Interactions (IV)

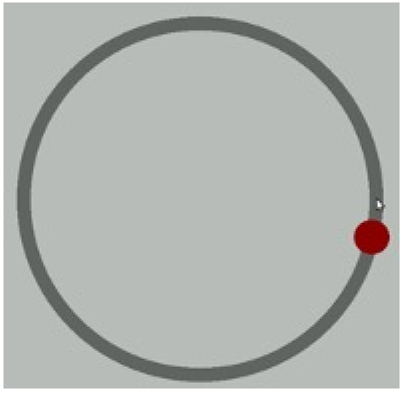
Last time...

- Factorial ANOVA

This time

- Wrapping up Factorial ANOVA
- Power

The example data are from a simulated study in which 180 participants performed an eye-hand coordination task in which they were required to keep a mouse pointer on a red dot that moved in a circular motion.



Outcome: time of 10th failure. Dot speed: .5, 1, or 1.5 revolutions. Noise: no noise, controllable noise, and uncontrollable noise. 3x3 balanced.

Marginal/cell means

Noise	Slow	Medium	Fast	Marginal
None	630.72	525.29	329.28	495.10
Controllable	576.67	492.72	287.23	452.21
Uncontrollable	594.44	304.62	268.16	389.08
Marginal	600.61	440.88	294.89	

Running the analysis of variance

```
fit = lm(Time ~ Speed*Noise, data = Data)
anova(fit)
```

```
## Analysis of Variance Table
##
## Response: Time
##              Df  Sum Sq Mean Sq  F value    Pr(>F)
## Speed         2 2805871 1402936 109.3975 < 2.2e-16 ***
## Noise         2  341315  170658  13.3075 4.252e-06 ***
## Speed:Noise   4  295720   73930   5.7649 0.0002241 ***
## Residuals   171 2192939   12824
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
lsr::etaSquared(fit)
```

```
##                  eta.sq eta.sq.part
## Speed       0.49786168   0.5613078
## Noise       0.06056150   0.1346807
## Speed:Noise 0.05247123   0.1188269
```

An effect size, η^2, provides a simple way of indexing effect magnitude for ANOVA designs, especially as they get more complex.

η^2 represents the proportion of variance in Y explained by a single factor (or interaction of factors). It is identical in its calculation to R^2:

$$\eta^2 = \frac{SS_{\text{effect}}}{SS_{\text{Y}}}$$

In an experimental design, variance in Y is created, rather than observed, by manipulating participants. The ways in which an experimenter chooses to manipulate participants can chnage variance in Y, making it difficult to compare the effect of a single manipulation across studies with different designs.

Partial η^2 helps interpretation by considering only the variance associated with an effect and random variability, which is naturally occurring

$$\text{Partial}\,\eta^2 = \frac{SS_{\text{effect}}}{SS_{\text{effect}} + SS_{\text{within}}}$$

Differences in means

In a factorial design, marginal means or cell means must be calculated in order to interpret main effects and the interaction, respectively. The confidence intervals around those means likewise are needed.

Noise	Slow	Medium	Fast	Marginal
None	630.72	525.29	329.28	495.10
Controllable	576.67	492.72	287.23	452.21
Uncontrollable	594.44	304.62	268.16	389.08
Marginal	600.61	440.88	294.89	445.46

If the homogeneity of variances assumption holds, a common estimate of score variability (MS_{within}) underlies all of the confidence intervals.

$$SE_{mean} = \sqrt{\frac{MS_{within}}{N}}$$

$$CI_{mean} = Mean \pm t_{df_{within},\alpha=.05}\sqrt{\frac{MS_{within}}{N}}$$

The sample size, N, depends on how many cases are aggregated to create the mean.

