Course Business

- No class next week!
- Two weeks from now (10/21):
  - Midterm assignment due
  - Discuss papers & pros and cons of mixed effect models

- Variance in random effects
  - This is estimated variance in the population

- Interactions & main effects
  - Interaction of aphasia & passive sentence is the defined as special effect of having aphasia and a passive sentence—beyond either alone. Doesn’t make sense to consider interactions without main effect.
Week 7: Coding Predictors II

- Factors with More than 2 Levels
- Example: Treatment Coding
- Problem of Multiple Comparisons
- Orthogonal Contrasts
  - Example
  - Implementation
  - Definition
- Recap of Coding Systems
- Alternatives & Addenda
Alice in Um-derland (Fraundorf & Watson, 2011)


- How do disfluencies in speech (e.g., “uh”, “um”) change listener comprehension?

- Disfluencies more common before *more difficult material*, so might lead listeners to pay more attention

- But: Any benefit might be confounded with just *having more time* to process
  - Control: Speaker coughing, matched in duration
**Alice in Um-derland** *(Fraundorf & Watson, 2011)*


- Each participant hears stories based on *Alice in Wonderland*
  - Later, test recall (correct vs not) of 42 different plot elements

- **Conditions:**
  - Some plot points told fluently (control)
  - Some preceded by speech filler
  - Some by coughs matched in duration to the fillers
  - Each subject hears some points in all 3 conditions
  - Each item heard in all 3 conditions across subjects
Alice in *Um-derland* *(Fraundorf & Watson, 2011)*

- Mean performance in each condition:
  - `tapply(disfluency$Recalled, disfluency $InterruptionType, mean)`
  - “Take Recalled, separate it out by InterruptionType, and give me the mean”

<table>
<thead>
<tr>
<th>Condition</th>
<th>Mean Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control</td>
<td>0.61666667</td>
</tr>
<tr>
<td>Cough</td>
<td>0.6595238</td>
</tr>
<tr>
<td>Filler</td>
<td>0.7595238</td>
</tr>
</tbody>
</table>
Factors with More Than 2 Levels

- How can we code a variable with three categories?
  - Fluent = 0, Cough = 1, Filler = 2?
- Let’s imagine the equations:

  Fluent: $\text{Logit} = \gamma_{00} + \gamma_{100} \times \text{InterruptionType}$

  Cough: $\text{Logit} = \gamma_{00} + \gamma_{100} \times \text{InterruptionType}$

  Filler: $\text{Logit} = \gamma_{00} + \gamma_{100} \times \text{InterruptionType}$
**Factors with More Than 2 Levels**

- How can we code a variable with three categories?
  - Fluent = 0, Cough = 1, Filler = 2?
- Let’s imagine the equations:

<table>
<thead>
<tr>
<th>Category</th>
<th>Logit Formula</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fluent</td>
<td>$\logit = \gamma_{000} + \gamma_{100} \times \text{Fluent}$</td>
<td>0</td>
</tr>
<tr>
<td>Cough</td>
<td>$\logit = \gamma_{000} + \gamma_{100} \times \text{Cough}$</td>
<td>1</td>
</tr>
<tr>
<td>Filler</td>
<td>$\logit = \gamma_{000} + \gamma_{100} \times \text{Filler}$</td>
<td>2</td>
</tr>
</tbody>
</table>
Factors with More Than 2 Levels

• How can we code a variable with three categories?
  • Fluent = 0, Cough = 1, Filler = 2?
  • Let’s imagine the equations:

Fluent  \[ \text{Logit} = \gamma_{000} + \gamma_{100} \times \begin{array}{c} 0 \\ \downarrow \text{Differ by 1} \gamma_{100} \\ 1 \end{array} \]

Cough \[ \text{Logit} = \gamma_{000} + \gamma_{100} \times \begin{array}{c} 0 \\ \downarrow \text{Differ by 1} \gamma_{100} \\ 1 \end{array} \]

