Common Issues and Solutions in Regression Modeling (Mixed or not)

Day 2

Florian Jaeger

January 31, 2010
Acknowledgments

▷ I’ve incorporated slides prepared by:
  ▷ Victor Kuperman (Stanford)
  ▷ Roger Levy (UCSD)
... with their permission (naturalmente!)
▷ I am also grateful for feedback from:
  ▷ Austin Frank (Rochester)
  ▷ Previous audiences to similar workshops at CUNY, Haskins, Rochester, Buffalo, UCSD, MIT.
Hypothesis testing in psycholinguistic research

- Typically, we make predictions not just about the existence, but also the *direction* of effects.
- Sometimes, we’re also interested in effect *shapes* (non-linearities, etc.)
- Unlike in ANOVA, regression analyses reliably test hypotheses about effect direction and shape without requiring post-hoc analyses if (a) the *predictors in the model are coded appropriately* and (b) the *model can be trusted*.
- **Today**: Provide an overview of (a) and (b).
Overview

- Introduce sample data and simple models
- Towards a model with interpretable coefficients:
  - outlier removal
  - transformation
  - coding, centering, ...
  - collinearity
- Model evaluation:
  - fitted vs. observed values
  - model validation
  - investigation of residuals
  - case influence, outliers
- Model comparison
- Reporting the model:
  - comparing effect sizes
  - back-transformation of predictors
  - visualization
Data 1: Lexical decision RTs

- **Outcome:** log lexical decision latency RT
- **Inputs:**
  - factors Subject (21 levels) and Word (79 levels),
  - factor NativeLanguage *(English and Other)*
  - continuous predictors Frequency (log word frequency), and Trial (rank in the experimental list).

<table>
<thead>
<tr>
<th>Subject</th>
<th>RT</th>
<th>Trial</th>
<th>NativeLanguage</th>
<th>Word</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A1 6.340359</td>
<td>23</td>
<td>English</td>
<td>owl</td>
<td>4.859812</td>
</tr>
<tr>
<td>2</td>
<td>A1 6.308098</td>
<td>27</td>
<td>English</td>
<td>mole</td>
<td>4.605170</td>
</tr>
<tr>
<td>3</td>
<td>A1 6.349139</td>
<td>29</td>
<td>English</td>
<td>cherry</td>
<td>4.997212</td>
</tr>
<tr>
<td>4</td>
<td>A1 6.186209</td>
<td>30</td>
<td>English</td>
<td>pear</td>
<td>4.727388</td>
</tr>
<tr>
<td>5</td>
<td>A1 6.025866</td>
<td>32</td>
<td>English</td>
<td>dog</td>
<td>7.667626</td>
</tr>
<tr>
<td>6</td>
<td>A1 6.180017</td>
<td>33</td>
<td>English</td>
<td>blackberry</td>
<td>4.060443</td>
</tr>
</tbody>
</table>
Data 2: Lexical decision response

- **Outcome**: Correct or incorrect response (Correct)
- **Inputs**: same as in linear model

```r
> lmer(Correct == "correct" ~ NativeLanguage +
+ Frequency + Trial +
+ (1 | Subject) + (1 | Word),
+ data = lexdec, family = "binomial")
```

**Random effects:**
- **Groups** | **Name** | **Variance** | **Std.Dev.**
  - Word | (Intercept) | 1.01820 | 1.00906
  - Subject | (Intercept) | 0.63976 | 0.79985

**Number of obs:** 1659, **groups:** Word, 79; Subject, 21

**Fixed effects:**

|              | Estimate | Std. Error | z value | Pr(>|z|) |
|--------------|----------|------------|---------|----------|
| (Intercept)  | -1.746e+00 | 8.206e-01 | -2.128  | 0.033344 * |
| NativeLanguageOther | -5.726e-01 | 4.639e-01 | 1.234   | 0.217104 |
| Frequency    | 5.600e-01  | 1.570e-01 | -3.567  | 0.000361 *** |
| Trial        | 4.443e-06  | 2.965e-03 | 0.001   | 0.998804 |

Building an interpretable model

Data exploration
Transformation
Coding
Centering
Interactions and modeling of non-linearities
Collinearity
What is collinearity?
Detecting collinearity
Dealing with collinearity

Model Evaluation
Beware overfitting
Detect overfitting: Validation
Goodness-of-fit
Aside: Model Comparison

Reporting the model
Describing Predictors
What to report
Back-transforming coefficients
Comparing effect sizes
Visualizing effects
Interpreting and reporting interactions

Discussion

Modeling schema

Generalized Linear Mixed Models
Florian Jaeger
Data exploration

- outcome
- input var 1
- input var 2
- input var 3
- input var 4
- ...
- input var n

Generalized Linear Mixed Models
Florian Jaeger

Building an interpretable model
Data exploration
Transformation
Coding
Centering
Interactions and modeling of non-linearities
Collinearity
What is collinearity?
Detecting collinearity
Dealing with collinearity

Model Evaluation
Beware overfitting
Detect overfitting:
Validation
Goodness-of-fit
Aside: Model Comparison

Reporting the model
Describing Predictors
What to report
Back-transforming coefficients
Comparing effect sizes
Visualizing effects
Interpreting and reporting interactions

Discussion
Data exploration

- Select and understand input variables and outcome based on a-priori theoretical consideration
  - How many parameters does your data afford (overfitting)?
- Data exploration: Before fitting the model, explore inputs and outputs
  - Outliers due to missing data or measurement error (e.g. RTs in SPR < 80msecs).
  - NB: postpone distribution-based outlier exclusion until after transformations
  - Skewness in distribution can affect the accuracy of model’s estimates (transformations).
Understanding variance associated with potential random effects

- explore candidate predictors (e.g., Subject or Word) for level-specific variation.

> boxplot(RT ~ Subject, data = lexdec)

→ Huge variance.
Random effects (cnt’d)

▶ explore variation of level-specific slopes.

> xyloewss.fnc(RT ~ Trial | Subject,
> type = c("g", "smooth"), data = lexdec)

→ not too much variance.

▶ random effect inclusion test via \(\sim\) model comparison
Understanding input variables

- Explore:
  - correlations between predictors (collinearity).
  - non-linearities may become obvious (lowess).

