Workshop 3 - General Linear Model (ANOVA)

Andrew Stewart

Andrew.Stewart@manchester.ac.uk

@ajstewart_lang

https://github.com/ajstewartlang

Workshop	Topic
1	Reproducibility and R
2	General Linear Model (Regression)
3	General Linear Model (ANOVA)
4	Mixed Models
5	Data Simulation and Advanced Data Visualisation
6	Reproducible Computational Environments and Presentations

Assignment

Assignment to be completed by Semester 1 exam period.

- We're going to have our first look at the Analysis of Variance (ANOVA).

- We'll look at ANOVA for within-subjects, between-subjects and mixed designs.

- ANOVA is an important statistical test and (in various forms) is used widely across many areas of psychology.

- We'll finish today by seeing how ANOVA and regression are both really the same thing and based on the GLM.

Why ANOVA, why not t-tests?

- So, t-tests are fine if we're just comparing two means.

- In the real world of psychology, we often have more than two conditions.

- How could we analyse our data ?

- One possibility could be that we do multiple t-tests – but there's a problem with that.

- With one t-test, at $p < 0.05$ level of significance there is a 5% chance of falsely rejecting our null hypothesis (type 1 error).

- If we have three conditions, then we have three pairs of means to compare (condition 1 vs condition 2, condition 2 vs condition 3 and condition 1 vs condition 3).

- For each test, there is 0.95 probability of not having a type 1 error.

- But when we do three tests the probability is 0.95 x 0.95 x 0.95 which equals 0.857.

- So that means there is a 14.3% chance of us falsely rejecting the null hypothesis (1-0.857) x 100 = 14.3

The familywise error rate

- This is known as the <u>familywise</u> error rate.

$$familywise\ error = 1 - (0.95)^n$$

- If we had 5 conditions, and hence 10 t-tests to conduct, our error rate would be 0.4 – which means there is a 40% chance of having made at least one type I error (i.e., thinking we have an effect when none is present).

Similarities between t-tests and the ANOVA

- t-tests tell us whether or not two samples have the same mean.

- ANOVA tells us whether two or more samples have the same mean.

- As the t-test produced the t-statistic, the ANOVA gives us an F-statistic or F-ratio which compares the amount of systematic variance with the amount of unsystematic variance.

- ANOVA can tell us that there is a difference between means – so for three samples it tells us that $\overline{X}_1 = \overline{X}_2 = \overline{X}_3$ is <u>not</u> true.

- But it doesn't tell us where the difference is.

- It doesn't tell us whether $\overline{X}_1$ differs from both $\overline{X}_2$ and $\overline{X}_3$ or whether $\overline{X}_2$ differs from $\overline{X}_3$ but not $\overline{X}_2$ etc.

ANOVA

- Imagine we're interested in the impact of caffeine consumption on an individual's motor performance.

- It's a between-subjects design with 3 conditions:

  - low amount of caffeine (single espresso)

  - large amount of caffeine (double espresso)

  - placebo group (water)

- We conduct an ANOVA and find a significant F-ratio.

- What does it mean?

- The single espresso people could have performed better from the double espresso and water group.

- Or maybe they performed the same as the water group but better than the double espresso group.

- Or maybe (unexpectedly) they performed <u>worse</u> than both the double espresso and water groups.

- To know what is the case we need to do planned contrasts (similar to 1 tailed tests) or post hoc tests (similar to 2 tailed tests).

- We know that at least one of our means differs from at least one of our other means but (so far) we don't know where that difference lies…...

- Luckily things easy for us as we can conduct what are known as post hoc tests. These will tell us which means differ from which other means (and allow us to begin to tell a story….)

Post hocs tests

- Work by doing pairwise comparisons on all the different combinations of experimental groups…..

- They control for the familywise error rate though to get round that problem.

- Bonferroni method divides our critical p value (0.05) by the number of tests. If we are conducting ten tests, then for each test the critical p is 0.005 – but this increases our chances of a type II error – missing an effect when it's there.

When deciding which post hoc test to use :

Does it control the Type I error rate ?

Does it control the Type II error rate ?

Is it reliable when ANOVA assumptions have been violated ?

LSD, Bonferroni, and Tukey tests.

- The least significant differences test (LSD) doesn't control the Type I error and is like doing multiple t-tests on the data (but only if the ANOVA is significant).

- Bonferroni and Tukey both control for Type I errors but are conservative. Bonferroni works by dividing the critical alpha level by the number of tests conducted.

- Tukey is less conservative than Bonferroni.

The Packages

```
library(tidyverse) #load the tidyverse packages

library(afex) #load afex for running ANOVA

library(emmeans) #load emmeans for running pairwise comparisons
```

ANOVA

We have 45 participants, a between participants condition with 3 levels (Water *vs.* Single Espresso *vs.* Double Espresso), and Ability as our DV measured on a continuous scale.

```
cond <- read_csv("data_files/cond.csv")

cond$Condition <- as.factor(cond$Condition)
```

```
> cond
# A tibble: 45 x 3
   Participant Condition Ability
         <dbl> <fct>       <dbl>
 1           1 Water        4.82
 2           2 Water        5.41
 3           3 Water        5.73
 4           4 Water        4.36
 5           5 Water        5.47
 6           6 Water        5.50
 7           7 Water        5.07
 8           8 Water        5.08
 9           9 Water        5.07
10          10 Water        4.94
# … with 35 more rows
```

We have three columns - Participant number, Condition, and Ability. Condition is our IV, and Ability our DV. Note, our data are in tidy format with one observation per row.

Generating Descriptives and Visualising the Data

```
cond %>%
  group_by(Condition) %>%
  summarise(mean = mean(Ability), sd = sd(Ability))
```

```
# A tibble: 3 x 3
  Condition            mean     sd
  <fct>               <dbl> <dbl>
1 Double Espresso      8.89 0.467
2 Single Espresson     6.99 0.419
3 Water                5.17 0.362
```

```
cond %>%
  ggplot(aes(x = Condition, y = Ability, colour = Condition)) +
  geom_violin() +
  geom_jitter(width = .1) +
  guides(colour = FALSE) +
  stat_summary(fun.data = "mean_cl_boot", colour = "black") +
  theme(text = element_text(size = 15))
```
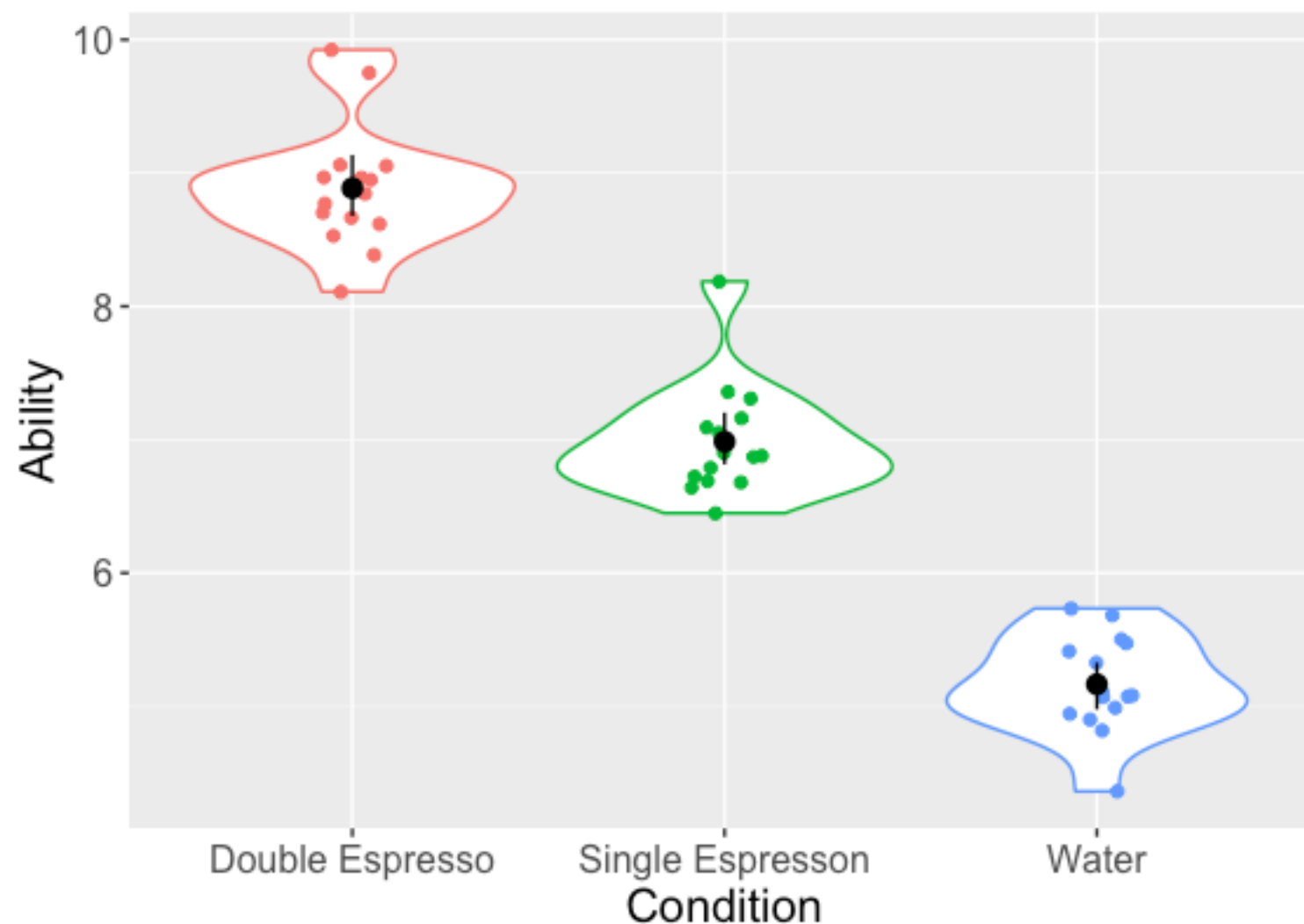
```
library(afex)

model <- aov_4(Ability ~ Condition + (1 | Participant),
data = cond)
```

This is our DV   **This is our IV**   **This is our random effect**

```
> summary(model)
Anova Table (Type 3 tests)

Response: Ability
          num Df den Df     MSE      F      ges     Pr(>F)
Condition      2     42 0.17484 297.05 0.93397 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

To determine what's driving the effect we can use the `emmeans::emmeans()` to run pairwise comparisons (note, default is Tukey correction).

```
> emmeans(model, pairwise ~ Condition)
$emmeans
 Condition          emmean     SE df lower.CL upper.CL
 Double Espresso      8.89 0.108 42     8.67     9.10
 Single Espresson     6.99 0.108 42     6.77     7.20
 Water                5.17 0.108 42     4.95     5.38

Confidence level used: 0.95

$contrasts
 contrast                              estimate     SE df t.ratio
p.value
 Double Espresso - Single Espresson        1.90 0.153 42 12.453  <.0001
 Double Espresso - Water                   3.72 0.153 42 24.372  <.0001
 Single Espresson - Water                  1.82 0.153 42 11.920  <.0001

P value adjustment: tukey method for comparing a family of 3
estimates
```

Measure of Effect Size

- The effect size is measured by *ges* which stands for generalised effect size or generalised eta squared (η_G^2).

- For designs with more than one factor it can be a useful indicator of how much variance in the dependent variable can be explained by each factor (plus any interactions between factors).

