Course Business

- Midterm assignment: Review a journal article in your area that uses mixed-effects models
  - See CourseWeb document for specific requirements and grading rubric
  - Due on CourseWeb on March 1st at 2:00 PM—2 weeks from today
  - Unsure if an article is suitable? Can run it by me

- New dataset on CourseWeb

- Next 3 weeks:
  - This week: Finish categorical predictors
  - Next week: Categorical outcomes
  - 2 weeks: Discuss midterm projects
Week 7: Coding Predictors II

- Distributed Practice
- Factors with More than 2 Levels
- Treatment Coding
- Problem of Multiple Comparisons
- Orthogonal Contrasts
  - Example
  - Implementation
  - Definition
  - Practice
- Overview of Coding Systems
- Understanding Interactions
- Alternatives & Addenda
Distributed Practice!

- Your research team is modeling the effect of out-of-class study on college achievement. You recruit a sample of 300 students. Each term over their college career, the students report the number of hours they spent studying for their final exam week that term as well as their GPA for that term. Your first model is:
  \[
  \text{model1} <- \text{lmer}(\text{GPA} \sim 1 + \text{HoursOfStudy} + (1|\text{Subject}), \text{data=x})
  \]
  But, your team thinks \text{HoursOfStudy} may show a stronger effect on \text{GPA} for some students than others (i.e., some people make better use of study time). How can your new model reflect this?
  \[
  \text{model2} <- \text{lmer}(\text{GPA} \sim 1 + \text{HoursOfStudy} + \boxed{???}, \text{data=x})
  \]

- Albert says, “We should use \((1|\text{Subject}) + (1|\text{HoursOfStudy})\) because we’re adding \text{HoursOfStudy} as another random effect.”
- Betsy says, “We can use \((1+\text{HoursOfStudy}|\text{Subject})\) to make both the intercept and slope different for each subject.”
- Carlos says, “We want to capture both subject differences and \text{HoursOfStudy} differences, so it’s \((1|\text{Subject}+\text{HoursOfStudy})\)”
- Dipika says, “\text{HoursOfStudy} is a between-subjects variable, so this question makes no sense.”
Distributed Practice!

- Your research team is modeling the effect of out-of-class study on college achievement. You recruit a sample of 300 students. Each term over their college career, the students report the number of hours they spent studying for their final exam week that term as well as their GPA for that term. Your first model is:

  ```r
  model1 <- lmer(GPA ~ 1 + HoursOfStudy + (1|Subject), data=x)
  ```

- But, your team thinks `HoursOfStudy` may show a stronger effect on `GPA` for some students than others (i.e., some people make better use of study time). How can your new model reflect this?

  ```r
  model2 <- lmer(GPA ~ 1 + HoursOfStudy + ???, data=x)
  ```

- Albert says, “We should use `(1|Subject) + (1|HoursOfStudy)` because we’re adding `HoursOfStudy` as another random effect.”

- Betsy says, “We can use `(1+HoursOfStudy|Subject)` to make both the intercept and slope different for each subject.”

- Carlos says, “We want to capture both subject differences and `HoursOfStudy` differences, so it’s `(1|Subject+HoursOfStudy)`”

- Dipika says, “`HoursOfStudy` is a between-subjects variable, so this question makes no sense.”
Distributed Practice!

- Alyssa is a chemistry professor experimenting with online quizzes. Half of her students take a quiz on the Web, and half take it on paper. In Alyssa’s R dataframe (called quizzes), that variable looks like this:

  ![QuizType
  Paper:50
  Web :50](image)

  Alyssa is interested in:
  - The **overall average** quiz score, and
  - The effect of **Web** quizzes relative to paper quizzes

- Given the eventual model:
  - `model2<-lmer(Score ~ 1+QuizType + (1|Year), data=quizzes)`
- What R code will create contrasts for QuizType that will tell her both (a) and (b) in one model?
Distributed Practice!

- Alyssa is a chemistry professor experimenting with online quizzes. Half of her students take a quiz on the Web, and half take it on paper. In Alyssa’s R dataframe (called quizzes), that variable looks like this:

  ![QuizType](Paper:50
  Web :50)

  Alyssa is interested in:
  a. The overall average quiz score, and
  b. The effect of Web quizzes relative to paper quizzes

- Given the eventual model:
  - `model2 <- lmer(Score ~ 1 + QuizType + (1 | Year), data=quizzes)`

- What R code will create contrasts for QuizType that will tell her both (a) and (b) in one model?
  - `contrasts(quizzes$QuizType) <-`
Distributed Practice!