The MS_{within} is common to all calculations if HoV met.

```r
anova(fit)
```

```
## Analysis of Variance Table
##
## Response: Time
##              Df  Sum Sq Mean Sq  F value    Pr(>F)
## Speed         2 2805871 1402936 109.3975 < 2.2e-16 ***
## Noise         2  341315  170658  13.3075 4.252e-06 ***
## Speed:Noise   4  295720   73930   5.7649 0.0002241 ***
## Residuals   171 2192939   12824
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
library(emmeans)
(time_rg = ref_grid(fit))
```

```
## 'emmGrid' object with variables:
##     Speed = Slow, Medium, Fast
##     Noise = None, Controllable, Uncontrollable
```

```r
summary(time_rg)
```

```
##  Speed  Noise          prediction   SE  df
##  Slow   None                  631 25.3 171
##  Medium None                  525 25.3 171
##  Fast   None                  329 25.3 171
##  Slow   Controllable          577 25.3 171
##  Medium Controllable          493 25.3 171
##  Fast   Controllable          287 25.3 171
##  Slow   Uncontrollable        594 25.3 171
##  Medium Uncontrollable        305 25.3 171
##  Fast   Uncontrollable        268 25.3 171
```

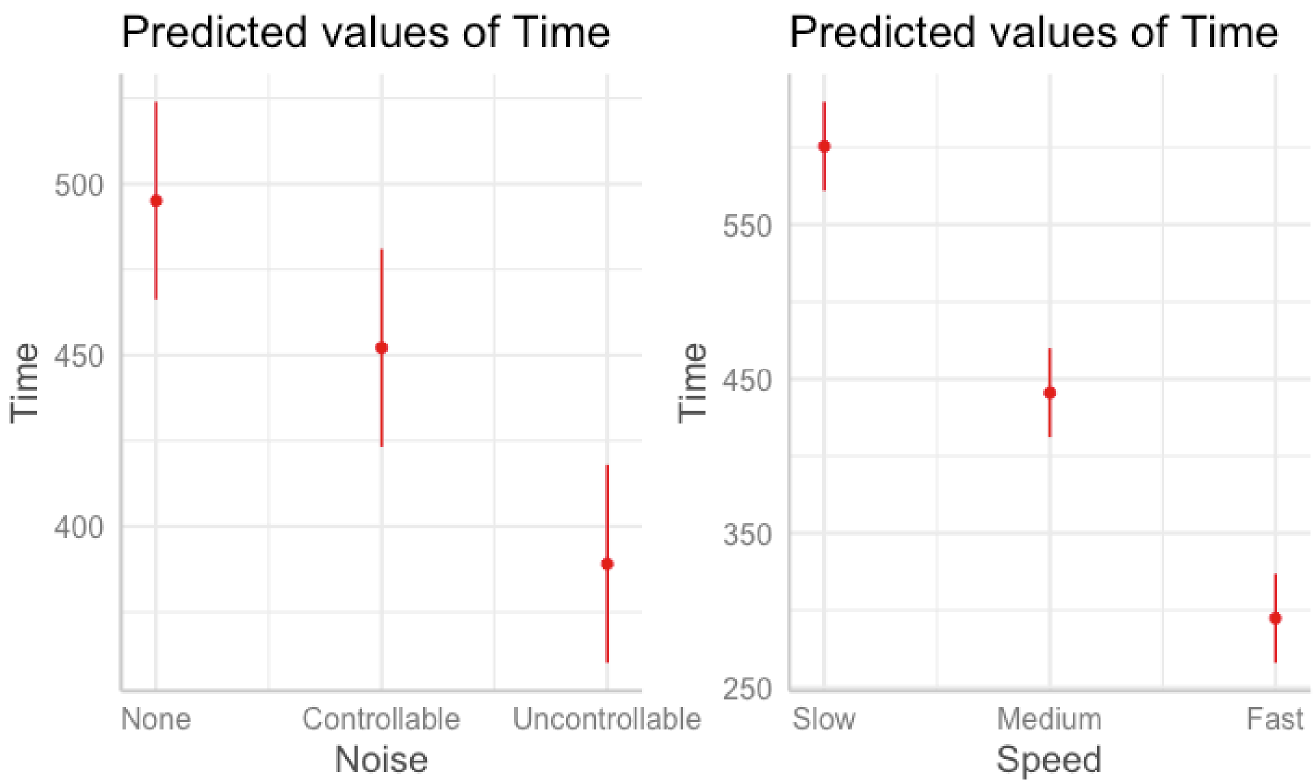
Takes in a model that's already been fitted, makes a reference grid needed to calculate the marginal means.

