Week 9 - GLM - ANOVA part 2

Andrew Stewart

Andrew.Stewart@manchester.ac.uk

@ajstewart_lang

https://github.com/ajstewartlang

Week	Topic
1	Introduction, Open Science, and Power
2	Introduction to R
3	Data Wrangling and Visualisation
4	General Linear Model - Regression
5	General Linear Model - Regression
6	No Timetabled Lecture - Reading Week
7	Consolidation Lab
8	General Linear Model - ANOVA
9	General Linear Model - ANOVA
10	Tidy Thursday Data Wrangling & Visualisation Challenge
11	Reproducing your Computational Environment using Binder
12	Dynamic, Reproducible Presentations Using xaringan

Semester 1 Assignments

Data wrangling and visualisation – Due December 5th

ANOVA/ANCOVA – Due January 17th

More ANOVA...

- 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

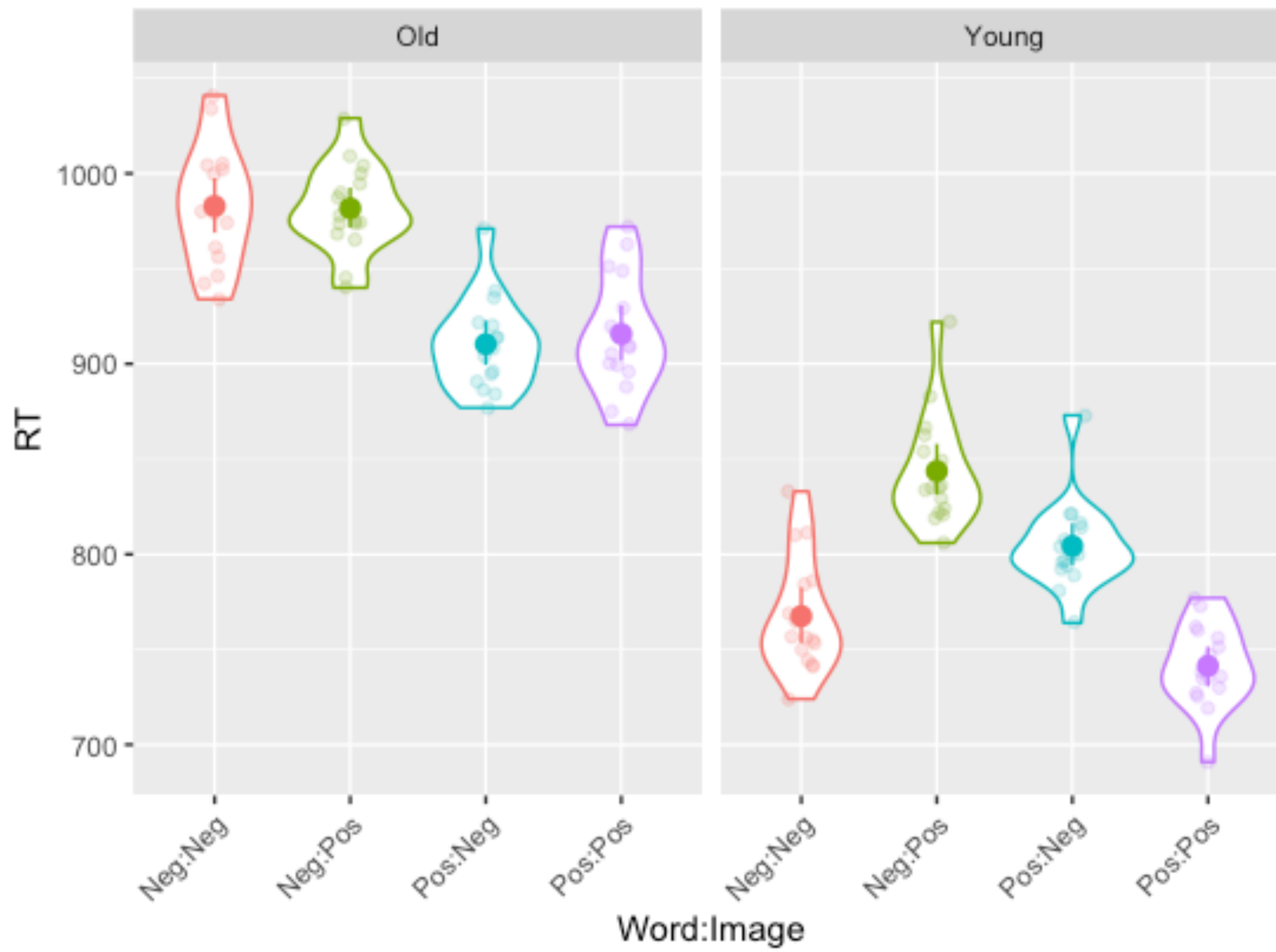
- Similar to our 2 x 2 ANOVA from last time, but let's say we now have Age as an additional factor. It's a between subjects factor. We might think that priming effects might be different for young vs old people.
- So we need to run a 2 x 2 x 2 ANOVA. The first two factors are still within subjects (i.e., 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.

```r
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), data)
```

