Wrapping up

Loose ends and more content than we'll get to

Daniel Anderson

Week 10

Agenda

- Review Homework 3
- RQ to models practice
- Finishing up the last bits from last time
- Quick discussion on missing data
- Cross–classified models
    - And a brief foray into multiple membership models

Bonus info for you if you want to cover on your own

- Fitting a 1PL model through a mixed effects framework

Review Homework 3

Research Questions to models

Data

Read in the `alcohol-adolescents.csv` data.

```
library(tidyverse)
alc <- read_csv(here::here("data", "alcohol-adolescents.csv"))
```

02:00

Data

```
alc
```

```
## # A tibble: 246 x 6
##       id   age   coa  male   alcuse      peer
##    <dbl> <dbl> <dbl> <dbl>    <dbl>     <dbl>
##  1     1    14     1     0 1.732051 1.264911
##  2     1    15     1     0 2        1.264911
##  3     1    16     1     0 2        1.264911
##  4     2    14     1     1 0        0.8944272
##  5     2    15     1     1 0        0.8944272
##  6     2    16     1     1 1        0.8944272
##  7     3    14     1     1 1        0.8944272
##  8     3    15     1     1 2        0.8944272
##  9     3    16     1     1 3.316625 0.8944272
## 10     4    14     1     1 0        1.788854
## # … with 236 more rows
```

RQ 1

> Does alcohol use increase with age?

Actually no *one* right answer, but some are more correct than others.

02:00

How I would start

- First, estimate the model that makes the most theoretical sense, to me – random intercepts and slopes
    - If that model fits fine, just go forward with it and interpret
    - If that model does *not* fit well (i.e., convergence warnings), contrast that model with one that constrains the slope to be constant across partcipants

Theoretical model

```
library(lme4)
alc <- alc %>%
  mutate(age_c = age - 14)

rq1a <- lmer(alcuse ~ age_c + (1 + age_c | id),
             data = alc)
```

Model summary

```
summary(rq1a)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: alcuse ~ age_c + (1 + age_c | id)
##    Data: alc
##
## REML criterion at convergence: 643.2
##
## Scaled residuals:
##     Min       1Q   Median       3Q      Max
## -2.48287 -0.37933 -0.07858  0.38876  2.49284
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  id       (Intercept) 0.6355   0.7972
##           age_c       0.1552   0.3939   -0.23
##  Residual             0.3373   0.5808
## Number of obs: 246, groups:  id, 82
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  0.65130    0.10573   6.160
## age_c        0.27065    0.06284   4.307
##
## Correlation of Fixed Effects:
##       (Intr)
```

RQ 2

> Do children of alcoholics have higher levels of alcohol use?

Keyword here: **levels**

This implies a change in the intercept, not the slope

```
rq2 <- lmer(alcuse ~ age_c + coa + (1 + age_c | id),
            data = alc)
```

02:00

RQ 3

> Do male children of alcoholics have higher alcohol use than female children of alcoholics?

Important – we're specifically interested in `coa == 1` cases.

However: We have to model main effects for `coa` and `male` to get at `male:coa`.

```
rq3 <- lmer(alcuse ~ age_c + coa + male + coa:male +
              (1 + age_c | id),
            data = alc)
```

02:00

RQ 4

> Do adolescents' alcohol use trajectories differ between males and females?

```
rq4 <- lmer(alcuse ~ age_c + male + age_c:male +
              (1 + age_c | id),
            data = alc)
```

02:00

RQ 5

> Does the trajectory of alcohol use differ for male children of alcoholics than female children of alcoholics?

Three–way interaction

```
rq5 <- lmer(alcuse ~ age_c + coa + male +
              age_c:male + age_c:coa + coa:male +
              age_c:male:coa +
              (1 + age_c | id),
            data = alc)
```

02:00

Alternative syntax

This syntax also includes the `coa:male` interaction

```
rq5b <- lmer(alcuse ~ age_c * coa * male +
               (1 + age_c | id),
             data = alc)
```

```
arm::display(rq5)
```

```
## lmer(formula = alcuse ~ age_c + coa + male + age_c:male + age_c:coa +
##     coa:male + age_c:male:coa + (1 + age_c | id), data = alc)
##                coef.est coef.se
## (Intercept)     0.42     0.19
## age_c           0.22     0.12
## coa             0.76     0.29
## male           -0.22     0.27
## age_c:male      0.15     0.17
## age_c:coa      -0.23     0.18
## coa:male        0.00     0.40
## age_c:coa:male  0.32     0.25
##
## Error terms:
##  Groups   Name        Std.Dev. Corr
##  id       (Intercept) 0.72
##           age_c       0.37     -0.20
##  Residual             0.58
## ---
## number of obs: 246, groups: id, 82
## AIC = 653.4, DIC = 597.1
## deviance = 613.3
```

```
arm::display(rq5b)
```

```
## lmer(formula = alcuse ~ age_c * coa * male + (1 + age_c | id),
##     data = alc)
##                coef.est coef.se
## (Intercept)     0.42     0.19
## age_c           0.22     0.12
## coa             0.76     0.29
## male           -0.22     0.27
## age_c:coa      -0.23     0.18
## age_c:male      0.15     0.17
## coa:male        0.00     0.40
## age_c:coa:male  0.32     0.25
##
## Error terms:
##  Groups   Name        Std.Dev. Corr
##  id       (Intercept) 0.72
##           age_c       0.37     -0.20
##  Residual             0.58
## ---
## number of obs: 246, groups: id, 82
## AIC = 653.4, DIC = 597.1
## deviance = 613.3
```

New data

Load in the `three-lev.csv` data

```
threelev <- read_csv(here::here("data", "three-lev.csv"))
threelev
```

```
## # A tibble: 7,230 x 12
##    schid       sid  size lowinc mobility female black hispanic retained
##    <dbl>     <dbl> <dbl>  <dbl>    <dbl>  <dbl> <dbl>    <dbl>    <dbl>
##  1  2020 273026452   380   40.3     12.5      0     0        1        0
##  2  2020 273026452   380   40.3     12.5      0     0        1        0
##  3  2020 273026452   380   40.3     12.5      0     0        1        0
##  4  2020 273030991   380   40.3     12.5      0     0        0        0
##  5  2020 273030991   380   40.3     12.5      0     0        0        0
##  6  2020 273030991   380   40.3     12.5      0     0        0        0
##  7  2020 273030991   380   40.3     12.5      0     0        0        0
##  8  2020 273030991   380   40.3     12.5      0     0        0        0
##  9  2020 273059461   380   40.3     12.5      0     0        1        0
## 10  2020 273059461   380   40.3     12.5      0     0        1        0
## # … with 7,220 more rows
```

01:00

RQ 1

> To what extent do math scores, and changes in math scores, vary between students versus between schools?

This is complicated

Two part analysis – first estimate the model, then compute the ICC for the intercept and the ICC for the slope

The model

```
rq1_math <- lmer(math ~ 1 + year +
                   (year | sid) + (year | schid),
                 data = threelev)
```

Variances

```
as.data.frame(VarCorr(rq1_math))
```

```
##        grp        var1 var2       vcov     sdcor
## 1      sid (Intercept) <NA> 0.64047114 0.8002944
## 2      sid        year <NA> 0.01125765 0.1061021
## 3      sid (Intercept) year 0.04678624 0.5509911
## 4    schid (Intercept) <NA> 0.16857056 0.4105735
## 5    schid        year <NA> 0.01126367 0.1061304
## 6    schid (Intercept) year 0.01734150 0.3979749
## 7 Residual        <NA> <NA> 0.30143334 0.5490295
```

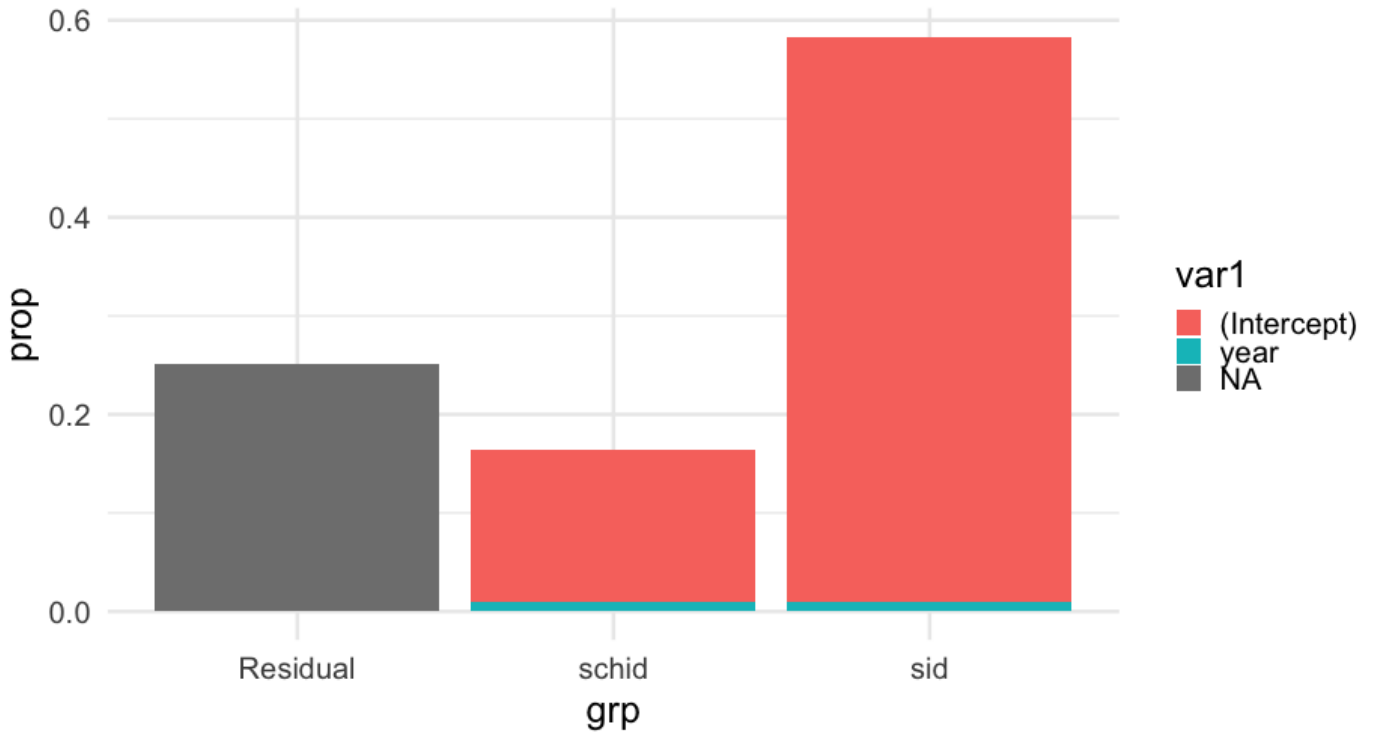
Proportions

```
as.data.frame(VarCorr(rq1_math)) %>%
  mutate(prop = vcov / sum(vcov))
```

```
##        grp        var1 var2       vcov     sdcor        prop
## 1      sid (Intercept) <NA> 0.64047114 0.8002944 0.535008142
## 2      sid        year <NA> 0.01125765 0.1061021 0.009403910
## 3      sid (Intercept) year 0.04678624 0.5509911 0.039082201
## 4    schid (Intercept) <NA> 0.16857056 0.4105735 0.140812939
## 5    schid        year <NA> 0.01126367 0.1061304 0.009408942
## 6    schid (Intercept) year 0.01734150 0.3979749 0.014485963
## 7 Residual        <NA> <NA> 0.30143334 0.5490295 0.251797903
```

Plot

```
as.data.frame(VarCorr(rq1_math)) %>%
  mutate(prop = vcov / sum(vcov)) %>%
  ggplot(aes(grp, prop)) +
  geom_col(aes(fill = var1))
```



RQ 2

> What is the average difference in math scores among students coded Black or Hispanic, versus those who are not?

```
rq2_math <- lmer(math ~ year + black + hispanic +
                   (year | sid) + (year | schid),
                 data = threelev,
                 control = lmerControl(optimizer = "bobyqa"))
```

02:00

Last one

> To what extent do the differences in gains between students coded Black or Hispanic, versus those who are not, vary between schools?