Filler \[ \text{Logit} = \gamma_{000} + \gamma_{100} \times \begin{array}{c} 0 \\ \downarrow \text{Differ by 1} \gamma_{100} \\ 2 \end{array} \]

• This coding scheme assumes Fluent & Cough differ by the same amount as Cough & Filler
  • Probably not true. Not a safe assumption
Factors With More Than 2 Levels

- To actually represent three levels, we need two sets of codes
  - “InterruptionType1” and “InterruptionType2”

- If a factor has 3 levels, R automatically creates multiple sets of codes
  - `contrasts(disfluency$InterruptionType)`

One set of codes (“InterruptionType1”). 1 for Cough, 0 for everything else.

Another, different set of codes (“InterruptionType2”). 1 for Filler, 0 for everything else.
Factors With More Than 2 Levels

• Annoying R “feature”: If you take a subset that includes only some levels...

  \[
  \text{disfluency.NoCoughs} \leftarrow \text{subset(\text{disfluency, InterruptionType} \neq \text{'Cough'})}
  \]

• ...R still remembers all of the possible levels...

• Solution: Re-make into a factor with \text{factor}():

  \[
  \text{disfluency.NoCoughs}\$\text{InterruptionType} \leftarrow \text{factor(\text{disfluency.NoCoughs}\$\text{InterruptionType})}
  \]
Week 7: Coding Predictors II

- Factors with More than 2 Levels
  - Example: Treatment Coding
  - Problem of Multiple Comparisons
- Orthogonal Contrasts
  - Example
  - Implementation
  - Definition
- Recap of Coding Systems
- Alternatives & Addenda
Treatment Coding With >2 Levels

• The two sets of codes are 2 separate variables in the underlying regression equation:

**Fluent**

\[ \text{Logit} = \gamma_{000} + \gamma_{100} \times \text{InterruptionType1} + \gamma_{200} \times \text{InterruptionType2} \]

**Cough**

\[ \text{Logit} = \gamma_{000} + \gamma_{100} \times \text{InterruptionType1} + \gamma_{200} \times \text{InterruptionType2} \]

**Filler**

\[ \text{Logit} = \gamma_{000} + \gamma_{100} \times \text{InterruptionType1} + \gamma_{200} \times \text{InterruptionType2} \]
**Treatment Coding With >2 Levels**

- The two sets of codes are 2 separate variables in the underlying regression equation:

  - Fluent: \[ \text{Logit} = y_{000} + y_{100} \times 0 + y_{200} \times 0 \]
  - Cough: \[ \text{Logit} = y_{000} + y_{100} \times \text{InterruptionType1} + y_{200} \times \text{InterruptionType2} \]
  - Filler: \[ \text{Logit} = y_{000} + y_{100} \times \text{InterruptionType1} + y_{200} \times \text{InterruptionType2} \]
Treatment Coding With >2 Levels

- The two sets of codes are 2 separate variables in the underlying regression equation:

Fluent

\[ \text{Logit} = y_{000} \]

Once again, the intercept is just performance in the baseline level (the one coded with all 0s)

Cough

\[ \text{Logit} = y_{000} + y_{100} \times 1 + y_{200} \times 0 \]

Filler

\[ \text{Logit} = y_{000} + y_{100} \times \text{InterruptionType1} + y_{200} \times \text{InterruptionType2} \]
Treatment Coding With >2 Levels

- The two sets of codes are 2 separate variables in the underlying regression equation:

  Fluent: \[ \text{Logit} = \gamma_{000} \]
  
  Cough: \[ \text{Logit} = \gamma_{000} + \gamma_{100} \]
  
  Filler: \[ \text{Logit} = \gamma_{000} + \gamma_{100} \times 0 + \gamma_{200} \times 1 \]

  Once again, the intercept is just performance in the baseline level (the one coded with all 0s)

