The General Linear Model – ANOVA Part 2

Dr Andrew J. Stewart

E: drandrewjstewart@gmail.com
T: @ajstewart_lang
G: ajstewartlang

Previously

We examined using the `afex` package for 1-way ANOVA for between subjects and within subjects (repeated measures) designs.

We used the `emmeans` package for running follow-up tests and discussed issues around the need to correct for multiple comparisons (familywise error).

We examined how to build models for factorial ANOVA and how to interpret interaction effects using `emmeans`.

Analysis of Covariance (ANCOVA)

ANCOVA can be thought of as a mix of ANOVA and regression (both of which are the GLM at their core).

One of our examples from the previous workshop 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 another factor that we are not manipulating (time spent playing computer games) might also influence the DV.

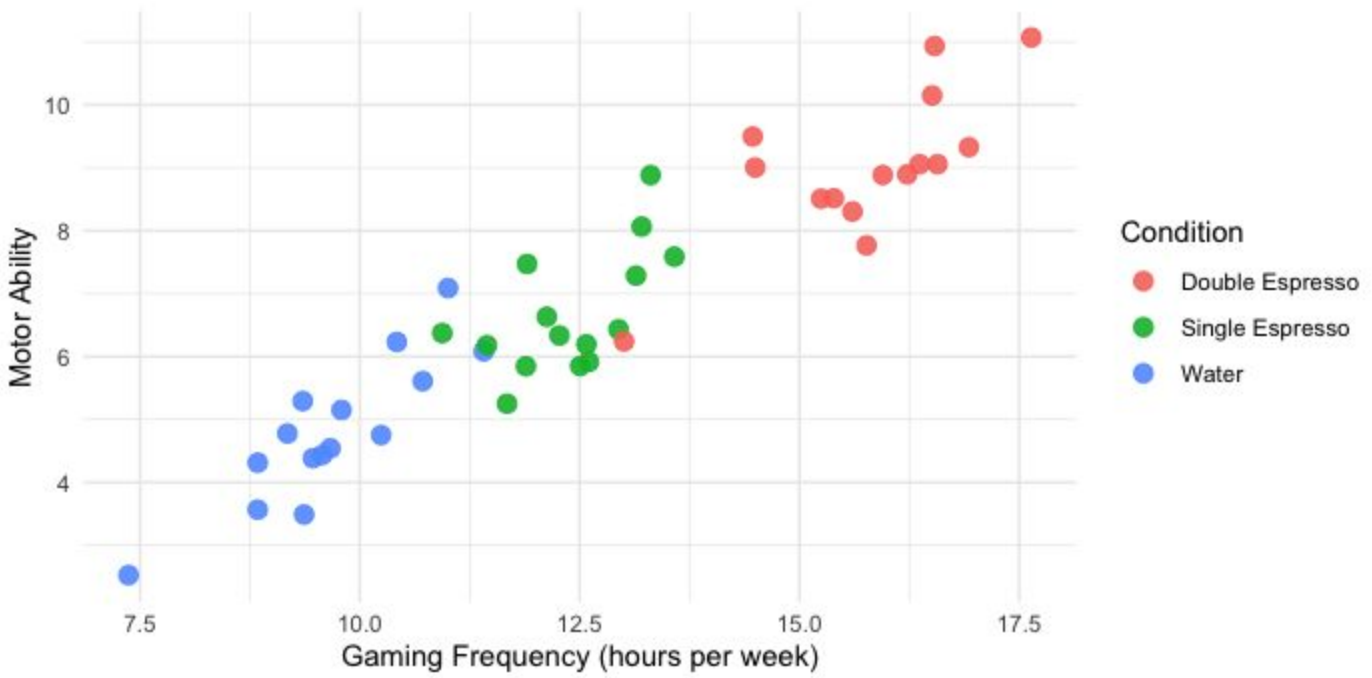
What we want is to be able to see the effect on our DV of our IV *after* we have removed the influence of computer game playing frequency.

Analysis of Covariance (ANCOVA)

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 (Motor Ability) on the y-axis, and our covariate (Gaming Frequency) on the x-axis.



So we can see there’s a relationship between our DV (Motor Ability) and our covariate (Gaming Frequency).

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

Let’s run a 1-way between participants ANOVA and initially ignore the covariate.

```
anova_model <-aov_4(Ability ~ Condition + (1 | Participant), data = my_data)
anova(anova_model)

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 might then follow this up with some pairwise comparisons.

```
> emmeans(anova_model, pairwise ~ Condition)
$emmeans
 Condition        emmean    SE df lower.CL upper.CL
 Double Espresso    9.02 0.288 42     8.43     9.60
 Single Espresso    6.69 0.288 42     6.11     7.27
 Water              4.82 0.288 42     4.24     5.40

Confidence level used: 0.95

$contrasts
 contrast                          estimate    SE df t.ratio p.value
 Double Espresso - Single Espresso     2.33 0.407 42   5.720  <.0001
 Double Espresso - Water               4.20 0.407 42  10.317  <.0001
 Single Espresso - Water               1.87 0.407 42   4.597  0.0001

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

We might then conclude we have a significant effect of Condition, and that each group differs from each other condition with the Double Espresso group scoring highest on the task, then the Single Espresso group, and then the Water group scoring lowest.

But now let’s control for the effect of our co-variate.

```
model_ancova <- aov_4(Ability ~ Gaming + Condition + (1 | Participant), data = my_data,
factorize = FALSE)
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 the covariate to our model means a lot of the variance we previously attributed to our experimental factor is actually explained by the covariate.

Adjusted Means

The mean for each group of our experimental factor (Condition) is adjusted to take into consideration the influence of our covariate within that group.

