Generalized Linear Mixed/Multilevel Models (GLMMs)

An applied tutorial

T. Florian Jaeger

October 29, 2020

Contents

1 Reading and assignments in *preparation* of this class **1**
1.1 Additional readings . . . 2

2 Quick recap: The generalized linear model (GLM) and generalized linear mixed model (GLMM) **2**
2.1 GLMs . . . 2
2.2 Assumptions of the GLM . . . 4
2.3 GLMMs . . . 4
2.4 Assumptions of the GLMM . . . 5
2.5 Writing up the results of a GLMM analysis . . . 5

3 The data for this document **6**
3.1 Background . . . 6
3.2 Goals . . . 6
3.3 Methods . . . 7
3.4 Overview . . . 7
3.5 Available data formats . . . 8

4 Working through an example **8**
4.1 GLM (logistic regression) . . . 8
4.1.1 Prepare for class . . . 9
4.2 GLMM (mixed-effects logistic regression) . . . 9
4.2.1 Prepare for class . . . 10

5 Using the GLMM to obtain subject-specific estimate **11**
5.1 Why use GLMMs to obtain subject-specific estimate? The benefits of partial pooling . . . 11
5.2 An example . . . 11
5.2.1 Prepare for class . . . 12
5.2.2 Discussion questions for class . . . 12

6 Issues during GLMM fitting **12**
6.1 Convergence failures . . . 13
6.2 Singular fits . . . 13

7 Session info **13**

1 Reading and assignments in *preparation* of this class

Please make sure you have read Gelman & Hill (2007, p. 301-310 in Ch 14) as well as McElreath (2019, Ch. 10.2). The latter is particularly concise and contains some beautifully clear graphs on a) the relation between different

types of GLMs and b) the consequences of the link function—relating the 'original' scale (e.g., counts for the Poisson model) to the scale on which the linear predictor is fit (log-counts for the Poisson model).

Then read and *work through* this document. We will use class to go through the important concepts and to address any questions that you have about the readings or the problem sets in this document. **Please note that this document is providing R code only.** Some of the steps described here might have less direct solutions in Matlab, so it is recommended that you start early. I have, however, tried to describe each step in a way that does not depend on R or Matlab.

In this document, you will find sections labeled "Prepare for class". Please work through those examples and write up your answers. If there are a few questions, you cannot answer, elicit help on the slack channel. In addition, there are sections called "Discussion questions for class". Please think about these questions, but don't worry if you get stuck. These are some questions we can go through during class.

1.1 Additional readings

For logistic regression another *optional* reading is James et al. (2013, Ch. 4 up to but *not* including 4.3.4). To learn more about psychometric models, you might also find Gilchrist et al. (2005) and Wichmann and Hill (2001a, b) helpful (both in *Perception & Psychophysics*).

2 Quick recap: The generalized linear model (GLM) and generalized linear mixed model (GLMM)

A generalized linear model extends the linear model (LM) to outcome variables drawn from—or *assumed* to be drawn from—the exponential family of distributions. Concise introductions are provided in Gelman & Hill (2007, Ch. 6) or MacElreath (2019, Ch. 10.2).

2.1 GLMs

The GLM has three components: the linear predictor (η), the link function (g), and the outcome distribution (f), as shown in Figure 1a. Building on our notation for the LM:

$$\eta = X\beta \tag{1}$$
$$\mu = g^{-1}(\eta) \tag{2}$$
$$y \sim f(\mu) \tag{3}$$

As for the LM, X is is called the *model (or design) matrix* and β is a vector of parameter values. Note that η and μ are fully determined by X and β (i.e., they do not introduce degrees of freedom or latent structure). The LM is a GLM for which g is the identity function I, and f is the Normal distribution with unknown residual variance σ_{resid}:

$$\eta = X\beta \tag{4}$$
$$\mu = I^{-1}(\eta) = \eta = X\beta \tag{5}$$
$$y \sim \mathcal{N}(\mu, \sigma_{resid}) \quad \Rightarrow y \sim \mathcal{N}(X\beta, \sigma_{resid}) \tag{6}$$

The exponential family of distributions—not to be confused with the exponential distribution, which is a member of the exponential family of distributions—contains such well-known members as the Normal, Bernoulli, categorical (the Bernoulli-equivalent for categorical outcomes with more than 2 levels), Poisson, Dirichlet, gamma, and inverse Wishart distributions. This provides plausible distributions for binary data (e.g., correct/incorrect), count data, ordered and unorderer categorical data, variance or sum of square data, etc. A few types of GLMs are shown in Figure 2.

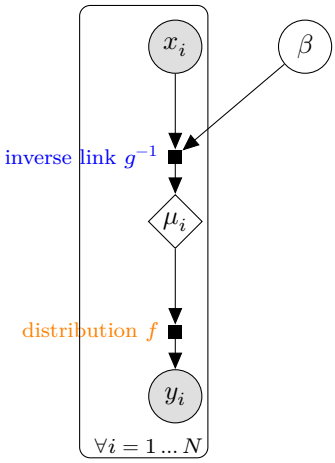
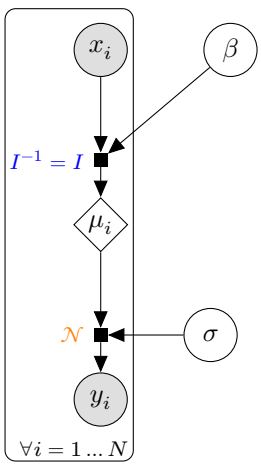
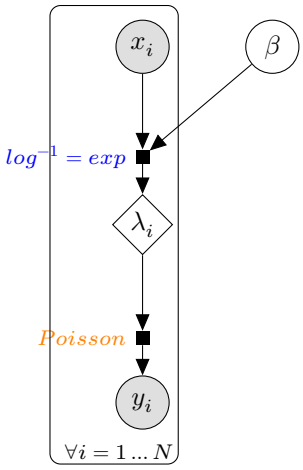


Figure 1: Generalized linear model (GLM)

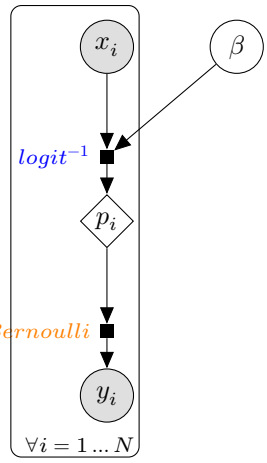
Figure 2: The three components of the GLM: the linear predictor ($X\beta$), the link function g, and the distribution f.



