Lecture 7 - ANOVA part 1

Andrew Stewart

Andrew.Stewart@manchester.ac.uk

@ajstewart_lang

Session	Topic	Lecturer
1	Introduction, Open Science, and Power	Andrew Stewart
2	Introduction to R	Andrew Stewart
3	Data Wrangling and Visualisation	Andrew Stewart
4	General Linear Model - Regression	Andrew Stewart
5	General Linear Model - Regression	Andrew Stewart
6	Consolidation Lab	Bo Yao
7	General Linear Model - ANOVA	Andrew Stewart
8	General Linear Model - ANOVA	Andrew Stewart
9	Signal Detection Theory	Ellen Poliakoff
10	Signal Detection Theory	Ellen Poliakoff
11	Revision Session	Andrew Stewart

Semester 1 Assignments

ANOVA – Due start December

Signal Detection Analysis – Due around mid-January

- We're going to have our first look at the Analysis of Variance (ANOVA).
- This week 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.
- It assumes that our data are parametric.

Assessment

- The assessment will be on the ANOVA lectures. It will require you to conduct an ANOVA and to produce a report using R Markdown - we'll cover that next week.

- The assessment question will be of a similar type to the ones we'll look at in the lab classes over the next couple of weeks.

Reporting ANOVA

- Say what type of ANOVA it was, say what factors you had (and with labels for each level).
- Report the results of main effects first, then interactions.
- Report F values, exact *p*-values and effect size values.
- Remember to interpret interactions further - either with further ANOVA or pairwise comparisons.
- When you have main effects, say which direction the effect goes.
- Avoid sillies - e.g., mixing up < and > or saying p = .000

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 familywise error rate.

$$\text{familywise error} = 1 – (0.95)^n$$

- 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 not 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 worse 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 ?

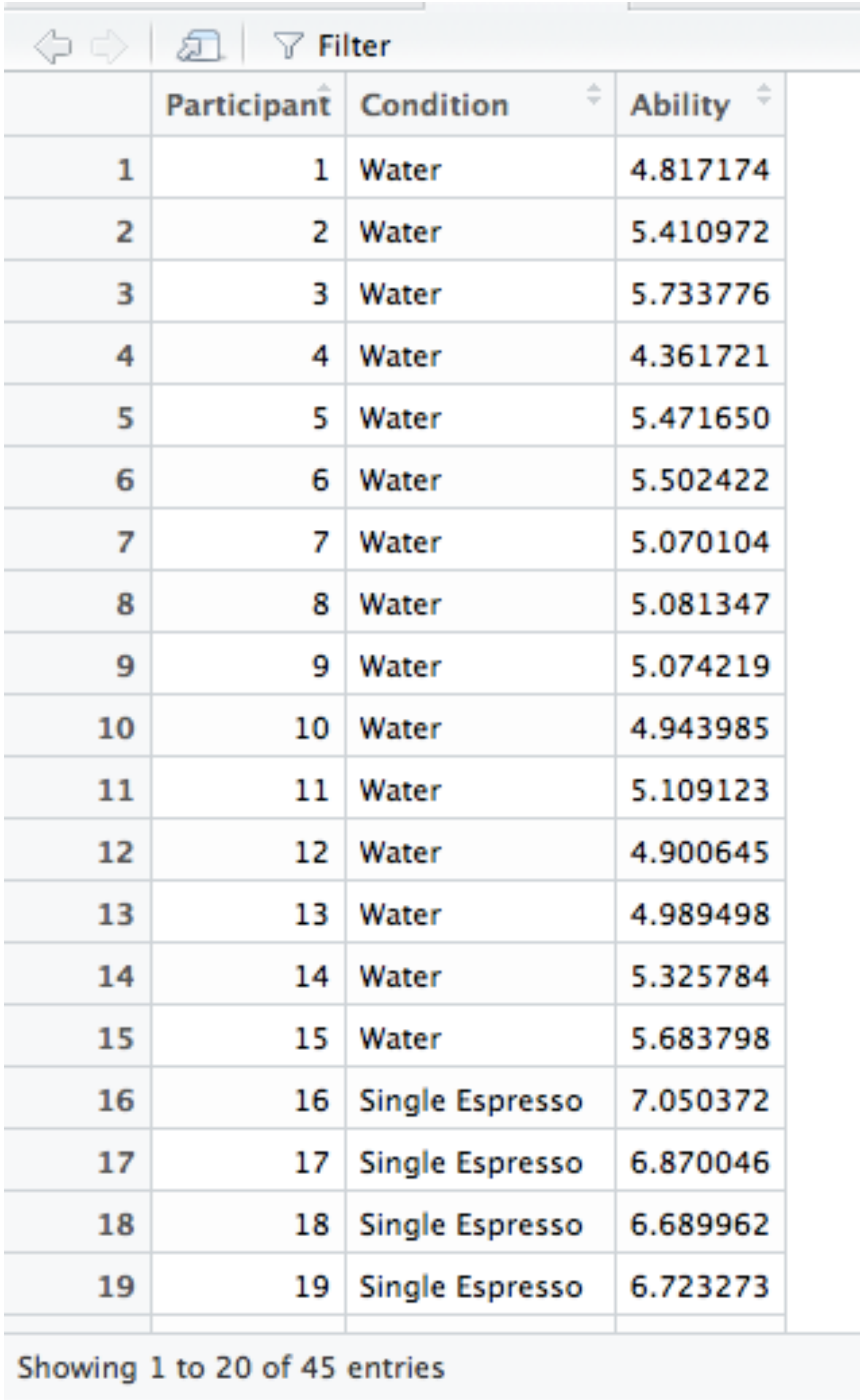
Is it reliable when ANOVA assumptions have been violated ?

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.

Our data look like this:

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 scale of 1-10.



	Participant	Condition	Ability
1	1	Water	4.817174
2	2	Water	5.410972
3	3	Water	5.733776
4	4	Water	4.361721
5	5	Water	5.471650
6	6	Water	5.502422
7	7	Water	5.070104
8	8	Water	5.081347
9	9	Water	5.074219
10	10	Water	4.943985
11	11	Water	5.109123
12	12	Water	4.900645
13	13	Water	4.989498
14	14	Water	5.325784
15	15	Water	5.683798
16	16	Single Espresso	7.050372
17	17	Single Espresso	6.870046
18	18	Single Espresso	6.689962
19	19	Single Espresso	6.723273

Showing 1 to 20 of 45 entries

First we need to load the packages we're going to use:

```
library(tidyverse) #load the tidyverse packages
library(psych) #load the psych packages for generating descriptives
library(yarrr) #load yarrr for pirate plots
library(afex) #load afex for running factorial ANOVA
library(DescTools) #load DescTools for calculating effect sizes
library(emmeans) #load emmeans for running pairwise comparisons
```

If you haven’t installed a package previously, remember to type

```
>install.packages(“packagename”)
```

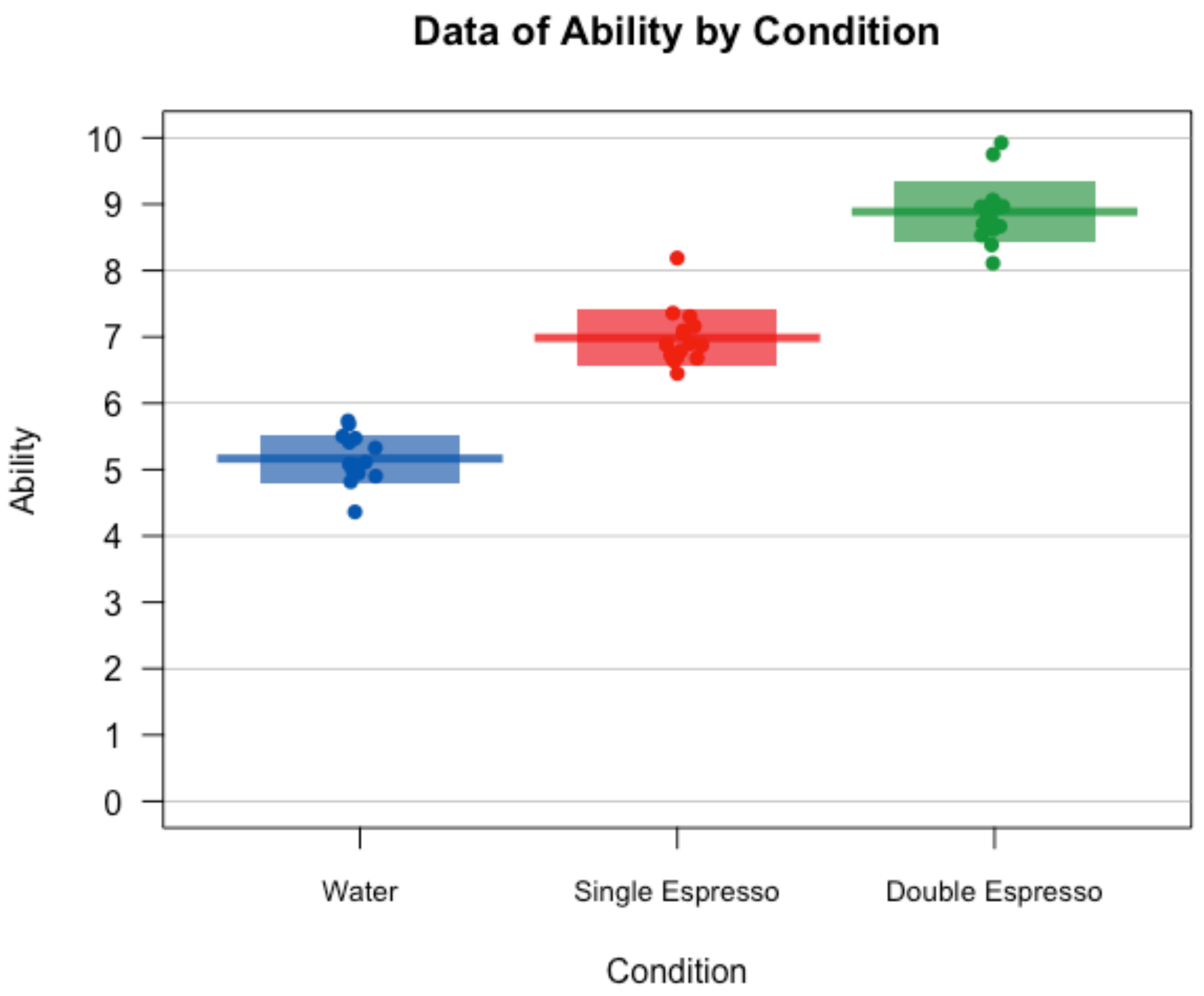
first.

