# STAD29 / STA 1007 assignment 5

Due Tuesday Feb 25 at 11:59pm on Quercus

Packages for this one:

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Hand in questions 2 and 3 of the following:

1. Work through, or read through, chapter 22 of PASIAS. Problem 22.2 will prepare you for question 2, and problems 22.3, 22.5, and 22.6 will prepare you for question 3.

2. The number of hours that a battery operates might depend on the material it is made of, and the temperature at which it is operated. In an experiment, three materials (labelled A, B, and C) were tested, and three temperatures: Low (-10 ° C), Medium (20 ° C) or High (45 ° C). Twelve batteries were randomly selected from each material type and were then randomly allocated to each temperature level. The resulting life of all 36 batteries is shown in `http://ritsokiguess.site/STAD29/batteries.txt`, with the data values separated by spaces.

   (a) (2 marks) Read in and display (some of) the data.

   **Solution:** Separated by spaces means `read_delim`:

```
my_url="http://ritsokiguess.site/STAD29/batteries.txt"
batteries <- read_delim(my_url, " ")
```

```
## Parsed with column specification:
## cols(
##   material = col_character(),
##   temperature = col_character(),
##   life_hours = col_double()
## )
```

```
batteries
```

```
## # A tibble: 36 x 3
##    material temperature life_hours
##    <chr>    <chr>            <dbl>
##  1 A        low                130
##  2 A        low                155
##  3 A        low                 74
##  4 A        low                180
```

```
##  5 A        medium              34
##  6 A        medium              40
##  7 A        medium              80
##  8 A        medium              75
##  9 A        high                20
## 10 A        high                70
## # ... with 26 more rows
```

Extra: more data-reading fun. I got the data from `http://www.statstutor.ac.uk/resources/uploaded/coventrytwowayanova.pdf`. My first step was to reproduce it in the form you see there:

```
data <- tribble(
  ~material, ~low, ~medium, ~high,
  "A", "130, 155, 74, 180", "34, 40, 80, 75", "20, 70, 82, 58",
  "B", "150, 188, 159, 126", "136, 122, 106, 115", "25, 70, 58, 45",
  "C", "138, 110, 168, 180", "174, 120, 150, 139", "96, 104, 82, 60"
)
data
```

```
## # A tibble: 3 x 4
##   material low               medium             high
##   <chr>    <chr>             <chr>              <chr>
## 1 A        130, 155, 74, 180  34, 40, 80, 75     20, 70, 82, 58
## 2 B        150, 188, 159, 126 136, 122, 106, 115 25, 70, 58, 45
## 3 C        138, 110, 168, 180 174, 120, 150, 139 96, 104, 82, 60
```

I figured this would minimize transcription errors, and then I could deal with it in the `tidyverse`.

Next, get those quartets of lifetimes in hours into one column:

```
data %>% pivot_longer(-material, names_to="temperature",
                      values_to="life_hours")
```

```
## # A tibble: 9 x 3
##   material temperature life_hours
##   <chr>    <chr>       <chr>
## 1 A        low         130, 155, 74, 180
## 2 A        medium      34, 40, 80, 75
## 3 A        high        20, 70, 82, 58
## 4 B        low         150, 188, 159, 126
## 5 B        medium      136, 122, 106, 115
## 6 B        high        25, 70, 58, 45
## 7 C        low         138, 110, 168, 180
## 8 C        medium      174, 120, 150, 139
## 9 C        high        96, 104, 82, 60
```

Now, those are four lifetimes, separated by commas, in each cell. There is a handy technique for dealing with those, that looks like this:

```
data %>% pivot_longer(-material, names_to="temperature", values_to="life_hours") %>%
  separate_rows(life_hours, sep=",", convert=T)
```

```
## # A tibble: 36 x 3
##    material temperature life_hours
##    <chr>    <chr>            <int>
##  1 A        low                130
##  2 A        low                155
##  3 A        low                 74
##  4 A        low                180
```

```
##  5 A        medium              34
##  6 A        medium              40
##  7 A        medium              80
##  8 A        medium              75
##  9 A        high                20
## 10 A        high                70
## # ... with 26 more rows
```

The last `convert` is to take the numbers, which were previously text, and convert them to what they look like (numbers) after splitting them up.

This is what I saved for you.

(b) (3 marks) Which order are the temperatures in, as far as R is concerned? Does that make sense? If not, put them in the right order by creating a column in the data frame that has the right order. (Hint: what order are they in in the data frame?)