(a) Linear model



(b) Poisson regression



(c) Logistic regression

Figure 3: Three different GLMs with their canonical link functions, and using typical notation (e.g., λ instead of μ for the expected value of the Poisson model).

2.2 Assumptions of the GLM

The GLM adjusts several of the assumptions of the LM. Foremost of all, non-normal GLMs do *not* assume that the outcome follows a normal distribution around the mean. From that it also follows that that these GLMs do *not* assume normally distributed residuals or equality of variance. In fact, many types of GLMs assume that the variance depends on the mean, and is thus systematically non-homogeneous. This is the case, for example, for the Poisson and logistic regression. If analyzing data that follow the Poisson or Bernoulli distribution, this removes the primary problem of LMs—the assumption of homogeneity of variances can lead to highly misleading results for such data (see Jaeger, 2008, for illustration). The GLM provides a way around this issue. It should be noted, however, that each specific GLM assumes a specific distribution of the data. As always, assumptions should thus be checked and, if violated, the consequences of those violations should be investigated.

The GLM inherits from the LM the assumptions of:

- Independence of observations/errors
- Validity/exhaustivity

We are also still assuming:

- Additivity of effects
- Linearity of each effects

But do so on the scale described by the link function g.

2.3 GLMMs

The GLMM extends the GLM in *exactly* the same way as the linear mixed model (LMM) extends the LM: by allowing the βs to vary between levels of one or more grouping factors, while assuming that the differences between levels follow a (multivariate) Normal distribution. Grouping variables are is sometimes called *random effects*. The name grouping variable highlights the fact that the data are grouped by this variable. For example, when we have many observations from each subject (repeated measures data), the data can be thought of as grouped by subjects.

For a GLMM with one grouping variable, this can written as follows (see Figure 3 for a side-by-side comparison of the GLM and GLMM):

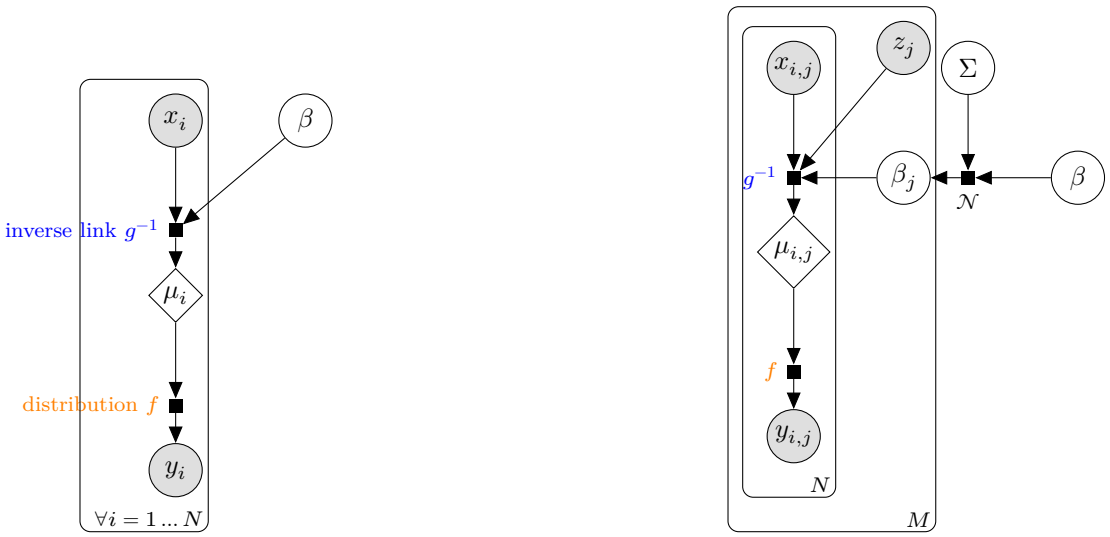
$$\eta = X\beta \tag{7}$$
$$\mu = g^{-1}(\eta) \tag{8}$$
$$y \sim f(\mu) \tag{9}$$
$$\tag{10}$$

$$\beta = \beta_{population} + \beta_{group\ level}, \tag{11}$$
$$\beta_{group\ level} \sim \mathcal{N}(0, \Sigma) \tag{12}$$

GLMMs can have more than one grouping variable. For example, for two crossed random effects:

$$\beta = \beta_{population} + \beta_{group1\ level} + \beta_{group2\ level}, \tag{13}$$
$$\beta_{group1\ level} \sim \mathcal{N}(0, \Sigma_{group1}) \tag{14}$$
$$\beta_{group2\ level} \sim \mathcal{N}(0, \Sigma_{group2}) \tag{15}$$
$$\mathcal{N}(0, \Sigma_{group1}) \perp \mathcal{N}(0, \Sigma_{group2}) \tag{16}$$

The grouping variables can also be hierarchically organized (e.g., a national sample of students taking a college entrance exam like the SAT could be hierarchically grouped by school, county, and state). In this case, we refer to the model as a *multilevel* model (which conveniently also abbreviates to an "M").



(a) Generalized linear model (GLM)

(b) Generalized linear mixed model (GLMM)

Figure 4: The GLM and GLMM side by side. For the GLMM, z_j is the jth level of the grouping variable z, which select which β_j is to be chosen for the present case (i.e., j here corresponds to *group level* in equation 11.

2.4 Assumptions of the GLMM

The GLMM softens the independence assumption of the GLM, instead assuming *conditional independence* (conditional on the assumed random effects). This is achieved via random effects that are distributed to follow normal distributions with unknown standard deviations, centered around zero. For each grouping factor, one can further model the correlations between the random effects, in which case the by-group adjustments $\beta_{group\ level}$ to the fixed effects ($\beta_{population}$) in equation 11 are assumed to follow a multivariate normal distribution. Put differently, we assume that the β vector in equation 7 as being drawn from a multivariate normal distribution with mean $\beta_{population}$ and covariance matrix Σ_{group}.

For multiple random effects, we further assume that the multivariate Normal distributions for the different groups are independent of each (that's what the $\perp$ means in equation 16). All other assumptions are inherited from the GLM.

2.5 Writing up the results of a GLMM analysis

The write-up of a GL(M)M analysis will in many ways follow the write up of an L(M)M analysis, with a few additions. In addition to the outcome, predictors, and how we coded them, we need to clearly state what *type* of GL(M)M we're fitting. Second, since our predictors now include random effects, we need to describe those predictors, too. That is, we need to describe the *random effect structure* of the model. Like with everything else, we should also *motivate* that structure unless it is abundantly clear why this is the structure we chose.

