Assignment 6

Instructions: Make an R Notebook and in it answer the questions below. When you are done, hand in on Quercus the *output* from Previewing (or Knitting) your Notebook, probably an `html` or `pdf` file. An `html` file is easier for the grader to deal with. Do *not* hand in the Notebook itself. You want to show that you can (i) write code that will answer the questions, (ii) run that code and get some sensible output, (iii) write some words that show you know what is going on and that reflect your conclusions about the data. Your goal is to convince the grader that you *understand* what you are doing: not only doing the right thing, but making it clear that you know *why* it's the right thing.

Do *not* expect to get help on this assignment. The purpose of the assignments is for you to see how much *you* have understood. You will find that you also learn something from grappling with the assignments. The time to get help is after you watch the lectures and work through the problems from PASIAS, via tutorial and the discussion board, that is *before* you start work on the assignment. The only reasons to contact the instructor while working on an assignment are to report (i) something missing like a data file that cannot possibly be read, (ii) something *beyond your control* that makes it impossible to finish the assignment in time after you have started it.

There is a time limit on this assignment (you will see Quercus counting down the time remaining).

1. An experimenter is interested in evaluating the effect of anxiety and muscular tension on a learning task. There are twelve subjects altogether. Each subject is given an anxiety test and is classified as High or Low anxiety, and the six subjects in each anxiety group are randomly assigned to either high or low tension (the subjects have to hold down a button, which is harder or easier to keep held down, while they are learning a task). None of the subjects know which anxiety and tension groups they are in. Each subject has four separate attempts ("trials") to perform the task, all of which are recorded. The trials are numbered in sequential order. The response variable is the number of errors they make on each trial, so a smaller number of errors is better.

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

   (a) What feature of this experiment makes it a repeated-measures design? Explain briefly.

   Solution:

   There is more than one observation from each subject. Specifically, each subject has four trials and a number of errors on each. This means that the observations are not all independent (the ones for each subject will be dependent; for example some subjects may tend to score high or low regardless of their treatment).

   (b) Read in and display (some of) the data.

   Solution:

   I gave you a `.csv` again, so:

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

```
##
## -- Column specification ------------------------------------------------------------
## cols(
##   anxiety = col_character(),
##   tension = col_character(),
##   subject = col_double(),
##   trial1 = col_double(),
##   trial2 = col_double(),
##   trial3 = col_double(),
##   trial4 = col_double()
## )
```

```
learning
```

```
## # A tibble: 12 x 7
##    anxiety tension subject trial1 trial2 trial3 trial4
##    <chr>   <chr>     <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
##  1 low_a   low_t         1     18     14     12      6
##  2 low_a   low_t         2     19     12      8      4
##  3 low_a   low_t         3     14     10      6      2
##  4 low_a   high_t        4     16     12     10      4
##  5 low_a   high_t        5     12      8      6      2
##  6 low_a   high_t        6     18     10      5      1
##  7 high_a  low_t         7     16     10      8      4
##  8 high_a  low_t         8     18      8      4      1
##  9 high_a  low_t         9     16     12      6      2
## 10 high_a  high_t       10     19     16     10      8
## 11 high_a  high_t       11     16     14     10      9
## 12 high_a  high_t       12     16     12      8      8
```

There are 12 subjects, and four trials for each one. (For me, all the data displayed; you might see only the first ten rows.)

(c) Run a suitable repeated-measures ANOVA and display the results.

Solution:

Steps:

1. make a response variable out of the four columns of errors (in `trial1` through `trial4`)
2. run `lm` with that response on the between-subject factors (two of them, here)
3. Get hold of the different "times" (here, the trials)
4. Make a dataframe out of them
5. Run the right `Manova`.

Step one:

```
y <- with(learning, cbind(trial1, trial2, trial3, trial4))
y
```

```
##       trial1 trial2 trial3 trial4
##  [1,]     18     14     12      6
##  [2,]     19     12      8      4
##  [3,]     14     10      6      2
##  [4,]     16     12     10      4
##  [5,]     12      8      6      2
##  [6,]     18     10      5      1
##  [7,]     16     10      8      4
##  [8,]     18      8      4      1
##  [9,]     16     12      6      2
## [10,]     19     16     10      8
## [11,]     16     14     10      9
## [12,]     16     12      8      8
```