> pairscor.fnc(lexdec[,c("RT", "Frequency", "Trial", "NativeLanguage")])
Non-linearities

- Consider Frequency (already log-transformed in lexdec) as predictor of RT:

→ Assumption of a linearity may be inaccurate.
  - Select appropriate \( transformation \): log, power, sinusoid, etc.
  - or use polynomial \( \text{poly}() \) or splines \( \text{rcs}(), \text{bs}() \), etc. to \( \text{model non-linearities} \).
Transformation

Generalized Linear Mixed Models
Florian Jaeger

Building an interpretable model
Data exploration
Transformation
Coding
Centering
Interactions and modeling of non-linearities
Collinearity
What is collinearity?
Detecting collinearity
Dealing with collinearity
Model Evaluation
Beware overfitting
Detect overfitting: Validation
Goodness-of-fit
Aside: Model Comparison
Reporting the model
Describing Predictors
What to report
Back-transforming coefficients
Comparing effect sizes
Visualizing effects
Interpreting and reporting interactions
Discussion
Transformation

- Reasons to transform:
  - Conceptually motivated (e.g. log-transformed probabilities)
  - Can reduce non-linear to linear relations (cf. previous slide)
  - Remove skewness (e.g. by log-transform)
- Common transformation: log, square-root, power, or inverse transformation, etc.

![Graphs showing density plots of raw and log-transformed RT and frequency distributions.](image)
Coding and centering predictors

- Transforming
- Coding centering
- Input var 1 → predictor 1
- Input var 2 → predictor 2
- Input var 3 → predictor 3
- Input var 4 → predictor 4
- Input var n → predictor m
- Outlier removal

Generalized Linear Mixed Models
Florian Jaeger

Building an interpretable model
- Data exploration
- Transformation
- Coding
- Centering
- Interactions and modeling of non-linearities
- Collinearity
- What is collinearity?
- Detecting collinearity
- Dealing with collinearity

Model Evaluation
- Beware overfitting
- Detecting overfitting: Validation
- Goodness-of-fit
- Aside: Model Comparison

Reporting the model
- Describing Predictors
- What to report
- Back-transforming coefficients
- Comparing effect sizes
- Visualizing effects
- Interpreting and reporting interactions

Discussion
Coding affects interpretation

Consider a simpler model:

```r
> lmer(RT ~ NativeLanguage +
+ (1 | Word) + (1 | Subject), data = lexdec)
```

```

<table>
<thead>
<tr>
<th></th>
<th>AIC</th>
<th>BIC</th>
<th>logLik</th>
<th>deviance</th>
<th>REMLdev</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>886.1</td>
<td>853.6</td>
<td>449.1</td>
<td>926.2</td>
<td>898.1</td>
</tr>
</tbody>
</table>

Random effects:

<table>
<thead>
<tr>
<th>Groups</th>
<th>Name</th>
<th>Variance</th>
<th>Std.Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word</td>
<td>(Intercept)</td>
<td>0.0045808</td>
<td>0.067682</td>
</tr>
<tr>
<td>Subject</td>
<td>(Intercept)</td>
<td>0.0184681</td>
<td>0.135897</td>
</tr>
<tr>
<td>Residual</td>
<td></td>
<td>0.0298413</td>
<td>0.172746</td>
</tr>
</tbody>
</table>

Number of obs: 1659, groups: Word, 79; Subject, 21

Fixed effects:

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>Std. Error</th>
<th>t value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>6.32358</td>
<td>0.03783</td>
<td>167.14</td>
</tr>
<tr>
<td>NativeLanguageOther</td>
<td>0.15003</td>
<td>0.05646</td>
<td>2.66</td>
</tr>
</tbody>
</table>
```

- **Treatment (a.k.a. dummy) coding** is standard in most stats programs
  - NativeLanguage coded as 1 if “other”, 0 otherwise.
  - Coefficient for (Intercept) reflects reference level English of the factor NativeLanguage.
  - Prediction for NativeLanguage = Other is derived by \(6.32358 + 0.15003 = 6.47361\) (log-transformed reaction times).
Recoding

- Coding affects interpretation of coefficients.
- E.g., we can recode NativeLanguage into NativeEnglish:

```r
> lexdec$NativeEnglish = ifelse(lexdec$NativeLanguage == "English", 1, 0)
> lmer(RT ~ NativeEnglish + Frequency +
+ (1 | Word) + (1 | Subject), data = lexdec)
<...>

AIC    BIC   logLik deviance REMLdev
-886.1 -853.6  449.1   -926.6  -898.1

Random effects:
  Groups     Name    Variance  Std.Dev.
  Word     (Intercept)  0.0045808  0.067682
  Subject  (Intercept)  0.0184681  0.135897
  Residual              0.0298413  0.172746

Number of obs: 1659, groups: Word, 79; Subject, 21

Fixed effects:                           Estimate  Std. Error t value  
(Intercept)                         6.32358    0.03783   167.14
NativeEnglish                    -0.15003    0.05646    2.66
<...>
```

- NB: Goodness-of-fit (AIC, BIC, loglik, etc.) is not affected by choice between different sets of orthogonal contrasts.
Other codings of factor

- Treatment coding . . .
  - makes intercept hard to interpret.
  - leads to collinearity with interactions
- Sum (a.k.a. contrast) coding avoids that problem (in balanced data sets) and makes intercept interpretable (in factorial analyses of balanced data sets).
  - Corresponds to ANOVA coding.
  - Centers for balanced data set.
  - Caution when reporting effect sizes! (R contrast codes as $-1$ vs. $1 \rightarrow$ coefficient estimate is only half of estimated group difference).
- Other contrasts possible, e.g. to test hypothesis that levels are ordered (contr.poly(), contr.helmert()).
Centering predictors

- **Centering**: removal of the mean out of a variable...
  - makes coefficients more interpretable.
  - if all predictors are centered $\rightarrow$ intercept is estimated grand mean.
  - reduces $\sim$ collinearity of predictors
    - with intercept
    - higher-order terms that include the predictor (e.g. interactions)

- **Centering** does not change...
  - coefficient estimates (it’s a linear transformations); including random effect estimates.
  - $\sim$ Goodness-of-fit of model (information in the model is the same)
Centering: An example

- Re-consider the model with NativeEnglish and Frequency. Now with a centered predictors:

```r
> lexdec$cFrequency = lexdec$Frequency - mean(lexdec$Frequency)
> lmer(RT ~ cNativeEnglish + cFrequency +
+     (1 | Word) + (1 | Subject), data = lexdec)
<...>
Fixed effects:
                    Estimate Std. Error t value
(Intercept)         6.385090  0.030570 208.87
cNativeEnglish      -0.155821  0.060532  -2.57
cFrequency          -0.042872  0.005827  -7.36

Correlation of Fixed Effects:
                       (Intr) cNtvEn
(Intercept)             0.000
NativeEnglish           0.000  0.000
Frequency               0.000  0.000
<...>
```

→ Correlation between predictors and intercept gone.
→ Intercept changed (from 6.678 to 6.385 units): now grand mean (previously: prediction for Frequency=0!)
→ NativeEnglish and Frequency coefs unchanged.
Centering: An interaction example