On a terminological note, it can become somewhat confusing to talk about random effects "for" a grouping factor (e.g., subjects) since we also need to talk about random slopes "for" a fixed effect predictor. I strongly recommend the use of "by" when talking about grouping factors, and "for" when talking about slopes. For example:

We analyzed the trial-level data with mixed-effects logistic regression (for introduction, see Jaeger, 2008), using the function `glmer` from the `lmer` package (citation with version) of the software `R` (citation with version). We predict correct responses (1 = correct vs. 0 = incorrect) from letter size (centered), condition (deviation-coded: .5 = *crowded* vs. -.5 = *uncrowded*) and their interaction. Following standards of the field, we used the maximal random effect structure required by the design, random by-subject intercepts and slopes for condition, size, and their interaction.

Another difference in reporting GL(M)Ms is that the linear predictor is often on a scale that is not particularly intuitive. It is important to remember that *there is a good reason why this is the case*: many of the outcomes for

which GL(M)Ms offer a better solution than LMs do not lend themselves to intuitve linear analyses on the original outcome scale (e.g., because that scale is bounded and/or reflects the result of multiple non-additive processes). I.e., the unintuitive scale on which the effects of the GLMM are expressed are a direct (inevitable) consequence of our efforts to avoid the Type I and II error inflation that would result from analyzing such data with an LM.

Unfortunately, the use of LM, ANOVA, and *t*-tests for, e.g., count and proportion data remains common practice, although we have known for more than 70 years that this practice is problematic (for summaries directed at cognitive scientists, see Dixon, 2008; Jaeger, 2008; Johnson, 2009).

For simple factorial designs, the unintuitive scales on which the effects of the GL(M)M are expressed typically aren't much of a problem since we should accompany our analysis with plots and/or tables that summarize the relevant means anyway. Those figures/tables tend to provide an intuitive summary of the data. However, if we do need to somehow translate our effects into more intuitive scales, there are a number of ways to do so. See e.g., the "divide by four rule" for mean proportions close to .5, described in Gelman and Hill, 2007. Or take the fact that we can describe additive effects on log counts in terms of multiplicative effects on counts (similarly, we can describe effects additive effects on log-odds in terms of multiplicative effects on odds). If push comes to shove, we can always describe what the predicted outcome is at meaningful values of our predictor (e.g., the 5th and 95th quantile of a continous predictor), and those point predictions can be translated to the scale of the original outcome: simply plug those predictor values into the linear predictor, apply the inverse link function, and you have the predicted mean.

3 The data for this document

For this tutorial we are continuing to use Ashley Clark's data from her experiment on visual crowding effects on foveal processing. **Do not share this data with Ashley's permission. Thank you.**

3.1 Background

Crowding is a visual phenomenon that has puzzled scientists for decades; an object in isolation can be perceived without problems, yet surrounding it with similar objects makes it harder to see. While crowding has been studied extensively in the visual periphery, humans normally orient objects in their center of gaze, the high-acuity region of the retina called the foveola. While the foveola is less than 1.5mm wide (or ~1 visual deg2), it contains more cones than rest of the retina combined. Human's ability to actively perceive the visual world relies heavily on not only the foveola itself, but also how and where the eye is positioned. The few studies that have examined crowding within foveal vision have produced contradictory results. Some reasons for these discrepancies include using relatively large stimuli, a small number of participants, abnormal stimuli presentation, and having indefinite stimulus presentation times with no eye tracking. More recent research, however, has highlighted the importance of precise eye tracking due to the eye's constant movement during even fixation. These small and constant movements of the eye, called fixational eye movements (FEMs), are beneficial or both high acuity vision, as well as daily tasks such as reading and face recognition. FEM's have never been examined in the context of crowding, and it remains unknown how individuals in previous crowding studies directed their foveola over stimuli, or even maintained fixation.

3.2 Goals

The goals of this research are to (1) investigate the effect of crowding within the foveola, and (2) examine if and how fixational eye movements influence crowding at this scale. Based on previous research, we hypothesize that crowding will be detrimental to foveal vision, as it is in peripheral vision, but on a finer scale. Further, based on the recent findings that FEMs are beneficial for high-acuity vision, I expect a relationship between FEMs and the strength of crowding within the foveola, with larger and less precise FEMs increasing the negative effects visual crowding.

3.3 Methods

Studying FEMs during visual crowding within the foveola requires high-precision eye tracking and accuracy in localizing the center of gaze. Current video eye trackers do not have the required spatial precision, as the error of gaze localization is as large as the foveola itself. However, by using a custom built state-of-the-art eye tracking system with arcminute precision, we will be able to examine exactly how FEMs contribute to crowding within the foveola. Stimuli consist of a number-font designed specifically for studying crowding in the fovea, as it allows for recognition even when numbers are closer together than traditional optotypes used in other crowding studies.6 Two conditions will be examined, the uncrowded (where a single number is presented), and the crowded (where the same size number is presented, but with four surrounding numbers). The size of the number and spacing between the numbers in the crowded condition change throughout the experiment based on the subject's performance using an adaptive procedure. The stimuli presented will vary in size, ranging from 0.5 arcminutes to 4 arcminutes in width. To determine the number-width threshold, we use a standard psychophysics procedure measuring the width of the stimulus at which a subject performs above chance level.

3.4 Overview

Subject-level variables include:

- Subject
- Condition: "crowded" or "uncrowded"
- Threshold: threshold inferred from a previous Weibull model fit to the subjects trial-level data
- DiffusionConstant
- Span
- Area
- Curvature
- Speed

Size-level variables include:

- Size: the width (or strokewidth) of the stimulus during that group of trials.
- Performance: What the overall performance for the size stimulus was.

Trial-level variables:

- Response: What number the subject guessed.
- Answer: What the number presented on the screen was.
- Correct: Whether the response was correct or not (combining information from Response and Answer)
- Response Time: response time of subject. Note* subjects were not told to response as quickly as possible during experiment.
- Traces: the x and y traces of each trial in this size stimulus group.
- Trial Curvature: Same as curvature above, but for each individual trial.
- Trial Speed: Same as speed above, but for each individual trial.

