More Bayes

And multilevel binomial logistic regression

Daniel Anderson

Week 8

Agenda

- Missing values can be treated as unknown variables (parameters) in the model, and imputed from the posterior
- Fitting multilevel logistic regression models with **brms**
- Plotting **brms** fits

Popular data

```r
popular <- read_csv(here::here("data", "popularity.csv"))
popular
```

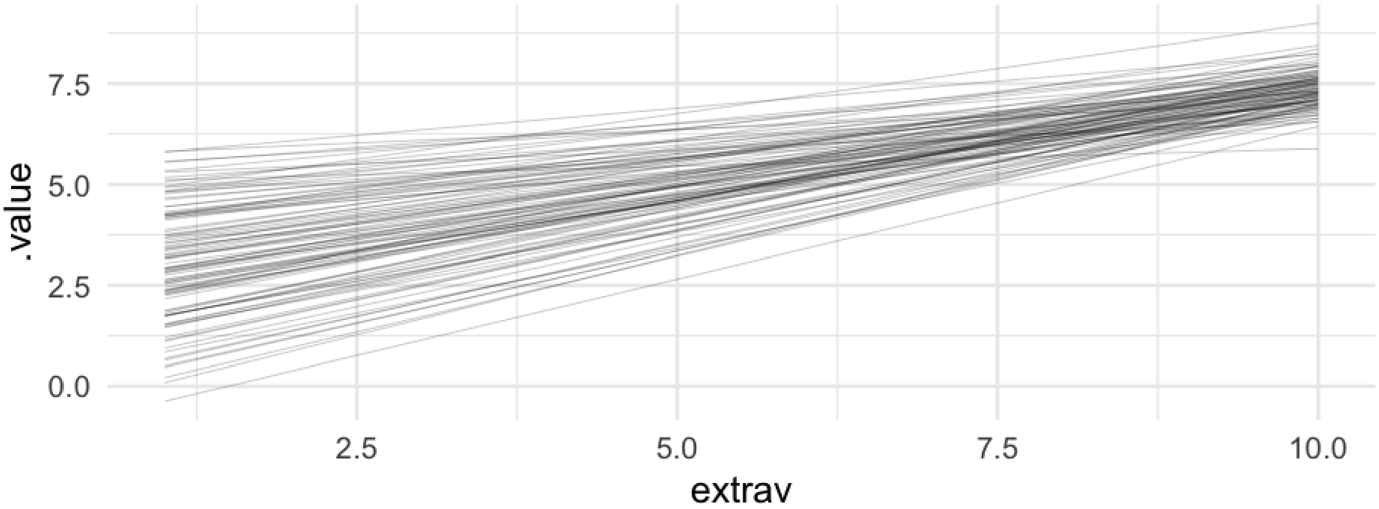
```
## # A tibble: 2,000 x 7
##    pupil class extrav sex    texp popular popteach
##    <dbl> <dbl>  <dbl> <chr> <dbl>   <dbl>    <dbl>
##  1     1     1      5 girl     24     6.3        6
##  2     2     1      7 boy      24     4.9        5
##  3     3     1      4 girl     24     5.3        6
##  4     4     1      3 girl     24     4.7        5
##  5     5     1      5 girl     24     6          6
##  6     6     1      4 boy      24     4.7        5
##  7     7     1      5 boy      24     5.9        5
##  8     8     1      4 boy      24     4.2        5
##  9     9     1      5 boy      24     5.2        5
## 10    10     1      5 boy      24     3.9        3
## # … with 1,990 more rows
```

```r
m_brms <- brm(popular ~ extrav + (extrav|class), popular)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: popular ~ extrav + (extrav | class)
##    Data: popular (Number of observations: 2000)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Group-Level Effects:
## ~class (Number of levels: 100)
##                         Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## sd(Intercept)               1.72      0.17     1.42     2.08 1.00     1462
## sd(extrav)                  0.16      0.03     0.11     0.22 1.00     1030
## cor(Intercept,extrav)      -0.95      0.03    -1.00    -0.89 1.01      610
##                         Tail_ESS
## sd(Intercept)               2382
## sd(extrav)                  2233
## cor(Intercept,extrav)        885
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     2.47      0.20     2.07     2.87 1.00      783     1620
## extrav        0.49      0.03     0.44     0.54 1.00     1390     2126
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.95      0.02     0.92     0.98 1.00     5563     2674
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
```

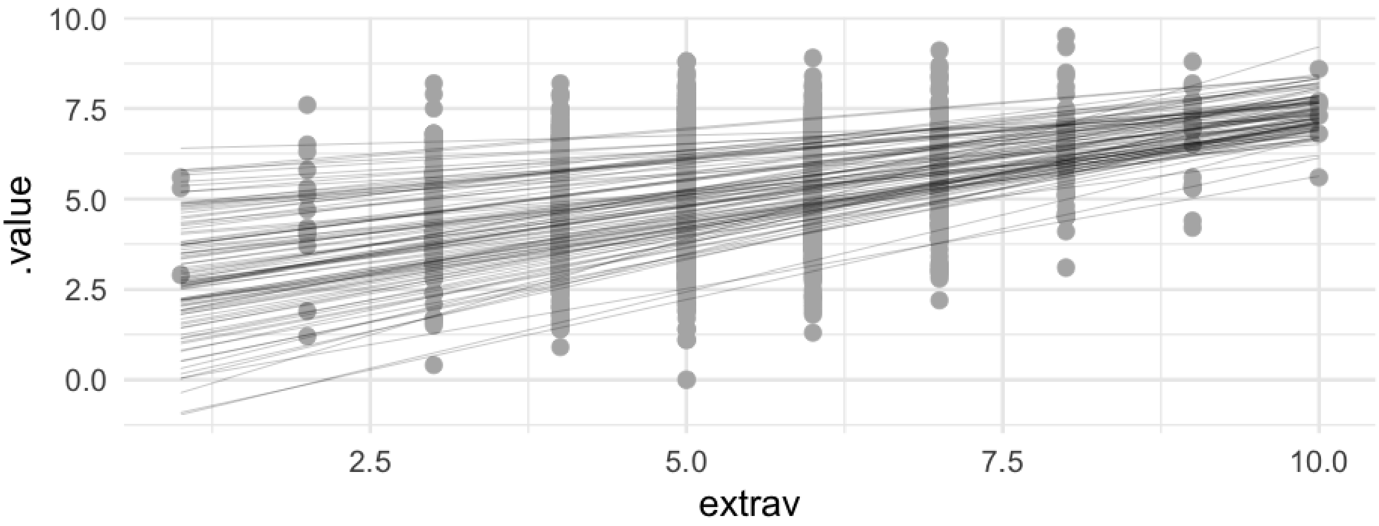
Plotting the Bayes fit

```r
tibble(extrav = 1:10, class = 0) %>%
  tidybayes::add_fitted_draws(m_brms,
                              allow_new_levels = TRUE,
                              n = 100) %>%
  ggplot(aes(extrav, .value)) +
  geom_line(aes(group = .draw), size = 0.1)
```



Add in raw data

```r
library(tidybayes)
tibble(extrav = 1:10, class = 0) %>%
  add_fitted_draws(m_brms, allow_new_levels = TRUE, n = 100) %>%
  ggplot(aes(extrav, .value)) +
  geom_point(aes(extrav, popular), data = popular,
             color = "gray70") +
  geom_line(aes(group = .draw), size = 0.1)
```



New data

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. Allow the intercept to vary by the doctor ID. Fit the model using **brms**

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

```
##
-

\

|

/

-

\
```