```
> summary(model)
Anova Table (Type 3 tests)

Response: Ability
          num Df den Df     MSE       F      ges       Pr(>F)
Condition      2     42 0.17484 297.05 0.93397 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

So, to make sense of our output

- We found a significant effect of Beverage type (F (2,42) = 297.05, p < .001, generalised η2 = .93). Bonferroni comparisons revealed that the Water group differed significantly worse than the Single Espresso Group (p < .001), that the Water group differed significantly worse the Double Espresso Group (p < .001), and that the Single Espresso Group permed significantly worse than the Double Espresso Group (p < .001).

- In other words, drinking a some coffee improves motor performance relative to drinking water, and drinking a lot of coffee improves motor performance even more.

ANOVA for repeated measures designs

- Let's imagine we have an experiment where we asked 32 participants to memorise words of differing levels of spelling complexity - Very Easy, Easy, Hard, and Very Hard.

- They were presented with these words in an initial exposure phrase. After a 30 minute break we tested them by asking them to write down all the words. We scored them as number correct for each condition.

- We want to know whether there is a difference in the number of words they remembered for each level of spelling complexity.

```
rm_data <- read_csv("data_files/rm_data.csv")
rm_data$Condition <- as.factor(rm_data$Condition)
rm_data

# A tibble: 128 x 3
   Participant Condition Score
         <dbl> <fct>     <dbl>
 1           1 Very Easy    80
 2           2 Very Easy    86
 3           3 Very Easy    89
 4           4 Very Easy    75
 5           5 Very Easy    86
 6           6 Very Easy    87
 7           7 Very Easy    82
 8           8 Very Easy    82
 9           9 Very Easy    82
10          10 Very Easy    81
# … with 118 more rows
```

Generating Descriptives and Visualising the Data

```
rm_data %>%
  group_by(Condition) %>%
  summarise(mean = mean(Score), sd = sd (Score))

# A tibble: 4 x 3
  Condition    mean     sd
  <fct>       <dbl> <dbl>
1 Easy         81.6   4.28
2 Hard         72.4   6.24
3 Very Easy    83.5   3.62
4 Very Hard    54.0   5.50
```

```
rm_data %>%
  ggplot(aes(x = fct_reorder(Condition, Score), y = Score, colour =
Condition)) +
  geom_violin() +
  geom_jitter(width = .1) +
  guides(colour = FALSE) +
  stat_summary(fun.data = "mean_cl_boot", colour = "black") +
  theme(text = element_text(size = 15)) +
  labs(x = "Condition")
```
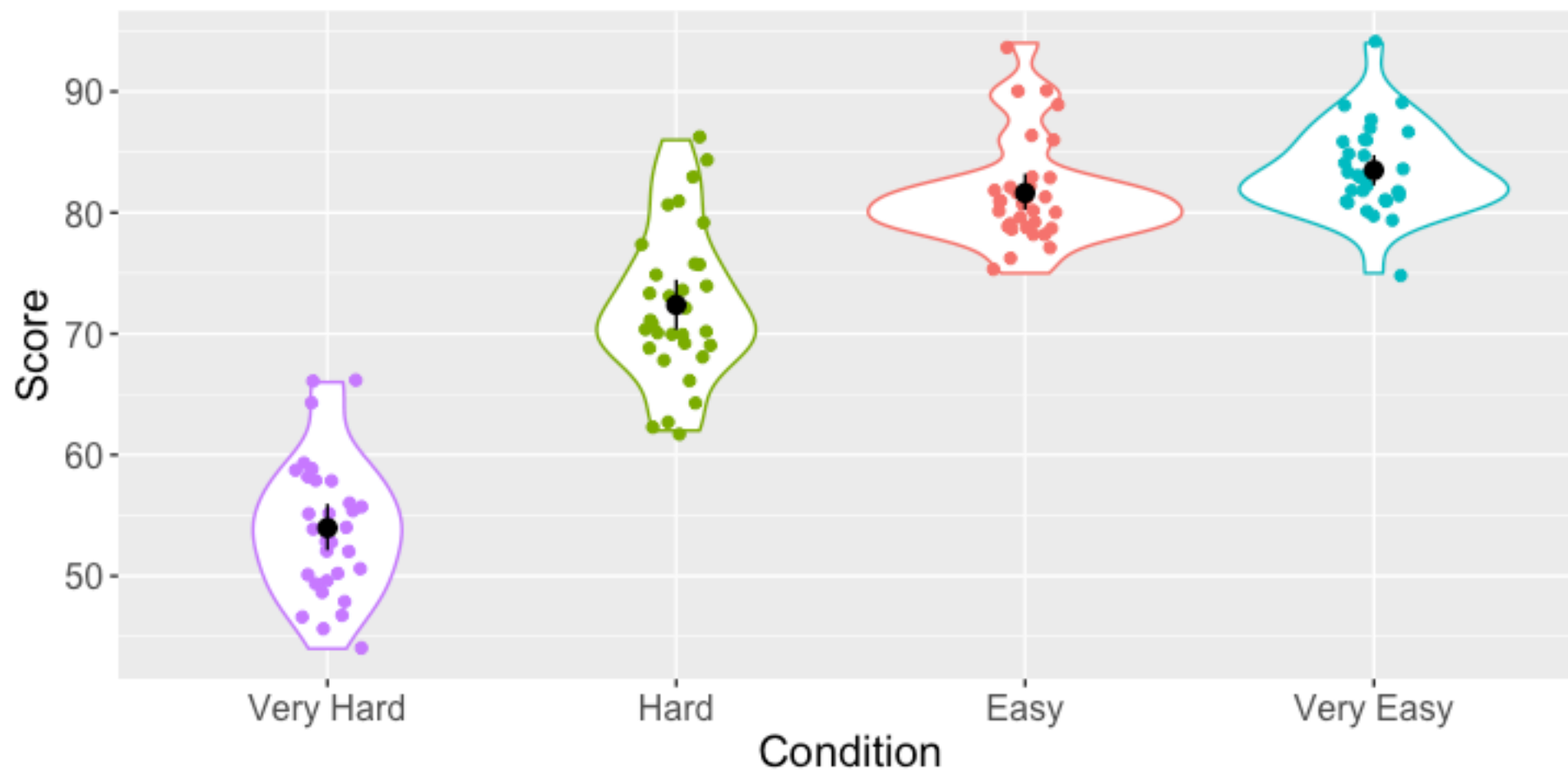
This is the our ANOVA model - we have a significant effect of Condition.

```
> model <- aov_4(Score ~ Condition + (1 + Condition | Participant), data = rm_data)
> summary(model)

Univariate Type III Repeated-Measures ANOVA Assuming Sphericity

                SS num Df Error SS den Df        F    Pr(>F)
(Intercept) 679632      1   936.49     31 22497.36 < 2.2e-16 ***
Condition    17509      3  2179.48     93   249.04 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Mauchly Tests for Sphericity

          Test statistic p-value
Condition        0.90603 0.71042


Greenhouse-Geisser and Huynh-Feldt Corrections
 for Departure from Sphericity

          GG eps Pr(>F[GG])
Condition 0.9401  < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

            HF eps   Pr(>F[HF])
Condition 1.043895 2.615157e-44
```

```
> anova(model)
Anova Table (Type 3 tests)

Response: Score
          num Df den Df    MSE       F       ges      Pr(>F)
Condition 2.8203  87.43 24.928 249.04 0.84892 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The effect size is measured by *ges* which stands for generalised effect size (η_G^2) - this is the recommended effect size measure for repeated measures designs (Bakeman, 2005). We get this by using the `anova()` function on our model. Note the dfs in this output are always corrected as if there is a violation of sphericity - to be conservative (and to avoid Type 1 errors) we might be better off to always choose these corrected dfs.

Where does the difference lie?

```
> emmeans(model, pairwise ~ Condition, adjust = "Bonferroni")
$emmeans
 Condition emmean     SE  df lower.CL upper.CL
 Easy           81.6 0.886 122     79.9     83.4
 Hard           72.4 0.886 122     70.6     74.1
 Very.Easy      83.5 0.886 122     81.7     85.3
 Very.Hard      54.0 0.886 122     52.2     55.7

Warning: EMMs are biased unless design is perfectly balanced
Confidence level used: 0.95

$contrasts
 contrast                    estimate    SE df t.ratio p.value
 Easy - Hard                     9.25 1.21 93   7.643  <.0001
 Easy - Very.Easy               -1.88 1.21 93  -1.549  0.7483
 Easy - Very.Hard               27.66 1.21 93  22.852  <.0001
 Hard - Very.Easy              -11.12 1.21 93  -9.192  <.0001
 Hard - Very.Hard               18.41 1.21 93  15.209  <.0001
 Very.Easy - Very.Hard          29.53 1.21 93  24.401  <.0001

