Course Business

• Course evaluation e-mails going out this week

• Scott out of town Wednesday through Sunday

• Final paper rubric is now posted—due Dec. 2

• Final project presentation schedule:
  • Nov. 25: Cristina, Hilary, Kevin, Laura, Leanne
  • Dec. 2: Becca, Caitlin, Cory, Evelyn, Michelle, Ruizhe, Xiaoping, Zhaohong

• Project presentation prep:
  • Can use my laptop or your own
    • If you e-mail me file beforehand, I can check if it displays correctly on my computer
  • Plan to ask at least 1 question over the two days
Week 13: Data Management & Level-2 Variables

- Follow-ups
- Data Management in R
  - rbind()
  - merge()
  - melt()
- Level-2 Fixed & Random Effects
  - What do level-2 variables do?
  - Continuous or categorical?
    - Median splits
    - Extreme groups design
  - Good measurement
    - Reliability
    - Validity
Follow-Ups

- Follow-up on MAR vs MNAR:
  - Issue isn’t about what variable physically “causes” the missingness
  - It’s whether about whether some values of a variable are more likely to end up missing (whatever the reason)
- We probably can’t know for sure the degree to which our data is MAR vs. MNAR
  - Unless we know a variable that was used to screen people
- But, be aware of the issue
  - Affects how justifiable case deletion or imputation is
Follow-Ups

• Running `glmer()` without `family=binomial` did not work for me
• My results have e+ or e- in them!

Scientific notation shows up if at least one effect is very large or very small
• 9.638e-03 = 9.638 x 10^{-3} = .009638
• e-03: Move the decimal three places to the left
• e+03: Move the decimal three places to the right
• Copy & paste the scientific notation into the R command line to get the regular number:

> 9.638e-03
[1] 0.009638
Follow-ups

Data Management in R
- rbind()
- merge()
- melt()

Level-2 Fixed & Random Effects
- What do level-2 variables do?
- Continuous or categorical?
  - Median splits
  - Extreme groups design
- Good measurement
  - Reliability
  - Validity
Week 13: Data Management & Level-2 Variables

- Lots of data today (we’ll be talking about how to combine it):
**rbind()**

- Paste together the rows from two (or more) dataframes to create a new one:
  - `allschools <- rbind(school1, school2, school3)`

- Useful when observations are spread across files
  - Or, to create a dataframe that consists of 2 subsets
- Requires these to have the same columns
  - Do before calculating new variables
- “More of the same”
Week 13: Data Management & Level-2 Variables

- Follow-ups
- Data Management in R
  - `rbind()`
  - `merge()`
  - `melt()`
- Level-2 Fixed & Random Effects
  - What do level-2 variables do?
  - Continuous or categorical?
    - Median splits
    - Extreme groups design
  - Good measurement
    - Reliability
    - Validity
merge()

- Sometimes different files/dataframes contain different variables relevant to the same observations.
- Common scenario in mixed effects models context: Level-2 variables are in a different file than Level-1 measurements.

<table>
<thead>
<tr>
<th>School</th>
<th>Classroom</th>
<th>Student</th>
<th>HoursOfStudy</th>
<th>StudentSES</th>
<th>Pretest</th>
<th>Posttest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jefferson</td>
<td>C001</td>
<td>S0001</td>
<td>1</td>
<td>0.78035730</td>
<td>0.51378000</td>
<td>1.5399431</td>
</tr>
<tr>
<td>Jefferson</td>
<td>C001</td>
<td>S0002</td>
<td>3</td>
<td>-0.21536230</td>
<td>0.26349070</td>
<td>1.3080398</td>
</tr>
<tr>
<td>Jefferson</td>
<td>C001</td>
<td>S0003</td>
<td>0</td>
<td>0.12904320</td>
<td>0.52329010</td>
<td>1.4550667</td>
</tr>
<tr>
<td>Jefferson</td>
<td>C001</td>
<td>S0004</td>
<td>3</td>
<td>1.68735930</td>
<td>0.36402300</td>
<td>0.6022264</td>
</tr>
<tr>
<td>Jefferson</td>
<td>C001</td>
<td>S0005</td>
<td>3</td>
<td>0.21965170</td>
<td>0.78668840</td>
<td>1.2517459</td>
</tr>
<tr>
<td>Jefferson</td>
<td>C001</td>
<td>S0006</td>
<td>5</td>
<td>-0.29315090</td>
<td>1.28626590</td>
<td>1.6046956</td>
</tr>
</tbody>
</table>