For the present purpose, I'm renaming "Crowded" as "Condition", "Answer" as "ResponseExpected", "Correct" to "ResponseCorrect". I'll also prefix all subject-level variables with "Subject" and remove the "Trial" prefix that is used for some (but not all) of the trial-level variables. Finally, since performance refers to the average performance for a specific letter size, I change it to "Size.AvgPerformance". I then order the column from the least quickly varying (across rows) to the most quickly varying.

```
tibble [6,539 x 17] (S3: tbl_df/tbl/data.frame)
 $ Subject               : Factor w/ 10 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ Condition             : Factor w/ 2 levels "uncrowded","crowded": 1 1 1 1 1 1 1 1 1 1 ...
 $ Threshold.Subj        : num [1:6539] 1.39 1.39 1.39 1.39 1.39 ...
 $ DiffusionConstant.Subj: num [1:6539] 11.1 11.1 11.1 11.1 11.1 ...
 $ Area.Subj             : num [1:6539] 137 137 137 137 137 ...
 $ Span.Subj             : num [1:6539] 3.13 3.13 3.13 3.13 3.13 ...
 $ Speed.Subj            : num [1:6539] 40.6 40.6 40.6 40.6 40.6 ...
 $ Curvature.Subj        : num [1:6539] 12 12 12 12 12 ...
 $ Size                  : num [1:6539] 0.505 0.505 0.505 0.505 0.505 ...
 $ Size.AvgPerformance   : num [1:6539] 0.135 0.135 0.135 0.135 0.135 ...
 $ Response              : num [1:6539] 1 4 1 4 3 1 2 3 3 4 ...
```

```
 $ ResponseExpected      : num [1:6539] 1 4 4 2 2 4 4 1 2 1 ...
 $ ResponseCorrect       : num [1:6539] 1 1 0 0 0 0 0 0 0 0 ...
 $ ResponseTime          : num [1:6539] 2032 1537 2203 2266 2385 ...
 $ Curvature             : num [1:6539] 15.5 16.1 14.6 14.1 14.1 ...
 $ Speed                 : num [1:6539] NaN 33.6 38.5 43.5 39.8 ...
 $ Span                  : num [1:6539] 2.55 3.03 2.25 2.9 2.13 ...
```

3.5 Available data formats

The git repository contains the data in three formats: a .mat file, and .Rdata file with a single tibble/data.frame object (d), and a .csv file that contains all information except the X and Y traces.

4 Working through an example

4.1 GLM (logistic regression)

We first deviation-code the crowdedness condition, and center letter size:

```
levels(d$Condition)
```

```
[1] "uncrowded" "crowded"
```

```
contrasts(d$Condition) = cbind("Crowded.vs.Uncrowded" = c(-.5, .5))

d %<>%
  mutate(Size = Size - mean(Size))
```

Now we analyze the data from one subject (Subject 4) in an ordinary (non-mixed) logististic regression:

```
l = glm(
  ResponseCorrect ~ 1 + Size + Condition + Size:Condition,
  data = d,
  family = binomial)

summary(l)
```

```
Call:
glm(formula = ResponseCorrect ~ 1 + Size + Condition + Size:Condition,
    family = binomial, data = d)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.9824  -1.1972   0.7427   0.9919   1.6762

Coefficients:
                                  Estimate Std. Error z value Pr(>|z|)
(Intercept)                        0.67989    0.03212  21.169  < 2e-16 ***
Size                               1.06677    0.04961  21.505  < 2e-16 ***
ConditionCrowded.vs.Uncrowded     -0.65333    0.06424 -10.171  < 2e-16 ***
Size:ConditionCrowded.vs.Uncrowded -0.61150    0.09921  -6.164 7.11e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 8641.5  on 6538  degrees of freedom
Residual deviance: 8039.5  on 6535  degrees of freedom
AIC: 8047.5

Number of Fisher Scoring iterations: 4
```

4.1.1 Prepare for class

1. In your own words, describe in 1-2 sentences the effect of size (for this subject). Is it significant? Is it positive or negative? What does that mean intuitively (describe the effect on the outcome).
2. Do the same for the effect of condition. Use the terms crowded and uncrowded. What is the predicted difference in log-odds of an accurate answer between the two conditions (for this subject).
3. What's the conceptual interpretation of the interaction for this subject? Try to describe it in your own words.
4. For the uncrowded condition, how would you go from this to calculate the required letter size to achieve 62.5% threshold accuracy?

4.2 GLMM (mixed-effects logistic regression)

The we fit the mixed-effects logistic regression with the full random effect structure required by the design:

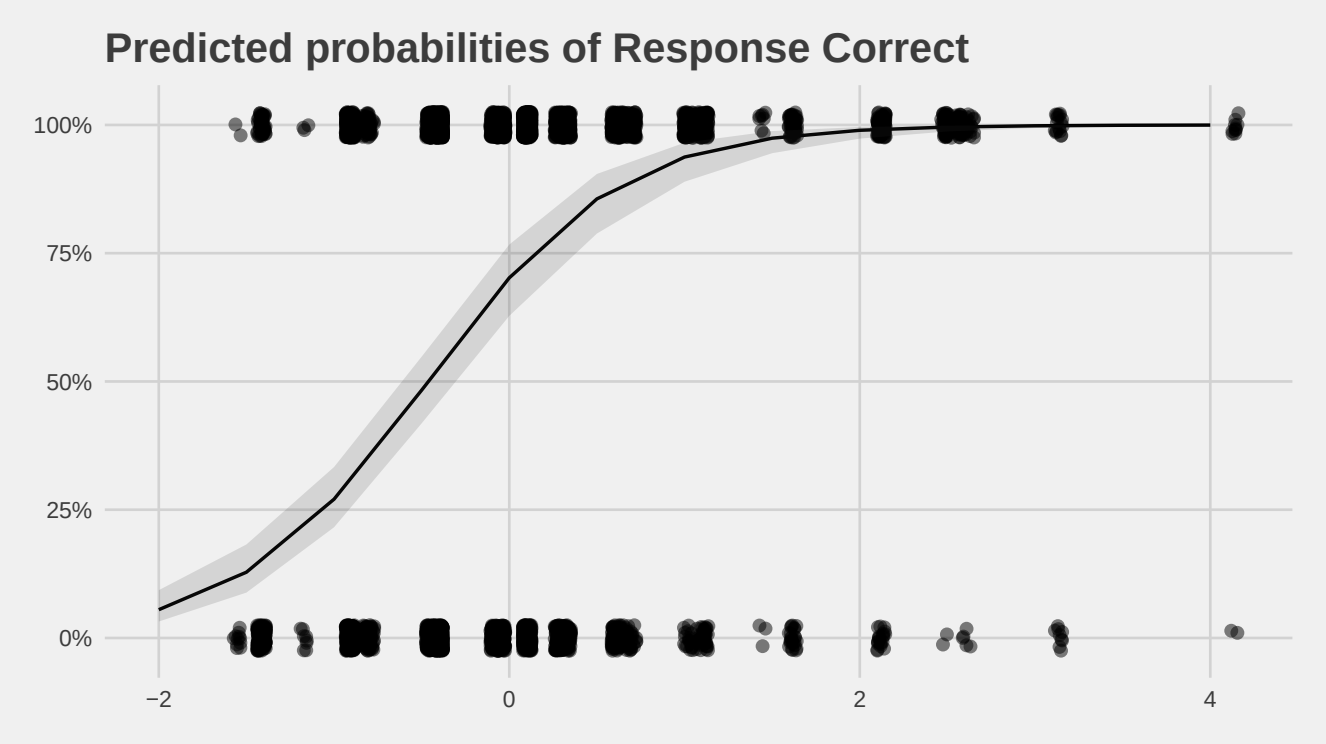
```
m = glmer(
  ResponseCorrect ~ 1 + Size + Condition + Size:Condition + (1 + Size + Condition + Size:Condition | Subject),
  data = d,
  family = binomial)
```