P value adjustment: bonferroni method for 6 tests
```

So far we have looked at ANOVA for designs when we have one factor which is between subjects (i.e., each participant appears in one condition), and for designs when we have one factor that is repeated measures (each participant appears in all conditions. These are examples of 1-way ANOVA.

Now we're going to look at factorial ANOVA - this is for cases where we have more than one factor and we might be interested in how the two factors interact with each other. If we have two factors, we have a 2-way ANOVA, three factors a 3-way ANOVA etc.

- Imagine we have 2 factors. Factor 1 with two levels, Factor 2 with three. Our analysis might reveal a main effect of Factor 1 (i.e., a difference between the two levels), a main effect of Factor 2 (i.e., a difference between the three levels) or an interaction between the two…..
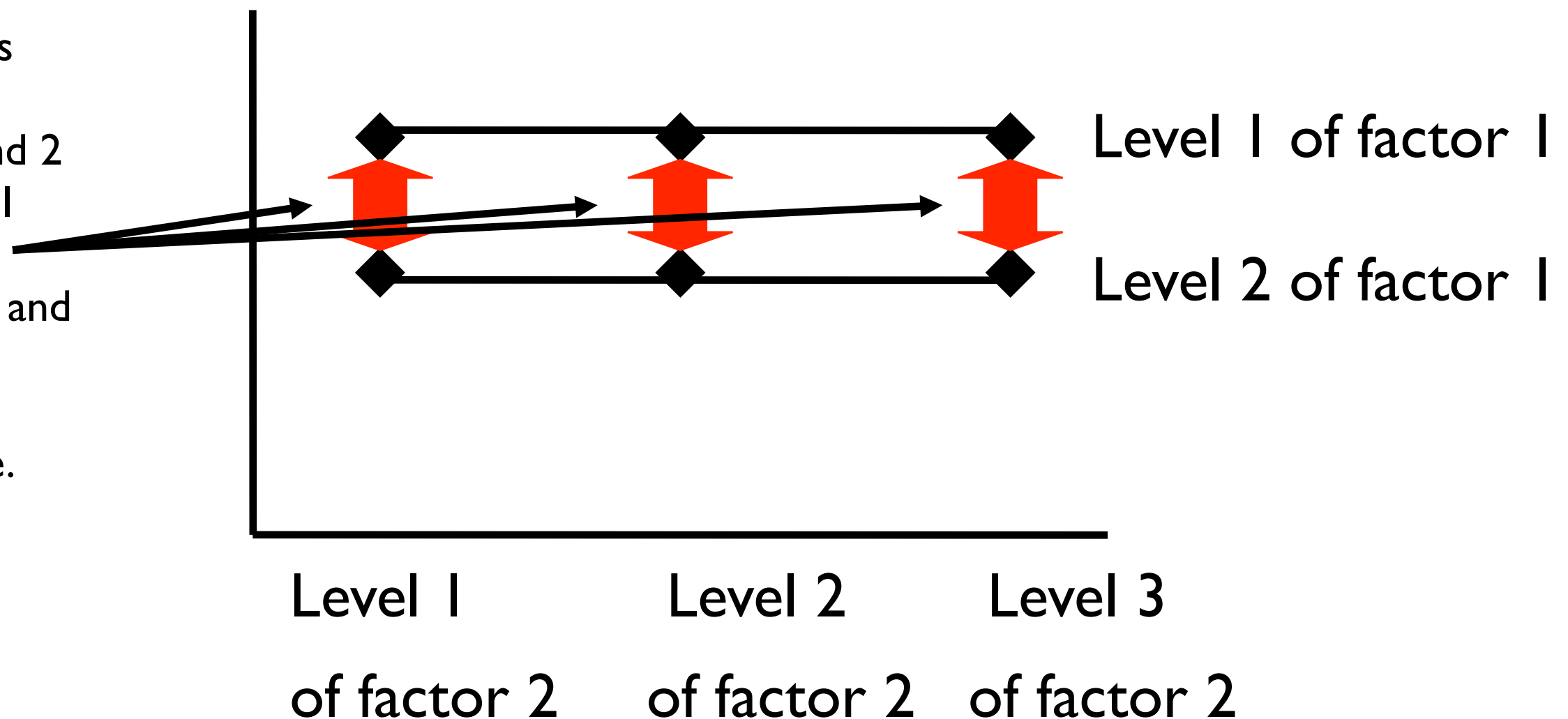
- This is a ②x③ANOVA

Corresponds to Factor 1 – it has two levels.

Corresponds to Factor 2 – it has three levels.

Main effect of Factor 1, no main effect of Factor 2 and no interaction

The differences between levels 1 and 2 of Factor 1 are all significant and are of the same magnitude.
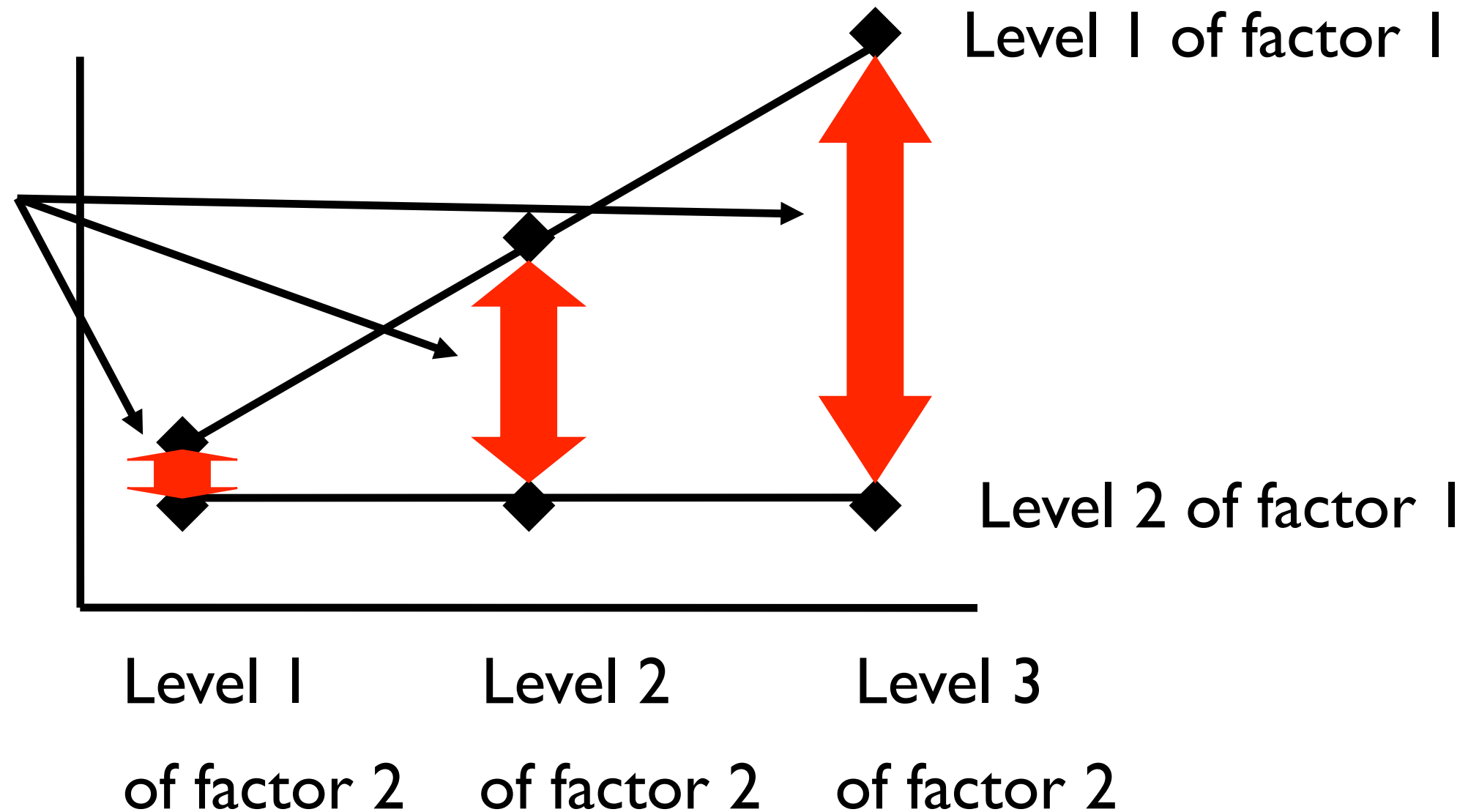
Level 1 of factor 1

Level 2 of factor 1

Level 1 of factor 2

Level 2 of factor 2

Level 3 of factor 2

No main effect of Factor 1, main effect of Factor 2 and no interaction

Level 1 of factor 1

The differences between levels 1 & 2 and 2 & 3 of Factor 2 are all significant and are of the same magnitude. There are no significant differences between levels 1 and 2 of Factor 1.

Level 2 of factor 1

Level 1 of factor 2

Level 2 of factor 2
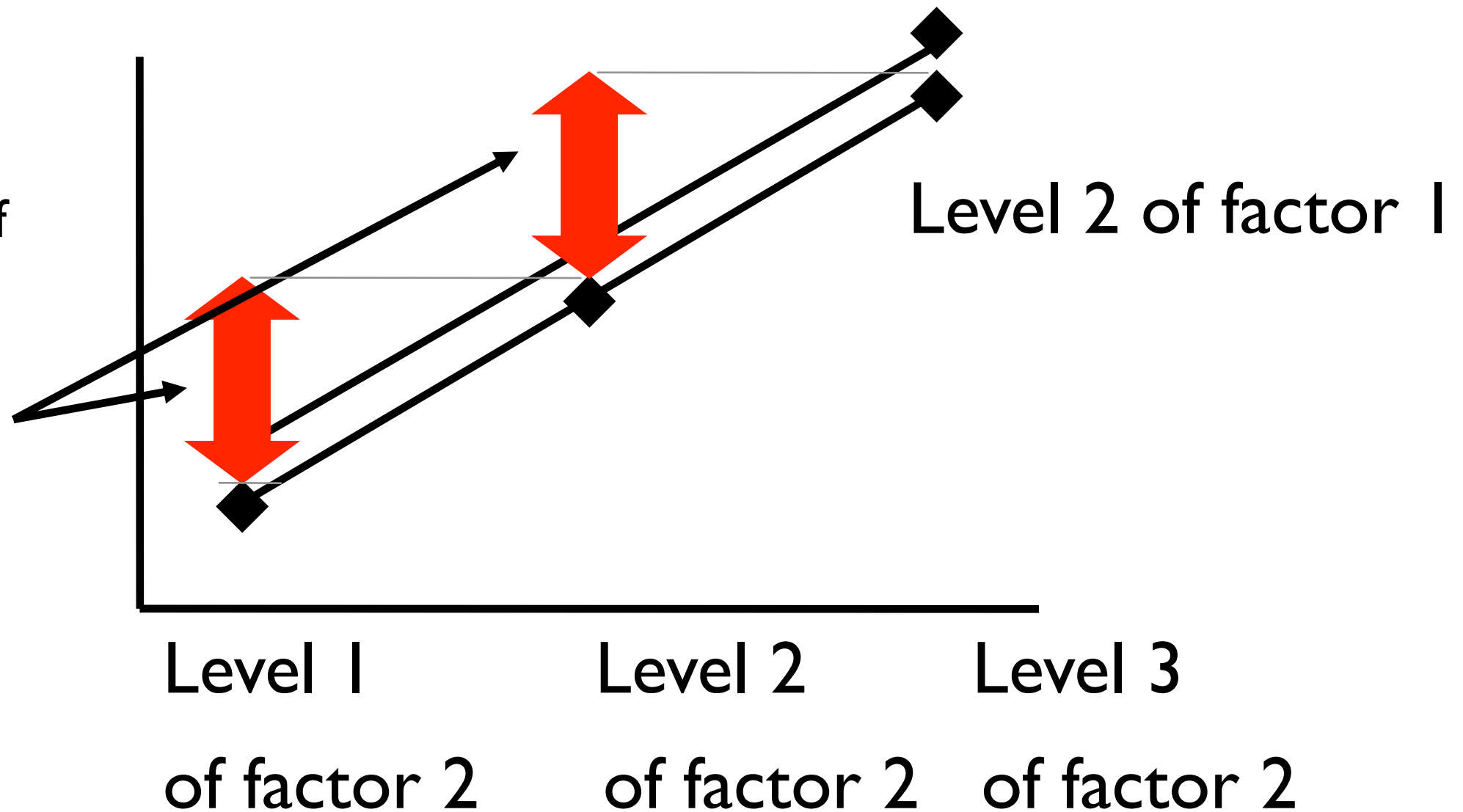
Level 3 of factor 2

Main effect of Factor 1, main effect of Factor 2 and an interaction

The differences between the two levels of factor 1 change as a function of factor 2.

Level 1 of factor 1

Level 2 of factor 1

Level 1 of factor 2

Level 2 of factor 2

Level 3 of factor 2

No main effect of Factor 1, no main effect of Factor 2 but an interaction

The difference between levels 1 & 2 of Factor 1 at Level 1 of Factor 2 is different from the same difference at Levels 2 and 3 of Factor 2. This is a crossover interaction as the polarity of the difference flips.

Level 1 of factor 1

Level 2 of factor 1

Level 1 of factor 2

Level 2 of factor 2

Level 3 of factor 2
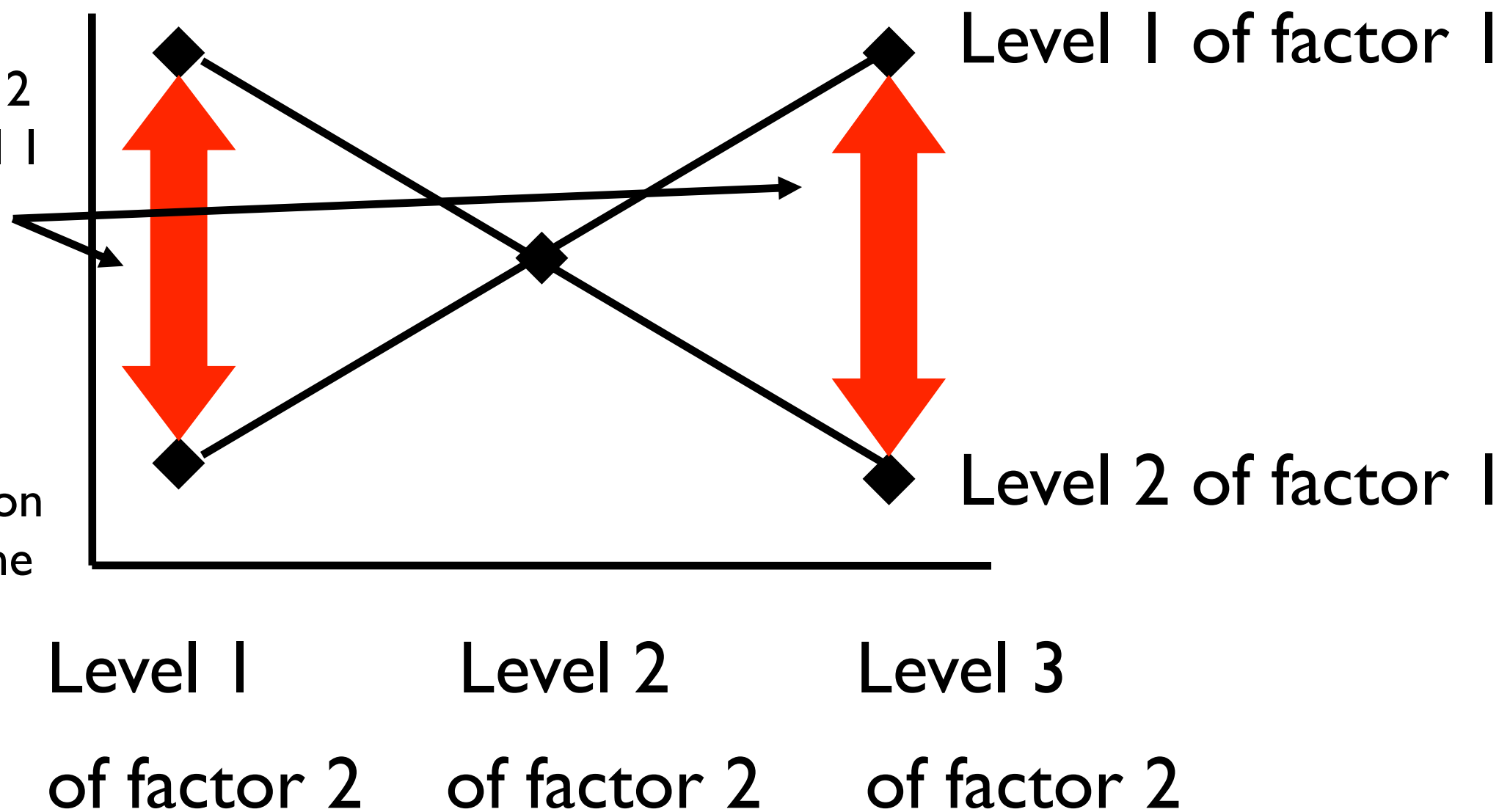
Factorial ANOVA

- Imagine the case where we're interested in the effect of positive vs. negative contexts on how quickly (in milliseconds) people respond to positive vs negative sentences. We think there might be a priming effect (i.e., people are quicker to respond to positive sentences after positive contexts vs. after negative contexts - and vice versa).

- So, we have two factors, each with two levels. This is what's known as a full factorial design where every subject participates in every condition.

