Lecture 8 - ANOVA part 2

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

Assignment

- The data for the assignment were all generated by random sampling from the normal (and multivariate normal) distributions. Assume the assumptions of ANOVA have been met for the purposes of this assignment (i.e., you don't have to test/report them). We'll cover how to test these assumptions in Semester 2.

R Markdown

- R Markdown is a language in and of itself that allows you to produce documents in many formats (e.g., .html, .pdf, .doc) that contain your R code, the output of that code, and narrative that you write describing what you are doing (and why).

- R Markdown documents can be produced from within RStudio.

- R Markdown cheat sheet is in the R documentation folder on Blackboard.

- First we need to create a new R Markdown file:
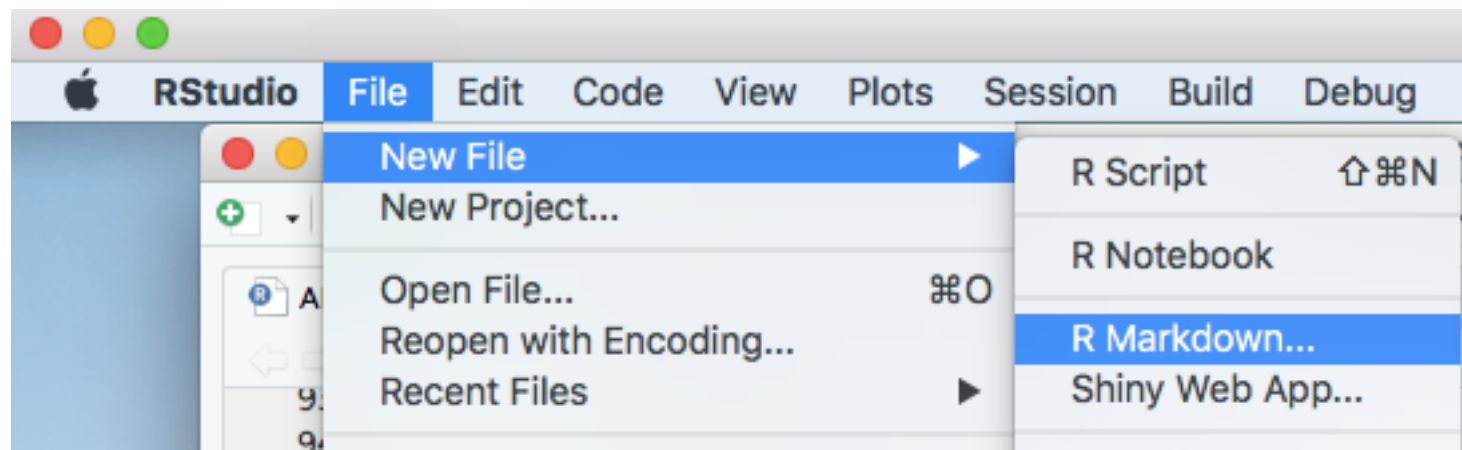
Type the title of your document here.

New R Markdown

Document

Title: ANOVA example

Presentation

Author: Andrew Stewart

Shiny

From Template

Important - replace your name here with your student ID number.

Default Output Format:

◉ HTML

Recommended format for authoring (you can switch to PDF or Word output anytime).

Select the kind of file you want to be generated by your Markdown - you can change this later btw.

○ PDF

PDF output requires TeX (MiKTeX on Windows, MacTeX 2013+ on OS X, TeX Live 2013+ on Linux).

○ Word

Previewing Word documents requires an installation of MS Word (or Libre/Open Office on Linux).

OK   Cancel

You will now see a document like the above - it contains lots of example narrative (with a white background) and R code (with a grey background). We could actually 'knit' this document by clicking on ⬤ Knit to see what is produced...

What you will get is an html file (because that's the type of document we asked to be produced) that contains the R code, the associated output, plus the narrative. Notice how the ## symbols increasing the font size to allow us to generate headings in our narrative.

ANOVA example

Andrew Stewart

18/09/2018

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```
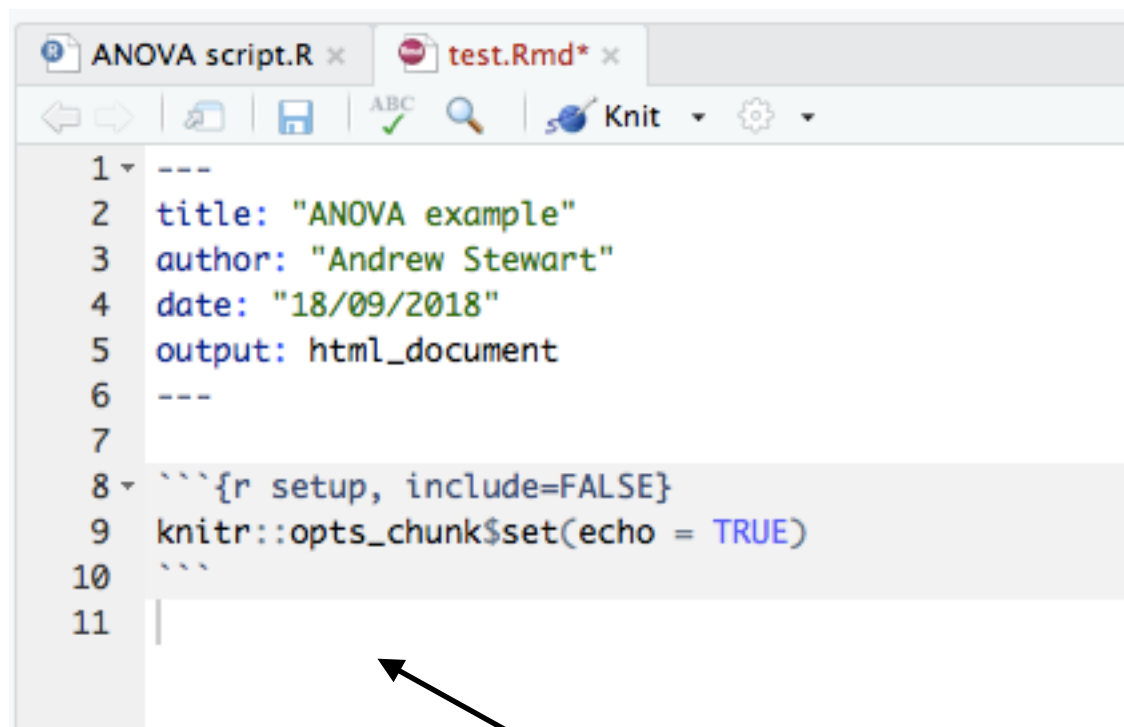
Including Plots