I displayed mine to make sure it looked right. Up to you whether you do or not. (It's only 12 rows, so there's no problem displaying it.) Here is another way:

```
learning %>% select(starts_with("trial")) %>%
  as.matrix()
```

```
##       trial1 trial2 trial3 trial4
##  [1,]     18     14     12      6
##  [2,]     19     12      8      4
##  [3,]     14     10      6      2
##  [4,]     16     12     10      4
##  [5,]     12      8      6      2
##  [6,]     18     10      5      1
##  [7,]     16     10      8      4
##  [8,]     18      8      4      1
##  [9,]     16     12      6      2
## [10,]     19     16     10      8
## [11,]     16     14     10      9
## [12,]     16     12      8      8
```

This starts off easier, but needs more thinking to finish it off. `Manova` needs a `matrix` as its response, and the output of select is a dataframe (`tibble`), so you have to take its values and turn them into a matrix. If you go this way, you'll need to save the result into `y` or `response` or whatever with `->`.

Step two:

There are (this time) *two* between-subjects factors `anxiety` and `tension`. You can tell these are between subjects because each subject has only one level of each of them (unlike `trial`, that each subject has four of). These go on the right of the `lm`, *including the interaction* (since the dependence might be on the anxiety-tension *combination*), with the response variable you just made on the left. There's no particular value in looking at the results:

```
learning.1 <- lm(y ~ anxiety*tension, data=learning)
```

Steps three and four:

```
times <- colnames(y)
times.df <- data.frame(times = factor(times))
```

Step five:

Set up all the right inputs for `Manova` (from package `car`). I always use the same names for these, which makes the coding easier, though I really should have called these ones `trials` (since that's what they are):

```
Manova(learning.1, idata = times.df, idesign = ~times)

##
## Type II Repeated Measures MANOVA Tests: Pillai test statistic
##                        Df test stat approx F num Df den Df    Pr(>F)
## (Intercept)             1   0.98310   465.45      1      8 2.244e-08 ***
## anxiety                 1   0.10891     0.98      1      8   0.35171
## tension                 1   0.09174     0.81      1      8   0.39494
## anxiety:tension         1   0.49257     7.77      1      8   0.02368 *
## times                   1   0.98458   127.69      3      6 7.977e-06 ***
## anxiety:times           1   0.75558     6.18      3      6   0.02885 *
## tension:times           1   0.63940     3.55      3      6   0.08751 .
## anxiety:tension:times   1   0.67209     4.10      3      6   0.06692 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(d) Interpret any significant effects in the context of the data. You may ignore the intercept, and in these models it is reasonable to interpret main effects even when a higher-order interaction involving time is significant.

Solution:

My last point is that in these models, you cannot "remove time", so we have to work around this.

There are three significant effects of interest:

- the interaction between anxiety and tension
- the time (trials) main effect
- the interaction between anxiety and time (trials)

These say:

- the effect of tension on number of errors is different for high and low anxiety. You could also say it the other way around.
- the number of errors depends on time, that is, the number of trial you are looking at.
- the effect of anxiety depends on time (trial number).

We cannot yet say any more about what kind of effects these are; this takes a spaghetti plot (which we will look at shortly).

(e) Rearrange your data frame to be a suitable format from which to make a spaghetti plot.

Solution:

This means, as usual for `ggplot`, one observation per row, so we have to make those four `trial` columns longer:

```
learning %>%
  pivot_longer(starts_with("trial"),
               names_to = "trial",
               values_to = "errors") -> learning_long
learning_long
```