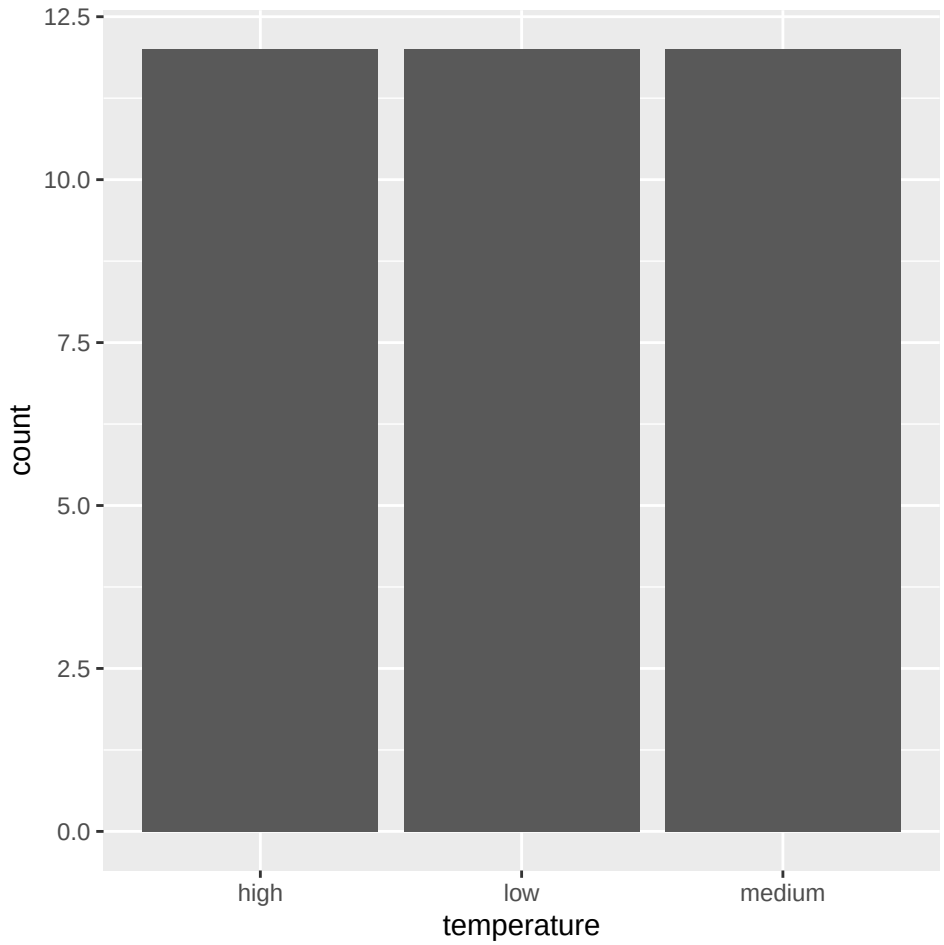
**Solution:** You might guess that R thinks they are in alphabetical order, high, low, medium, which is indeed correct:

```
batteries %>% distinct(temperature)
```

```
## # A tibble: 3 x 1
##   temperature
##   <chr>
## 1 low
## 2 medium
## 3 high
```

or even

```
ggplot(batteries, aes(x=temperature)) + geom_bar()
```




This does not make any logical sense.

To fix that, use `fct_inorder`, since the temperatures are in a sensible order in the data. I'm overwriting my old temperature column, but you can certainly create a new one, *which you need to use for the rest of the question*:[1]

```
batteries %>% mutate(temperature=fct_inorder(temperature)) -> batteries
```