```
fact_data <- read_csv("data_files/fact_data.csv")
fact_data$Sentence <- as.factor(fact_data$Sentence)
fact_data$Context <- as.factor(fact_data$Context)

fact_data

# A tibble: 1,680 x 5
   Subject   Item     RT Sentence Context
     <dbl>  <dbl>  <dbl> <fct>    <fct>
 1       1      3   1270 Positive Negative
 2       1      7    739 Positive Negative
 3       1     11    982 Positive Negative
 4       1     15   1291 Positive Negative
 5       1     19   1734 Positive Negative
 6       1     23   1757 Positive Negative
 7       1     27   1052 Positive Negative
 8       2      4   1706 Positive Negative
 9       2      8    533 Positive Negative
10       2     12   1009 Positive Negative
# … with 1,670 more rows
```
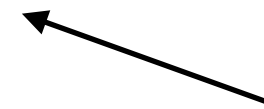
Generating Descriptives and Visualising the Data

```
fact_data %>%
  group_by(Context, Sentence) %>%
  summarise(mean = mean(RT), sd = sd(RT))

# A tibble: 4 x 4
# Groups:    Context [2]
  Context   Sentence  mean      sd
  <fct>     <fct>     <dbl> <dbl>
1 Negative  Negative  1474.   729.
2 Negative  Positive    NA      NA
3 Positive  Negative    NA      NA
4 Positive  Positive  1579.   841.
```
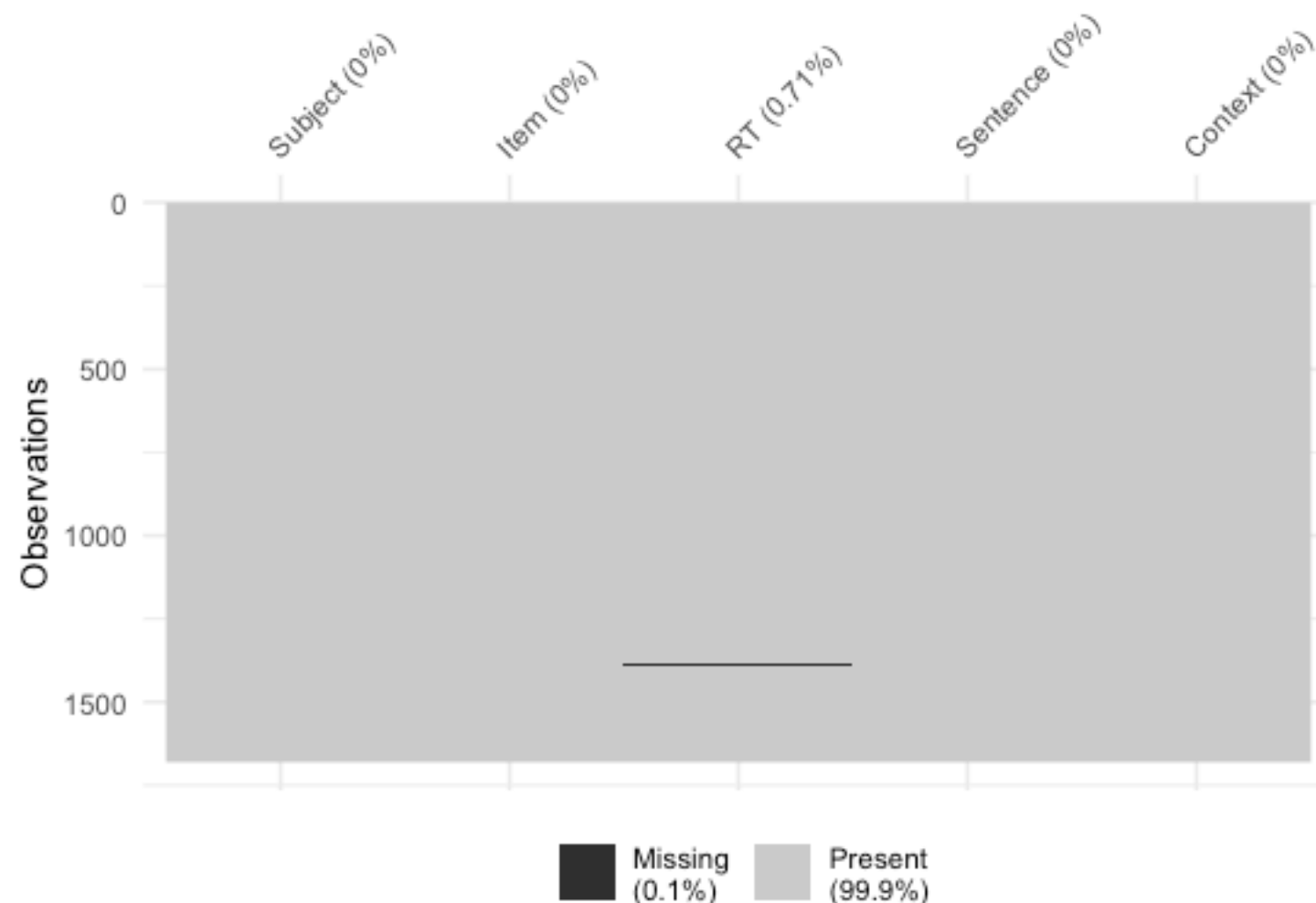
What's going on here?

Let's visualise the whole dataset...

We can use a function in a package without loading it into our library using `package_name::function_name` like this:

```
visdat::vis_miss(fact_data)
```

Let's ignore the missing data (NAs) - one way:

```
fact_data %>%
  filter(!is.na(RT)) %>%
  group_by(Context, Sentence) %>%
  summarise(mean = mean(RT), sd = sd(RT))

# A tibble: 4 x 4
# Groups:   Context [2]
  Context  Sentence  mean     sd
  <fct>    <fct>     <dbl> <dbl>
1 Negative Negative  1474.  729.
2 Negative Positive  1595.  887.
3 Positive Negative  1633.  877.
4 Positive Positive  1579.  841.
```

or another way...

```
fact_data %>%
  group_by(Context, Sentence) %>%
  summarise(mean = mean(RT, na.rm = TRUE),
     sd = sd(RT, na.rm = TRUE))

# A tibble: 4 x 4
# Groups:   Context [2]
  Context   Sentence   mean     sd
  <fct>     <fct>      <dbl> <dbl>
1 Negative  Negative   1474.  729.
2 Negative  Positive   1595.  887.
3 Positive  Negative   1633.  877.
4 Positive  Positive   1579.  841.
```