Note – this is a model that is difficult to fit. Bayes might help.

```
rq3_math <- lmer(math ~ year + black + hispanic +
                   year:black + year:hispanic +
                   (year | sid) +
                   (year + year:black + year:hispanic | schid),
                 data = threelev)
```

02:00

Low variance components make convergence difficult

```
arm::display(rq3_math)
```

```
## lmer(formula = math ~ year + black + hispanic + year:black +
##     year:hispanic + (year | sid) + (year + year:black + year:hispanic |
##     schid), data = threelev)
##                coef.est coef.se
## (Intercept)   -0.36     0.08
## year           0.78     0.02
## black         -0.59     0.08
## hispanic      -0.38     0.09
## year:black    -0.05     0.02
## year:hispanic  0.04     0.03
##
## Error terms:
##  Groups   Name           Std.Dev. Corr
##  sid      (Intercept)    0.79
##           year           0.10     0.56
##  schid    (Intercept)    0.34
##           year           0.11      0.39
##           year:black     0.07     -0.40 -0.46
##           year:hispanic  0.06      0.03 -0.38  0.90
##  Residual                0.55
## ---
## number of obs: 7230, groups: sid, 1721; schid, 60
## AIC = 16327.4, DIC = 16229.3
## deviance = 16258.4
```

Finishing up from last class

Reminder

Lung cancer data: Patients nested in doctors

```
hdp <- read_csv("https://stats.idre.ucla.edu/stat/data/hdp.csv")
  janitor::clean_names() %>%
  select(did, tumorsize, pain, lungcapacity, age, remission)
hdp
```

```
## # A tibble: 8,525 x 6
##      did tumorsize  pain lungcapacity      age remission
##    <dbl>     <dbl> <dbl>        <dbl>    <dbl>     <dbl>
##  1     1  67.98120     4    0.8010882 64.96824         0
##  2     1  64.70246     2    0.3264440 53.91714         0
##  3     1  51.56700     6    0.5650309 53.34730         0
##  4     1  86.43799     3    0.8484109 41.36804         0
##  5     1  53.40018     3    0.8864910 46.80042         0
##  6     1  51.65727     4    0.7010307 51.92936         0
##  7     1  78.91707     3    0.8908539 53.82926         0
##  8     1  69.83325     3    0.6608795 46.56223         0
##  9     1  62.85259     4    0.9088714 54.38936         0
## 10     1  71.77790     5    0.9593268 50.54465         0
## # … with 8,515 more rows
```

Predict remission

Build a model where age, lung capacity, and tumor size predict whether or not the patient was in remission.

- Build the model so you can evaluate whether or not the relation between the tumor size and likelihood of remission depends on age
- Allow the intercept to vary by the doctor ID.
- Fit the model using **brms**

Lung cancer remission model

```
library(brms)
lc <- brm(
  remission ~ age * tumorsize + lungcapacity + (1|did),
  data = hdp,
  family = bernoulli(link = "logit"),
  cores = 4,
  backend = "cmdstan"
)
```

```
##
-

\

|

/

-

\
```

Variance by Doctor

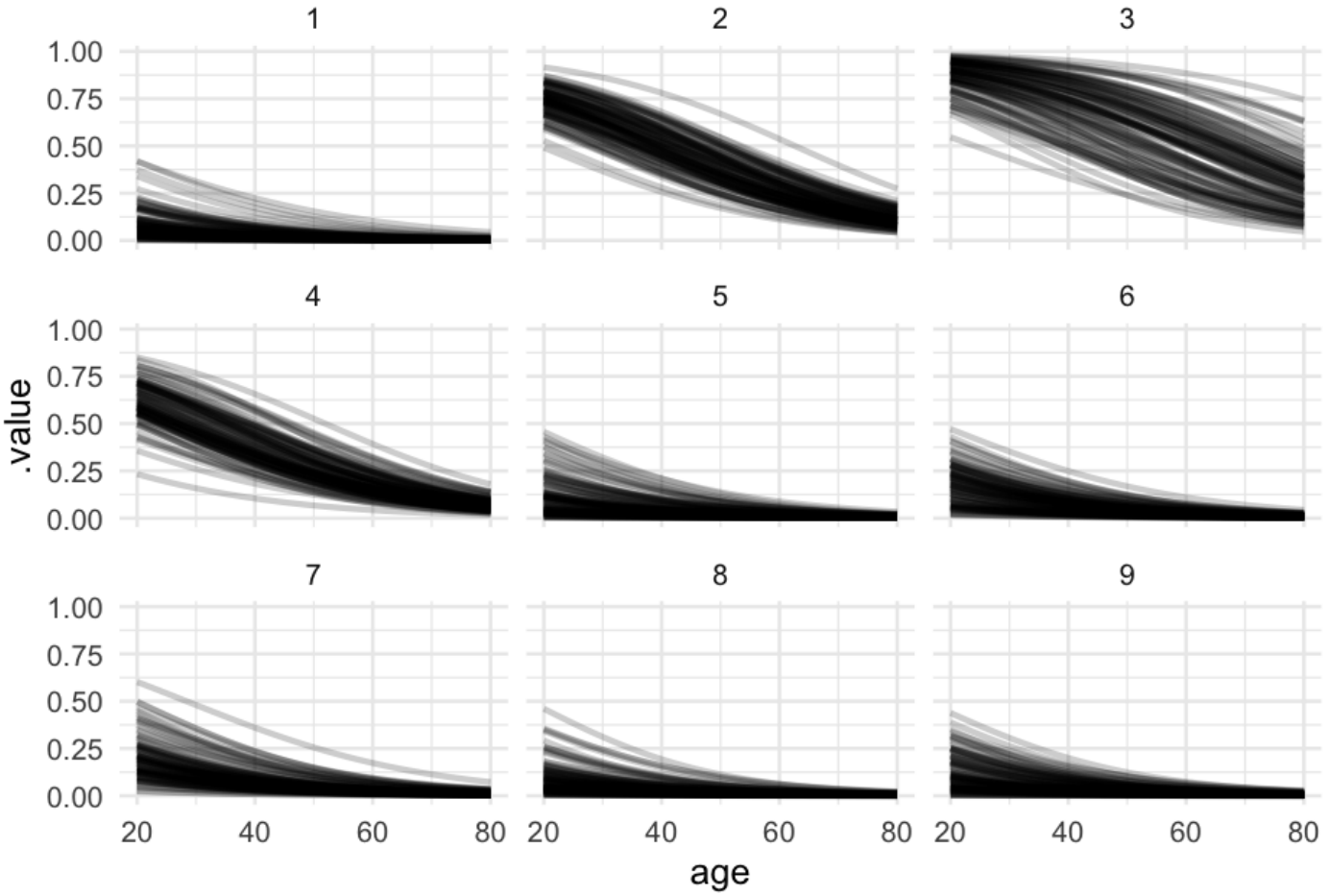
Let's look at the relation between age and probability of remission for each of the first nine doctors.

```
library(tidybayes)
pred_age_doctor <- expand.grid(
    did = 1:9,
    age = 20:80,
    tumorsize = mean(hdp$tumorsize),
    lungcapacity = mean(hdp$lungcapacity)
  ) %>%
  add_fitted_draws(model = lc, n = 100)
```

```
pred_age_doctor
```

```
## # A tibble: 54,900 x 9
## # Groups:   did, age, tumorsize, lungcapacity, .row [549]
##      did   age tumorsize lungcapacity  .row .chain .iteration .draw
##    <int> <int>     <dbl>        <dbl> <int>  <int>      <int> <int>
##  1     1    20  70.88067    0.7740865     1     NA         NA     7 0.06
##  2     1    20  70.88067    0.7740865     1     NA         NA    13 0.10
##  3     1    20  70.88067    0.7740865     1     NA         NA    99 0.07
##  4     1    20  70.88067    0.7740865     1     NA         NA   113 0.06
##  5     1    20  70.88067    0.7740865     1     NA         NA   121 0.02
##  6     1    20  70.88067    0.7740865     1     NA         NA   122 0.10
##  7     1    20  70.88067    0.7740865     1     NA         NA   141 0.08
##  8     1    20  70.88067    0.7740865     1     NA         NA   183 0.04
##  9     1    20  70.88067    0.7740865     1     NA         NA   224 0.22
## 10     1    20  70.88067    0.7740865     1     NA         NA   288 0.08
## # … with 54,890 more rows
```

```
ggplot(pred_age_doctor, aes(age, .value)) +
  geom_line(aes(group = .draw), alpha = 0.2) +
  facet_wrap(~did)
```