Our data frame is called `cond` and has the following structure:

```
> str(cond)
'data.frame':   45 obs. of  3 variables:
 $ Participant: num  1 2 3 4 5 6 7 8 9 10 ...
 $ Condition  : Factor w/ 3 levels "Water","Double Espresso",..: 1 1 1 1 1 1 
1 1 1 1 ...
 $ Ability    : num  4.82 5.41 5.73 4.36 5.47 …

> head (cond)
  Participant Condition  Ability
1           1     Water 4.817174
2           2     Water 5.410972
3           3     Water 5.733776
4           4     Water 4.361721
5           5     Water 5.471650
6           6     Water 5.502422
```

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.

Let’s visualise the data first



Now some descriptives…

We’re going to do this by using the *describeBy* function in the *Psych* package.

```
> describeBy(cond$Ability, group = cond$Condition)
```

```
> describeBy (cond$Ability, group = cond$Condition)

 Descriptive statistics by group
group: Water
   vars  n mean   sd median trimmed  mad  min  max range  skew kurtosis   se
X1    1 15 5.17 0.36   5.08    5.18 0.36 4.36 5.73  1.37 -0.27    -0.49 0.09
-----------------------------------------------------------------------------------
group: Single Espresso
   vars  n mean   sd median trimmed mad  min  max range skew kurtosis   se
X1    1 15 6.99 0.42   6.88    6.93 0.3 6.45 8.19  1.74  1.4     1.83 0.11
-----------------------------------------------------------------------------------
group: Double Espresso
   vars  n mean   sd median trimmed  mad  min  max range skew kurtosis   se
X1    1 15 8.89 0.47   8.85    8.87 0.31 8.11 9.92  1.81 0.72     0.05 0.12
```

Or alternatively using functions from the `dplyr` package:

```
> cond %>% group_by(Condition) %>% summarise(mean = mean(Ability),
sd = sd(Ability), count = n())
# A tibble: 3 x 4
  Condition          mean    sd count
  <fct>             <dbl> <dbl> <int>
1 Water              5.17 0.362    15
2 Single Espresso    6.99 0.419    15
3 Double Espresso    8.89 0.467    15
```

Now let’s run the 1-way ANOVA using the *aov* function (part of base R). We are going to assign it to a variable we are calling *model*.

```
> model <- aov(Ability ~ Condition, data = cond)
> anova(model)
Analysis of Variance Table

Response: Ability
          Df  Sum Sq Mean Sq F value    Pr(>F)    
Condition  2 103.872  51.936  297.05 < 2.2e-16 ***
Residuals 42   7.343   0.175                      
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Here’s the output we get – the F value is the ratio of systematic variance to unsystematic variation. It is the Mean SS of Condition divided by Mean Residual SS.

To get the Mean Square values we divide the Sum of Squares by the associated degrees of freedom (e.g., 7.343 / 42 = 0.175).

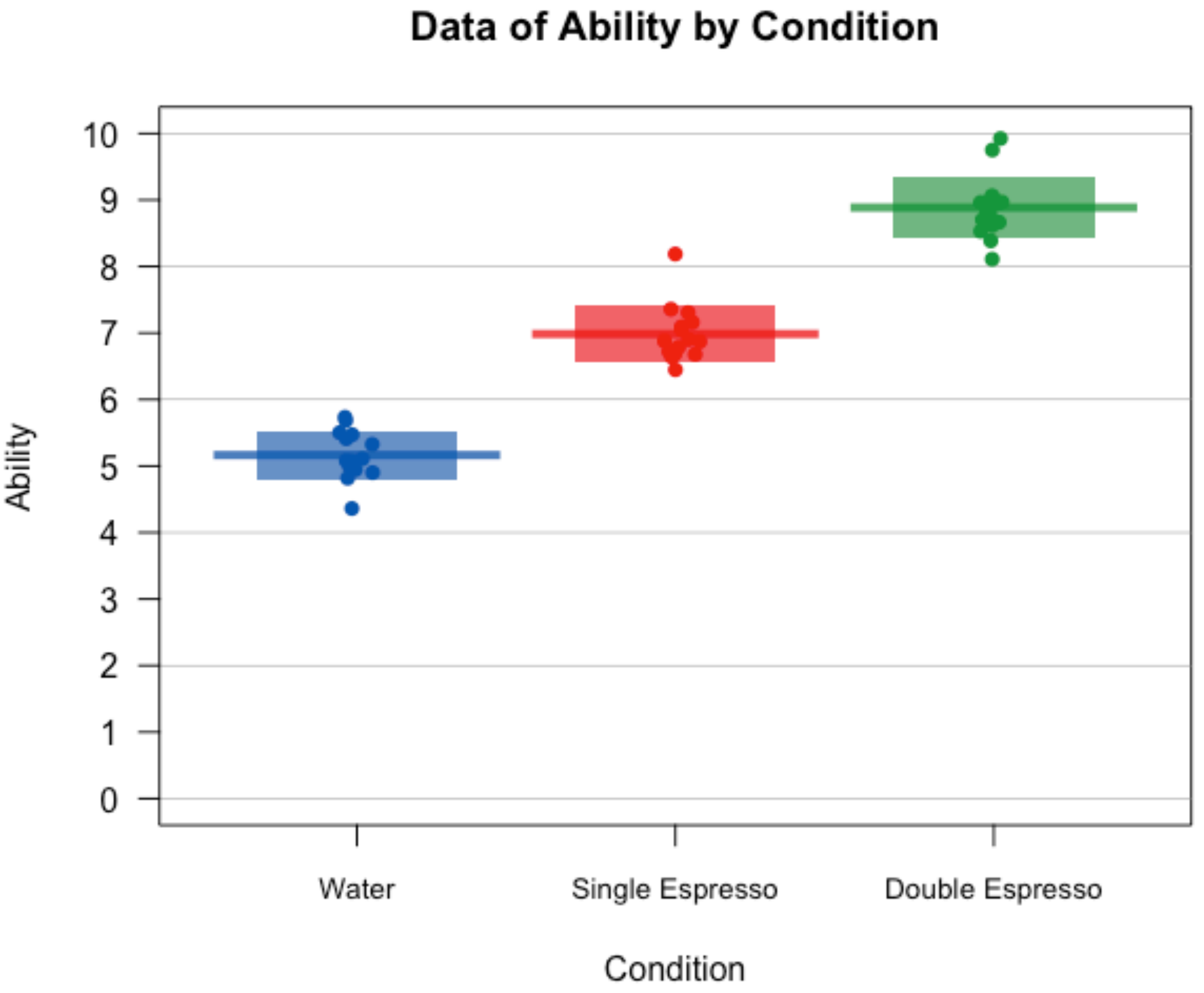
The ANOVA tells us we have an effect somewhere of Condition, but we don’t yet know which level of this factor differs from which other level(s).

We need to conduct post hoc tests to figure this out. We can conduct both Bonferroni and Tukey pairwise comparisons using the *emmeans* function - Bonferroni is slightly more conservative than Tukey.

```
> emmeans(model, pairwise ~ Condition, adjust = “Bonferroni")
$emmeans
 Condition         emmean        SE df lower.CL upper.CL
 Water           5.165081 0.1079627 42 4.947204 5.382959
 Single Espresso 6.985001 0.1079627 42 6.767124 7.202879
 Double Espresso 8.886287 0.1079627 42 8.668409 9.104164

Confidence level used: 0.95

$contrasts
 contrast                           estimate        SE df t.ratio p.value
 Water - Single Espresso           -1.819920 0.1526824 42 -11.920  <.0001
 Water - Double Espresso           -3.721205 0.1526824 42 -24.372  <.0001
 Single Espresso - Double Espresso -1.901285 0.1526824 42 -12.453  <.0001

P value adjustment: bonferroni method for 3 tests
```

```
> emmeans(model, pairwise ~ Condition, adjust = “Tukey")
$emmeans
 Condition         emmean        SE df lower.CL upper.CL
 Water           5.165081 0.1079627 42 4.947204 5.382959
 Single Espresso 6.985001 0.1079627 42 6.767124 7.202879
 Double Espresso 8.886287 0.1079627 42 8.668409 9.104164

Confidence level used: 0.95

$contrasts
 contrast                          estimate        SE df t.ratio p.value
 Water - Single Espresso          -1.819920 0.1526824 42 -11.920  <.0001
 Water - Double Espresso          -3.721205 0.1526824 42 -24.372  <.0001
 Single Espresso - Double Espresso -1.901285 0.1526824 42 -12.453  <.0001

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

We could set `adjust = “none”` if we wanted uncorrected *p*-values. But in this case, both Bonferroni and Tukey comparisons tell us the same thing - each condition differs from each other condition (which fits with what we saw in the graph).


Data of Ability by Condition
Ability
10
9
8
7
6
5
4
3
2
1
0
Water
Single Espresso
Double Espresso
Condition

Measure of Effect Size

- Effect size measures tell us how much variance can be explained by our experimental factors.
- partial η2 is a correlation between the dependent variable and different levels of a factor.
- 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).

```
> EtaSq(model, type = 3, anova = TRUE)
              eta.sq eta.sq.part         SS df         MS        F  p
Condition 0.93397251   0.9339725 103.871817  2 51.9359084 297.0494  0
Residuals 0.06602749          NA   7.343252 42  0.1748393       NA NA
```

So, to make sense of our output

- We found a significant effect of Beverage type (F (2,42) = 297.05, p < .001, partial 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 factorial designs

- A particularly good package for factorial ANOVA is by Henrik Singmann and called `afex`.
- Built to work like ANOVA in SPSS - uses Type III Sums of Squares with *effect* coding of contrasts. This overrides the default contrast coding in *R* which is for *dummy* coding.