```
> emmeans(model_ancova, pairwise ~ Condition)
$emmeans
 Condition       emmean    SE df lower.CL upper.CL
 Double Espresso   6.32 0.415 41     5.48     7.16
 Single Espresso   6.87 0.193 41     6.48     7.26
 Water             7.33 0.393 41     6.53     8.12
```

These adjusted means contrast with the *unadjusted* ones which are:

```
  Condition                Mean
1 Double Espresso         9.02
2 Single Espresso         6.69
3 Water                   4.82
```

Base R `aov()` *vs.* `afex::aov_4()`

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 from the `afex` package.

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).

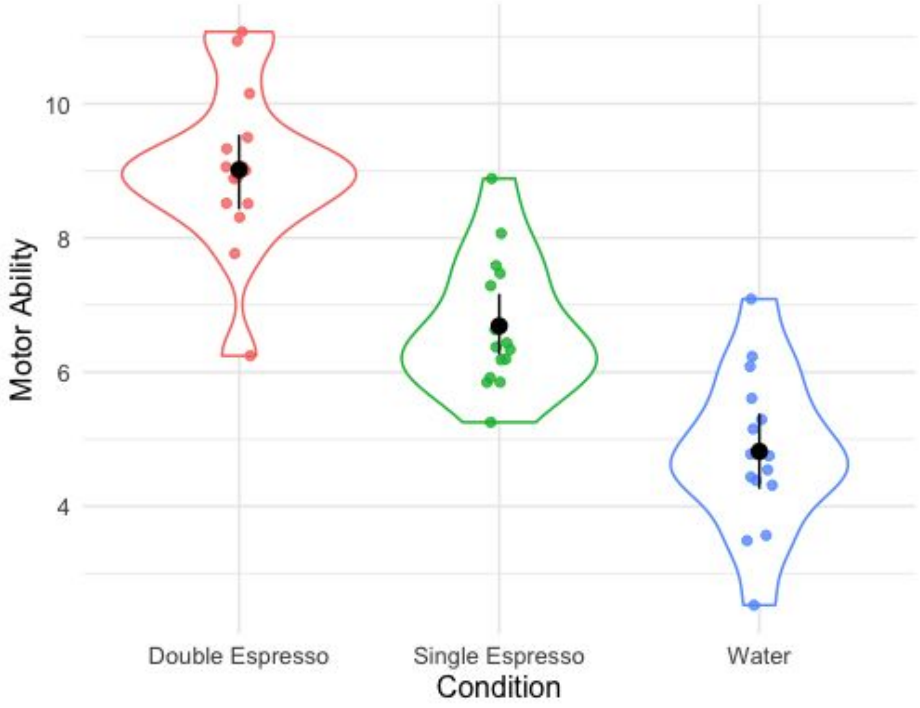
Type III Sum of Squares tests for effects adjusted for the presence of the other effects (so does not depend on the order of terms).

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).



AN(C)OVA as a special case of regression

First we use dummy (treatment) coding for the levels of our experimental factor.

```
my_data <- my_data %>%
  mutate(Condition = fct_relevel(Condition,
                     levels = c("Water", "Double Espresso", "Single Espresso")))

contrasts(my_data$Condition)
                Double Espresso Single Espresso
Water                         0               0
Double Espresso               1               0
Single Espresso               0               1
```

Ability = Intercept + β1(Double Espresso) + β2(Single Espresso)

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

AN(C)OVA as a special case of regression

Ability = Intercept + β1(Double Espresso) + β2(Single Espresso)

We want to calculate β1 and β2.

```
model_lm <- lm(Ability ~ Condition, data = my_data)
model_lm

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

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 is 4.199, and β2 is 1.871

AN(C)OVA as a special case of regression

To work out the mean Ability of our Double Espresso Group, we use the coding for the Double Espresso group (1, 0) with our equation:

Ability = Intercept + β1(Double Espresso) + β2(Single Espresso)

Ability = 4.817 + 4.199(1) + 1.871(0)
Ability = 4.817 + 4.199
Ability = 9.016

To work out the mean Ability of our Single Espresso Group, we use the coding for the Single Espresso group (0, 1) with our equation:

Ability = 4.817 + 4.199(0) + 1.871(1)
Ability = 4.817 + 1.871
Ability = 6.688

AN(C)OVA as a special case of regression

Which are the exact same means generated by the ANOVA…

AN(C)OVA as a special case of regression

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

AN(C)OVA as a special case of regression

Ability = Intercept + β1(Gaming) + β2(Double Espresso) + β3(Single Espresso)

Add the covariate to our model before the experimental factor:

```
model_ancova <- lm(Ability ~ Gaming + Condition, data = my_data)
model_ancova

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

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

AN(C)OVA as a special case of regression

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 (= -1.0085) while $\beta 3$ is the difference between the Single Espresso and Water group adjusted means (= -0.4563)

AN(C)OVA as a special case of regression

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 Espresso Group is 0.46.

AN(C)OVA as a special case of regression

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).

Ability = Intercept + β1(Gaming) + β2(Double Espresso) + β3(Single Espresso)
Ability = -3.4498 + 0.8538(12.62296) + (- 1.0085)(0) + (-0.4563)(0)
Ability = -3.4498 + 10.777
Ability = 7.33

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

AN(C)OVA as a special case of regression

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

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

A great overview by Jonas Kristoffer Lindeløv.

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