This is now good:

```
batteries %>% distinct(temperature)
```

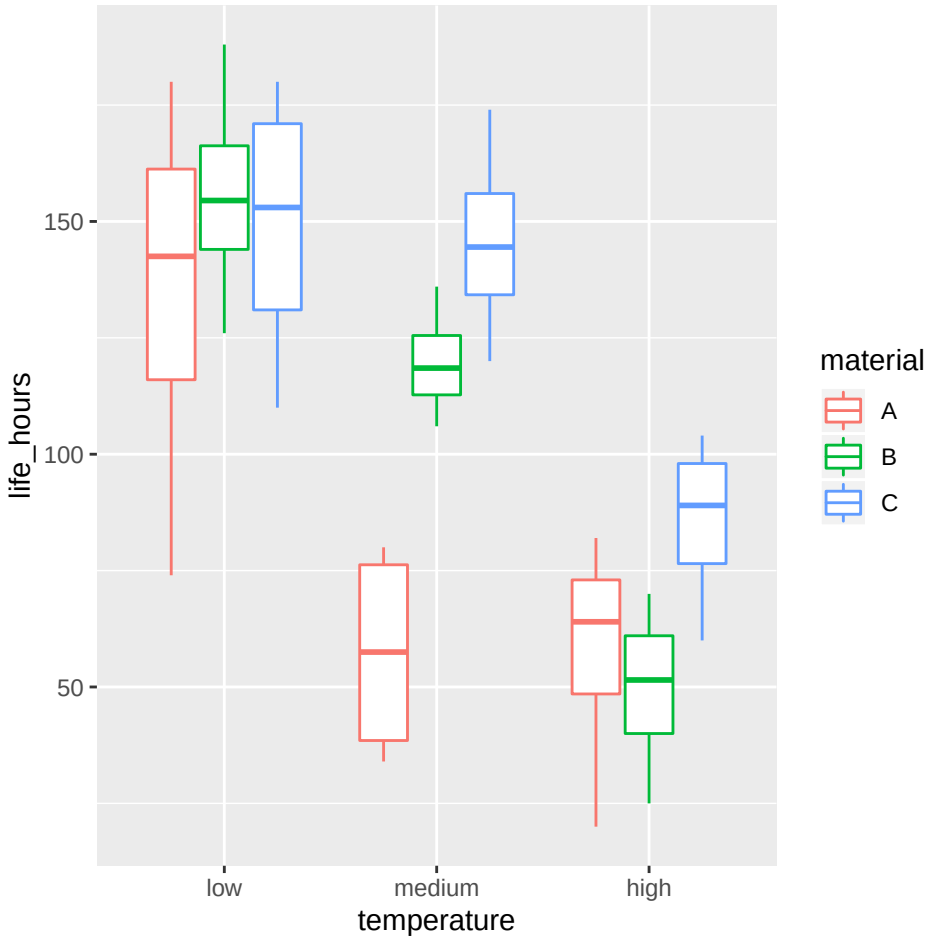
```
## # A tibble: 3 x 1
##   temperature
##   <fct>
## 1 low
## 2 medium
## 3 high
```

(c) (2 marks) Make a suitable plot of these data, given the number and types of variables you have. Put temperature on the $x$-axis.

**Solution:** Go back to C32: one quantitative and two categorical variables, so you want grouped boxplots:

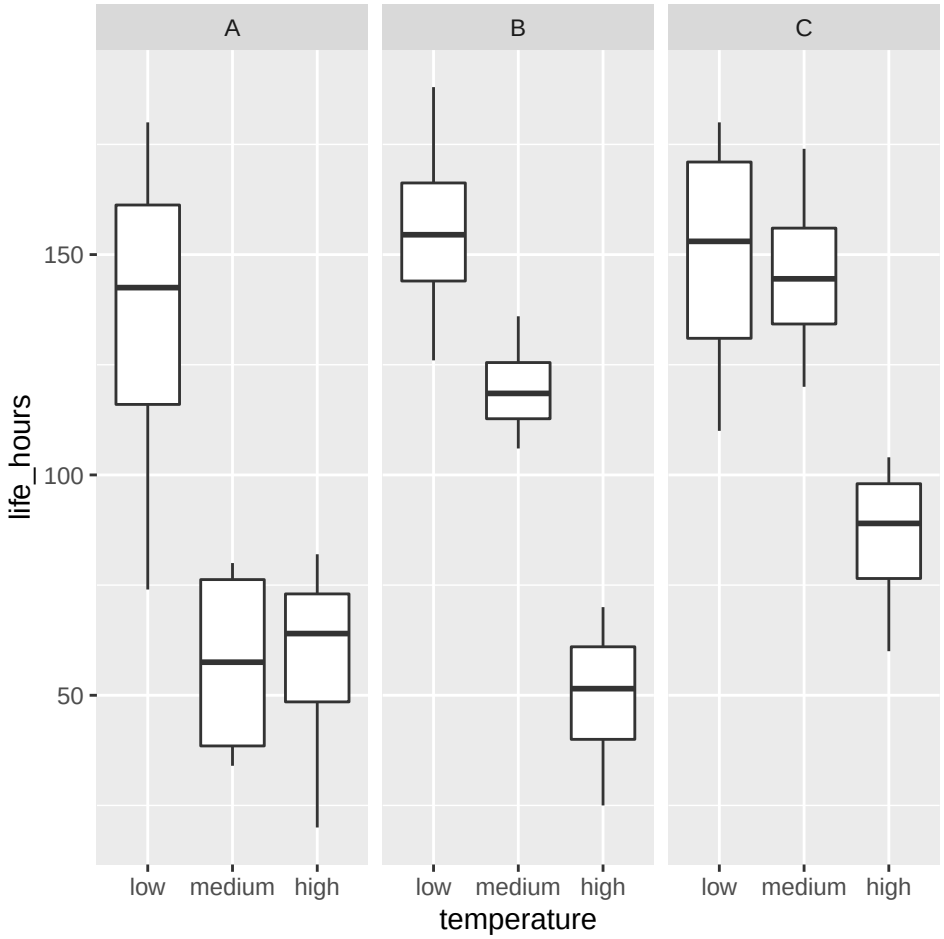
```
batteries %>% ggplot(aes(x=temperature, y=life_hours, colour=material)) +
```

```
    geom_boxplot()
```