allschools: 1 row per student

Each classroom appears in multiple rows

tutoruse.csv:

<table>
<thead>
<tr>
<th>Classroom</th>
<th>Tutor</th>
</tr>
</thead>
<tbody>
<tr>
<td>C001</td>
<td>No</td>
</tr>
<tr>
<td>C002</td>
<td>Yes</td>
</tr>
<tr>
<td>C003</td>
<td>No</td>
</tr>
<tr>
<td>C004</td>
<td>Yes</td>
</tr>
<tr>
<td>C005</td>
<td>No</td>
</tr>
<tr>
<td>C006</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Each class has only one row—did this class use the tutor or not?
merge()

- Sometimes different files/dataframes contain different variables relevant to the same observations
- Common scenario in mixed effects models context: Level-2 variables are in a different file than Level-1 measurements

**lexicaldecision.csv:** Each word appears in multiple rows

<table>
<thead>
<tr>
<th>Subject</th>
<th>Word</th>
<th>PrevTrials</th>
<th>RT</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>panther</td>
<td>0 703.877</td>
<td></td>
</tr>
<tr>
<td>S1</td>
<td>drive</td>
<td>1 532.387</td>
<td></td>
</tr>
<tr>
<td>S1</td>
<td>monorail</td>
<td>2 731.882</td>
<td></td>
</tr>
<tr>
<td>S13</td>
<td>peony</td>
<td>0 808.392</td>
<td></td>
</tr>
<tr>
<td>S13</td>
<td>monorail</td>
<td>1 489.479</td>
<td></td>
</tr>
<tr>
<td>S13</td>
<td>aardvark</td>
<td>2 875.799</td>
<td></td>
</tr>
</tbody>
</table>

**subtlexus.csv:** Each word has only one row with its frequency

<table>
<thead>
<tr>
<th>Word</th>
<th>WordFreq</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>6.1766</td>
</tr>
<tr>
<td>to</td>
<td>6.0632</td>
</tr>
<tr>
<td>a</td>
<td>6.0175</td>
</tr>
<tr>
<td>you</td>
<td>6.3293</td>
</tr>
<tr>
<td>and</td>
<td>5.8343</td>
</tr>
<tr>
<td>it</td>
<td>5.9839</td>
</tr>
</tbody>
</table>
merge()

- Sometimes different files/dataframes contain different variables relevant to the same observations
- Common scenario in mixed effects models context: Level-2 variables are in a different file than Level-1 measurements

1 row per trial

Each subject has multiple rows

Each subject has only one row with his or her Reading Span score
merge()

- “Look up word frequency from the other dataframe”
- We can combine these dataframes if they have at least one column in common
- **Word** tells us which word was presented on an individual trial, and it also identifies the word in our database of word frequency

**lexicaldecision.csv**: 1 row per **trial**
- Each word appears in multiple rows

**subtlexus.csv**: Each word has only one row with its frequency
merge()

- `lexdec2 <- merge(lexicaldecision, subtlexus, by='Word')`
- New dataframe has both the columns from `lexicaldecision` (Subject, PrevTrials, RT) and the columns from `subtlexus` (WordFreq)
- Matches the observations using the `Word` column
merge()

- `lexdec2 <- merge(lexicaldecision, subtlexus, by='Word')`
  - New dataframe has both the columns from `lexicaldecision` (Subject, PrevTrials, RT) and the columns from `subtlexus` (WordFreq)
  - Matches the observations using the `Word` column