The `lsmeans( )` function produces marginal and cell means along with their confidence intervals. These are the marginal means for the Noise main effect.

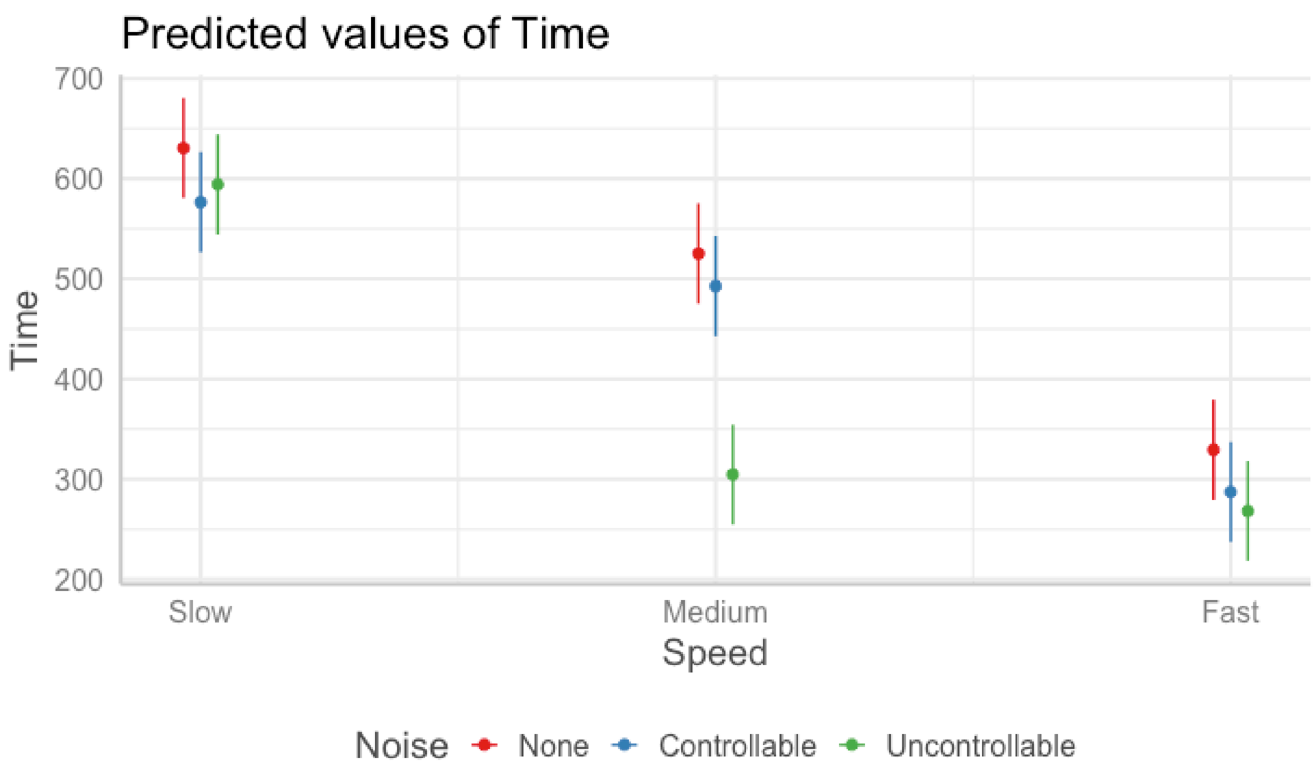
```
noise_lsm = emmeans::lsmeans(time_rg, "Noise")
noise_lsm
```

```
##  Noise            lsmean   SE  df lower.CL upper.CL
##  None                495 14.6 171      466      524
##  Controllable        452 14.6 171      423      481
##  Uncontrollable      389 14.6 171      360      418
##
## Results are averaged over the levels of: Speed
## Confidence level used: 0.95
```

```r
library(sjPlot)
noise_m = plot_model(fit, type = "emm", terms = c("Noise")) +
  theme_sjplot(base_size = 20)
speed_m = plot_model(fit, type = "emm", terms = c("Speed")) +
  theme_sjplot(base_size = 20)
```



```r
plot_model(fit, type = "pred", terms = c("Speed", "Noise")) +
  theme_sjplot(base_size = 20) +
  theme(legend.position = "bottom")
```