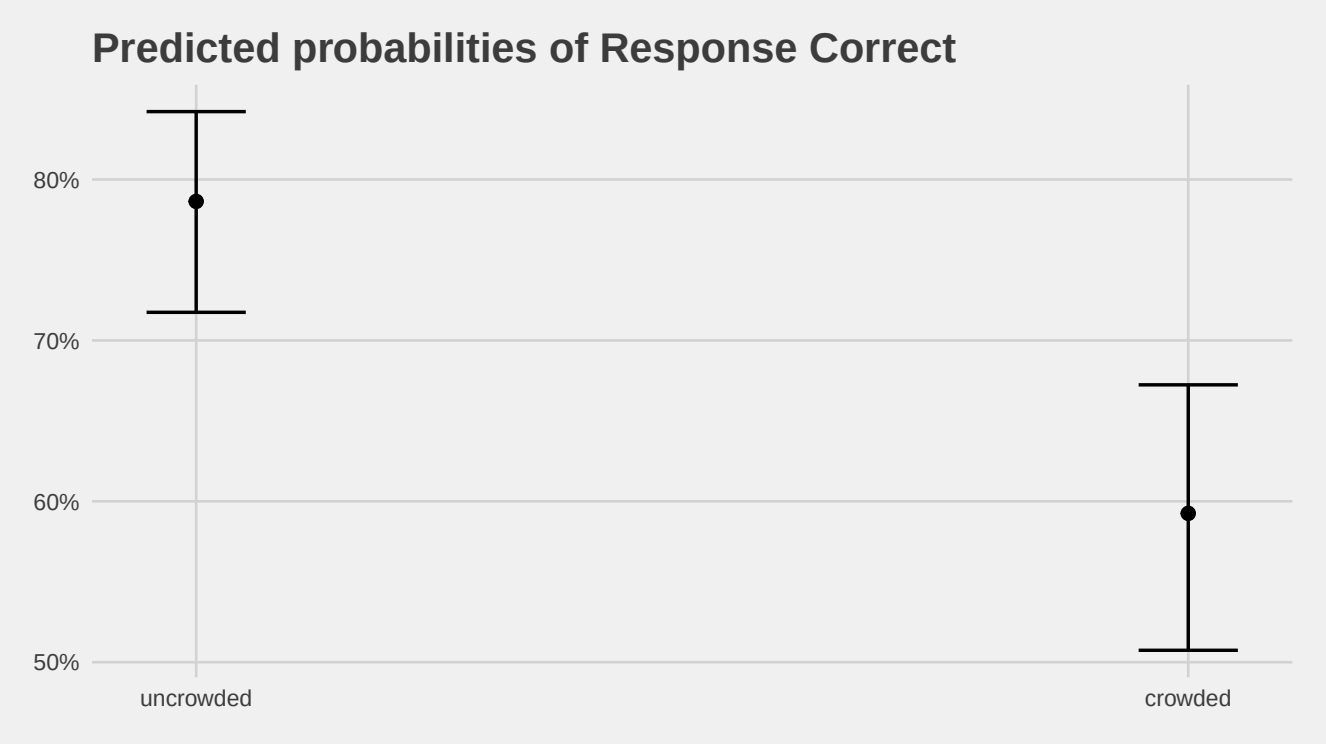
```
boundary (singular) fit: see ?isSingular
```

```
summary(m)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']
 Family: binomial  ( logit )
Formula: ResponseCorrect ~ 1 + Size + Condition + Size:Condition + (1 +      Size + Condition + Size:Condition | Subject)
   Data: d

     AIC      BIC   logLik deviance df.resid
  7816.3   7911.3  -3894.2   7788.3     6525

Scaled residuals:
     Min       1Q   Median       3Q      Max
-14.2768  -0.9618   0.4980   0.7375   2.5184

Random effects:
 Groups  Name                              Variance Std.Dev. Corr
 Subject (Intercept)                       0.27094  0.5205
         Size                              0.26705  0.5168    0.66
         ConditionCrowded.vs.Uncrowded     0.10770  0.3282   -0.08 -0.14
         Size:ConditionCrowded.vs.Uncrowded 0.07958  0.2821   -0.48  0.11  0.63
Number of obs: 6539, groups:  Subject, 10

Fixed effects:
                                  Estimate Std. Error z value Pr(>|z|)
(Intercept)                         0.8388     0.1701   4.933 8.12e-07 ***
Size                                1.8470     0.1818  10.160  < 2e-16 ***
ConditionCrowded.vs.Uncrowded      -0.9290     0.1342  -6.923 4.43e-12 ***
Size:ConditionCrowded.vs.Uncrowded -0.1015     0.1783  -0.569    0.569
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr) Size   CnC..U
Size         0.618
CndtnCrw..U -0.109 -0.227
Sz:CndtC..U -0.331  0.050  0.463
convergence code: 0
boundary (singular) fit: see ?isSingular
```

We cover convergence problems and warnings of possible singular fits below. For now, let's ignore this warning and plot the model's predictions:





4.2.1 Prepare for class

1. Drawing on what you've learned about LMs, GLMs, and GLMMs, write a result section for the above model, and submit it on Slack prior to the class. (The model is exactly the one described in the method write-up at the end of Section 2).
2. How could you assess whether the effect of condition is fully mediated (subsumed) by effect of crowded on the

speed of eye movements, which then causes the effect on accuracy? I.e., how would you test the hypothesis that the causal model for your data is crowdedness of display → speed of eye movements → accuracy (indirect causation), rather than crowdedness of display → accuracy (direct causation)? Describe what you'd want to do in your own words, or try it out (in R, for example, you can compare glmer fits with the anova() function in just the same way that you can compare lm() or glm() fits).