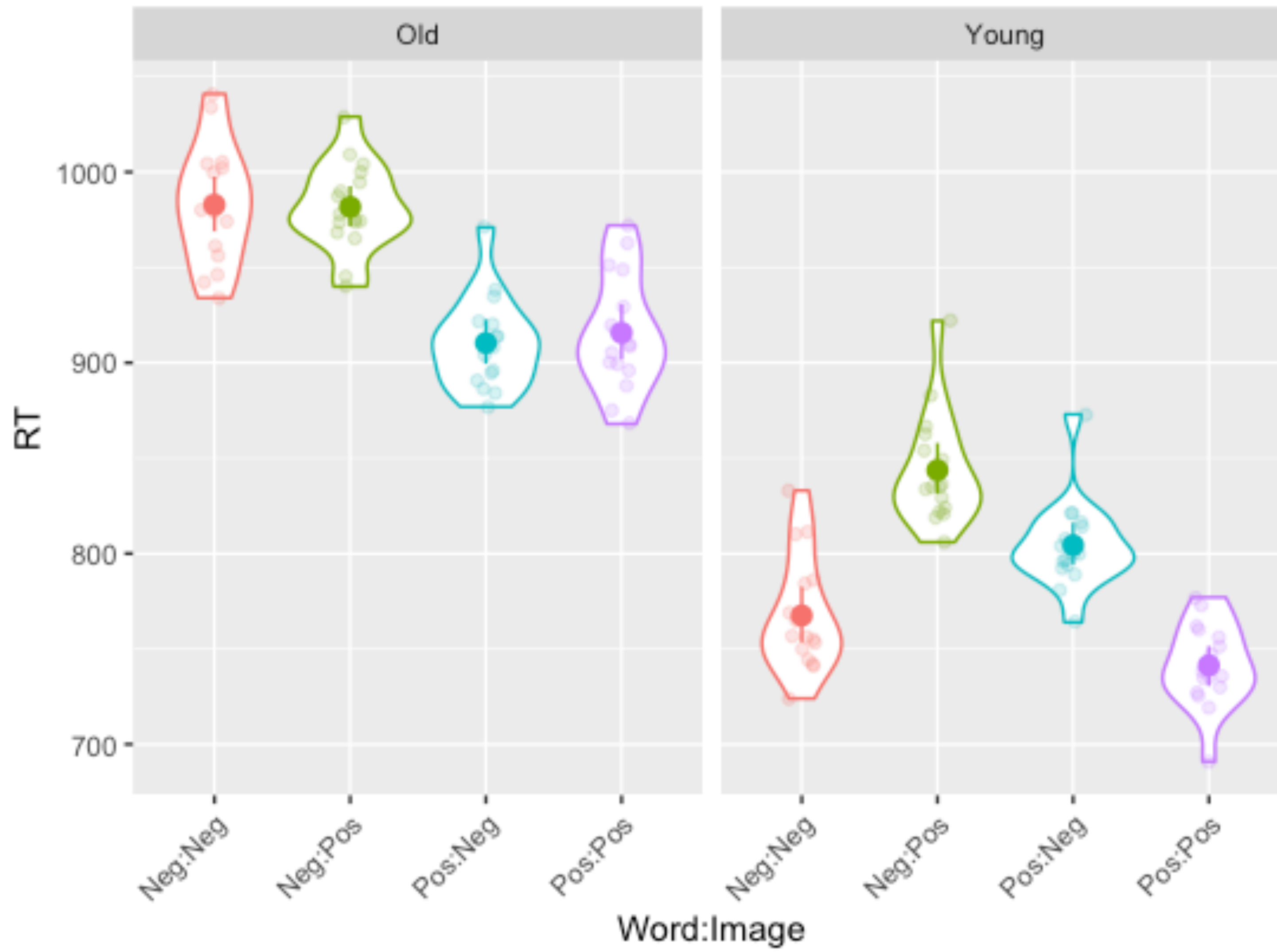
We are asking for the model to be built using the three factors - this will give us three possible main effects, 3 possible 2-way interactions, and a possible 3-way interaction…