Precision

A reminder that comparing the confidence intervals for two means (overlap) is not the same as the confidence interval for the difference between two means.

$$SE_{\text{mean}} = \sqrt{\frac{MS_{within}}{N}}$$

$$SE_{\text{mean difference}} = \sqrt{MS_{within}[\frac{1}{N_1} + \frac{1}{N_2}]}$$

$$SE_{\text{mean difference}} = \sqrt{\frac{2MS_{within}}{N}}$$

Follow-up comparisons

Interpretation of the main effects and interaction in a factorial design will usually require follow-up comparisons. These need to be conducted at the level of the effect.

Interpretation of a main effect requires comparisons among the marginal means.

Interpretation of the interaction requires comparisons among the cell means.

The `emmeans` package makes these comparisons very easy to conduct.

Cohen's d

η^2 is useful for comparing the relative effect sizes of one factor to another. If you want to compare the differences between groups, Cohen's d is the more appropriate metric. Like in a t-test, you'll divide the differences in means by the pooled standard deviation. The pooled variance estimate is the MS_{error}

```
fit = lm(Time ~ Speed*Noise, data = Data)
anova(fit)[,"Mean Sq"]
```

```
## [1] 1402935.71  170657.62   73929.93   12824.20
```

```
MS_error = anova(fit)[,"Mean Sq"][4]
```

Cohen's d

So to get the pooled standard deviation:

```
pooled_sd = sqrt(MS_error)
pooled_sd
```

```
## [1] 113.244
```

Cohen's d

```
noise_df = as.data.frame(noise_lsm)
noise_df
```

```
##  Noise              lsmean       SE  df lower.CL upper.CL
##  None             495.0976 14.61974 171 466.2392 523.9560
##  Controllable     452.2079 14.61974 171 423.3495 481.0663
##  Uncontrollable 389.0759 14.61974 171 360.2175 417.9343
##
## Results are averaged over the levels of: Speed
## Confidence level used: 0.95
```

```
d_none_control = diff(noise_df[c(1,2), "lsmean"])/pooled_sd
d_none_control
```

```
## [1] -0.3787367
```

Cohen's d the easy way

```
em.ef <- emmeans(fit, pairwise ~ Noise)
```

```
## NOTE: Results may be misleading due to involvement in interactions
```

```
eff_size(em.ef, sigma = sigma(fit), edf = 171)
```

```
## Since 'object' is a list, we are using the contrasts already present.
```

```
##  contrast                          effect.size    SE  df lower.CL upper.
##  (None - Controllable)                   0.379 0.184 171   0.0161    0.7
##  (None - Uncontrollable)                 0.936 0.189 171   0.5622    1.3
##  (Controllable - Uncontrollable)         0.557 0.185 171   0.1922    0.9
##
## Results are averaged over the levels of: Speed
## sigma used for effect sizes: 113.2
## Confidence level used: 0.95
```

