Are you going to go through the whole data frame, note down the numbers of the columns that begin with R, and then take the columns by that number? *Inefficient.* You can do it much better.

(d) (3 marks) Some of the biking data as you rearranged it in part (a) is shown in Figure 10, as a data frame called `biking_long`. Use this data frame for the rest of the question.

Display the number of times each route was ridden, together with the mean time for each route.

My answer: `group_by` and `summarize`, using `n()` to count:

```
biking_long %>% group_by(street) %>%
    summarize(n=n(), mean=mean(time))
## # A tibble: 3 x 3
##   street     n  mean
##   <chr>  <int> <dbl>
## 1 Oxnard    15  753.
## 2 Rice      19  705.
## 3 Rose      18  756.
```

"Along with" implies that you need to get both in one shot for full marks. This, therefore, is worth only two altogether:

```
biking_long %>% count(street)
## # A tibble: 3 x 2
##   street     n
##   <chr>  <int>
## 1 Oxnard    15
## 2 Rice      19
## 3 Rose      18
biking_long %>% group_by(street) %>% summarize(mean=mean(time))
## # A tibble: 3 x 2
##   street  mean
##   <chr>  <dbl>
## 1 Oxnard  753.
## 2 Rice    705.
## 3 Rose    756.
```

Extra: getting these results from the original data frame `biking` is much harder, involving `summarize_all` and some trickery:

```
biking %>% summarize_all(list(~n(), ~mean(., na.rm=T)))
## # A tibble: 1 x 6
##   Oxnard_n Rose_n Rice_n Oxnard_mean Rose_mean Rice_mean
##      <int>  <int>  <int>       <dbl>     <dbl>     <dbl>
## 1       19     19     19        753.      756.      705.
```

and that doesn't even do it right! It counts the number of rows in each column, not the number of non-missing ones.

(e) (2 marks) Display the rows numbered 2 through 6 of `biking_long`.

My answer: `slice`:

```
biking_long %>% slice(2:6)
## # A tibble: 5 x 2
```

```
##   street  time
##   <chr>   <dbl>
## 1 Rose      869
## 2 Rice      694
## 3 Oxnard    842
## 4 Rose      648
## 5 Rice      629
```

This, if you insist on doing it, also works (and is therefore full marks):

```
biking_long %>% mutate(r=row_number()) %>%
    filter(r>=2, r<=6)
```

```
## # A tibble: 5 x 3
##   street  time     r
##   <chr>   <dbl> <int>
## 1 Rose      869     2
## 2 Rice      694     3
## 3 Oxnard    842     4
## 4 Rose      648     5
## 5 Rice      629     6
```

but it is much easier to remember `slice`. (Why make it difficult for yourself for no reason?)

(f) (3 marks) Display the rows of `biking_long` where the street is Rice.

My answer: Filter, and two equals signs:

```
biking_long %>% filter(street=="Rice")
```

```
## # A tibble: 19 x 2
##    street  time
##    <chr>   <dbl>
##  1 Rice      694
##  2 Rice      629
##  3 Rice      863
##  4 Rice      748
##  5 Rice      767
##  6 Rice      574
##  7 Rice      628
##  8 Rice      637
##  9 Rice      620
## 10 Rice      752
## 11 Rice      608
## 12 Rice      983
## 13 Rice      765
## 14 Rice      666
## 15 Rice      727
## 16 Rice      729
## 17 Rice      605
```

```
## 18 Rice     717
## 19 Rice     679
```

(g) (3 marks) Display the times of the rides in `biking_long` which took 600 seconds or less, along with the street that each one was on.

My answer: Also a `filter`. You don't need anything special to get the streets, because `filter` gives you the whole row:

```
biking_long %>% filter(time <= 600)
## # A tibble: 3 x 2
##   street  time
##   &lt;chr&gt;  &lt;dbl&gt;
## 1 Rice     574
## 2 Rose     552
## 3 Rose     578
```

Read the question carefully: that's less than *or equal* to 600, so if you have strictly less than, you'll lose a point. (The result is the same *here*, but would not be in general.)

(h) (3 marks) Display the times of the five slowest rides, along with the streets they were on.

My answer: This is the five rides that take the *largest* number of seconds. Strategy: sort in descending order, slice off the first five:

```
biking_long %>% arrange(desc(time)) %>%
    slice(1:5)
## # A tibble: 5 x 2
##   street  time
##   <chr>  <dbl>
## 1 Rose    1045
## 2 Rose    1029
## 3 Rice     983
## 4 Rose     869
## 5 Rose     869
```

Expect to lose one if you sort the times without grabbing *only* the first five of them. (That is, two points for getting the `arrange` right, one for getting the `slice` right.)

There's no need to `select` the times and the streets, since that's all there is anyway, but no penalty if you do.

Question 5 (6 marks)

A study was carried out to assess the effects of smoking on exercise. Twenty-seven people were classified into three groups by smoking history as non-smokers (`non`), moderate smokers (`mod`) and heavy smokers (`heavy`). Each person was randomly assigned to one of three types of exercise: a stationary bicycle (`bike`), a treadmill (`tread`), or stair-climbing (`step`). There were three people in each combination of smoking history and exercise type. Each person was asked to begin exercising, and their time, in minutes, until "maximal oxygen uptake" was measured. The longer this time is, the fitter the person is (it means that they can exercise at a steady rate for a longer time).

The data as recorded are shown in Figure 11. The column `id` labels subject *within each group*.

(a) (4 marks) Give R code to arrange these data into a column of smoking-history group called `smoke`, a column of exercise types called `exercise`, and the time to maximal oxygen uptake, called `oxygen`. Bear in mind that there are 27 people in the data set, so your code will need to produce a data frame that has 27 rows. Your answer can contain a column `id` or not. Either is good.

My answer: This is more difficult than the `pivot_longer` from earlier, because each column encodes two things: a smoking-history group and an exercise type. This is one of those things that `pivot_longer` can deal with but `gather` cannot. So there are several possible approaches.

There is a one-step solution with `pivot_longer` that uses the "two things in `names_to`" idea from Problems 13.6 and 13.7 in PASIAS:

```
uptake %>%
    pivot_longer(-id, names_to=c("smoke", "exercise"),
                 names_sep="_", values_to="oxygen")
## # A tibble: 27 x 4
##       id smoke exercise oxygen
##    <dbl> <chr> <chr>     <dbl>
##  1     1 non   bike       12.8
##  2     1 non   tread      16.2
##  3     1 non   step       22.6
##  4     1 mod   bike       10.9
##  5     1 mod   tread      15.5
##  6     1 mod   step       20.1
##  7     1 heavy bike        8.7
##  8     1 heavy tread      14.7
##  9     1 heavy step       16.2
## 10     2 non   bike       13.5
## # ... with 17 more rows
```

This way needs: (i) "everything but `id`", (ii) a `names_to`, (iii) a `values_to`, (iv) a `names_sep`, because you have to say what the two columns to create are separated by in the original data.

Grading: lose one mark per error to a minimum of 1 if something substantial is correct.

In the likely event that you don't think of this, do it in two steps, using a "standard" `pivot_longer` or `gather` (this is the only way `gather` will do it):

```
uptake %>%
    pivot_longer(-id, names_to="treatment", values_to="oxygen")
## # A tibble: 27 x 3
##       id treatment   oxygen
```

```
##    <dbl> <chr>        <dbl>
##  1     1 non_bike      12.8
##  2     1 non_tread     16.2
##  3     1 non_step      22.6
##  4     1 mod_bike      10.9
##  5     1 mod_tread     15.5
##  6     1 mod_step      20.1
##  7     1 heavy_bike     8.7
##  8     1 heavy_tread   14.7
##  9     1 heavy_step    16.2
## 10     2 non_bike      13.5
## # ... with 17 more rows
```

and then use **separate** to make that **treatment** column into the right two things, giving

```
uptake %>%
    pivot_longer(-id, names_to="treatment", values_to="oxygen") %>%
    separate(treatment, into=c("smoke", "exercise"), sep="_")
```

```
## # A tibble: 27 x 4
##       id smoke exercise oxygen
##    <dbl> <chr> <chr>     <dbl>
##  1     1 non   bike       12.8
##  2     1 non   tread      16.2
##  3     1 non   step       22.6
##  4     1 mod   bike       10.9
##  5     1 mod   tread      15.5
##  6     1 mod   step       20.1
##  7     1 heavy bike        8.7
##  8     1 heavy tread      14.7
##  9     1 heavy step       16.2
## 10     2 non   bike       13.5
## # ... with 17 more rows
```

It actually works without the `sep` as well, since the default separator is an underscore. I think it's better to have it in (for clarity), but it works without, so that is also correct.

This way, two marks for the `pivot_longer` and two for the `separate`.