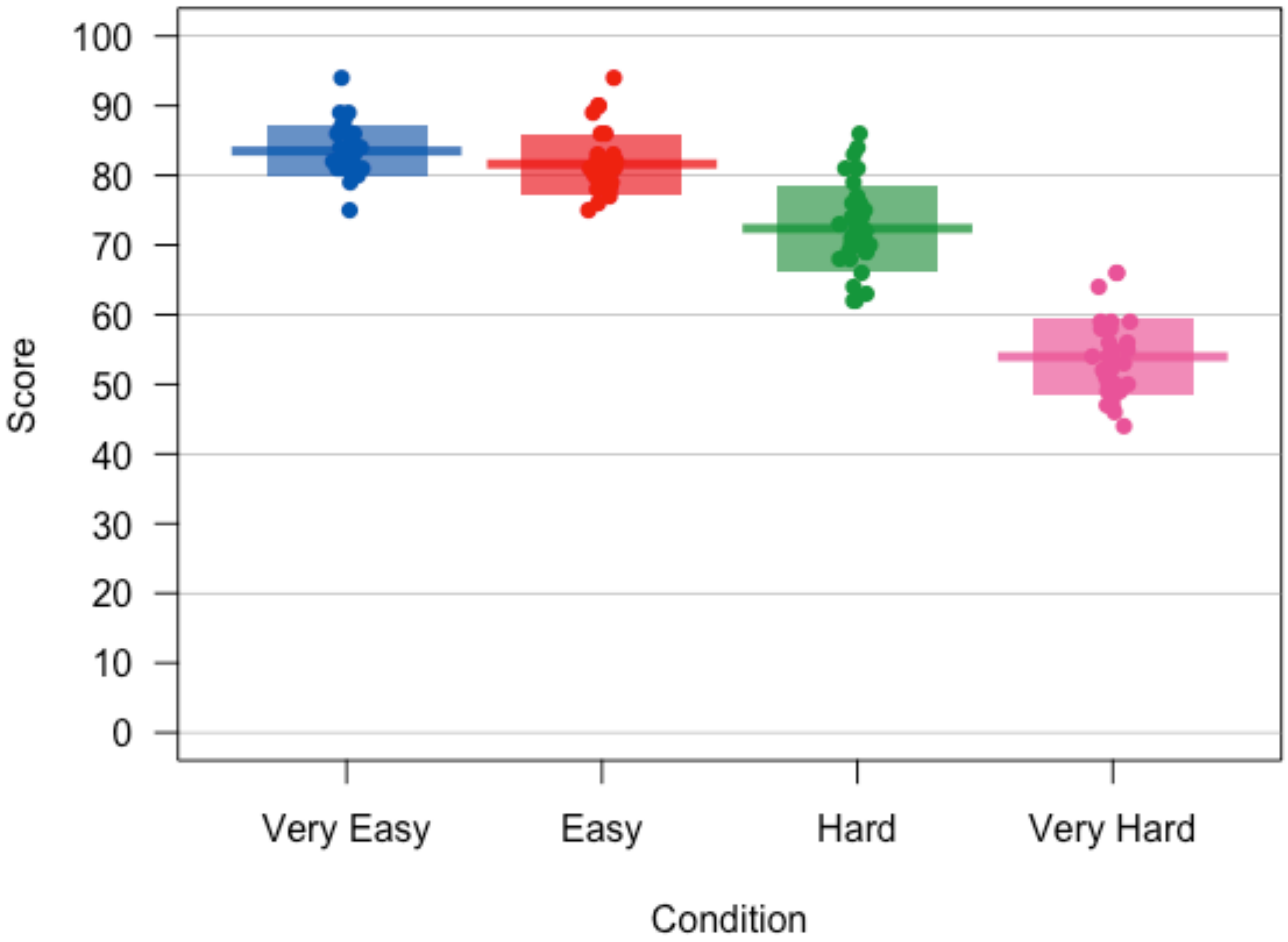
Repeated measures example - 1 Factor, 4 levels

- 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.

Our data are in tidy format with three columns - Participant, Condition, and Score and each row corresponding to one observation. We can use the `nrow()` function to find our how many rows we have:

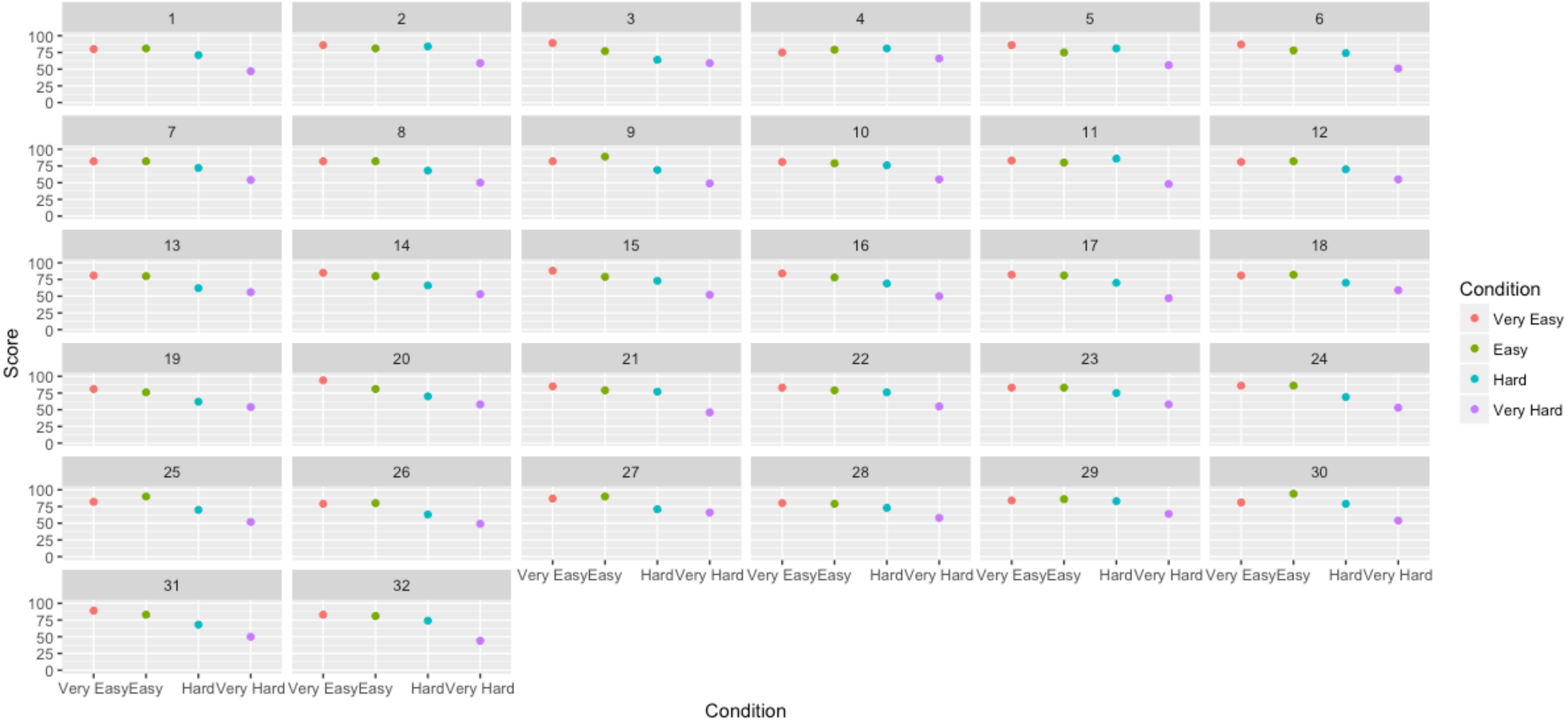
```
> head(data)
# A tibble: 6 x 3
  Participant Condition Score
  <chr>       <fct>     <int>
1 1           Very Easy    80
2 2           Very Easy    86
3 3           Very Easy    89
4 4           Very Easy    75
5 5           Very Easy    86
6 6           Very Easy    87

> nrow(data)
[1] 128
```

Let’s visualise the data first



We can use the *facet_wrap* function with *ggplot* to plot separate graphs for each participant on the same page:

```
> ggplot (data, aes (Condition, Score, colour =
Condition)) + ylim(0,100) + geom_point() +
facet_wrap(~ data$Participant)
```