Other emmeans functions

```
em <- emmeans(fit, ~ Speed : Noise )
em
```

```
##  Speed  Noise          emmean   SE  df lower.CL upper.CL
##  Slow   None              631 25.3 171      581      681
##  Medium None              525 25.3 171      475      575
##  Fast   None              329 25.3 171      279      379
##  Slow   Controllable      577 25.3 171      527      627
##  Medium Controllable      493 25.3 171      443      543
##  Fast   Controllable      287 25.3 171      237      337
##  Slow   Uncontrollable    594 25.3 171      544      644
##  Medium Uncontrollable    305 25.3 171      255      355
##  Fast   Uncontrollable    268 25.3 171      218      318
##
## Confidence level used: 0.95
```

```
pairs(em)
```

```
##  contrast                                       estimate   SE  df t.ratio
##  Slow None - Medium None                           105.4 35.8 171   2.944
##  Slow None - Fast None                             301.4 35.8 171   8.418
##  Slow None - Slow Controllable                      54.1 35.8 171   1.509
##  Slow None - Medium Controllable                   138.0 35.8 171   3.854
##  Slow None - Fast Controllable                     343.5 35.8 171   9.592
##  Slow None - Slow Uncontrollable                    36.3 35.8 171   1.013
##  Slow None - Medium Uncontrollable                 326.1 35.8 171   9.106
##  Slow None - Fast Uncontrollable                   362.6 35.8 171  10.124
##  Medium None - Fast None                           196.0 35.8 171   5.473
##  Medium None - Slow Controllable                   -51.4 35.8 171  -1.435
##  Medium None - Medium Controllable                  32.6 35.8 171   0.910
##  Medium None - Fast Controllable                   238.1 35.8 171   6.648
##  Medium None - Slow Uncontrollable                 -69.2 35.8 171  -1.931
##  Medium None - Medium Uncontrollable               220.7 35.8 171   6.162
##  Medium None - Fast Uncontrollable                 257.1 35.8 171   7.180
##  Fast None - Slow Controllable                    -247.4 35.8 171  -6.908
##  Fast None - Medium Controllable                  -163.4 35.8 171  -4.564
##  Fast None - Fast Controllable                      42.0 35.8 171   1.174
##  Fast None - Slow Uncontrollable                  -265.2 35.8 171  -7.405
##  Fast None - Medium Uncontrollable                  24.7 35.8 171   0.689
##  Fast None - Fast Uncontrollable                    61.1 35.8 171   1.707
##  Slow Controllable - Medium Controllable            84.0 35.8 171   2.344
##  Slow Controllable - Fast Controllable             289.4 35.8 171   8.082
```

```r
noise_lsm = emmeans::lsmeans(time_rg, "Noise")
pairs(noise_lsm, adjust = "holm")
```

```
##  contrast                      estimate   SE  df t.ratio p.value
##  None - Controllable               42.9 20.7 171   2.074  0.0395
##  None - Uncontrollable            106.0 20.7 171   5.128  <.0001
##  Controllable - Uncontrollable     63.1 20.7 171   3.053  0.0052
##
## Results are averaged over the levels of: Speed
## P value adjustment: holm method for 3 tests
```

Assumptions

You can check the assumptions of the factorial ANOVA in much the same way you check them for multiple regression; but given the categorical nature of the predictors, some assumptions are easier to check.

Assumptions

Homogeneity of variance, for example, can be tested using Levene's test, instead of examining a plot.

```
library(car)
leveneTest(Time ~ Speed*Noise, data = Data)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group   8  0.5879  0.787
##       171
```

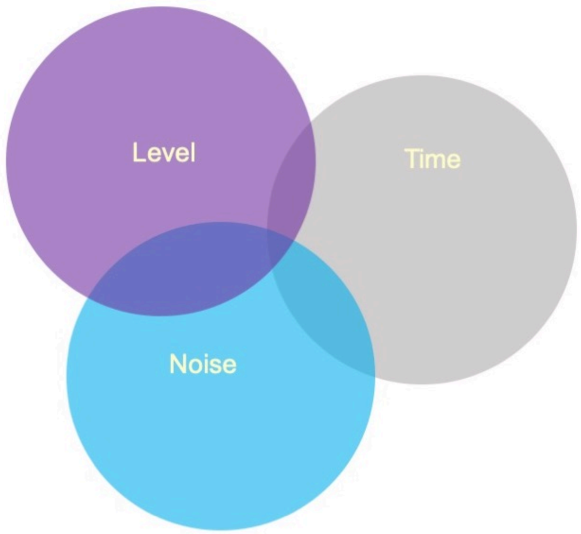
If designs are balanced, then the main effects and interaction effects are independent/orthogonal. In other words, knowing what condition a case is in on Variable 1 will not make it any easier to guess what condition they were part of in Variable 2.