<table>
<thead>
<tr>
<th>Word</th>
<th>Subject</th>
<th>PrevTrials</th>
<th>RT</th>
<th>WordFreq</th>
</tr>
</thead>
<tbody>
<tr>
<td>aardvark</td>
<td>S1</td>
<td>45</td>
<td>Min. : 0.00</td>
<td>Min. : 56.0</td>
</tr>
<tr>
<td>and</td>
<td>S10</td>
<td>45</td>
<td>1st Qu.: 12.00</td>
<td>1st Qu.: 460.0</td>
</tr>
<tr>
<td>astronaut</td>
<td>S11</td>
<td>45</td>
<td>Median : 25.00</td>
<td>Median : 579.5</td>
</tr>
<tr>
<td>boy</td>
<td>S12</td>
<td>45</td>
<td>Mean : 25.05</td>
<td>Mean : 589.8</td>
</tr>
<tr>
<td>breakfast</td>
<td>S13</td>
<td>45</td>
<td>3rd Qu.: 38.00</td>
<td>3rd Qu.: 703.3</td>
</tr>
<tr>
<td>carburetor</td>
<td>S14</td>
<td>45</td>
<td>Max. : 50.00</td>
<td>Max. : 2780.0</td>
</tr>
<tr>
<td>(Other)</td>
<td></td>
<td>1560</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Other)</td>
<td></td>
<td>1530</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**merge() – Renaming Columns**

- What if the columns have different names?
- **Item** in `lexicaldecision` tells us which **Word** to look for in `subtlexus` ... but R doesn’t know that!
- Easy solution is to rename the column
  ```r
  colnames(lexicaldecision)[colnames(lexicaldecision)=='Item'] <- 'Word'
  ```
- Then do the `merge()`
**merge() – all.x and all.y**

- `nrow(lexicaldecision)` 2040
- `nrow(lexdec2)` 1800
- Six words don’t have a frequency measurement
- Default behavior of `merge()` is to drop rows that can’t be matched
- `lexdec2 <- merge(lexicaldecision, subtlexus, by='WORD', all.x=TRUE)`

Keep the rows in `lexicaldecision` where we can’t find the matching `WORD` in `subtlexus`

WordFreq will be NA in these rows
merge() – all.x and all.y

- `nrow(lexicaldecision)`: 2040
- `nrow(lexdec2)`: 1800
- Six words don’t have a frequency measurement
- Default behavior of `merge()` is to drop rows that can’t be matched
- `lexdec2 <- merge(lexicaldecision, subtlexus, by='WORD', all.x=TRUE, all.y=TRUE)`

Adding `all.y=TRUE` would also include rows for all of the words in the word frequency database, even the words that weren’t used in our experiment.

We DON’T need or want that.
merge() – Matching by Multiple Columns

- Sometimes, one column isn’t enough to uniquely match things across files/dataframes
- Can use multiple columns in `merge()`
  ```r
  lexdec2 <- merge(lexicaldecision, subtlexus, by=c('Word', 'Country'))
  ```
- This is a logical AND. Has to match both Word and Country

Imagine doing our task in both the US and UK. Word frequency differs somewhat between American English & British English, so now we need both Word and Country to look up the frequency.
**merge() – Troubleshooting**

- If you leave out `by=`:
  - R tries to figure out the matching columns on its own

- If you leave out `by=` and NO columns match:
  - R creates a massive dataframe in which every row in dataframe 1 is paired with every row in dataframe 2
  - `nrow(trials) * nrow(subtlexus)`

- Symptoms:
  - You end up with an enormous dataframe with tens of thousands of observations
  - The `merge()` takes so long that it seems like your computer has frozen

- Hit STOP and check your `merge()` call
**merge() – Practice!**

- Remember our math tutoring data?:

<table>
<thead>
<tr>
<th>School</th>
<th>Classroom</th>
<th>Student</th>
<th>HoursOfStudy</th>
<th>StudentSES</th>
<th>Pretest</th>
<th>Posttest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jefferson</td>
<td>C001</td>
<td>S0001</td>
<td>1</td>
<td>0.7803573</td>
<td>0.5137800</td>
<td>1.5399431</td>
</tr>
<tr>
<td>Jefferson</td>
<td>C001</td>
<td>S0002</td>
<td>3</td>
<td>-0.2153623</td>
<td>0.2634907</td>
<td>1.3080398</td>
</tr>
<tr>
<td>Jefferson</td>
<td>C001</td>
<td>S0003</td>
<td>0</td>
<td>0.1290432</td>
<td>0.5232901</td>
<td>1.4550667</td>
</tr>
<tr>
<td>Jefferson</td>
<td>C001</td>
<td>S0004</td>
<td>3</td>
<td>1.687393</td>
<td>0.3640230</td>
<td>0.6022264</td>
</tr>
<tr>
<td>Jefferson</td>
<td>C001</td>
<td>S0005</td>
<td>3</td>
<td>0.2196517</td>
<td>0.7866884</td>
<td>1.2517459</td>
</tr>
<tr>
<td>Jefferson</td>
<td>C001</td>
<td>S0006</td>
<td>5</td>
<td>-0.2931509</td>
<td>1.2862659</td>
<td>1.6046956</td>
</tr>
</tbody>
</table>

- Use `merge()` to add the tutor data from `tutoruse` to `allschools`
merge() – Practice!

- Remember our math tutoring data?:

```
merge()
```

- Use `merge()` to add the tutor data from `tutoruse` to `allschools`
  - `allschools <- merge(tutoruse, allschools, by='Classroom')`
Week 13: Data Management & Level-2 Variables

- Follow-ups
- Data Management in R
  - rbind()
  - merge()
  - melt()
- Level-2 Fixed & Random Effects
  - What do level-2 variables do?
  - Continuous or categorical?
    - Median splits
    - Extreme groups design
  - Good measurement
    - Reliability
    - Validity
melt()

- Need to install package `reshape`

- For `lmer()`, each observation needs its own row
  - “long” format
    - Time 1 row
    - Time 2 gets a separate row

- Sometimes data comes to us in “wide” format
  - Each repeated measure is a different column in the same row
    - Time 1 and Time 2 are considered separate variables
melt()

- Need to install package reshape
  - Then do `library(reshape)`

- `melt()` turns “wide” data into “long” data

- `melteddata <- melt(allschools, measure.vars=c('Pretest', 'Posttest'),` [Pretest and Posttest are the columns that we want to convert into separate observations](#)

  (often, repeated measures on the same individual)
**melt()**

- Need to install package **reshape**
  - Then do `library(reshape)`

- **melt()** turns "wide" data into "long" data

- `melteddata <- melt(allschools, measure.vars=c('Pretest', 'Posttest'))`

But, we need some way to preserve student, school, & classroom IDs and SES/hours of study.

Don’t want to treat Student SES as though it were the outcome from a 3rd session!
melt()

- Need to install package `reshape`
  - Then do `library(reshape)`

- `melt()` turns “wide” data into “long” data

```r
melteddata <- melt(allschools, measure.vars=c('Pretest', 'Posttest'), id.vars=c('Student', 'Classroom', 'School', 'StudentSES', 'HoursOfStudy'))
```

**id.vars** are columns that should stay as separate columns:
- IDs for students, classrooms, schools
- Between-subjects variables that are constant: StudentSES and HoursOfStudy
melt()

- Need to install package `reshape`
  - Then do `library(reshape)`

- `melt()` turns “wide” data into “long” data

```r
melteddata <- melt(allschools, measure.vars=c('Pretest', 'Posttest'), id.vars=c('Student', 'Classroom', 'School', 'StudentSES', 'HoursOfStudy'), variable_name='Session')
```

We’re creating a new variable to distinguish between the pretest & posttest sessions

Let’s call it Session (but could be anything you want)
melt(): The Results

- summary(melteddata)