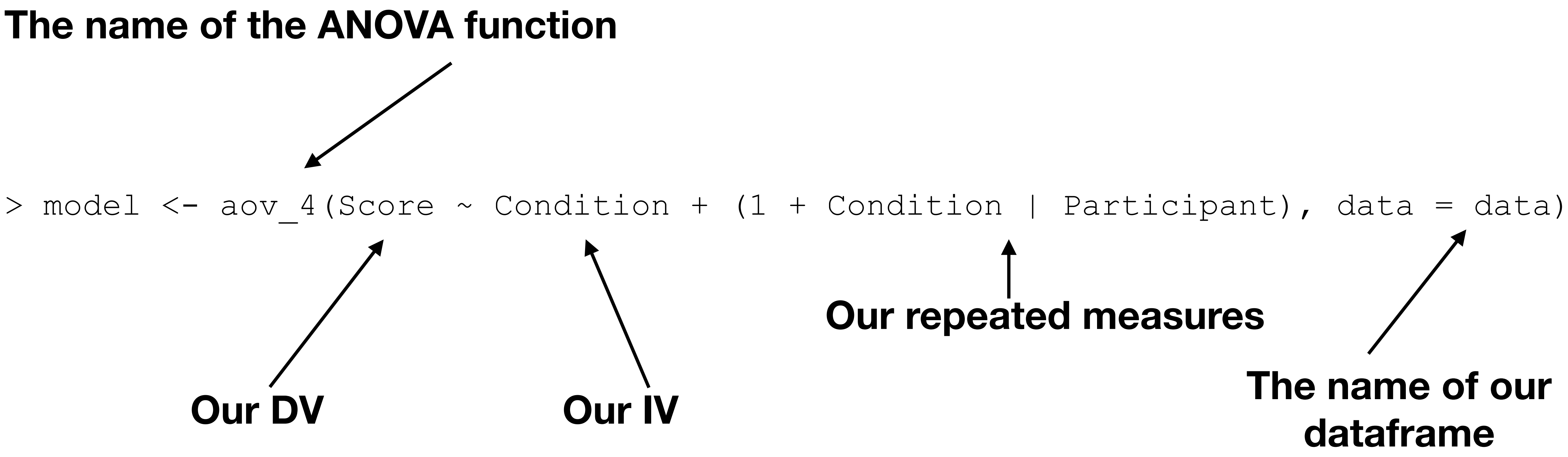
Now some descriptives…

We’re going to do this by using the *describeBy* function in the *Psych* package.

```
> describeBy (data$Score, group = data$Condition)

 Descriptive statistics by group
group: Very Easy
   vars  n mean   sd median trimmed  mad min max range skew kurtosis   se
X1    1 32 83.5 3.62     83   83.31 2.97  75  94    19 0.54     0.83 0.64
-------------------------------------------------------------------------------------
group: Easy
   vars  n  mean   sd median trimmed  mad min max range skew kurtosis   se
X1    1 32 81.62 4.28     81   81.15 2.97  75  94    19 1.14     0.83 0.76
-------------------------------------------------------------------------------------
group: Hard
   vars  n  mean   sd median trimmed  mad min max range skew kurtosis  se
X1    1 32 72.38 6.24     71   72.15 4.45  62  86    24 0.37    -0.56 1.1
-------------------------------------------------------------------------------------
group: Very Hard
   vars  n  mean  sd median trimmed  mad min max range skew kurtosis   se
X1    1 32 53.97 5.5     54   53.62 5.93  44  66    22 0.42    -0.37 0.97
```

Building the ANOVA model

We are mapping the output of our ANOVA model onto a new variable we are calling *model*.



The name of the ANOVA function

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

Our repeated measures

Our DV

Our IV

The name of our dataframe

This is the our ANOVA model - we have a significant effect of Condition.

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

```
> 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
```

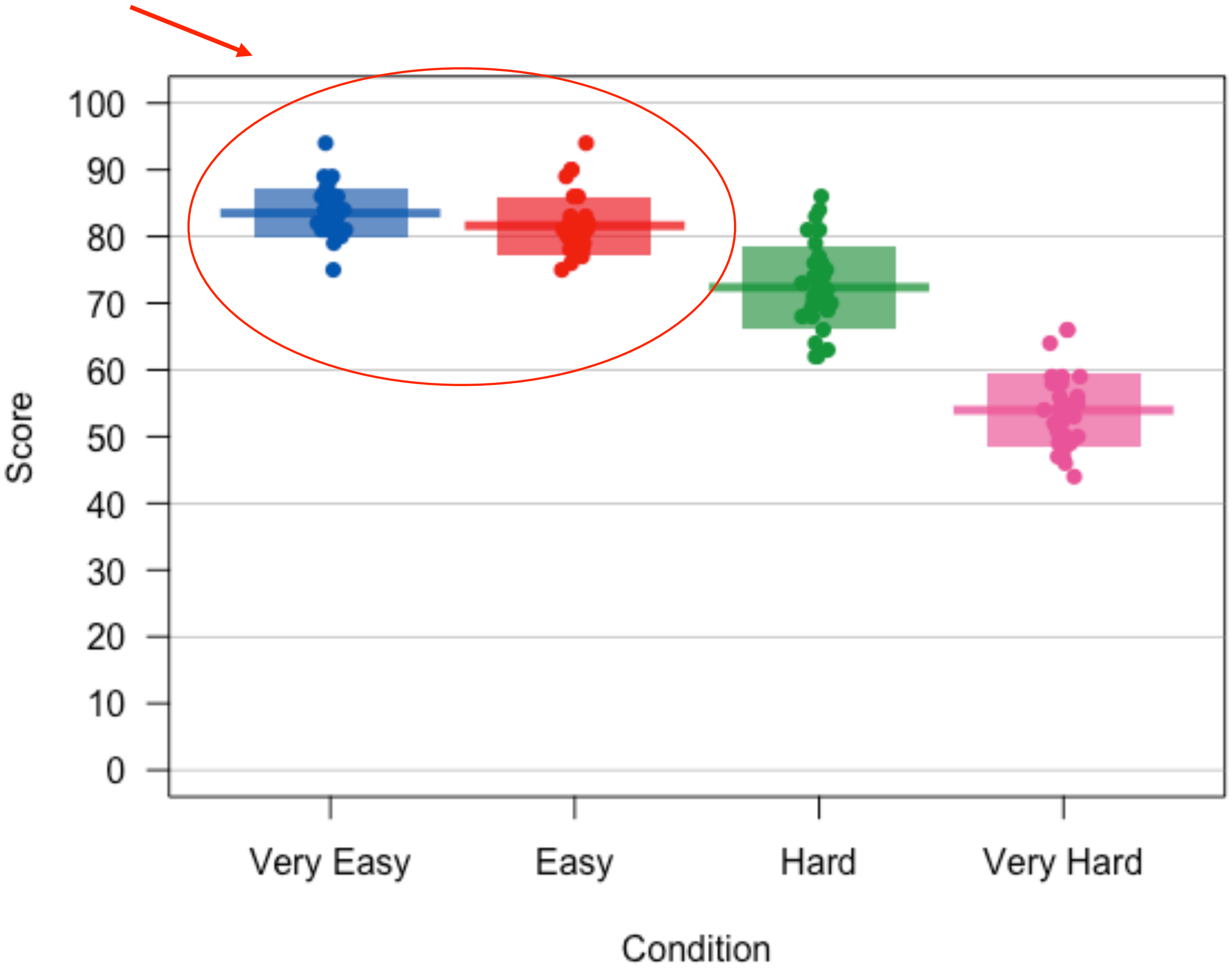
So we know we have an effect of Condition, but we don’t know where the difference lies…