Facetted boxplots also work, but less colourfully:

```
batteries %>% ggplot(aes(x=temperature, y=life_hours)) +
    geom_boxplot() + facet_wrap(~material)
```

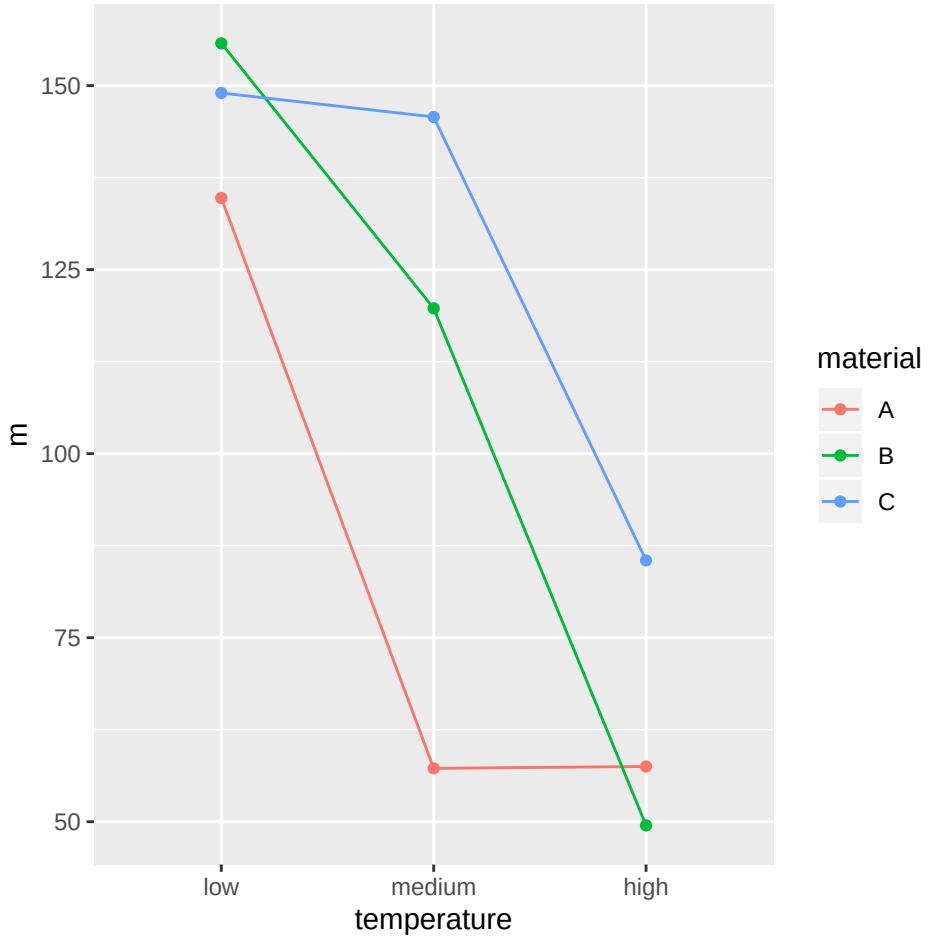



This looks different, because you now have three mini-$x$-axes, one for each material, so the order of the boxes is different.

(d) (3 marks) Make an interaction plot, again putting temperature on the $x$-axis.