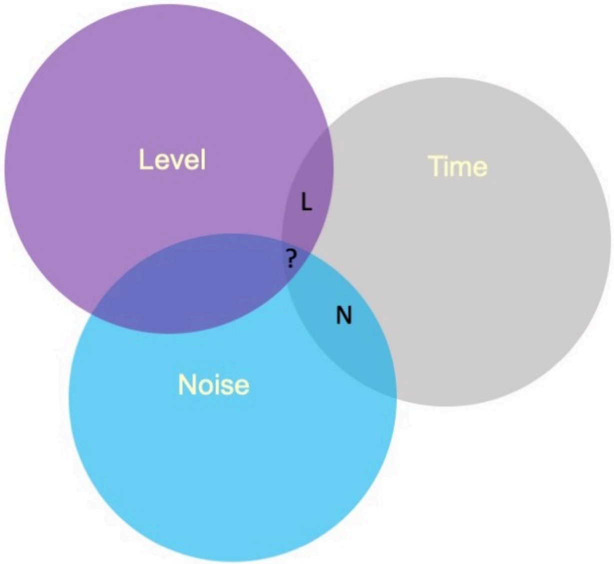
However, if your design is unbalanced, the main effects and interaction effect are partly confounded.

```
table(Data2$Level, Data2$Noise)
```

```
##
##         Controllable Uncontrollable
##   Soft            10             30
##   Loud            20             20
```

The overlap represents the variance shared by the variables. Now there is variance accounted for in the outcome (Time) that cannot be unambiguously attributed to just one of the predictors There are several options for handling the ambiguous regions.





Overlap handled in 3 ways: Type I, Type II, and Type III sums of squares.

All credit goes to Matt Cooper (no relation!) and this great blog post. I'm going to paraphrase from his post since I think he does a great job making it clear -- you can check it out here.

Sums of Squares

$SS(A, B, AB)$ is sum of squares for both main effects and interaction

$SS(A, B)$ is sum of squares for a model with no interaction

$SS(B, AB)$ is sum of squares for model that does not account for the effects from factor A

Sum of Squares

> The influence of particular factors (including interactions) can be tested by examining the differences between models. For example, to determine the presence of an interaction effect, an F-test of the models $SS(A, B, AB)$ and the no-interaction model $SS(A, B)$ would be carried out.

Read the pipe | as "after"; $SS(AB|A, B)$ is *"the sum of squares for interaction after the main effects"*

Type 1 SS (aka sequential)

Test for main effect of A, followed by main effect of B, followed by interaction. ORDER MATTERS! The overlapping part of the venn diagram goes to whichever factor is listed first (a priority rule). Rule needs to be justified or else you might be p-hacking.

Effect	Type I	Type II	Type III
A	SS(A)		
B	SS(B|A)		
A:B	SS(AB|A,B)		

If a design is quite unbalanced, different orders of effects can produce quite different results.

```
fit_1 = aov(Time ~ Noise + Level, data = Data2)
summary(fit_1)
```

```
##              Df  Sum Sq Mean Sq F value   Pr(>F)    
## Noise         1  579478  579478   41.78 8.44e-09 ***
## Level         1  722588  722588   52.10 3.20e-10 ***
## Residuals    77 1067848   13868                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lsr::etaSquared(fit_1, type = 1)
```

```
##          eta.sq eta.sq.part
## Noise 0.2445144   0.3517689
## Level 0.3049006   0.4035822
```

If a design is quite unbalanced, different orders of effects can produce quite different results.

```
fit_1 = aov(Time ~ Level + Noise, data = Data2)
summary(fit_1)
```

```
##             Df  Sum Sq Mean Sq F value   Pr(>F)    
## Level        1 1035872 1035872   74.69 5.86e-13 ***
## Noise        1  266194  266194   19.20 3.68e-05 ***
## Residuals   77 1067848   13868                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lsr::etaSquared(fit_1, type = 1)
```

```
##          eta.sq eta.sq.part
## Level 0.4370927   0.4924002
## Noise 0.1123222   0.1995395
```

Type 2 SS (aka hierarchical)

Test for each of the main effects after the other main effect. **No signifiant interaction is assumed**. So you'll need to first test for your interaction. If it's not significant, then you can examine your main effects with Type 2 SS.