Let’s do some post hoc tests with Bonferroni corrected *p*-values…

```
> emmeans(model, pairwise ~ Condition, adjust = “Bonferroni")
$emmeans
 Condition   emmean        SE     df lower.CL upper.CL
 Very.Easy 83.50000 0.8861571 122.33 81.74581 85.25419
 Easy      81.62500 0.8861571 122.33 79.87081 83.37919
 Hard      72.37500 0.8861571 122.33 70.62081 74.12919
 Very.Hard 53.96875 0.8861571 122.33 52.21456 55.72294

Confidence level used: 0.95

$contrasts
 contrast              estimate       SE df t.ratio p.value
 Very.Easy - Easy       1.87500 1.210249 93   1.549  0.7483
 Very.Easy - Hard      11.12500 1.210249 93   9.192  <.0001
 Very.Easy - Very.Hard 29.53125 1.210249 93  24.401  <.0001
 Easy - Hard            9.25000 1.210249 93   7.643  <.0001
 Easy - Very.Hard      27.65625 1.210249 93  22.852  <.0001
 Hard - Very.Hard      18.40625 1.210249 93  15.209  <.0001

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

- We see each level differs from each other, apart from Very Easy *vs*. Easy (where *p* = .75).

These two are equivalent, while other pairwise differences are significant.

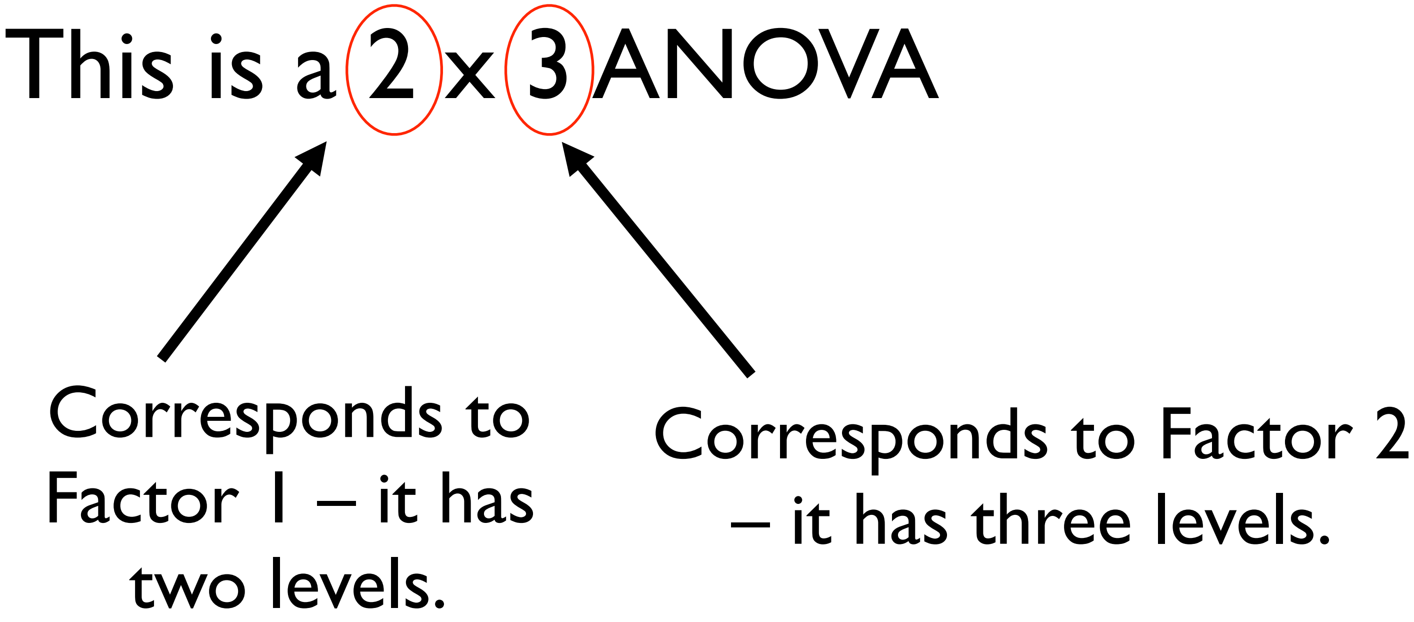


So far we have looked at ANOVA for designs when we have one factor which is between subjects (i.e., each participant appears in one condition), and for designs when we have one factor that is repeated measures (each participant appears in all conditions. These are examples of 1-way ANOVA.

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…..

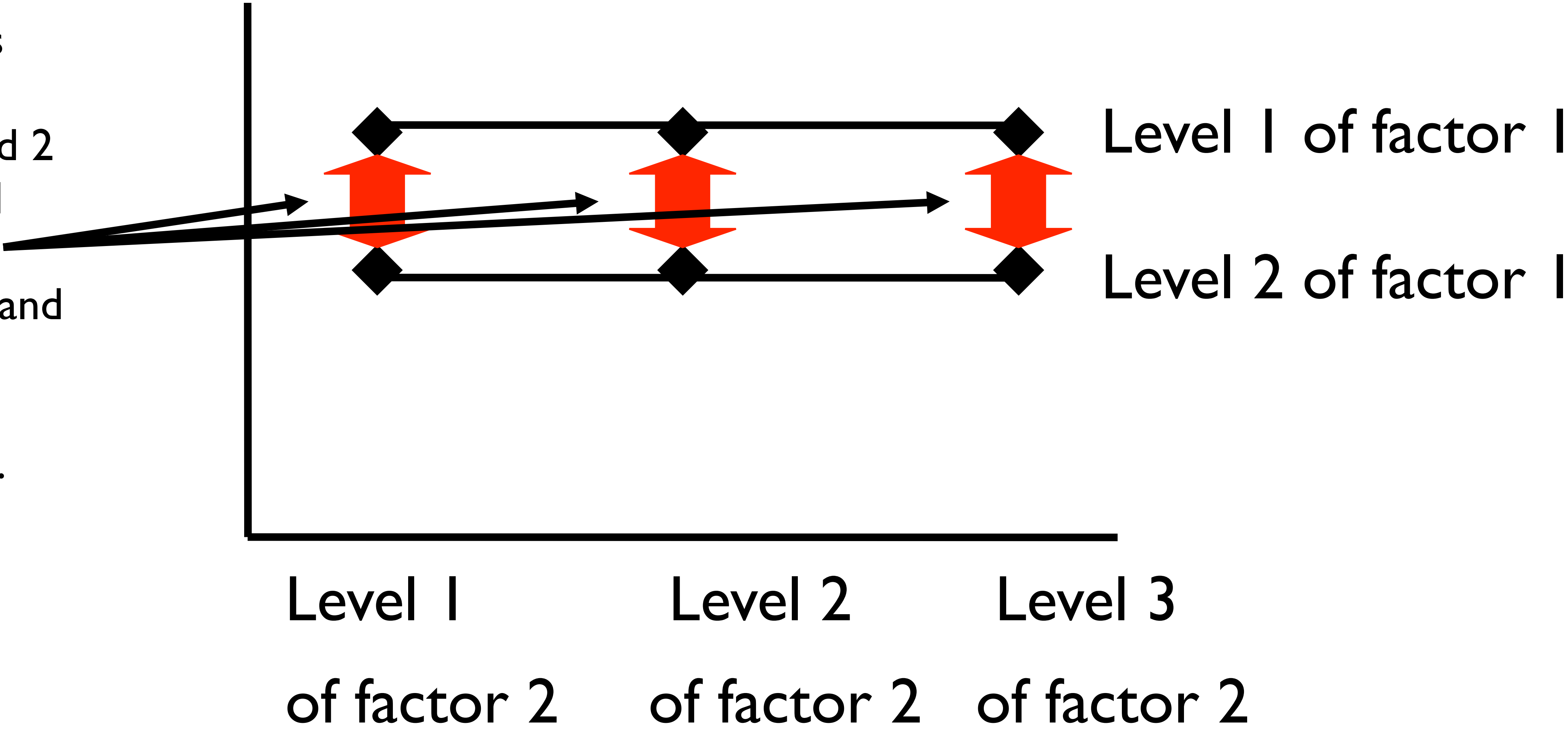
- This is a 2 x 3 ANOVA



Corresponds to Factor 1 – it has two levels.

Corresponds to Factor 2 – it has three levels.

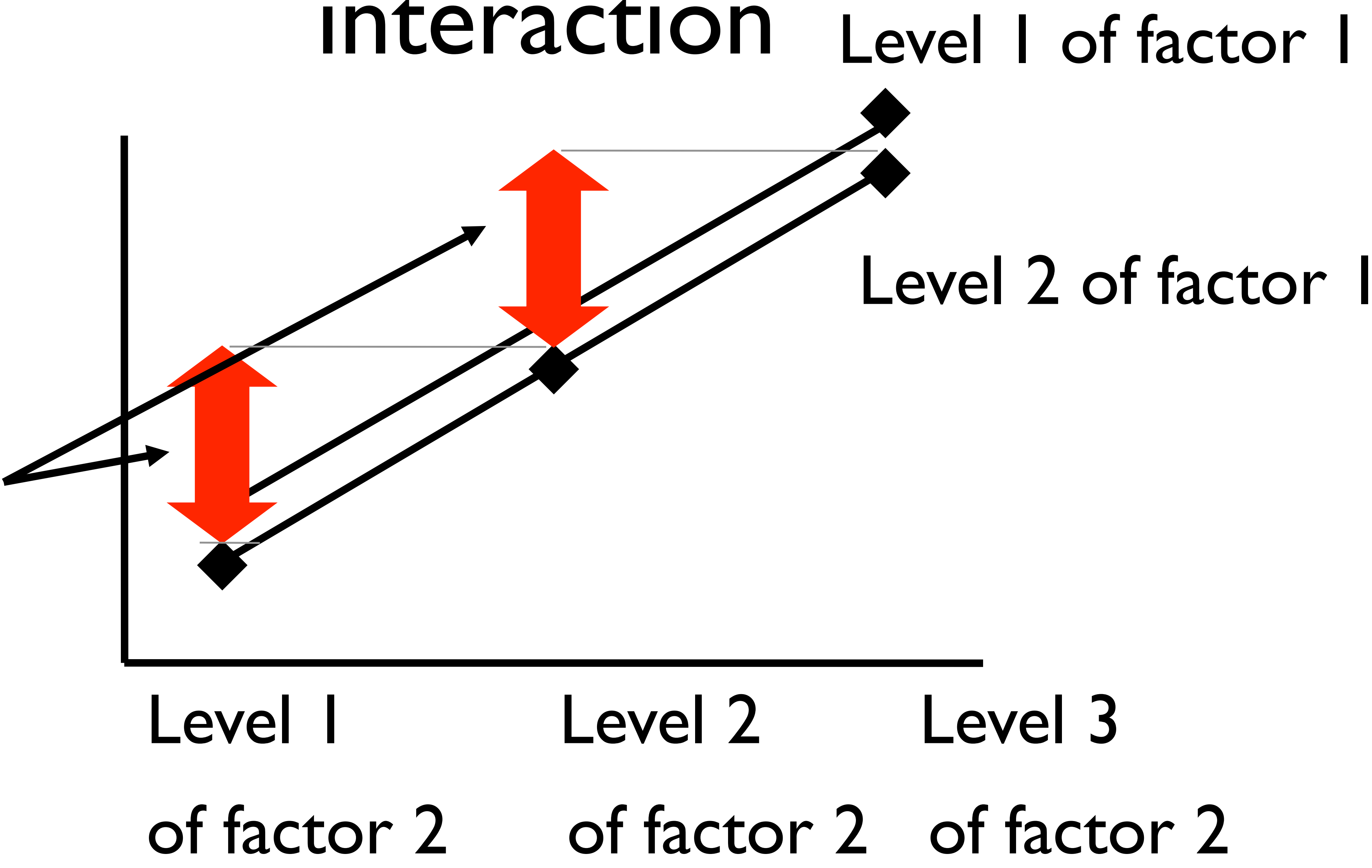
Main effect of Factor 1, no main effect of Factor 2 and no interaction