Look at our variables

```
get_variables(lc)
```

```
##   [1] "b_Intercept"           "b_age"                 "b_tumorsize"
##   [7] "Intercept"             "r_did[1,Intercept]"    "r_did[2,Intercept]"
##  [13] "r_did[6,Intercept]"    "r_did[7,Intercept]"    "r_did[8,Intercept]"
##  [19] "r_did[12,Intercept]"   "r_did[13,Intercept]"   "r_did[14,Intercept]
##  [25] "r_did[18,Intercept]"   "r_did[19,Intercept]"   "r_did[20,Intercept]
##  [31] "r_did[24,Intercept]"   "r_did[25,Intercept]"   "r_did[26,Intercept]
##  [37] "r_did[30,Intercept]"   "r_did[31,Intercept]"   "r_did[32,Intercept]
##  [43] "r_did[36,Intercept]"   "r_did[37,Intercept]"   "r_did[38,Intercept]
##  [49] "r_did[42,Intercept]"   "r_did[43,Intercept]"   "r_did[44,Intercept]
##  [55] "r_did[48,Intercept]"   "r_did[49,Intercept]"   "r_did[50,Intercept]
##  [61] "r_did[54,Intercept]"   "r_did[55,Intercept]"   "r_did[56,Intercept]
##  [67] "r_did[60,Intercept]"   "r_did[61,Intercept]"   "r_did[62,Intercept]
##  [73] "r_did[66,Intercept]"   "r_did[67,Intercept]"   "r_did[68,Intercept]
##  [79] "r_did[72,Intercept]"   "r_did[73,Intercept]"   "r_did[74,Intercept]
##  [85] "r_did[78,Intercept]"   "r_did[79,Intercept]"   "r_did[80,Intercept]
##  [91] "r_did[84,Intercept]"   "r_did[85,Intercept]"   "r_did[86,Intercept]
##  [97] "r_did[90,Intercept]"   "r_did[91,Intercept]"   "r_did[92,Intercept]
## [103] "r_did[96,Intercept]"   "r_did[97,Intercept]"   "r_did[98,Intercept]
## [109] "r_did[102,Intercept]" "r_did[103,Intercept]" "r_did[104,Intercept
## [115] "r_did[108,Intercept]" "r_did[109,Intercept]" "r_did[110,Intercept
## [121] "r_did[114,Intercept]" "r_did[115,Intercept]" "r_did[116,Intercept
## [127] "r_did[120,Intercept]" "r_did[121,Intercept]" "r_did[122,Intercept
## [133] "r_did[126,Intercept]" "r_did[127,Intercept]" "r_did[128,Intercept
## [139] "r_did[132,Intercept]" "r_did[133,Intercept]" "r_did[134,Intercept
```

Multiple comparisons

One of the nicest things about Bayes is that any comparison you want to make can be made without jumping through a lot of additional hoops (e.g., adjusting α).

Scenario

Imagine a **35** year old has a tumor measuring **58 millimeters** and a lung capacity rating of **0.81**.

What would we estimate as the probability of remission if this patient had `did == 1` versus `did == 2`?

Fixed effects

Not really "fixed", but rather just average relation

```
fe <- lc %>%
  spread_draws(b_Intercept, b_age, b_tumorsize, b_lungcapacity,
               `b_age:tumorsize`)
fe
```

```
## # A tibble: 4,000 x 8
##    .chain .iteration .draw b_Intercept      b_age b_tumorsize b_lungcapa
##     <int>      <int> <int>       <dbl>      <dbl>       <dbl>          <
##  1      1          1     1     5.42858  -0.119206  -0.0544173      -0.362
##  2      1          2     2     4.46859  -0.100647  -0.0450108      -0.341
##  3      1          3     3     1.96492  -0.0468593 -0.0102471      -0.070
##  4      1          4     4     0.917227 -0.0301529  0.00952746     -0.068
##  5      1          5     5     1.53674  -0.0396987 -0.00207063     -0.067
##  6      1          6     6     0.881689 -0.0345708  0.00102342      0.051
##  7      1          7     7     1.64537  -0.0490206 -0.00906014      0.144
##  8      1          8     8     1.38195  -0.0418099 -0.00452064     -0.082
##  9      1          9     9     0.580196 -0.0293446  0.00953349     -0.214
## 10      1         10    10     3.48012  -0.0843227 -0.0282885       0.226
## # … with 3,990 more rows
```

Data

```
age <- 35
tumor_size <- 58
lung_cap <- 0.81
```

Population–level predictions (there's other ways we could do this, but it's good to remind ourselves the "by hand" version too)

```
pop_level <-
  fe$b_Intercept +
  (fe$b_age * age) +
  (fe$b_tumorsize * tumor_size) +
  (fe$b_lungcapacity * lung_cap) +
  (fe$`b_age:tumorsize` * (age * tumor_size))
pop_level
```

```
##    [1] -0.400649170 -0.548156420 -0.530601573 -0.565097334 -0.496463910
##   [12] -0.273899070 -0.164508090 -0.505821720 -0.786540482 -0.641475783
##   [23] -0.276976520 -0.574581568 -0.613817338 -0.457796800 -0.571542413
##   [34] -0.394965420 -0.272091990 -0.233677040 -0.145021650 -0.203014758
```

Plot population level

```
pd <- tibble(population_level = pop_level)

ggplot(pd, aes(population_level)) +
  geom_histogram(fill = "#61adff",
                 color = "white") +
  geom_vline(xintercept = median(pd$population_level),
             color = "magenta",
             size = 2)
```

Add in did estimates

```
dids <- spread_draws(lc, r_did[did, ])

did1 <- filter(dids, did == 1)
did2 <- filter(dids, did == 2)

pred_did1 <- pop_level + did1$r_did
pred_did2 <- pop_level + did2$r_did
```

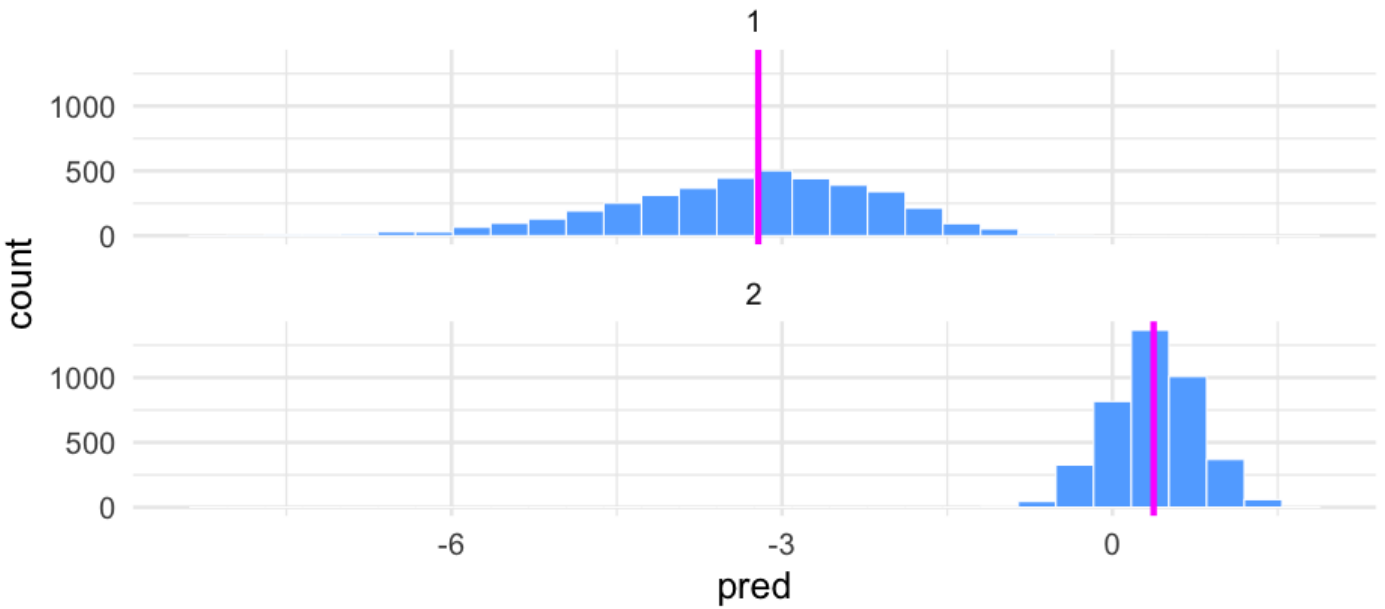
Distributions

```
did12 <- tibble(did = rep(1:2, each = length(pred_did1)),
                pred = c(pred_did1, pred_did2))

did12_medians <- did12 %>%
  group_by(did) %>%
  summarize(did_median = median(pred))

did12_medians
```

```
## # A tibble: 2 x 2
##     did did_median
##   <int>      <dbl>
## 1     1 -3.216032
## 2     2  0.3734396
```

Plot

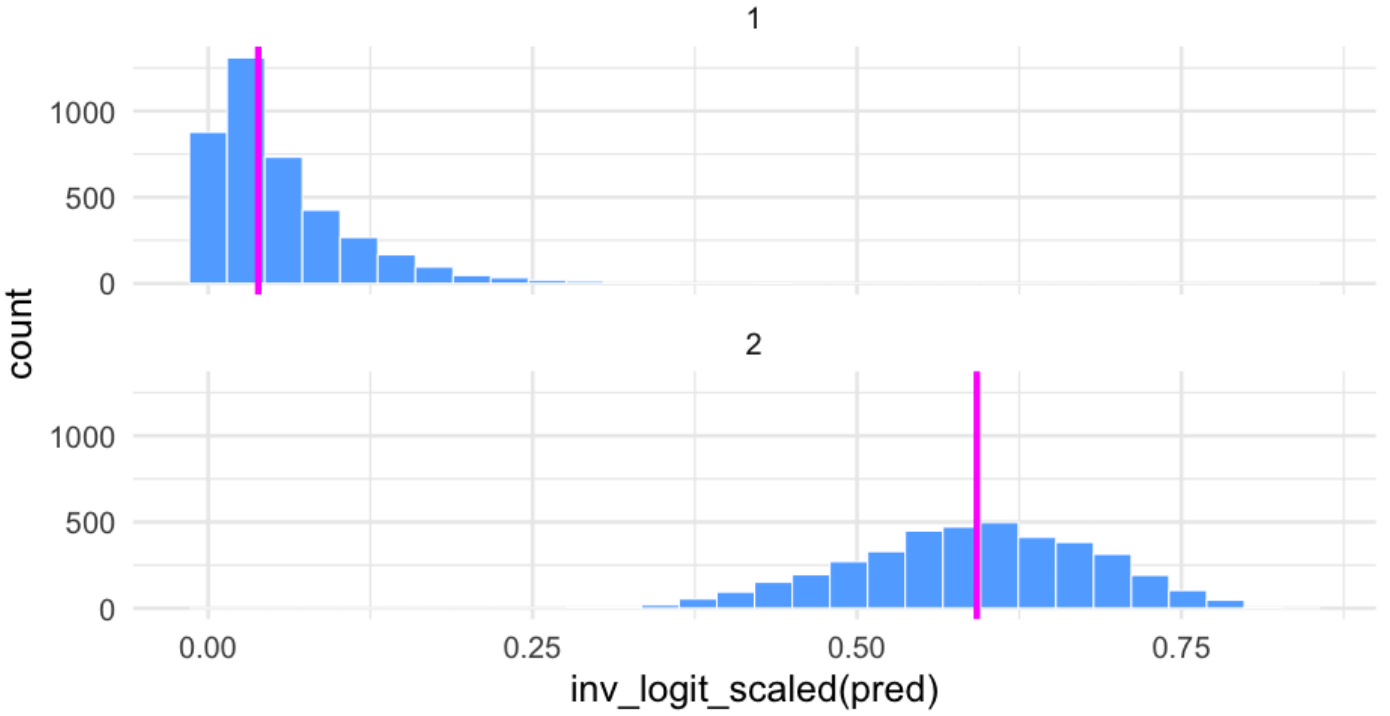
```
ggplot(did12, aes(pred)) +
  geom_histogram(fill = "#61adff",
                 color = "white") +
  geom_vline(aes(xintercept = did_median), data = did12_medians,
             color = "magenta",
             size = 2) +
  facet_wrap(~did, ncol = 1)
```