- Alyssa is a chemistry professor experimenting with online quizzes. Half of her students take a quiz on the Web, and half take it on paper. In Alyssa’s R dataframe (called quizzes), that variable looks like this:

  Alyssa is interested in:
  
  a. The *overall average* quiz score, and
  b. The effect of *Web* quizzes relative to paper quizzes

- Given the eventual model:
  
  ```r
  model2 <- lmer(Score ~ 1 + QuizType + (1|Year), data=quizzes)
  ```

- What R code will create contrasts for QuizType that will tell her both (a) and (b) in one model?
  
  ```r
  contrasts(quizzes$QuizType) <- c(???, ???)
  ```
Distributed Practice!

- Alyssa is a chemistry professor experimenting with online quizzes. Half of her students take a quiz on the Web, and half take it on paper. In Alyssa’s R dataframe (called quizzes), that variable looks like this:

  ![quiz_type_table]

- Alyssa is interested in:
  a. The overall average quiz score, and
  b. The effect of Web quizzes relative to paper quizzes

- Given the eventual model:
  ```r
  model2 <- lmer(Score ~ 1 + QuizType + (1|Year), data=quizzes)
  ```

- What R code will create contrasts for QuizType that will tell her both (a) and (b) in one model?
  ```r
  contrasts(quizzes$QuizType) <- c(-0.5, 0.5)
  ```
Week 7: Coding Predictors II

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

- disfluency.csv on CourseWeb

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

- Disfluencies more common with *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)

• disfluency.csv on CourseWeb

• Each participant hears stories based on Alice in Wonderland
  • Later, test recall of each chapter – scored from 0 to 10

• Conditions:
  • Some chapters told fluently (control)
  • Some chapters contain speech fillers
  • Some have coughs matched in duration to the fillers
  • Each subject hears some chapters in all 3 conditions
  • Each chapter heard in all 3 conditions across subjects
Alice in Um-derland (Fraundorf & Watson, 2011)

- Average memory score in each condition:
  - `tapply(disfluency$MemoryScore, disfluency$InterruptionType, mean)`
  - “Take MemoryScore, separate it out by InterruptionType, and give me the mean”

<table>
<thead>
<tr>
<th>Control</th>
<th>Cough</th>
<th>Filler</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.05625</td>
<td>6.20000</td>
<td>7.34375</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: \( \text{Score} = \gamma_{000} + \gamma_{100} \times \text{InterruptionType} \)

Cough: \( \text{Score} = \gamma_{000} + \gamma_{100} \times \text{InterruptionType} \)

Filler: \( \text{Score} = \gamma_{000} + \gamma_{100} \times \text{InterruptionType} \)
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>Score Equation</th>
<th>Score Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fluent</td>
<td>( y_{000} + y_{100} )</td>
<td>0</td>
</tr>
<tr>
<td>Cough</td>
<td>( y_{000} + y_{100} )</td>
<td>1</td>
</tr>
<tr>
<td>Filler</td>
<td>( y_{000} + y_{100} )</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:

\[
\text{Score} = \gamma_{000} + \gamma_{100} \]

- 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...
  - `disfluency.NoCoughs <- subset(disfluency, InterruptionType != 'Cough')`
  - ...R still remembers all of the possible levels...

- Solution: Re-make into a factor with `factor()`:
  - `disfluency.NoCoughs$InterruptionType <- factor(disfluency.NoCoughs$InterruptionType),posssible_levels

<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>

<table>
<thead>
<tr>
<th></th>
<th>Filler</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control</td>
<td>0</td>
</tr>
<tr>
<td>Filler</td>
<td>1</td>
</tr>
</tbody>
</table>
Week 7: Coding Predictors II

- Distributed Practice
- Factors with More than 2 Levels
- Treatment Coding
- Problem of Multiple Comparisons
- Orthogonal Contrasts
  - Example
  - Implementation
  - Definition
  - Practice
- Overview of Coding Systems
- Understanding Interactions
- Alternatives & Addenda
Treatment Coding With >2 Levels

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

Fluent: \( \text{Score} = \gamma_{000} + \gamma_{100} \times \text{InterruptionType1} + \gamma_{200} \times \text{InterruptionType2} \)