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

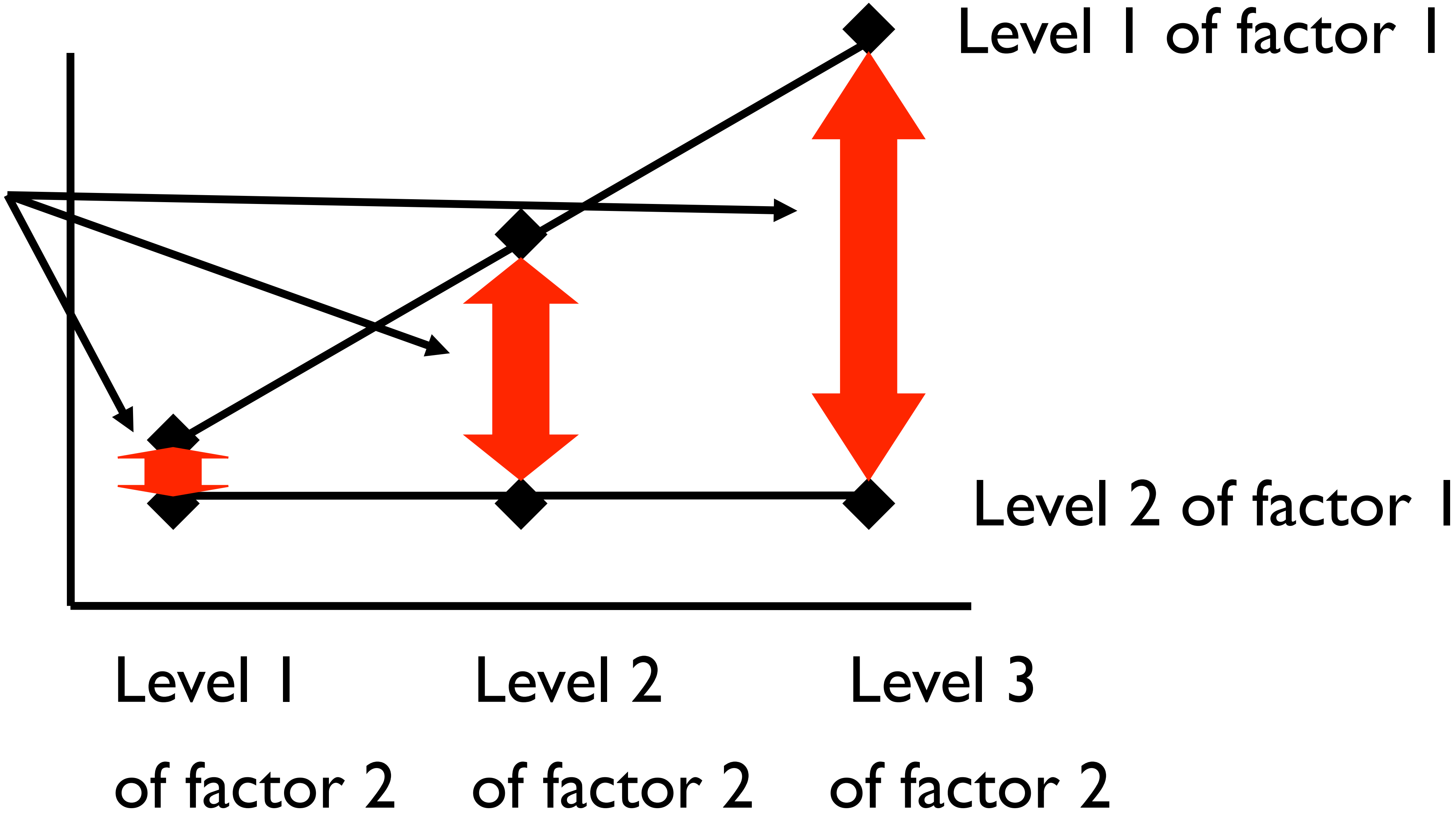
No main effect of Factor 1, main effect of Factor 2 and no interaction

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.



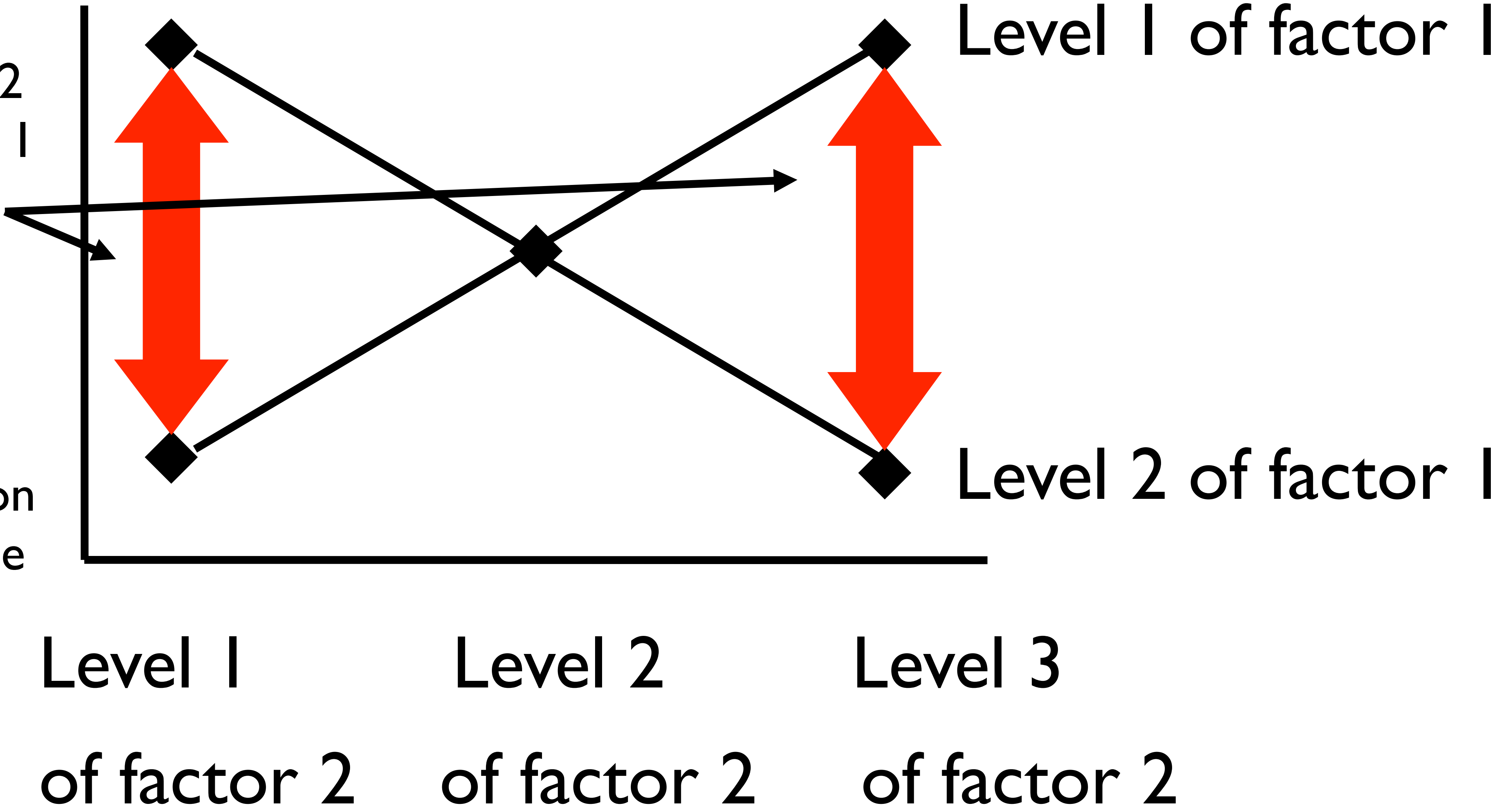
Main effect of Factor 1, main effect of Factor 2 and an interaction

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



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

2 x 2 Example

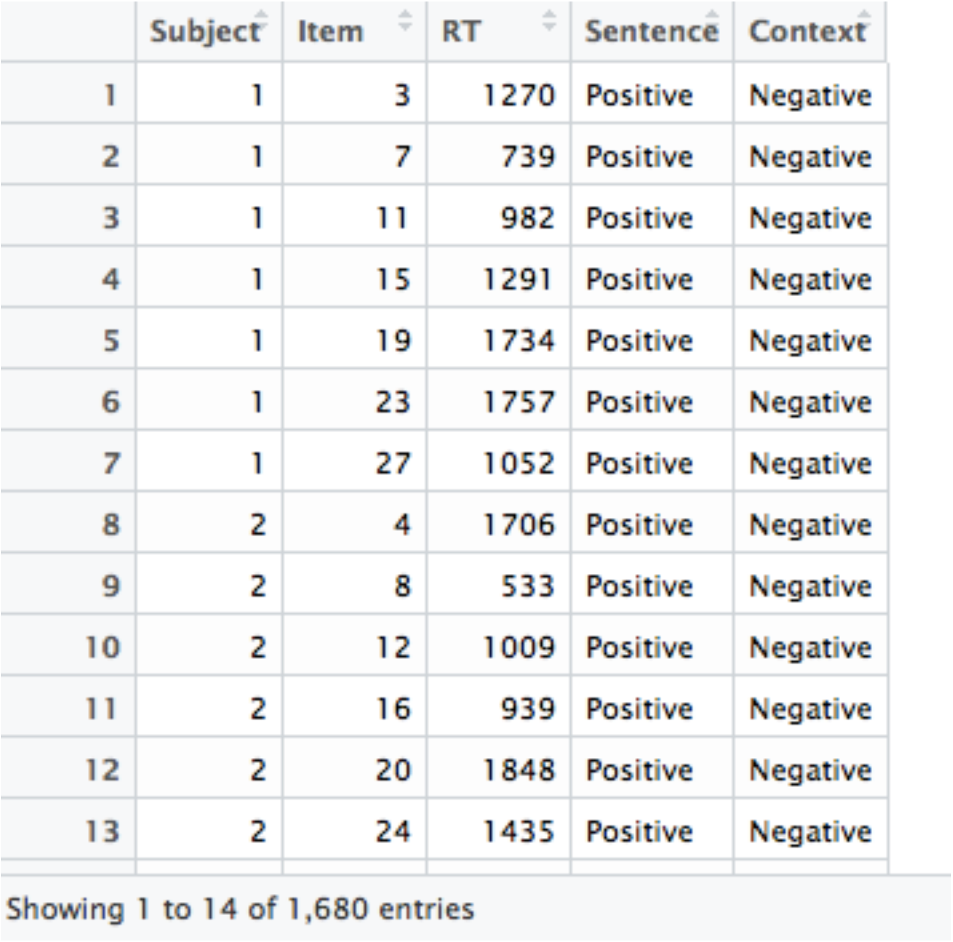
- Imagine the case where we're interested in the effect of positive vs. negative words on how quickly (in milliseconds) people respond to positive vs negative images. We think there might be a priming effect (i.e., people are quicker to respond to positive images after positive words vs. after negative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

2 x 2 Example

- A 2 x 2 repeated measures design with the factors Sentence Type (Positive vs. Negative) and Context (Positive vs. Negative). DV is reaction time (RT).

- The data file is called DV and is in *long* format (i.e., each row is one observation):



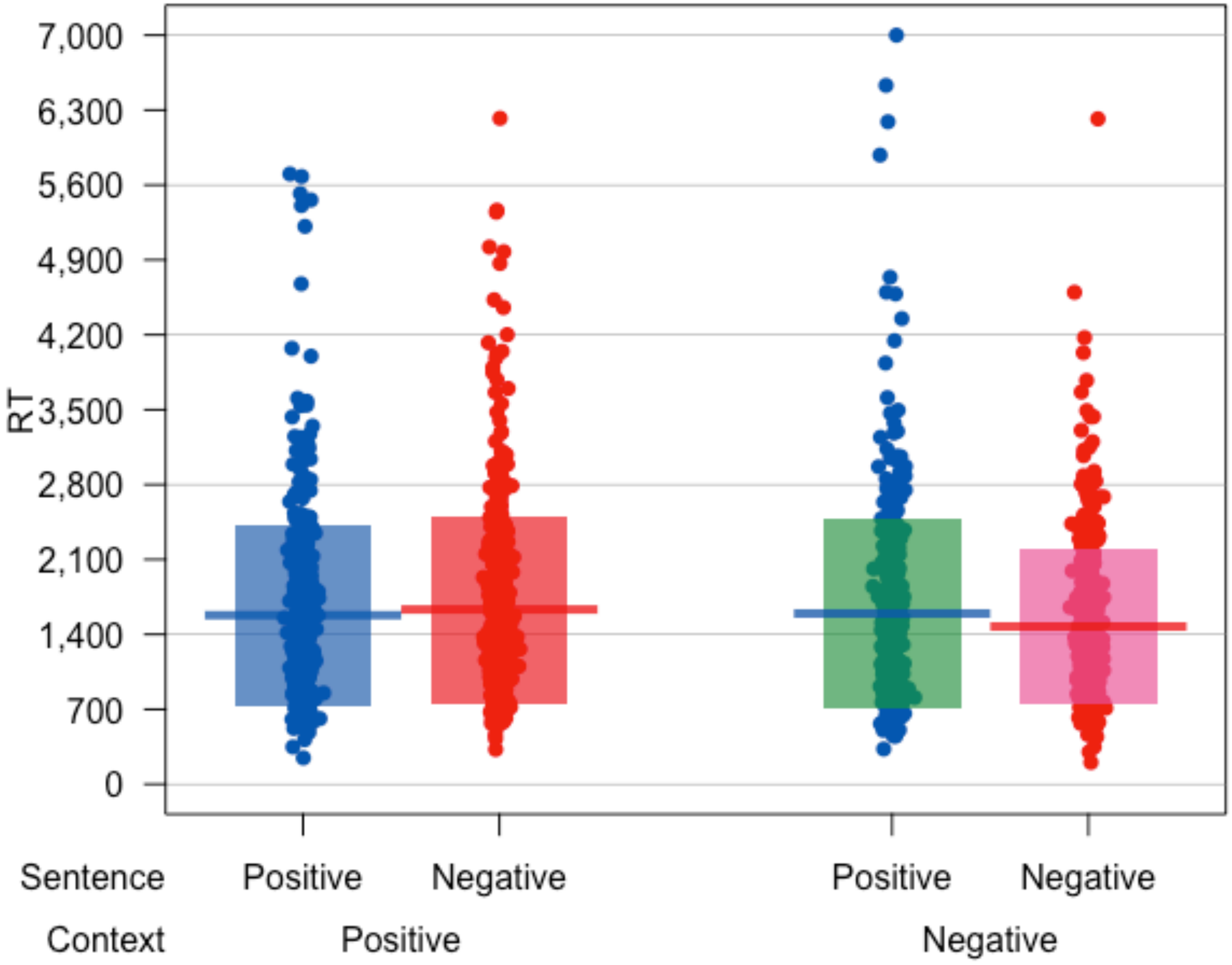
	Subject	Item	RT	Sentence	Context
1	1	3	1270	Positive	Negative
2	1	7	739	Positive	Negative
3	1	11	982	Positive	Negative
4	1	15	1291	Positive	Negative
5	1	19	1734	Positive	Negative
6	1	23	1757	Positive	Negative
7	1	27	1052	Positive	Negative
8	2	4	1706	Positive	Negative
9	2	8	533	Positive	Negative
10	2	12	1009	Positive	Negative
11	2	16	939	Positive	Negative
12	2	20	1848	Positive	Negative
13	2	24	1435	Positive	Negative

Showing 1 to 14 of 1,680 entries

Generating Descriptives

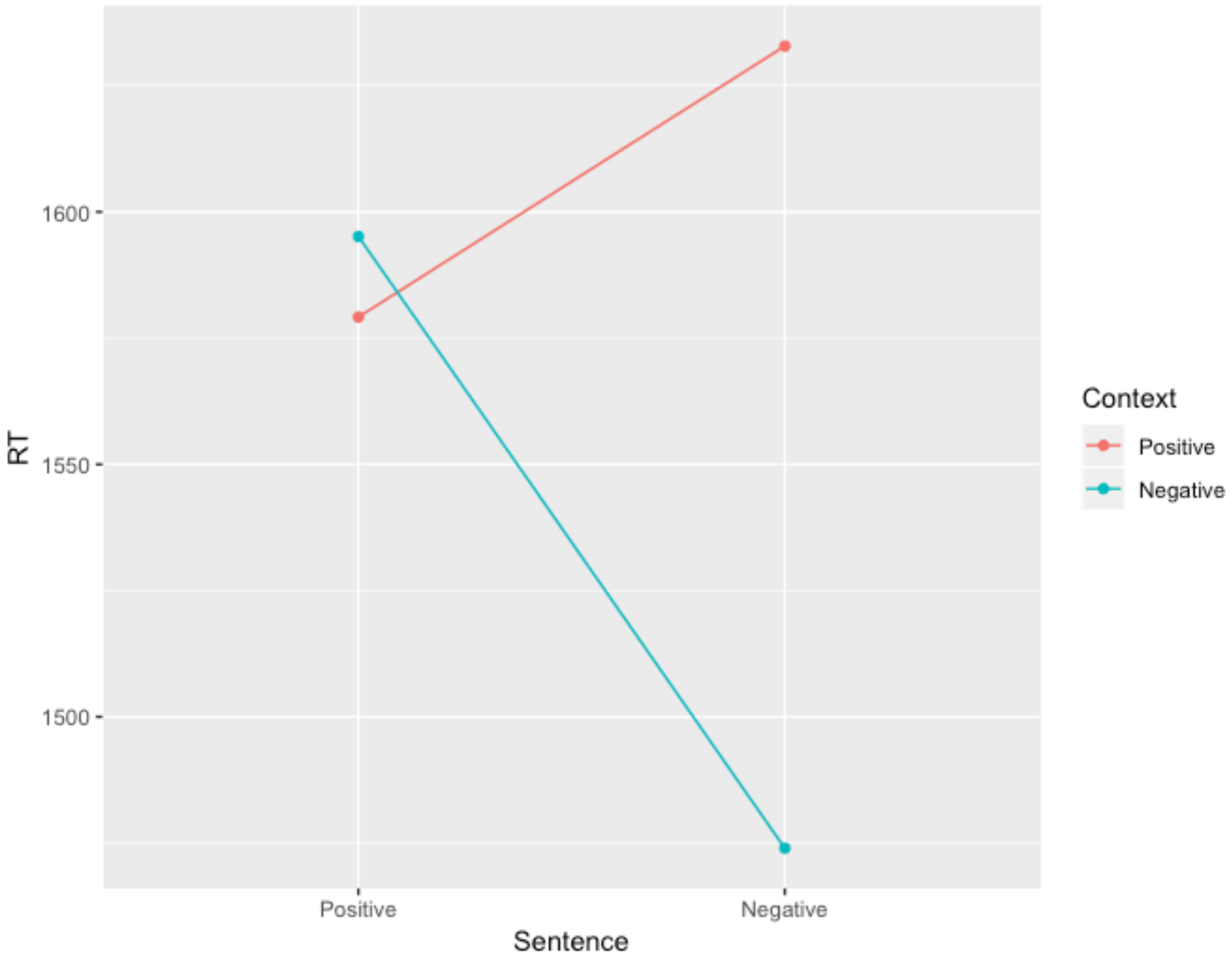
```
> describeBy(DV$RT, group = list(DV$Sentence, DV$Context))