Transform

Let's look at this again on the probability scale using `brms::inv_logit_scaled()` to make the transformation.

```
ggplot(did12, aes(inv_logit_scaled(pred))) +
  geom_histogram(fill = "#61adff",
                 color = "white") +
  geom_vline(aes(xintercept = inv_logit_scaled(did_median)),
             data = did12_medians,
             color = "magenta",
             size = 2) +
  facet_wrap(~did, ncol = 1)
```


1
2
count
1000
500
0
0.00
0.25
0.50
0.75
inv_logit_scaled(pred)

Difference

- The difference in the probability of remission for our theoretical patient is large between the two doctors.
- The median difference in log–odds is

```
diff(did12_medians$did_median)
```

```
## [1] 3.589472
```

Exponentiation

We can exponentiate the log–odds to get normal odds

These are fairly interpretable (especially when greater than 1)

```
# probability
inv_logit_scaled(did12_medians$did_median)
```

```
## [1] 0.03856683 0.59228985
```

```
# odds
exp(did12_medians$did_median)
```

```
## [1] 0.0401139 1.4527229
```

```
# odds of the difference
exp(diff(did12_medians$did_median))
```

```
## [1] 36.21495
```

We estimate that our theoretical patient is about 36 times **more likely** (!) to go into remission if they had `did` 2, instead of 1.

Confidence in difference?

Everything is a distribution

Just compute the difference in these distributions, and we get a new distribution, which we can use to summarize our uncertainty

```
did12_wider <- tibble(
  did1 = pred_did1,
  did2 = pred_did2
) %>%
  mutate(diff = did2 - did1)

did12_wider
```

```
## # A tibble: 4,000 x 3
##          did1        did2     diff
##         <dbl>       <dbl>    <dbl>
##  1 -4.732689   1.005471   5.73816
##  2 -4.845796   0.8612836  5.70708
##  3 -2.986972   0.4373794  3.424351
##  4 -0.9940453  0.1152017  1.109247
```

Summarize

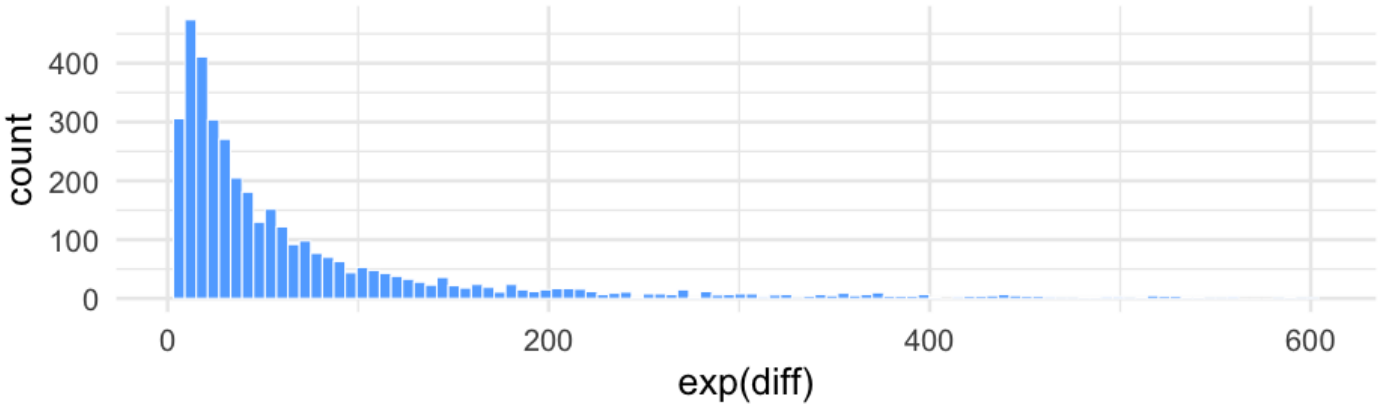
```
qtile_diffs <- quantile(did12_wider$diff,
                        probs = c(0.025, 0.5, 0.975))

exp(qtile_diffs)
```

```
##       2.5%        50%      97.5%
##   4.796723  36.883655 600.266651
```

Plot distribution

Show the most likely 95% of the distribution

```
# filter a few extreme observations
did12_wider %>%
  filter(diff > qtile_diffs[1] & diff < qtile_diffs[3]) %>%
ggplot(aes(exp(diff))) +
  geom_histogram(fill = "#61adff",
                 color = "white",
                 bins = 100)
```



Directionality

Let's say we want to simplify the question to directionality.

Is there a greater chance of remission for `did` 2 than 1?

```
table(did12_wider$diff > 0) / 4000
```

```
##
## TRUE
##    1
```

The distributions are not overlapping at all – therefore, we are as certain as we can be that the odds of remission are higher with `did` 2 than 1.

One more quick example

Let's do the same thing, but comparing `did` 2 and 3.

```
did3 <- filter(dids, did == 3)
pred_did3 <- pop_level + did3$r_did

did23 <- did12_wider %>%
  select(-did1, -diff) %>%
  mutate(did3 = pred_did3,
         diff = did3 - did2)
did23
```

```
## # A tibble: 4,000 x 3
##           did2        did3        diff
##          <dbl>       <dbl>       <dbl>
##  1  1.005471    0.8228108  -0.18266
##  2  0.8612836   0.8277436  -0.03354000
##  3  0.4373794   1.004198    0.566819
##  4  0.1152017   1.194783    1.079581
##  5  0.1754881   1.432726    1.257238
##  6  0.6139410   1.354041    0.7401000
##  7  0.6795638   2.767964    2.0884
##  8  0.3006396   1.548721    1.248081
##  9 -0.07275537  2.912974    2.985729
## 10  0.8960439  -0.06564811 -0.961692
```

Directionality

```
table(did23$diff > 0) / 4000
```

```
##
##   FALSE    TRUE
## 0.1415 0.8585
```

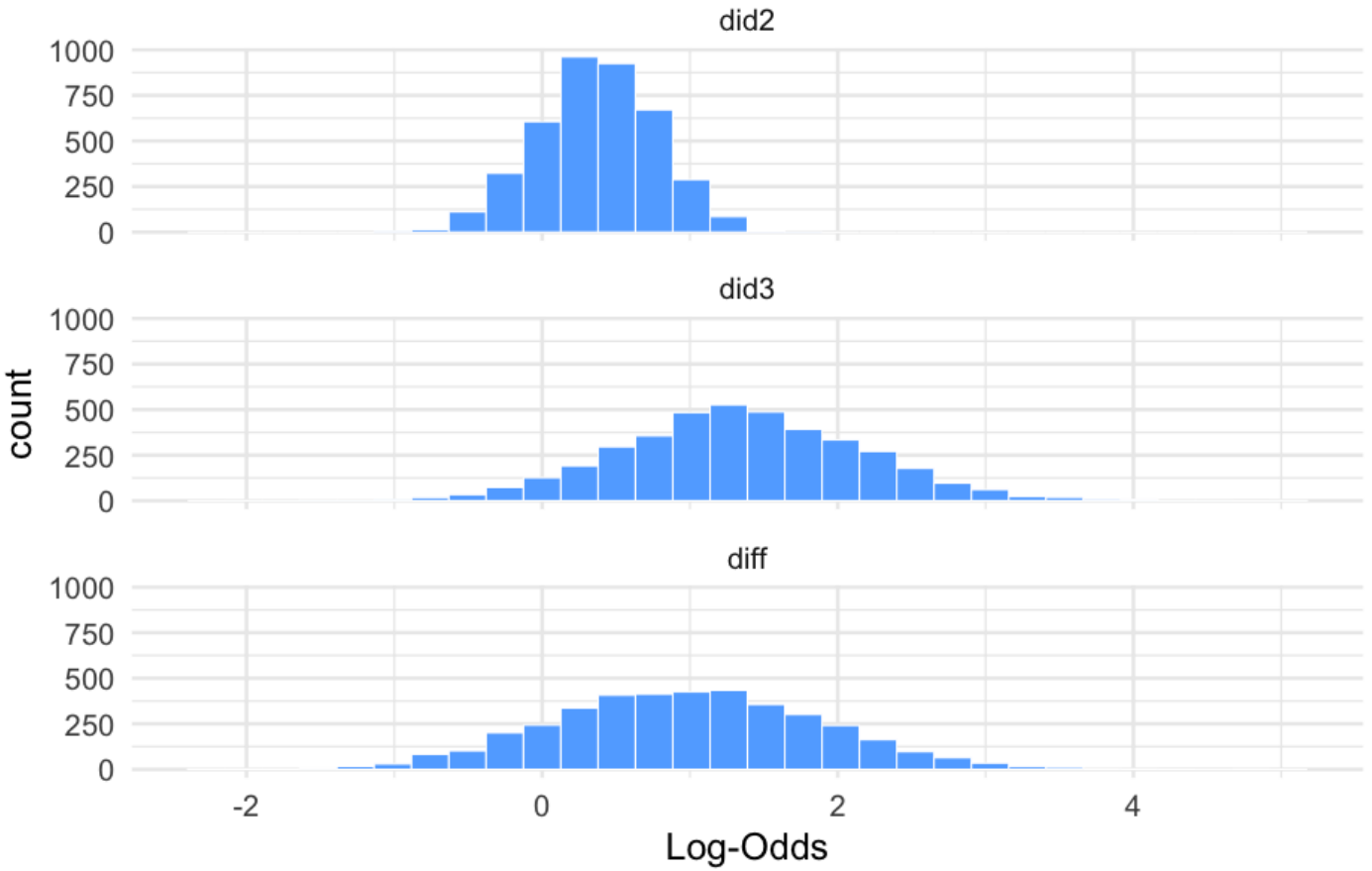
So there's roughly an 86% chance that the odds of remission are higher with with `did` 3 than 2.