- Now we have 2 rows per student: A “Pretest” row and a “Posttest” row
- Can now include Session as a predictor variable in lmer
- This column is named Session because that’s what we set the variable_name parameter to:
  - melteddata <- melt(allschools, 
    measure.vars=c('Pretest', 'Posttest'), 
    id.vars=c('Student', 'Classroom', 'School', 'StudentSES', 'HoursOfStudy'), variable_name='Session')
melt(): The Results

- `summary(melteddata)`

- DV is just called `value` by default because R has no way of knowing what it represents
- We can change that:
  - `colnames(melteddata)[colnames(melteddata) == 'value'] <- 'MathScore'`
melt(): Extra Practice!

- If you completed the `merge()` practice earlier, `allschools` will also have a `Tutor` column that we want to preserve when we convert to long format.
- Where should we add this in the `melt()` call?
**melt(): Extra Practice!**

- If you completed the `merge()` practice earlier, `allschools` will also have a `Tutor` column that we want to preserve when we convert to long format.
- Where should we add this in the `melt()` call?

```r
melteddata <- melt(allschools, 
measure.vars = c('Pretest', 'Posttest'),
id.vars = c('Student', 'Classroom', 'School',
           'StudentSES', 'HoursOfStudy', 'Tutor'),
variable_name = 'Session')
```

*id.vars are the columns that should stay as-is*
melt()

- Need to install package `reshape`
  - Then do `library(reshape)`

- `melt()` turns “wide” data into “long” data

- Also a corresponding function, `cast()`, to turn “long” format data into “wide” format data
  - Analogy: Casting molten steel

- Other, newer packages for reshaping data: `reshape2` and `dplyr`
Summary

- Data is already in one data frame but you need to rearrange it:

- Same variables in more than one file:

- Different variables in more than one file:
Summary

- Data is already in one data frame but you need to rearrange it:
  - `melt()`

- Same variables in more than one file:
  - `rbind()`

- Different variables in more than one file:
  - `merge()`
Week 13: Data Management & Level-2 Variables

- Follow-ups
- Data Management in R
  - rbind()
  - merge()
  - melt()
- Level-2 Fixed & Random Effects
  - What do level-2 variables do?
  - Continuous or categorical?
    - Median splits
    - Extreme groups design
  - Good measurement
    - Reliability
    - Validity
**Level-2 Fixed and Random Effects**

- Let’s consider one model of our lexical decision data:
  - `model1 <- lmer(RT ~ 1 + PrevTrials + (1|Subject) + (1|Word), data=lexdec2)`

- Hierarchical linear model notation for this:
  - **Lv.2 (Item):** \( Y_{00k} = u_{00k} \)
  - **Lv.2 (Subj.):** \( Y_{0j0} = u_{0j0} \)
  - **Lv.1 (Trial):** \( Y_{ijk} = Y_{000} + \gamma_{100}\text{PrevTrials} + Y_{0j0} + Y_{00k} + e_{ijk} \)

- Level 2 model predicts the effect of item \( k \)

- Could substitute random intercept into the level 1 model
Level-2 Fixed and Random Effects

- Now let's add a fixed effect of word frequency:
  - `model2 <- lmer(RT ~ 1 + PrevTrials + WordFreq + (1|Subject) + (1|Word), data=lexdec2)`

- Which level does this characterize?:
  - **Lv.2 (Item):** $Y_{00k} = u_{00k}$
  - **Lv.2 (Subj.):** $Y_{0j0} = u_{0j0}$
  - **Lv.1 (Trial):** $Y_{ijk} = Y_{000} + Y_{100} \text{PrevTrials} + Y_{0j0} + Y_{00k} + e_{ijk}$

  Level 2 model predicts the effect of item $k$

  Could substitute random intercept into the level 1 model
Level-2 Fixed and Random Effects

- Now let’s add a fixed effect of word frequency:
  - `model2 <- lmer(RT ~ 1 + PrevTrials + WordFreq + (1|Subject) + (1|Word), data=lexdec2)`

- Helps us predict the effect of a particular item:
  - Lv.2 (Item): $Y_{00k} = Y_{001}\text{WordFreq} + u_{00k}$
  - Lv.2(Subj.): $Y_{0j0} = u_{0j0}$
  - Lv.1(Trial): $Y_{ijk} = Y_{000} + Y_{100}\text{PrevTrials} + Y_{0j0} + Y_{00k} + e_{ijk}$
What Changes?