  Interruption Type 1 = Difference between fluent story & coughs
Treatment Coding With >2 Levels

- The two sets of codes are 2 separate variables in the underlying regression equation:

  - Fluent: \( \text{Logit} = \gamma_{000} \)
  - Cough: \( \text{Logit} = \gamma_{000} + \gamma_{100} \)
  - Filler: \( \text{Logit} = \gamma_{000} + \gamma_{200} \)

  Once again, the intercept is just performance in the baseline level (the one coded with all 0s)

  - InterruptionType1 = Difference between fluent story & coughs
  - InterruptionType2 = Difference between fluent story & fillers
Treatment Coding: Model

• We want to analyze whether or not each plot point was recalled based on the InterruptionType condition
  • Each subject hears some points in all 3 conditions
  • Each item heard in all 3 conditions across subjects

• What’s wrong with this picture?:
  • `dummycode.Maximal <- lmer(Recalled ~ 1 + InterruptionType + (1 + InterruptionType|Subject) + (1|Item), data=disfluency)`
Treatment Coding: Model

- We want to analyze whether or not each plot point was recalled based on the InterruptionType condition
  - Each subject hears some points in all 3 conditions
  - Each item heard in all 3 conditions across subjects

- Correct model call:
  - `dummycode.Maximal <- glmer(Recalled ~ 1 + InterruptionType + (1 + InterruptionType|Subject) + (1 + InterruptionType|Item), data=disfluency, family=binomial)`
Treatment Coding: Results

Intercept: Log odds of recall in the control condition

Cough effect: Marginally greater recall with coughs than control fluent condition

Filler effect: Greater recall with speech fillers than control fluent condition

See Levy reading on CourseWeb for how to do the likelihood ratio test
Week 7: Coding Predictors II

- Factors with More than 2 Levels
- Example: Treatment Coding
- Problem of Multiple Comparisons
- Orthogonal Contrasts
  - Example
  - Implementation
  - Definition
- Recap of Coding Systems
- Alternatives & Addenda
The Problem of Multiple Comparisons

- It would be nice to have a direct comparison of fillers vs. coughs.
- Actually, there are a lot of other comparisons we could consider...
  - Fillers vs. coughs
  - Fluent story vs. any kind of interruption
  - Fillers vs. mean performance in this task
  - Cough vs. mean performance in this task
The Problem of Multiple Comparisons

- But: These comparisons aren’t all independent
- Our model already showed us:
  - Log odds of recall are 0.18 logits greater for **coughs** than **fluent**
  - Log odds of recall are 0.68 logits greater for **fillers** than **fluent**
  - Difference between **coughs** and **fillers** is already known: 0.68 – 0.18 = 0.50
The Problem of Multiple Comparisons

• But: These comparisons aren’t all independent
• Our model already showed us:
  • Log odds of recall are 0.18 logits greater for **coughs** than **fluent**
  • Log odds of recall are 0.68 logits greater for **fillers** than **fluent**
  • Difference between **coughs** and **fillers** is *already known*: \(0.68 - 0.18 = 0.50\)
• In general: With \(g\) levels, \(g-1\) comparisons fully describe data
  • Could position all \(g\) conditions on a number line just based on \(g-1\) comparisons
The Problem of Multiple Comparisons

• Every comparison between levels has some possibility of type I error (sampling error)

Maybe we underestimated performance in the fluent control condition

\( \alpha = .05 \)
The Problem of Multiple Comparisons

• Every comparison between levels has some possibility of type I error (sampling error)
• If wrong, other similar comparisons have a higher probability of being wrong
• They’re not independent

Maybe we underestimated performance in the fluent control condition

Would affect both the fluent vs coughs and fluent vs fillers comparisons

\[ \alpha > .05 \]
The Problem of Multiple Comparisons

- Every comparison between levels has some possibility of type I error (sampling error)
- If wrong, other similar comparisons have a higher probability of being wrong
  - They’re not independent
- $p$-values are too low

$\alpha > .05$
The Problem of Multiple Comparisons

- Going back to our treatment coding results...