Plot data

```
pd23 <- did23 %>%
  pivot_longer(did2:diff,
               names_to = "Distribution",
               values_to = "Log-Odds")
pd23
```

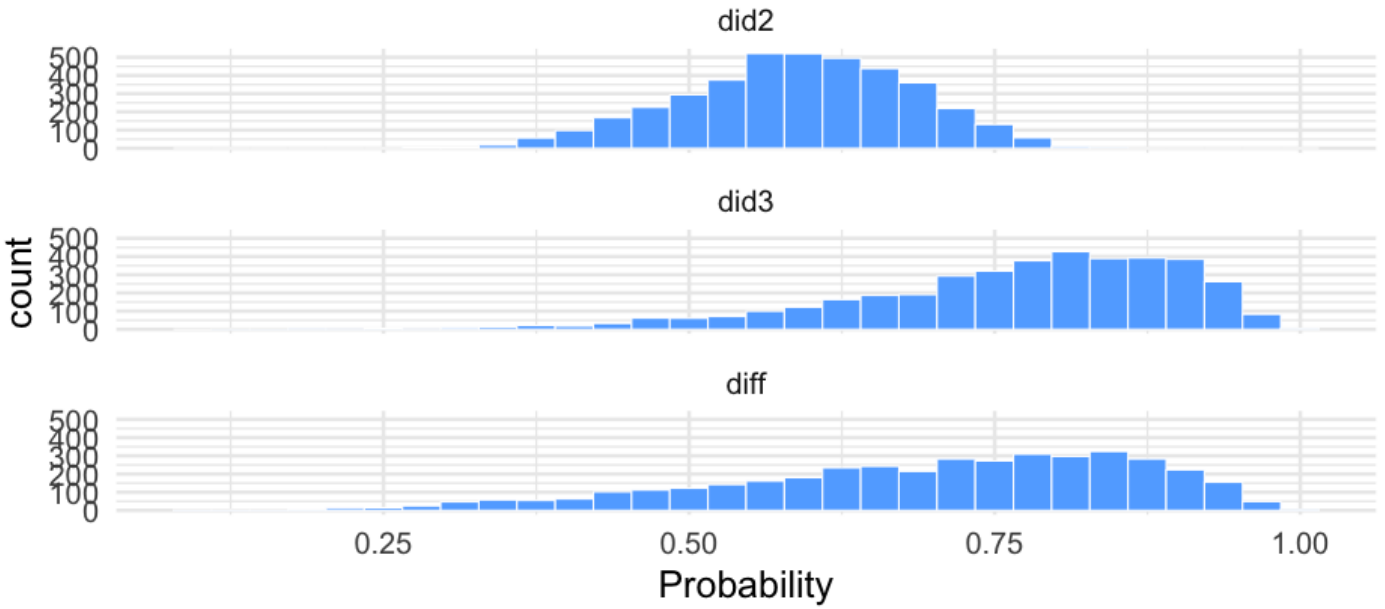
```
## # A tibble: 12,000 x 2
##    Distribution  `Log-Odds`
##    <chr>              <dbl>
##  1 did2            1.005471
##  2 did3            0.8228108
##  3 diff           -0.18266
##  4 did2            0.8612836
##  5 did3            0.8277436
##  6 diff           -0.03354000
##  7 did2            0.4373794
##  8 did3            1.004198
##  9 diff            0.566819
## 10 did2            0.1152017
## # … with 11,990 more rows
```

```
ggplot(pd23, aes(`Log-Odds`)) +
  geom_histogram(fill = "#61adff",
                 color = "white") +
  facet_wrap(~Distribution, ncol = 1)
```



Probability scale

```
pd23 %>%
  mutate(Probability = inv_logit_scaled(`Log-Odds`)) %>%
ggplot(aes(Probability)) +
  geom_histogram(fill = "#61adff",
                 color = "white") +
  facet_wrap(~Distribution, ncol = 1)
```



Break

05:00

Missing data

Disclaimer

- Missing data is a **massive** topic
- I'm hoping/assuming you've covered it some in other classes
- This is mostly about implementation options

Missing data on the DV

- Mostly what we tend to talk about in classes
- Also regularly the least problematic
- If we can assume MAR (missing at random conditional on covariates), most modern models do a pretty good job

Missing data on the IVs

- Much more problematic, no matter the model or application
- Remove all cases with any missingness on any IV?
  - Limits your sample size
  - Might (probably?) introduces new sources of bias
- Impute?
  - Often ethical challenges here – do you really want to impute somebody's gender?


I GIVE
UP
GIFSec.com

Solution?

- There really isn't a great one. Be clear about the decisions you do make.
- If you do choose imputation, use multiple imputation
    - This will allow you to have uncertainty in your imputation
- The purpose is to get unbiased population estimates for your parameters (not make inferences about an individual for whom data were imputed)

Missing IDs

- In multilevel models, you always have IDs linking the data to the higher levels
- If you are missing these IDs, I'm not really sure what to tell you
    - This is particularly common with longitudinal data (e.g., missing prior school IDs)
    - In rare cases, you can make assumptions and impute, but those are few and far between, in my experience, and the assumptions are still pretty dangerous

Let's do it

Multiple imputation

This part is general, and not specific to multilevel modeling

First, install/load the **{mice}** package (you might also check out **{Amelia}**)

```
library(mice)
```

Data

We'll impute data from the nhanes dataset, which comes with **{mice}**

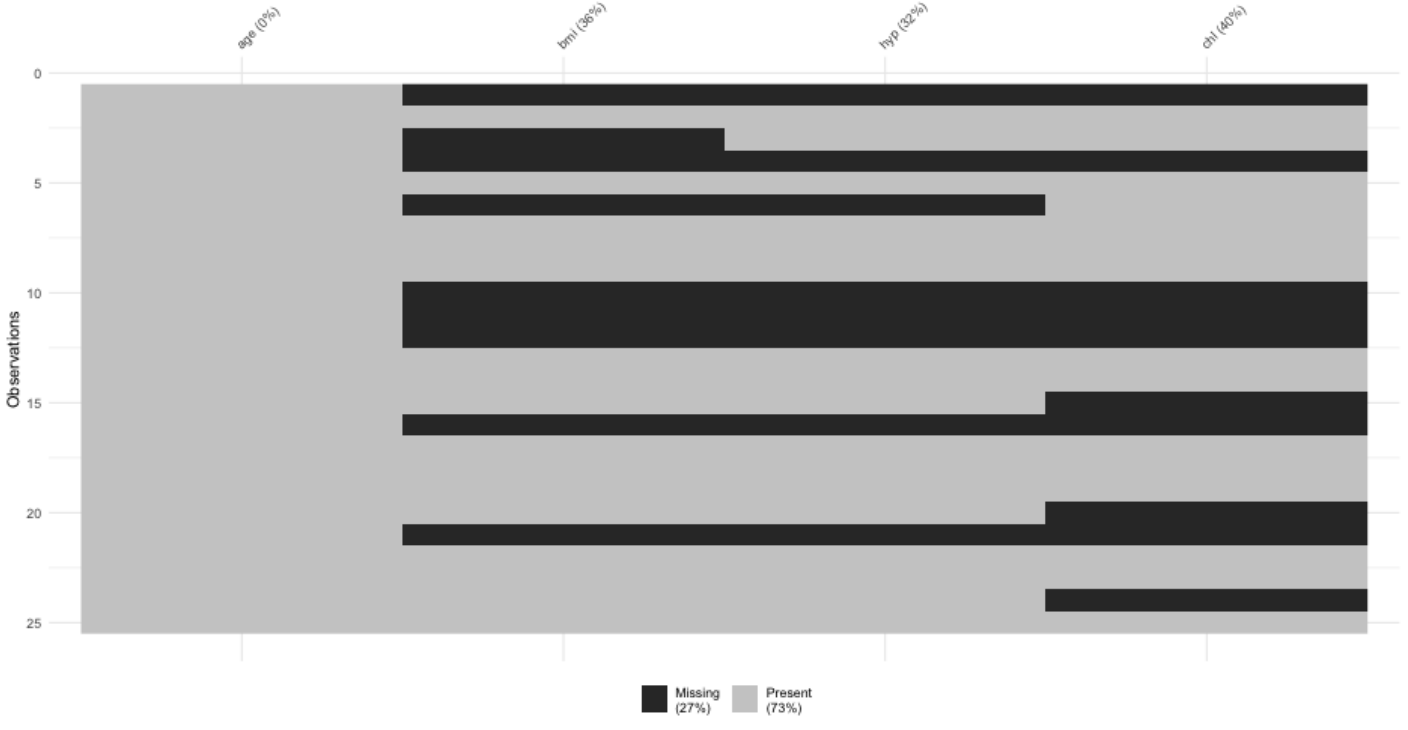
```
head(nhanes)
```

```
##   age  bmi hyp chl
## 1   1   NA  NA  NA
## 2   2 22.7   1 187
## 3   1   NA   1 187
## 4   3   NA  NA  NA
## 5   1 20.4   1 113
## 6   3   NA  NA 184
```

How much missingness

A lot

```
#install.packages("naniar")
naniar::vis_miss(nhanes)
```



Multiple imputation

- First, we're going to create 5 new dataset, each one with the missing data imputed

```
mi_nhanes <- mice(nhanes, m = 5, print = FALSE)
```

MI for BMI

```
mi_nhanes$imp$bmi
```

```
##       1    2    3    4    5
## 1  30.1 22.7 35.3 30.1 30.1
## 3  27.2 22.0 33.2 29.6 26.3
## 4  22.5 24.9 27.4 21.7 21.7
## 6  22.5 20.4 22.5 21.7 20.4
## 10 27.4 25.5 27.2 20.4 33.2
## 11 20.4 27.5 22.5 29.6 22.5
## 12 25.5 27.5 24.9 27.5 33.2
## 16 27.2 25.5 27.4 35.3 22.0
## 21 22.5 35.3 35.3 30.1 30.1
```

Fit model w/brms

Now just feed the **{mice}** object to `brms::brm_multiple()` as your data.

Note – this is considerably easier than it is with **lme4**, but it is do–able

```
m_mice <- brm_multiple(bmi ~ age * chl,
                       data = mi_nhanes)
```

Alternative

- A neat thing we can do with Bayes, is to impute *on the fly* using the posterior
- We still get uncertainty because of the repeated samples we're taking from the posterior anyway
- With **{brms}**, we can do this by just passing a slightly more complicated formula

Missing formula

We specify a model for each column that has missingness

We have missing data in `bmi` and `chl` (not `age`).

`bmi` is our outcome, and it will be modeled by `age` and the *complete* (missing data imputed) `chl` variable, as well as their interaction

The missing data in `chl` will be imputed via a model with `age` as its predictor!

We're basically fitting two models at once.