- Let's add an interaction between NativeEnglish and Frequency.
- Prior to centering: interaction is collinear with main effects.

```r
> lmer(RT ~ NativeEnglish * Frequency +
+       (1 | Word) + (1 | Subject), data = lexdec)
<...>
Fixed effects:  Estimate  Std. Error t value  
(Intercept)      6.752403 0.056810 118.86 
NativeEnglish   -0.286343 0.068368  -4.19 
Frequency        -0.058570 0.006969  -8.40 
NativeEnglish:Frequency  0.027472 0.006690   4.11 

Correlation of Fixed Effects:  
            (Intr)    NtvEng   Frqncy
NativEngls  -0.688 
Frequency   -0.583  0.255
NtvEnglish:F  0.320 -0.465 -0.549
<...>
```
Centering: An interaction example (cnt’d)

- After centering:

```
<...>
Fixed effects:               Estimate  Std. Error t value
(Intercept)        6.385090  0.030572  208.85
cNativeEnglish    -0.155821  0.060531  -2.57
cFrequency        -0.042872  0.005827  -7.36
cNativeEnglish:cFrequency  0.027472  0.006690   4.11

Correlation of Fixed Effects:
                          (Intr)     cNtvEn   cFrqnc
cNativeEnglish     0.000
cFrequency         0.000  0.000
cNativeEnglish:F   0.000  0.000  0.000
<...>
```
Interactions and modeling of non-linearities

- Data exploration
- Transformation
- Coding
- Centering
- Interactions and modeling of non-linearities
- Collinearity
  - What is collinearity?
  - Detecting collinearity
  - Dealing with collinearity
- Model Evaluation
  - Beware overfitting
  - Detect overfitting: Validation
  - Goodness-of-fit
- Aside: Model Comparison
- Reporting the model
  - Describing Predictors
  - What to report
  - Back-transforming coefficients
  - Comparing effect sizes
  - Visualizing effects
  - Interpreting and reporting interactions
- Discussion
Interactions and non-linearities

- Include interactions after variables are centered → avoids unnecessary \( \bowtie \) collinearity.
- The same holds for higher order terms when non-linearities in continuous (or ordered) predictors are modeled. Though often centering will not be enough.
- See for yourself: a polynomial of (back-transformed) frequency

```r
> lexdec$rawFrequency <- round(exp(lexdec$Frequency),0)
> lmer(RT ~ poly(rawFrequency,2) +
+ (1 | Word) + (1 | Subject), data = lexdec)
```

- ... vs. a polynomial of the centered (back-transformed) frequency

```r
> lexdec$crawFrequency = lexdec$rawFrequency - mean(lexdec$rawFrequency)
> lmer(RT ~ poly(crawFrequency,2) +
+ (1 | Word) + (1 | Subject), data = lexdec)
```
Collinearity

- Data exploration
- Transformation
- Coding
- Centering
- Interactions and modeling of non-linearities
- What is collinearity?
- Detecting collinearity
- Dealing with collinearity

Model Evaluation
- Beware overfitting
- Detect overfitting: Validation
- Goodness-of-fit
- Aside: Model Comparison

Reporting the model
- Describing Predictors
- What to report
- Back-transforming coefficients
- Comparing effect sizes
- Visualizing effects
- Interpreting and reporting interactions

Discussion
Definition of collinearity

 ► **Collinearity**: a predictor is collinear with other predictors in the model if there are high (partial) correlations between them.

 ► Even if a predictor is not highly correlated with any single other predictor in the model, it can be highly collinear with the combination of predictors → collinearity will affect the predictor

 ► This is not uncommon!
  ▶ in models with many predictors
  ▶ when several somewhat related predictors are included in the model (e.g. word length, frequency, age of acquisition)
Consequences of collinearity

→ standard errors $SE(\beta)$s of collinear predictors are biased (inflated).
  → tends to underestimate significance (but see below)
→ coefficients $\beta$ of collinear predictors become hard to interpret (though not biased)
  ▶ ‘bouncing betas’: minor changes in data might have a major impact on $\beta$s
  ▶ coefficients will flip sign, double, half
→ coefficient-based tests don’t tell us anything reliable about collinear predictors!
Extreme collinearity: An example

- Drastic example of collinearity: meanWeight (rating of the weight of the object denoted by the word, averaged across subjects) and meanSize (average rating of the object size) in lexdec.

```
lmer(RT ~ meanSize + (1 | Word) + (1 | Subject), data = lexdec)
```

<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>6.3891053</td>
<td>0.0427533</td>
<td>149.44</td>
</tr>
<tr>
<td>meanSize</td>
<td>-0.0004282</td>
<td>0.0094371</td>
<td>-0.05</td>
</tr>
</tbody>
</table>

- n.s. correlation of meanSize with RTs.
- similar n.s. weak negative effect of meanWeight.
- The two predictors are highly correlated ($r > 0.999$).
Extreme collinearity: An example (cnt’d)

- If the two correlated predictors are included in the model …

```r
> lmer(RT ~ meanSize + meanWeight +
+ (1 | Word) + (1 | Subject), data = lexdec)

Fixed effects:

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>Std. Error</th>
<th>t value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>5.7379</td>
<td>0.1187</td>
<td>48.32</td>
</tr>
<tr>
<td>meanSize</td>
<td>1.2435</td>
<td>0.2138</td>
<td>5.81</td>
</tr>
<tr>
<td>meanWeight</td>
<td>-1.1541</td>
<td>0.1983</td>
<td>-5.82</td>
</tr>
</tbody>
</table>

Correlation of Fixed Effects:

<table>
<thead>
<tr>
<th></th>
<th>(Intr)</th>
<th>meanSz</th>
</tr>
</thead>
<tbody>
<tr>
<td>meanSize</td>
<td>-0.949</td>
<td></td>
</tr>
<tr>
<td>meanWeight</td>
<td>0.942</td>
<td>-0.999</td>
</tr>
</tbody>
</table>
```

- SE($\beta$)s are hugely inflated (more than by a factor of 20)
- large and highly significant significant counter-directed effects ($\beta$s) of the two predictors
- collinearity needs to be investigated!
Extreme collinearity: An example (cnt’d)

- Objects that are perceived to be unusually heavy for their size tend to be more frequent (accounts for 72% of variance in frequency).

- Both effects apparently disappear though when frequency is included in the model (but cf. residualization → meanSize or meanWeight still has small expected effect beyond Frequency).

Fixed effects:

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>Std. Error</th>
<th>t value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>6.64846</td>
<td>0.06247</td>
<td>106.43</td>
</tr>
<tr>
<td>cmeanSize</td>
<td>-0.11873</td>
<td>0.35196</td>
<td>-0.34</td>
</tr>
<tr>
<td>cmeanWeight</td>
<td>0.13788</td>
<td>0.33114</td>
<td>0.42</td>
</tr>
<tr>
<td>Frequency</td>
<td>-0.05543</td>
<td>0.01098</td>
<td>-5.05</td>
</tr>
</tbody>
</table>
So what does collinearity do?