03:00

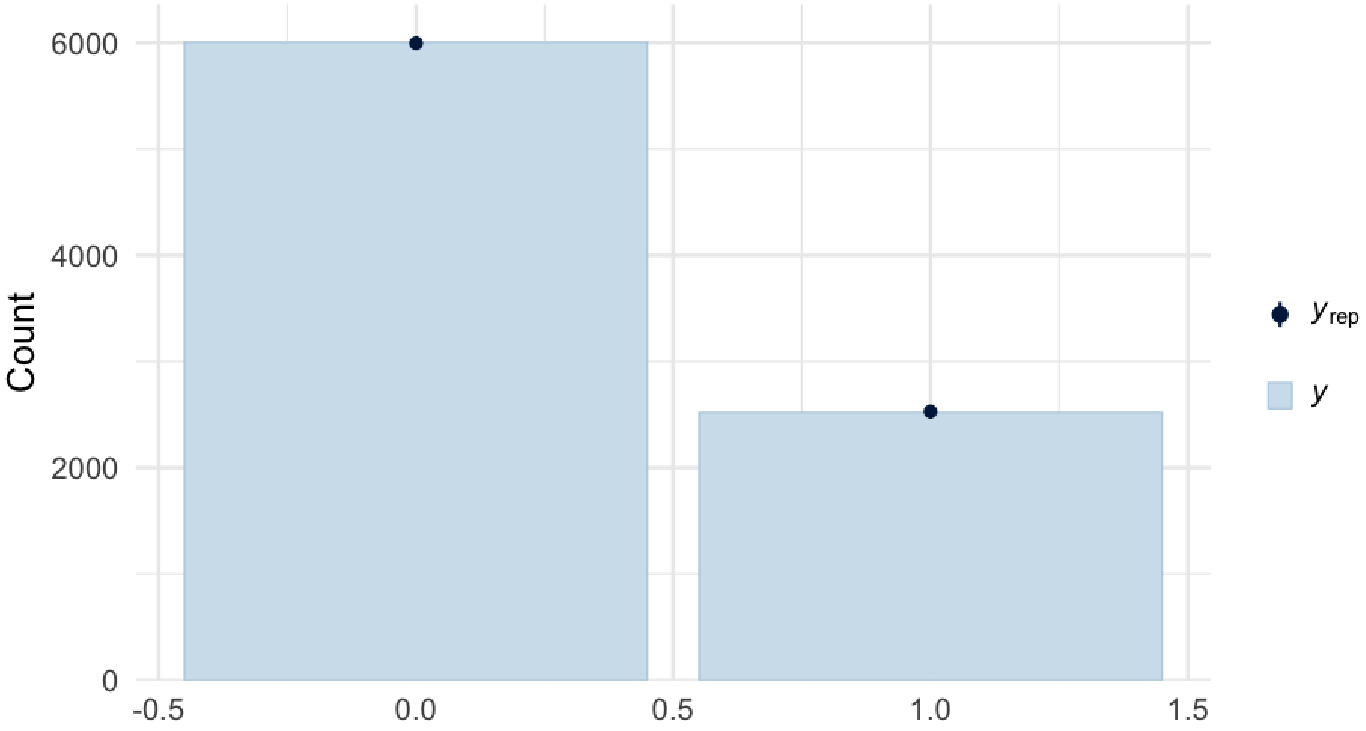
Model summary

```
summary(lc)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: remission ~ age * tumorsize + lungcapacity + (1 | did)
##    Data: hdp (Number of observations: 8525)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Group-Level Effects:
## ~did (Number of levels: 407)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ES
## sd(Intercept)     2.02      0.10     1.82     2.23 1.00      668     151
##
## Population-Level Effects:
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ES
## Intercept         1.69      1.52    -1.37     4.63 1.00     1654     273
## age              -0.05      0.03    -0.10     0.01 1.00     1710     267
## tumorsize        -0.01      0.02    -0.05     0.04 1.00     1678     264
## lungcapacity      0.07      0.19    -0.30     0.43 1.00     3560     275
## age:tumorsize    -0.00      0.00    -0.00     0.00 1.00     1689     264
##
## Samples were drawn using sample(hmc). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potenti
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Posterior predictive check

```
pp_check(lc, type = "bars")
```

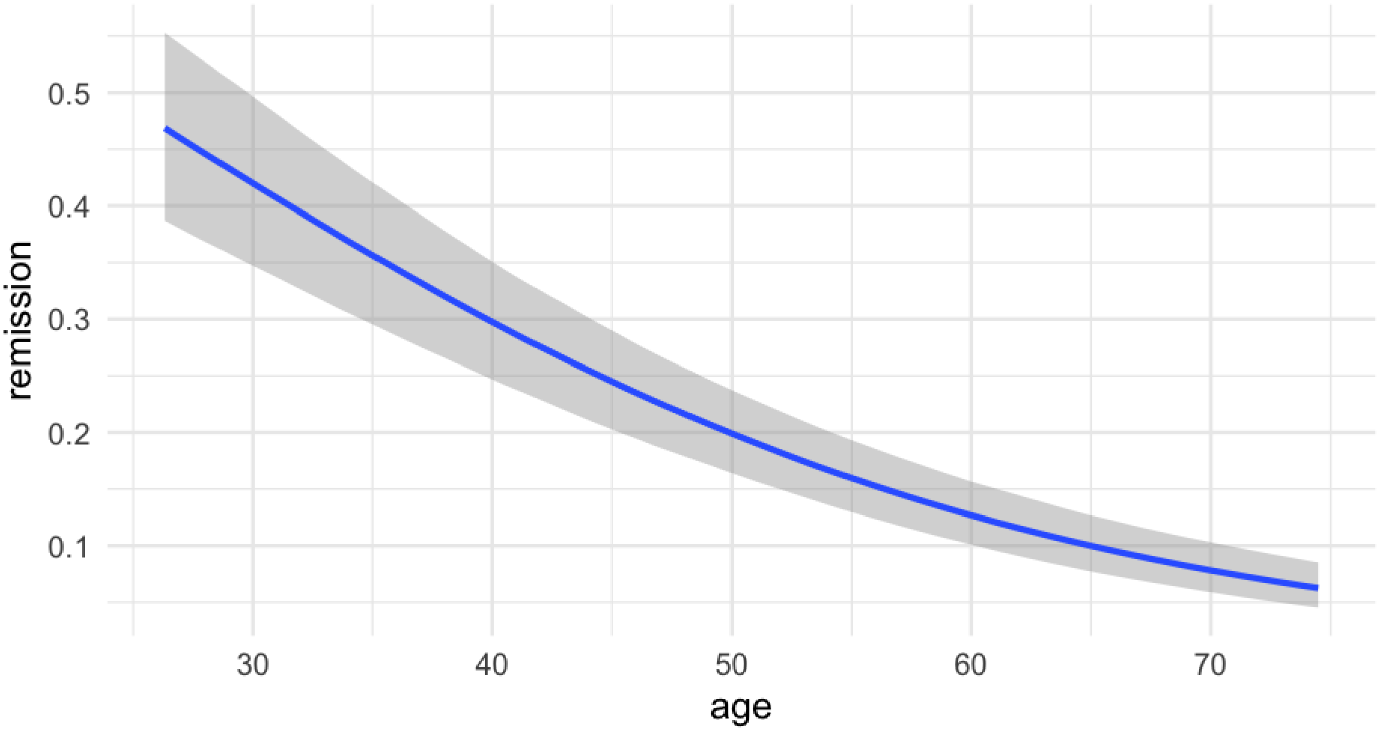


Chains