Cough: \( \text{Score} = \gamma_{000} + \gamma_{100} \times \text{InterruptionType1} + \gamma_{200} \times \text{InterruptionType2} \)

Filler: \( \text{Score} = \gamma_{000} + \gamma_{100} \times \text{InterruptionType1} + \gamma_{200} \times \text{InterruptionType2} \)
**Treatment Coding With >2 Levels**

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

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

  **Fluent**
  \[
  \text{Score} = \gamma_{000} + \gamma_{100} \times 0 + \gamma_{200} \times 0
  \]

  **Cough**
  \[
  \text{Score} = \gamma_{000} + \gamma_{100} \times \text{InterruptionType1} + \gamma_{200} \times \text{InterruptionType2}
  \]

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

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

\[
\text{Fluent: } \text{Score} = \gamma_{000} \quad \text{Once again, the intercept is just performance in the } \text{baseline level (the one coded with all 0s)}
\]

\[
\begin{align*}
\text{Cough: } \text{Score} & = \gamma_{000} + \gamma_{100} \times 1 + \\
& + \gamma_{200} \times 0 \\
\text{Filler: } \text{Score} & = \gamma_{000} + \gamma_{100} \times \text{InterruptionType1} + \\
& + \gamma_{200} \times \text{InterruptionType2}
\end{align*}
\]
Treatment Coding With >2 Levels

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

  ![Equations](image)

  - Fluent: Score = $\gamma_{000}$
  - Cough: Score = $\gamma_{000} + \gamma_{100}$
  - Filler: Score = $\gamma_{000} + \gamma_{100} * 0 + \gamma_{200} * 1$

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

  InterruptionType1 = 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: \[ \text{Score} = \gamma_{000} \]

  Cough: \[ \text{Score} = \gamma_{000} + \gamma_{100} \]

  Filler: \[ \text{Score} = \gamma_{000} + \gamma_{200} \]

  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 how well each chapter was recalled based on the InterruptionType condition.
  - Each subject hears some chapters in each of the 3 conditions.
  - Each chapter heard in all 3 conditions across subjects.
- Finish the maximal random effects model:
  - `dummycode.Maximal <- lmer(MemoryScore ~ 1 + InterruptionType + (1 + InterruptionType | Subject) + (1 + InterruptionType | Chapter), data=disfluency)`
Treatment Coding: Model

• We want to analyze how well each chapter was recalled based on the InterruptionType condition
  • Each subject hears some chapters in each of the 3 conditions
  • Each chapter heard in all 3 conditions across subjects
• Finish the maximal random effects model:
  • `dummycode.Maximal <- lmer(MemoryScore ~ 1 + InterruptionType + (1 + InterruptionType|Subject) + 
    (1 + InterruptionType|Chapter),
    data=disfluency)`
Treatment Coding: Model

• We want to analyze how well each chapter was recalled based on the InterruptionType condition
  • Each subject hears some chapters in each of the 3 conditions
  • Each chapter heard in all 3 conditions across subjects
• Finish the maximal random effects model:
  • `dummycode.Maximal <- lmer(MemoryScore ~ 1 + InterruptionType + (1 + InterruptionType|Subject) + (1 + InterruptionType|Chapter), data=disfluency)`
**Treatment Coding: Results**

Intercept: Baseline score in the control condition

Cough effect: Numerically greater recall with coughs than control fluent condition ... but not statistically significant

Filler effect: Greater recall with speech fillers than control fluent condition
Week 7: Coding Predictors II

- Distributed Practice
- Factors with More than 2 Levels
- Treatment Coding
- Problem of Multiple Comparisons
- Orthogonal Contrasts
  - Example
  - Implementation
  - Definition
  - Practice
- Overview of Coding Systems
- Understanding Interactions
- 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
- But there is a problem running too many comparisons
Recap of Hypothesis Tests

- Under null hypothesis of no effect, extreme z- or t-values are *improbable*
Recap of Hypothesis Tests

- Under null hypothesis of no effect, extreme $z$- or $t$-values are **improbable**
- A $t$-value with < 5% probability: Significant evidence against null hypothesis
- But, **possible** (just unlikely) even if no real effect