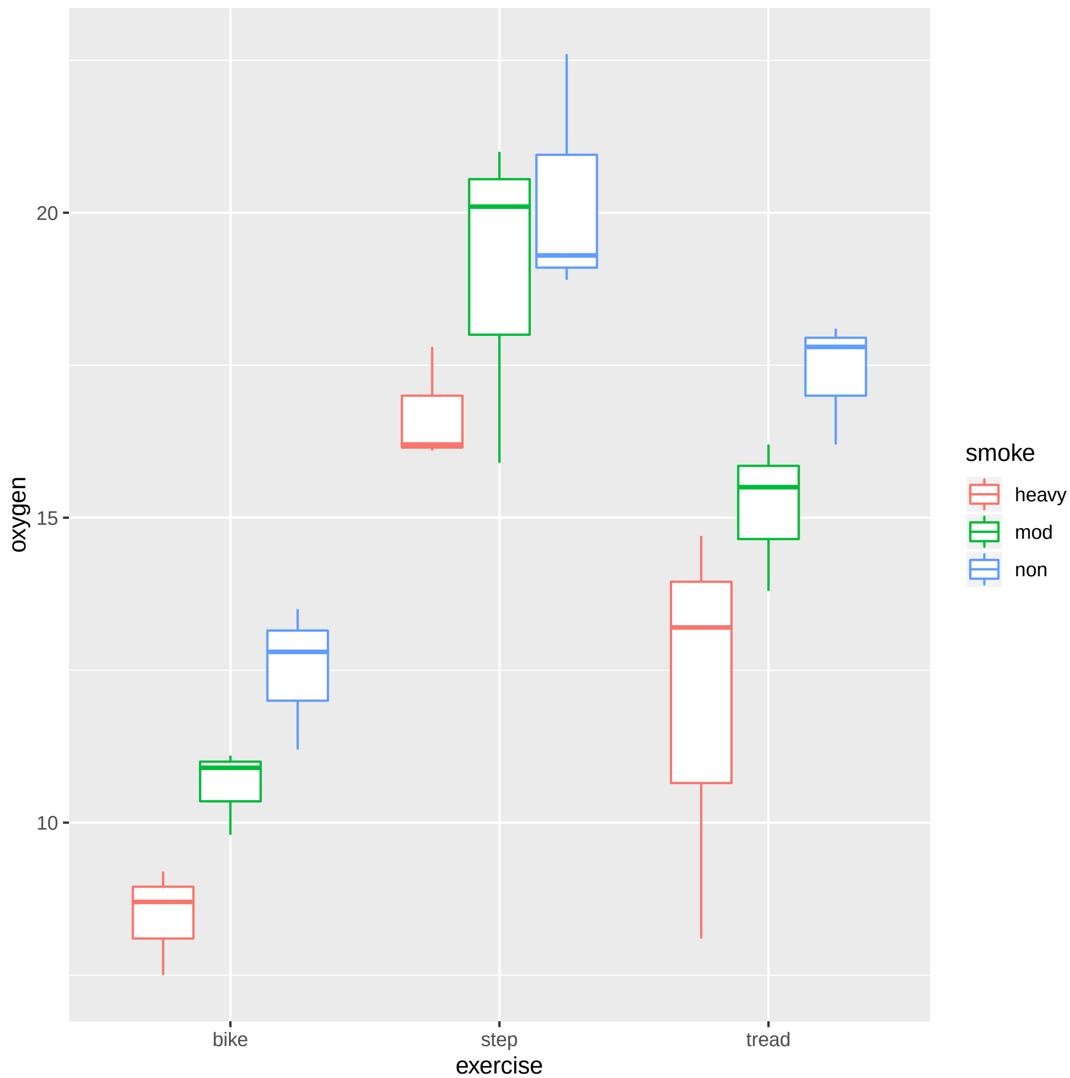
(b) (2 marks) Part of the data set, after you have finished tidying it, is shown in Figure 12. Describe *in words* a suitable graph for this data set, and justify your choice briefly. Think about whether there is any value in including `id` in your graph.

My answer: There is actually *no* point including ID, since (for example) ID 1 refers to nine different people, and each person only provides one measurement. There is no relationship between the nine different people with ID 1. (The time to include ID on a graph is in a matched pairs or repeated measures situation, where you want to know which measurements were on the same person. This is not that.)

Otherwise, this is actually just like the one on the midterm (for 2019 students): one quantitative and two categorical variables, so a grouped boxplot is the thing.

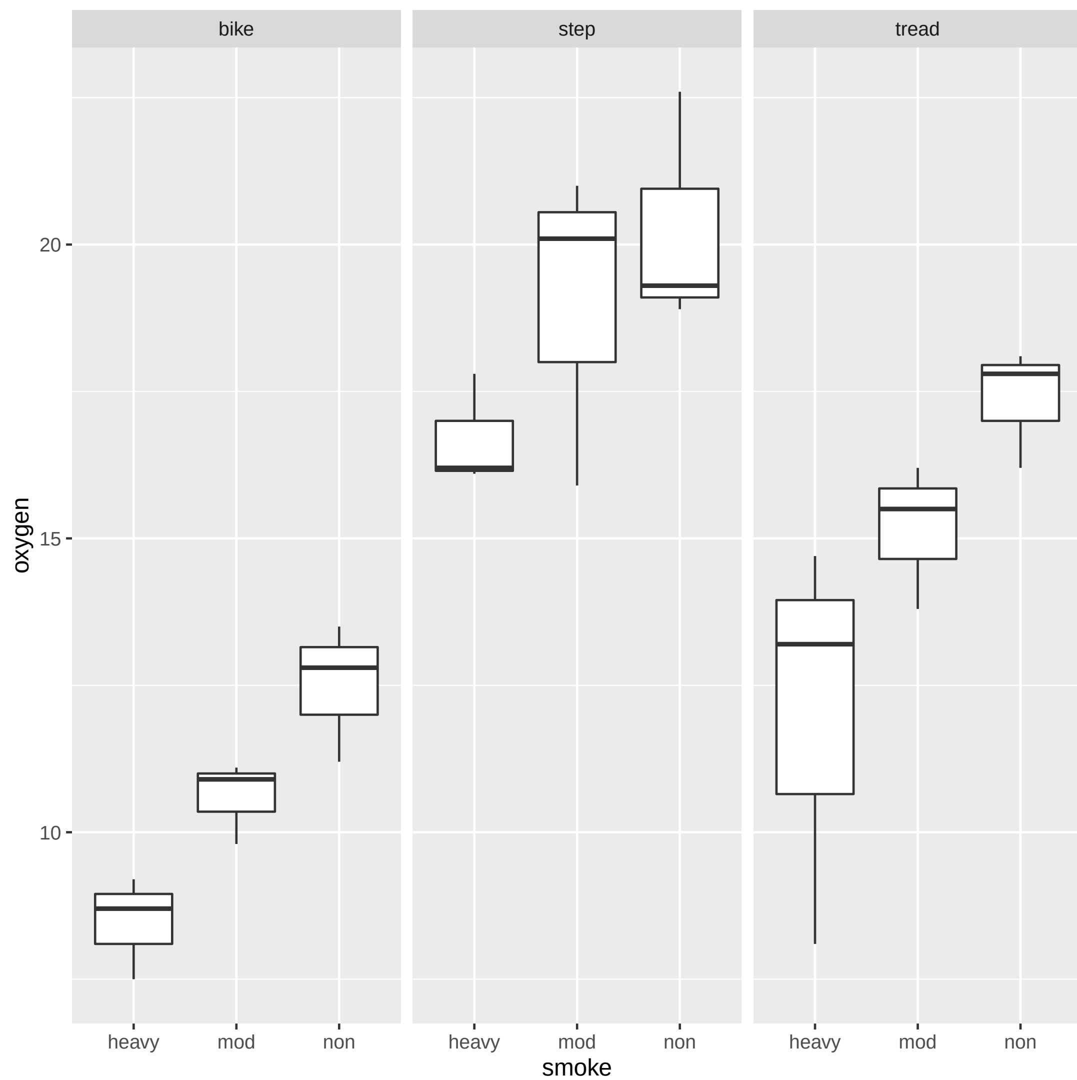
I called my tidied dataset **uptake2**, which is probably a bad name, but I won't be doing any modelling with it here:

```
ggplot(uptake2, aes(x=exercise, y=oxygen, colour=smoke)) + geom_boxplot()
```



If you can make the case for it, facetted boxplots also work (as on the 2019 midterm), such as this one:

```
ggplot(uptake2, aes(x=smoke, y=oxygen)) + geom_boxplot() +
    facet_wrap(~exercise)
```



If you go this way, you should explain how you are going to make boxplots using one of the categorical variables and facets using the other one.

The conclusion from this is pretty clear, even with only three observations per exercise-smoking combination: for any of the exercises, it seems that the more you smoke, the shorter the time until maximum oxygen uptake.

No points for code. Get used to the idea of explaining what you are going to do, in words, to people who know less statistics than you do.

Question 6 (12 marks)

Earlier, we worked with the data in Figure 2. This concerned an analysis of electricity pricing, and the satisfaction of customers with different pricing plans. In each of the parts below, give SAS code to accomplish the task described. This data is stored in the SAS data set `tod`.

(a) (3 marks) Create a new SAS data set from `tod` that contains only the column `satisfaction`.

My answer:

Thus:

```
data tod2;
  set tod;
  keep satisfaction;

proc print;
```

```
Obs    satisfaction

  1          25
  2          26
  3          28
  4          27
  5          31
  6          26
  7          29
  8          27
  9          24
 10          25
 11          28
 12          26
 13          26
 14          29
 15          27
 16          30
 17          25
 18          30
 19          24
 20          26
 21          33
 22          25
 23          28
 24          27
 25          22
 26          25
 27          20
 28          21
 29          33
 30          25
 31          27
 32          27
 33          30
 34          26
 35          31
 36          27
```

I don't need the `proc print`; you can have it or not. I have it here to show the results (to verify that my code works).

(b) (3 marks) Create a new SAS data set that contains the first 8 rows (inclusive) of the data set `tod`.

My answer: This uses the special variable `_N_` and an `if` to grab the rows you want:

```
data tod3;
  set tod;
  if _N_<=8;

proc print;
```

```
Obs    length      ratio    satisfaction

  1    06-hours     2-1              25
  2    06-hours     2-1              26
  3    06-hours     2-1              28
  4    06-hours     2-1              27
  5    06-hours     4-1              31
  6    06-hours     4-1              26
  7    06-hours     4-1              29
  8    06-hours     4-1              27
```

Anything else that works is also good; for example, you could have an `_N_>=1` in there as well, or you could even have something like `if _N_ in (1,2,3,4,5,6,7,8)`.