```
plot(lc)
```

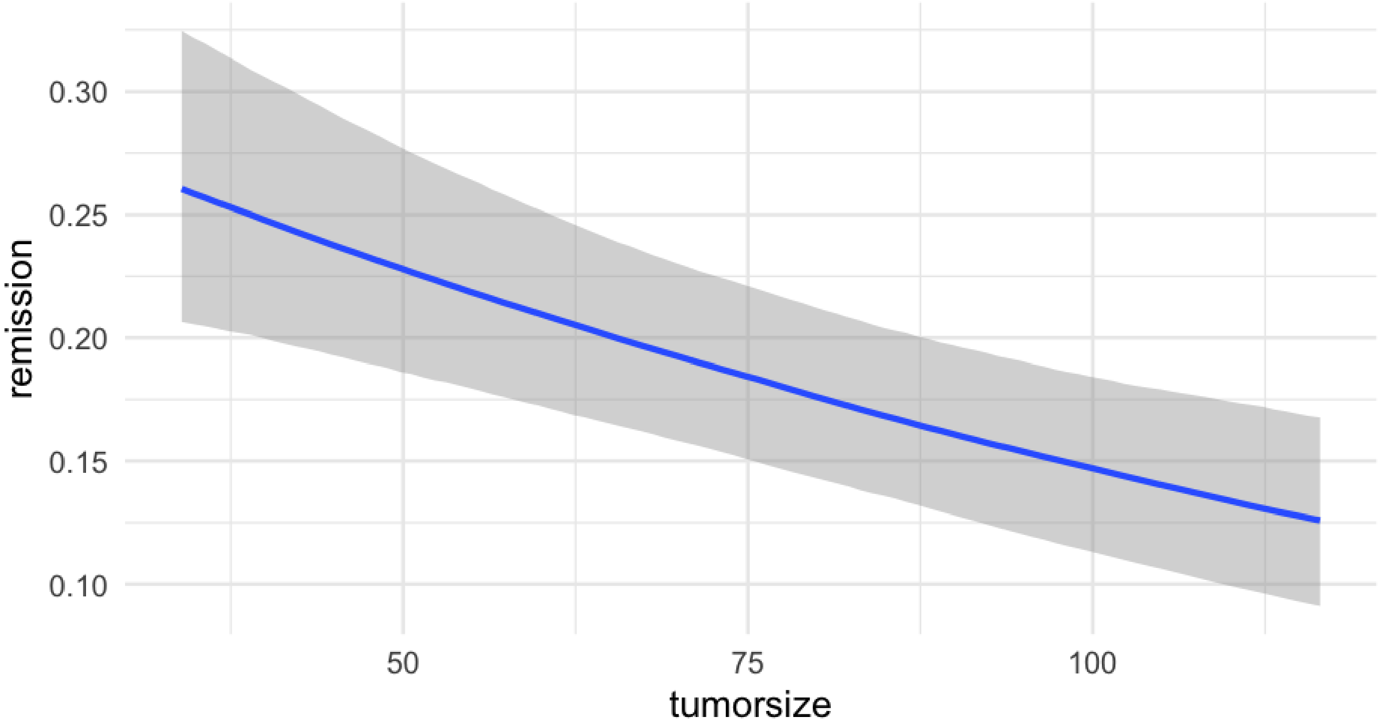
Marginal predictions: Age

```
conditional_effects(lc, "age")
```



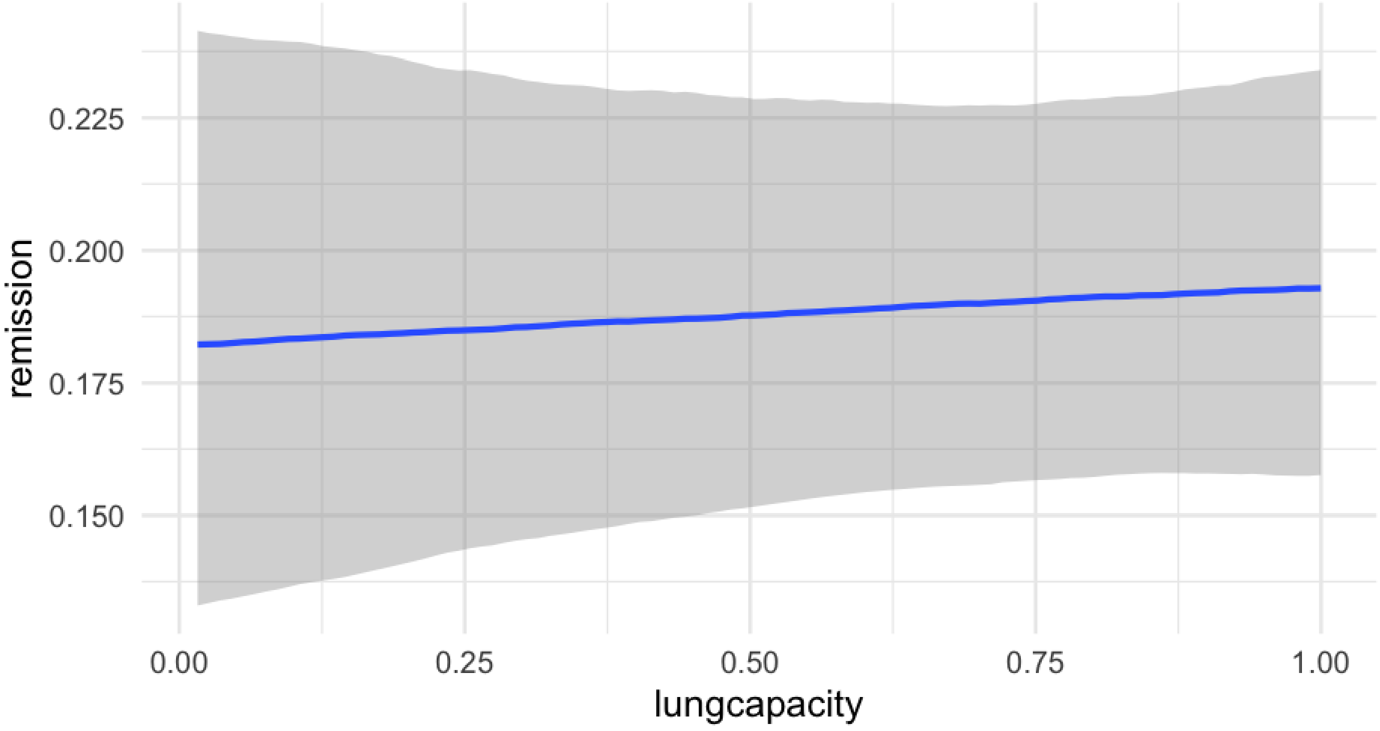
Marginal predictions: tumor size

```r
conditional_effects(lc, "tumorsize")
```



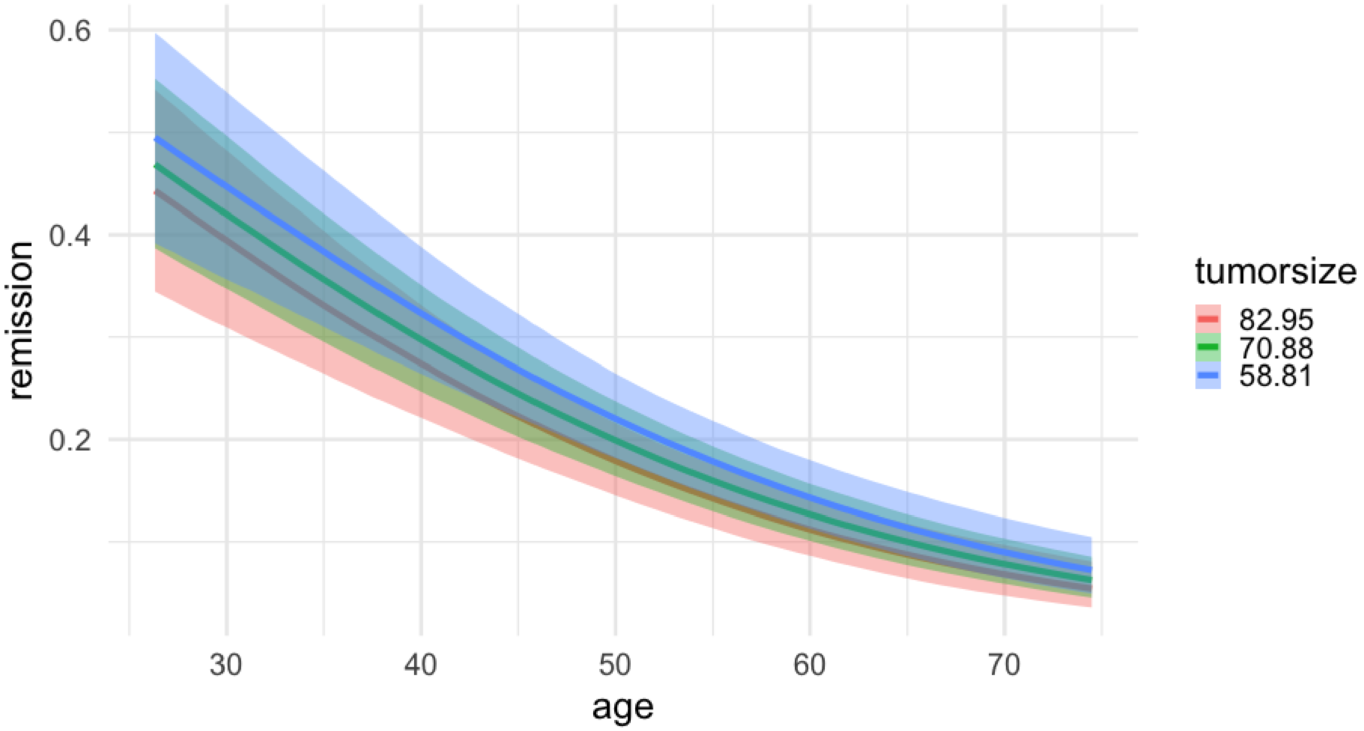
Marginal predictions: lung capacity