In Code

The `| mi()` part says to include missing data

```
bayes_impute_formula <-
  bf(bmi | mi() ~ age * mi(chl)) +  # base model
  bf(chl | mi() ~ age) + # model for chl missingness
  set_rescor(FALSE) # we don't estimate the residual correlation
```

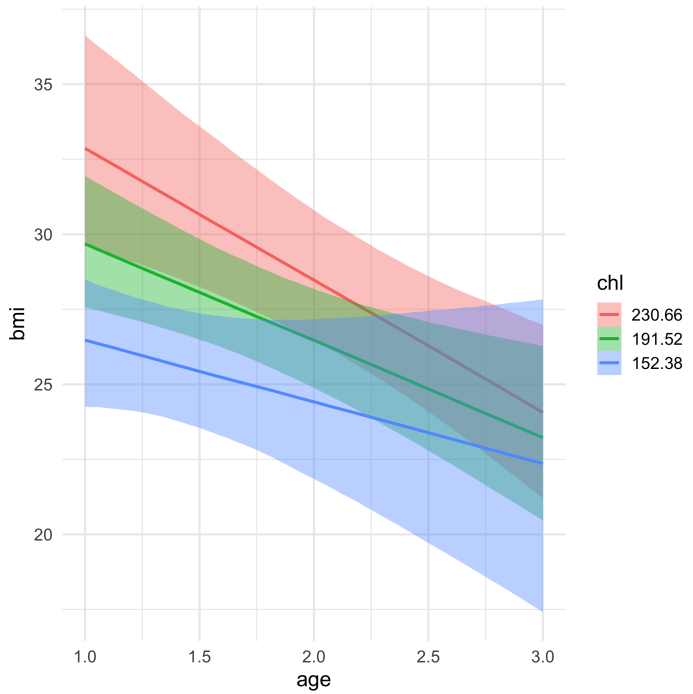
Fit

```
m_onfly <- brm(bayes_impute_formula, data = nhanes)
```

Comparison

Multiple imputation before modeling

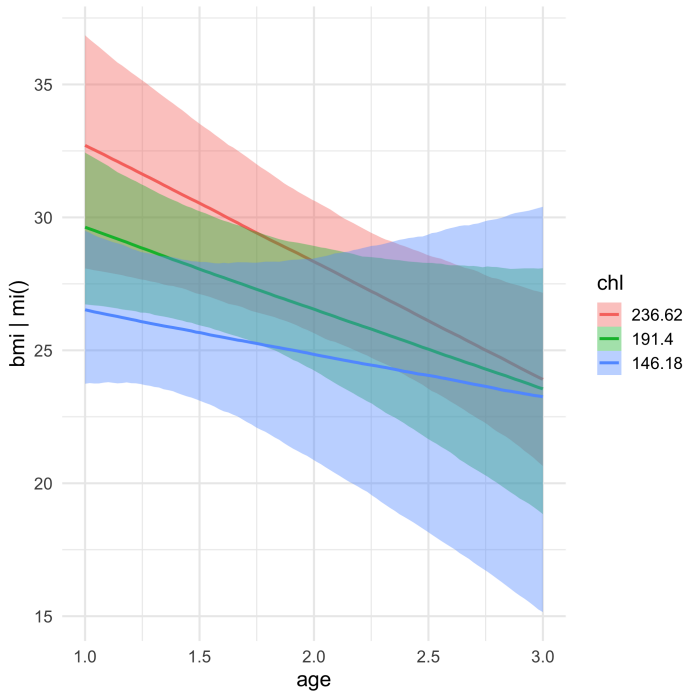
```
conditional_effects(m_mice, "age:chl", resp = "bmi")
```



Comparison

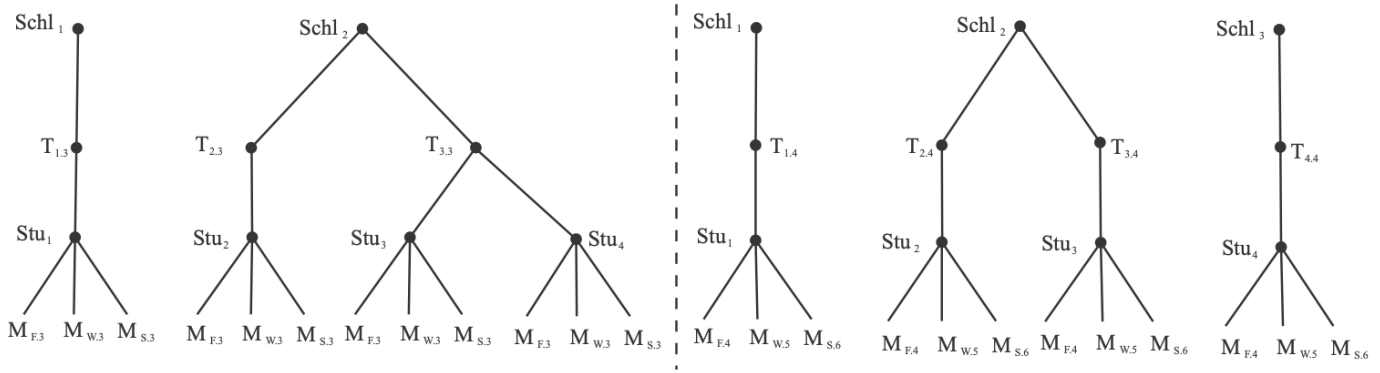
Imputation during modeling

```
conditional_effects(m_onfly, "age:chl", resp = "bmi")
```

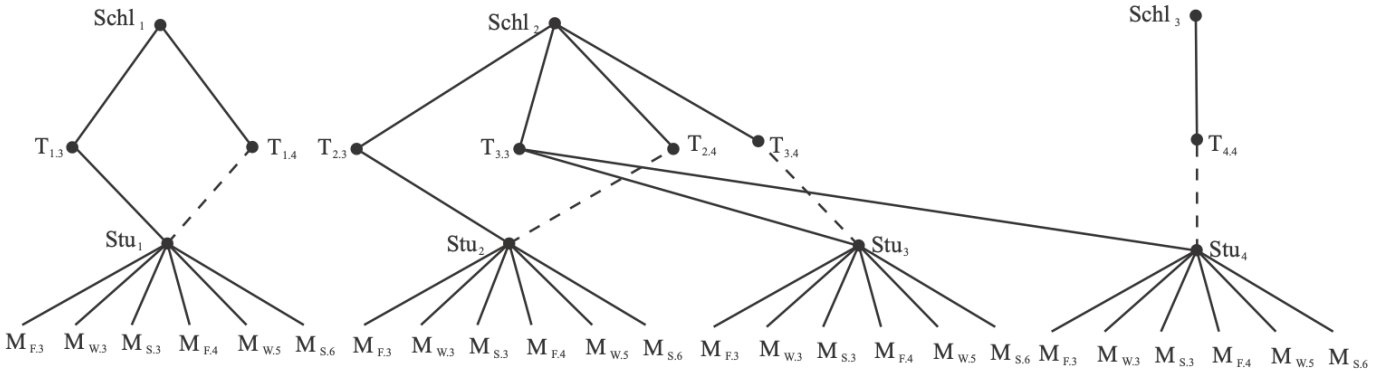


Cross classified models

Nested model



Crossed


Schl 1
T1.3
T1.4
Stu1
MF.3 MW.3 MS.3 MF.4 MW.5 MS.6
Schl 2
T2.3
T3.3
T2.4
T3.4
Stu2
Stu3
MF.3 MW.3 MS.3 MF.4 MW.5 MS.6
MF.3 MW.3 MS.3 MF.4 MW.5 MS.6
Schl 3
T4.4
Stu4
MF.3 MW.3 MS.3 MF.4 MW.5 MS.6

Fitting models

First, look at the data

```
library(lme4)

Penicillin <- Penicillin %>%
  arrange(sample)

head(Penicillin)
```

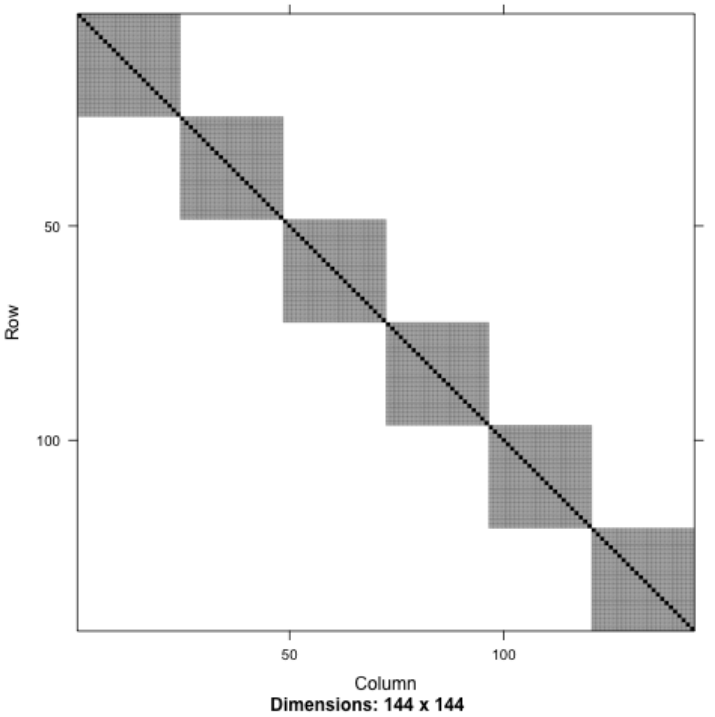
```
##   diameter plate sample
## 1       27     a      A
## 2       27     b      A
## 3       25     c      A
## 4       26     d      A
## 5       25     e      A
## 6       24     f      A
```

Nested model

```
nested1 <- lmer(diameter ~ 1 + (1 | sample),
                data = Penicillin)
```

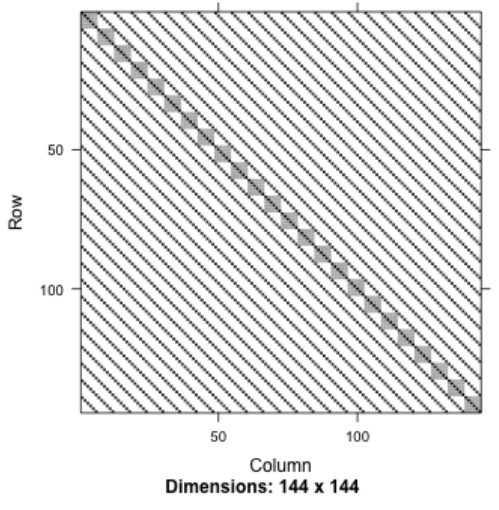
Nested image

```
library(sundry)
pull_residual_vcov(nested1) %>%
  image()
```