(c) (3 marks) Create a new SAS data set that contains only those observations where the ratio is `4-1`.

My answer: This puts the logical condition in the `if`:

```
data tod4;
  set tod;
  if ratio="4-1";

proc print;
```

```
Obs    length      ratio    satisfaction

  1    06-hours     4-1              31
  2    06-hours     4-1              26
  3    06-hours     4-1              29
  4    06-hours     4-1              27
  5    09-hours     4-1              25
  6    09-hours     4-1              30
  7    09-hours     4-1              24
  8    09-hours     4-1              26
  9    12-hours     4-1              33
 10    12-hours     4-1              25
 11    12-hours     4-1              27
 12    12-hours     4-1              27
```

The `ratio` is text, so needs to be in quotes.

(d) (3 marks) Obtain *only* the mean `satisfaction` for only those observations whose `ratio` is `8-1`. That is, your code must not calculate any other summary statistics, and must not calculate any means for other ratios. Do this without creating any new datasets.

My answer: The prescription asks for only the mean, not any of the other summary statistics that `proc means` gives, so you have to ask for the mean specifically. The other part of the prescription prevents you from using a `class` in `proc means` (to get means for *all* the ratios). What I am trying to get you to do is this:

```
proc means mean data=tod;
  where ratio="8-1";
  var satisfaction;
```

```
                    The MEANS Procedure

             Analysis Variable : satisfaction

                            Mean
                    ------------
                      27.5000000
                    ------------
```

Three marks for that. Strictly speaking, the `data=tod` is needed because the last data set we created is not the one we want to use here, but I'm willing to forgive that. Two marks for getting the `where` but missing the `mean` on the first line. One for a `proc means` with a `class`. One also for making a new data set with those `ratio` values and finding the mean `satisfaction` in that.

To show why some of the other possibilities don't work:

```
proc means data=tod;
  class ratio;
  var satisfaction;
```

```
                        The MEANS Procedure

                  Analysis Variable : satisfaction

          N
ratio   Obs    N         Mean        Std Dev        Minimum        Maximum
--------------------------------------------------------------------------
2-1      12   12   25.5000000      3.1188576     20.0000000     30.0000000

4-1      12   12   27.5000000      2.7136021     24.0000000     33.0000000

8-1      12   12   27.5000000      2.6798914     24.0000000     33.0000000
--------------------------------------------------------------------------
```

This gets the standard deviation and the min and max as well, and also gets the mean for the other `ratio`s as well. (So this one is wrong twice.)

Or, this:

```
data tod5;
  set tod;
  if ratio="8-1";

proc means mean;
  var satisfaction;
```

```
                    The MEANS Procedure

             Analysis Variable : satisfaction

                           Mean
                   ------------
                     27.5000000
                   ------------
```

This one gets the answer, but does so by creating a new data set, so fails to answer the question for that reason.

Question 7 (15 marks)

Dorothy sells life insurance. She does this by visiting her clients' homes. We want to find out whether it is true that the more home visits she makes, the more people buy insurance from her. She collects data on the number of home visits she makes each week, and the number of life insurance policies she sells. The data are shown in Figure 13.

(a) (3 marks) A scatterplot is shown in Figure 14. Describe any relationship you see. Hint: linear or curved? Up or down? Strong, moderate or weak? Briefly justify your choice about the strength of the relationship.

> **My answer:**
> The relationship looks to me linear (not obviously curved as I see it), clearly going up rather than down, maybe a moderate relationship because I can see an upward trend but there is a lot of scatter.
>
> Make the case, in the last one, for what you think. That is, say *something* plausible about why you think it's moderate (or strong, or weak, or whatever you think).
>
> One point for each of linear, upward. A half point for each of a strength of trend and a justification for it. If you think it's curved, a half point for saying that and a half for supporting that somehow.
>
> (Whether the slope is large or small is unrelated to the strength of the trend, because you could have an only slightly uphill trend that the points are very closely clustered about.)

(b) (2 marks) A regression was fitted to predict sales from visits, with the results shown in Figure 15. Which numbers from the output support your conclusions in (a) about (i) the direction (up/down) of the trend, (ii) the strength of the trend? Explain briefly (one number and a brief explanation for each of (i) and (ii)).

> **My answer:** The upward direction that you hopefully found in (a) is supported by the slope, 0.29, being positive. The moderate strength of the trend is supported by the moderate R-squared of 0.62. (Use the same adjective to describe the R-squared as you did to describe the strength of the relationship; for example, calling the relationship and the R-squared "weak" is fine as long as you did it in both places.)
>
> The P-value for the slope is small, which tells you that the slope is not zero, but that doesn't say anything directly about how strong the relationship is (that's what R-squared does). You can get a significantly non-zero slope and still have the trend look weak (especially if you have a lot of data).

(c) (2 marks) What do you conclude from the P-value on the Intercept line in Figure 15? Does this make sense in the context of the data? Explain briefly.

> **My answer:** The P-value 0.1748 is not small, so the intercept is not significantly different from zero. This makes sense because the intercept is the number of sales Dorothy would make if she makes no visits at all; if she makes no visits, she would expect to make no sales.
>
> One point for "the intercept is not significantly different *from zero*", and one for saying why that actually does make sense. A lot of people only got as far as saying that the P-value was not

significant, which is only 0.5 by itself. (Since Crowdmark adds up the marks, and I no longer have to do it, I am more generous with 0.5s than I used to be. In the old days, if you didn't get as far as "the intercept is not significantly different *from zero*", you didn't get anything.)

Don't get confused here with the test for the *slope* (next part). I want to see that you understand the difference between the intercept (y when $x = 0$) and the slope (increase in y when x increases by 1).

(d) (2 marks) Does Figure 15 support a hypothesis that there really is a relationship between the number of visits in a week and the number of sales in that week? Explain briefly.

My answer: The key here is "really is", which implies a test with a P-value. (That is, there looks as if there is a relationship *for these data*, but the question is whether there is a relationship in *all* weeks, not just the ones sampled.) The P-value required here is the one for the slope, 0.0004, which is very small. This says that the slope is not zero, so that the number of visits is definitely related to the number of sales. (The relationship is stronger than chance.)

Strictly speaking, if you want to say that more visits goes with *more* sales, you ought to say something that indicates that the slope is significantly *greater* than zero, but I'm not going to insist on that here, because we looked at the up/down thing earlier, and that would really be a repeat.

Another way to tackle this is to say that we should do a bigger study with other potentially confounding variables, and eliminate them as a source of the cause and effect. I'd go for that.

(e) (1 mark) Using Figure 15, what would you predict Dorothy's sales to be in a week where she makes 12 visits? (Use your calculator if you need to.)

My answer: This is just substituting 12 into the regression equation:

```
1.62597 + 0.29278*12
## [1] 5.13933
```

I didn't ask for an explanation, but you would do well to at least show your calculation, because there might be a half point if we could see that you made a small error. An answer that is wrong, without explanation or calculation, is a fast zero.

This kind of thing is close enough:

```
1.63 + 0.29*12
## [1] 5.11
```

If you didn't have a calculator, note that, and write down the calculation you would have done. Since we are not using the answer for anything, I'm happy with that. (But you have to tell me that you don't have a calculator; otherwise, I want the answer.)

Only one mark for this, since you have probably known how to do this since your *first* statistics course. But I wanted to have something concrete to talk about in the next part.

The point of doing a regression is that you have a straight-line relationship and you want to use it to predict with. Going back to the data and trying to find a number of visits close to 12 (and seeing how many sales there were that week) doesn't take advantage of the fact that the *whole* relationship appears to be straight.

I know we said earlier that the intercept was not significantly different from zero, but if you want to do a prediction, you use the best values you have for everything, including 1.63 for the intercept. The other way to go is to fit a regression where the intercept is constrained to be zero, which will give you a *different* slope. R will always include an intercept unless you explicitly tell it not to. The intercept is called `1` in R, and you remove it thus:

```
## Parsed with column specification:
## cols(
##   visits = col_double(),
```

```
##  sales = col_double()
## )
insurance.1a <- lm(sales~visits-1, data=insurance)
summary(insurance.1a)
##
## Call:
## lm(formula = sales ~ visits - 1, data = insurance)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5231 -0.7681  0.1209  0.8626  3.8593
##
## Coefficients:
##        Estimate Std. Error t value Pr(>|t|)
## visits  0.37582    0.02576   14.59 7.35e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.795 on 14 degrees of freedom
## Multiple R-squared:  0.9383,Adjusted R-squared:  0.9339
## F-statistic: 212.8 on 1 and 14 DF,  p-value: 7.353e-10
```

(My model-numbering scheme got messed up: there is a model `insurance.2` later, and I didn't want to re-number everything.)

You see that there is no intercept, and now the slope is a bit bigger than it was before. (Also, the R-squared appears to be much bigger, but this is deceiving because R-squared is calculated *by a different method* when you don't have an intercept, and the values of R-squared with and without intercept are not comparable).

You could then do a prediction from this model, which really would be 12 times the slope:

```
12*0.37582
## [1] 4.50984
```

This is at least in the same ballpark as the prediction from the output I gave you. We haven't done anything like this in class, so I'm by no means expecting you to come up with it yourself, but the point is that you do a prediction using all the estimates you have (the intercept and slope in our case). If you want to use only some of the estimates (because, say, you believe that the intercept should be zero), you first have to fit a model without the things you believe have zero estimates, and then base your predictions on the new model.