You can also embed plots, for example:
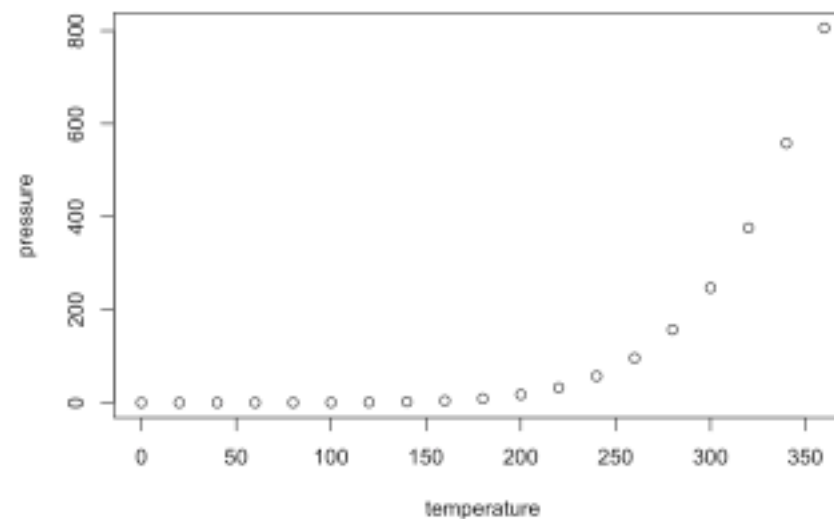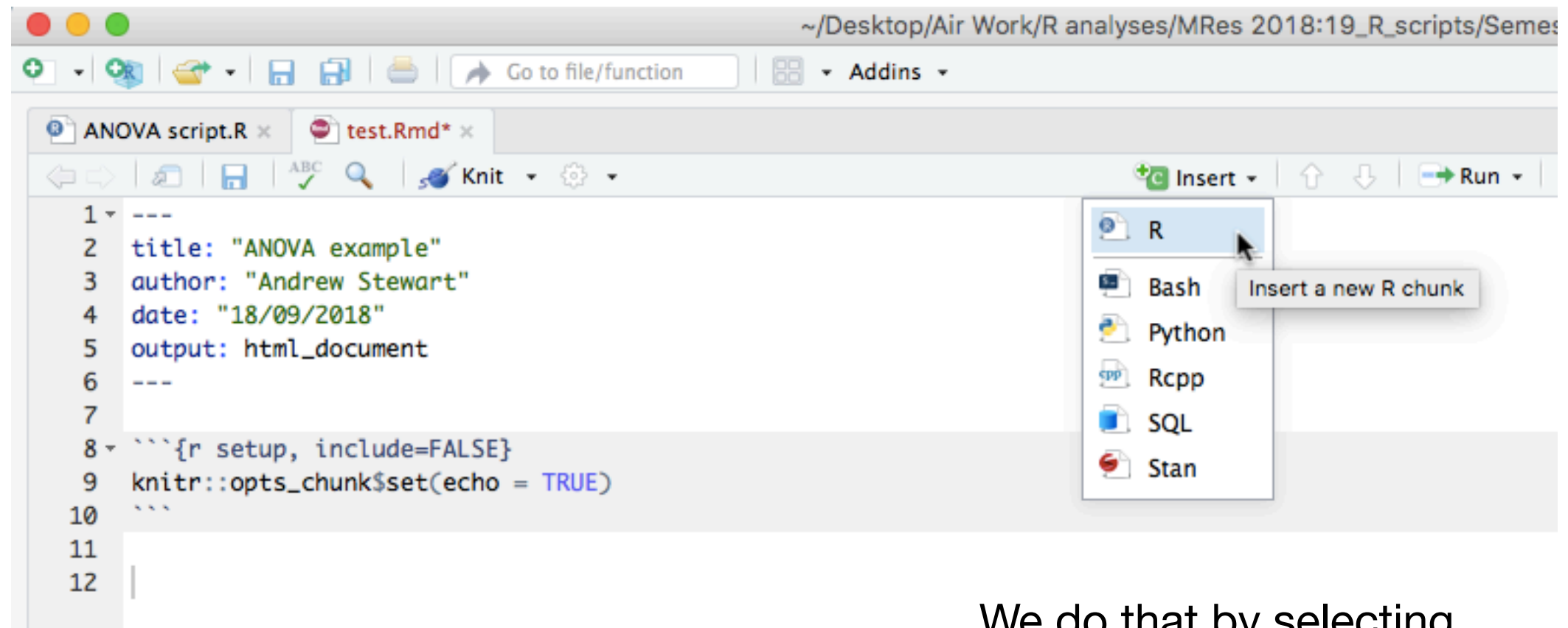
Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