Total probability of a $z$-score here under H0 = .05
Recap of Hypothesis Tests

• Under null hypothesis of no effect, extreme $z$- or $t$-values are improbable
• A $t$-value with < 5% probability: Significant evidence against null hypothesis
• But, possible (just unlikely) even if no real effect
  • Could just result from chance! (sampling error)
  • We’d conclude that there is an effect, but it doesn’t really exist
    • False positive or Type I error
  • 5% probability of this happening
  • $\alpha = .05$
Problem of Multiple Comparisons

• With $\alpha = .05$ (Type I error rate of 5%), we’d expect 1 in 20 comparisons to be significant just by chance
  • Problem if we want to run lots of comparisons!
  • Even if aren’t running 20 comparisons, Type I error will be inflated with >1 comparison

• Probability of fully avoiding Type I error:
  
\[
(1 - \alpha) \times (1 - \alpha) \approx 90\%
\]
Problem of Multiple Comparisons

- But with $\alpha = .05$ (Type I error rate of 5%), we’d expect 1 in 20 comparisons to be significant just by chance
- Problem if we want to run lots of comparisons!
  - Even if aren’t running 20 comparisons, Type I error will be inflated with >1 comparison

- Probability of fully avoiding Type I error:

\[
1 - \alpha \times (1 - \alpha) \times \ldots = (1 - \alpha)^c
\]

- Familywise error rate: $\alpha_{FW} = 1 - (1 - \alpha)^c$
  - Probability of making a Type I error somewhere
News

Green Jelly Beans Linked to Acne!

95% Confidence

Only 5% Chance of Coincidence!

Scientists...
The Problem of Multiple Comparisons

- Situation gets even more complicated if we use some of the same data in >1 comparison

Maybe we underestimated performance in the fluent control condition

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

- Situation gets even more complicated if we use some of the same data in >1 comparison
- 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
The Problem of Multiple Comparisons

- How many independent comparisons?
- Our model already showed us that:
  - Recall score is 0.14 points greater for *coughs* than fluent
  - Recall score is 1.28 points greater for *fillers* than fluent
  - Difference between *coughs* and *fillers* is *already known*: \( 1.28 - 0.14 = 1.14 \)
The Problem of Multiple Comparisons

• How many independent comparisons?
• Our model already showed us that:
  • Recall score is 0.14 points greater for coughs than fluent
  • Recall score is 1.28 points greater for fillers than fluent
  • Difference between coughs and fillers is already known: 1.28 – 0.14 = 1.14
• 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

- 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
  • Not new, independent comparisons … inflated Type I error rate

• In fact, even our current comparisons aren’t totally independent
  • Filler vs control
  • Coughs vs control
  • If we have underestimated (or overestimated) performance in Control condition, both comparisons will be affected
Here Comes Trouble!

• These reasons are why R doesn’t perform all possible comparisons between all levels
  • Not new, independent comparisons … inflated Type I error rate

• So, here’s what let’s do:
  • First, let’s look at how to run comparisons that are *truly* independent
  • And, we can discuss how to control Type I error rate if we need comparisons that aren’t independent
Week 7: Coding Predictors II

- Distributed Practice
- Factors with More than 2 Levels
- Treatment Coding
- Problem of Multiple Comparisons
- Orthogonal Contrasts
  - Example
  - Implementation
  - Definition
  - Practice
- Overview of Coding Systems
- Understanding Interactions
- 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

**FLUENT**

**CONTROL**

**COUGHS**

**FILLERS**

*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

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

<table>
<thead>
<tr>
<th>FLUENT CONTROL</th>
<th>COUGHS</th>
<th>FILLERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-1/2</td>
<td>1/2</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>FLUENT</th>
<th>COUGHS</th>
<th>FILLERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-1/2</td>
<td>1/2</td>
</tr>
</tbody>
</table>

Centered around mean of 0!

Do coughs and fillers differ?

CONTRAST 2

<table>
<thead>
<tr>
<th>FLUENT</th>
<th>COUGHS</th>
<th>FILLERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2/3</td>
<td>1/3</td>
<td>1/3</td>
</tr>
</tbody>
</table>

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

- Distributed Practice
- Factors with More than 2 Levels
- Treatment Coding
- Problem of Multiple Comparisons
- Orthogonal Contrasts
  - Example
    - Implementation
    - Definition
    - Practice
- Overview of Coding Systems
- Understanding Interactions
- Alternatives & Addenda
Let’s Change the Contrasts