- Type II error increases → power loss

```r
h <- function(n) {
  x <- runif(n)
  y <- x + rnorm(n, 0, 0.01)
  z <- ((x + y) / 2) + rnorm(n, 0, 0.2)

  m <- lm(z ~ x + y)
  signif.m.x <- ifelse(summary(m)$coef[2, 4] < 0.05, 1, 0)
  signif.m.y <- ifelse(summary(m)$coef[3, 4] < 0.05, 1, 0)

  mx <- lm(z ~ x)
  my <- lm(z ~ y)
  signif.mx.x <- ifelse(summary(mx)$coef[2, 4] < 0.05, 1, 0)
  signif.my.y <- ifelse(summary(my)$coef[2, 4] < 0.05, 1, 0)
  return(c(cor(x, y), signif.m.x, signif.m.y, signif.mx.x, signif.my.y))
}
result <- sapply(rep(M, n), h)
print(paste("x in combined model:", sum(result[2,])))
print(paste("y in combined model:", sum(result[3,])))
print(paste("x in x-only model:", sum(result[4,])))
print(paste("y in y-only model:", sum(result[5,])))
print(paste("Avg. correlation:", mean(result[1,])))
```
So what does collinearity do?

- Type II error increases → power loss
- Type I error does not increase much (5.165% Type I error for two predictors with $r > 0.9989$ in joined model vs. 5.25% in separate models; 20,000 simulation runs with 100 data points each)

```r
set.seed(1)
n <- 100
M <- 20000
f <- function(n) {
x <- runif(n)
y <- x + rnorm(n, 0, 0.01)
z <- rnorm(n, 0, 5)
m <- lm(z ~ x + y)
mx <- lm(z ~ x)
my <- lm(z ~ y)
signifmin <- ifelse(min(summary(m)$coef[2:3,4]) < 0.05, 1, 0)
signifx <- ifelse(min(summary(mx)$coef[2,4]) < 0.05, 1, 0)
signify <- ifelse(min(summary(my)$coef[2,4]) < 0.05, 1, 0)
signifxory <- ifelse(signifx == 1 | signify == 1, 1, 0)
return(c(cor(x,y), signifmin, signifx, signify, signifxory))
}
result <- sapply(rep(n,M), f)
sum(result[2,])/M # joined model returns >=1 spurious effect
sum(result[3,])/M
sum(result[4,])/M
sum(result[5,])/M # two individual models return >=1 spurious effect
min(result[1,])
```
So what does collinearity do?

- Type II error increases → power loss
- Type I error does not increase (much)

★ But small differences between highly correlated predictors can be highly correlated with another predictors and create ‘apparent effects’ (like in the case discussed).
  → Can lead to misleading effects (not technically spurious, but if they we interpret the coefficients causally we will have a misleading result!).
  - This problem is not particular to collinearity, but it frequently occurs in the case of collinearity.

- When coefficients are unstable (as in the above case of collinearity) treat this as a warning sign - check for mediated effects.
Detecting collinearity

- Mixed model output in R comes with correlation matrix (cf. previous slide).
  - Partial correlations of fixed effects *in the model*.
- Also useful: correlation matrix (e.g. cor(); use Spearman option for categorical predictors) or pairscor.fnc() in languageR for visualization.
  - apply to predictors (not to untransformed input variables)!

```r
> cor(lexdec[,c(2,3,10, 13)])
```

<table>
<thead>
<tr>
<th></th>
<th>RT</th>
<th>Trial</th>
<th>Frequency</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>RT</td>
<td>1.0000000</td>
<td>-0.0524113</td>
<td>-0.2132495</td>
<td>0.146738111</td>
</tr>
<tr>
<td>Trial</td>
<td>-0.0524113</td>
<td>1.000000000</td>
<td>-0.006849117</td>
<td>0.009865814</td>
</tr>
<tr>
<td>Frequency</td>
<td>-0.2132495</td>
<td>-0.006849117</td>
<td>1.000000000</td>
<td>-0.427338136</td>
</tr>
<tr>
<td>Length</td>
<td>0.1467381</td>
<td>0.009865814</td>
<td>-0.427338136</td>
<td>1.000000000</td>
</tr>
</tbody>
</table>
Formal tests of collinearity

- Variance inflation factor (VIF, `vif()`).
  - generally, $VIF > 10 \rightarrow$ absence of absolute collinearity in the model cannot be claimed.
  - $VIF > 4$ are usually already problematic.
  - $VIF > 2$ can lead inflated standard errors.

- Kappa (e.g. `collin.fnc()` in `languageR`)
  - generally, $\kappa > 10 \rightarrow$ mild collinearity in the model.

- Applied to current data set, . . .

```r
> collin.fnc(lexdec[,c(2,3,10,13)])$cnumber
```

- . . . gives us a $\kappa > 90 \rightarrow$ Houston, we have a problem.
Dealing with collinearity
Dealing with collinearity

- **Good news**: Estimates are only problematic for those predictors that are collinear.

→ If collinearity is in the nuisance predictors (e.g. certain controls), nothing needs to be done.

- **Somewhat good news**: If collinear predictors are of interest but we are not interested in the direction of the effect, we can use \(\Rightarrow\) **model comparison** (rather than tests based on the standard error estimates of coefficients).

- If collinear predictors are of interest and we are interested in the direction of the effect, we need to reduce collinearity of those predictors.
Reducing collinearity

► **Centering** ↝: reduces collinearity of predictor with intercept and higher level terms involving the predictor.
  ► **pros**: easy to do and interpret; often improves interpretability of effects.
  ► **cons**: none?

► **Re-express the variable** based on conceptual considerations (e.g. ratio of spoken vs. written frequency in lexdec; rate of disfluencies per words when constituent length and fluency should be controlled).
  ► **pros**: easy to do and relatively easy to interpret.
  ► **cons**: only applicable in some cases.
Reducing collinearity (cnt’d)

- **Stratification**: Fit separate models on subsets of data holding correlated predictor A constant.
- If effect of predictor B persists $\rightarrow$ effect is probably real.

  - **pros**: Still relatively easy to do and easy to interpret.
  - **cons**: harder to do for continuous collinear predictors; reduces power, $\rightarrow$ extra caution with null effects; doesn’t work for multicollinearity of several predictors.

- **Principal Component Analysis (PCA)**: for $n$ collinear predictors, extract $k < n$ most important orthogonal components that capture $> p\%$ of the variance of these predictors.