When you come to create a document using R Markdown, you'll first want to delete a lot of the example code that appears when you first start a new Markdown file:

```
 ANOVA script.R ×     test.Rmd* ×

 1 ▾ ---
 2   title: "ANOVA example"
 3   author: "Andrew Stewart"
 4   date: "18/09/2018"
 5   output: html_document
 6   ---
 7
 8 ▾ ```{r setup, include=FALSE}
 9   knitr::opts_chunk$set(echo = TRUE)
10   ```
11   |
```

We now want to paste in chunks of our R code here

Go to file/function   Addins

ANOVA script.R ×   **test.Rmd*** ×

ABC   Knit ⚙   Insert   Run

```
1  ---
2  title: "ANOVA example"
3  author: "Andrew Stewart"
4  date: "18/09/2018"
5  output: html_document
6  ---
7
8  ```{r setup, include=FALSE}
9  knitr::opts_chunk$set(echo = TRUE)
10  ```
11
12  |
```

R

Bash   Insert a new R chunk

Python

Rcpp

SQL

Stan

We do that by selecting "Insert" and then "R" - this will allow us to add some new R code.

ANOVA script.R ×   **Untitled1*** ×

ABC   Knit ⚙

```
1  ---
2  title: "ANOVA"
3  author: "Andrew Stewart"
4  date: "18/09/2018"
5  output: html_document
6  ---
7
8  ```{r setup, include=FALSE}
9  knitr::opts_chunk$set(echo = TRUE)
10  ```
11
12  ```{r}
13  |
14  ```
15
16
```

We can now insert our new R code here.

Screenshot 1 (ANOVA script.R / Untitled1*):

```
1  ---
2  title: "ANOVA"
3  author: "Andrew Stewart"
4  date: "18/09/2018"
5  output: html_document
6  ---
7
8  ```{r setup, include=FALSE}
9  knitr::opts_chunk$set(echo = TRUE)
10 ```
11
12 ```{r}
13 library(tidyverse)
14 ```
15
16 |
```
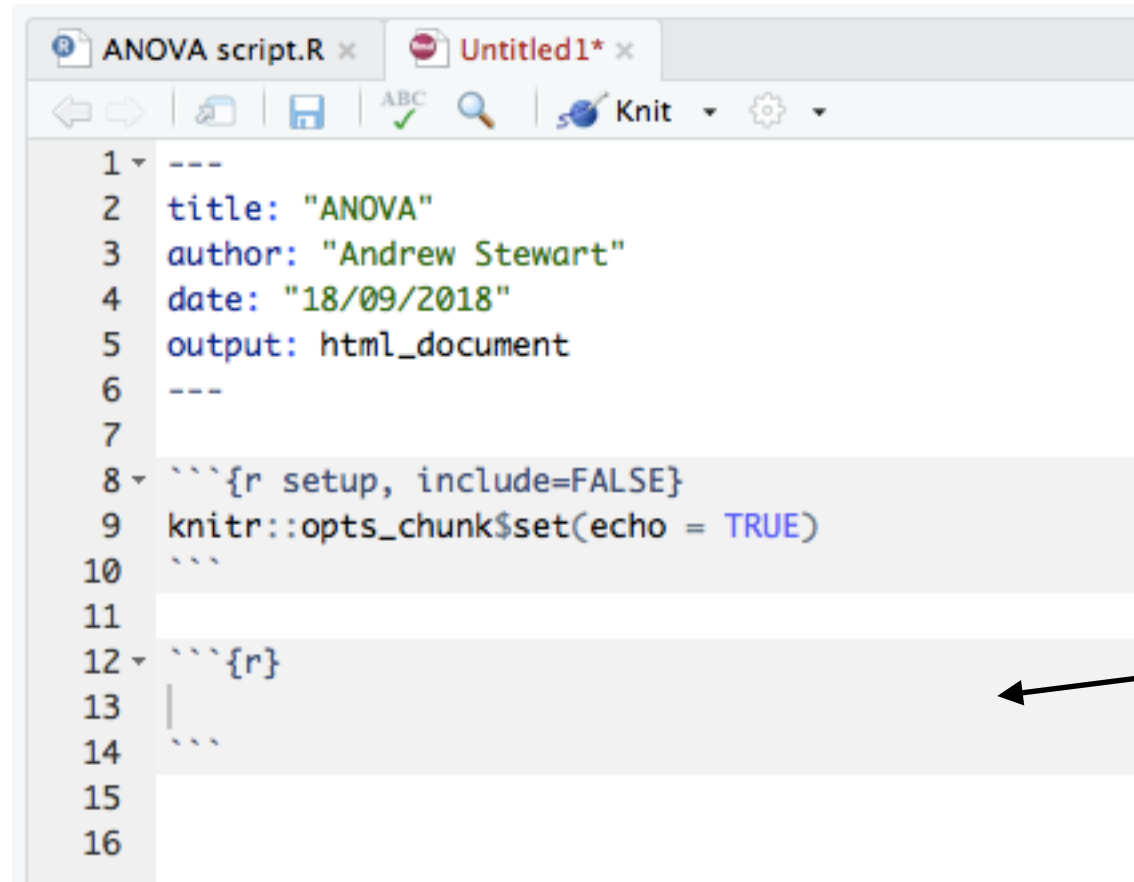
We load the tidyverse packages first.

But maybe we should precede that with some narrative (and a heading) explaining what we're doing.