This idea is actually the same as doing predictions from multiple regression: either you use *all* the slopes you have, or you get rid of the x's that you think are not important (which is the same thing as making their slopes zero), *re-fit the model*, and then base your predictions on the new slopes, which will be different from the old ones.

(f) (2 marks) How accurate would you expect your prediction of the previous part to be: highly accurate, moderately accurate, not accurate at all, or something else? Cite something from the output to support your answer, and explain briefly.

My answer: If the regression fits well, the data are all close to the line, and you would expect the prediction to be accurate; if not, not. So cite something like R-squared, and use it to support your answer: for example, I would say that the R-squared is only moderately high, and so the prediction will be only moderately accurate. (You can choose your adjective here; if you want to call this R-squared "low", do so and say that the prediction will not be very accurate.)

Another approach would be to go back to the scatterplot in Figure 14, and say (for example) that the points are not very close to a line, so predictions based on these data will not be very accurate.

I think you can only use the small P-value for the slope if you also say something like "the slope is accurately estimated, therefore the prediction is likely to be good". This is still not really complete, because the *intercept* is poorly estimated (the estimate of the intercept is 1.62 which is still not significantly different from zero). If you were able to say that both slope and intercept were accurately estimated, then you would have support for predictions being accurate also. If you talk about accuracy of estimation for both slope *and* intercept (one good, the other bad, and therefore overall moderate), I'm OK with that. It depends on how far from zero the numbers of visits are; if they were all way above zero, it would matter less that the intercept was poorly estimated.

Pick *one* thing and use it to support your answer. I might be OK with two things, but three or more things, especially if you don't say which one you prefer, makes me think that you are just guessing. Expect only one point if you are not focused enough.

Careful about what you think R-squared means: it is *the proportion of variability (variance) in the response explained by the fact that the response depends on the explanatory variable.* "Explained by the data" is sloppy. You can do better.

(g) (3 marks) Two more plots, ones that normally go with a regression, are shown in Figures 16 and 17. What are these plots, and what do you conclude from them? Explain briefly. (You need to say three things: a conclusion from each plot (including a statement of what it is), and an overall conclusion.)

My answer: Figure 16 is a normal quantile plot of the residuals. I think this is OK except for the most positive residual at the top right, which indicates an outlier (a point far above the line). If you look back at Figure 14, this seems to be the week in which Dorothy made 19 visits and had an unusually high 11 sales. I think you need to notice this outlier. You could also say something along the lines of the plot showing a slight upward-opening curve, which would indicate a right-skewed distribution of residuals. I like this less, because I would like the curve to be more pronounced for you to say this (the middle points are basically on the line, rather than below it). Minus a half for claiming skewness, I think.

I tightened up the question so that I make sure you understand that this is a *normal quantile plot of the residuals*, and not some kind of scatterplot. The clue that it's a normal quantile plot is the labels on the axes; the clue that it's a normal quantile plot *of the residuals* is given in the question, since no other normal quantile plots typically go with a regression. I would accept "normal quantile plot" as a description of what it is, but in your explanation you need to say something about what it is that is or is not normal (the residuals).

Figure 17 is a plot of the residuals against the fitted values. (Look at the axes again.) This shows a nice random pattern except possibly for that point at the top which is the very positive residual again. (Feel free to describe this plot as random; spotting the outlier at the top here is optional.)

The overall conclusion seems to be that the regression is satisfactory apart from that one outlier. If you prefer, "the regression is not satisfactory because of the one outlier". If you talked about skewness in the normal quantile plot, you should here express dissatisfaction with the whole regression, because skewness implies a problem with the whole distribution of residuals. Remember that you want both plots to be satisfactory, and if not, there is a problem with the regression (that ought to be fixed, but that is farther than I want to go here).

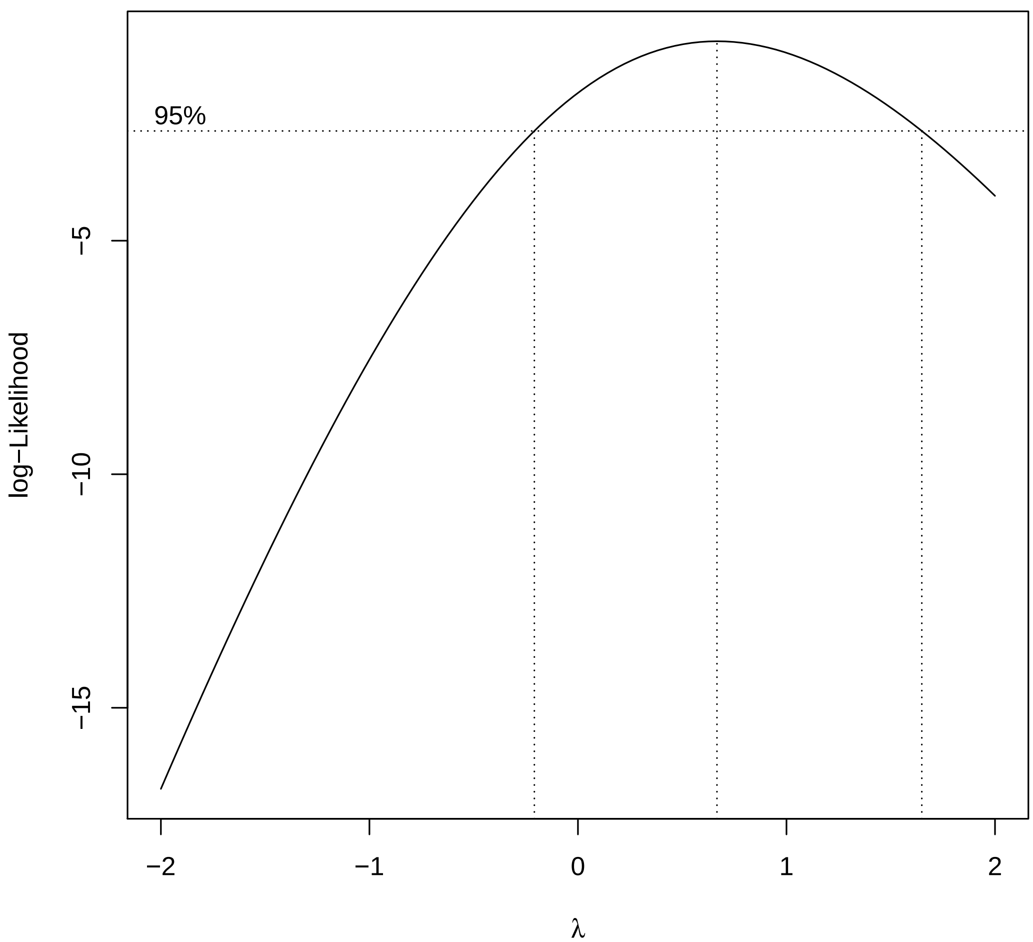
In addition to saying what each plot is, it is smart if you say what you are looking for: points close to the line in the first one, and a random scatter of points in the second. Your overall conclusion needs to match what you said about the two plots; for example, if you said that there were no problems anywhere, you would lose something for not seeing the outlier (on the normal quantile plot), but it would be correct then to say that the regression as a whole is satisfactory.

I really don't think you can find any other fault with the second plot (other than, perhaps, by looking at it too long!).

I am being fairly picky on the grading on this one, because I want you to demonstrate to me that you really know what you are doing: it's a normal quantile plot *of the residuals*, from which we conclude that the residuals are normally distributed as they should be (or are not, depending); the second plot is used to demonstrate that the *residuals* are random (or have no relationship to the fitted values), and the overall conclusion is that the regression is (or is not) satisfactory, based on what you think about the graphs (are they both good enough?). A point for each of your discussions of Figure 16, 17 and overall conclusion. I gave quite a lot of half points (if I thought you were definitely missing something, but you also had something substantial correct, such as not seeing the outlier on the normal quantile plot, or if you didn't express clearly enough what the plot was or what it was for). Hint: if you find yourself using words like "the data" on a question like this, it's a sign that you can say things more clearly; what is it about "the data" that has to be normal, or random, or whatever?

Extra: if you find a problem like skewness in the distribution of the residuals, what that would suggest is a transformation of the response variable, the number of policies sold. This is a count, and counts often respond well to something like a square root transformation (that brings the big values down a bit, but not as much as a log transformation would):

```
library(MASS)

##
## Attaching package:  'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

boxcox(sales~visits, data=insurance)
```