Effect	Type I	Type II	Type III
A	SS(A)	SS(A|B)	
B	SS(B|A)	SS(B|A)	
A:B	SS(AB|A,B)	---	

Type 3 SS

Tests the presence of a main effect after the other main effect and interaction. Basically, the overlapping area of the venn diagram goes unclaimed. Useful for interactions. Less useful for main effect (what does a main effect tell you if you have a significant interaction?)

Effect	Type I	Type II	Type III
A	SS(A)	SS(A|B)	SS(A|B,AB)
B	SS(B|A)	SS(B|A)	SS(B|A,AB)
A:B	SS(AB|A,B)	SS(AB|A,B)	SS(AB|A,B)

The `aov( )` function in R produces Type I sums of squares. The `Anova( )` function from the `car` package provides Type II and Type III sums of squares.

```
Anova(aov(fit), type = "II")
```

```
## Anova Table (Type II tests)
##
## Response: Time
##              Sum Sq  Df  F value    Pr(>F)
## Speed       2805871   2 109.3975 < 2.2e-16 ***
## Noise        341315   2  13.3075 4.252e-06 ***
## Speed:Noise  295720   4   5.7649 0.0002241 ***
## Residuals   2192939 171
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

All of the between-subjects variance is accounted for by an effect in Type I sums of squares. The sums of squares for each effect and the residual will equal the total sum of squares.

For Type II and Type III sums of squares, the sums of squares for effects and residual will be less than the total sum of squares. Some variance (in the form of SS) goes unclaimed.

The highest order effect (assuming standard ordering) has the same SS in all three models.

When a design is balanced, Type I, II, and III sums of squares are equivalent.

Summary (ANOVA and Regression)

Factorial ANOVA is the method by which we can examine whether two (or more) categorical IVs have joint effects on a continuous outcome of interest.

Like all general linear models, factorial ANOVA is a specific case of multiple regression. However, we may choose to use an ANOVA framework for the sake of interpretability.

Summary (ANOVA and Regression)

Factorial ANOVA

Interaction tests whether there are differences in differences.

A main effect is the effect of one IV on the DV **ignoring the other variable(s)**.

Regression

Interaction tests whether slope changes.

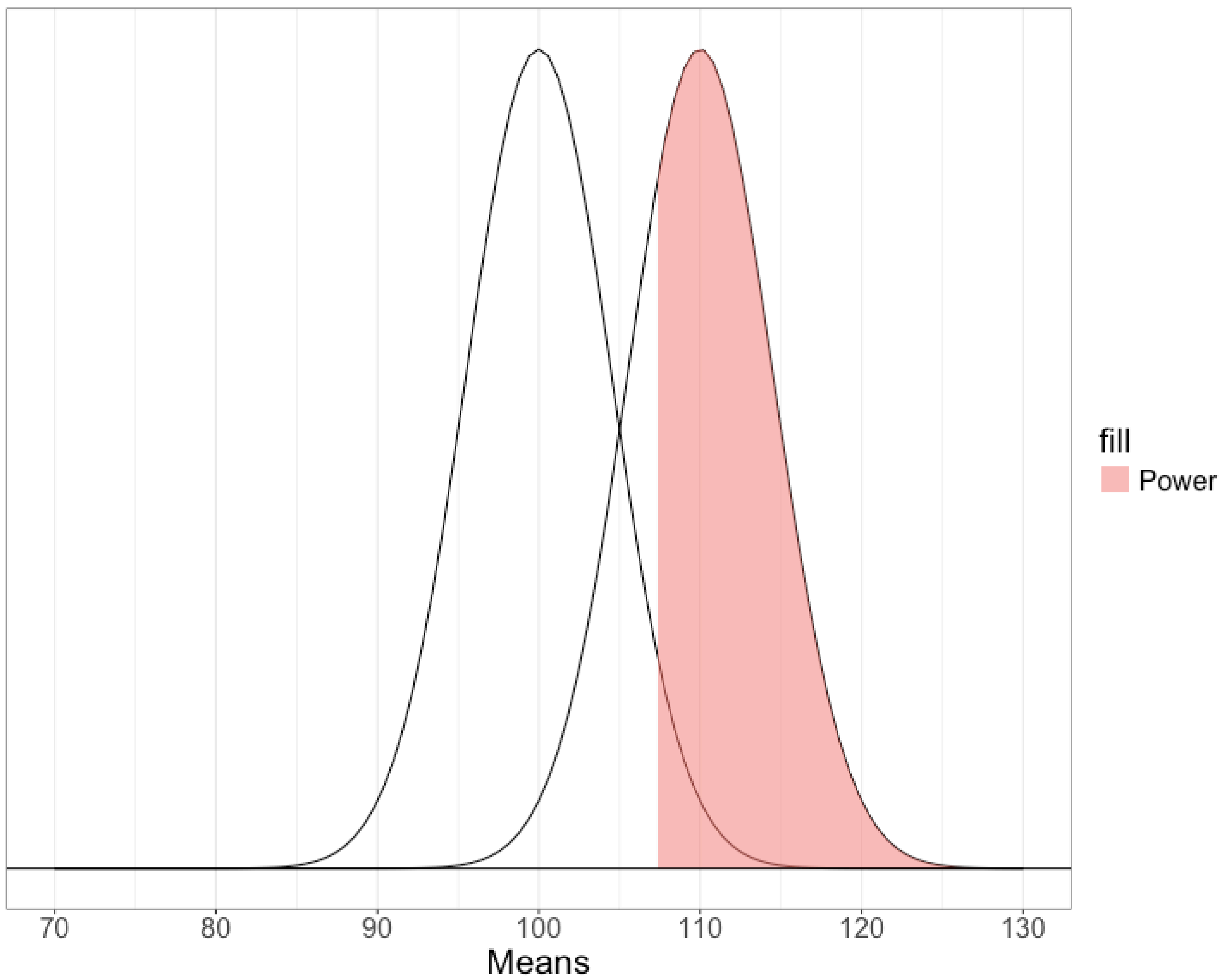
A conditional effect is the effect of IV on the DV **assuming all the other variables are 0**.

Power

What is statistical power? How can we increase power?

The likelihood of finding an effect *if the effect actually exists.* Power gets larger as we:

- increase our sample size
- reduce (error) variance
- raise our Type I error rate
- study larger effects


70
80
90
100
110
120
130
Means
fill
Power

Power in multiple regression (additive effects)

When calculating power for the omnibus test, use the expected multiple R^2 value to calculate an effect size:

$$f^2 = \frac{R^2}{1 - R^2}$$

Omnibus power