Screenshot 2 (ANOVA script.R / Untitled1*):

```
1  ---
2  title: "ANOVA"
3  author: "Andrew Stewart"
4  date: "18/09/2018"
5  output: html_document
6  ---
7
8  ```{r setup, include=FALSE}
9  knitr::opts_chunk$set(echo = TRUE)
10 ```
11
12 ## This is my first R Markdown document
13
14 First we load the tidyverse.
15 ```{r}
16 library(tidyverse)
17 ```
18
```

What happens if we now knit this short document?

ANOVA

Andrew Stewart

18/09/2018

This is my first R Markdown document

First we load the tidyverse.

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.2.1 ──
```

```
## ✔ ggplot2 3.0.0     ✔ purrr   0.2.5
## ✔ tibble  1.4.2     ✔ dplyr   0.7.6
## ✔ tidyr   0.8.1     ✔ stringr 1.3.1
## ✔ readr   1.1.1     ✔ forcats 0.3.0
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

Hmm, so we get some messages and warnings in our document - how can we get rid of these?

```
14   First we load the tidyverse.
15 ▾ ```{r, message=FALSE}
16   library(tidyverse)
17   ```
18
```

We can set the option for this chuck of R code not to display any messages by setting messages=FALSE within the first curly bracket. There are lots of other options available to use (e.g., warnings=FALSE).

Everything you need (and probably lots you don't) can be found in this cheatsheet here:

http://www.rstudio.com/wp-content/uploads/2016/03/rmarkdown-cheatsheet-2.0.pdf

and here is the 'R Markdown: The Definitive Guide':

https://bookdown.org/yihui/rmarkdown/

I've even put together a video tutorial talking you through how to make an R Markdown document in the BB folder…

https://youtu.be/cWGqWhfVkuw

The ANOVA Assessment

- The ANOVA assessment this semester and the mixed models one next semester both need to be produced using R Markdown.

- Top tips:

  Have lots of narrative explaining what you're doing (and why).

  Use small chunks of code bookended by narrative - the ideal code chunk length is where you can have one comment that describes what that bit of code does.

  If you find you're commenting about two things that one chunk of R code does, that chunk should probably be split into two…

Now onto ANOVA part 2

- Last week 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, 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(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 = 90, hjust = 1)) +
    facet_wrap(~ Age) + guides(colour = FALSE)
```
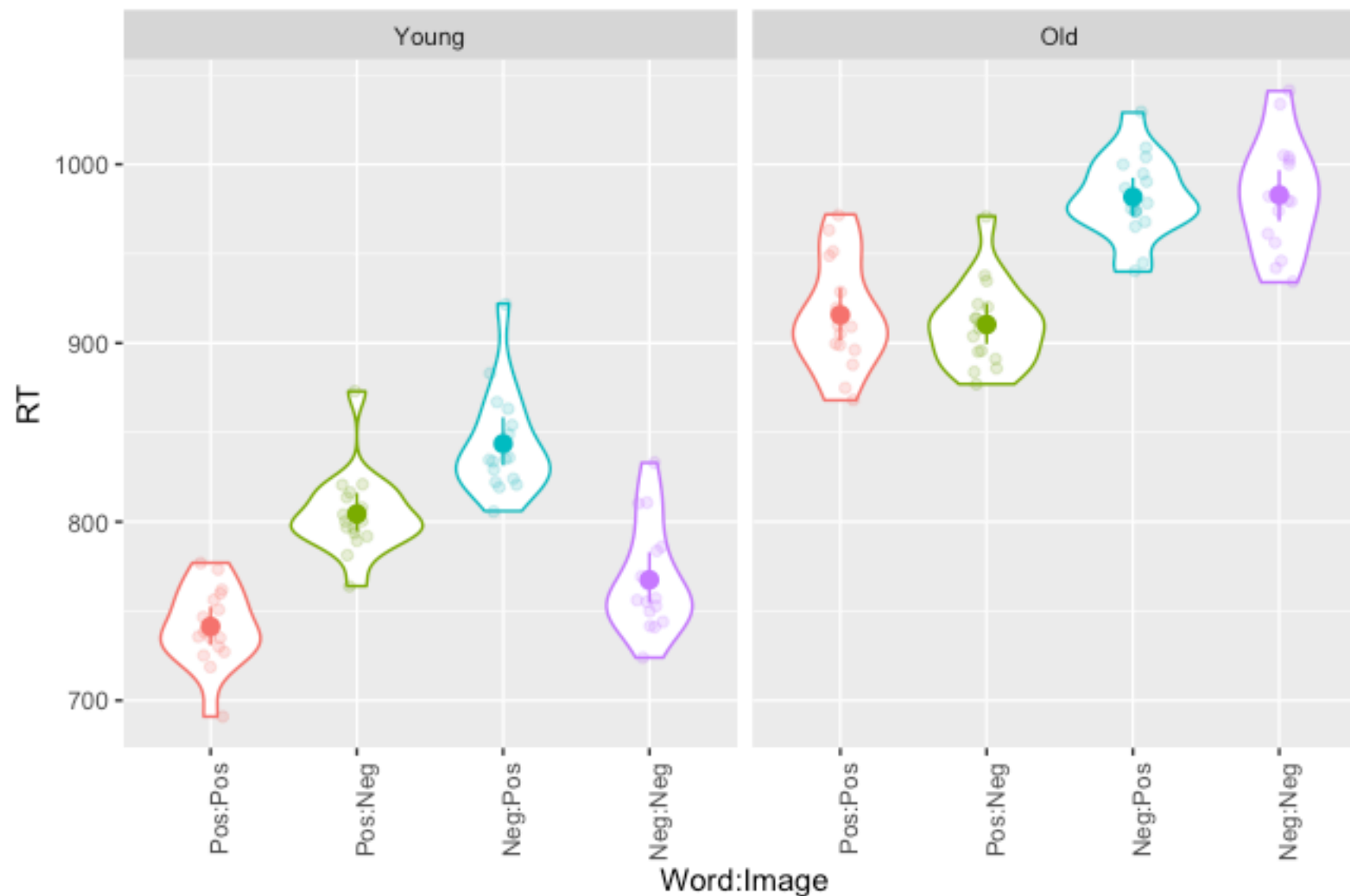

We can see it looks like the Young and Old groups are behaving a little differently.

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), data)
```