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(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

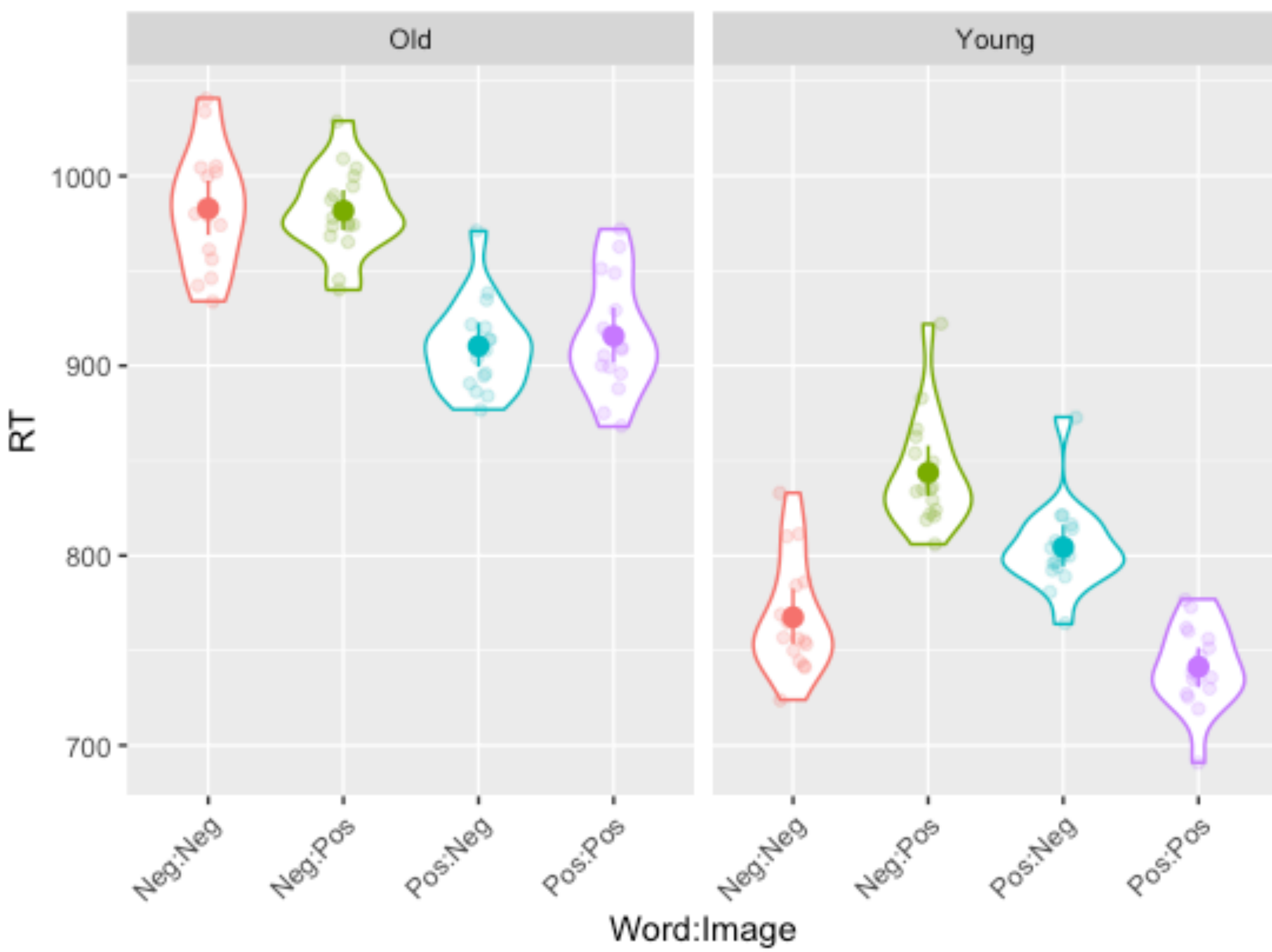
But what about the Old group?

```
> old_filter <- filter(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

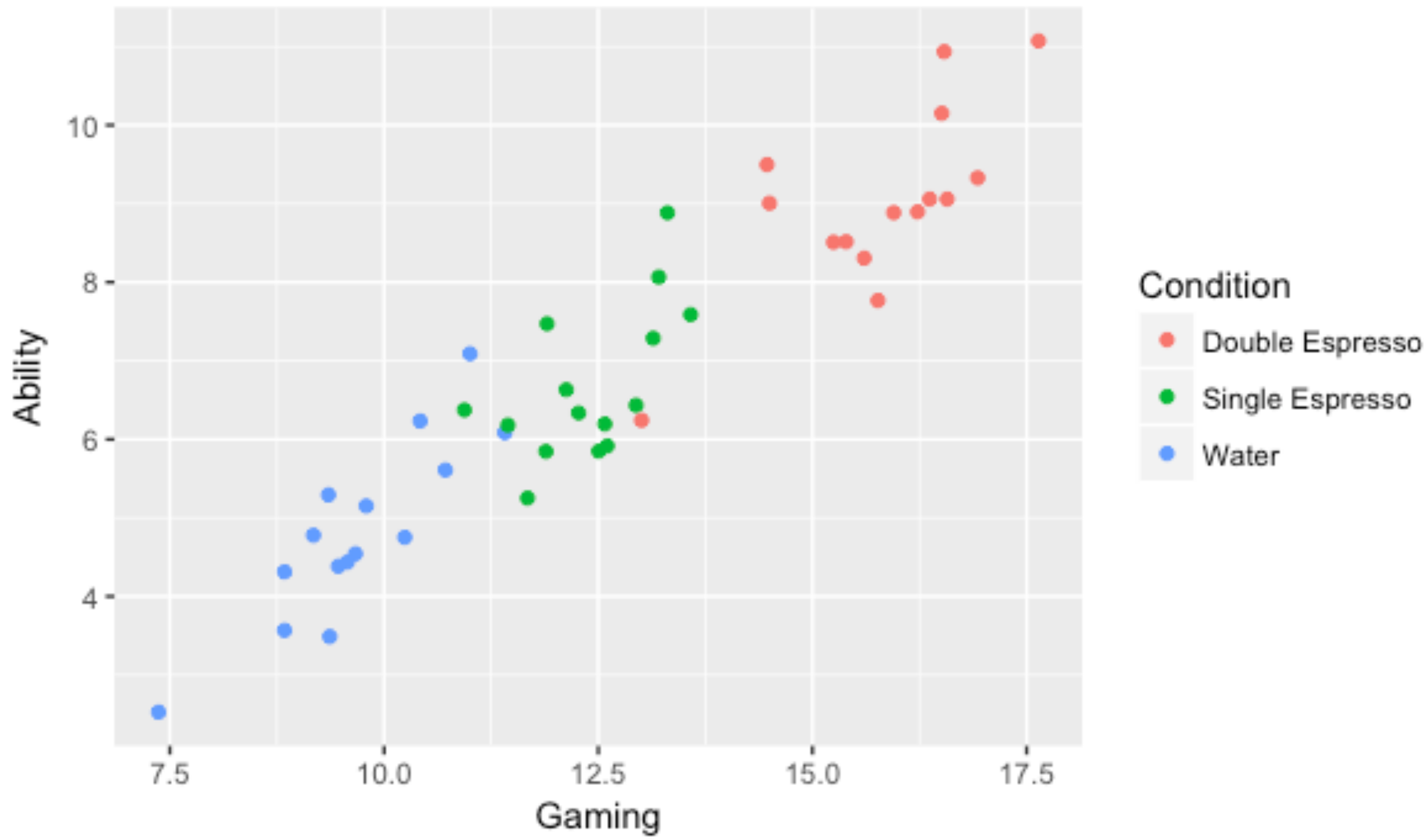
- One of our examples from last week looked at how double espresso vs. single espresso vs. water drinking (our IV) might influence motor performance (our DV).
- Imagine we sampled from a new group of participants - and we think other factors that we are not manipulating might also influence the DV – 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()
```