```
fact_data %>%
  ggplot(aes(x = Context:Sentence, y = RT, colour = Context:Sentence)) +
  geom_violin() +
  geom_jitter(width = .1, alpha = .25) +
  guides(colour = FALSE) +
  stat_summary(fun.data = "mean_cl_boot", colour = "black") +
  theme(text = element_text(size = 15), axis.text.x = element_text(angle =
45, hjust = 1)) +
  labs(x = "Condition")
```
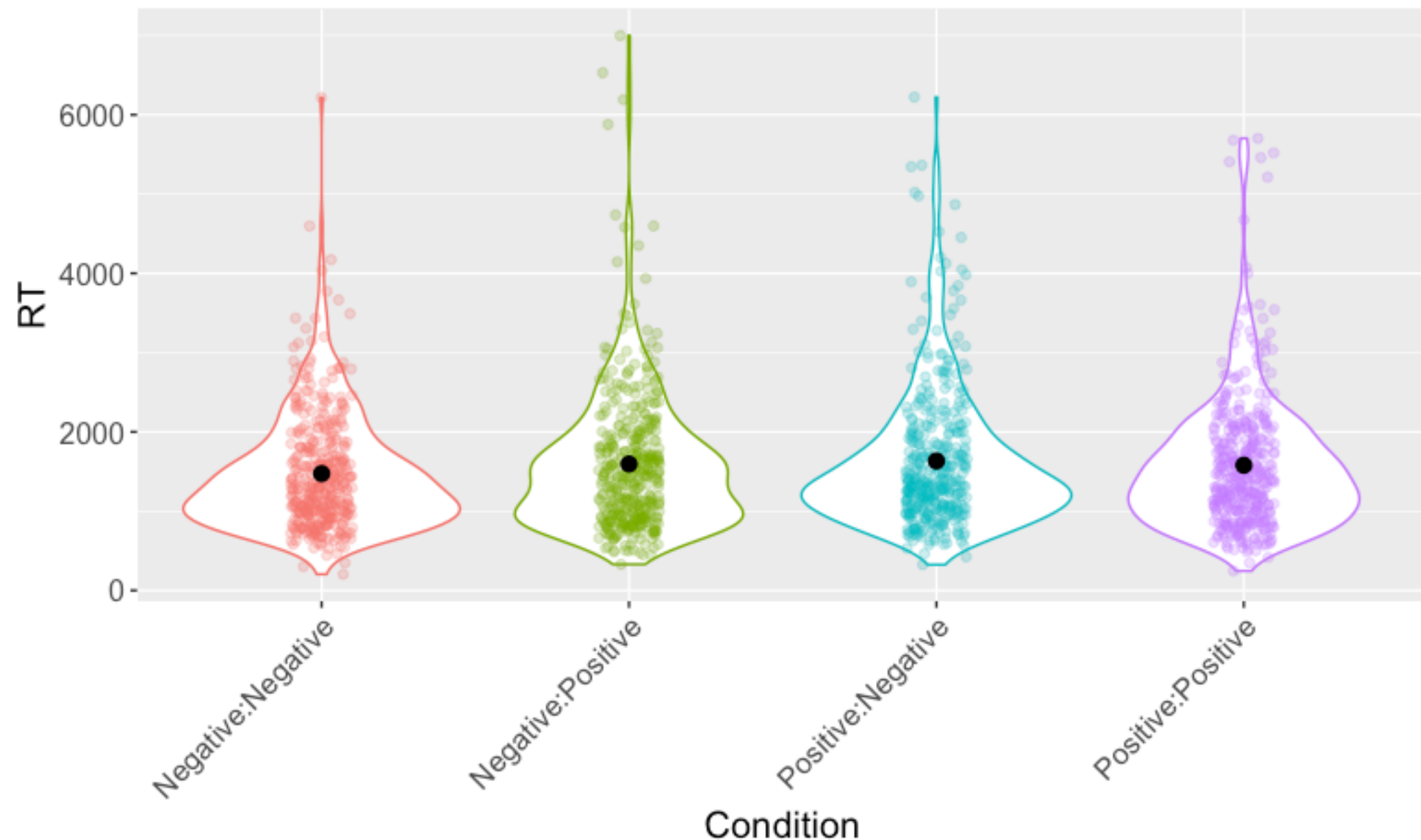
By Subjects

```
model_subjects <- aov_4(RT ~ Context * Sentence + (1 + Context
* Sentence | Subject), data = fact_data, na.rm = TRUE)
```

- Syntax corresponds to RT being predicted by the two factors (`Context * Sentence`) corresponds to two main effects plus the interaction) plus the random effect by Subjects using the datafile called DV. By setting na.rm to be TRUE, we are telling the analysis to ignore individual trials where there might be missing data - effectively this calculates the condition means over the data that is present (and ignores trial where it is missing).

- aov_4 aggregates over the grouping term in the random effect. Simply change to (1 + `Context * Sentence` | Item) for by-item (i.e., F2) analysis. This requires the data to contain the individual observations (not aggregated as means).

```
> anova(model_subjects)
Anova Table (Type 3 tests)

Response: RT
                   num Df den Df     MSE      F        ges   Pr(>F)
Context                 1     59   90195 3.1767 0.0060231 0.07984 .
Sentence                1     59  124547 0.6283 0.0016524 0.43114
Context:Sentence        1     59   93889 4.5967 0.0090449 0.03616 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- The output contains the main effect of Sentence, the main effect of Context, and the interaction between the two. Associated with each are the dfs, the Mean Squared Error, the F ratio, the generalized eta-squared, and p-value. Note, you can ask for partial eta-squared as effect size measure too.

By Items

```
> model_items <- aov_4(RT ~ Context * Sentence + (1 + Context * Sentence | Item),
data = DV, na.rm = TRUE)

> anova(model_items_

Anova Table (Type 3 tests)

Response: RT
                 num Df den Df      MSE        F        ges    Pr(>F)
Context               1     27   39844  4.0013  0.0080150  0.05561 .
Sentence              1     27  203164  0.1221  0.0012553  0.72951
Context:Sentence      1     27   40168  5.7687  0.0116070  0.02346 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- With the same datafile and just by changing *one* word in the analysis code.

Interpreting Interactions

We can build the model as before and pass the model to the function `emmeans` (remember to load the `emmeans` package) and ask for pairwise comparisons with no correction - we need to work out the Bonferroni corrected value ourselves…

```
> emmeans(model_subjects, pairwise ~ Context * Sentence, adjust = "none")
$emmeans
 Context  Sentence emmean   SE  df lower.CL upper.CL
 Negative Negative   1474 57.8 138     1360     1588
 Positive Negative   1628 57.8 138     1514     1742
 Negative Positive   1595 57.8 138     1481     1709
 Positive Positive   1579 57.8 138     1465     1693

Warning: EMMs are biased unless design is perfectly balanced
Confidence level used: 0.95

$contrasts
 contrast                                estimate   SE  df t.ratio p.value
 Negative,Negative - Positive,Negative     -153.9 55.4 118  -2.779  0.0064
 Negative,Negative - Negative,Positive     -120.9 60.3 116  -2.004  0.0474
 Negative,Negative - Positive,Positive     -105.2 59.8 115  -1.759  0.0813
 Positive,Negative - Negative,Positive       33.0 59.8 115   0.551  0.5824
 Positive,Negative - Positive,Positive       48.7 60.3 116   0.807  0.4213
 Negative,Positive - Positive,Positive       15.7 55.4 118   0.284  0.7772
```

Results

We conducted a 2 (Context: Positive vs. Negative) x 2 (Sentence: Positive vs. Negative) repeated measures ANOVA to investigate the influence of context valence on reaction times to words of the same or different valence. The ANOVA revealed no effect of Sentence ($F < 1$), no effect of Context ($F(1, 59) = 3.18$, $p = .080$, $\eta_G^2 = .006$), but an interaction between Sentence and Context ($F(1, 59) = 4.60$, $p = .036$, $\eta_G^2 = .009$).

The interaction was interpreted by conducting Bonferroni-corrected pairwise companions. These comparisons revealed that the interaction was driven by Negative sentences being processed faster in Negative vs. Positive contexts (1,474 ms. vs. 1,628 ms., $t(118) = 2.78$, $p = .006$) while Positive sentences were read equivalently in Negative vs. Positive contexts (1,595 ms. vs. 1,579 ms., $t(118) = .284$, $p = .777$).

To ANOVA_part_1_lab_script_questions...

ANOVA part 2...

- Earlier we looked at 1-way between participants ANOVA, 1-way repeated measures ANOVA, and 2-way repeated measures ANOVA.

- We used the `afex` package for building our models as it uses Type III Sums of Squares with effect coding of contrasts (allowing us to more easily interpret our results when we have interactions).

A slightly more complex study

- Similar to our 2 x 2 ANOVA from last time in that we're looking at priming (this time a word following an image), but let's say we now have Age as an additional factor. It's a between subjects factor. We might think that priming effects might be different for young vs old people.

- So we need to run a 2 x 2 x 2 ANOVA. The first two factors are still within subjects (i.e., repeated measures), but our new one (age) is between subjects and has two levels.

	Participant	Image	Word	Age	RT
1	1	Pos	Pos	Young	719
2	2	Pos	Pos	Young	756
3	3	Pos	Pos	Young	777
4	4	Pos	Pos	Young	691
5	5	Pos	Pos	Young	760
6	6	Pos	Pos	Young	762
7	7	Pos	Pos	Young	735
8	8	Pos	Pos	Young	736
9	9	Pos	Pos	Young	735
10	10	Pos	Pos	Young	727
11	11	Pos	Pos	Young	738
12	12	Pos	Pos	Young	725
13	13	Pos	Pos	Young	730
14	14	Pos	Pos	Young	751
15	15	Pos	Pos	Young	773
16	16	Pos	Pos	Young	747
17	1	Pos	Neg	Young	834
18	2	Pos	Neg	Young	822

Showing 1 to 18 of 128 entries

Remember, for the `aov_4` function we need each factor to be in its own column and for each row to be one observation - this is long or tidy format data.

```
ggplot(my_data, aes(x = Word:Image, y = RT, colour = Word:Image)) +
    geom_violin() +
    geom_jitter(width = .1, alpha = .2) +
    stat_summary(fun.data = "mean_cl_boot") +
    theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
    facet_wrap(~ Age) +
    guides(colour = FALSE)
```
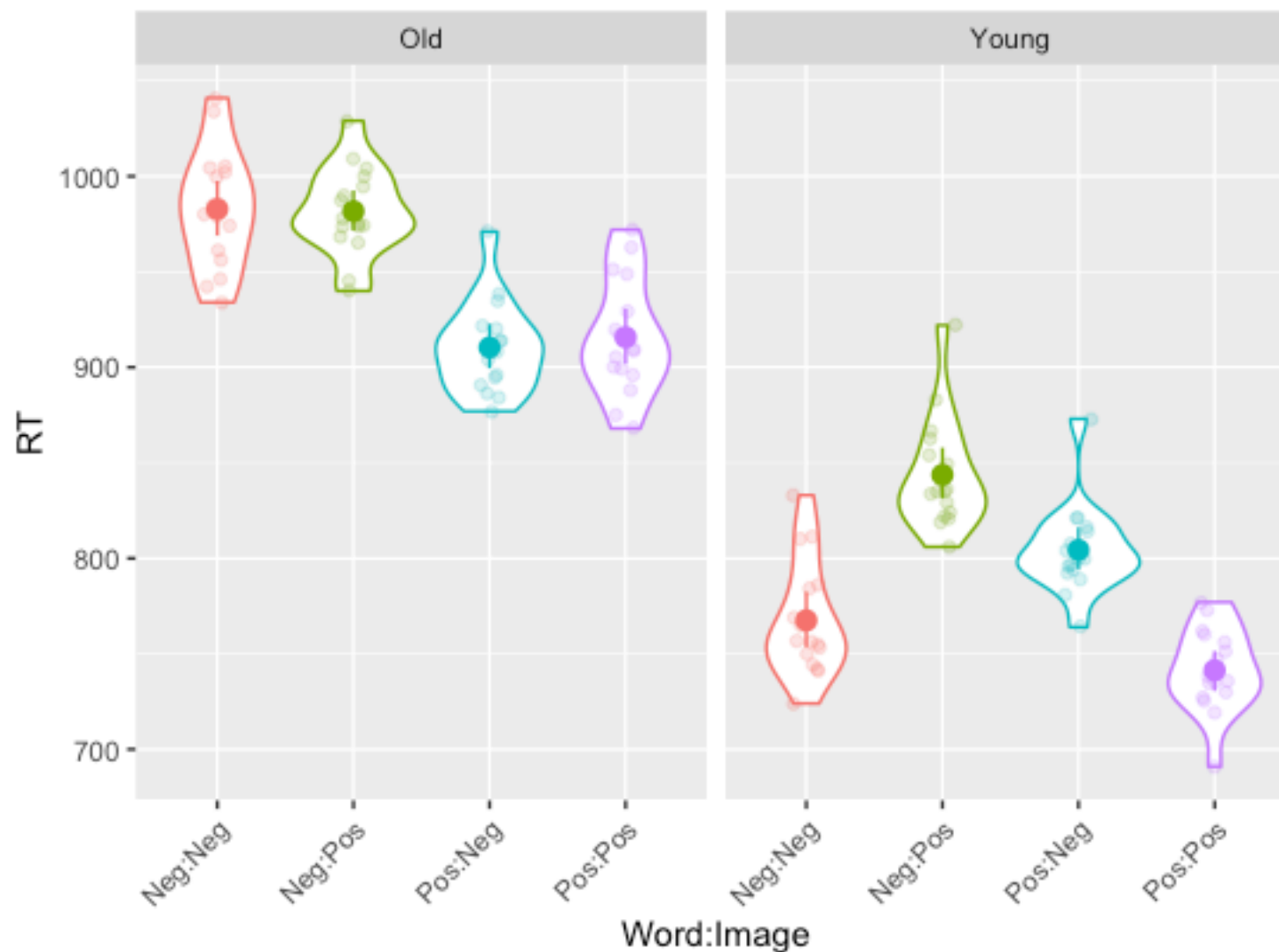

We can see it looks like the Young and Old groups are behaving a little differently.

We need to build our model with two repeated and one between participants factor…