We are asking for the anova to be run on 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 doesn't need to be told which factors are repeated, and which are between - it can work this out itself from the data 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
```
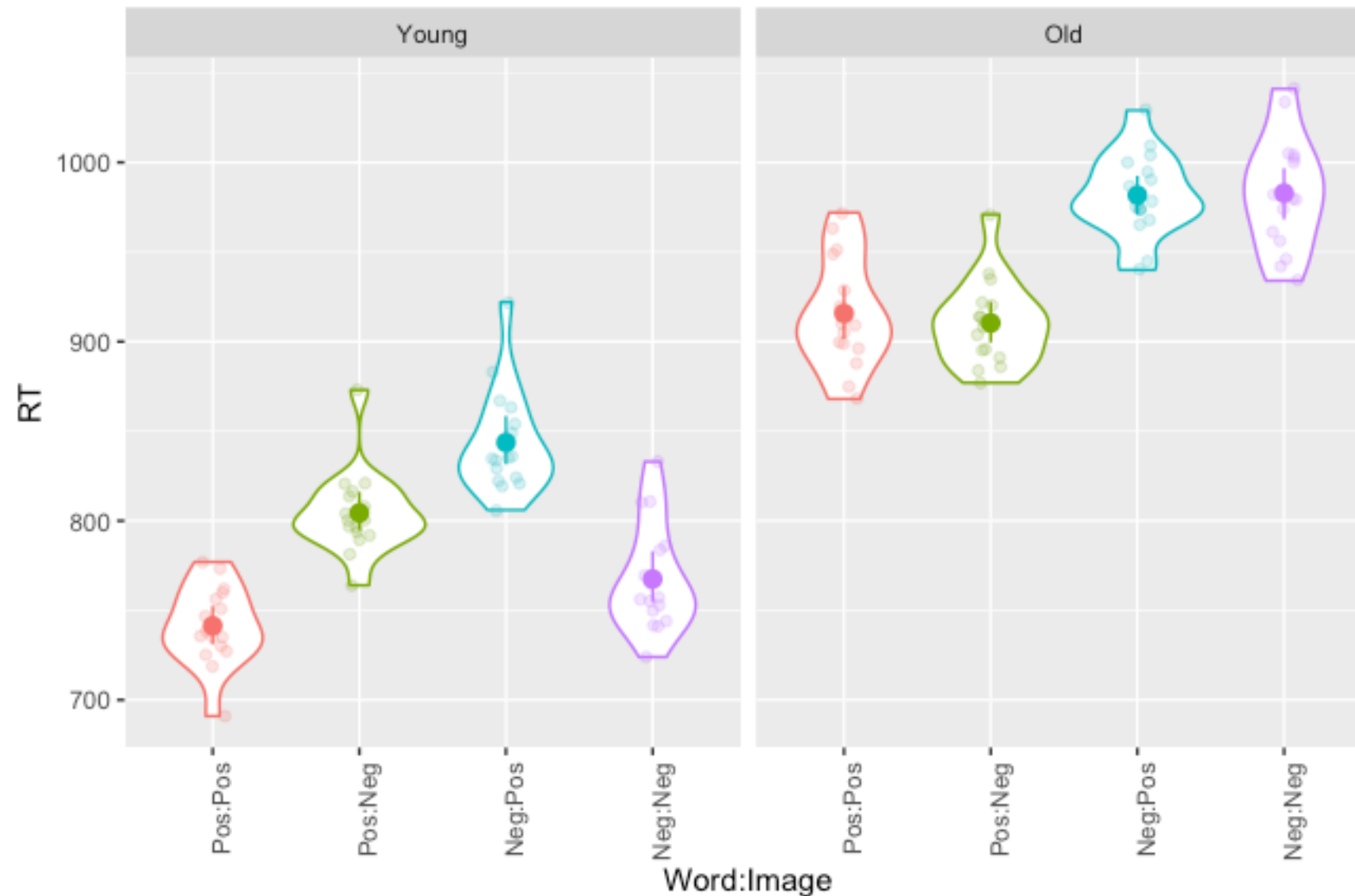
From this we can see we have main effects of Age and Word, no main effect of Image, significant 2-way interactions of Age x Word, and of Word x Image and, crucially, a 3-way interaction between all three factors - Age x Word x Image…

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(data, Age == "Young")
> model_young <- aov_4(RT ~ Word * Image + (1 + Word * Image | Participant),
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
  Pos  Pos    741.3750 6.571636 57.69 728.2189 754.5311
  Neg  Pos    843.6875 6.571636 57.69 830.5314 856.8436
  Pos  Neg    804.3750 6.571636 57.69 791.2189 817.5311
  Neg  Neg    767.5000 6.571636 57.69 754.3439 780.6561

Confidence level used: 0.95

$contrasts
 contrast              estimate        SE     df t.ratio p.value
  Pos,Pos - Neg,Pos  -102.3125  8.812773 29.72 -11.610  <.0001
  Pos,Pos - Pos,Neg   -63.0000  9.615140 28.34  -6.552  <.0001
  Pos,Pos - Neg,Neg   -26.1250 10.000957 29.36  -2.612  0.0842
  Neg,Pos - Pos,Neg    39.3125 10.000957 29.36   3.931  0.0028
  Neg,Pos - Neg,Neg    76.1875  9.615140 28.34   7.924  <.0001
  Pos,Neg - Neg,Neg    36.8750  8.812773 29.72   4.184  0.0014

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

Key pairwise comparisons are significant

But what about the Old group?