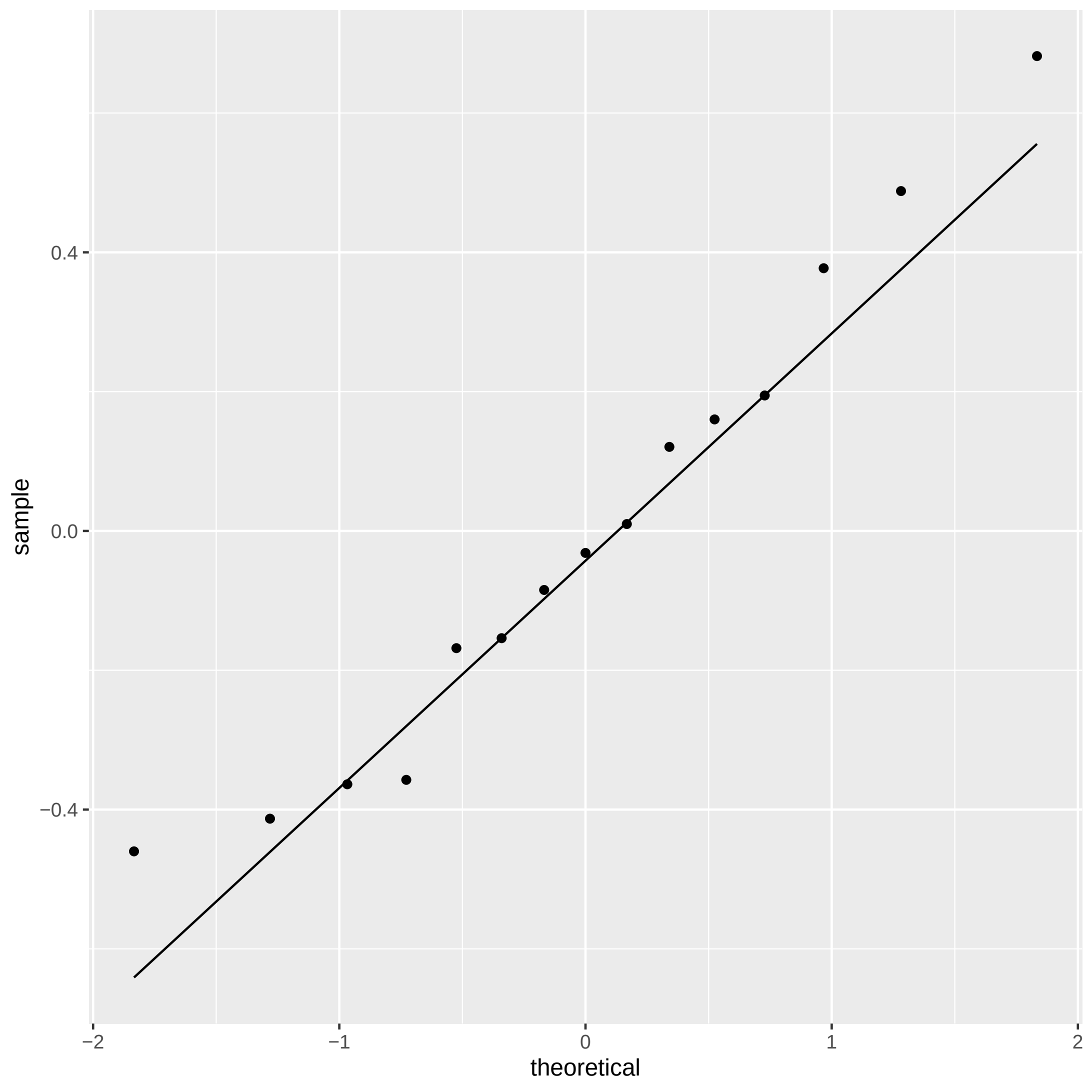
Power 0.5 (square root) is inside the confidence interval, but so is power 1 (do nothing), so there isn't really any justification for doing anything transformation-wise. However, the big residual was a positive one, so maybe bringing it down a bit will help:

```
insurance.2=lm(sqrt(sales)~visits, data=insurance)
summary(insurance.2)
```

```
##
## Call:
## lm(formula = sqrt(sales) ~ visits, data = insurance)
##
## Residuals:
```

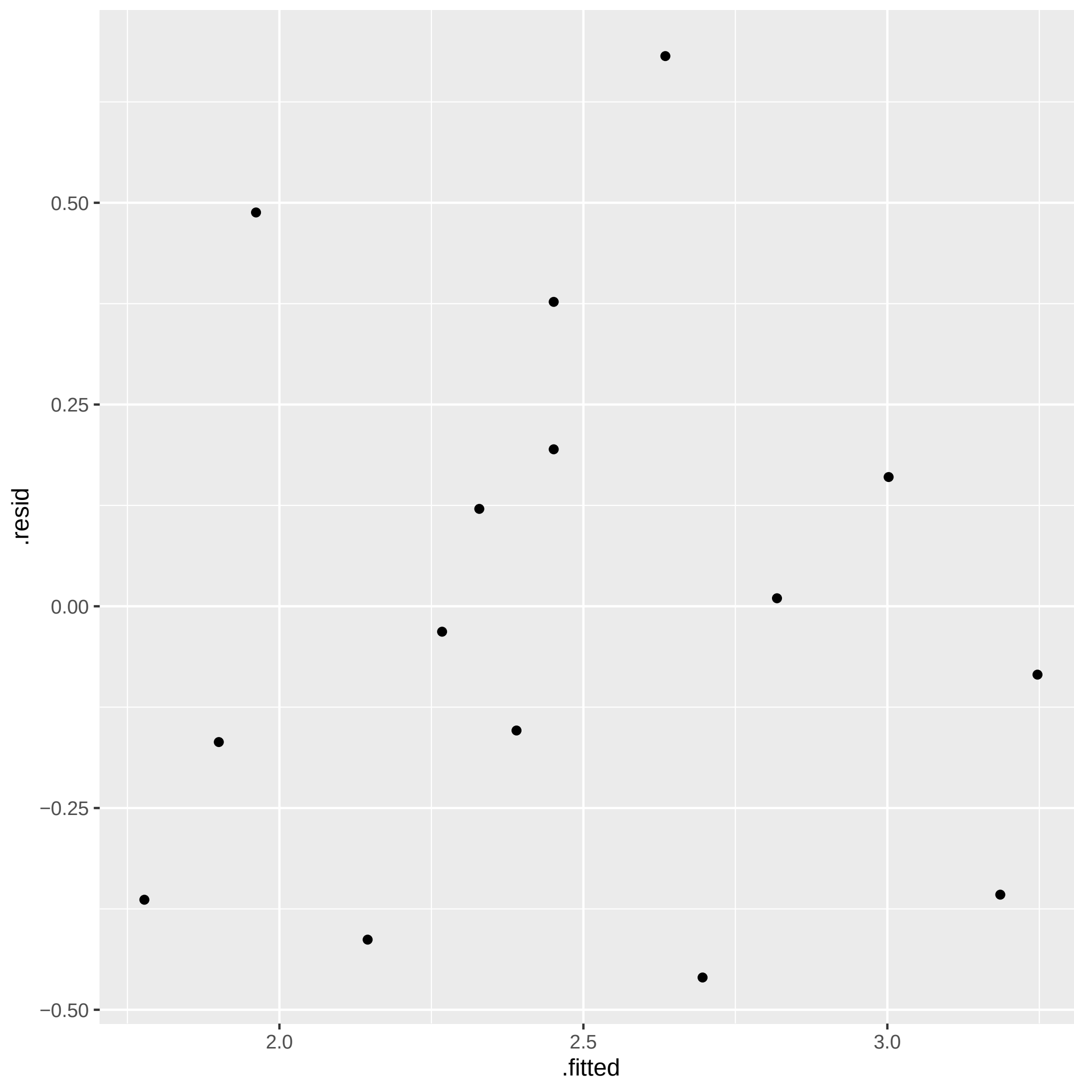
```
##      Min       1Q   Median       3Q      Max
## -0.45999 -0.26278 -0.03149  0.17735  0.68178
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.47179    0.23145   6.359  2.5e-05 ***
## visits       0.06121    0.01286   4.759 0.000374 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3536 on 13 degrees of freedom
## Multiple R-squared:  0.6353,Adjusted R-squared:  0.6072
## F-statistic: 22.64 on 1 and 13 DF,  p-value: 0.0003736
```

```
ggplot(insurance.2, aes(sample=.resid)) + stat_qq() + stat_qq_line()
```



and, for confirmation:

```
ggplot(insurance.2, aes(x=.fitted, y=.resid)) + geom_point()
```



My opinion is that the residuals vs. fitted values is still good, but the lowest values on the normal quantile plot have been brought up a bit too much, while bringing the highest value down, so there is a little more evidence of skewness there than there was before.

As ever, you would make a call and defend it. There isn't a "best" answer here.

Question 8 (16 marks)

The life expectancy is measured as the number of years a baby born today can expect to live. This typically depends on the country a baby is born in, or on variables that say something about that society.

The data in Figure 18 show, for a number of countries, the life expectancy in years (male and female averaged), the number of televisions per person, and the number of doctors per 1000 people.

(a) (3 marks) A regression model is shown in Figure 19. Would you have expected each of the three numbers shown in the Estimate column to be positive, given what the data represent? Explain briefly, for each one.

My answer: If TVs per person is bigger, the positive slope means that the life expectancy will be bigger. The same is true if the number of doctors per 1000 people is bigger; the life expectancy will also be bigger. These make sense because the more TVs or doctors a country has, the more developed it is and the longer its people are likely to live. (In the case of doctors, this is a direct connection: if there are more doctors, there is more likely to be a culture of going to see the doctor when you get sick, and serious illnesses will be caught sooner.)

The intercept is positive as well: this is the predicted life expectancy for a country with no TVs or doctors, and even in that case, people will live for some positive number of years. (They cannot live for a negative number of years, in any case.)

In each case, I would like to see some comment about what each Estimate value represents, and why its being positive makes sense for the kind of data we have. (This is meant to be a warmup for the rest of the question.)