```
model <- aov_4(RT ~ Word * Image * Age + (1 + Word * Image |
Participant), my_data)
```

We are asking for the model to be built using the three factors - this will give us three possible main effects, 3 possible 2-way interactions, and a possible 3-way interaction…

The `aov_4` function knows which factors are repeated and which are between from the model structure - note that between participant factors shouldn't appear in the random effects term - if you have only between factors then the term should be something like (1 | Participant)…

```
> anova(model)
Anova Table (Type 3 tests)

Response: RT
                num Df den Df     MSE          F      ges      Pr(>F)
Age                  1     30  788.86  1017.8790  0.90328  < 2.2e-16 ***
Word                 1     30  750.85   110.7147  0.49157  1.380e-11 ***
Age:Word             1     30  750.85    14.1946  0.11029  0.0007202 ***
Image                1     30  752.62     0.8022  0.00697  0.3775568
Age:Image            1     30  752.62     0.2152  0.00188  0.6460346
Word:Image           1     30  573.74    61.4309  0.29074  9.553e-09 ***
Age:Word:Image       1     30  573.74    73.9247  0.33033  1.363e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```
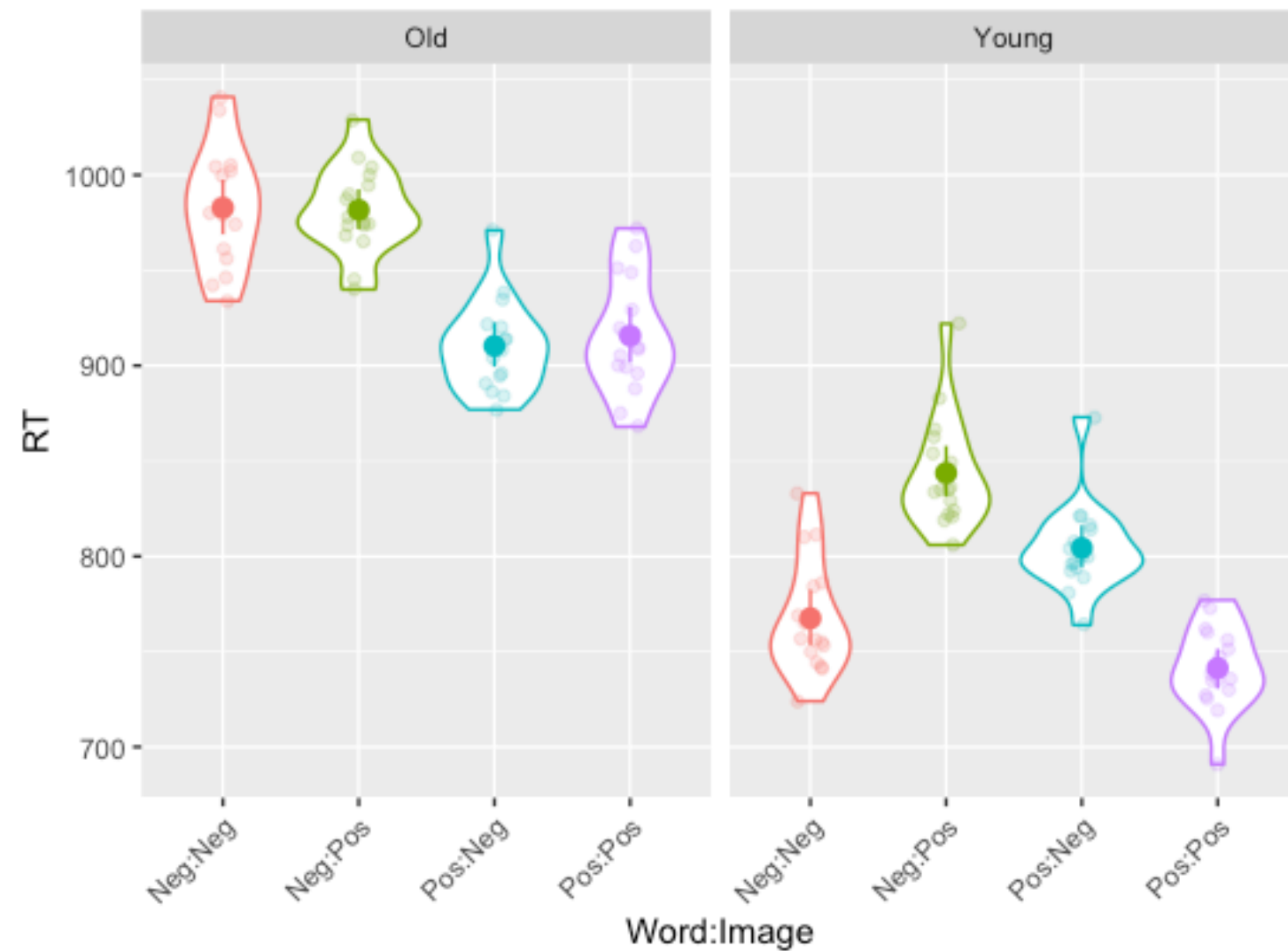
From this we can see we have main effects of Age and Word, no main effect of Image, significant 2-way interactions of Age x Word, and of Word x Image and, crucially, a 3-way interaction between all three factors - Age x Word x Image…

The 3-way interaction suggests that the Word x Image interaction is different for Young vs. Old people (which is supported by what we see in the graph…)

To interpret this 3-way, we should examine the Word x Image interaction separately for Young and Old people…

We can do this by filtering our dataset:

```
> young_filter <- filter(my_data, Age == "Young")
> model_young <- aov_4(RT ~ Word * Image + (1 + Word * Image | Participant), data =
young_filter)
> anova(model_young)
Anova Table (Type 3 tests)

Response: RT
           num Df den Df      MSE          F      ges      Pr(>F)
Word             1      15 681.87   25.1197 0.29236 0.0001547 ***
Image            1      15 918.44    0.7574 0.01650 0.3978527
Word:Image       1      15 560.77 138.1887 0.65146 5.725e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Significant
interaction

We need to follow this up with pairwise comparisons.

```
> emmeans(model_young, pairwise ~ Word * Image, adjust = "Bonferroni")
$emmeans
 Word Image emmean   SE    df lower.CL upper.CL
 Neg  Neg      768 6.57 57.7      754      781
 Pos  Neg      804 6.57 57.7      791      818
 Neg  Pos      844 6.57 57.7      831      857
 Pos  Pos      741 6.57 57.7      728      755

Warning: EMMs are biased unless design is perfectly balanced
Confidence level used: 0.95

$contrasts
 contrast             estimate     SE    df t.ratio p.value
 Neg,Neg - Pos,Neg      -36.9   8.81 29.7 -4.184   0.0014
 Neg,Neg - Neg,Pos      -76.2   9.62 28.3 -7.924   <.0001
 Neg,Neg - Pos,Pos       26.1 10.00 29.4   2.612   0.0842
 Pos,Neg - Neg,Pos      -39.3 10.00 29.4 -3.931   0.0028
 Pos,Neg - Pos,Pos       63.0   9.62 28.3   6.552   <.0001
 Neg,Pos - Pos,Pos      102.3   8.81 29.7 11.610   <.0001

P value adjustment: bonferroni method for 6 tests
```
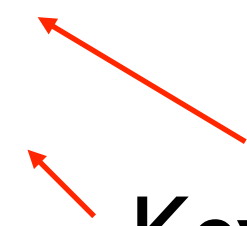
Key pairwise comparisons are significant

But what about the Old group?

```
> old_filter <- filter(my_data, Age == "Old")
> model_old <- aov_4(RT ~ Word * Image + (1 + Word * Image | Participant), data =
old_filter)
> anova (model_old)
Anova Table (Type 3 tests)

Response: RT
           num Df den Df    MSE        F       ges     Pr(>F)
Word          1      15 819.83 93.5066 0.63260 7.757e-08 ***
Image         1      15 586.81  0.1195 0.00157    0.7343
Word:Image    1      15 586.70  0.2825 0.00371    0.6028
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```
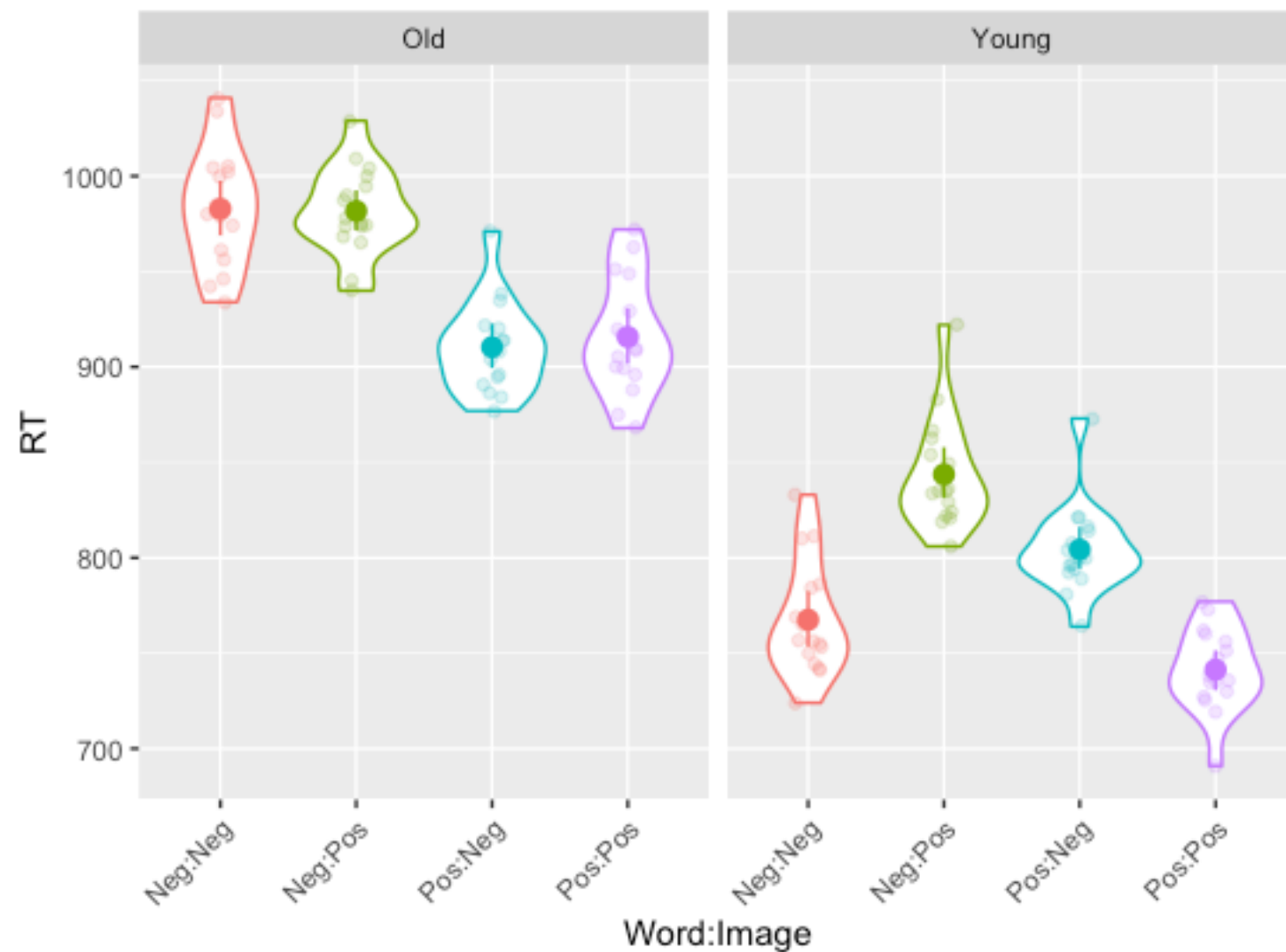
Interaction
not significant