```
conditional_effects(lc, "lungcapacity")
```



Interaction

```
conditional_effects(lc, "age:tumorsize")
```



Make predictions

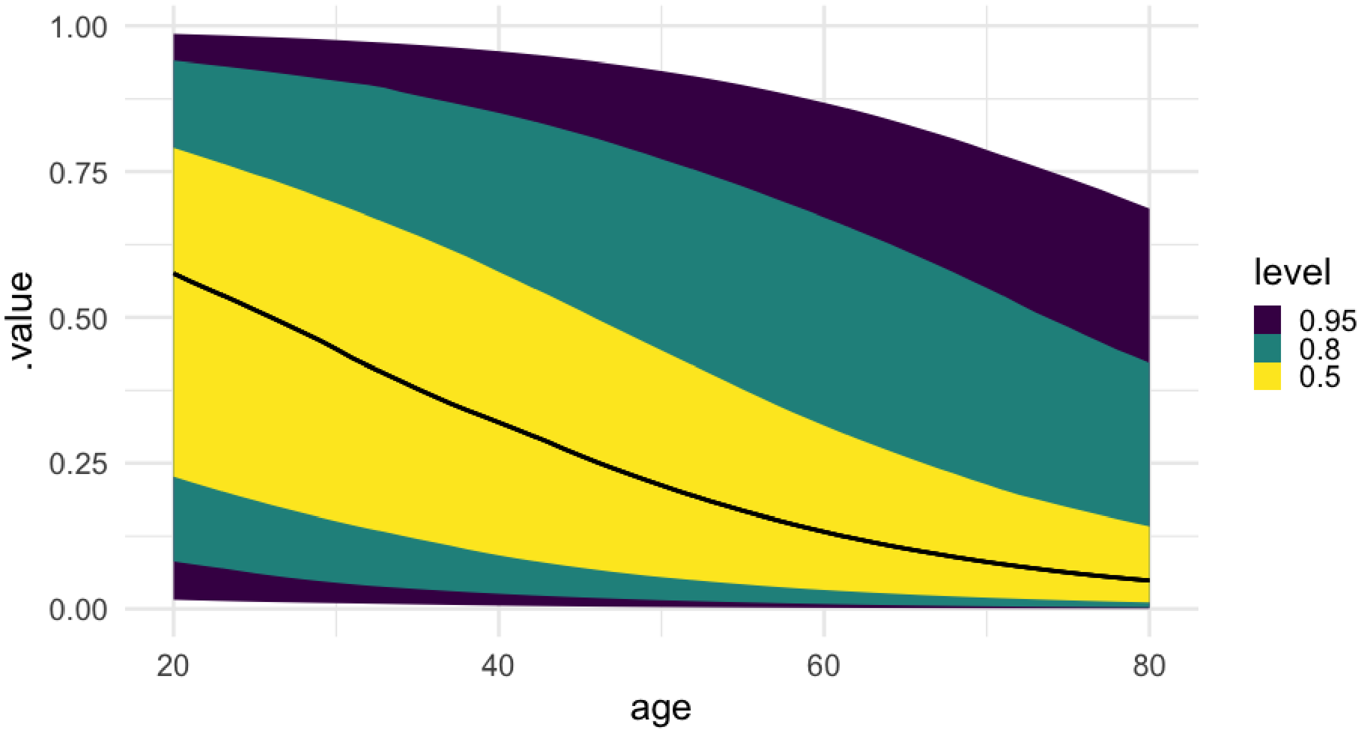
Check the relation for tumor size

```
library(tidybayes)
pred_tumor <- expand.grid(
    age = 20:80,
    lungcapacity = mean(hdp$lungcapacity),
    tumorsize = 30:120,
    did = -999
  ) %>%
  add_fitted_draws(model = lc, n = 100,
                   allow_new_levels = TRUE)
pred_tumor
```

```
## # A tibble: 555,100 x 9
## # Groups:   age, lungcapacity, tumorsize, did, .row [5,551]
##      age lungcapacity tumorsize   did  .row .chain .iteration .draw     .
##    <int>        <dbl>     <int> <dbl> <int>  <int>      <int> <int>
##  1    20    0.7740865        30  -999     1     NA         NA   295 0.68
##  2    20    0.7740865        30  -999     1     NA         NA   298 0.65
##  3    20    0.7740865        30  -999     1     NA         NA   370 0.98
##  4    20    0.7740865        30  -999     1     NA         NA   435 0.56
##  5    20    0.7740865        30  -999     1     NA         NA   472 0.76
##  6    20    0.7740865        30  -999     1     NA         NA   481 0.54
```

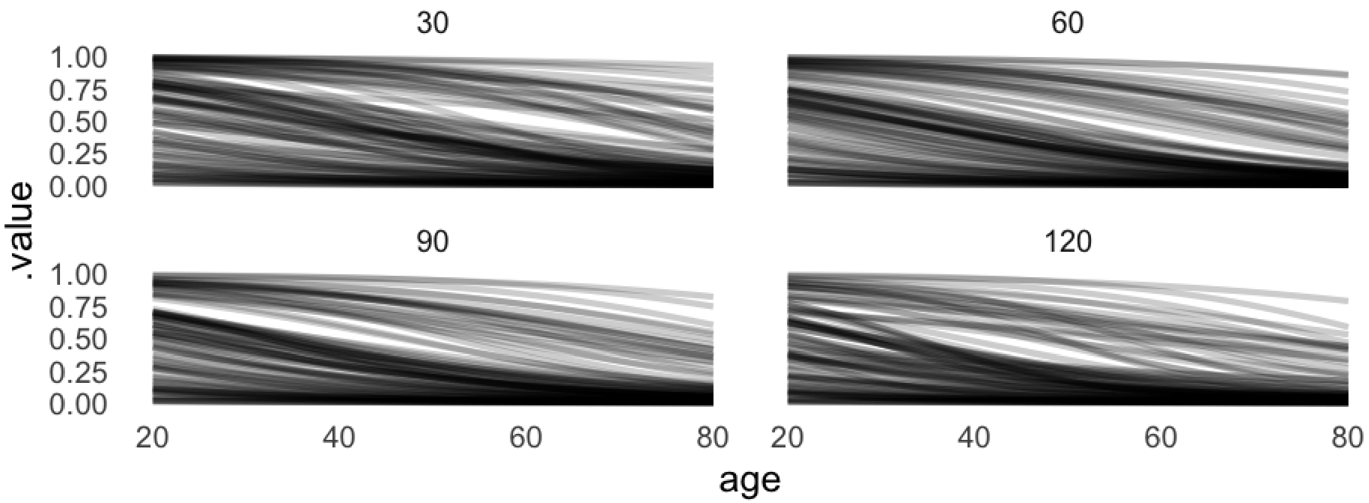
Plot

```
ggplot(pred_tumor, aes(age, .value)) +
  stat_lineribbon()
```



Different plot