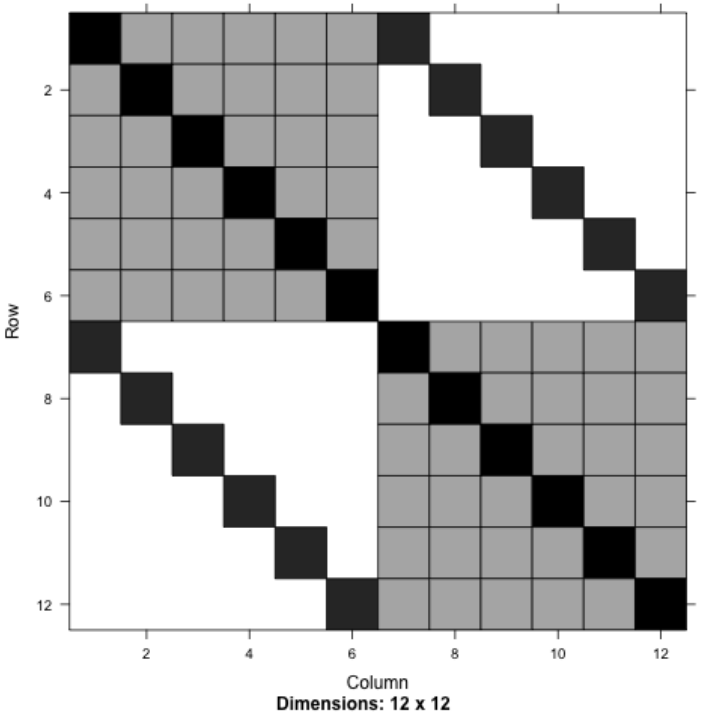
Crossed model

```
Penicillin <- Penicillin %>%
  arrange(plate)

crossed1 <- lmer(diameter ~ 1 + (1 | sample) + (1|plate),
                 data = Penicillin)

pull_residual_vcov(crossed1) %>%
  image()
```



Closer look

```
pull_residual_vcov(crossed1)[1:12, 1:12] %>%
  image()
```



Second example

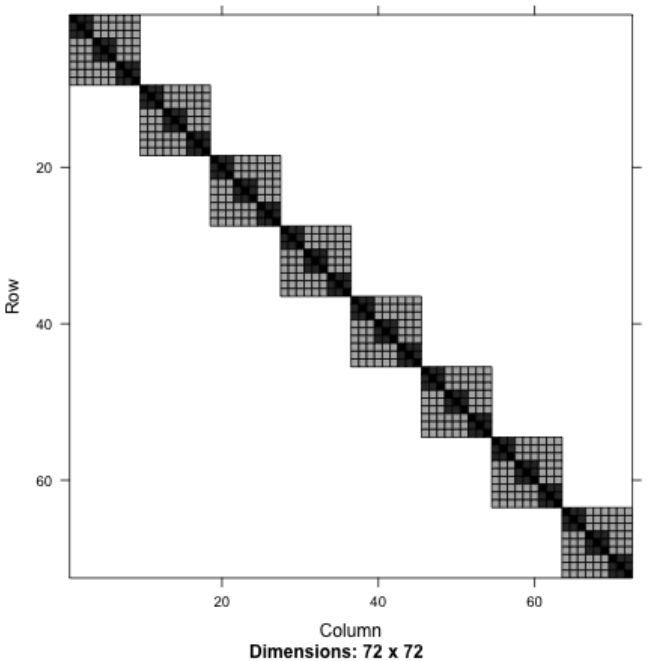
```
data(Oxide, package = "nlme")
head(Oxide)
```

```
## Grouped Data: Thickness ~ 1 | Lot/Wafer
##   Source Lot Wafer Site Thickness
## 1      1   1     1    1      2006
## 2      1   1     1    2      1999
## 3      1   1     1    3      2007
## 4      1   1     2    1      1980
## 5      1   1     2    2      1988
## 6      1   1     2    3      1982
```

Nested

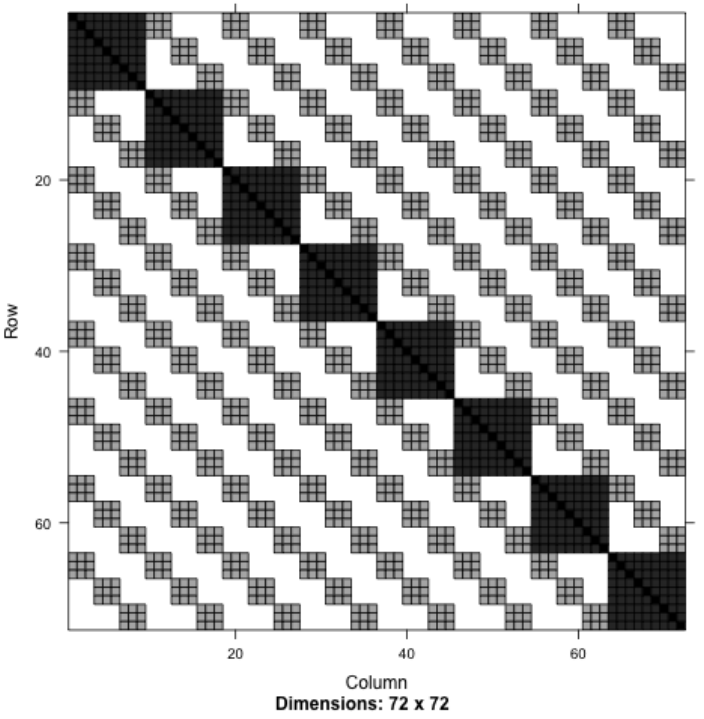
This data actually are nested

```
nested2 <- lmer(Thickness ~ 1 + (1|Lot/Wafer), data = Oxide)
pull_residual_vcov(nested2) %>%
  image()
```



Errant crossing

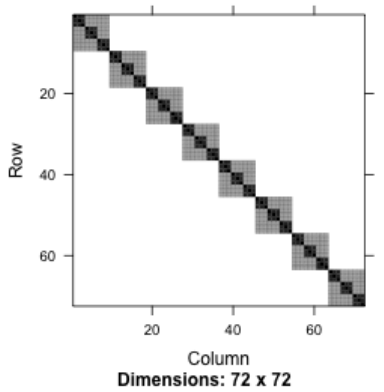
```
crossed2 <- lmer(Thickness ~ 1 + (1|Lot) + (1|Wafer),
                 data = Oxide)
pull_residual_vcov(crossed2) %>%
  image()
```



Fix

I tend to like to fix the *data*, so the model is the same regardless of the model syntax.

```
Oxide <- Oxide %>%
  mutate(Wafer = paste0(Lot, Wafer))

nested3 <- lmer(Thickness ~ 1 + (1|Lot) + (1|Wafer),
                data = Oxide)
pull_residual_vcov(nested3) %>%
  image()
```



Compare

Nested syntax

```
summary(nested2)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Thickness ~ 1 + (1 | Lot/Wafer)
##    Data: Oxide
##
## REML criterion at convergence: 454
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.8746 -0.4991  0.1047  0.5510  1.7922
##
## Random effects:
##  Groups    Name        Variance Std.Dev.
##  Wafer:Lot (Intercept)  35.87    5.989
##  Lot       (Intercept) 129.91   11.398
##  Residual               12.57    3.545
## Number of obs: 72, groups:  Wafer:Lot, 24; Lot, 8
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept) 2000.153      4.232   472.6
```

Compare

Explicit IDs

```
summary(nested3)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Thickness ~ 1 + (1 | Lot) + (1 | Wafer)
##    Data: Oxide
##
## REML criterion at convergence: 454
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.8746 -0.4991  0.1047  0.5510  1.7922
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Wafer    (Intercept)  35.87    5.989
##  Lot      (Intercept) 129.91   11.398
##  Residual              12.57    3.545
## Number of obs: 72, groups:  Wafer, 24; Lot, 8
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept) 2000.153      4.232   472.6
```

Switching schools

See this paper for more complete information

The data

Primary and secondary schools

```
achievement <- read_csv(here::here("data", "pupils.csv"))
achievement
```

```
## # A tibble: 1,000 x 8
##    PUPIL primary_school_id secondary_school_id achievement sex    ses
##    <dbl>             <dbl>                <dbl>       <dbl> <chr>  <chr>
##  1     1                 1                    2         6.6 female highes
##  2     2                 1                    1         5.7 male   lowest
##  3     3                 1                   17         4.5 male   2
##  4     4                 1                    3         4.4 male   2
##  5     5                 1                    4         5.8 male   3
##  6     6                 1                    4         5   female 4
##  7     7                 1                    4         4.9 male   3
##  8     8                 1                   17         5.3 female 2
##  9     9                 1                   14         9   male   highes
## 10    10                 1                   22         5.4 male   lowest
## # … with 990 more rows
```

About the data

Students transitioned from primary to secondary schools

We don't know how long they spent in each

It's logical to assume that scores would vary across both primary and secondary schools

How do we model this?

Cross–classified model!

```
ccrem <- lmer(achievement ~ 1 +
                (1|primary_school_id) + (1|secondary_school_id),
              data = achievement)

arm::display(ccrem)
```

```
## lmer(formula = achievement ~ 1 + (1 | primary_school_id) + (1 |
##     secondary_school_id), data = achievement)
## coef.est  coef.se
##     6.35     0.08
##
## Error terms:
##  Groups               Name        Std.Dev.
##  primary_school_id    (Intercept) 0.41
##  secondary_school_id (Intercept) 0.26
##  Residual                         0.72
## ---
## number of obs: 1000, groups: primary_school_id, 50; secondary_school_id,
## AIC = 2329.1, DIC = 2314.6
## deviance = 2317.8
```

Random effects

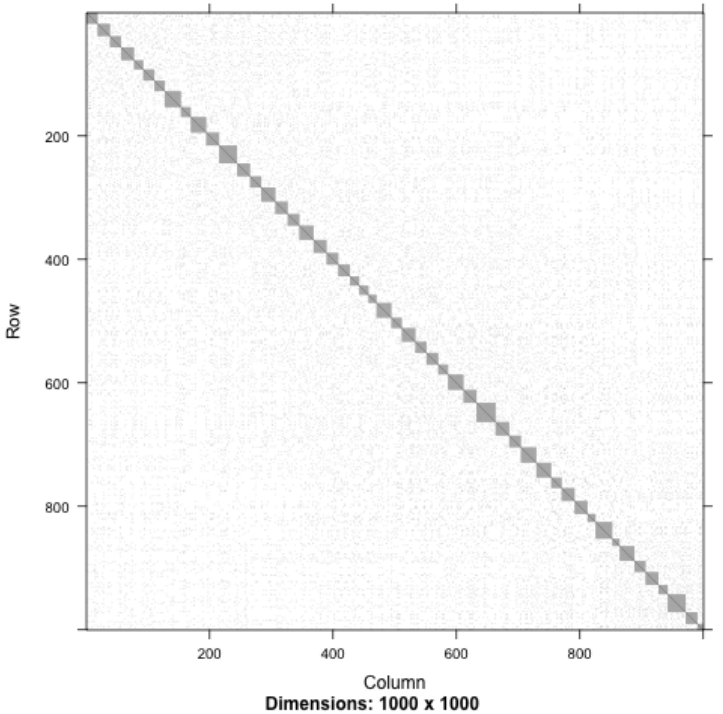
Note that each random effect would be added to each person's score. This is a little complicated.

```
str(ranef(ccrem))
```

```
## List of 2
##  $ primary_school_id  :'data.frame':    50 obs. of  1 variable:
##   ..$ (Intercept): num [1:50] -0.38 0.209 0.528 0.523 0.312 ...
##   ..- attr(*, "postVar")= num [1, 1, 1:50] 0.0264 0.0241 0.026 0.023 0.0
##  $ secondary_school_id:'data.frame':    30 obs. of  1 variable:
##   ..$ (Intercept): num [1:30] -0.3854 0.0802 -0.0141 -0.1074 0.1071 ...
##   ..- attr(*, "postVar")= num [1, 1, 1:30] 0.0165 0.0148 0.0169 0.0139 0
##  - attr(*, "class")= chr "ranef.mer"
```