```
> old_filter <- filter(data, Age == "Old")
> model_old <- aov_4(RT ~ Word * Image + (1 + Word * Image | Participant),
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…
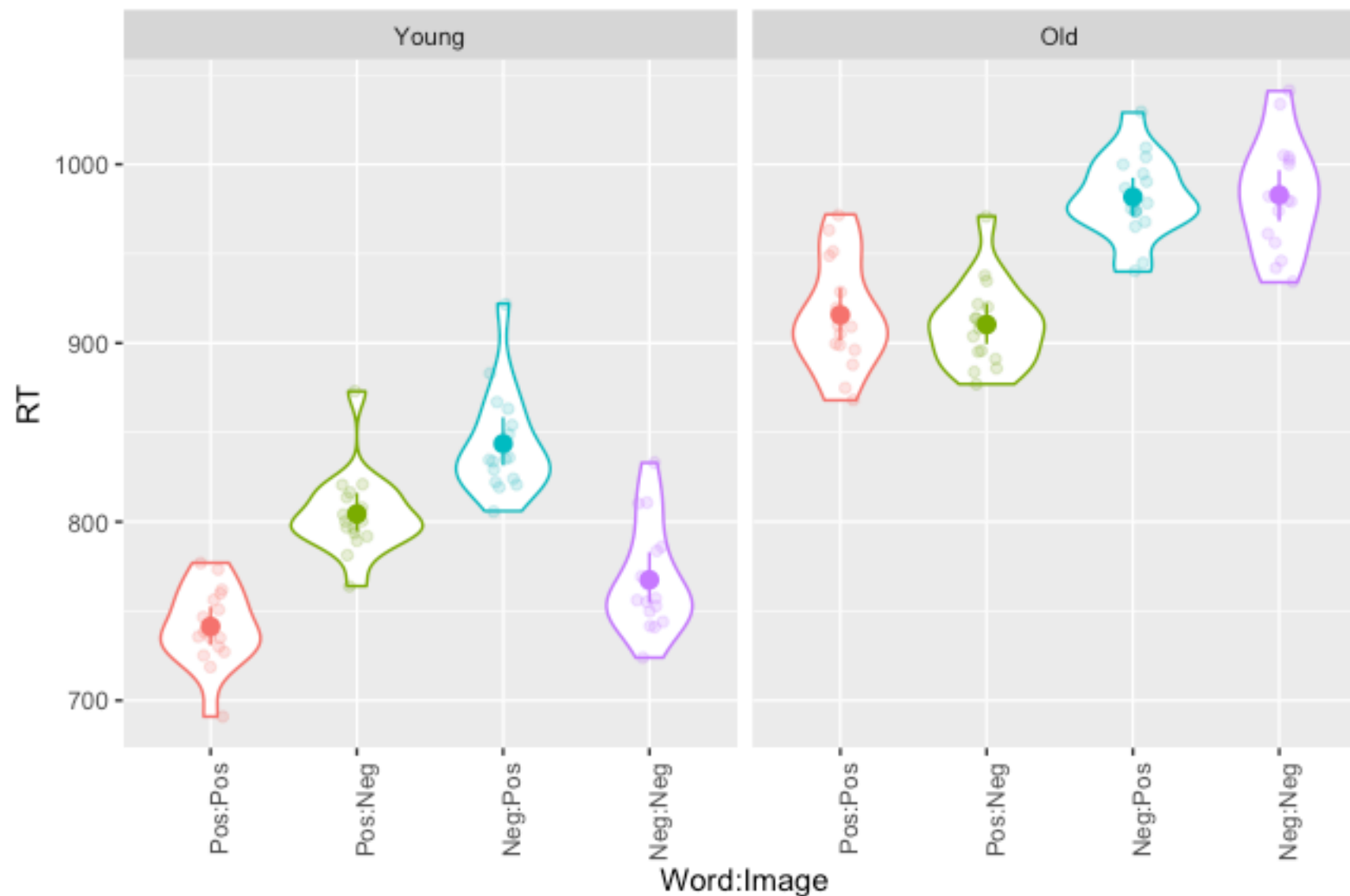
ANCOVA