5 Using the GLMM to obtain subject-specific estimate

As it is a GL*M*M, we can also obtain estimates of the individual differences in the effects (at least for any predictor for which we were able to fit random slopes by subject). These estimates can be used to obtain subject-specific psychometric thresholds under the *partial pooling* assumption of mixed models.

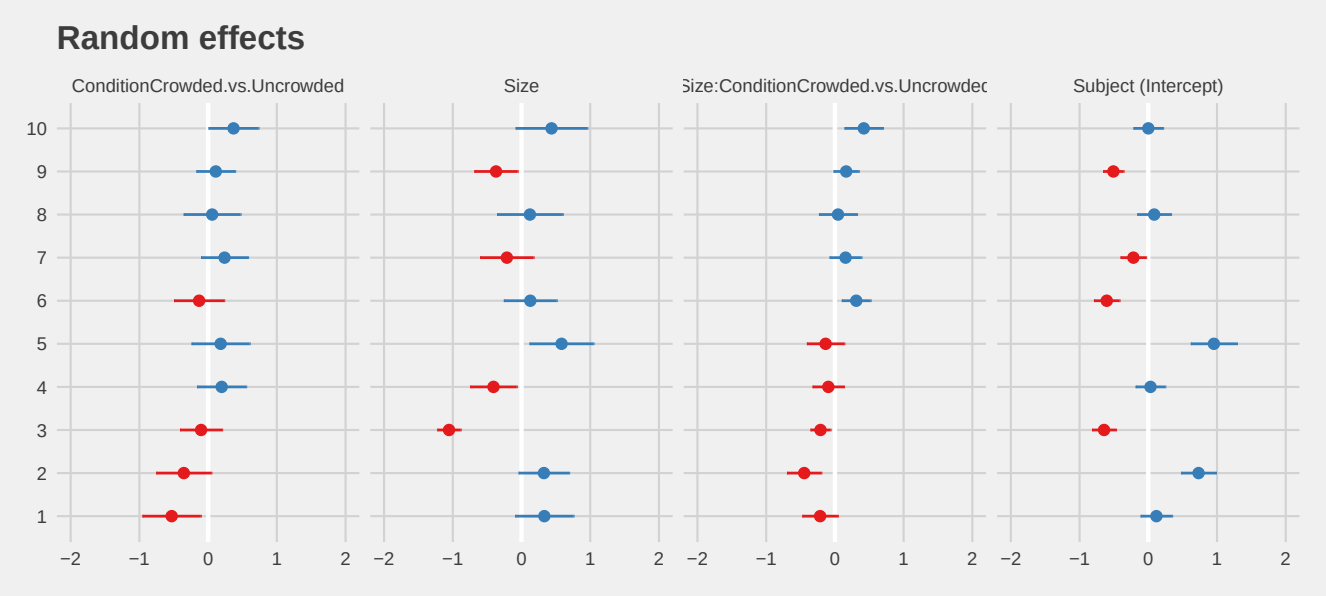
5.1 Why use GLMMs to obtain subject-specific estimate? The benefits of partial pooling

You might wonder **why we would want to use a GLMM to estimate subject-specific thresholds, rather than individual GLMs fit separately to each subject's data.** There is actually something rather appealing about the GLMM approach: it takes into account *all* data while estimating each subject-specific parameters. Essentially, **the assumption of normally distributed individual differences acts as a *regularizing prior* that 'shrinks' the subject-specific estimates towards the population mean, thereby reducing the probability of overfitting the model(s) to the data.** Such shrinkage based on the overall mean is generally a good feature for prediction. It essentially trades off fit for the sample (reducing it) against likely fit against as of yet unseen data (increasing predictive accuracy). McElreath (2019) summarizes this very nicely for GLMMs (e.g., on p. 419). Shrinkage is also discussed in length in Gelman & Hill (2007, Section 12.3). I encourage you to read those sections.

The benefit of shrinkage to the population mean is particularly noteworthy when we have little data per subject but a lot of subjects. It does not come for free, however: if the assumption of normally distributed individual differences is wrong it can create bias. For the type of data we tend to analyze in the psychological sciences the normality assumption about the individual differences is usually sufficiently close to the truth (at least for our typical population of college students; this might differ if one works with highly heterogeneous patient populations).

5.2 An example

The following plot shows the posterior distribution of by-subject differences in each of the four coefficients of the model fit above:



As is evidence from these plots, the GLMM returns a *distribution* over possible subject-specific adjustment in the intercept and effects. As a *point estimate* of the subject-specific adjustment, we typically use the so-called best unbiased linear predictors (BLUPs)—the posterior *mode* of the random effects for each subjects (the value with the highest estimated density. Whereas the points in the above plot reflect the mean of the by-subject effect estimated for each subject, the BLUPs are the mode of that distribution. Typically, these two values will be closely related. Take, for example, the BLUP for subject 2's intercept (.7317…) and compare it to the plot (second to last row, last column).

```
$Subject
     (Intercept)       Size ConditionCrowded.vs.Uncrowded Size:ConditionCrowded.vs.Uncrowded
1   0.1182922778  0.3322312                  -0.53041250                        -0.21492599
2   0.7317493573  0.3260495                  -0.35344773                        -0.44648010
3  -0.6435440297 -1.0546417                  -0.10229036                        -0.20925712
4   0.0323276366 -0.4076431                   0.19711393                        -0.09537583
5   0.9548535156  0.5828423                   0.18242857                        -0.13492389
6  -0.6042431174  0.1282141                  -0.13120895                         0.31068822
7  -0.2167805860 -0.2129506                   0.23921633                         0.15542642
8   0.0860655408  0.1226124                   0.05852554                         0.04531094
9  -0.5078199618 -0.3705270                   0.11171866                         0.16430669
10  0.0006444035  0.4369235                   0.37012741                         0.42049974

with conditional variances for "Subject"
```

5.2.1 Prepare for class

1. What intercept does this model predict for Subject 9?
2. Is that intercept smaller or larger than the predicted intercept for subject 8? Is it smaller or larger than the predicted population intercept?
3. What intercept does the model predict for a novel Subject that was not part of the data we fit the model to?

5.2.2 Discussion questions for class

4. Calculate the threshold size for Subject 2 in the uncrowded condition?
5. How much does this threshold size differ from that in the crowded condition?

6 Issues during GLMM fitting

6.1 Convergence failures

In particular for full random effect structures, GLMM fitting might not converge, as indicated by the error message. This will trigger a convergence error in R. **You cannot rely on the output of models that have not converged. Do not ignore this error message.**

When a model fails to converge, we can try to increase the number of iterations that the fitting algorithm uses to find the best-fitting parameters. For example, in R we might invest additional compute time to achieve more accurate computation, and refit the model.