- Random item variance is greatly reduced.
- Word frequency accounts for a lot of the variance among items.
- Word frequency explains a lot of the “Item $k$” effect we’re substituting into the level 1 equation. No longer just a random intercept.
What Didn’t Change?

- Level 1 fixed effect (PrevTrials) and error term essentially unchanged.
- Doesn’t matter what explains the “Item $k$” effect; still substituting into the same Lv 1 model.
- Note that WordFreq & PrevTrials effects are slightly correlated (due to random sampling of item orders); otherwise, there’d be no change.
What Didn’t Change?

- Estimated variance in subject intercept also essentially the same
- Explaining where the “Item k” effect comes from doesn’t change the “Subject j” effect
Week 13: Data Management & Level-2 Variables

- Follow-ups
- Data Management in R
  - rbind()
  - merge()
  - melt()
- Level-2 Fixed & Random Effects
  - What do level 2 variables do?
  - Continuous or categorical?
    - Median splits
    - Extreme groups design
  - Good measurement
    - Reliability
    - Validity
In ANOVA, subject & item differences typically examined as *categorical* variables.

- **e.g. median split:**
  - `median(lexcdec2$WordFreq, na.rm=TRUE) = 3.30`
  - Word frequencies above the median are in category A and words below it are in category B.

---

**Continuous or Categorical Predictors?**

- In ANOVA, subject & item differences typically examined as *categorical* variables.
- **e.g. median split:**
  - `median(lexcdec2$WordFreq, na.rm=TRUE) = 3.30`
  - Word frequencies above the median are in category A and words below it are in category B.
Continuous or Categorical Predictors?

- In ANOVA, subject & item differences typically examined as **categorical** variables
- e.g. **median split**: median(lexcdec2$WordFreq, na.rm=TRUE) = 3.30
- Word frequencies above the median are in category A and words below it are in category B
Evaluating Median Splits

- Median splits are noisy and discard info.
- Ignores all within-category variation

Median split considers these both equally “low-frequency” words

- *pomegranate* (WF: 1.1461)
- *glasses* (WF: 3.2279)
Evaluating Median Splits

- Median splits are noisy and discard info.
  - Ignores all within-category information
  - High probability of misclassification

If our measures of word frequency were even slightly off, these words could have ended up in the opposite categories!

- glasses (WF: 3.2279)
- chair (WF: 3.400)
Evaluating Median Splits

- Median splits are noisy and discard info.
  - Ignores all within-category information
  - High probability of misclassification
- Greatly reduces power and estimated effect size (Cohen, 1983)

![Word frequency vs. Mean RT scatter plot]

- X-axis: Word frequency
- Y-axis: Mean RT
- Dashed line represents median split
Evaluating Median Splits

- Median splits are **noisy** and **discard info.**
  - Ignores all within-category information
  - High probability of misclassification
- Greatly reduces power and estimated effect size (Cohen, 1983)
- Also, comparing two categories can’t tell us about the form of the relationship (as polynomial contrasts can)

![Graphs showing different trends](image1.png)

- If continuous variation (in word frequency, second language proficiency, etc.) measured, better to include it in the model
Week 13: Data Management & Level-2 Variables

- Follow-ups
- Data Management in R
  - rbind()
  - merge()
  - melt()
- Level-2 Fixed & Random Effects
  - What do level-2 variables do?
  - Continuous or categorical?
    - Median splits
      - Extreme groups design
  - Good measurement
    - Reliability
    - Validity
In some cases, we might deliberately sample only very low- and very high-frequency words. Extreme group design.