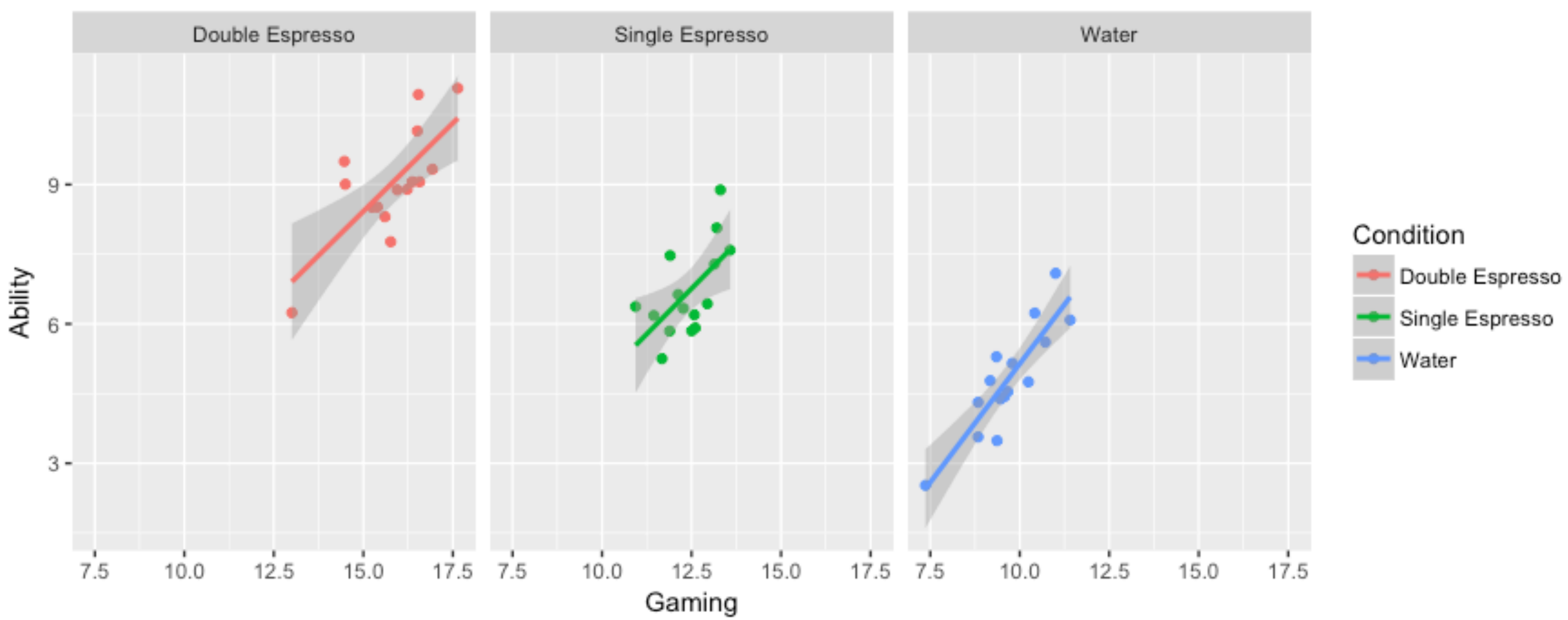


- So we can see there's a relationship between our DV (Ability) and our covariate (Gaming Frequency)…

- 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 not 