```r
pred_tumor %>%
  filter(tumorsize %in% c(30, 60, 90, 120)) %>%
ggplot(aes(age, .value)) +
  geom_line(aes(group = .draw), alpha = 0.2) +
  facet_wrap(~tumorsize) +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
```



Variance by Doctor

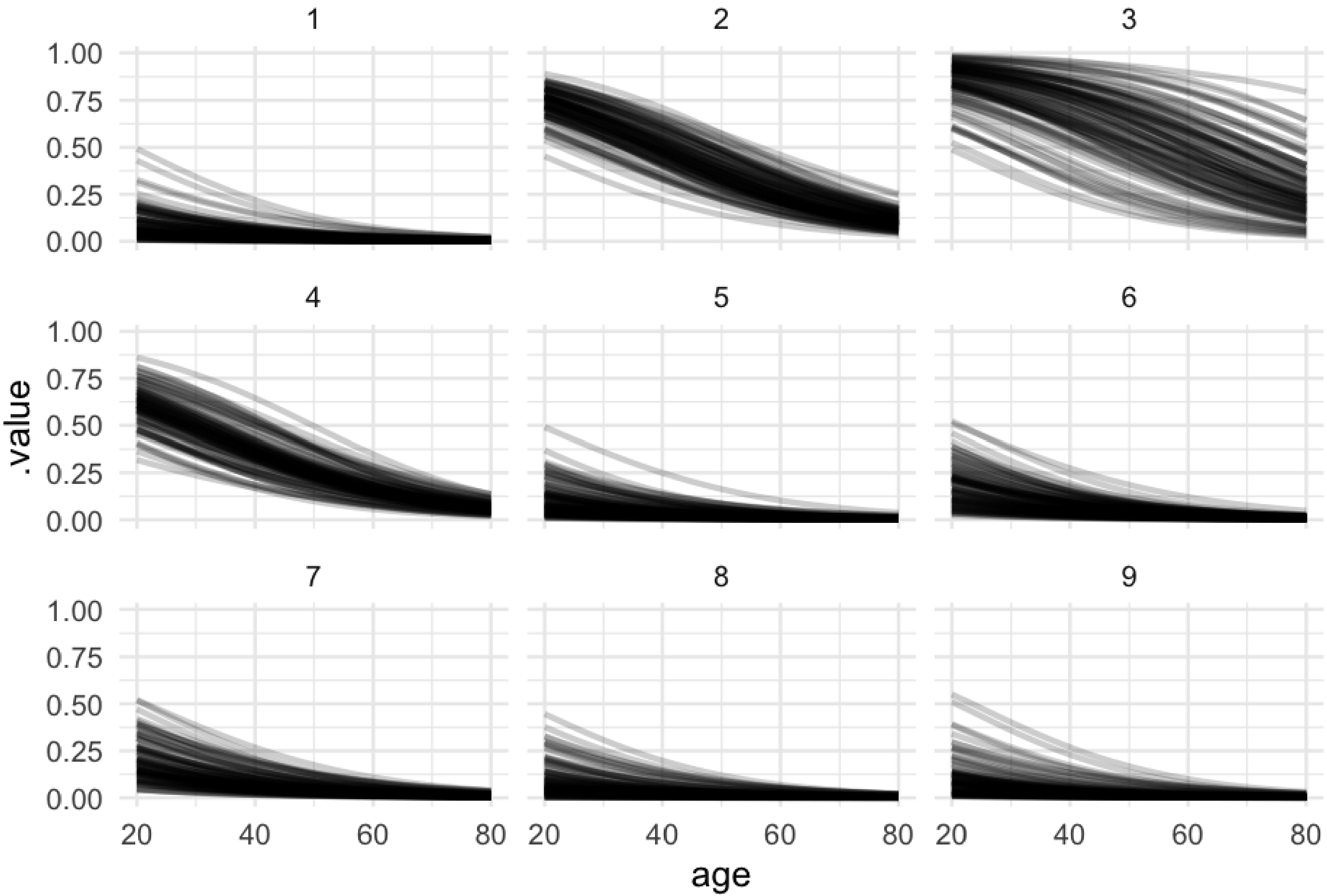
Let's look at the relation between age and proability of remission for each of the first nine doctors.

```
pred_age_doctor <- expand.grid(
    did = unique(hdp$did)[1:9],
    age = 20:80,
    tumorsize = mean(hdp$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
##    <dbl> <int>     <dbl>        <dbl> <int>  <int>      <int> <int>
##  1     1    20  70.88067    0.7740865     1     NA         NA    31
##  2     1    20  70.88067    0.7740865     1     NA         NA    39
##  3     1    20  70.88067    0.7740865     1     NA         NA    48
##  4     1    20  70.88067    0.7740865     1     NA         NA    96
##  5     1    20  70.88067    0.7740865     1     NA         NA   100
##  6     1    20  70.88067    0.7740865     1     NA         NA   113
##  7     1    20  70.88067    0.7740865     1     NA         NA   117
##  8     1    20  70.88067    0.7740865     1     NA         NA   158
##  9     1    20  70.88067    0.7740865     1     NA         NA   214
## 10     1    20  70.88067    0.7740865     1     NA         NA   311
## # … with 54,890 more rows, and 1 more variable: .value <dbl>
```

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



Going further

- We can pull lots of different things from our model
- Let's start by looking at what's actually in the model

In this case `r_*` implies "random". These are the deviations from the average.

```
get_variables(lc)
```

```
##   [1] "b_Intercept"          "b_age"                "b_tumorsize"
##   [4] "b_lungcapacity"       "b_age:tumorsize"      "sd_did__Intercept"
##   [7] "Intercept"            "r_did[1,Intercept]"   "r_did[2,Intercept]"
##  [10] "r_did[3,Intercept]"   "r_did[4,Intercept]"   "r_did[5,Intercept]"
##  [13] "r_did[6,Intercept]"   "r_did[7,Intercept]"   "r_did[8,Intercept]"
##  [16] "r_did[9,Intercept]"   "r_did[10,Intercept]"  "r_did[11,Intercept]
##  [19] "r_did[12,Intercept]"  "r_did[13,Intercept]"  "r_did[14,Intercept]
##  [22] "r_did[15,Intercept]"  "r_did[16,Intercept]"  "r_did[17,Intercept]
##  [25] "r_did[18,Intercept]"  "r_did[19,Intercept]"  "r_did[20,Intercept]
##  [28] "r_did[21,Intercept]"  "r_did[22,Intercept]"  "r_did[23,Intercept]
##  [31] "r_did[24,Intercept]"  "r_did[25,Intercept]"  "r_did[26,Intercept]
##  [34] "r_did[27,Intercept]"  "r_did[28,Intercept]"  "r_did[29,Intercept]
##  [37] "r_did[30,Intercept]"  "r_did[31,Intercept]"  "r_did[32,Intercept]
##  [40] "r_did[33,Intercept]"  "r_did[34,Intercept]"  "r_did[35,Intercept]
```

Get all draws

Let's look at the intercept