Now, we don’t know what the full relation is.
In some cases, we might deliberately sample only very low- and very high-frequency words:

- Extreme group design
- Now, we don’t know what the full relation is

![Graph showing the relationship between word frequency and mean RT.](image-url)
Extreme Group Designs

- In some cases, we might deliberately sample only very low- and very high-frequency words
- Extreme group design
- Now, we don’t know what the full relation is
Extreme Group Designs

- In some cases, we might deliberately sample only very low- and very high-frequency words
  - Extreme group design
- Now, we don’t know what the full relation is
  - *Should* treat this as a categorical variable (reflects design)
**Extreme Group Designs: Evaluation**

- May overestimate effect size
- Still, better than median splits if you want to do a categorical design (Conway et al., 2005)
  - e.g., you only care whether a difference exists (not its size / shape)
Breakpoints

- When you have a continuous variable, but you think there’s a qualitative shift at some point in the range
  - e.g., below vs above the poverty line
- Add a categorical variable that represents whether or not you’re above the point at which the shift happens

Main effect of breakpoint only – single shift downward but same slope

Main effect of breakpoint & interaction – slopes also changes
Week 13: Data Management & Level-2 Variables

- **Follow-ups**
- **Data Management in R**
  - `rbind()`
  - `merge()`
  - `melt()`
- **Level-2 Fixed & Random Effects**
  - What do level 2 variables do?
  - Continuous or categorical?
    - Median splits
    - Extreme groups design
- **Good measurement**
  - Reliability
  - Validity
Good Measurement: Reliability

- Suppose we find that a measure of working memory is unrelated to people’s moral judgments
  - Maybe these are truly unrelated
  - Or, maybe we just failed to accurately measure WM and/or moral reasoning

- Not all measures are good measures
  - Measures may be noisy
  - Measures may not measure a stable or meaningful characteristic of people/items/schools
Good Measurement: Reliability

- Good measures produce consistent scores
  - Across times (test-retest reliability)
  - Across items (internal consistency)
  - Across judges (inter-rater reliability)
- Shows you’re measuring something real

- If measures can’t even predict themselves, they can’t predict anything else!
Week 13: Data Management & Level-2 Variables

- Follow-ups
- Data Management in R
  - rbind()
  - merge()
  - melt()
- Level-2 Fixed & Random Effects
  - What do level 2 variables do?
  - Continuous or categorical?
    - Median splits
    - Extreme groups design
- Good measurement
  - Reliability
  - Validity
Good Measurement: Validity

- Even if we have a *reliable* measure, no guarantee it measures the thing we *think* it measures
  - You’re measuring *something*, but what is it?
  - *Examples of tests that produce consistent results but don’t measure what we want:* BMI, polygraph, stopped clock
Good Measurement: Validity

- Valid measures should show (among other things):
  - **Convergent validity**: Correlate with *other* measures of this construct

**Reading Span**
- Task: Remember words while verifying sentences

**Operation Span**
- Task: Remember words while verifying equations

**An official who manages a state is called a governor.** (T / F)?

**3 x 4 = 12** (T / F)?

Here, two tasks designed to measure working memory correlate.
Good Measurement: Validity

- Valid measures should show (among other things):
  - **Convergent validity**: Correlate with *other* measures of this construct
  - **Divergent validity**: *Don’t* correlate with things that are supposed to be *different*
    - If “working memory” task correlates with years of education or socioeconomic status, might not be measuring what we thought
Good Measurement: Validity

- Valid measures should show (among other things):
  - **Convergent validity:** Correlate with *other* measures of this construct
  - **Divergent validity:** *Don’t* correlate with things that are supposed to be *different*
    - Do higher Working Memory scores predict second language learning just because subjects who are “smarter” or more motivated do well on both tasks?
    - Or is this unique to WM?
    - Measuring only 1 construct makes it difficult to tell where the locus of an effect lies