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).

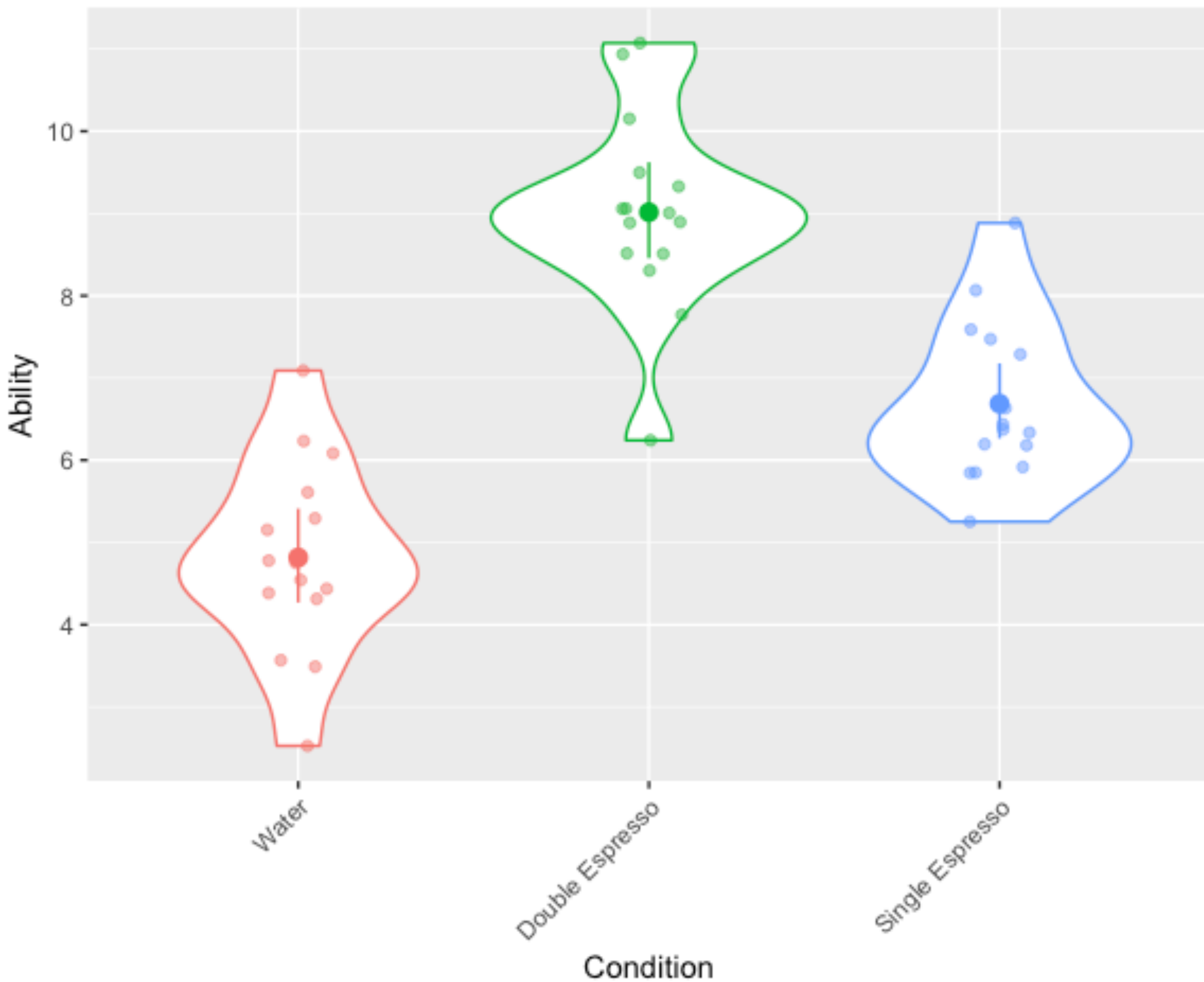
- Type III Sum of Squares tests for effects adjusted for the presence of the other effects (so does not depend on the order of terms).