  - **pros**: Powerful way to deal with multicollinearity.
  - **cons**: Hard to interpret ($\rightarrow$ better suited for control predictors that are not of primary interest); technically complicated; some decisions involved that affect outcome.
Reduce collinearity (cnt’d)

- **Residualization**: Regress collinear predictor against combination of (partially) correlated predictors
  - usually using ordinary regression (e.g. `lm()`), `ols()`).
  - **pros**: systematic way of dealing with multicollinearity; directionality of (conditional) effect interpretable
  - **cons**: effect sizes hard to interpret; judgment calls: what should be residualized against what?
An example of moderate collinearity (cnt’d)

▶ Consider two moderately correlated variables 
\( (r = -0.49) \), (centered) word length and (centered log) frequency:

\[
\begin{align*}
&> \text{lmer(RT ~ cLength + cFrequency +} \\
&+ \quad (1 | \text{Word}) + (1 | \text{Subject}), \text{ data = lexdec) \\
&<...>
\end{align*}
\]

Fixed effects:

\[
\begin{array}{ccc}
\text{Estimate} & \text{Std. Error} & \text{t value} \\
\text{(Intercept)} & 6.385090 & 0.034415 & 185.53 \\
cLength & 0.009348 & 0.004327 & 2.16 \\
cFrequency & -0.037028 & 0.006303 & -5.87 \\
\end{array}
\]

Correlation of Fixed Effects:

\[
\begin{array}{ccc}
\text{(Intr)} & \text{cLength} \\
\text{cLength} & 0.000 \\
cFrequency & 0.000 & 0.429 \\
\end{array}
\]

▶ Is this problematic? Let’s remove collinearity via residualization
Residualization: An example

▶ Let’s regress word length vs. word frequency.

```r
> lexdec$rLength = residuals(lm(Length ~ Frequency, data = lexdec))
```

▶ \( rLength \): difference between actual length and length as predicted by frequency. Related to actual length \((r > 0.9)\), but crucially not to frequency \((r \ll 0.01)\).

▶ Indeed, collinearity is removed from the model:

```
<...>
Fixed effects:
   Estimate Std. Error t value
(Intercept)  6.385090  0.034415 185.53
rLength     0.009348  0.004327  2.16
cFrequency  -0.042872  0.005693  7.53

Correlation of Fixed Effects:
   (Intr) rLngth
cLength  0.000
rLength  0.000  0.000
cFrequency  0.000  0.000  0.000
<...>
```

→ \( \text{SE}(\beta) \) estimate for frequency predictor decreased
→ larger \( t \)-value
Residualization: An example (cnt’d)

- **Q:** What precisely is \( r\text{Length} \)?
- **A:** Portion of word length that is not explained by (a linear relation to log) word frequency.

→ Coefficient of \( r\text{Length} \) needs to be interpreted as such
- No trivial way of back-transforming to \( \text{Length} \).

- **NB:** We have granted frequency the entire portion of the variance that cannot unambiguously attributed to either frequency or length!

→ If we choose to residualize frequency on length (rather than the inverse), we may see a different result.
Understanding residualization

- So, let’s regress frequency against length.
- Here: no qualitative change, but word length is now *highly* significant (random effect estimates unchanged)

```r
> lmer(RT ~ cLength + rFrequency +
+       (1 | Word) + (1 | Subject), data = lexdec)
<...>
Fixed effects:

<table>
<thead>
<tr>
<th>Estimate</th>
<th>Std. Error</th>
<th>t value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>6.385090</td>
<td>0.034415</td>
</tr>
<tr>
<td>cLength</td>
<td>0.020255</td>
<td>0.003908</td>
</tr>
<tr>
<td>rFrequency</td>
<td>-0.037028</td>
<td>0.006303</td>
</tr>
</tbody>
</table>

Correlation of Fixed Effects:

<table>
<thead>
<tr>
<th></th>
<th>(Intr)</th>
<th>cLength</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intr)</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>cLength</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>
<...>
```

→ Choosing what to residualize, changes interpretation of $\beta$s and hence the hypothesis we’re testing.
Extreme collinearity: ctn’d

- we can now residualize `meanWeight` against `meanSize` and `Frequency`, and
- and residualize `meanSize` against `Frequency`.
- include the transformed predictors in the model.

```r
> lexdec$rmeanSize <- residuals(lm(cmeanSize ~ Frequency + cmeanWeight,
+                             data=lexdec))
> lexdec$rmeanWeight <- residuals(lm(cmeanWeight ~ Frequency,
+                                    data=lexdec))
> lmer(RT ~ rmeanSize + rmeanWeight + Frequency + (1|Subject) + (1|Word),
+      data=lexdec)
```

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>Std. Error</th>
<th>t value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>6.588778</td>
<td>0.043077</td>
<td>152.95</td>
</tr>
<tr>
<td>rmeanSize</td>
<td>-0.118731</td>
<td>0.351957</td>
<td>-0.34</td>
</tr>
<tr>
<td>rmeanWeight</td>
<td>0.026198</td>
<td>0.007477</td>
<td>3.50</td>
</tr>
<tr>
<td>Frequency</td>
<td>-0.042872</td>
<td>0.005470</td>
<td>-7.84</td>
</tr>
</tbody>
</table>

- NB: The frequency effect is stable, but the `meanSize` vs. `meanWeight` effect depends on what is residualized against what.
Residualization: Which predictor to residualize?

- What to residualize should be based on conceptual considerations (e.g. rate of disfluencies = number of disfluencies \sim number of words).

- **Be conservative** with regard to your hypothesis:
  - If the effect only holds under some choices about residualization, *the result is inconclusive*.
  - We usually want to show that a hypothesized effect holds *beyond what is already known* or that it *subsumes other effects*.

  → **Residualize** effect of interest.
    - E.g. if we hypothesize that a word’s predictability affects its duration beyond its frequency \Rightarrow residuals(lm(Predictability \sim Frequency, data)).
    - (if effect *direction* is not important, see also \sim model comparison)
Modeling schema

Generalized Linear Mixed Models
Florian Jaeger

Building an interpretable model
- Data exploration
- Transformation
- Coding
- Centering
- Interactions and modeling of non-linearities
- Collinearity
- What is collinearity?
- Detecting collinearity
- Dealing with collinearity

Model Evaluation
- Beware overfitting
- Detect overfitting: Validation
- Goodness-of-fit
- Aside: Model Comparison

Reporting the model
- Describing Predictors
- What to report
- Back-transforming coefficients
- Comparing effect sizes
- Visualizing effects
- Interpreting and reporting interactions

Discussion

Modeling schema diagram:
- Outcome
- Log transformation
- Transforming
- Coding
- Centering
- Interactions
- Predictors
- Residualization
- Reducing collinearity
- Model quality?
Overfitting: Fit might be too tight due to the exceeding number of parameters (coefficients). The maximal number of predictors that a model allows depends on their distribution and the distribution of the outcome.