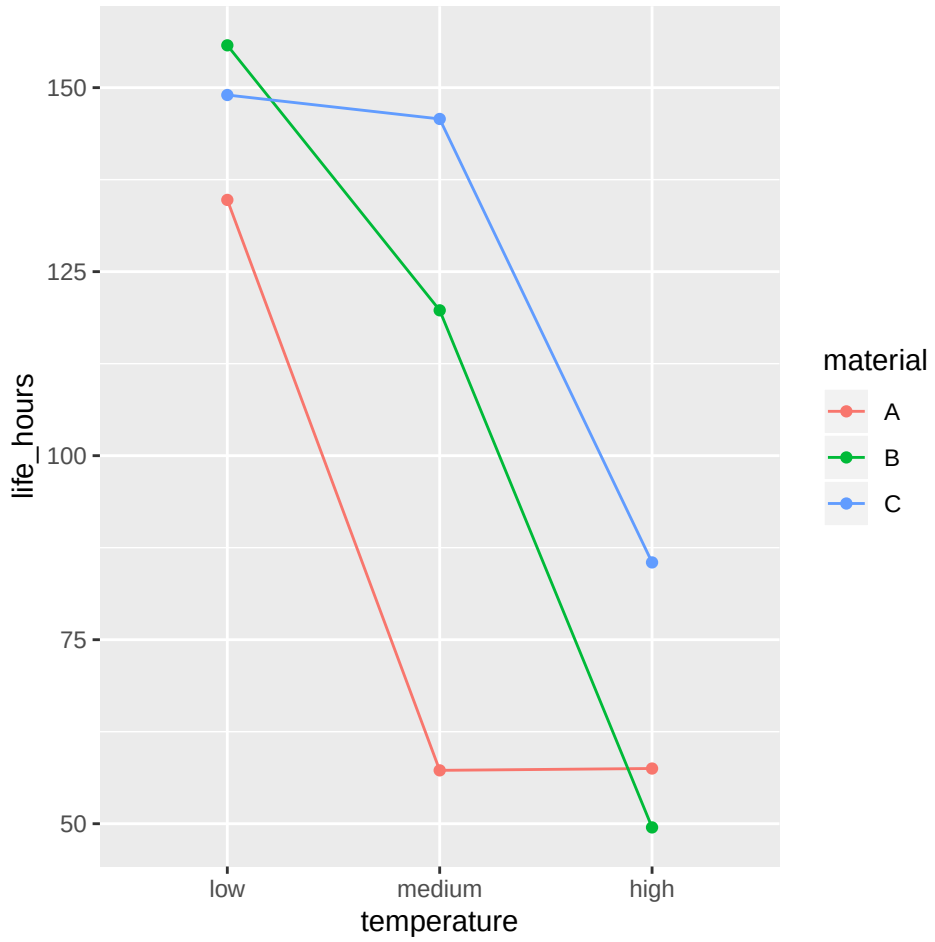
**Solution:** This means using material as the "trace", which you do by setting it as `colour` *and* `group`. Remember that you need the mean `life_hours` first for each combination of material and temperature (in either order). This is how I did it:

```
batteries %>% group_by(material, temperature) %>%
  summarize(m=mean(life_hours)) %>%
  ggplot(aes(x=temperature, y=m, colour=material, group=material)) +
    geom_point() + geom_line()
```




There's no problem in doing the group-by and summarize *first*, saving the result, and then plotting *that*. Alternatively, you can do it the clever way that avoids actually finding the means yourself:

```
ggplot(batteries, aes(x=temperature, y=life_hours,
                      colour=material, group=material)) +
    stat_summary(fun.y=mean, geom="point") +
    stat_summary(fun.y=mean, geom="line")
```




Either way, including the points as well as the lines is optional.

(e) (2 marks) What do you conclude from your interaction plot? By looking at your first plot, explain briefly why your conclusion from your interaction plot makes sense.

**Solution:** The lines on the interaction plot do not look close to parallel, so we would expect to see a significant interaction. (One point.) As to why that is, my grouped boxplots say that the distinction between materials is not consistent over all temperatures; sometimes the three materials have about the same lifetime (low temperature) but sometimes they have very different lifetimes (medium temperature). Or, flip it around and say that material C is best at medium and high temperatures, but not at low.

To get the second point, you need to say something about how the effect of one explanatory variable on battery life depends on the level of the other one. This is what would be behind a significant interaction.

(f) (3 marks) Run a suitable analysis of variance, including interaction, and display the results. Was your interaction significant?

**Solution:** This is `aov` but with the right kind of model formula:

```
batteries.1 <- aov(life_hours~temperature*material, data=batteries)
```

```
summary(batteries.1)
##                      Df Sum Sq Mean Sq F value  Pr(>F)
## temperature           2  40713   20357  28.670 2.1e-07 ***
## material              2  11488    5744   8.090 0.00177 **
## temperature:material  4   8843    2211   3.113 0.03143 *
## Residuals            27  19171     710
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

It's actually OK to use either the original temperature or your ordered one, since the analysis will come out the same either way.

My P-value for interaction is less than 0.05, so my interaction is significant, as I guessed from my interaction plot.

(g) (2 marks) Explain briefly why simple effects would be a useful technique for this data set, and give an example of a comparison you would be able to make with them.

**Solution:** Simple effects work when you have a significant interaction (a fast one point). The idea is that you look at the effects of one of the explanatory variables *at each level* of the other: for example, you see how the materials compare at low temperature, or (flipping it around) how the temperatures compare for material C. Any example is good, but I want you to say something specific that applies to this data set.[2]

(h) (4 marks) Find the simple effects of material at each temperature. That is, for each temperature, compare the materials at that temperature using `aov` and (if necessary) Tukey, and state your conclusions in the context of the data.

**Solution:** This is long, but rather repetitive. The easiest way is (i) to use `filter` to make a data frame containing only the temperature you want, (ii) run a one-way ANOVA predicting `life_hours` from materia for that temperature, (iii) if needed, run Tukey from that ANOVA.

Thus, low temperature first:

```
batteries %>% filter(temperature=="low") %>%
  aov(life_hours~material, data=.) %>% summary()
##             Df Sum Sq Mean Sq F value Pr(>F)
## material     2    920   459.8   0.373  0.699
## Residuals    9  11103  1233.7
```

If you prefer, make the data frame with `filter`, save it, then run `aov`. Either way works.

At low temperatures, there is no difference in hours of life between the materials.

Medium temperature:

```
batteries %>% filter(temperature=="medium") %>%
  aov(life_hours~material, data=.) -> res
summary(res)
##             Df Sum Sq Mean Sq F value   Pr(>F)
## material     2  16553    8276   20.26 0.000465 ***
## Residuals    9   3676     408
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This time, there *are* differences to find, so we run Tukey to find them:

```
TukeyHSD(res)
##   Tukey multiple comparisons of means
```

```
##     95% family-wise confidence level
##
## Fit: aov(formula = life_hours ~ material, data = .)
##
## $material
##     diff       lwr       upr     p adj
## B-A 62.5  22.59911 102.40089 0.0045670
## C-A 88.5  48.59911 128.40089 0.0004209
## C-B 26.0 -13.90089  65.90089 0.2177840
```

Material A is significantly worse than the other two.

Finally, high temperature:

```
batteries %>% filter(temperature=="high") %>%
  aov(life_hours~material, data=.) %>% summary()
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## material     2   2859  1429.3    2.93  0.105
## Residuals    9   4391   487.9
```

No significant differences between materials here either.

The overall conclusion is that it is only at medium temperature that it matters at all which material is used, and at that temperature, material A is significantly worse than the others.

If you now go back and look at your grouped boxplot, you will see that these conclusions are not at all surprising given what you see there. The only mild surprise, maybe, is that the difference between materials B and C is not significant anywhere.

A faster but much more complicated way coding-wise is this:

```
library(broom)
batteries %>%
  nest(-temperature) %>%
  mutate(anova=map(data, ~aov(life_hours~material, data=.))) %>%
  mutate(p_value=map_dbl(anova, ~glance(.)$p.value)) %>%
  mutate(tukey=map(anova, ~TukeyHSD(.))) %>%
  mutate(tukey_table=map(tukey, ~tidy(.))) %>%
  unnest(tukey_table) %>%
  select(-data, -anova, -tukey)
```

```
## Warning: All elements of `...` must be named.
## Did you want `data = c(material, life_hours)`?
```

```
## # A tibble: 9 x 8
##   temperature  p_value term   comparison estimate conf.low conf.high adj.p.value
##   <fct>          <dbl> <chr>  <chr>         <dbl>    <dbl>     <dbl>       <dbl>
## 1 low         0.699    mater~ B-A           21      -48.3       90.3    0.686
## 2 low         0.699    mater~ C-A           14.2    -55.1       83.6    0.837
## 3 low         0.699    mater~ C-B           -6.75   -76.1       62.6    0.960
## 4 medium      0.000465 mater~ B-A           62.5     22.6      102.     0.00457
## 5 medium      0.000465 mater~ C-A           88.5     48.6      128.     0.000421
## 6 medium      0.000465 mater~ C-B           26      -13.9       65.9    0.218
## 7 high        0.105    mater~ B-A           -8.     -51.6       35.6    0.867
## 8 high        0.105    mater~ C-A           28.0    -15.6       71.6    0.226
## 9 high        0.105    mater~ C-B           36.0     -7.61      79.6    0.106
```

Gosh. This produces, for each temperature, the ANOVA P-value for comparing materials at that temperature, and all three Tukey comparisons for materials at that temperature. The results are the same as before.[3]

Coding steps, starting from the `nest` line, which you should run the code one line at a time to understand:

1. create a new data frame with one row for each temperature and everything else in a list-column called `data` for that temperature.
2. for each of those data frames in `data`, run `aov` predicting `life_hours` from `material`.
3. Pull out the P-values of each ANOVA. This is rather tricky, but the easiest way is to use `glance` from `broom`.
4. Run Tukey on each ANOVA.
5. Use `tidy` from `broom` to get a digestible output from the Tukey[4]
6. Display all the Tukey tables.
7. Get rid of the columns I no longer care about.

3. R has a number of built-in data sets. One of them is called `PlantGrowth`. This consists of 30 observations from an experiment to compare plant yield (measured by the dried weight of plants) under two treatment conditions and a control condition. We have two research hypotheses to consider: whether the average of the two treatments is different from the control, and whether the two treatments differ from one another.