```r
R2 = .10
(f = R2/(1-R2))
```

```
## [1] 0.1111111
```

Omnibus power

```
library(pwr)
pwr.f2.test(u = 3, # number of predictors in the model
            f2 = f,
            sig.level = .05, #alpha
            power =.90) # desired power
```

```
##
##      Multiple regression power calculation
##
##               u = 3
##               v = 127.5235
##              f2 = 0.1111111
##       sig.level = 0.05
##           power = 0.9
```

v is the denominator df of freedom, so the number of participants needed is v + k (number of predictors) + 1.

Coefficient power

To estimate power for a single coefficient, you need to consider (1) how much variance is accounted for by just the variable and (2) how much variance you'll account for in Y overall.

$$f^2 = \frac{R^2_Y - R^2_{Y.X}}{1 - R^2_Y}$$

Coefficient power

```
R2 = .10
RX1 = .03
(f = (R2-RX1)/(1-R2))
```

```
## [1] 0.07777778
```

Coefficient power

```
pwr.f2.test(u = 3, # number of predictors in the model
            f2 = f,
            sig.level = .05, #alpha
            power =.90) # desired power
```

```
##
##      Multiple regression power calculation
##
##               u = 3
##               v = 182.1634
##              f2 = 0.07777778
##       sig.level = 0.05
##           power = 0.9
```

v is the denominator df of freedom, so the number of participants needed is v + k + 1.

Next time...

Powering interactions

Polynomials