- Cough vs fluent and filler vs fluent effects are somewhat positively correlated.

- If we underestimate performance in the fluent condition (for instance), both comparisons would be affected.
The Problem of Multiple Comparisons

- Also, running more than $g-1$ comparisons can result in nonsensical / paradoxical results

- Condition C > Condition A, $p < .05$
- Condition A and Condition B don’t significantly differ
- Condition B and Condition C don’t significantly differ
Here Comes Trouble!

- These reasons are why R doesn’t perform all possible comparisons between all levels
  - Statisticians who designed R would point out that $g-1$ comparisons fully describe everything
- But, standard practice in some fields IS to have every comparison between levels
  - Not fully correct statistically, but expected by readers/reviewers
  - Mixed effects models may not be giving you what you want
- Case where statistics & field standards conflict!
Week 7: Coding Predictors II

- Factors with More than 2 Levels
- Example: Treatment Coding
- Problem of Multiple Comparisons
  - Orthogonal Contrasts
    - Example
    - Implementation
    - Definition
- Recap of Coding Systems
- Alternatives & Addenda
Orthogonal Contrasts

- Another set of comparisons...

CONTRAST 1
- FLUENT CONTROL
- COUGHS
- FILLERS

Do coughs and fillers differ?

CONTRAST 2

Do interruptions (in general) differ from fluent speech?
Orthogonal Contrasts

- These comparisons are independent ("orthogonal")
- Knowing that interruptions differ from fluent speech doesn’t tell us *anything* about which type (if any) is better

**Contrast 1**

Do coughs and fillers differ?

**Contrast 2**

Do interruptions (in general) differ from fluent speech?
Orthogonal Contrasts

- In each contrast, compares the positive-coded level(s) to the negative-coded level(s)
- Ignore the level(s) coded as zero

**CONTRAST 1**

- FLUENT CONTROL
- COUGHS
- FILLERS

- 0
- -1/2
- 1/2

Do coughs and fillers differ?

Centered around mean of 0!

**CONTRAST 2**

Do interruptions (in general) differ from fluent speech?
Orthogonal Contrasts

• In each contrast, compares the positive-coded level(s) to the negative-coded level(s)
• Ignore the level(s) coded as zero

CONTRAST 1

FLUENT CONTROL

COUGHS

FILLERS

CONTRAST 2

Do coughs and fillers differ?

Centered around mean of 0!

Do interruptions (in general) differ from fluent speech?
Orthogonal Contrasts

- In each contrast, compares the positive-coded level(s) to the negative-coded level(s)
- Ignore the level(s) coded as zero

**CONTRAST 1**
- FLUENT CONTROL
  - 0
- COUGHS
  - -1/2
- FILLERS
  - 1/2

Centered around mean of 0!

**CONTRAST 2**
- FLUENT CONTROL
  - -2/3
- COUGHS
  - 1/3
- FILLERS
  - 1/3

Do coughs and fillers differ?

Do interruptions (in general) differ from fluent speech?
Week 7: Coding Predictors II

- Factors with More than 2 Levels
- Example: Treatment Coding
- Problem of Multiple Comparisons
- Orthogonal Contrasts
  - Example
  - Implementation
  - Definition
- Recap of Coding Systems
- Alternatives & Addenda
Let’s Change the Contrasts

• As before, we use `<-` to change the contrasts

<table>
<thead>
<tr>
<th></th>
<th>Cough</th>
<th>Filler</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Cough</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Filler</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

• Now, we’re trying to create a \textbf{matrix} of numbers

• Need to stick two columns together with \texttt{cbind}:

\begin{verbatim}
contrasts(disfluency$InterruptionType) <- cbind(c(0,-1/2,1/2), c(-2/3, 1/3,1/3))
\end{verbatim}