   (a) (1 mark) Display (some of) the data set.

**Solution:** You *definitely* don't want to overthink this one!

```
PlantGrowth
```

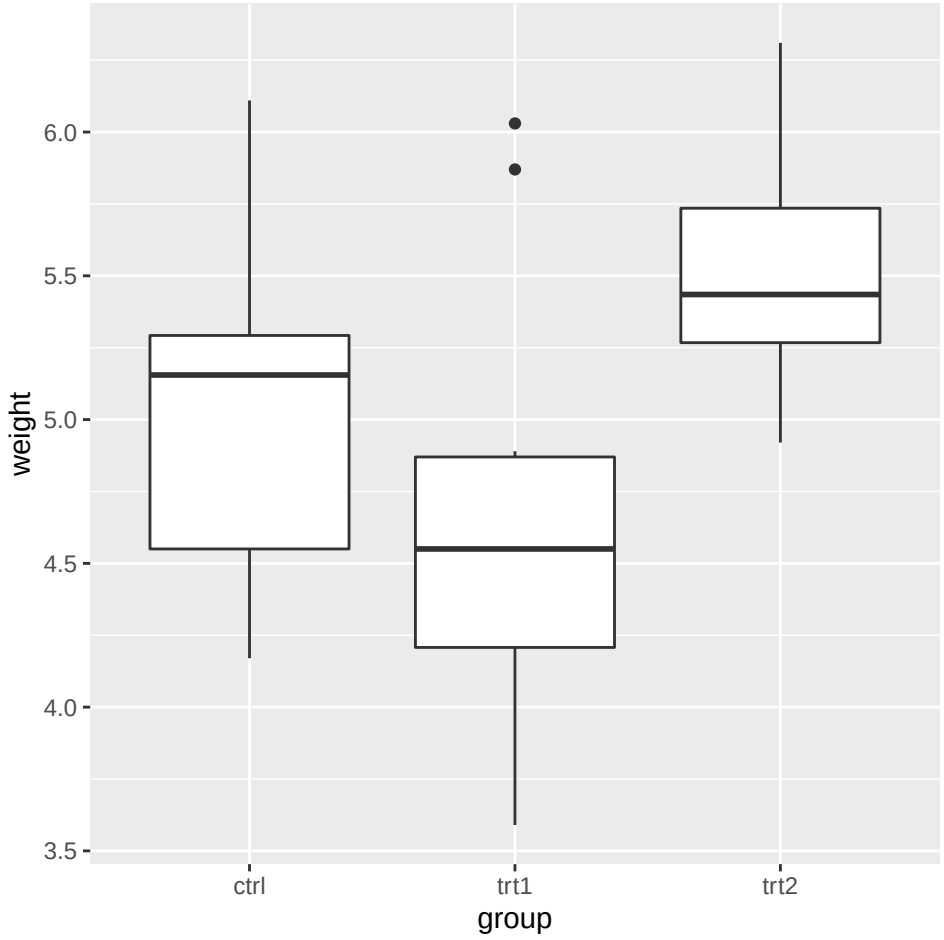
```
##    weight group
## 1    4.17  ctrl
## 2    5.58  ctrl
## 3    5.18  ctrl
## 4    6.11  ctrl
## 5    4.50  ctrl
## 6    4.61  ctrl
## 7    5.17  ctrl
## 8    4.53  ctrl
## 9    5.33  ctrl
## 10   5.14  ctrl
## 11   4.81  trt1
## 12   4.17  trt1
## 13   4.41  trt1
## 14   3.59  trt1
## 15   5.87  trt1
## 16   3.83  trt1
## 17   6.03  trt1
## 18   4.89  trt1
## 19   4.32  trt1
## 20   4.69  trt1
## 21   6.31  trt2
## 22   5.12  trt2
## 23   5.54  trt2
## 24   5.50  trt2
```

```
## 25   5.37  trt2
## 26   5.29  trt2
## 27   4.92  trt2
## 28   6.15  trt2
## 29   5.80  trt2
## 30   5.26  trt2
```

(b) (2 marks) Make a suitable plot of the data.

**Solution:** Two variables, one quantitative and one categorical, so a boxplot, as ever for an ANOVA-type design:

```
ggplot(PlantGrowth, aes(x=group, y=weight)) + geom_boxplot()
```




Extra: you should probably be concerned by the two outliers in Treatment 1. We will ignore them for this question. Apart from those, the three groups are something like symmetric with something like equal spread, so we are going to act as if all is well.

(c) (2 marks) Why is this a situation where contrasts would be helpful? Explain briefly.

