KG2

Gina Nichols

4/14/2020

Trying to use a Bayesian approach

Based on Julia Silge's blog.

Get libraries

```
library(tidyverse)
library(brms)
```

Use the mixed linear model I decided was 'best': Need to learn LaTex

```
lmer(log(totseeds_m2) ~ site_sys * cc_trt2 + (1|blockID), data = filter(dstat, totseeds_m2 < 15000))
```

I fit the model using the Bayesian approach. Seems OK.

This is what I did:

```
options(mc.cores = parallel::detectCores())

fit_bayes1 <- brm(
  log(totseeds_m2) ~ 1 +(1 | blockID) + site_sys*cc_trt2,
  data = filter(dstat, totseeds_m2 < 15000))
```

I then use the **tidybayes** package to extract the distributions

```
library(tidybayes)
res_bayes1 <- fit_bayes1 %>%
  tidybayes::gather_draws(`b_.*`, regex = TRUE) %>%
  ungroup()
```

```
res_bayes1
```

```
## # A tibble: 32,000 x 5
##    .chain .iteration .draw .variable   .value
##     <int>      <int> <int> <chr>        <dbl>
##  1      1          1     1 b_Intercept   6.26
##  2      1          2     2 b_Intercept   6.05
##  3      1          3     3 b_Intercept   5.73
##  4      1          4     4 b_Intercept   5.57
##  5      1          5     5 b_Intercept   5.92
##  6      1          6     6 b_Intercept   5.70
##  7      1          7     7 b_Intercept   6.10
##  8      1          8     8 b_Intercept   5.40
##  9      1          9     9 b_Intercept   5.95
## 10      1         10    10 b_Intercept   5.72
## # ... with 31,990 more rows
```

I tidy up the results. Seems ok.

```
library(janitor)

res_bayes2 <- res_bayes1  %>%
  mutate(.variable = str_remove_all(.variable, "b_|cc_trt2|site_sys")) %>%
  pivot_wider(names_from = .variable,
              values_from = .value) %>%
  janitor::clean_names()

res_bayes2
```

```
## # A tibble: 4,000 x 11
##    chain iteration  draw intercept boyd_silage funcke_grain stout_grain   ryecc
##    <int>     <int> <int>     <dbl>       <dbl>        <dbl>       <dbl>   <dbl>
##  1     1         1     1      6.26       1.34          2.95       0.267  -0.214
##  2     1         2     2      6.05       1.14          3.18     -0.0787  0.0336
##  3     1         3     3      5.73       2.03          3.20       0.136   0.358
##  4     1         4     4      5.57       1.64          3.71       0.943   0.884
##  5     1         5     5      5.92       2.11          2.99       0.949   0.618
##  6     1         6     6      5.70       1.55          3.43       0.879   0.0663
##  7     1         7     7      6.10       0.506         3.31       0.265   0.609
##  8     1         8     8      5.40       2.48          2.92       0.815   0.456
##  9     1         9     9      5.95       0.889         3.22       0.255   0.0791
## 10     1        10    10      5.72       1.76          2.88       0.240   0.409
## # ... with 3,990 more rows, and 3 more variables: boyd_silage_ryecc <dbl>,
## #   funcke_grain_ryecc <dbl>, stout_grain_ryecc <dbl>
```

Interpreting the different intercept/slopes. Now I struggle.

The intercept is actually *boyd_grain_none*. How do I compare, for example, *boyd_grain_none* with *boyd_grain_rye*?