<table>
<thead>
<tr>
<th></th>
<th>[,1]</th>
<th>[,2]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control</td>
<td>0.0</td>
<td>-0.6666667</td>
</tr>
<tr>
<td>Cough</td>
<td>-0.5</td>
<td>0.3333333</td>
</tr>
<tr>
<td>Filler</td>
<td>0.5</td>
<td>0.3333333</td>
</tr>
</tbody>
</table>
Naming the Contrasts

• Default contrast names are just “1” and “2”

• We can change the names of these columns with `colnames()`
  • `colnames(contrasts(disfluency $InterruptionType))) <- c('FillerVsCough', 'InterruptionVsFluent')`

• Optional—it just makes the output easier to read
Orthogonal Contrasts: Results

- summary(orthogonal.Maximal)

| Fixed effects:                                      | Estimate | Std. Error | z value | Pr(>|z|) |
|-----------------------------------------------------|----------|------------|---------|----------|
| (Intercept)                                          | 0.76674  | 0.04787    | 16.018  | < 2e-16  *** |
| InterruptionTypeFillerVsCough                       | 0.49557  | 0.11041    | 4.489   | 7.17e-06 *** |
| InterruptionTypeInterruptionVsFluent                | 0.43202  | 0.09010    | 4.795   | 1.63e-06 *** |

---

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:

- (Intr) InTFVC
- IntrrpTFVC 0.125
- IntrrpTIVF 0.058

Note that the two contrasts are now uncorrelated. They’re independent / orthogonal.

**Intercept:** Mean performance across conditions (remember this is in terms of log odds not probability)

**Contrast 1:** Fillers produce higher recall than coughs

**Contrast 2:** Speech with pauses/interruptions better remembered than totally fluent speech
Which Model Fits Better?

- `summary(dummycode.Maximal)`

<table>
<thead>
<tr>
<th></th>
<th>AIC</th>
<th>BIC</th>
<th>logLik</th>
<th>deviance</th>
<th>df.resid</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3151.4</td>
<td>3238.9</td>
<td>-1560.7</td>
<td>3121.4</td>
<td>2505</td>
</tr>
</tbody>
</table>

- `summary(orthogonal.Maximal)`

<table>
<thead>
<tr>
<th></th>
<th>AIC</th>
<th>BIC</th>
<th>logLik</th>
<th>deviance</th>
<th>df.resid</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3151.4</td>
<td>3238.9</td>
<td>-1560.7</td>
<td>3121.4</td>
<td>2505</td>
</tr>
</tbody>
</table>

- Changing coding schemes will **not** change the overall fit of the model
  - The same information is available to the model either way
  - We’re just dividing it up differently
Why -0.5 & 0.5?

<table>
<thead>
<tr>
<th>FILLER</th>
<th>CONTRAST CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>.5</td>
<td>}1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COUGH</th>
<th>CONTRAST CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>-.5</td>
<td>}1</td>
</tr>
</tbody>
</table>

1 unit change in contrast IS the difference between conditions

<table>
<thead>
<tr>
<th>FILLER</th>
<th>CONTRAST CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>}2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COUGH</th>
<th>CONTRAST CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>}2</td>
</tr>
</tbody>
</table>

1 unit change in contrast IS only *half* of the difference between conditions
Why -0.5 & 0.5?

- What if we used \((0, -1, 1)\) instead?
- Doesn't affect **significance test**
- Does make it harder to interpret the estimate
  - Parameter estimate is only half of the actual difference between conditions

**CONTRAST 1:**
c(0, -0.5, 0.5)

| Parameter Estimate | Std. Error | z value | Pr(>|z|) |
|--------------------|------------|---------|----------|
| (Intercept)        | 0.76674    | 0.04787 | 16.018   |
|                    |            |         | < 2e-16  |
| InterruptionTypeFillerVsCough | 0.49557    | 0.11041 | 4.489    |
|                    |            |         | 7.17e-06 |
|                    |            |         | ***      |
| InterruptionTypeInterruptionVsFluent | 0.43202    | 0.09010 | 4.795    |
|                    |            |         | 1.63e-06 |
|                    |            |         | ***      |