If that does not work, we can simplify the random effect structure by removing the most complex terms. We start by removing the random correlations. In R, we can use the shorthand "||" instead of "|" to indicate that we are willing to assume that the correlations are between the random variance are zero. If that still does not yield convergence, we remove the random by-subject slopes for the interaction, etc. I.e., we reduce the model step by step:

$$ResponseCorrect\ 1 + Size * Condition + (1 + Size * Condition|Subject) \quad (17)$$

$$ResponseCorrect\ 1 + Size * Condition + (1 + Size * Condition||Subject) \quad (18)$$

$$ResponseCorrect\ 1 + Size * Condition + (1 + Size + Condition||Subject) \quad (19)$$

$$etc. \quad (20)$$

6.2 Singular fits

Even when a model converges, it might reflect a singular fit (R will print a *warning* if that is the case). Essentially, this means that one of the variances of the random effects might be zero, or that one of the correlations between the random effects might be -1 or 1. From the R help file on `isSingular`:

> Complex mixed-effect models (i.e., those with a large number of variance-covariance parameters) frequently result in singular fits, i.e. estimated variance-covariance matrices with less than full rank. Less technically, this means that some "dimensions" of the variance-covariance matrix have been estimated as exactly zero. For scalar random effects such as intercept-only models, or 2-dimensional random effects such as intercept+slope models, singularity is relatively easy to detect because it leads to random-effect variance estimates of (nearly) zero, or estimates of correlations that are (almost) exactly -1 or 1. However, for more complex models (variance-covariance matrices of dimension >=3) singularity can be hard to detect; models can often be singular without any of their individual variances being close to zero or correlations being close to +/-1. ...
>
> While singular models are statistically well defined (it is theoretically sensible for the true maximum likelihood estimate to correspond to a singular fit), there are real concerns that (1) singular fits correspond to overfitted models that may have poor power; (2) chances of numerical problems and mis-convergence are higher for singular models (e.g. it may be computationally difficult to compute profile confidence intervals for such models); (3) standard inferential procedures such as Wald statistics and likelihood ratio tests may be inappropriate.

While singular fits are not technically an error, we might want to check whether the conclusions we would draw from this analysis would hold under a slightly simpler model that avoids the singularity. This would be achieved by following the same stepwise simpiflication of the random effect structure used above when the model failed to converge. Alternatively, and perhaps to be preferred, we could employ a Bayesian approach, in which case weakly regularizing priors are often sufficient to help to model converge.