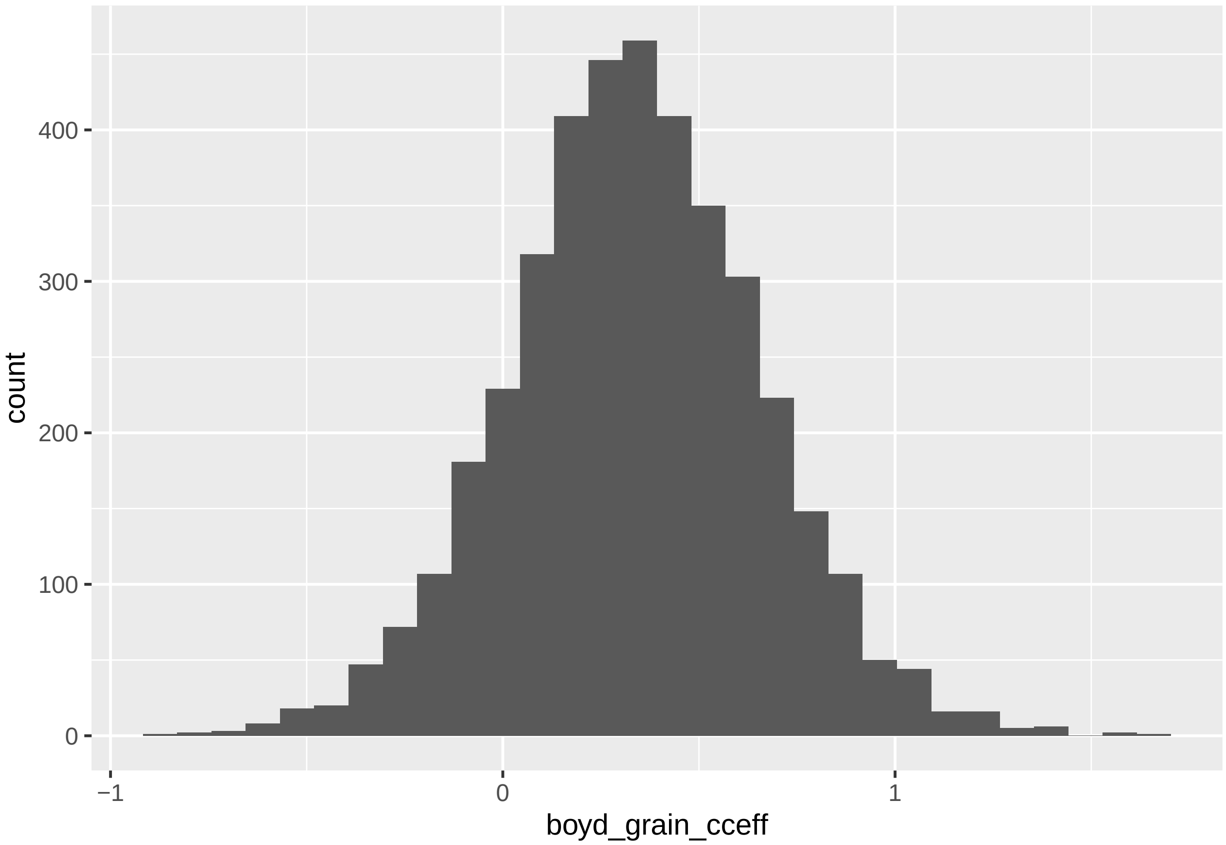
```
res_bayes2 %>%
  select(
    chain, iteration, draw,
         intercept, #--intercept is actually boyd_grain_none
         ryecc #--need to add to intercept to get boyd_grain_ryecc?
  ) %>%
  mutate(
    boyd_grain_cceff = ryecc #--this should be the effect of rye at boyd_grain?
  ) %>%
  select(chain, iteration, draw, boyd_grain_cceff) %>%
  summarise(mean = mean(boyd_grain_cceff))
```

```
## # A tibble: 1 x 1
##    mean
##   <dbl>
## 1 0.336
```

Ummm, so this is similar to the number I got from the 'back-transformed' values using **emmeans** on the **lmer** output for the same model. So maybe it's ok?

My credible interval is calculated from the distribution, seems ok.

```
res_bayes2 %>%
  select(
    chain, iteration, draw,
         intercept, #--intercept is actually boyd_grain_none
         ryecc #--need to add to intercept to get boyd_grain_ryecc?
  ) %>%
  mutate(
    boyd_grain_cceff = ryecc #--this should be the effect of rye at boyd_grain?
  ) %>%
  select(chain, iteration, draw, boyd_grain_cceff) %>%
  ggplot(aes(boyd_grain_cceff)) +
  geom_histogram()
```



PROBLEM

HOw would I get other contrasts? For example how would I calculate the effect of rye in *boyd_silage*?

```
res_bayes2 %>%
  select(
    chain, iteration, draw,
    intercept, #--intercept is actually boyd_grain_none
    boyd_silage, #--this will add the effect of boyd_silage to the boyd_grain_none, making it boyd_sila
    boyd_silage_ryecc #--this will be what I compare that to?
    ) %>%
  mutate(
```

```r
    boyd_silage_cceff = boyd_silage_ryecc - (intercept + boyd_silage) #--this should be the effect of r
  ) %>%
  select(chain, iteration, draw, boyd_silage_cceff) %>%
  summarise(mean = mean(boyd_silage_cceff))
```

```
## # A tibble: 1 x 1
##    mean
##   <dbl>
## 1 -8.56
```

This is definitely wrong. Help?