**CONTRAST 1:**
c(0, -1, 1)

| Parameter Estimate | Std. Error | z value | Pr(>|z|) |
|--------------------|------------|---------|----------|
| (Intercept)        | 0.76674    | 0.04787 | 16.019   |
|                    |            |         | < 2e-16  |
| InterruptionTypeFillerVsCough | 0.24779    | 0.05520 | 4.489    |
|                    |            |         | 7.17e-06 |
|                    |            |         | ***      |
| InterruptionTypeInterruptionVsFluent | 0.43200    | 0.09010 | 4.795    |
|                    |            |         | 1.63e-06 |
|                    |            |         | ***      |
Factors with More than 2 Levels
Example: Treatment Coding
Problem of Multiple Comparisons
Orthogonal Contrasts
  Example
  Implementation
  Definition
Recap of Coding Systems
Alternatives & Addenda
What Makes Contrasts Orthogonal?
**What Makes Contrasts Orthogonal?**

- **Criterion 1:** Codes within contrast sum to 0
- **and Criterion 2:**
  - Multiply codes for each level across contrasts
  - Then sum across the levels
  - Needs to sum to 0

---

**FLUENT**

<table>
<thead>
<tr>
<th>CONTRAST 1</th>
<th>CONTRAST 2</th>
<th>PRODUCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-0.67</td>
<td>0</td>
</tr>
<tr>
<td>-0.5</td>
<td>0.33</td>
<td>-0.165</td>
</tr>
<tr>
<td>0.5</td>
<td>0.33</td>
<td>0.165</td>
</tr>
</tbody>
</table>

= 0
= 0
= 0

Yes, orthogonal!
What Makes Contrasts Orthogonal?

• **Criterion 1:** Codes within contrast sum to 0
• **and Criterion 2:**
  • Multiply codes for each level across contrasts
  • Then sum across the levels
  • Needs to sum to 0

No, not orthogonal
What Makes Contrasts Orthogonal?

- **Criterion 1:** Codes within contrast sum to 0
- **and Criterion 2:**
  - Multiply codes for each level across contrasts
  - Then sum across the levels
  - Needs to sum to 0

Treatment codes are not orthogonal!
What Makes Contrasts Orthogonal?

- Multiply codes for each level across contrasts
- Then sum across the levels
- Needs to sum to 0
- Codes within a contrast must also sum to 0

- Interpretation given earlier...
  - Each contrast compares the + and – levels
  - And ignores the 0-coded levels
  - ...is valid only if each pair of contrasts is orthogonal
Week 7: Coding Predictors II

- Factors with More than 2 Levels
- Example: Treatment Coding
- Problem of Multiple Comparisons
- Orthogonal Contrasts
  - Example
  - Implementation
  - Definition
- Recap of Coding Systems
- Alternatives & Addenda
Treatment / Dummy Coding

- Coding:
  - Baseline level always coded as 0
  - Each other level is coded as 1 in one of the \( g-1 \) contrasts
  - \texttt{contr.treatment()} in R

- Interpretation:
  - Each contrast compares one condition to the \textit{baseline}

- Examples:
  - Compare each of 2 different interventions (talk therapy & medication) to control w/ no intervention
  - In reading time, compare each of helpful and unhelpful context to version with \textit{no} context
Orthogonal Contrasts

• Coding:
  • Each contrast sums to 0
  • Product of weights across contrasts also sums to 0

• Interpretation:
  • Within each contrast, positively coded levels are compared to negative ones

• Examples:
  • Second language learning. Contrast 1 compares words related to 1st language with unrelated words. Contrast 2 compares two types of relations.
Helmert Contrasts

• Coding:
  • A subtype of orthogonal contrast
  • `contr.helmert()` in R

• Interpretation:
  • Each level is compared to the mean of all previous ones

• Use when categories are ordered:
  • Changes in time / across phases of an experiment
  • “Easy,” “medium,” or “hard” items
Orthogonal Polynomials

- **Coding:**
  - A *subtype* of orthogonal contrast
  - `contr.poly()` in R

- **Interpretation:**
  - Is there a linear effect across levels?
  - Is there a quadratic effect across levels?
  - + cubic, quartic, etc...