Rules of thumb:

- **linear models**: $> 20$ observations per predictor.
- **logit models**: the less frequent outcome should be observed $> 10$ times more often than there predictors in the model.
- Predictors count: one per each random effect + residual, one per each fixed effect predictor + intercept, one per each interaction.
Validation allows us to detect **overfitting**:

- How much does our model depend on the exact data we have observed?
- Would we arrive at the same conclusion (model) if we had only slightly different data, e.g. a subset of our data?

**Bootstrap-validate** your model by repeatedly sampling from the population of speakers/items with replacement. Get estimates and confidence intervals for fixed effect coefficients to see how well they generalize (Baayen, 2008:283; cf. `bootcov()` for ordinary regression models).
Visualize validation

- Plot predicted vs. observed (averaged) outcome.
- E.g. for logit models, `plot.logistic.fit.fnc` in `languageR` or similar function (cf. [http://hlplab.wordpress.com](http://hlplab.wordpress.com))
- The following shows a badly fitted model:

```r
> lexdec$NativeEnglish = ifelse(lexdec$NativeLanguage == "English", 1, 0)
> lexdec$cFrequency = lexdec$Frequency - mean(lexdec$Frequency)
> lexdec$cNativeEnglish = lexdec$NativeEnglish - mean(lexdec$NativeEnglish)
> lexdec$Correct = ifelse(lexdec$Correct == "correct", T, F)
> l <- glmer(Correct ~ cNativeEnglish * cFrequency + Trial +
+ (1 | Word) + (1 | Subject),
+ data = lexdec, family="binomial")
```
Fitted values

So far, we’ve been worrying about coefficients, but the real model output are the **fitted values**.

**Goodness-of-fit** measures assess the relation between fitted (a.k.a. predicted) values and actually observed outcomes.

- **linear models**: Fitted values are predicted numerical outcomes.

<table>
<thead>
<tr>
<th>RT</th>
<th>fitted</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6.340359</td>
</tr>
<tr>
<td>2</td>
<td>6.308098</td>
</tr>
<tr>
<td>3</td>
<td>6.349139</td>
</tr>
<tr>
<td>4</td>
<td>6.186209</td>
</tr>
</tbody>
</table>

- **logit models**: Fitted values are predicted log-odds (and hence predicted probabilities) of outcome.

<table>
<thead>
<tr>
<th>Correct</th>
<th>fitted</th>
</tr>
</thead>
<tbody>
<tr>
<td>correct</td>
<td>0.9933675</td>
</tr>
<tr>
<td>correct</td>
<td>0.9926289</td>
</tr>
<tr>
<td>correct</td>
<td>0.9937420</td>
</tr>
<tr>
<td>correct</td>
<td>0.9929909</td>
</tr>
</tbody>
</table>
Goodness-of-fit measures: Linear Mixed Models

- $R^2 = \text{correlation(observed, fitted)}^2$.
  - Random effects usually account for much of the variance → obtain separate measures for partial contribution of fixed and random effects (Gelman & Hill 2007:474).
  - E.g. for

```r
> cor(l$RT, fitted(lmer(RT ~ cNativeEnglish * cFrequency + Trial +
+ (1 | Word) + (1 | Subject), data = l)))^2
```

- ...yields $R^2 = 0.52$ for model, but only 0.004 are due to fixed effects!
Measures built on data likelihood

- **Data likelihood**: What is the probability that we would observe the data we have given the model (i.e. given the predictors we chose and given the ‘best’ parameter estimates for those predictors).

- Standard model output usually includes such measures, e.g. in R:

<table>
<thead>
<tr>
<th>AIC</th>
<th>BIC</th>
<th>logLik</th>
<th>deviance</th>
<th>REMLdev</th>
</tr>
</thead>
<tbody>
<tr>
<td>-96.48</td>
<td>-63.41</td>
<td>55.24</td>
<td>-123.5</td>
<td>-110.5</td>
</tr>
</tbody>
</table>

- **log-likelihood**, \( \log(L) \). This is the maximized model’s log data likelihood, no correction for the number of parameters. **Larger (i.e. closer to zero) is better.** The value for log-likelihood should always be **negative**, and AIC, BIC etc. are positive. → current bug in the `lmer()` output for linear models.
Measures built on data likelihood (contd’)

- Other measures trade off goodness-of-fit (data likelihood) and model complexity (number of parameters; cf. Occam’s razor; see also model comparison).
  - **Deviance:** -2 times log-likelihood ratio. Smaller is better.
  - **Aikaike Information Criterion,** AIC = \( k - 2\ln(L) \), where \( k \) is the number of parameters in the model. Smaller is better.
  - **Bayesian Information Criterion,**
    \[ \text{BIC} = k \times \ln(n) - 2\ln(L) \], where \( k \) is the number of parameters in the model, and \( n \) is the number of observations. Smaller is better.
  - also **Deviance Information Criterion**
Likelihood functions used for the fitting of linear mixed models

- **Linear models:**
  - **Maximum Likelihood** function, ML: Find $\theta$-vector for your model parameters that maximizes the probability of your data given the model’s parameters and inputs. Great for point-wise estimates, but provides biased (anti-conservative) estimates for variances.
  - In practice, the estimates produced by ML and REML are nearly identical (Pinheiro and Bates, 2000:11).

→ hence the two deviance terms given in the standard model output in R.
Goodness-of-fit: Mixed Logit Models

- Best available right now:
  - some of the same measures based on data likelihood as for mixed models

<table>
<thead>
<tr>
<th>AIC</th>
<th>BIC</th>
<th>logLik</th>
<th>deviance</th>
</tr>
</thead>
<tbody>
<tr>
<td>499.1</td>
<td>537</td>
<td>-242.6</td>
<td>485.1</td>
</tr>
</tbody>
</table>

- but no known closed form solution to likelihood function of mixed logit models → current implementations use Penalized Quasi-Likelihoods or better Laplace Approximation of the likelihood (default in R; cf. Harding & Hausman, 2007)

- Discouraged:
  - pseudo-$R^2$ a la Nagelkerke (cf. along the lines of [http://www.ats.ucla.edu/stat/mult_pkg/faq/general/Psuedo_RSquareds.htm](http://www.ats.ucla.edu/stat/mult_pkg/faq/general/Psuedo_RSquareds.htm))

  - classification accuracy: If the predicted probability is $< 0.5$ → predicted outcome $= 0$; otherwise 1. Needs to be compared against baseline. (cf. Somer’s $D_{xy}$ and $C$ index of concordance).
Model comparison