- 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 the linear model…

- 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 + *β1*(Double Espresso) + *β2*(Single Espresso) + *ε*

We want to calculate *β1* and *β2*

```
> 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), *β1 i*s 4.2, and *β2* 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 + $\beta 1$(Double Espresso) + $\beta 2$(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 + $\beta 1$(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 + $\beta 1$(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…

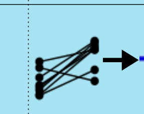
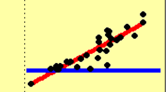
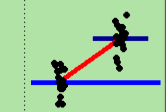
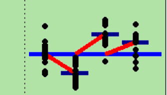
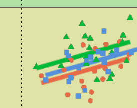
You can now build ANOVA models in R for different kinds of designs, add between participant co-variates, factor out the influence of these co-variates, and you also know why AN(C)OVA is a special case of the linear model (with dummy coding of variables)…

Actually, many statistical models can be built as a variation of the linear model!

Common statistical tests are linear models

Last updated: 28 June, 2019.Also check out the [Python version](#)!

See worked examples and more details at the accompanying notebook: https://lindeloev.github.io/tests-as-linear

	Common name	Built-in function in R	Equivalent linear model in R	Exact?	The linear model in words	Icon
Simple regression: lm(y ~ 1 + x)	**y is independent of x** P: One-sample t-test N: Wilcoxon signed-rank	t.test(y) wilcox.test(y)	lm(y ~ 1) lm(signed_rank(y) ~ 1)	✓ for N >14	One number (intercept, i.e., the mean) predicts **y**. - (Same, but it predicts the *signed rank* of **y**.)	
	P: Paired-sample t-test N: Wilcoxon matched pairs	t.test(y_1, y_2, paired=TRUE) wilcox.test(y_1, y_2, paired=TRUE)	lm(y_2 - y_1 ~ 1) lm(signed_rank(y_2 - y_1) ~ 1)	✓ for N >14	One intercept predicts the pairwise **y_2-y_1** differences. - (Same, but it predicts the *signed rank* of **y_2-y_1**.)	
	y ~ continuous x P: Pearson correlation N: Spearman correlation	cor.test(x, y, method='Pearson') cor.test(x, y, method='Spearman')	lm(y ~ 1 + x) lm(rank(y) ~ 1 + rank(x))	✓ for N >10	One intercept plus **x** multiplied by a number (slope) predicts **y**. - (Same, but with *ranked* **x** and **y**)	
	y ~ discrete x P: Two-sample t-test P: Welch's t-test N: Mann-Whitney U	t.test(y_1, y_2, var.equal=TRUE) t.test(y_1, y_2, var.equal=FALSE) wilcox.test(y_1, y_2)	lm(y ~ 1 + G_2)[A] gls(y ~ 1 + G_2, weights=...[B])[A] lm(signed_rank(y) ~ 1 + G_2)[A]	✓ ✓ for N >11	An intercept for **group 1 (**plus a difference if **group 2**) predicts **y**. - (Same, but with one variance *per group* instead of one common.) - (Same, but it predicts the *signed rank* of **y**.)	
Multiple regression: lm(y ~ 1 + x_1 + x_2 +...)	P: One-way ANOVA N: Kruskal-Wallis	aov(y ~ group) kruskal.test(y ~ group)	lm(y ~ 1 + G_2 + G_3 +...+ G_N)[A] lm(rank(y) ~ 1 + G_2 + G_3 +...+ G_N)[A]	✓ for N >11	An intercept for **group 1** (plus a difference if group ≠ 1) predicts **y**. - (Same, but it predicts the *rank* of **y**.)	