```
## # A tibble: 48 x 5
##    anxiety tension subject trial  errors
##    <chr>   <chr>     <dbl> <chr>   <dbl>
##  1 low_a   low_t         1 trial1     18
##  2 low_a   low_t         1 trial2     14
##  3 low_a   low_t         1 trial3     12
##  4 low_a   low_t         1 trial4      6
##  5 low_a   low_t         2 trial1     19
##  6 low_a   low_t         2 trial2     12
##  7 low_a   low_t         2 trial3      8
##  8 low_a   low_t         2 trial4      4
##  9 low_a   low_t         3 trial1     14
## 10 low_a   low_t         3 trial2     10
## # ... with 38 more rows
```

Any other way of specifying the columns to make longer also works. This is what I thought was easiest.

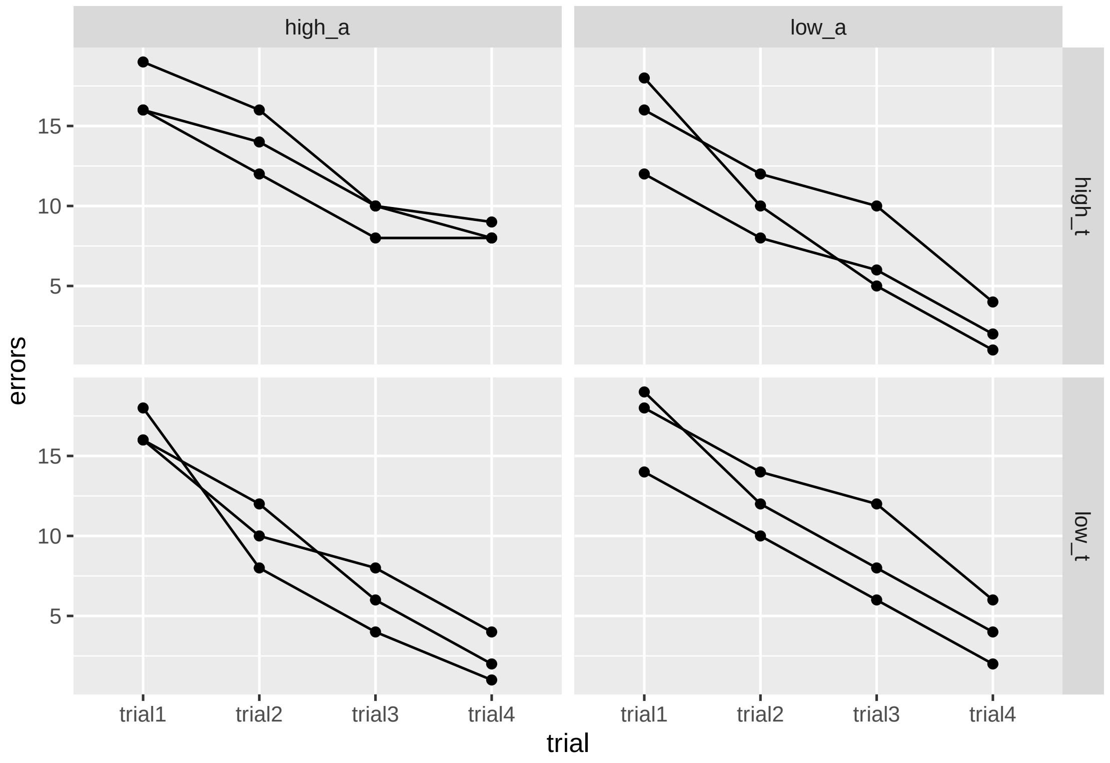
(f) Make a suitable spaghetti plot for these data. There are several possible approaches; if you use one that is not in the course materials, tell the grader where you got it from.

Solution:

The first thing to remember is that `ggplot` likes long-format data, which is why I had you create that just now.

In this plot, we have *two* between-subject factors. This is like the exercise example from lecture, and you can approach it the same way with facets:

```
ggplot(learning_long, aes(x=trial, y=errors, group=subject)) +
  geom_point() + geom_line() +
  facet_grid(tension ~ anxiety)
```

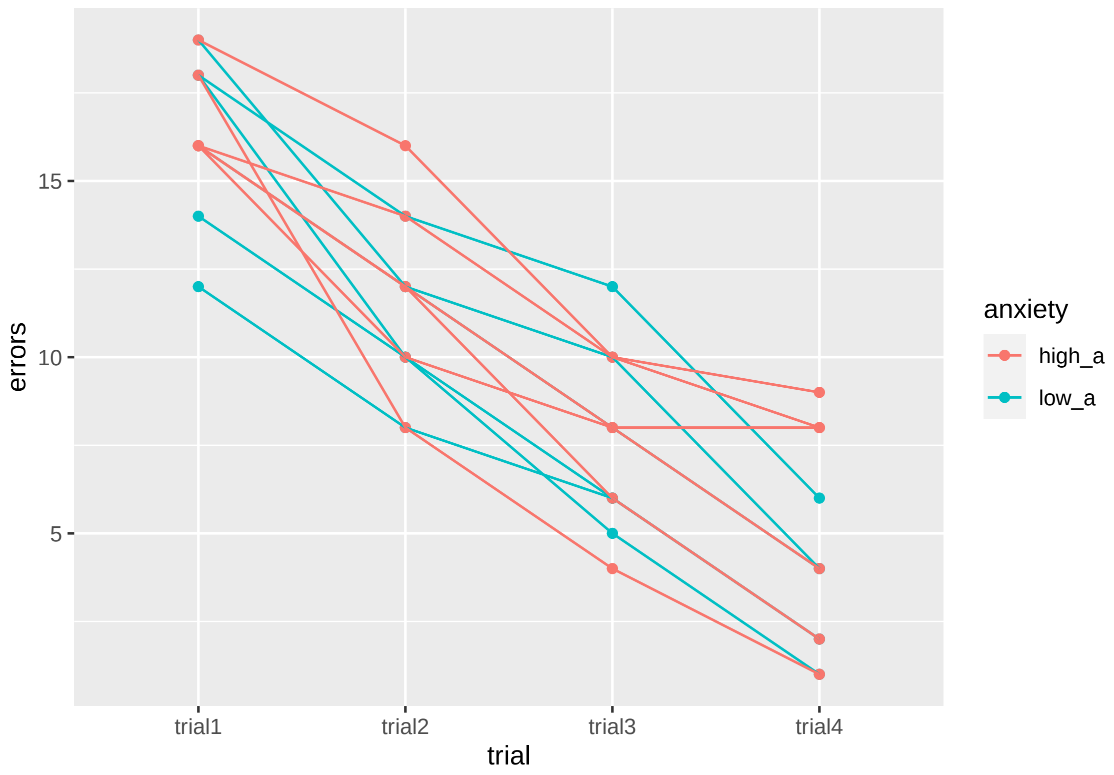


The way I drew mine, tension is up the side, and anxiety is across the top. I gave the high and low levels of tension and anxiety different labels so that, on a plot like this, you wouldn't get confused about which high and low is which.

This is a good way to graph these. Here is another.

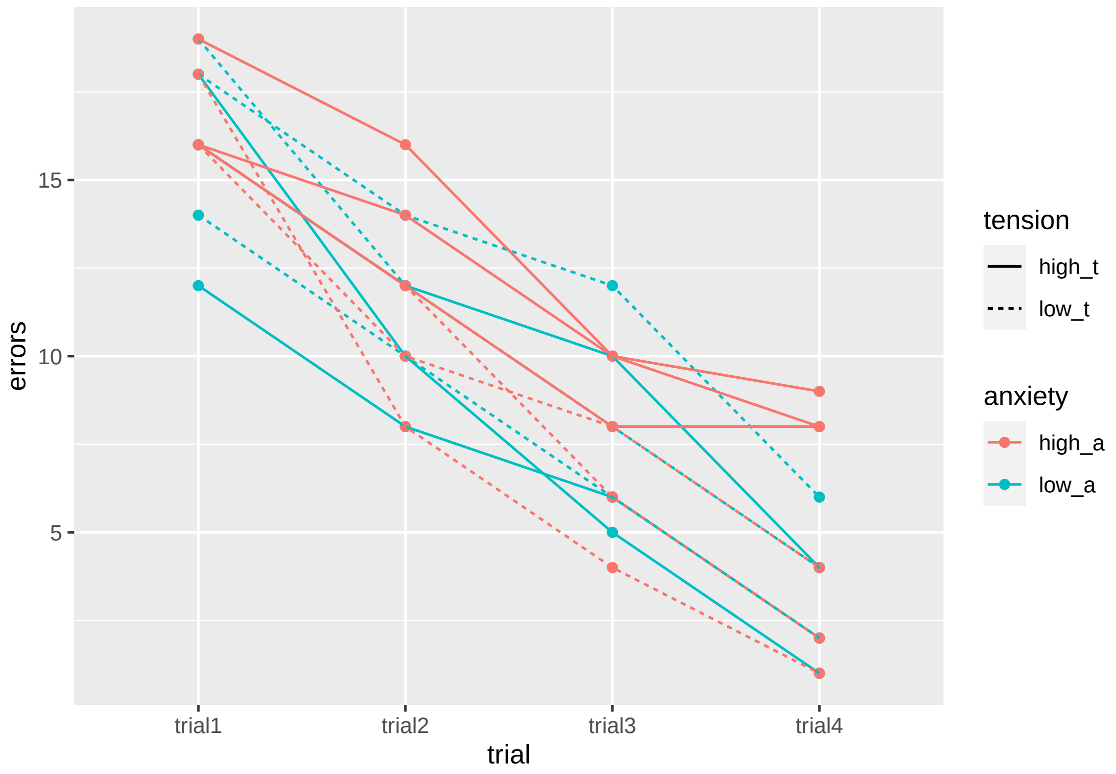
The way we saw how to draw originally, with only one between-subject factor, would look something like this for this one:

```
ggplot(learning_long, aes(x=trial, y=errors, colour=anxiety, group=subject)) +
  geom_point() + geom_line()
```



but this makes no mention of `tension`. One way to do that is to use different kinds of line joining the points, solid or dotted or dashed or whatever, depending on whether they are high or low tension. This is where you may need to fire up Google and see what comes out. I searched for “ggplot line style”. This was my first hit, and by scrolling down a bit I found `linetype`. The one there is not quite the same as what we want, since we want the line type to depend on a variable’s value, but this is exactly how `aes` works, so put it in there:

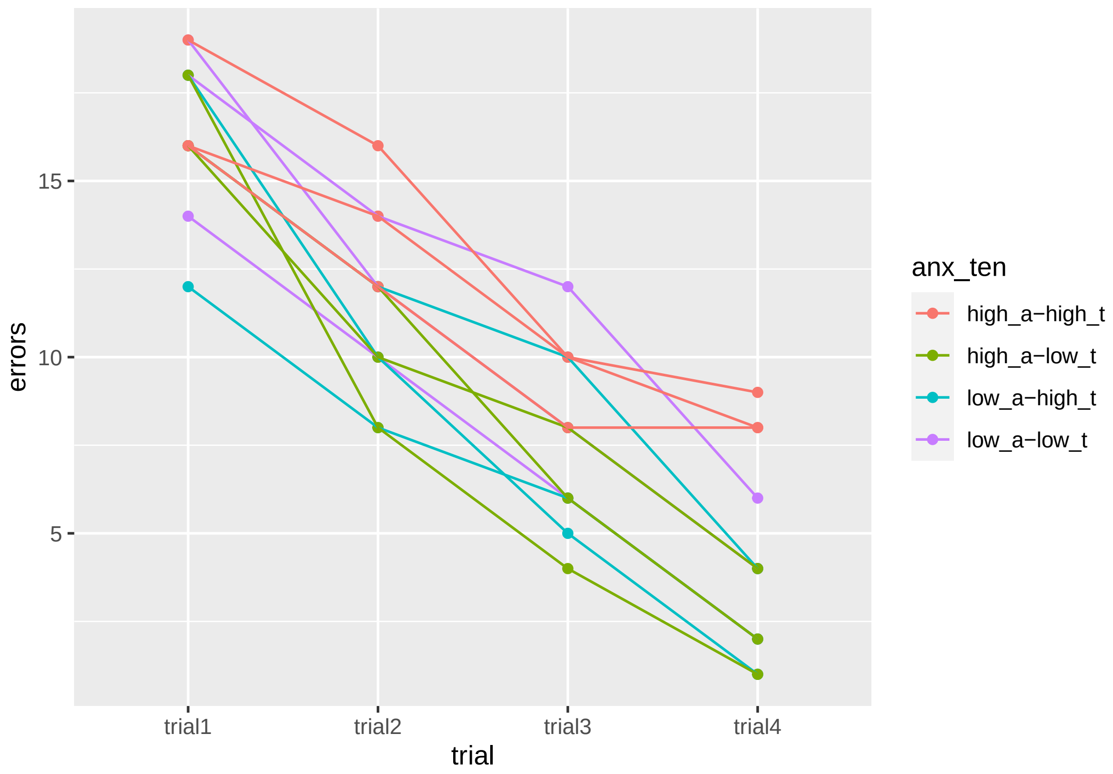
```
ggplot(learning_long, aes(x=trial, y=errors, colour=anxiety, linetype=tension, group=subject))
  geom_point() + geom_line()
```



The lines have different colours and types according to the subject's values of `anxiety` and `tension`, as we want, so this works.

Another way you might come up with is to make an anxiety-tension *combination*, and use that for colour:

```
learning_long %>%
  mutate(anx_ten = str_c(anxiety, "-", tension)) %>%
  ggplot(aes(x=trial, y=errors, colour=anx_ten, group=subject)) +
  geom_point() + geom_line()
```

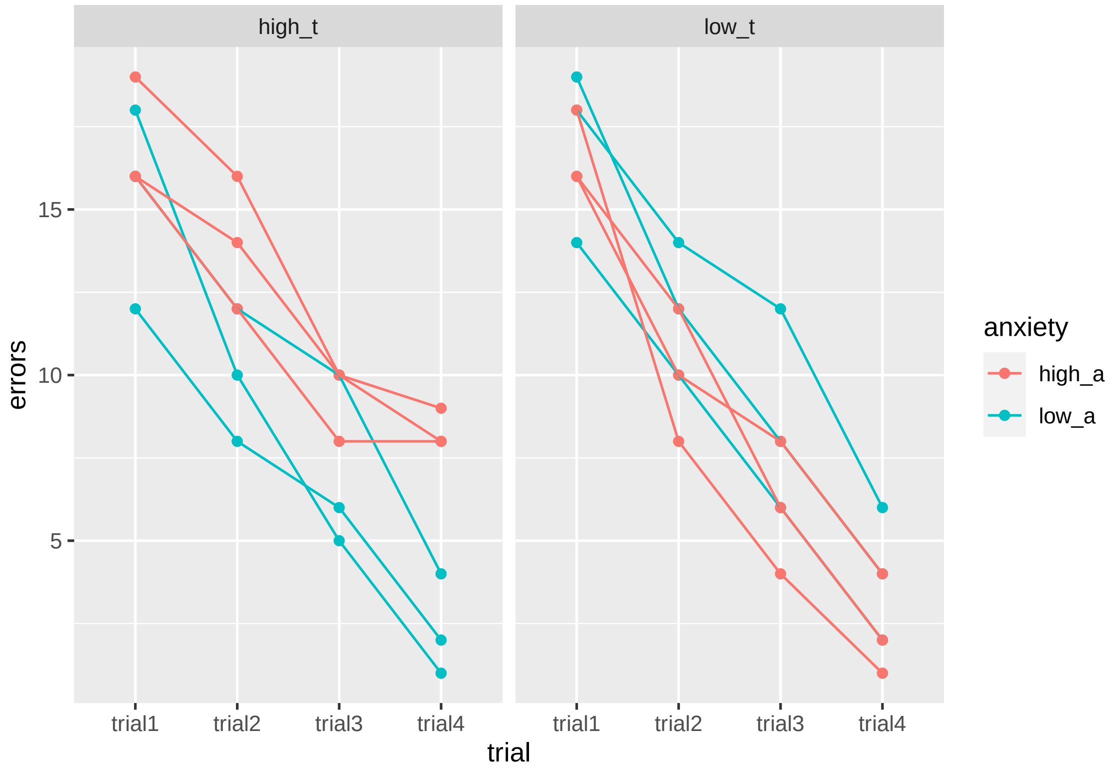


This works, especially for what we'll be doing with it, but if you were looking for an anxiety main effect, say, you'd have to remember to compare red with blue and green with purple, to hold `tension` constant. (This is not quite so easy to think about as comparing the red and blue solid lines and then the red and blue dashed lines on the other graph.)

If you consult Google for ideas here, you must *cite* any webpages that you use. Good scholarship requires this, and you're a good scholar, aren't you?

You don't have to facet on both explanatory variables. You can make one of them colours and use `facet_wrap` for the other one, eg:

```
ggplot(learning_long, aes(x=trial, y=errors, colour=anxiety, group=subject)) +
  geom_point() + geom_line() +
  facet_wrap(~tension)
```



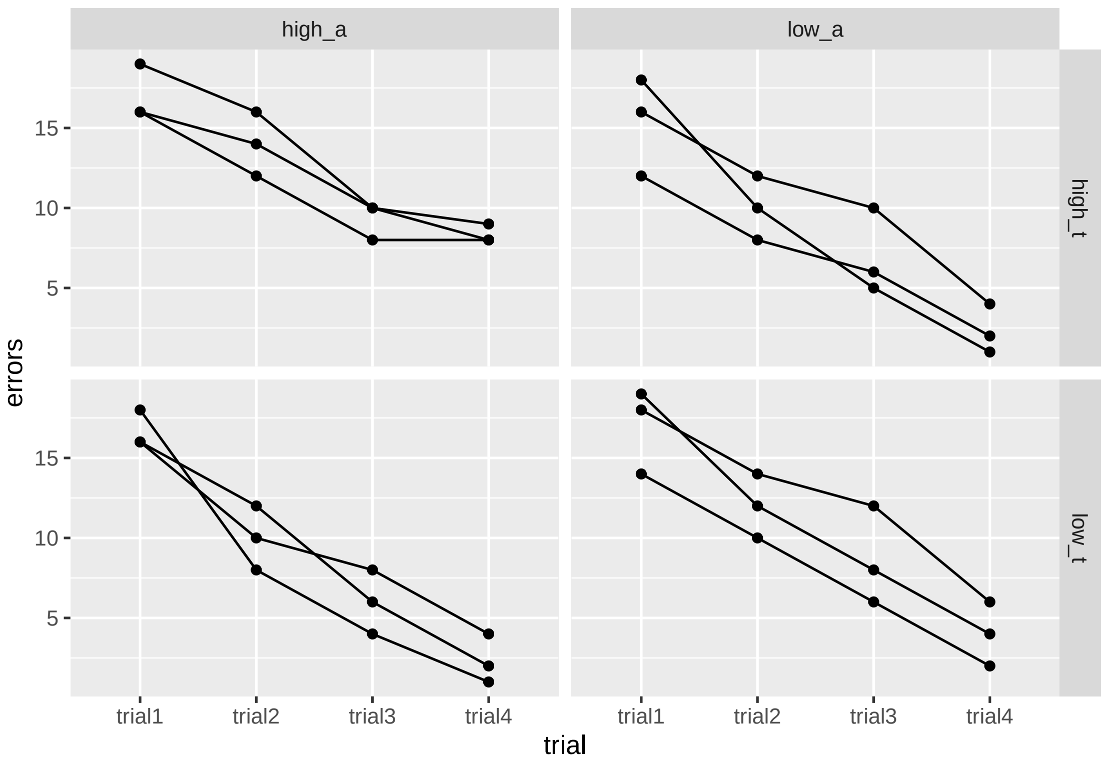
For making comparisons (which we do next), it's not clear to me whether it's better to have all the traces on the same graph, or to have separate facets with fewer traces on each one. This is mostly a matter of taste.

(g) What does your spaghetti plot tell you about the nature of the significant effects you found from your repeated-measures analysis earlier? Explain briefly.

Solution:

I'm going to reproduce one of my graphs so that I'm not scrolling back. You will probably have only one graph, which will be the last thing in the previous part, so you won't need to do this:

```r
ggplot(learning_long, aes(x=trial, y=errors, group=subject)) +
  geom_point() + geom_line() +
  facet_grid(tension ~ anxiety)
```



We found three significant effects before (or talk about whatever you found, if you found something different).

The most obvious effect is that of time (trial), which had a very small P-value. This shows up in all those traces on your spaghetti plot(s) going downhill, meaning that people were making fewer errors over time, regardless of which treatment they were on. This is even clear from looking at the data (you might have noticed when you read it in as something "obvious").

There was an anxiety-tension interaction, which says that the effect of tension depends on the level of anxiety. If the anxiety is high (on the left side on my graph), the high-tension traces (top) seem to come down less fast than the low-tension ones (bottom). If the anxiety is low, however (right), there seems to be not much difference between high and low tension: the top and bottom traces look very similar. That is to say, whether or not there is a tension effect depends on whether anxiety is high or low. This didn't have so small a P-value, so it's harder to see.

Finally, there was an anxiety-time interaction. This means that the difference between high and low anxiety changes over time, regardless of the level of tension. On mine, look at what is happening over time in the two graphs on the top (high tension) and on the bottom two (low tension). I think the evidence for this is coming from the top left vs. top right graph, where the error counts for low anxiety keep decreasing over time (right), but for high anxiety they stop decreasing after trial 3 (the results for trial 4 are about the same). This, likewise, is not strongly significant, so it is not so easy to see.

If your graph is one of the others, your reasoning might look different, but I think your conclusions will end up in about the same place.

(h) Run a suitable mixed model using `lmer`, displaying the output. Make a brief comparison of the results with those of the repeated measures analysis. (You do not need to interpret the results of this model.)

Solution:

This uses the long dataframe. You include (at least for now) all the interactions between the three factors, treating time (which I properly called `trial` here) no differently from anything else. Then you add a subject random effect as shown, which is how this model allows for some of the numbers of errors being correlated because they come from the same person:

```
learning.2 <- lmer(errors~anxiety*tension*trial+(1|subject), data=learning_long)
drop1(learning.2, test="Chisq")
```

```
## Single term deletions
##
## Model:
## errors ~ anxiety * tension * trial + (1 | subject)
##                       npar    AIC    LRT Pr(Chi)
## <none>                     208.71
## anxiety:tension:trial    3 210.58 7.8718 0.04874 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In this case, we need to keep the three-way interaction. Before, its P-value was 0.067, so there is actually not much difference in P-value even though the conclusions are opposite. This one says that the effect of the anxiety-tension combination itself changes over time.

Notice that the random effect for subject is not up for grabs here; we are taking it for granted that the subjects *do* differ one from another, and so the random effect has to stay in the model.