- One of our examples from last week 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()
```
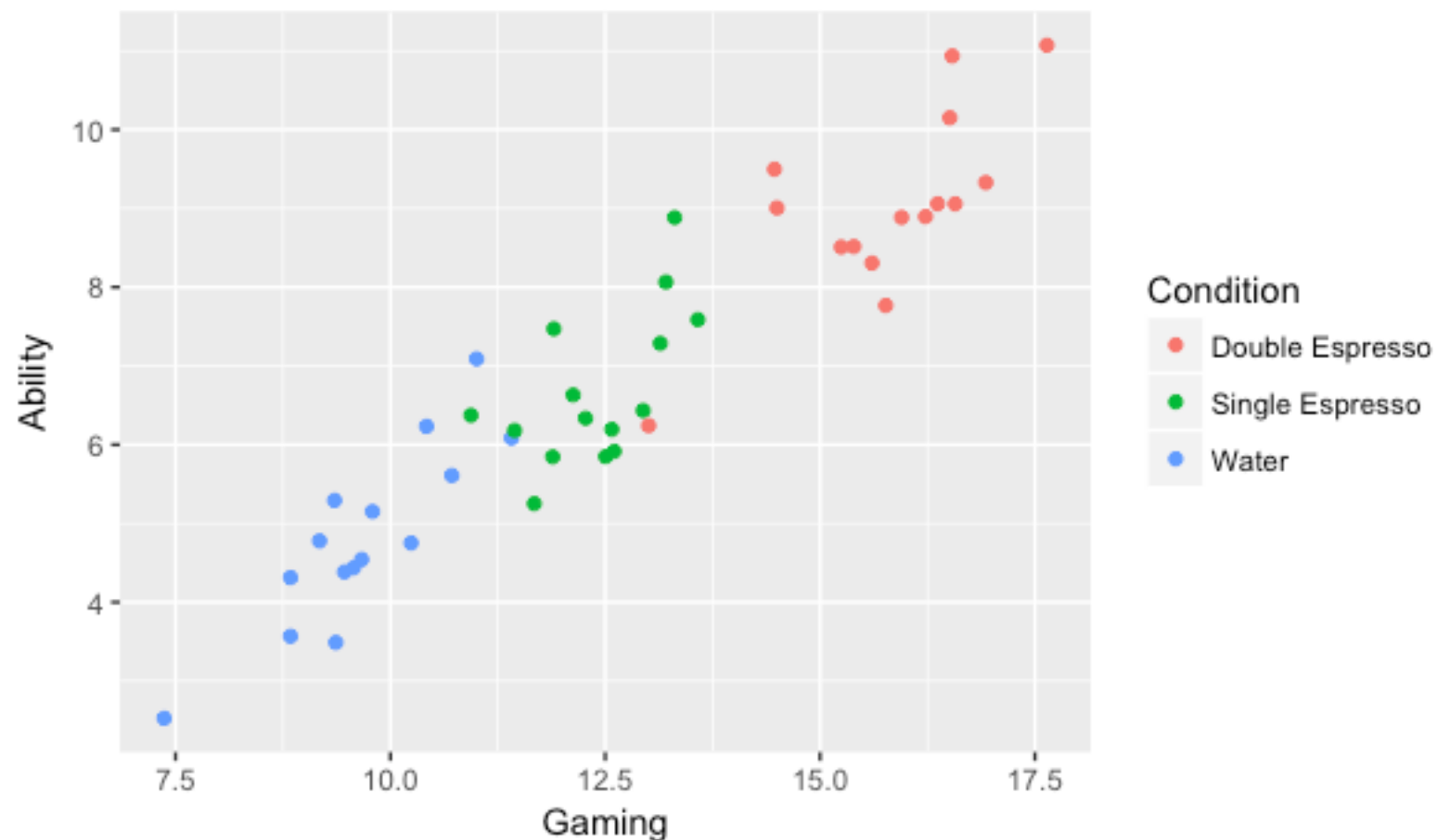

- So we can see there's a relationship between our DV (Ability) and our covariate (Gaming Frequency)…

- 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')
```
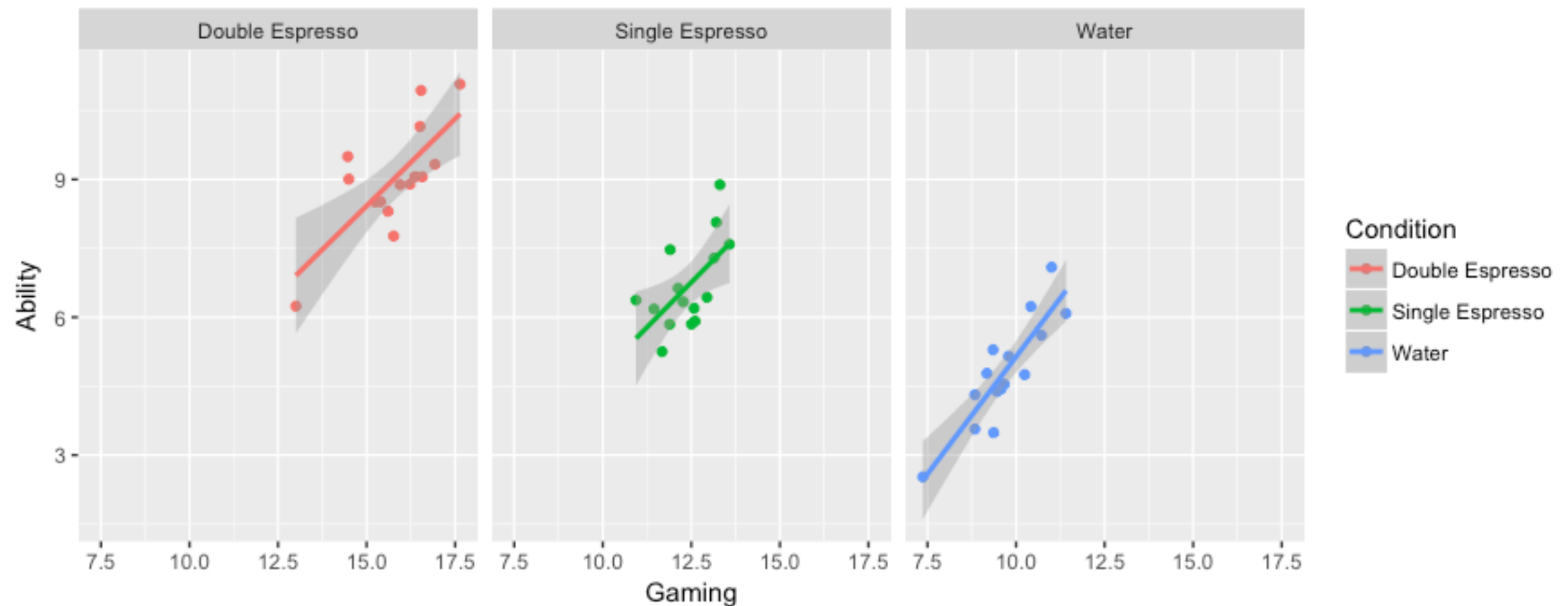
Running a 1-way between participants ANOVA (and ignoring the covariate)…

```
> 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

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

 Descriptive statistics by group
group: Double Espresso
   vars  n mean   sd median trimmed  mad  min   max range  skew kurtosis   se
X1    1 15 9.02 1.19   9.01    9.07 0.73 6.24 11.07  4.83 -0.26     0.16 0.31
-------------------------------------------------------------------------------
group: Single Espresso
   vars  n mean   sd median trimmed  mad  min  max range skew kurtosis   se
X1    1 15 6.69 0.98   6.37    6.63 0.78 5.25 8.88  3.63 0.69    -0.53 0.25
-------------------------------------------------------------------------------
group: Water
   vars  n mean   sd median trimmed mad  min  max range skew kurtosis  se
X1    1 15 4.82 1.16   4.75    4.82 0.8 2.53 7.09  4.56 0.03    -0.57 0.3
```