```
int <- lc %>%
  spread_draws(b_Intercept)
int
```

```
## # A tibble: 4,000 x 4
##    .chain .iteration .draw b_Intercept
##     <int>      <int> <int>       <dbl>
##  1      1          1     1    2.71656
##  2      1          2     2    4.20511
##  3      1          3     3    2.29535
##  4      1          4     4    3.07332
##  5      1          5     5    4.28472
##  6      1          6     6    2.24793
##  7      1          7     7    4.5452
##  8      1          8     8    0.796449
##  9      1          9     9    1.10624
## 10      1         10    10    1.89963
## # … with 3,990 more rows
```

Plot the distribution

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

Grab random effects

- The random effect name is `r_did`
- We use brackets to assign new names

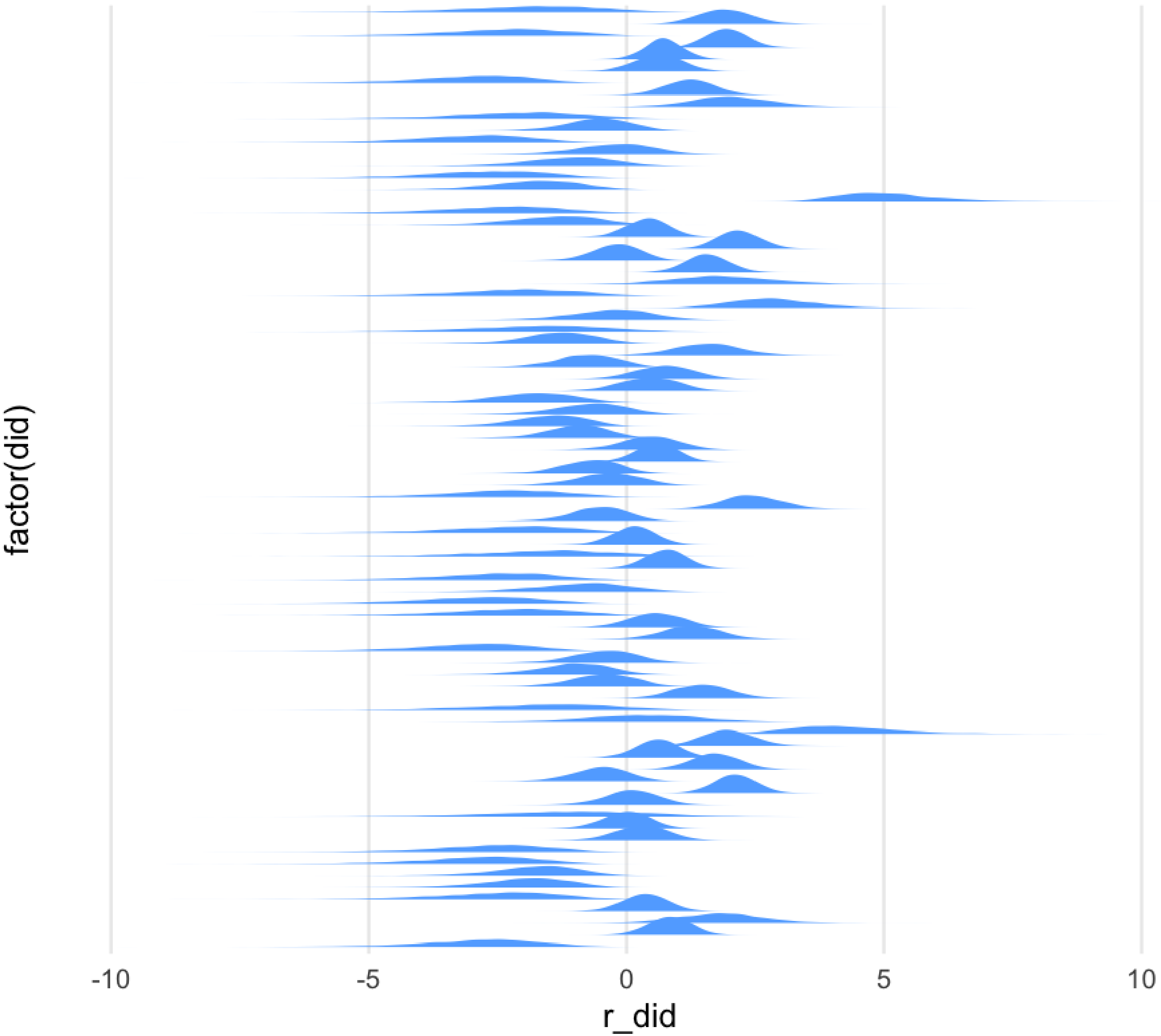
```
spread_draws(lc, r_did[did, term])
```

```
## # A tibble: 1,628,000 x 6
## # Groups:   did, term [407]
##      did term           r_did .chain .iteration .draw
##    <int> <chr>          <dbl>  <int>      <int> <int>
##  1     1 Intercept -3.0229         1          1     1
##  2     1 Intercept -1.74345        1          2     2
##  3     1 Intercept -5.47016        1          3     3
##  4     1 Intercept -0.561637       1          4     4
##  5     1 Intercept -4.68276        1          5     5
##  6     1 Intercept -1.02352        1          6     6
##  7     1 Intercept -3.5834         1          7     7
##  8     1 Intercept -2.75596        1          8     8
##  9     1 Intercept -2.28068        1          9     9
## 10     1 Intercept -2.12253        1         10    10
## # … with 1,627,990 more rows
```

Look at did distributions

First 80 doctors

```
dids <- spread_draws(lc, r_did[did, ]) # all terms, which is jus
dids %>%
  filter(did %in% 1:80) %>%
  ggplot(aes(x = r_did, y = factor(did))) +
  ggridges::geom_density_ridges(color = NA, fill = "#61adff") +
  theme(panel.grid.major.y = element_blank(),
        panel.grid.minor = element_blank(),
        axis.text.y = element_blank())
```


factor(did)
-10
-5
0
5
10
r_did

Long format

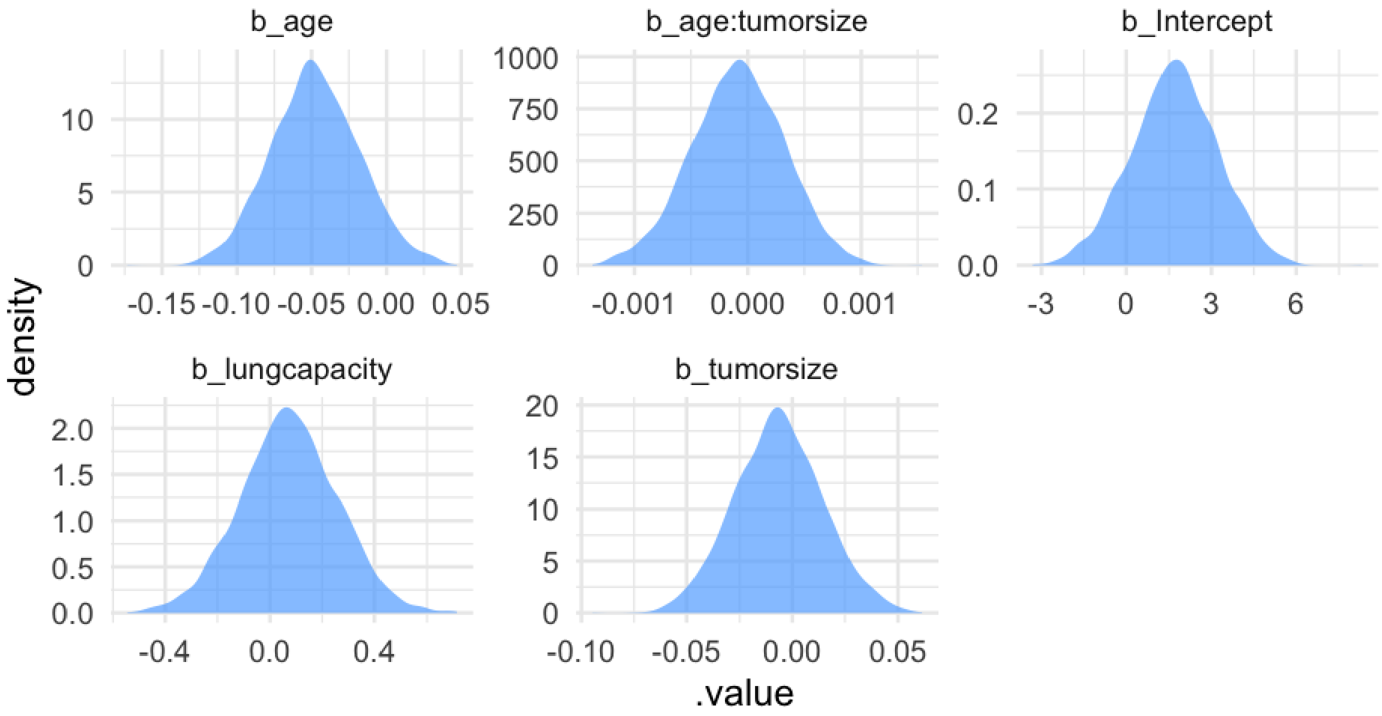
Use `gather_draws()` to return a long format, suitable for plotting (and many other things)

```
fixed_l <- lc %>%
  gather_draws(b_Intercept, b_age, b_tumorsize, b_lungcapacity,
               `b_age:tumorsize`)
fixed_l
```

```
## # A tibble: 20,000 x 5
## # Groups:   .variable [5]
##    .chain .iteration .draw .variable     .value
##     <int>      <int> <int> <chr>          <dbl>
##  1      1          1     1 b_Intercept 2.71656
##  2      1          2     2 b_Intercept 4.20511
##  3      1          3     3 b_Intercept 2.29535
##  4      1          4     4 b_Intercept 3.07332
##  5      1          5     5 b_Intercept 4.28472
##  6      1          6     6 b_Intercept 2.24793
##  7      1          7     7 b_Intercept 4.5452
##  8      1          8     8 b_Intercept 0.796449
##  9      1          9     9 b_Intercept 1.10624
## 10      1         10    10 b_Intercept 1.89963
## # … with 19,990 more rows
```

Plot the densities