If you look at the `summary` output:

```
summary(learning.2)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: errors ~ anxiety * tension * trial + (1 | subject)
##    Data: learning_long
##
## REML criterion at convergence: 145.7
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.36874 -0.54342 -0.09531  0.40758  1.89797
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  subject  (Intercept) 2.035    1.426
##  Residual             2.174    1.474
## Number of obs: 48, groups:  subject, 12
##
## Fixed effects:
##                                         Estimate Std. Error t value
## (Intercept)                              17.0000     1.1844  14.353
## anxietylow_a                             -1.6667     1.6750  -0.995
## tensionlow_t                             -0.3333     1.6750  -0.199
## trialtrial2                              -3.0000     1.2038  -2.492
## trialtrial3                              -7.6667     1.2038  -6.369
## trialtrial4                              -8.6667     1.2038  -7.200
## anxietylow_a:tensionlow_t                 2.0000     2.3688   0.844
## anxietylow_a:trialtrial2                 -2.3333     1.7024  -1.371
## anxietylow_a:trialtrial3                 -0.6667     1.7024  -0.392
## anxietylow_a:trialtrial4                 -4.3333     1.7024  -2.545
## tensionlow_t:trialtrial2                 -3.6667     1.7024  -2.154
## tensionlow_t:trialtrial3                 -3.0000     1.7024  -1.762
## tensionlow_t:trialtrial4                 -5.6667     1.7024  -3.329
## anxietylow_a:tensionlow_t:trialtrial2   4.0000     2.4075   1.661
## anxietylow_a:tensionlow_t:trialtrial3   3.0000     2.4075   1.246
## anxietylow_a:tensionlow_t:trialtrial4   5.6667     2.4075   2.354
```

```
##
## Correlation matrix not shown by default, as p = 16 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

you'll see that there are two random effects ("sources of variation", if you will): the subjects and "residual". There is estimation here (the "REML" thing at the top), but there aro no tests for the random effects. If there is a lot of variability due to subjects (the `Variance` in the subjects line is large), that means that the subjects were very different from each other, and *not* including a subject effect could make the numbers of errors look very variable. If there is very little variability due to subjects, then the subjects are almost the same, and it makes little difference whether we include the subjects or not. In this case, the variance due to subjects is about as big as the variance due to error, so it looks to me as if having a subject random effect was worthwhile.

To interpret a 3-way interaction, you'd have to go back to your spaghetti plot and try to interpret how the effects of the anxiety-tension combination differ over time. I suspect that this goes back to the errors for the high-tension and high-anxiety combination coming down until trial 3 and then stopping, while the errors for the other combinations keep decreasing. I didn't want you to get into that, however.

Extra: I wondered what would happen if we removed the three-way interaction (pretending we were working at $\alpha = 0.01$, say). This goes most smoothly using `update`:

```r
learning.3 <- update(learning.2, .~.-anxiety:tension:trial)
drop1(learning.3, test = "Chisq")
```

```
## Single term deletions
##
## Model:
## errors ~ anxiety + tension + trial + (1 | subject) + anxiety:tension +
##     anxiety:trial + tension:trial
##                 npar    AIC    LRT  Pr(Chi)
## <none>                210.58
## anxiety:tension    1 216.72 8.1407 0.004328 **
## anxiety:trial      3 208.97 4.3888 0.222427
## tension:trial      3 210.76 6.1842 0.102985
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can use `drop1` here the same as with any other model, to see what we can get rid of. (`step` appears not to work here, but we are about to mimic it.)

First, take out `anxiety:trial`:

```
learning.4 <- update(learning.3, .~.-anxiety:trial)
drop1(learning.4, test = "Chisq")
```

```
## Single term deletions
##
## Model:
## errors ~ anxiety + tension + trial + (1 | subject) + anxiety:tension +
##     tension:trial
##                  npar    AIC    LRT  Pr(Chi)
## <none>                208.97
## anxiety:tension    1 215.11 8.1407 0.004328 **
## tension:trial      3 208.49 5.5260 0.137089
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

`tension:trial` can come out as well:

```
learning.5 <- update(learning.4, .~.-tension:trial)
drop1(learning.5, test = "Chisq")
```

```
## boundary (singular) fit: see ?isSingular
```

```
## Single term deletions
##
## Model:
## errors ~ anxiety + tension + trial + (1 | subject) + anxiety:tension
##                  npar    AIC    LRT   Pr(Chi)
## <none>                208.49
## trial              3 301.08 98.583 < 2.2e-16 ***
## anxiety:tension    1 214.63  8.141  0.004328 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

and this is where we stop. This is like the model we got the `Manova` way, ignoring the things that were not significant, which we had to do there. It is however not the same. In this kind of model, though, we can act as we normally do, actually removing things that are not significant, re-fitting and re-evaluating.