- Models can be compared for performance using any goodness-of-fit measures. Generally, an advantage in one measure comes with advantages in others, as well.
- **To test whether one model is significantly better than another model:**
  - **likelihood ratio test** (for nested models only)
  - (DIC-based tests for non-nested models have also been proposed).
Likelihood ratio test for nested models

- -2 times ratio of likelihoods (or difference of log likelihoods) of nested model and super model.
- Distribution of likelihood ratio statistic follows asymptotically the $\chi^2$-square distribution with $DF(model_{super}) - DF(model_{nested})$ degrees of freedom.
- $\chi^2$-square test indicates whether sparing extra df’s is justified by the change in the log-likelihood.
  - in R: `anova(model1, model2)`
  - NB: use restricted maximum likelihood-fitted models to compare models that differ in random effects.
Example of model comparison

\[ \text{super.lmer} = \text{lmer}(\text{RT} \sim \text{rawFrequency} + (1 \mid \text{Subject}) + (1 \mid \text{Word}), \text{data} = \text{lexdec}) \]

\[ \text{nested.lmer} = \text{lmer}(\text{RT} \sim \text{rawFrequency} + (1 + \text{Trial}\mid \text{Subject}) + (1 \mid \text{Word}), \text{data} = \text{lexdec}) \]

\[ \text{anova} \left( \text{super.lmer}, \text{nested.lmer} \right) \]

<table>
<thead>
<tr>
<th>Df</th>
<th>AIC</th>
<th>BIC</th>
<th>logLik</th>
<th>Chisq</th>
<th>Chi Df</th>
<th>Pr(&gt;Chisq)</th>
</tr>
</thead>
<tbody>
<tr>
<td>super.lmer</td>
<td>5</td>
<td>-910.41</td>
<td>-883.34</td>
<td>460.20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nested.lmer</td>
<td>7</td>
<td>-940.71</td>
<td>-902.81</td>
<td>477.35</td>
<td>34.302</td>
<td>2</td>
</tr>
</tbody>
</table>

Change in log-likelihood justifies inclusion

Subject-specific slopes for Trial, and the correlation parameter between trial intercept and slope.
Model comparison: Trade-offs

- Compared to tests based on \( SE(\beta) \), model comparison ...
  - robust against collinearity
  - does not test directionality of effect

★ Suggestion: In cases of high collinearity ...
  - first determine which predictors are subsumed by others (model comparison, e.g. \( p > 0.7 \)) → remove them,
  - then use \( SE(\beta) \)-based tests (model output) to test effect direction on simple model (with reduced collinearity).
Reporting the model’s performance

- for the overall performance of the model, report goodness-of-fit measures:
  - for linear models: report $R^2$. Possibly, also the amount of variance explained by fixed effects over and beyond random effects, or predictors of interest over and beyond the rest of predictors.
  - for logistic models: report $D_{xy}$ or concordance C-number. Report the increase in classification accuracy over and beyond the baseline model.
- for model comparison: report the p-value of the log-likelihood ratio test.
Before you report the model coefficients

- **Transformations, centering**, (potentially standardizing), **coding**, **residualization** should be described as part of the predictor summary.
  - Where possible, give theoretical, and/or empirical arguments for any decision made.
  - Consider reporting scales for outputs, inputs and predictors (e.g., range, mean, sd, median).
Some considerations for good science

- **Do not** report effects that heavily depend on the choices you have made;
- **Do not** fish for effects. There should be a strong theoretical motivation for what variables to include and in what way.
- To the extent that different ways of entering a predictor are investigated (without a theoretical reason), **do** make sure your conclusions hold for all ways of entering the predictor or that the model you choose to report is superior (**model comparison**).
What to report about effects

- **Effect size** (What is that actually?)
- Effect direction
- Effect shape (tested by significance of non-linear components & superiority of transformed over un-transformed variants of the same input variable); plus visualization
Reporting the model coefficients

- **Linear models:** report (at least) coefficient estimates, *MCMC-based* confidence intervals (HPD intervals) and *MCMC-based* p-values for each fixed and random effect (cf. `pvals.fnc()` in `languageR`).

  ```
  $fixed
  Estimate  MCMCmean  HPD95lower  HPD95upper  pMCMC   Pr(>|t|)
  (Intercept)  6.3183 6.3180 6.2537 6.3833 0.0001 0.0000
  cFrequency  -0.0429 -0.0429 -0.0541 -0.0321 0.0001 0.0000
  NativeLanguageOther 0.1558 0.1557 0.0574 0.2538 0.0032 0.0101
  $random
  Groups Name Std.Dev. MCMCmedian MCMCmean HPD95lower HPD95upper
  1 Word (Intercept) 0.0542 0.0495 0.0497 0.0377 0.0614
  2 Subject (Intercept) 0.1359 0.1089 0.1101 0.0824 0.1386
  3 Residual 0.1727 0.1740 0.1741 0.1679 0.1802
  ```

- **Logit models:** for now, simply report the coefficient estimates given by the model output (but see e.g. Gelman & Hill 2006 for Bayesian approaches, more akin to the MCMC-sampling for linear models).
Interpretation of coefficients

<table>
<thead>
<tr>
<th>Fixed effects:</th>
<th>Estimate</th>
<th>Std. Error</th>
<th>t value</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>6.323783</td>
<td>0.037419</td>
<td>169.00</td>
<td></td>
</tr>
<tr>
<td>NativeLanguageOther</td>
<td>0.150114</td>
<td>0.056471</td>
<td>2.66</td>
<td></td>
</tr>
<tr>
<td>cFrequency</td>
<td>-0.039377</td>
<td>0.005552</td>
<td>-7.09</td>
<td></td>
</tr>
</tbody>
</table>

- The increase in 1 log unit of cFrequency comes with a -0.039 log units decrease of RT.
- Utterly **uninterpretable**!
- To get estimates in sensible units we need to back-transform both our predictors and our outcomes.
  - decentralize cFrequency, and
  - exponentially-transform logged Frequency and RT.
  - if necessary, we de-residualize and de-standardize predictors and outcomes.
Getting interpretable effects

- estimate the effect in ms across the frequency range and then the effect for a unit of frequency.

```r
> intercept = as.vector(fixef(lexdec.lmer4)[1])
> betafreq = as.vector(fixef(lexdec.lmer4)[3])
> eff = exp(intercept + betafreq * max(lexdec$Frequency)) - 
   exp(intercept + betafreq * min(lexdec$Frequency))
> eff/range * 100
```

- Report that the full effect of Frequency on RT is a 109 ms decrease.