```
ggplot(fixed_l, aes(.value)) +
  geom_density(fill = "#61adff", alpha = 0.7, color = NA) +
  facet_wrap(~.variable, scales = "free")
```



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 odds 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_lungcap
##     <int>      <int> <int>       <dbl>      <dbl>        <dbl>
##  1      1          1     1    2.71656  -0.0705399  -2.03108e-2      0.26
##  2      1          2     2    4.20511  -0.099448   -4.21165e-2      0.08
##  3      1          3     3    2.29535  -0.0546174  -1.32831e-2      0.27
##  4      1          4     4    3.07332  -0.0844623  -3.4302 e-2      0.32
##  5      1          5     5    4.28472  -0.0996404  -4.20281e-2      0.20
##  6      1          6     6    2.24793  -0.0580755  -1.31931e-2      0.17
##  7      1          7     7    4.5452   -0.103602   -4.22678e-2     -0.09
##  8      1          8     8    0.796449 -0.026492    1.25579e-2     -0.25
##  9      1          9     9    1.10624  -0.0412785  -3.64035e-3      0.25
## 10      1         10    10    1.89963  -0.0491901  -2.99624e-5      0.09
## # … with 3,990 more rows, and 1 more variable: b_age:tumorsize <dbl>
```

Data

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

population-level predictions

```r
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] -0.28595340 -0.32296642 -0.30876278 -0.51610507 -0.27539923
##    [6] -0.37883284 -0.26486025 -0.48828568 -0.52913791 -0.22763422
##   [11] -0.28639752 -0.34646963 -0.42096777 -0.20235639 -0.39264974
##   [16] -0.40923439 -0.66818081 -0.26361550 -0.38686267 -0.41100890
##   [21] -0.81131743 -0.39119771 -0.71284405 -0.65015638 -0.73476382
##   [26] -0.43831524 -0.37782157 -0.43385727 -0.64368528 -0.64279999
##   [31] -0.61004680 -0.54325605 -0.65055400 -0.46189436 -0.49659387
##   [36] -0.70644828 -0.80191926 -0.81185245 -0.93141718 -0.67029875
```

Plot

```r
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

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

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

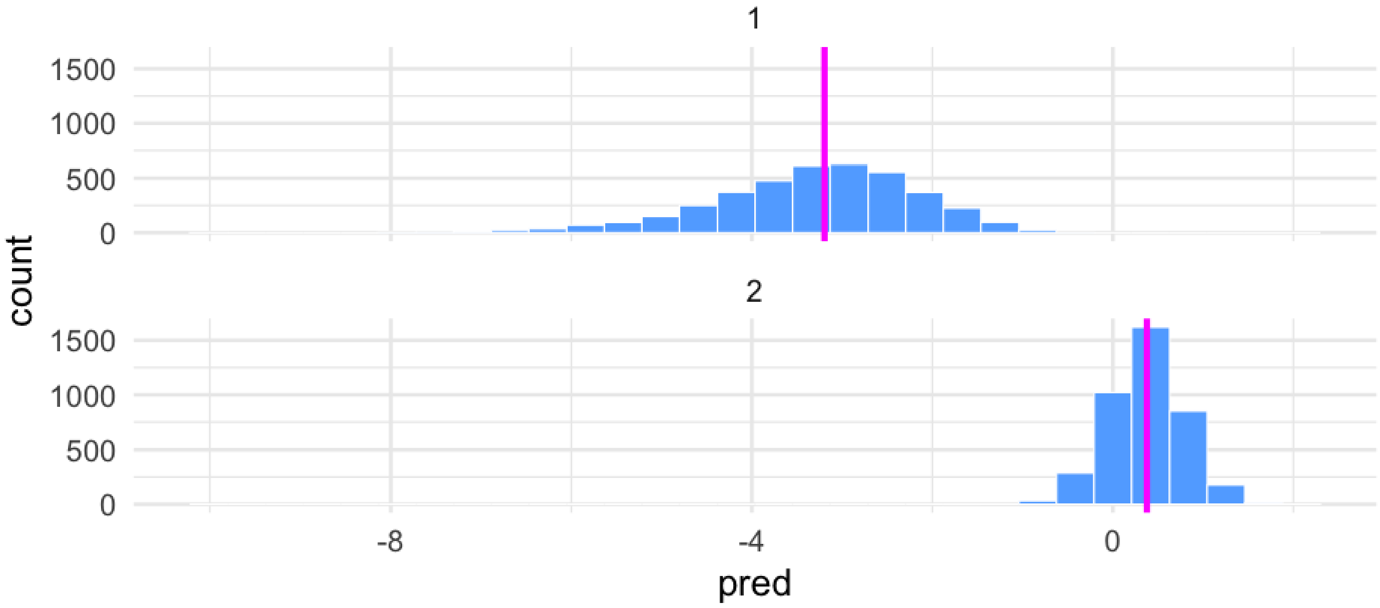
Distributions

```r
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 -3.193489
## 2     2  0.3779643
```

Plot

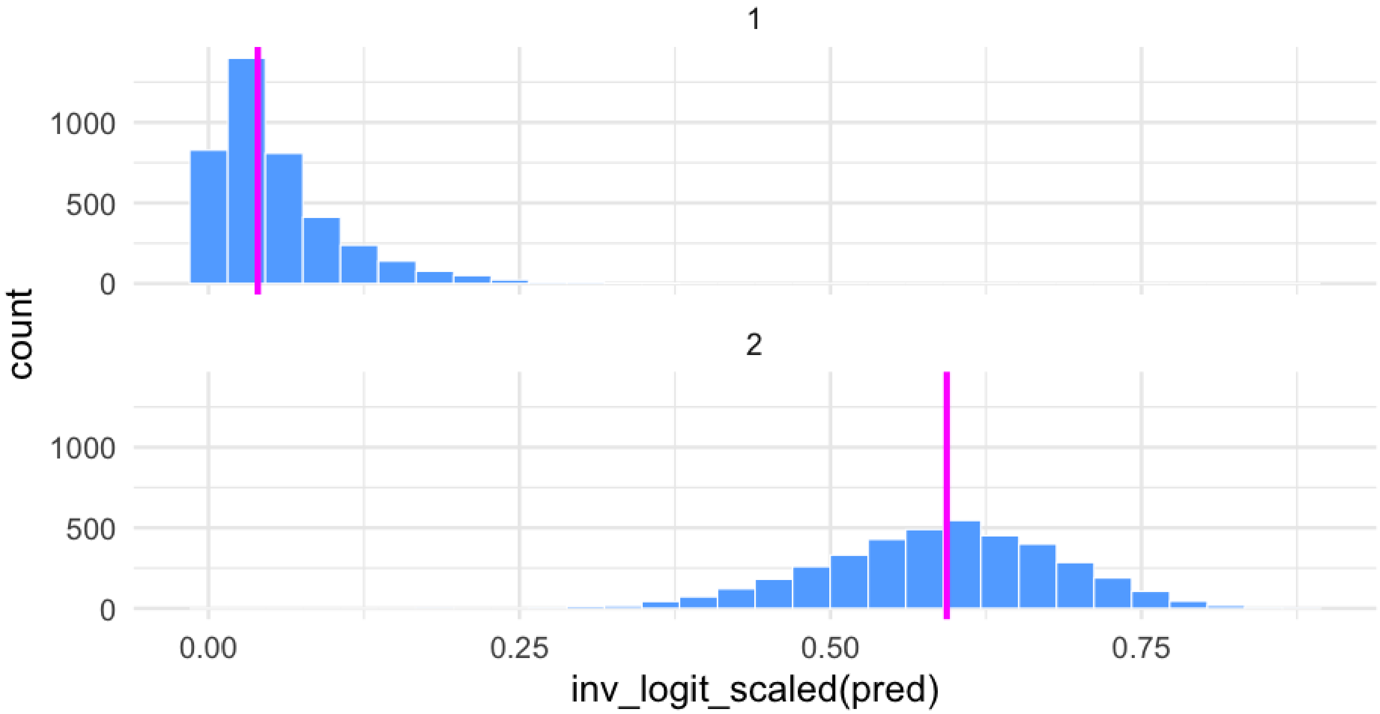
```r
ggplot(did12, aes(pred)) +
  geom_histogram(fill = "#61adff",
                 color = "white") +
  geom_vline(aes(xintercept = did_median), data = did12_medians,
             color = "magenta",
             size = 2) +
  facet_wrap(~did, ncol = 1)
```



Transform

Let's look at this again on the probability scale. Note I'm 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
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] 3.571454
```

so the patient is about 3.5 times **more likely** to have their cancer go into remission if they had did 2, instead of 1.