**Solution:** We want to make specific, pre-planned comparisons (the two research hypotheses), rather than comparing all three pairs of groups. For example, there is no value to us in

comparing treatment 1 with the control, since this is not something that interests us.

(d) (3 marks) Set up contrasts for the two hypotheses of interest. That is, define two vectors with mnemonic names whose values reflect what you want to compare with what. (To get the order right, think about what order the treatment groups came out on your plot.)

**Solution:** On the plot, `control` was first, then `trt1`, then `trt2`, so the three numbers in each contrast have to be in that order.

The easier one is treatment 1 vs. treatment 2:

```
c_t1_vs_t2 <- c(0, 1, -1)
```

Or have the 1 and $-1$ switched, or use any number in the second place and minus that same number in the third. The first number has to be zero because control is not part of this contrast.

For the other one, we need to compare control vs. the average of the two real treatments, so the two real treatments need to get half weight and be of opposite sign to the control:

```
c_trt_vs_ctrl <- c(1, -0.5, -0.5)
```

Or flip the signs all the way through, or use 2, $-1$, $-1$, or any combination of those. If your contrasts are different from mine, some of your numbers will differ from mine later, but there is no problem as long as your contrasts capture the right thing.

(e) (2 marks) Demonstrate that your two contrasts are orthogonal.

**Solution:** Multiply them together (elementwise) and demonstrate that what you get adds up to zero, most easily:

```
sum(c_t1_vs_t2*c_trt_vs_ctrl)
## [1] 0
```

Or just do the elementwise multiplication and add up the three results in your head:

```
c_t1_vs_t2*c_trt_vs_ctrl
## [1]  0.0 -0.5  0.5
```

A zero and plus and minus the same thing, so this must add up to zero and therefore the two contrasts are orthogonal.

Extra: if you've done linear algebra, this is exactly the same concept as two vectors being orthogonal there.

(f) (2 marks) Use your two contrasts to set up for `lm` to test them via `summary`.

**Solution:** Two things to do: make a matrix out of the two contrasts, and then make it be what is tested:

```
m <- cbind(c_t1_vs_t2,c_trt_vs_ctrl)
contrasts(PlantGrowth$group) <- m
```

In the second line, this is data frame, dollar sign, column that the contrasts are for comparing (groups).

(g) (2 marks) Fit an appropriate model and display its summary.

**Solution:** Fit the model *as an* `lm`:

```
growth.1 <- lm(weight~group, data=PlantGrowth)
summary(growth.1)
##
```

```
## Call:
## lm(formula = weight ~ group, data = PlantGrowth)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0710 -0.4180 -0.0060  0.2627  1.3690
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)          5.0730     0.1138  44.573  < 2e-16 ***
## groupc_t1_vs_t2     -0.4325     0.1394  -3.103  0.00446 **
## groupc_trt_vs_ctrl  -0.0410     0.1610  -0.255  0.80086
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6234 on 27 degrees of freedom
## Multiple R-squared:  0.2641,Adjusted R-squared:  0.2096
## F-statistic: 4.846 on 2 and 27 DF,  p-value: 0.01591
```

I don't mind how you combine the work in (f) and (g), as long as code like those four lines is there somewhere.

(h) (3 marks) What do you conclude? Explain briefly why that makes sense by looking at the plot you drew earlier.

**Solution:** The two "slope" lines in the output encode the two contrasts you want to test (which is why giving them mnemonic names is helpful). The first one compares treatment 1 against treatment 2; with a P-value of 0.0045, this is significant, so there is a significant difference between the two treatments. The second one, with a P-value of 0.80, is not anywhere near significant, so there is no difference between the average of the two treatments and the control.

Why does that make sense? Well, on the boxplot, treatment 1 came out highest, treatment 2 came out lowest, and the control was in the middle. That big difference between the two treatments came out significant, but if you average the two treatments together, they come out about the same as the control, so that contrast was not anywhere near significant.

Extra 1: if you write your contrasts differently from mine, your numbers in the Estimate column may differ from mine, but you should end up with the same P-values. This is what I meant about it not mattering if you switch your signs all the way through a contrast, or multiply a contrast through by something like 2.

Extra 2: having things come out this way is kind of unusual. Typically what happens is that all the treatments are at least as good as the control, so that the contrast with the control comes out significant, but the treatments may or may not be different from each other, so the contrast comparing them may or may not be significant. But this one is different from that.

# Notes

[1] Otherwise, why make it?

[2] It is not enough to be able to parrot the definition; you need to *understand* the definition well enough to be able to use it.

[3] You should not look at the Tukey comparisons when the ANOVAs were non-significant, but here none of those are significant anyway.

[4] As for regression summaries, the Tukey output is for looking at rather than computing with.