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

Now, we’re trying to create a matrix of numbers

• Need to stick two columns together with `cbind`:

\[
\text{contrasts(\text{disfluency}\$\text{InterruptionType}) <- cbind(c(0,-1/2,1/2), c(-2/3, 1/3,1/3))}
\]

Contrast 1         Contrast 2

\[
\begin{array}{c|c}
\text{Control} & 0.0 & -0.6666667 \\
\text{Cough} & -0.5 & 0.3333333 \\
\text{Filler} & 0.5 & 0.3333333 \\
\end{array}
\]
Naming the Contrasts

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

<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>

• We can change the names of these columns with `colnames()`

```
colnames(contrasts(disfluency $InterruptionType)) <- c('FillerVsCough', 'InterruptionVsFluent')
```

<table>
<thead>
<tr>
<th></th>
<th>FillerVsCough</th>
<th>InterruptionVsFluent</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>

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

- **summary(orthogonal.Maximal)**

<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>6.53284</td>
<td>0.14917</td>
<td>43.79</td>
</tr>
<tr>
<td>InterruptionTypeFillerVsCough</td>
<td>1.14604</td>
<td>0.18322</td>
<td>6.25</td>
</tr>
<tr>
<td>InterruptionTypeInterruptionVsFluent</td>
<td>0.71441</td>
<td>0.09911</td>
<td>7.21</td>
</tr>
</tbody>
</table>

  **Intercept**: Mean score across conditions (because these are centered)

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

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

- `anova(dummycode.Maximal, orthogonal.Maximal)`

<table>
<thead>
<tr>
<th></th>
<th>Df</th>
<th>AIC</th>
<th>BIC</th>
<th>logLik</th>
<th>deviance</th>
</tr>
</thead>
<tbody>
<tr>
<td>dummycode.Maximal</td>
<td>16</td>
<td>1478.3</td>
<td>1545</td>
<td>-723.13</td>
<td>1446.3</td>
</tr>
<tr>
<td>orthogonal.Maximal</td>
<td>16</td>
<td>1478.3</td>
<td>1545</td>
<td>-723.13</td>
<td>1446.3</td>
</tr>
</tbody>
</table>

- Overall model fit is **identical!**
  - Same total amount of variance explained
  - 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
Week 7: Coding Predictors II

- Distributed Practice
- Factors with More than 2 Levels
- Treatment Coding
- Problem of Multiple Comparisons
- Orthogonal Contrasts
  - Example
  - Implementation
    - Definition
    - Practice
- Overview of Coding Systems
- Understanding Interactions
- Alternatives & Addenda
What Makes Contrasts Orthogonal?
What Makes Contrasts Orthogonal?

• Criterion 1: Codes within contrast sum to 0

<table>
<thead>
<tr>
<th>CONTRAST 1</th>
<th>CONTRAST 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLUENT</td>
<td>-0.67</td>
</tr>
<tr>
<td>CONTROL</td>
<td>0.33</td>
</tr>
<tr>
<td>COUGHS</td>
<td>0.5</td>
</tr>
<tr>
<td>FILLERS</td>
<td>0.33</td>
</tr>
</tbody>
</table>

= 0  = 0
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

![Orthogonal Contrast Calculation](image)
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

- Distributed Practice
- Factors with More than 2 Levels
- Treatment Coding
- Problem of Multiple Comparisons
- Orthogonal Contrasts
  - Example
  - Implementation
  - Definition
  - Practice
- Overview of Coding Systems
- Understanding Interactions
- Alternatives & Addenda
Orthogonal Contrasts Practice

• Hitomi is a clinical psychologist investigated the effectiveness of talk therapy. She examines the severity of depressive symptoms in three groups: waitlisted controls, people receiving medication, and people receiving medication and talk therapy. In her dataframe, depression, this Treatment variable looks like this:

```
<table>
<thead>
<tr>
<th>TreatmentType</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control:100</td>
<td></td>
</tr>
<tr>
<td>Medication:100</td>
<td></td>
</tr>
<tr>
<td>MedicationAndTalk:100</td>
<td></td>
</tr>
</tbody>
</table>
```

• Hitomi wants to compare:
  • The two groups receiving any treatment vs. controls
  • Medication + talk therapy vs. medication only
• Create some R code to set these contrasts:
Orthogonal Contrasts Practice

• Hitomi is a clinical psychologist investigated the effectiveness of talk therapy. She examines the severity of depressive symptoms in three groups: waitlisted controls, people receiving medication, and people receiving medication and talk therapy. In her dataframe, depression, this Treatment variable looks like this:

<table>
<thead>
<tr>
<th>TreatmentType</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control</td>
<td>100</td>
</tr>
<tr>
<td>Medication</td>
<td>100</td>
</tr>
<tr>
<td>MedicationAndTalk</td>
<td>100</td>
</tr>
</tbody>
</table>

• Hitomi wants to compare:
  • The two groups receiving any treatment vs. controls
  • Medication + talk therapy vs. medication only
• Create some R code to set these contrasts:
  ```R
  contrasts(depression$Treatment) <-
  cbind(c(-2/3, 1/3, 1/3), c(0, -0.5, 0.5))
  ```
Orthogonal Contrasts Practice

For each set of contrasts, decide whether it IS orthogonal or IS not orthogonal.

(a) 

\[
\begin{array}{c|c|c}
\text{ [,1]} & 
\text{ [,2]} \\
\text{Divorced} & 1 & 0 \\
\text{Married} & 0 & 1 \\
\text{Single} & 0 & 0 \\
\end{array}
\]

(b) 

```
> contrasts(L2Vocab$RelationType)

\[
\begin{array}{c|c|c}
\text{ [,1]} & 
\text{ [,2]} \\
\text{SimilarMeaning} & 0.3333333 & 0.5 \\
\text{SimilarSound} & 0.3333333 & -0.5 \\
\text{Unrelated} & -0.6666667 & 0.0 \\
\end{array}
\]
```

(c) 

\[
\begin{array}{c|c|c}
\text{ [,1]} & 
\text{ [,2]} \\
\text{Young} & -1 & -1 \\
\text{MiddleAge} & 1 & -1 \\
\text{Older} & 0 & 2 \\
\end{array}
\]

(d) 

```
> contrasts(L2Vocab$RelationType)

\[
\begin{array}{c|c|c|c}
\text{Northeast} & \text{South} & \text{West} \\
\text{Midwest} & 0 & 0 & 0 \\
\text{Northeast} & 1 & 0 & 0 \\
\text{South} & 0 & 1 & 0 \\
\text{West} & 0 & 0 & 1 \\
\end{array}
\]
```
Orthogonal Contrasts Practice

• For each set of contrasts, decide whether it IS orthogonal or IS not orthogonal.

(a) NOT orthogonal

(b) orthogonal

(c) orthogonal

(d) NOT orthogonal
Orthogonal Contrasts Practice

• Zebulon is a health psychologist examining the (potentially) protective effect of physical exercise on cognition in older adults. Measures of working memory are obtained from older adults from each of three groups: a control group that just stretches, a group that performs low-intensity exercises, and a group that performs moderate-intensity exercises.

• What is a reasonable set of orthogonal comparisons that Zebulon might make?
Orthogonal Contrasts Practice

• Zebulon is a health psychologist examining the (potentially) protective effect of physical exercise on cognition in older adults. Measures of working memory are obtained from older adults from each of three groups: a control group that just stretches, a group that performs low-intensity exercises, and a group that performs moderate-intensity exercises.

• What is a reasonable set of orthogonal comparisons that Zebulon might make?
  • One plausible set of comparisons is (a) control vs. any type of exercise, and (b) moderate- vs. low-intensity exercise. (Other answers may be possible.)
Week 7: Coding Predictors II

- Distributed Practice
- Factors with More than 2 Levels
- Treatment Coding
- Problem of Multiple Comparisons
- Orthogonal Contrasts
  - Example
  - Implementation
  - Definition
  - Practice
- Overview of Coding Systems
- Understanding Interactions
- 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

- Treatment coding is R’s default, but we might want to set this back (if we switched to something else but we want treatment coding back)

- **Shortcut** to this:
  - \( \text{contrasts(disfluency}\$\text{InterruptionTime)} <- \text{contr.treatment(n=3)} \)
  - \( n=3 \) because there are 3 groups
Treatment / Dummy Coding

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

- Interpretation:
  - Each contrast compares one condition to the 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 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
  • Control, mild anxiety, severe anxiety
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

- Distributed Practice
- Factors with More than 2 Levels
- Treatment Coding
- Problem of Multiple Comparisons
- Orthogonal Contrasts
  - Example
  - Implementation
  - Definition
  - Practice
- Overview of Coding Systems
- Understanding Interactions
- Alternatives & Addenda
Understanding Interactions

• If we have an interaction… what does it mean?