- But in this model there is no simple relation between RTs and frequency, so resist to report that “the difference in 100 occurrences comes with a 4 ms decrease of RT”.

```r
> eff/range * 100
```

[1] -4.606494
The magic of the ‘original’ scale

★ What’s the advantage of having an effect size in familiar units?
  ▶ Comparability across experiments?
  ▶ Intuitive idea of ‘how much’ factor (and mechanisms that predicts it to matter) accounts for?

★ But this may be misleadingly intuitive . . .
  ▶ If variables are related in non-linear ways, then that’s how it is.
  ▶ If residualization is necessary then it’s applied for a good reason → back-translating will lead to misleading conclusions (there’s only so much we can conclude in the face of collinearity).
  ▶ Most theories don’t make precise predictions about effect sizes on ‘original’ scale anyway.
  ▶ Comparison across experiments/data sets often only legit if similar stimuli (with regard to values of predictors).
Comparing effect sizes

- It ain’t trivial: What is meant by effect size?
  - Change of outcome if ‘feature’ is present? → coefficient
    - per unit?
    - overall range?
  - But that does not capture how much an effect affects language processing:
    - What if the feature is rare in real language use (‘availability of feature’)? Could use . . .
      → Variance accounted for (goodness-of-fit\[C\] improvement associated with factor)
      → **Standardized coefficient** (gives direction of effect)

★ **Standardization**: subtract the mean and divide by two standard deviations.
  - standardized predictors are on the same scale as binary factors (cf. Gelman & Hill 2006).
  - makes all predictors (relatively) comparable.
Plotting coefficients of linear models

Plotting (partial) effects of predictors allows for comparison and reporting of their effect sizes:

- partial fixed effects can be plotted, using `plotLMER.fnc()`. Option `fun` is the back-transformation function for the outcome. Effects are plotted on the same scale, easy to compare their relative weight in the model.

- confidence intervals (obtained by MCMC-sampling of posterior distribution) can be added.
Plotting posterior distributions (for linear mixed models)

- `pvals.fnc()` plots MCMC-sampling posterior distributions, useful for inspection of whether the distributions are well-bounded.
Plotting coefficients of mixed logit models

- Log-odd units can be automatically transformed to probabilities.
  - **pros:** more familiar space
  - **cons:** effects are linear in log-odds space, but non-linear in probability space; linear slopes are hard to compare in probability space; non-linearities in log-odd space are hard to interpret

![Graphs showing the relationship between frequency and correctness](image_url)

- Frequency vs. Correctness
- Native Language vs. Correctness
- Family Size vs. Correctness

---

Generalized Linear Mixed Models

Florian Jaeger

Building an interpretable model

- Data exploration
- Transformation
- Coding
- Centering
- Interactions and modeling of non-linearities
- Collinearity
  - What is collinearity?
  - Detecting collinearity
  - Dealing with collinearity

Model Evaluation

- Beware overfitting
- Detect overfitting: Validation
- Goodness-of-fit
- Aside: Model Comparison

Reporting the model

- Describing Predictors
- What to report
- Back-transforming coefficients
- Comparing effect sizes
- Visualizing effects
- Interpreting and reporting interactions

Discussion
Plotting coefficients of mixed logit models (contd’)

▶ For an alternative way, see http://hlplab.wordpress.com/:

```R
> data(lexdec)
> lexdec$NativeEnglish = ifelse(lexdec$NativeLanguage == "English", 1, 0)
> lexdec$rawFrequency = exp(lexdec$Frequency)
> lexdec$cFrequency = lexdec$Frequency - mean(lexdec$Frequency)
> lexdec$cNativeEnglish = lexdec$NativeEnglish - mean(lexdec$NativeEnglish)
> lexdec$Correct = ifelse(lexdec$Correct == "correct", T, F)
> l<- lmer(Correct ~ cNativeEnglish + cFrequency + Trial +
>         (1 | Word) + (1 | Subject), data = lexdec, family="binomial")
> my.glmerplot(l, "cFrequency", predictor= lexdec$rawFrequency,
>             predictor.centered=T, predictor.transform=log,
>             name.outcome="correct answer", xlab= ex, fun=plogis)
```
Plotting coefficients of mixed logit models (contd’)

- Great for outlier detection. Plot of predictor in log-odds space (actual space in which model is fit):

![Plot of predictor in log-odds space](image)
Plotting interactions

> plotLMER.fnc(l, pred = "FamilySize", intr = list("cFrequency", > quantile(lexdec$cFrequency), "end"), fun = exp)

▶ Can also be plotted as the FamilySize effect for levels of cFrequency. Plotting and interpretation depends on research hypotheses.
Reporting interactions

- Report the p-value for the interaction as a whole, not just p-values for specific contrasts. For linear models, use `aovlmer.fnc()` in `languageR`.

```r
> aovlmer.fnc(lmer(RT ~ NativeLanguage + cFrequency * FamilySize +
> (1| Subject) + (1|Word), data = lexdec), mcmcM = mcmcSamp)
```

Analysis of Variance Table

<table>
<thead>
<tr>
<th></th>
<th>Df</th>
<th>Sum Sq</th>
<th>Mean Sq</th>
<th>F value</th>
<th>F Df2</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>NativeLanguage</td>
<td>1</td>
<td>0.20</td>
<td>0.20</td>
<td>6.5830</td>
<td>6.5830</td>
<td>1654.00 0.01</td>
</tr>
<tr>
<td>cFrequency</td>
<td>1</td>
<td>1.63</td>
<td>1.63</td>
<td>54.6488</td>
<td>54.6488</td>
<td>1654.00 2.278e-13</td>
</tr>
<tr>
<td>FamilySize</td>
<td>1</td>
<td>0.05</td>
<td>0.05</td>
<td>1.6995</td>
<td>1.6995</td>
<td>1654.00 0.19</td>
</tr>
<tr>
<td>cFrequency:FamilySize</td>
<td>1</td>
<td>0.03</td>
<td>0.03</td>
<td>1.0353</td>
<td>1.0353</td>
<td>1654.00 0.31</td>
</tr>
</tbody>
</table>

→ FamilySize and its interaction with cFrequency do not reach significance in the model.
Some thoughts for discussion

- What do we do when *what’s familiar* (probability space; original scales such as msecs; linear effects) is not *what’s best/better*?

- More flexibility and power to explore and understand complex dependencies in the data do not come for free, they require additional education that is not currently standard in our field.

  - Let’s distinguish challenges that relate to complexity of our hypothesis and data vs. issues with method (regression).
  - cf. What’s the best measure of effect sizes? What to do when there is collinearity? Unbiased vs. biased variance estimates for ML-fitted models; accuracy of laplace approximation.

- cf. What’s the best measure of effect sizes? What to do when there is collinearity? Unbiased vs. biased variance estimates for ML-fitted models; accuracy of laplace approximation.