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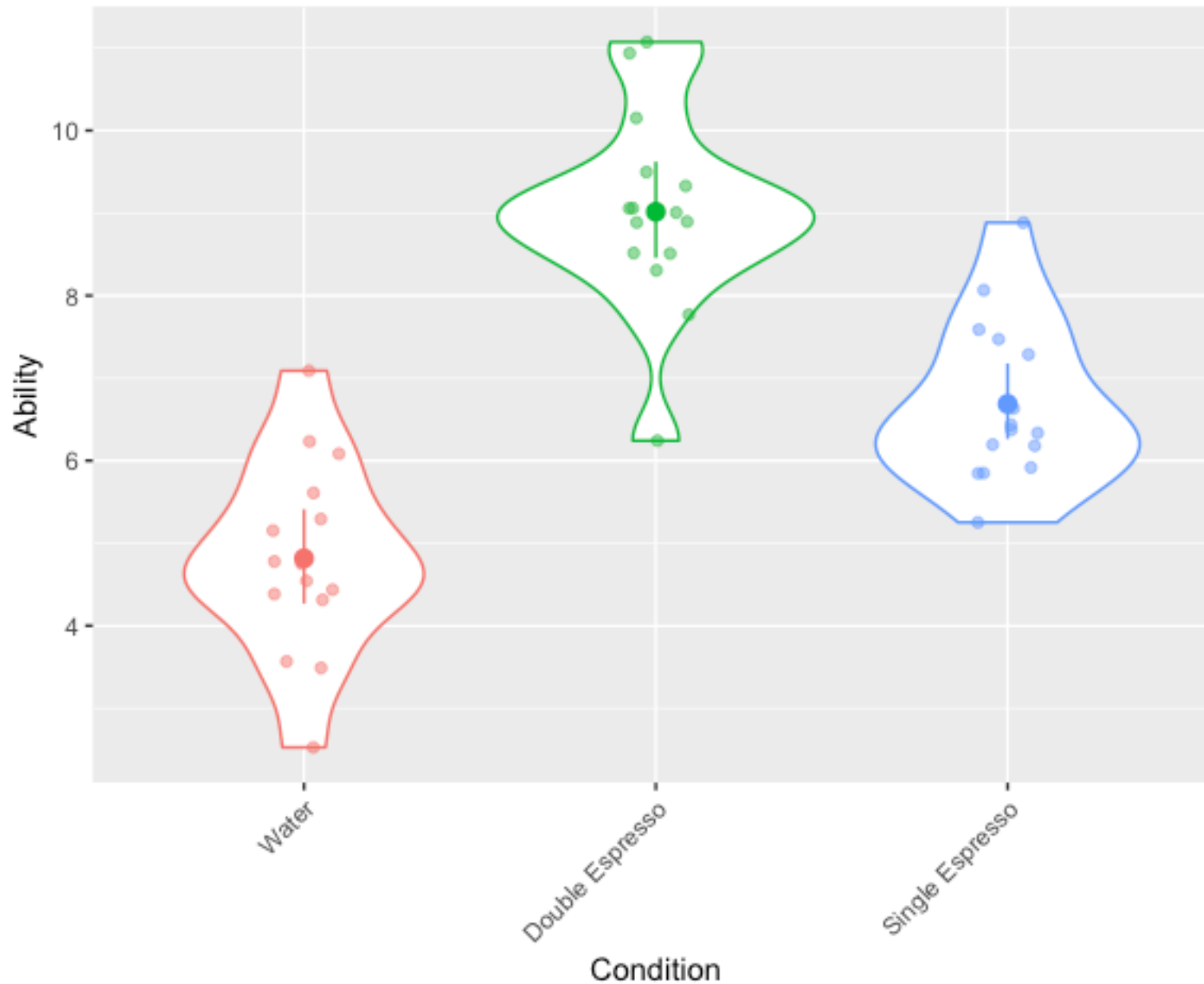
Water mean = 4.82
Double Espresso mean = 9.02
Single Espresso mean = 6.69

- 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 + $\beta 1$(Double Espresso) + $\beta 2$(Single Espresso) + ε

We want to calculate $\beta 1$ and $\beta 2$

```
> 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), $\beta 1$ is 4.2, and $\beta 2$ 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 + β_1(Double Espresso) + β_2(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 + *β1*(Gaming) + *β2*(Double Espresso) + *β3*(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 + $\beta 1$(Gaming) + $\beta 2$(Double Espresso) + $\beta 3$(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 the computer lab…