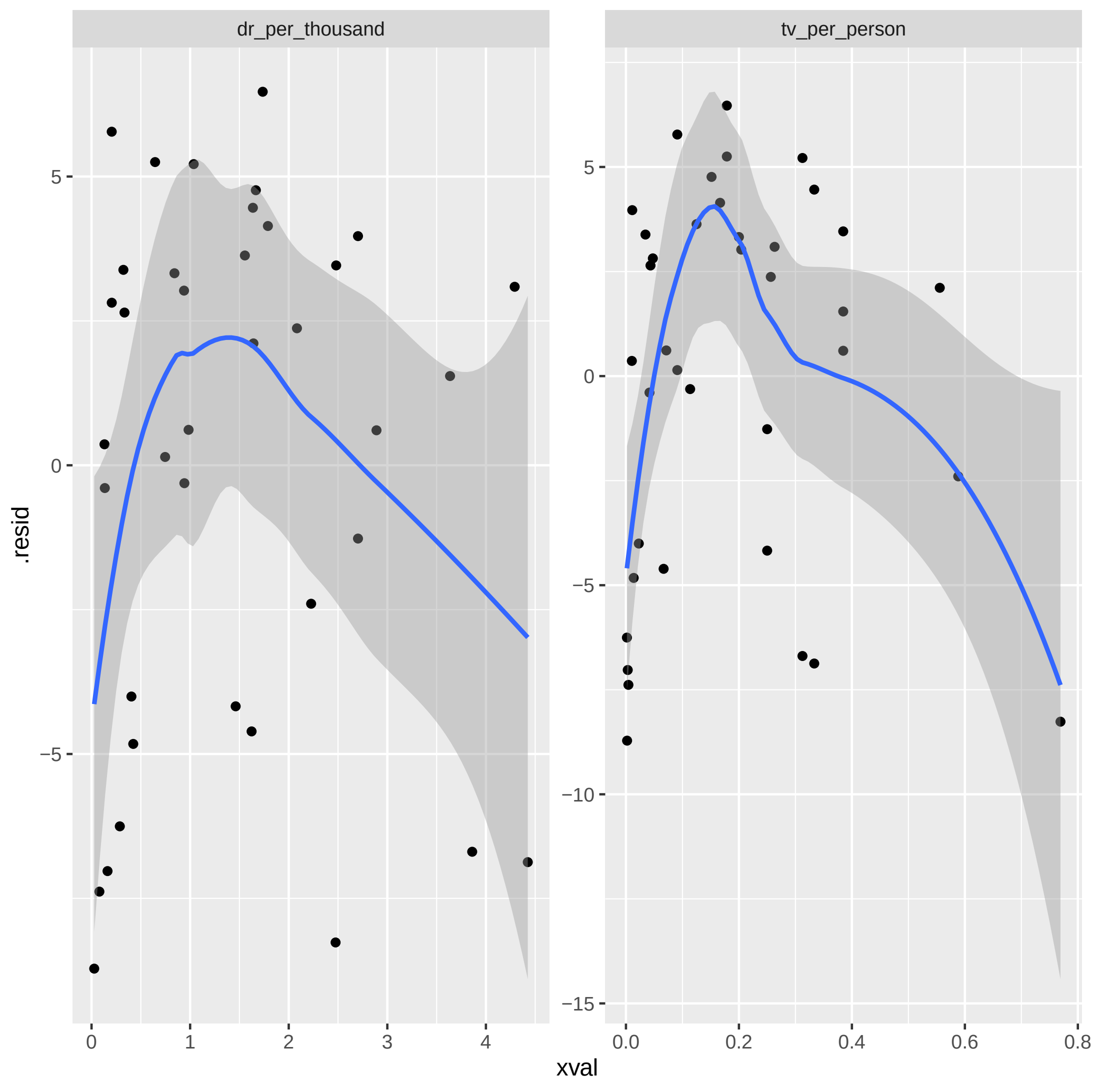
If this is not obvious to you, go back and look at the data in Figure 18. Find some countries where the life expectancy is large, and see what they have in common (being developed countries, or rich countries, or something like that). Or find where the life expectancy is small, and see what *they* have in common: being third-world countries, or being poor countries, or something like that. Eyeball the numbers of doctors per thousand and TVs per person for the countries you picked, and see that they are typically large (if you picked rich countries) or small (poor countries) compared to the others. (Of course, if you were sitting in front of a computer, you would draw some graphs, and this is probably the first thing I'd have you do if that were the case.)

(b) (4 marks) The plots of residuals against the explanatory variables are shown in Figure 20. Give the R code that was used to produce this plot, using the data frame shown in Figure 18, which is called `life`, and the regression model object `life.1`.

My answer:

This is what I did:

```
life.1 %>% augment(life) %>%
    pivot_longer(contains("per"), names_to="xname", values_to="xval") %>%
    ggplot(aes(x=xval, y=.resid)) + geom_point() + geom_smooth() +
    facet_wrap(~xname, scales="free")
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



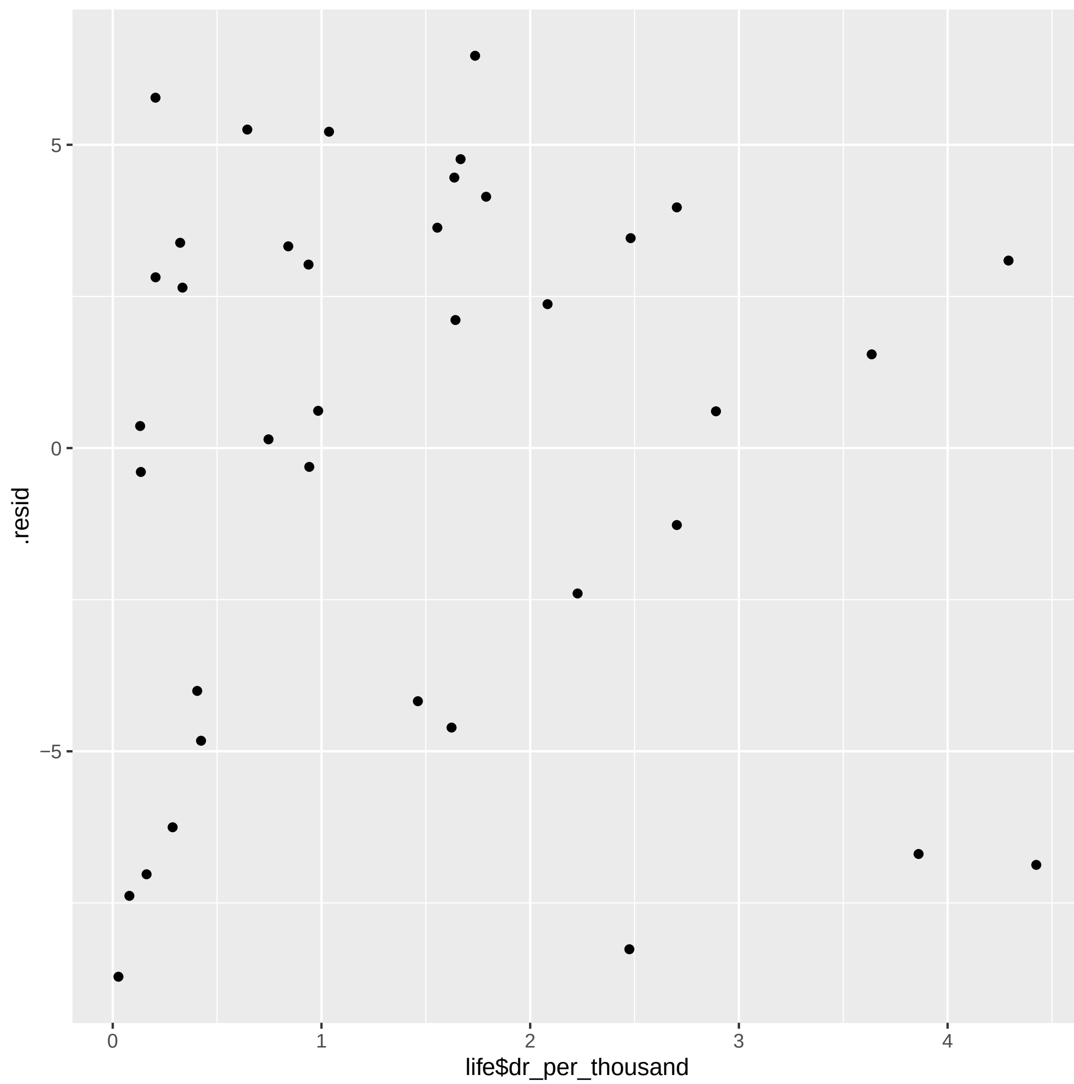
You need the `augment`, the `pivot_longer` or equivalent `gather`, the `ggplot`, and the `facet_wrap` with `scales="free"`. There is basically a point for each of those, so expect to lose a half point for each of the details you miss.

You have some freedom; the new names for the columns after the `pivot_longer` can be what you like, as long as you use them again when drawing the plot, and you can select the columns to pivot-longer in whatever way that will work, for example `tv_per_person:doctor_per_thousand`, or by joining them together with `c()`. Also, you can save the result of the `pivot_longer` in a data frame, and then use that data frame in your `ggplot`.

This is complicated, and getting it right shows good attention to detail.

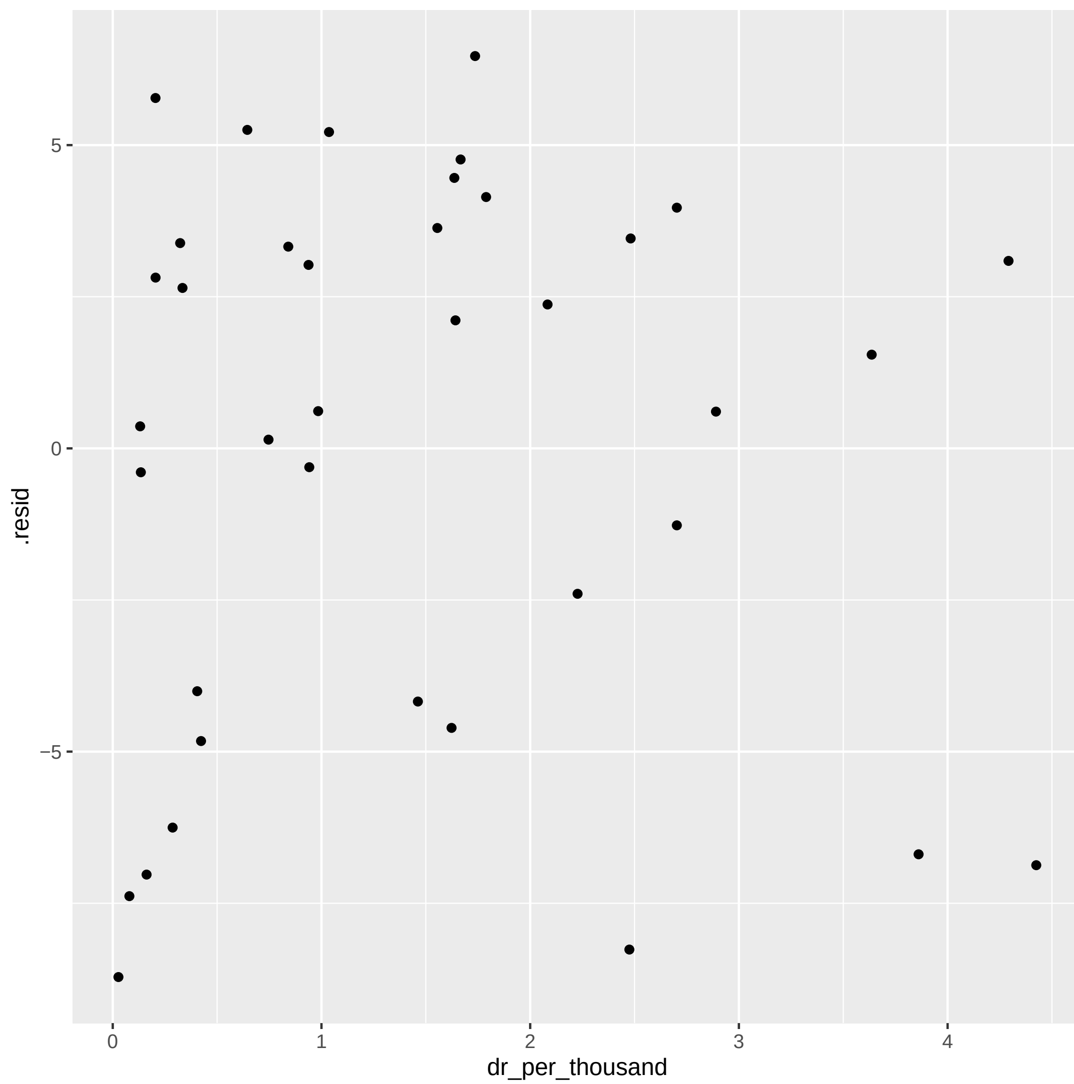
Expect to get about half marks for correctly producing the two residual-vs-x plots separately without a `facet_wrap`, since your instructions were to produce the plot in the Figure, which requires it. You will still need to get the residuals from the regression object and the x-values from the original data, using `augment` or otherwise, eg.:

```
ggplot(life.1, aes(x=life$dr_per_thousand, y=.resid)) + geom_point()
```



or

```
life.1 %>% augment(life) %>%
    ggplot(aes(x=dr_per_thousand, y=.resid)) + geom_point()
```



and then again for `tv_per_person`.

(c) (3 marks) What do you conclude from Figure 20, and thus how would you proceed with model-building? You may assume that the normal quantile plot of residuals and the plot of residuals against fitted values are satisfactory.

My answer: These two plots both show curves. (One point.) Since the other plots are satisfactory, the implication is to do something with these x's (one point), such as adding squared terms in them both. (One point.) (If you get the third point, that implies that you knew what to do for the second one, even if you didn't say so, and so if you note the curves and immediately talk about adding squared terms in TVs per person and doctors per thousand people, you are good.)

I'm also happy if you go a step further and note that the curves appear to go up fast and down slower, and so some other kind of curve might be better (since a squared term implies going up and down at the same rate). We haven't seen anything specific in the course like this that you might suggest, so this kind of comment is enough to make.

If you somehow come to the conclusion that both plots are satisfactory (which is of course wrong), and *then* say that the model that was fitted does not need to be changed, you get one point for making a valid deduction (do nothing) from an invalid conclusion (plots are satisfactory).

(d) (2 marks) I fitted another model (not shown) that was supposed to improve things. The residuals against fitted values and the normal quantile plot of the residuals are both again satisfactory. The residuals against the explanatory variables are shown in Figure 21. Do you think, on the basis of this Figure, that the model I fitted is now satisfactory? Explain briefly.

My answer: You're looking for randomness on both plots.

I would say that the patterns of points on both plots are now acceptably normal with no patterns, and so I would declare the regression satisfactory. I put the smooths on these, which is somewhat deceiving, because you don't want to take them too seriously; these are "inconsequential wiggles", not meaningful, because the points are all over the place and the grey envelopes include zero all the way across. Another way to see this is that the points are all over the place, not close to any curves you might see at all.

If you want to say something else, go ahead, but you need to be specific about what is wrong. For example, you might say that the right-hand TV-per-person plot has something like a cubic curve on it (one that bends twice). If you say that, you do well to say that we should try a cubic term in TVs per person. I don't think this is really supported by the data, though; the points on the graph don't really follow the up and down and up of the curve. (The "up-again" part is really caused by that point over on the right, and trends caused by single points are not really trends.)

Another possibility is fanning-in; the points on the left side of the TV-per-person graph are more spread out than the ones on the right. But I think the major reason for this is that there are *more* points on the left and *fewer* on the right, so the ones on the left have more opportunity to have residuals far from zero. (The range of a larger sample will typically be bigger than the range of a smaller sample, even if they are both samples from a distribution with the same SD. Fire up R and do some simulations to check this. There is some theory on this, related to "normal order statistics", that says (roughly) that if you take a sample of size n from a normal distribution, the smallest value in your sample will probably be smaller and the largest value will probably be bigger, the larger n gets. This is related to the "theoretical" values that R uses for a normal quantile plot: the extreme ones will also be more extreme if the sample is larger.) The remedy, if you think there is fanning-in, is "a different transformation of TVs per person"; you do well to say this much, even if we have not said more about this kind of phenomenon in this course.

If you want to say that these plots are not both satisfactory now, you will have to be *very* convincing to get full marks. I really don't there is any case to be made against the left-hand plot; it is about as good as you could ever wish to see. For the right-hand plot, you will need to convince me that there is a problem and it would help to suggest a remedy.

Extra: this is the model I fitted (which you will see again in the next part):

```
life.2=lm(life_exp~tv_per_person+dr_per_thousand+
            I(tv_per_person^2)+I(dr_per_thousand^2), data=life)
summary(life.2)
##
## Call:
## lm(formula = life_exp ~ tv_per_person + dr_per_thousand + I(tv_per_person^2) +
##     I(dr_per_thousand^2), data = life)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.9222 -2.7419  0.5167  2.1453  6.1920
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           56.7129     1.2969  43.729  < 2e-16 ***
## tv_per_person         58.3550    12.6006   4.631 5.45e-05 ***
## dr_per_thousand        4.8297     2.1168   2.282  0.02909 *
## I(tv_per_person^2)   -56.2349    16.2538  -3.460  0.00151 **
## I(dr_per_thousand^2)  -0.9009     0.4367  -2.063  0.04703 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.77 on 33 degrees of freedom
## Multiple R-squared:  0.7918,Adjusted R-squared:  0.7666
## F-statistic: 31.37 on 4 and 33 DF,  p-value: 8.006e-11
```

The squared terms help (so the relationship really curves). Does adding a cubic term in TVs per person help, in addition?

```
life.3 <- update(life.2, .~.+I(tv_per_person^3))
summary(life.3)
##
## Call:
## lm(formula = life_exp ~ tv_per_person + dr_per_thousand + I(tv_per_person^2) +
##     I(dr_per_thousand^2) + I(tv_per_person^3), data = life)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.1707 -2.4473 -0.0894  2.4756  6.9630
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            56.2363     1.3859  40.577  < 2e-16 ***
## tv_per_person          78.1069    23.7812   3.284  0.00248 **
## dr_per_thousand         4.3358     2.1773   1.991  0.05503 .
## I(tv_per_person^2)   -134.4757    81.5109  -1.650  0.10877
## I(dr_per_thousand^2)   -0.7898     0.4514  -1.750  0.08978 .
## I(tv_per_person^3)     74.3381    75.8879   0.980  0.33464
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.772 on 32 degrees of freedom
## Multiple R-squared:  0.7979,Adjusted R-squared:  0.7663
## F-statistic: 25.26 on 5 and 32 DF,  p-value: 3.053e-10
```

Absolutely not. So that maybe-cubic curve was an illusion.

(e) (4 marks) An alternative model is fitted, with output shown in Figure 22. This may or may not be a model that you would recommend, based on your earlier answers. Look at the two additional estimates, compared to the regression output shown in Figure 19. What do their signs (positive or negative) tell you about the form of this relationship? In addition, does this form of relationship with these estimates make sense in the context of these data? Explain briefly. Hint: a quadratic $y = ax^2 + bx + c$ has a maximum or minimum at $x = -b/2a$.

My answer: Both of the squared terms have *negative* estimates. This means that the relationships with both explanatory variables are curved with a *maximum*. One point. (I talked about this in class, or if you have calculus, you can remember that it depends on the *second* derivative, which is here a constant $2a$ (not depending on x); if the second derivative is positive, it's a minimum, and if it is negative, a maximum. That you can also reason out by noting that a negative second derivative means the slope is becoming more negative, so it starts out positive, passes through zero, and is then negative.)

To go further, figure out where those maxima are. These are two independent quadratics added together, so you can find the maximum of each one separately. For TVs per person:

```
-58.3550/(2*(-56.2349))
## [1] 0.5188504
```

and for doctors per thousand people:

```
-4.8297/(2*(-0.9009))
## [1] 2.680486
```

Use your calculator to work these out. If you didn't have a calculator at the exam, write down what you would calculate and, if you can, estimate what the answers would be. (The first one would be about a half, since it is a number divided by almost twice the same thing, and the second one is about 5 divided by 2, or 2.5, since the 0.9009 is close to 1.) Estimating will get you close enough for what you need later. (Save yourself some issues by noting that the formula for the maximum has a minus sign in it, and so does the estimate for each squared term, so the minus signs will cancel out and the answers will be positive.) A second point for figuring out the maxima.

For the third and fourth points, we need to figure out whether the relationship is going up and down again, just up (at a decreasing rate), or just down (at an increasing rate), for the data we have. There is no guarantee that a quadratic with a maximum *must* go up *and* come down again within the range of the data. (See the windmill example from class, where it basically kept going up and only came seriously down again above the range of wind velocities that we observed.) That means seeing where the maxima are relative to the data. Asserting that the curve has a maximum and thus that the relationship inappropriately (see below) goes up and down is at most three out of four, unless you somehow convince me that both the up and down parts are within the range of the data. The discussion I want is below. Assembling the ideas into a coherent whole (what I would call "joining the dots") is key here.

To figure out the range of the data, look back at one of the graphs of residuals vs. explanatory, such as Figure 21. This shows doctors per thousand going up to about 4.5 and TVs per person going up to about 0.8, starting from a minimum close to zero both times. Hence, some of the data values are above the maximum and some below. The third point.

Another way around this is to do some predictions yourself (using your calculator, since you don't have R as I do below). If you can find some values of the two x-variables, in the range of the data, such that the predicted life expectancy goes up and then down again, then that would show the form of the relationship and I don't need you to find the maxima using the hint in the question. Some values to try are the minima and maxima observed, or something close to them, and the value halfway between them. That should be enough to show the pattern. Vary one x and *fix* the other one, so that you only have three calculations to do each time. For example, if you're trying to see what happens with doctors per thousand people, you can try values 0, 2, and 4 for that, and use a middling value like 0.4 for TVs per person.

The fourth point is for what that actually *means* in terms of the data. The idea is that for both variables, some of the data values are less than the maximum and some are greater, so the curve goes up and down again. That means that once each explanatory variable passes the maximum, the predicted life expectancy goes down again. This makes no practical sense, because if there are more TVs or doctors in a country, you'd expect life expectancy to keep going *up*: maybe at a decreasing rate, but still up, and definitely not down.

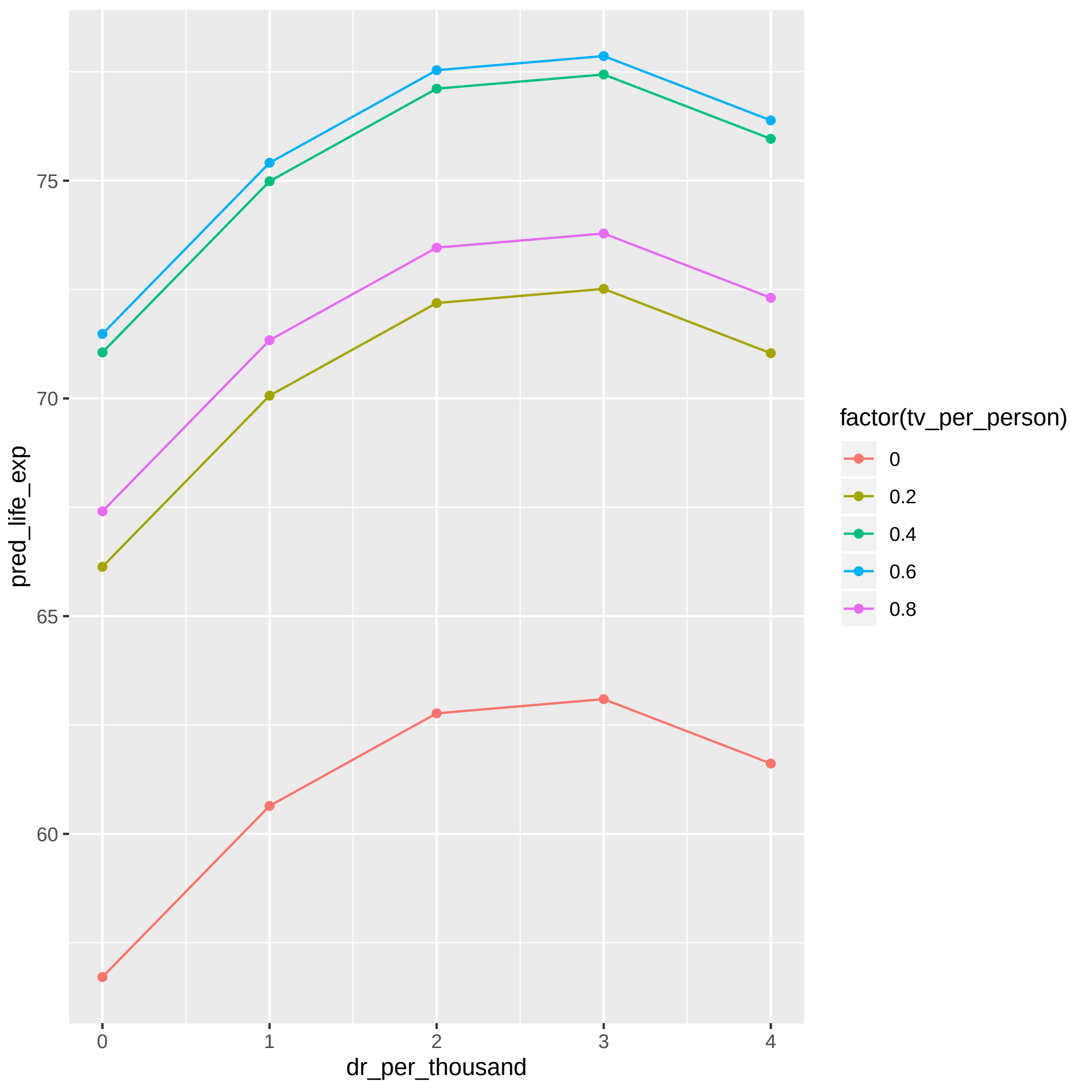
Yes, that's tricky. I wanted to see who could do some detective work and figure this out. The information is all there; you need to find it and make use of it. If you go the prediction way, it will take you some time, which is why I made this the last part of the last question.

Extra 1: I wanted to verify that my thinking was right (about the relationships going up and then down again). I can do that by running some predictions (rather in the spirit of one of those examples from PASIAS, such as part (h) of problem 14.15): pick some representative values for doctors per thousand and TVs per person, predict using our model with the squared terms, and examine:

```
docs <- 0:4
tvs <- seq(0, 0.8, 0.2)
new <- crossing(dr_per_thousand=docs, tv_per_person=tvs)
p <- predict(life.2, new)
preds <- bind_cols(new, pred_life_exp=p)
preds
## # A tibble: 25 x 3
##    dr_per_thousand tv_per_person pred_life_exp
##              <int>         <dbl>         <dbl>
##  1               0           0            56.7
##  2               0           0.2          66.1
##  3               0           0.4          71.1
##  4               0           0.6          71.5
##  5               0           0.8          67.4
##  6               1           0            60.6
##  7               1           0.2          70.1
##  8               1           0.4          75.0
##  9               1           0.6          75.4
## 10               1           0.8          71.3
## # ... with 15 more rows
```

This is hard to imagine, and would be clearer in a picture:

```
ggplot(preds, aes(x=dr_per_thousand, y=pred_life_exp,
                  colour=factor(tv_per_person))) +
    geom_point() + geom_line()
```



There are lots of details here: the `crossing` gives all combinations of values of the variables you feed it (the values in the first two lines of code). `predict` takes a model and predicts the response using all the values in `new` (this really belongs in D29). `preds` contains the values of doctors-per-thousand and TVs-per-person that I predicted for, along with the predicted life expectancy. In the graph, we actually have three quantitative variables, so I made `tv_per_person` categorical for the purposes of plotting. It has only a few distinct values, so that is not so bad.

The relationship between life expectancy and doctors per thousand people, *from the model*, clearly goes up (until about 3) and then down again. Also, if you check the coloured traces, the model says that life expectancy goes up with TVs per person until that reaches 0.6, and then it goes down again. These are consistent with the maximum values I found before. Because it makes no sense that the predictions go up and then back down, this means that the *model with squared terms* does not work for these data, even though it is satisfactory in other ways (eg. the residuals seem well-behaved).

Extra 2: what the up and down seems to mean is that we have the wrong kind of curve here, and maybe something like an asymptote model (like the windmill data) would make more sense. The problem with that here is that some of the countries (the poorest ones) have very few doctors or TVs, so that using something like $1/x$, which would be people per doctor or people per TV, will produce a few very large numbers. And I don't like regressions with unusually large x's, because such x's can be very influential over where the line goes.

The moral of this story is that squared terms can be good for curves, but not for curves that need to continue going up. This much is the same kind of story that we got from the windmill example, though I don't have a nice resolution here as I did there. (The place where I got the data from started with people per doctor and people per TV, and some of the values were *very* big.)

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