• 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>

- Baseline RT for control subj, active
- RT diff for aphasics, active
- RT diff for control subj, passive
- Special RT effect of aphasic+passive

• 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
Week 7: Coding Predictors II

- Distributed Practice
- Factors with More than 2 Levels
- Treatment Coding
- Problem of Multiple Comparisons
- Orthogonal Contrasts
  - Example
  - Implementation
  - Definition
  - Practice
- Overview of Coding Systems
- Understanding Interactions
- Alternatives & Addenda
**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
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
            Df Sum Sq  Mean Sq F value
InterruptionType  2  82.008  41.004  43.223
```

• Reading on CourseWeb for doing the likelihood ratio test version
Unbalanced Factors

• Sometimes, we may have differing numbers of observations per level

• Possible reasons:
  • Some categories naturally more common
    • e.g., college majors
  • Categories may be equally common in the population, but we have sampling error
    • e.g., ended up 60% female participants, 40% male
  • Study was designed so that some conditions are more common
    • e.g., more “control” subjects than “intervention” subjects
  • We wanted equal numbers of observations, but lost some because of errors or exclusion criteria
    • e.g., data loss due to computer problems
    • Dropping subjects below a minimum level of performance
Weighted Coding

• “For the average student, does course size predict probability of graduation?”
  • Random sample of 200 Pitt undergrads
  • 5 are student athletes and 195 are not

• How can we make the intercept reflect the “average student”?
  • We could try to apply effects coding to the StudentAthlete variable by centering around the mean and getting \((0.5, -0.5)\), but…
**Weighted Coding**

- An intercept at 0 would no longer correspond to the overall mean.

```
.5

ATHLETE (5)
```

- Zero is here.

```
0
```

- NOT ATHLETE (195)

```
-.475 -.5
```

- As a scale, this would be totally unbalanced.

- To fix balance, we need to assign a heavier weight to Athlete.

But “not athlete” is actually far more common.
Weighted Coding

- Change codes so the mean is 0
- \( c(.975, -0.025) \)
- `contr.helmert.weighted()` function in my psycholing package will calculate this
**Weighted Coding**

- **Weighted coding**: Change the codes so that the mean is 0 again
  - Used when the imbalance reflects something real
  - Like Type II sums of squares

- “For the average student, does course size predict graduation rates?”
  - Average student is *not* a student athlete, and our answer to the question about an “average student” should reflect this!
Unweighted Coding

• Oops! Our experiment loaded up the wrong image for one of our Passive sentences (“Groceries”)
  • It may have been sabotaged
  • UsableItem column is No for this item

• First, can we remove this from our data?
• Some possibilities:
  • `aphasia <- aphasia[aphasia$UsableItem == 'Yes', ]`
  • `aphasia2 <- aphasia[aphasia$UsableItem != 'No', ]`
  • `aphasia2 <- subset(aphasia, UsableItem == 'Yes')`
  • etc.
Unweighted Coding

• Oops! Our experiment loaded up the wrong image for one of our Passive sentences (“Groceries”)

• Now, there’s an imbalance, but it’s an accident and not meaningful
  • In fact, we’d like to get rid of it!

```r
> summary(aphasia2)

  Subject Item SubjectType SentenceType
S1   : 31 Astronaut: 30 Aphasia: 465 Active: 480
S11  : 31 Boy : 30
S12  : 31 Breakfast: 30
S13  : 31 Burglar : 30
S14  : 31 Cheese : 30
(Other): 744 (Other) : 750
```
Unweighted Coding

• Oops! Our experiment loaded up the wrong image for one of our Passive sentences (“Groceries”)

• Now, there’s an imbalance, but it’s an accident and not meaningful
  • In fact, we’d like to get rid of it!

• Retain the (-0.5, 0.5) codes
  • Weights the two conditions equally—because the imbalance isn’t meaningful
  • Like Type III sums of squares
  • Probably what you want for factorial experiments
Unbalanced Factors: Summary

- Weighted coding: Change the codes so that the mean is 0
  - Use when the imbalance reflects something real
  - Can be done with `contrast.helmert.weighted()`

Mean across each individual:

- Unweighted coding: Keep the codes as -0.5 and 0.5
  - Use when the imbalance is an accident that we want to eliminate

Mean of the two levels:

(with perfectly balanced factors, these two methods are identical)