 Descriptive statistics by group 
: Positive
: Positive
   vars   n    mean     sd median trimmed   mad min  max range skew kurtosis    se
X1    1 420 1579.18 840.61   1427 1467.34 660.5 246 5703  5457 1.92     5.78 41.02
-----------------------------------------------------------------------------------
: Negative
: Positive
   vars   n    mean     sd median trimmed    mad min  max range skew kurtosis    se
X1    1 409 1632.85 876.75   1379 1500.97 591.56 325 6223  5898 1.83     4.42 43.35
-----------------------------------------------------------------------------------
: Positive
: Negative
   vars   n    mean     sd median trimmed    mad min  max range skew kurtosis    se
X1    1 419 1595.13 886.86   1444 1479.01 748.71 329 7000  6671 2.16     7.97 43.33
-----------------------------------------------------------------------------------
: Negative
: Negative
   vars   n    mean     sd median trimmed    mad min  max range skew kurtosis    se
X1    1 420 1473.96 728.61 1308.5 1384.71 578.21 204 6218  6014 1.65     5.06 35.55
```

Visualising our Raw Data

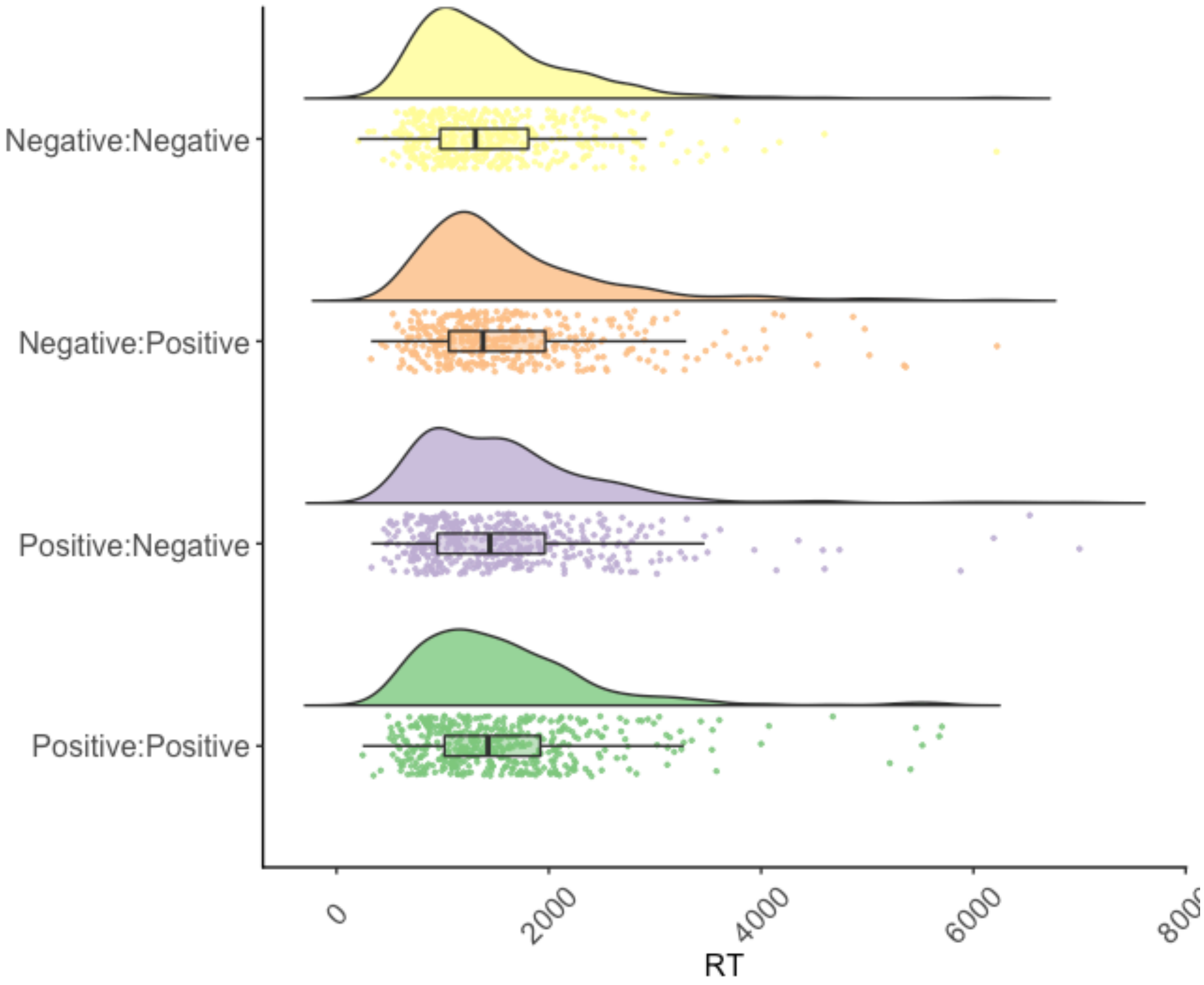


Visualising Our Aggregated Data