So we have found a 2-way interaction of Word x Image that *differs* between our two groups. The 2-way interaction is significant for our Young group, but not significant for our Old group. For our Old group, we simply have a main effect of Word...

ANCOVA

- One of our examples from earlier looked at how double espresso vs. single espresso vs. water drinking (our IV) might influence motor performance (our DV).

- Imagine we sampled from a new group of participants - and we think other factors that we are not manipulating might also influence the DV – e.g., practice with computer games.

- What we want is to be able to see the effect on our DV of our IV after we have removed the effects of other things (computer gaming frequency in this case).

- Now, imagine we have a measure of computer games frequency - perhaps hours per week people play computer games…

- So, in addition to manipulating the type of beverage we're giving people (i.e., double espresso vs. single espresso vs. water) we also measure how often they play computer games…

- Let's do a plot first with our DV (Ability) on the y-axis, and our covariate (Gaming Frequency) on the x-axis…

```
> ggplot(cond, aes(x = Gaming, y = Ability, colour = Condition)) + geom_point()
```
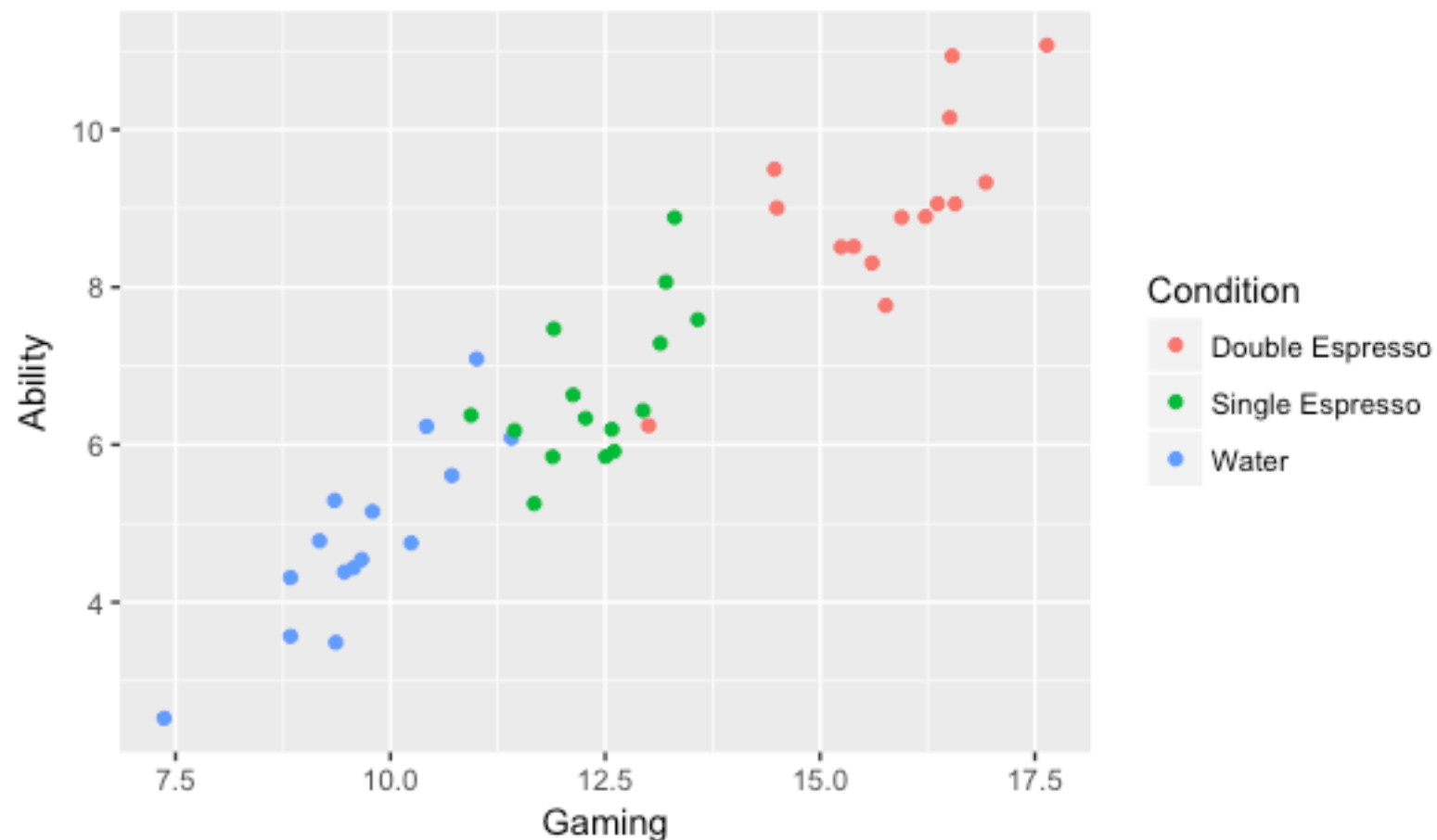

- So we can see there's a relationship between our DV (Ability) and our covariate (Gaming Frequency)…

- We can also see our Gaming Ability groups appear to be clustering in our data by Condition…

We can look at the data separately by condition using the *facet_wrap()* function:

```
> ggplot(cond, aes(x = Gaming, y = Ability, colour =
Condition)) + geom_point() + facet_wrap(~ Condition) +
geom_smooth(method = "lm")
```
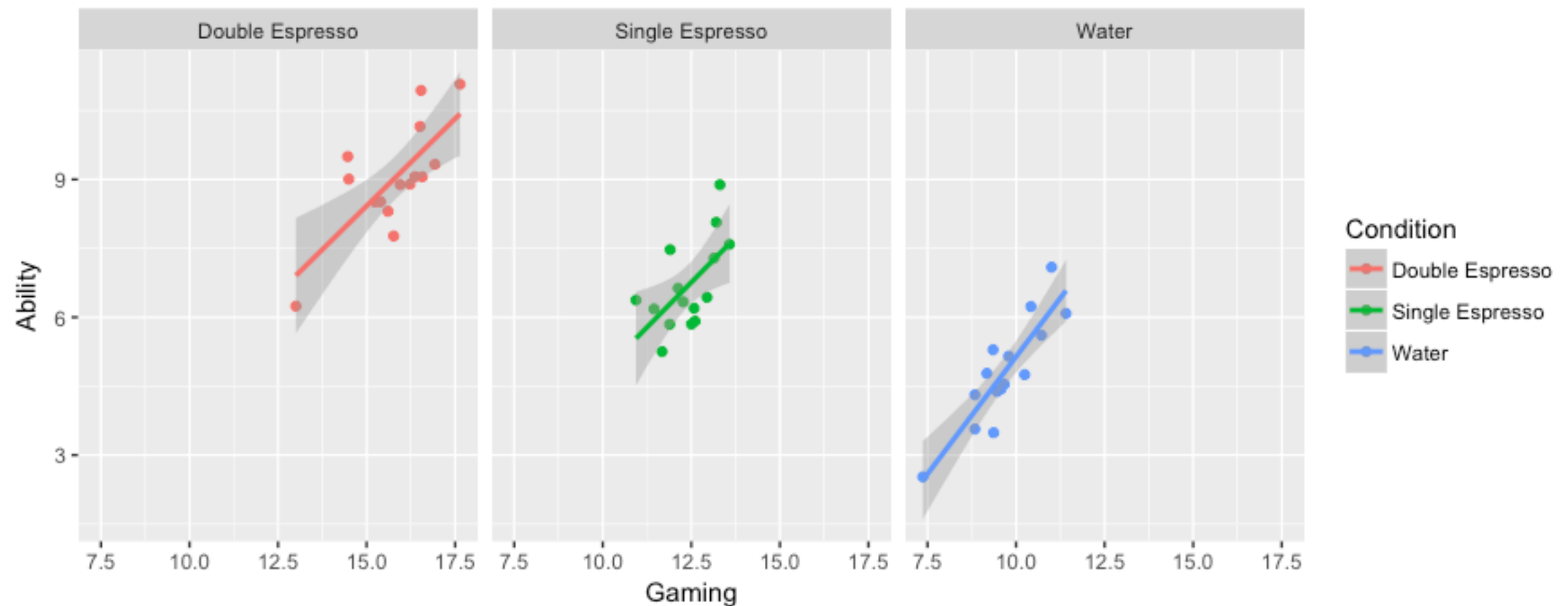
Running a 1-way between participants ANOVA (and ignoring the covariate)…

```
> model1 <- aov_4(Ability ~ Condition + (1 | Participant), data = cond)
Contrasts set to contr.sum for the following variables: Condition
> anova(model1)
Anova Table (Type 3 tests)

Response: Ability
          num Df  den Df    MSE       F       ges      Pr(>F)
Condition     2      42   1.2422  53.432  0.71786  2.882e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The factor Condition is significant with an F = 53.432. We would erroneously conclude that our manipulation has had an effect…

But now let's control for the effect of our co-variate (which we first need to scale and centre)…

```
> cond$Gaming <- scale(cond$Gaming)
> model_ancova <- aov_4(Ability ~ Gaming + Condition + (1 | Participant),
data = cond, factorize = FALSE)
Contrasts set to contr.sum for the following variables: Condition
> anova(model_ancova)
Anova Table (Type 3 tests)