	P: One-way ANCOVA	aov(y ~ group + x)	lm(y ~ 1 + G_2 + G_3 +...+ G_N + x)[A]	✓	- (Same, but plus a slope on **x**.) *Note: this is discrete AND continuous. ANCOVAs are ANOVAs with a continuous x.*	
	P: Two-way ANOVA	aov(y ~ group * sex)	lm(y ~ 1 + G_2 + G_3 +...+ G_N + S_2 + S_3 +...+ S_K + G_2*S_2 + G_3*S_3 + ... + G_N*S_K)	✓	Interaction term: changing **sex** changes the **y ~ group** parameters. *Note: **$G_{2 \text{ to } N}$** is an [indicator (0 or 1)](#) for each non-intercept levels of the **group** variable. Similarly for **$S_{2 \text{ to } K}$** for sex. The first line (with G_i) is main effect of group, the second (with S_i) for sex and the third is the **group × sex** interaction. For two levels (e.g. male/female), line 2 would just be "S_2" and line 3 would be S_2 multiplied with each G_i.*	*[Coming]*
	Counts ~ discrete x N: Chi-square test	chisq.test(groupXsex_table)	**Equivalent log-linear model** glm(y ~ 1 + G_2 + G_3 + ... + G_N + S_2 + S_3+ ... + S_K + G_2*S_2 + G_3*S_3 +...+ G_N*S_K, family=...)[A]	✓	Interaction term: (Same as Two-way ANOVA.) *Note: Run glm using the following arguments:* `glm(model, family=poisson())` *As linear-model, the Chi-square test is $\log(y_i) = \log(N) + \log(\alpha_i) + \log(\beta_j) + \log(\alpha_i\beta_j)$ where α_i and β_j are proportions. See more info in [the accompanying notebook](#).*	*Same as Two-way ANOVA*
	N: Goodness of fit	chisq.test(y)	glm(y ~ 1 + G_2 + G_3 +...+ G_N, family=...)[A]	✓	(Same as One-way ANOVA and see Chi-Square note.)	*1W-ANOVA*

List of common parametric (P) non-parametric (N) tests and equivalent linear models. The notation y ~ 1 + x is R shorthand for y = 1·b + a·x which most of us learned in school. Models in similar colors are highly similar, but really, notice how similar they *all* are across colors! For non-parametric models, the linear models are reasonable approximations for non-small sample sizes (see "Exact" column and click links to see simulations). Other less accurate approximations exist, e.g., Wilcoxon for the sign test and Goodness-of-fit for the binomial test. The signed rank function is `signed_rank = function(x) sign(x) * rank(abs(x))`. The variables G_i and S_i are ["dummy coded" indicator variables](#) (either 0 or 1) exploiting the fact that when Δx = 1 between categories the difference equals the slope. Subscripts (e.g., G_2 or y_1) indicate different columns in data. lm requires long-format data for all non-continuous models. All of this is exposed in greater detail and worked examples at https://lindeloev.github.io/tests-as-linear.

[A] See the note to the two-way ANOVA for explanation of the notation.
[B] Same model, but with one variance per group: `gls(value ~ 1 + G2, weights = varIdent(form = ~1|group), method="ML")`.

Jonas Kristoffer Lindeløv
https://lindeloev.net

https://lindeloev.github.io/tests-as-linear/#1_the_simplicity_underlying_common_tests

To the worksheet…