7 Session info

```
devtools::session_info()
```

```
- Session info ---------------------------------------------------------------------------------------------------------------------
 setting  value
```

```
 version  R version 4.0.2 (2020-06-22)
 os       macOS High Sierra 10.13.6
 system   x86_64, darwin17.0
 ui       X11
 language (EN)
 collate  en_US.UTF-8
 ctype    en_US.UTF-8
 tz       America/New_York
 date     2020-10-29

- Packages ---------------------------------------------------------------------------------------------------------------------------------
 package      * version    date       lib source
 assertthat     0.2.1      2019-03-21 [1] CRAN (R 4.0.2)
 backports      1.1.9      2020-08-24 [1] CRAN (R 4.0.2)
 bayestestR     0.7.2      2020-07-20 [1] CRAN (R 4.0.2)
 blob           1.2.1      2020-01-20 [1] CRAN (R 4.0.2)
 boot           1.3-25     2020-04-26 [1] CRAN (R 4.0.2)
 broom        * 0.7.0      2020-07-09 [1] CRAN (R 4.0.2)
 callr          3.4.4      2020-09-07 [1] CRAN (R 4.0.2)
 carData        3.0-4      2020-05-22 [1] CRAN (R 4.0.2)
 cellranger     1.1.0      2016-07-27 [1] CRAN (R 4.0.2)
 cli            2.0.2      2020-02-28 [1] CRAN (R 4.0.2)
 coda           0.19-3     2019-07-05 [1] CRAN (R 4.0.2)
 codetools      0.2-16     2018-12-24 [1] CRAN (R 4.0.2)
 colorspace     1.4-1      2019-03-18 [1] CRAN (R 4.0.2)
 crayon         1.3.4      2017-09-16 [1] CRAN (R 4.0.2)
 DBI            1.1.0      2019-12-15 [1] CRAN (R 4.0.2)
 dbplyr         1.4.4      2020-05-27 [1] CRAN (R 4.0.2)
 desc           1.2.0      2018-05-01 [1] CRAN (R 4.0.2)
 devtools       2.3.1      2020-07-21 [1] CRAN (R 4.0.2)
 digest         0.6.25     2020-02-23 [1] CRAN (R 4.0.2)
 dplyr        * 1.0.2      2020-08-18 [1] CRAN (R 4.0.2)
 effects        4.2-0      2020-08-11 [1] CRAN (R 4.0.2)
 effectsize     0.3.2      2020-07-27 [1] CRAN (R 4.0.2)
 ellipsis       0.3.1      2020-05-15 [1] CRAN (R 4.0.2)
 emmeans        1.5.0      2020-08-18 [1] CRAN (R 4.0.2)
 estimability   1.3        2018-02-11 [1] CRAN (R 4.0.2)
 evaluate       0.14       2019-05-28 [1] CRAN (R 4.0.1)
 fansi          0.4.1      2020-01-08 [1] CRAN (R 4.0.2)
 farver         2.0.3      2020-01-16 [1] CRAN (R 4.0.2)
 forcats      * 0.5.0      2020-03-01 [1] CRAN (R 4.0.2)
 fs             1.5.0      2020-07-31 [1] CRAN (R 4.0.2)
 generics       0.0.2      2018-11-29 [1] CRAN (R 4.0.2)
 ggeffects      0.15.1     2020-07-27 [1] CRAN (R 4.0.2)
 ggplot2      * 3.3.2      2020-06-19 [1] CRAN (R 4.0.2)
 ggthemes     * 4.2.0      2019-05-13 [1] CRAN (R 4.0.2)
 glmmTMB        1.0.2.1    2020-07-02 [1] CRAN (R 4.0.2)
 glue           1.4.2      2020-08-27 [1] CRAN (R 4.0.2)
 gtable         0.3.0      2019-03-25 [1] CRAN (R 4.0.2)
 haven          2.3.1      2020-06-01 [1] CRAN (R 4.0.2)
 hms            0.5.3      2020-01-08 [1] CRAN (R 4.0.2)
 htmltools      0.5.0      2020-06-16 [1] CRAN (R 4.0.2)
 httr           1.4.2      2020-07-20 [1] CRAN (R 4.0.2)
 insight        0.9.5      2020-09-07 [1] CRAN (R 4.0.2)
 jsonlite       1.7.1      2020-09-07 [1] CRAN (R 4.0.2)
 knitr        * 1.29.4     2020-08-23 [1] Github (yihui/knitr@bd64f2e)
 labeling       0.3        2014-08-23 [1] CRAN (R 4.0.2)
 lattice        0.20-41    2020-04-02 [1] CRAN (R 4.0.2)
 lifecycle      0.2.0      2020-03-06 [1] CRAN (R 4.0.2)
 lme4         * 1.1-23     2020-04-07 [1] CRAN (R 4.0.1)
 lubridate      1.7.9      2020-06-08 [1] CRAN (R 4.0.2)
 magrittr     * 1.5        2014-11-22 [1] CRAN (R 4.0.2)
 MASS           7.3-53     2020-09-09 [1] CRAN (R 4.0.2)
 Matrix       * 1.2-18     2019-11-27 [1] CRAN (R 4.0.2)
 memoise        1.1.0      2017-04-21 [1] CRAN (R 4.0.2)
 minqa          1.2.4      2014-10-09 [1] CRAN (R 4.0.2)
 mitools        2.4        2019-04-26 [1] CRAN (R 4.0.2)
 modelr         0.1.8      2020-05-19 [1] CRAN (R 4.0.2)
 multcomp       1.4-13     2020-04-08 [1] CRAN (R 4.0.2)
 munsell        0.5.0      2018-06-12 [1] CRAN (R 4.0.2)
 mvtnorm        1.1-1      2020-06-09 [1] CRAN (R 4.0.2)
```

```
 nlme           3.1-149    2020-08-23 [1] CRAN (R 4.0.2)
 nloptr         1.2.2.2    2020-07-02 [1] CRAN (R 4.0.2)
 nnet           7.3-14     2020-04-26 [1] CRAN (R 4.0.2)
 parameters     0.8.2      2020-07-24 [1] CRAN (R 4.0.2)
 performance    0.4.8      2020-07-27 [1] CRAN (R 4.0.2)
 pillar         1.4.6      2020-07-10 [1] CRAN (R 4.0.2)
 pkgbuild       1.1.0.9000 2020-08-06 [1] Github (r-lib/pkgbuild@3a87bd9)
 pkgconfig      2.0.3      2019-09-22 [1] CRAN (R 4.0.2)
 pkgload        1.1.0      2020-05-29 [1] CRAN (R 4.0.2)
 prettyunits    1.1.1      2020-01-24 [1] CRAN (R 4.0.2)
 processx       3.4.4      2020-09-03 [1] CRAN (R 4.0.2)
 ps             1.3.4      2020-08-11 [1] CRAN (R 4.0.2)
 purrr        * 0.3.4      2020-04-17 [1] CRAN (R 4.0.2)
 R6             2.4.1      2019-11-12 [1] CRAN (R 4.0.2)
 RColorBrewer   1.1-2      2014-12-07 [1] CRAN (R 4.0.2)
 Rcpp           1.0.5      2020-07-06 [1] CRAN (R 4.0.2)
 readr        * 1.3.1      2018-12-21 [1] CRAN (R 4.0.2)
 readxl         1.3.1      2019-03-13 [1] CRAN (R 4.0.2)
 remotes        2.2.0      2020-07-21 [1] CRAN (R 4.0.2)
 reprex         0.3.0      2019-05-16 [1] CRAN (R 4.0.2)
 rlang          0.4.7      2020-07-09 [1] CRAN (R 4.0.2)
 rmarkdown      2.3        2020-06-18 [1] CRAN (R 4.0.2)
 rprojroot      1.3-2      2018-01-03 [1] CRAN (R 4.0.2)
 rstudioapi     0.11       2020-02-07 [1] CRAN (R 4.0.2)
 rvest          0.3.6      2020-07-25 [1] CRAN (R 4.0.2)
 sandwich       2.5-1      2019-04-06 [1] CRAN (R 4.0.2)
 scales         1.1.1      2020-05-11 [1] CRAN (R 4.0.2)
 sessioninfo    1.1.1      2018-11-05 [1] CRAN (R 4.0.2)
 sjlabelled     1.1.7      2020-09-24 [1] CRAN (R 4.0.2)
 sjmisc         2.8.5      2020-05-28 [1] CRAN (R 4.0.2)
 sjPlot       * 2.8.4      2020-05-24 [1] CRAN (R 4.0.2)
 sjstats        0.18.0     2020-05-06 [1] CRAN (R 4.0.2)
 snakecase      0.11.0     2019-05-25 [1] CRAN (R 4.0.2)
 statmod        1.4.34     2020-02-17 [1] CRAN (R 4.0.2)
 stringi        1.4.6      2020-02-17 [1] CRAN (R 4.0.2)
 stringr      * 1.4.0      2019-02-10 [1] CRAN (R 4.0.2)
 survey         4.0        2020-04-03 [1] CRAN (R 4.0.2)
 survival       3.2-3      2020-06-13 [1] CRAN (R 4.0.2)
 testthat       2.3.2      2020-03-02 [1] CRAN (R 4.0.2)
 TH.data        1.0-10     2019-01-21 [1] CRAN (R 4.0.2)
 tibble       * 3.0.3      2020-07-10 [1] CRAN (R 4.0.2)
 tidyr        * 1.1.2      2020-08-27 [1] CRAN (R 4.0.2)
 tidyselect     1.1.0      2020-05-11 [1] CRAN (R 4.0.2)
 tidyverse    * 1.3.0      2019-11-21 [1] CRAN (R 4.0.2)
 TMB            1.7.18     2020-07-27 [1] CRAN (R 4.0.2)
 usethis        1.6.1      2020-04-29 [1] CRAN (R 4.0.2)
 vctrs          0.3.4      2020-08-29 [1] CRAN (R 4.0.2)
 withr          2.2.0      2020-04-20 [1] CRAN (R 4.0.2)
 xfun           0.17       2020-09-09 [1] CRAN (R 4.0.2)
 xml2           1.3.2      2020-04-23 [1] CRAN (R 4.0.2)
 xtable         1.8-4      2019-04-21 [1] CRAN (R 4.0.2)
 yaml           2.2.1      2020-02-01 [1] CRAN (R 4.0.2)
 zoo            1.8-8      2020-05-02 [1] CRAN (R 4.0.2)

[1] /Library/Frameworks/R.framework/Versions/4.0/Resources/library
```