```
> data_agg <- DV %>% group_by(Sentence, Context) %>% summarise_at("RT", c(Mean, sd), na.rm = T)
> colnames(data_agg) <- c("Sentence", "Context", "RT", "SD")
> ggplot(data_agg,aes(x = Sentence, y = RT, group = Context, colour = Context)) + geom_point() +
geom_line()
```



Visualising as a Raincloud Plot



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

- Syntax corresponds to RT being predicted by the two factors (Sentence*Context 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+Sentence*Context| Item) for by-item (i.e., F2) analysis. This requires the data to contain the individual observations (not aggregated as means).

By Subjects

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

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

Response: RT
                 num Df den Df    MSE      F       ges  Pr(>F)
Sentence              1     59 124547 0.6283 0.0016524 0.43114
Context               1     59  90195 3.1767 0.0060231 0.07984 .
Sentence:Context      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

```
> model1 <- aov_4(RT ~ Sentence * Context + (1 + Sentence * Context | Item),
data = DV, na.rm = TRUE)

> anova (model1)
Anova Table (Type 3 tests)

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

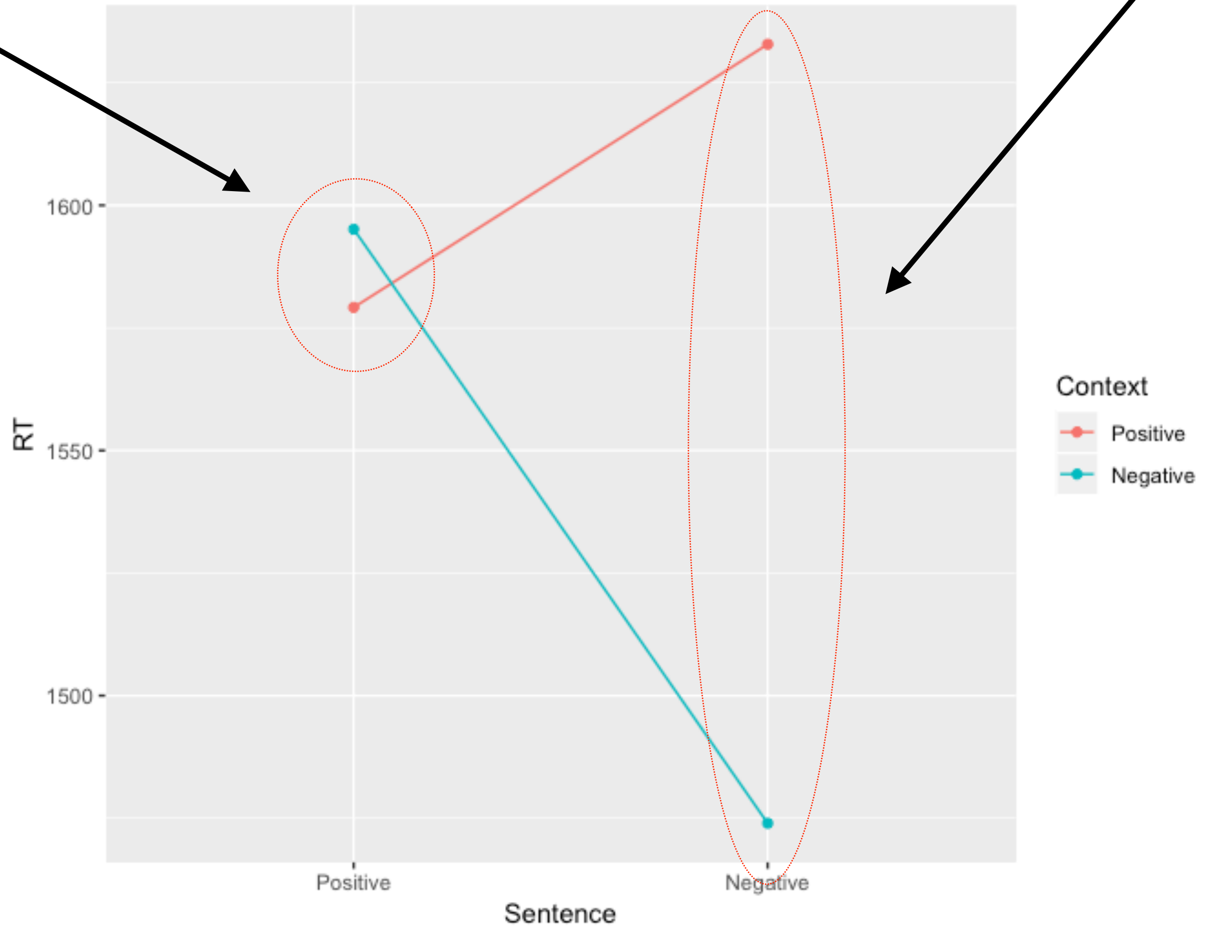
- With the same datafile and just by changing *one* word in the analysis code.

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, pairwise ~ Sentence * Context, adjust = "none")
$emmeans
 Sentence Context    emmean       SE     df lower.CL upper.CL
 Positive Positive 1579.181 57.78624 137.64 1464.917 1693.445
 Negative Positive 1627.877 57.78624 137.64 1513.614 1742.141
 Positive Negative 1594.889 57.78624 137.64 1480.625 1709.152
 Negative Negative 1473.962 57.78624 137.64 1359.698 1588.225

Confidence level used: 0.95

$contrasts
 contrast                                estimate       SE     df t.ratio p.value
 Positive,Positive - Negative,Positive -48.69643 60.33730 115.72  -0.807  0.4213
 Positive,Positive - Positive,Negative -15.70794 55.39009 117.95  -0.284  0.7772
 Positive,Positive - Negative,Negative 105.21905 59.82499 115.06   1.759  0.0813
 Negative,Positive - Positive,Negative  32.98849 59.82499 115.06   0.551  0.5824
 Negative,Positive - Negative,Negative 153.91548 55.39009 117.95   2.779  0.0064
 Positive,Negative - Negative,Negative 120.92698 60.33730 115.72   2.004  0.0474
```

The pairwise comparisons tell us that Positive Sentences are read at the same speed regardless of Context, and that Negative Sentences are read more quickly when they appear in a Negative Context relative to a Positive Context.



These two points are **not** statistically different from each other.

These two points **are** statistically different from each other.

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) = 3.177, p = .080, ges = .006), but an interaction between Sentence and Context (F(1, 59) = 4.60, p = .036, ges = .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(117.95) = 2.78, p = .0064) while Positive sentences were read equivalently in Negative vs. Positive contexts (1,595 ms. vs. 1,579 ms., t(117.95) = .284, p = .777).

Now for the lab…