How confident are we in this 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 <- did12 %>%
  mutate(.chain = rep(rep(1:4, each = 1000), 2),
         .draw = rep(1:1000, 8)) %>%
  pivot_wider(names_from = "did", values_from = "pred")

did12_wider
```

```
## # A tibble: 4,000 x 4
##    .chain .draw       `1`       `2`
##     <int> <int>     <dbl>     <dbl>
##  1      1     1 -3.308853 0.4307436
##  2      1     2 -2.066416 1.014934
##  3      1     3 -5.778923 0.4383612
##  4      1     4 -1.077742 0.2885519
##  5      1     5 -4.958159 0.3001608
##  6      1     6 -1.402353 0.3330352
##  7      1     7 -3.848260 0.2469467
##  8      1     8 -3.244246 0.8343543
```

Compute difference

```
did12_wider <- did12_wider %>%
  mutate(diff = `2` - `1`)

did12_wider
```

```
## # A tibble: 4,000 x 5
##    .chain .draw       `1`       `2`     diff
##     <int> <int>     <dbl>     <dbl>    <dbl>
##  1      1     1 -3.308853 0.4307436 3.739597
##  2      1     2 -2.066416 1.014934  3.08135
##  3      1     3 -5.778923 0.4383612 6.217284
##  4      1     4 -1.077742 0.2885519 1.366294
##  5      1     5 -4.958159 0.3001608 5.25832
##  6      1     6 -1.402353 0.3330352 1.735388
##  7      1     7 -3.848260 0.2469467 4.095207
##  8      1     8 -3.244246 0.8343543 4.0786
##  9      1     9 -2.809818 0.1763041 2.986122
## 10      1    10 -2.350164 0.8357758 3.18594
## # … with 3,990 more rows
```

Summarize

```r
quantile(did12_wider$diff,
         probs = c(0.025, 0.5, 0.975))
```

```
##     2.5%      50%    97.5%
## 1.677011 3.588097 6.357494
```

Plot distribution

```r
ggplot(did12_wider, aes(diff)) +
  geom_histogram(fill = "#61adff",
                 color = "white") +
  geom_vline(aes(xintercept = median(diff)),
             color = "magenta",
             size = 2)
```

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.

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

did23 <- did12_wider %>%
  select(-`1`, -diff) %>%
  mutate(`3` = pred_did3,
         diff = `3` - `2`)
did23
```

```
## # A tibble: 4,000 x 5
##    .chain .draw       `2`        `3`       diff
##     <int> <int>     <dbl>      <dbl>      <dbl>
##  1      1     1 0.4307436 2.475167    2.044423
##  2      1     2 1.014934  1.108304    0.093370
##  3      1     3 0.4383612 1.793237    1.354876
##  4      1     4 0.2885519 0.6536849   0.3651330
##  5      1     5 0.3001608 2.383981    2.08382
##  6      1     6 0.3330352 1.562827    1.229792
##  7      1     7 0.2469467 0.6939357   0.446989
##  8      1     8 0.8343543 0.8729243   0.03857
##  9      1     9 0.1763041 0.06621309 -0.110091
## 10      1    10 0.8357758 0.7470808  -0.088695
```

Directionality

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

```
## 
## FALSE  TRUE 
## 0.142 0.858
```

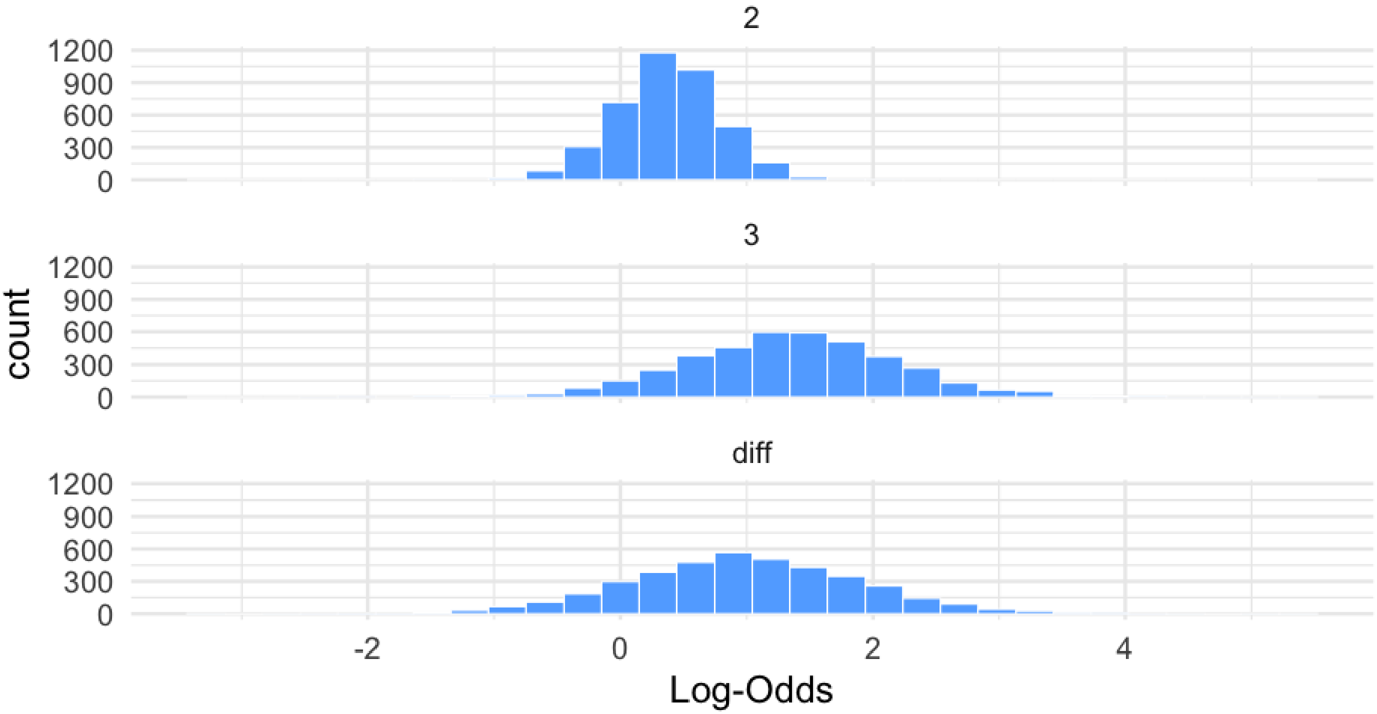
So there's roughly an 87% chance that the odds of remission are higher with with `did` 3 than 2.

Plot data

```
pd23 <- did23 %>%
  pivot_longer(`2`:diff,
               names_to = "Distribution",
               values_to = "Log-Odds")
pd23
```

```
## # A tibble: 12,000 x 4
##    .chain .draw Distribution `Log-Odds`
##     <int> <int> <chr>             <dbl>
##  1      1     1 2             0.4307436
##  2      1     1 3             2.475167
##  3      1     1 diff          2.044423
##  4      1     2 2             1.014934
##  5      1     2 3             1.108304
##  6      1     2 diff          0.093370
##  7      1     3 2             0.4383612
##  8      1     3 3             1.793237
##  9      1     3 diff          1.354876
## 10      1     4 2             0.2885519
## # … with 11,990 more rows
```

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



Next time

Growth Modeling 2

We'll continue to discuss and use Bayes