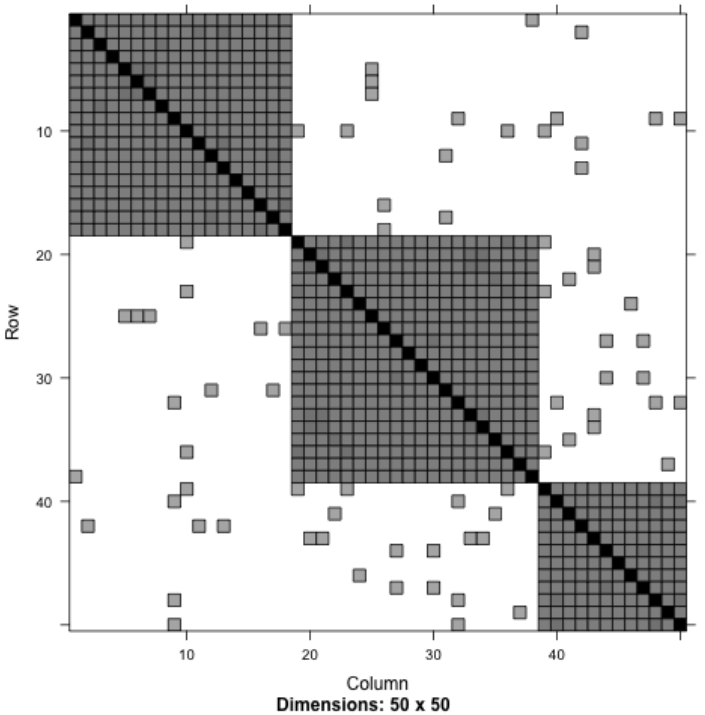
Image

```
pull_residual_vcov(ccrem) %>%
  image()
```



Zoomed in

```
pull_residual_vcov(ccrem)[1:50, 1:50] %>%
  image()
```



Multiple membership models

MM alternative

If we use **{brms}** instead, we can estimate a *multiple membership* model.

In this case, we would estimate a single random component for *school*, which would be a weighted component of each school the student attended

Fitting MM

```
mm_brms <- brm(
  achievement ~ 1 +
    (1 | mm(primary_school_id, secondary_school_id)),
  data = achievement
)
```

```
summary(mm_brms)
```

```
 Family: gaussian
  Links: mu = identity; sigma = identity
Formula: achievement ~ 1 + (1 | mm(primary_school_id, secondary_school_id))
   Data: achievement (Number of observations: 1000)
Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup samples = 4000

Group-Level Effects:
~mmprimary_school_idsecondary_school_id (Number of levels: 50)
              Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sd(Intercept)     0.53      0.07     0.40     0.68 1.00     1192     2145

Population-Level Effects:
          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept     6.33      0.08     6.18     6.50 1.00     1176     1819

Family Specific Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma     0.80      0.02     0.76     0.83 1.00     6783     2750

Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```

More complications

What if we knew how long they spent in each school?

Let's pretend we do?

Create new columns for the prortion of time spent in each school.

I'll just do this randomly.

Random proportion variables

```
set.seed(42)
achievement <- achievement %>%
  mutate(primary_school_time = abs(rnorm(1000)),
         secondary_school_time = abs(rnorm(1000))) %>%
  rowwise() %>%
  mutate(tot = primary_school_time + secondary_school_time,
         primary_school_time = primary_school_time / tot,
         secondary_school_time = secondary_school_time / tot) %>%
  ungroup()

achievement %>%
  select(PUPIL, ends_with("time"))
```

```
## # A tibble: 1,000 x 3
##    PUPIL primary_school_time secondary_school_time
##    <dbl>               <dbl>                 <dbl>
##  1     1           0.3709286             0.6290714
##  2     2           0.5186330             0.4813670
##  3     3           0.2722384             0.7277616
##  4     4           0.6266984             0.3733016
##  5     5           0.2887215             0.7112785
##  6     6           0.1508292             0.8491708
##  7     7           0.9014468             0.09855322
##  8     8           0.03131154            0.9686885
```

Include the weights

```
mm_brms2 <- brm(
  achievement ~ 1 +
    (1 | mm(primary_school_id, secondary_school_id,
            weights = cbind(primary_school_time,
                            secondary_school_time))),
  data = achievement)
```

```
summary(mm_brms2)
```

```
 Family: gaussian
  Links: mu = identity; sigma = identity
Formula: achievement ~ 1 + (1 | mm(primary_school_id, secondary_school_id, weights = cbind(primary_school_time, secondary_school_time)))
   Data: achievement (Number of observations: 1000)
Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup samples = 4000

Group-Level Effects:
~mmprimary_school_idsecondary_school_id (Number of levels: 50)
              Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sd(Intercept)     0.41      0.06     0.31     0.53 1.00     1410     2227

Population-Level Effects:
          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept     6.34      0.07     6.21     6.47 1.00     1999     2698

Family Specific Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma     0.81      0.02     0.77     0.85 1.00     7690     3027

Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```

One PL IRT model

More info

We won't really be able to get too far. Here's some additional resources

- Paper on using **{lme4}** to fit a multilevel Rasch model
- Paper on the equivalence of multilevel logistic regression and Rasch models.
- Chapter contrasting some of what we'll go through here.

The idea

- Students respond to a set of items, each scored correct/incorroct (0/1)
- Each item has a different difficulty
- Each person has a different ability
- Treat items as fixed effects and people as random effects

The data

A rural subsample of 8445 women from the Bangladesh Fertility Survey of 1989

The dimension of interest is women's mobility of social freedom.

Items

Women were asked whether they could engage in the following activities alone (1 = yes, 0 = no):

- Item 1: Go to any part of the village/town/city
- Item 2: Go outside the village/town/city
- Item 3: Talk to a man you do not know
- Item 4: Go to a cinema/cultural show
- Item 5: Go shopping
- Item 6: Go to a cooperative/mothers' club/other club
- Item 7: Attend a political meeting
- Item 8: Go to a health centre/hospital

Read in the data

```
mobility <- read_csv(here::here("data", "mobility.csv"))
mobility
```

```
## # A tibble: 8,445 x 9
##       id `Item 1` `Item 2` `Item 3` `Item 4` `Item 5` `Item 6` `Item 7`
##    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
##  1     1        1        1        1        1        0        0        0
##  2     2        0        0        0        0        0        0        0
##  3     3        0        0        0        0        0        0        0
##  4     4        0        0        0        0        0        0        0
##  5     5        0        0        0        0        0        0        0
##  6     6        0        0        0        0        0        0        0
##  7     7        1        1        0        0        0        0        0
##  8     8        0        0        0        0        0        0        0
##  9     9        0        0        0        0        0        0        0
## 10    10        1        0        0        0        0        0        0
## # … with 8,435 more rows
```

Restructuring

- The format the data came in with is common for item response data.
- This won't work for `lme4`
- Move to long

```
mobility_l <- mobility %>%
  pivot_longer(-id,
               names_to = "item",
               values_to = "response")
mobility_l
```

```
## # A tibble: 67,560 x 3
##       id item   response
##    <dbl> <chr>     <dbl>
##  1     1 Item 1        1
##  2     1 Item 2        1
##  3     1 Item 3        1
##  4     1 Item 4        1
##  5     1 Item 5        0
##  6     1 Item 6        0
##  7     1 Item 7        0
##  8     1 Item 8        0
##  9     2 Item 1        0
## 10     2 Item 2        0
## # … with 67,550 more rows
```

Estimate the model

```
onepl <- glmer(response ~ 0 + item + (1|id),
               data = mobility_l,
               family = binomial(link = "logit"),
               control = glmerControl(optimizer = "bobyqa"))
```

ltem estimates

```
library(ltm)
ltm_onepl <- rasch(mobility[ ,-1])
tibble(
  lme4_ests = fixef(onepl),
  ltm_ests = ltm_onepl$coefficients[ ,1]
)
```

```
## # A tibble: 8 x 2
##    lme4_ests   ltm_ests
##        <dbl>      <dbl>
## 1  2.547100   2.523787
## 2 -1.396082  -1.389602
## 3  2.082769   2.063258
## 4 -0.9773621 -0.9687520
## 5 -4.591452  -4.626544
## 6 -3.712541  -3.731430
## 7 -5.069683  -5.105589
## 8 -4.184125  -4.212841
```

Person estimates

```
ltm_theta <- factor.scores(ltm_onepl,
                           resp.patterns = mobility[ ,-1])
ltm_discrim <- unique(ltm_onepl$coefficients[ ,2])

lme4_theta <- ranef(onepl)$id

theta_compare <- tibble(
  ltm_theta = ltm_discrim*ltm_theta$score.dat$z1,
  lme4_theta = lme4_theta[ ,1]
)
theta_compare
```

```
## # A tibble: 8,445 x 2
##     ltm_theta lme4_theta
##         <dbl>      <dbl>
##  1  1.959870   1.942168
##  2 -3.364090  -3.346008
##  3 -3.364090  -3.346008
##  4 -3.364090  -3.346008
##  5 -3.364090  -3.346008
##  6 -3.364090  -3.346008
##  7 -0.4733693 -0.4751684
##  8 -3.364090  -3.346008
##  9 -3.364090  -3.346008
## 10 -1.879762  -1.878805
```

Discrimination estimate

```
unique(ltm_onepl$coefficients[ ,2])
```

```
## [1] 2.498525
```

```
VarCorr(onepl)
```

```
##  Groups Name        Std.Dev.
##  id     (Intercept) 2.4567
```

Extensions

Thinking about an IRT model through this framework allows us to extend the model easily – we could incorporate higher levels, more predictors, etc.

That's it!

Thanks so much for a great term