- **Use when categories are ordered and you’re interested in the form of the relation**
  - Linear increase from low->medium->high arousal, or is medium arousal the best?
Effects / Sum Coding

- Coding:
  - Code one level as -0.5 (or as -1)
  - Each other level is coded as 0.5 (or 1) in one of the $g-1$ contrasts
  - `contr.sum()` in R

- Interpretation:
  - Each contrast compares one condition to the overall mean

- Used when we don’t want to compare specific conditions & don’t have a clear baseline:
  - Compare students with various majors to the mean across majors
Week 7: Coding Predictors II

- Factors with More than 2 Levels
- Example: Treatment Coding
- Problem of Multiple Comparisons
- Orthogonal Contrasts
  - Example
  - Implementation
  - Definition
- Recap of Coding Systems
- Alternatives & Addenda
Testing an Overall Effect

- In ANOVA world, common to ask if there’s an overall effect of InterruptionType
  - “Are there any differences among conditions?”

- $t$-test version:
  - Package `car`
  - `anova(orthogonal.Maximal)`

Analysis of Variance Table

<table>
<thead>
<tr>
<th></th>
<th>Df</th>
<th>Sum Sq</th>
<th>Mean Sq</th>
<th>F value</th>
</tr>
</thead>
<tbody>
<tr>
<td>InterruptionType</td>
<td>2</td>
<td>41.352</td>
<td>20.676</td>
<td>20.676</td>
</tr>
</tbody>
</table>

- Reading on CourseWeb for doing the likelihood ratio test version
Multiple Comparisons

• “But I really want to run more than $g-1$ comparisons”

• Apply corrections to control Type I error
  • Note that if we’re doing post-hoc tests in an ANOVA, we should also be doing this. (It’s just that people often don’t.) Not really a mixed effects thing.

• e.g., Bonferroni: Multiple $p$-value by number of comparisons
  • “Most conservative correction” / worst case scenario
Understanding Interactions

- With an interaction, often want to characterize effects within particular conditions
- Last week: Aphasia x sentence structure

**W/ TREATMENT CODING:**

<table>
<thead>
<tr>
<th>Fixed effects</th>
<th>Estimate</th>
<th>Std. Error</th>
<th>t value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>1716.01</td>
<td>76.85</td>
<td>22.330</td>
</tr>
<tr>
<td>SubjectType1</td>
<td>84.52</td>
<td>76.37</td>
<td>1.107</td>
</tr>
<tr>
<td>SentenceType1</td>
<td>577.17</td>
<td>78.33</td>
<td>7.368</td>
</tr>
<tr>
<td>SubjectType1:SentenceType1</td>
<td>188.75</td>
<td>17.71</td>
<td>10.659</td>
</tr>
</tbody>
</table>

- Parameter estimates actually contain enough information to fully characterize data
- Could calculate RT in all four conditions
Understanding Interactions

- Looking at means often helps explain interaction
  - Numbers or plots

```r
> tapply(aphasia$RT, list(aphasia$SubjectType, aphasia $SentenceType), mean)
   Active  Passive
Aphasia 1800.528 2566.446
Control 1716.007 2293.173
```

- Can also do post-hoc tests within conditions
  - Test aphasics vs. control with just active sentences
  - Then, test with just passive sentences
  - Use `subset()` and then fit another model to each of the subsets
  - Just be aware that these tests are not fully independent of the original model