Response: Ability
          num Df den Df     MSE        F      ges     Pr(>F)
Gaming         1     41 0.55171 53.5636 0.56643 5.87e-09 ***
Condition      2     41 0.55171  0.8771 0.04103   0.4236
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The factor Condition is now <u>not</u> significant with an F < 1. However, our covariate *Gaming Frequency* is significant. Adding it means a lot of the variance we previously attributed to our experimental factor is actually explained by our covariate. Note, the F values are calculated using Type III Sum of Squares by the `aov_4()` function - more on that in a bit…

Rather than calculating over the raw means which are:

Water Group = 4.82

Double Espresso Group = 9.02

Single Espresso Group = 6.69

```
> cond %>%
    group_by(Condition) %>%
    summarise(mean_ability = mean(Ability), sd_ability = sd(Ability))
# A tibble: 3 x 3
  Condition        mean_ability sd_ability
  <chr>                   <dbl>      <dbl>
1 Double Espresso          9.02       1.19
2 Single Espresso          6.69       0.977
3 Water                    4.82       1.16
```

The calculation is performed over the *adjusted* means (which take into consideration the influence of the covariate):

Water Group = 7.33

Double Espresso Group = 6.32

Single Espresso Group = 6.87

```
> emmeans(model_ancova, pairwise ~ Condition, adjust = "none")
$emmeans
 Condition          emmean        SE df lower.CL upper.CL
 Double Espresso  6.319464 0.4152816 41 5.480786 7.158142
 Single Espresso  6.871614 0.1934303 41 6.480974 7.262255
 Water            7.327960 0.3931110 41 6.534056 8.121864

Confidence level used: 0.95
```

If our experimental factor in the ANCOVA *had* been significant, we could have looked at the pairwise comparisons reported by *emmeans* to determine what condition was different from what other condition…

```
$contrasts
 contrast                               estimate        SE df t.ratio p.value
 Double Espresso - Single Espresso   -0.5521505 0.4779448 41  -1.155  0.2547
 Double Espresso - Water             -1.0084959 0.7614421 41  -1.324  0.1927
 Single Espresso - Water             -0.4563454 0.4179276 41  -1.092  0.2812
```

But once we take account of the influence of our covariate we found no effect of Condition…

Note, if we had used the `aov()` function the F-tests would have been conducted using Type 1 (sequential) Sums of Squares. For Type III, we need to use the `aov_4()` function.

Type I vs. II vs. III Sums of Squares

- Type I Sum of Squares is calculated sequentially - e.g., first for Factor A main effect, then for Factor B main effect, then for the interaction. The order in which they are calculated matters and can be misleading for unbalanced design or cases where predictors are correlated. Total SS is the sum of the individual effect SS.

- Type II Sum of Squares assumes no interaction(s) when testing main effects or higher order interaction(s) when testing lower order interaction(s).

- Type III Sum of Squares tests for effects adjusted for the presence of the other effects (so does not depend on the order of terms).

- Much debate about which one is 'correct' - each has their own purpose - for factorial designs where you're interested in testing an interaction (or when your predictors correlate), Type III is most commonly used.

AN(C)OVA as a special case of regression…

- Let's return to the example we looked at for ANCOVA - and let's forget the co-variate for a moment…

- We looked at how double espresso vs. single espresso vs. water drinking (our IV) might influence people's gaming ability (our DV).
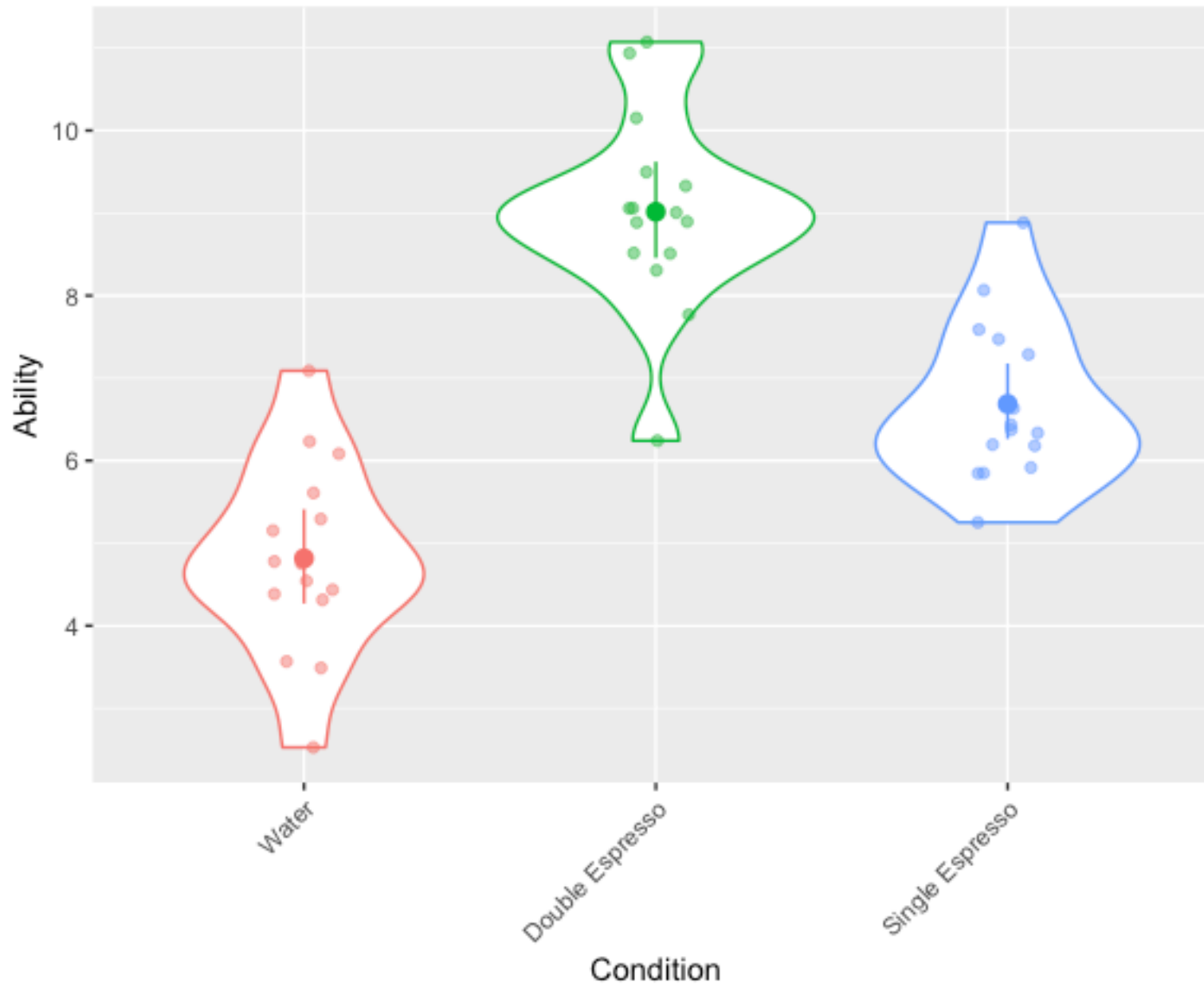
Water mean = 4.82
Double Espresso mean = 9.02
Single Espresso mean = 6.69

- First we need to use dummy coding of the levels of our experimental factor - which is the default coding in R for factors…

```
> # Set up the Water level as the reference level and check the contrasts
> cond$Condition <- relevel(cond$Condition, ref = 3)
> contrasts(cond$Condition)
                Double Espresso Single Espresso
Water                         0               0
Double Espresso               1               0
Single Espresso               0               1
```

Ability = Intercept + $\beta 1$(Double Espresso) + $\beta 2$(Single Espresso) + ε

The Intercept is our reference category (Water) with coding (0, 0), while the dummy coding for Double Espresso is (1, 0) and for Single Espresso (0, 1)

Ability = Intercept + $\beta1$(Double Espresso) + $\beta2$(Single Espresso) + ε

We want to calculate $\beta1$ and $\beta2$

```
> lm1 <- lm(Ability ~ Condition, data = cond)
> lm1

Call:
lm(formula = Ability ~ Condition, data = cond)

Coefficients:
            (Intercept)   ConditionDouble Espresso   ConditionSingle Espresso
                  4.817                      4.199                      1.871
```

The intercept is 4.817 (which is the mean of our Water group), $\beta1$ is 4.2, and $\beta2$ is 1.87

To work out the mean Ability of our Double Espresso Group:

Ability = Intercept + $\beta 1$(Double Espresso) + $\beta 2$(Single Espresso) + ε

Ability = 4.82 + 4.2(1) + 1.87(0) + ε

Ability = 4.82 + 4.2 + ε

Ability = 9.02 + ε

To work out the mean Ability of our Single Espresso Group:

Ability = Intercept + $\beta1$(Double Espresso) + $\beta2$(Single Espresso) + ε

Ability = 4.82 + 4.2(0) + 1.87(1) + ε

Ability = 4.82 + 1.87 + ε

Ability = 6.69 + ε

Which are the exact same means generated by the ANOVA…

Water mean = 4.82
Double Espresso mean = 9.02
Single Espresso mean = 6.69

We can do ANCOVA like this too - let's consider our co-variate of Gaming frequency…

The *adjusted* means from the ANCOVA (which take into consideration the influence of the covariate) were:

Water Group = 7.33

Double Espresso Group = 6.32

Single Espresso Group = 6.87

Ability = Intercept + $\beta 1$(Gaming) + $\beta 2$(Double Espresso) + $\beta 3$(Single Espresso) + ε

Add the covariate to our model *before* the experimental factor:

```
> lm2 <- lm(Ability ~ Gaming + Condition, data = cond)
> lm2

Call:
lm(formula = Ability ~ Gaming + Condition, data = cond)

Coefficients:
            (Intercept)                      Gaming  ConditionDouble Espresso  ConditionSingle Espresso
                -3.4498                      0.8538                   -1.0085                   -0.4563
```

Ability = Intercept + $\beta 1$(Gaming) + $\beta 2$(Double Espresso) + $\beta 3$(Single Espresso) + ε

The $\beta 2$ and $\beta 3$ coefficients tell us the difference between each group mean (i.e., the adjusted mean) compared to the reference Group (Water) when taking into account the covariate of Gaming frequency:

$\beta 2$ is the difference between the Double Espresso and Water group adjusted means (= -1.01) while $\beta 3$ is the difference between the Double Espresso and Water group adjusted means (= -0.46)…

Let's check - the following are the adjusted means output by the ANCOVA model:

Water Group = 7.33

Double Espresso Group = 6.32

Single Espresso Group = 6.87

Difference between the Water and Double Espresso Group is 1.01 and the difference between the Water and Single Espress Group is 0.46…

We can work out the mean of our reference group (Water) by plugging in the values to our equation - note that Gaming is not a factor and we need to enter the mean of this variable (which is 12.62296). So,…

Ability = Intercept + $\beta1$(Gaming) + $\beta2$(Double Espresso) + $\beta3$(Single Espresso) + ε

Ability = -3.4498 + 0.8538(12.62296) + (- 1.0085)(0) + (-0.4563)(0) + ε

Ability = -3.4498 + 10.777 + ε

Ability = 7.33 + ε

7.33 is the adjusted mean for the Water group…which is what we had from calling the `emmeans` function following the ANCOVA…

You can now build ANOVA models in R for different kinds of designs, add between participant co-variates, factor out the influence of these co-variates, and you also know why AN(C)OVA is a special case of regression (with dummy coding of variables)…

To ANOVA_part_2_lab_script_question…