Math 459 Lecture 22

Todd Kuffner

Hierarchical Linear Models (part 1)

- Mixed effect models
- Seedling weight example
- Non-Bayesian analysis (missing pvalues/CI method)
- Bayesian analysis in Stan
- Compute posterior probabilities and CIs

Basic idea of hierarchical linear models

Let j denote group, $j = 1, \ldots, g$ and i is observation, $i = 1, \ldots, n$.

Level 1 model:

$$Y_{ij} = \alpha_j + \beta_j X_{ij} + e_{ij}$$

Level 2 model:

$$\alpha_j = \gamma_\alpha + \delta_\alpha Z_j + u_{\alpha j},$$

$$\beta_j = \gamma_\beta + \delta_\beta + Z_j + u_{\beta j}$$

Thus the coefficients (intercept and slopes) in Level 1 model are modeled as response variables in Level 2 model. Can allow for random incercepts, random slopes, etc.

Notation

Standard notation for mixed-effect models:

$$y = X\beta + Zu + e$$

where

- y is an $n \times 1$ response vector
- X is an $n \times p$ design matrix for fixed effects
- β is a $p \times 1$ unknown fixed effect parameter vector
- Z is an $n \times q$ design matrix for random effects
- u is a $q \times 1$ unknown random effect parameter vector
- e is an $n \times 1$ unknown error vector

Assumptions

$$y = X\beta + Zu + e$$

Typically assume

- $E[u] = E[e] = 0$
- $V[u] = \Omega$ and $V[e] = \Lambda$
- $Cov[u, e] = 0$

These assumptions imply

- $E[y|\beta, \Omega, \Lambda] = X\beta$
- $V[y|\beta, \Omega, \Lambda] = Z\Omega Z' + \Lambda = \Sigma_y$

Common addition assumptions

- $V[e] = \Lambda = \sigma_e^2 I$,
- $V[u] = \Omega = \text{diag}\{\sigma_{u,\cdot}^2\}$, (or $V[u] = \Omega = \sigma_u^2 I$ for single source), and
- u and e are normally distributed.

Rewrite as a standard linear regression model

We can rewrite

$$y = X\beta + Zu + e$$

as

$$y = \tilde{X}\tilde{\beta} + e$$

where $\tilde{X}$ is $n \times (p + q)$ with

$$\tilde{X} = [X \ Z]$$

and $\tilde{\beta}$ is a $(p + q) \times 1$ vector with

$$\tilde{\beta} = \begin{bmatrix} \beta \\ u \end{bmatrix}.$$

The fixed and random effects have been concatenated into the same vector.

Hierarchical linear model

Assume $y \sim N(\tilde{X}\tilde{\beta}, \Lambda)$. A Bayesian analysis proceeds by assigning prior distributions to $\tilde{\beta}$ and Λ. In constructing the prior for $\tilde{\beta}$, consider the components β and u separately. Assume

$$\beta \sim N(\beta_0, \Sigma_\beta), \quad \text{and} \quad u \sim N(0, \Omega)$$

independently.

For the

- **fixed** effects β, we select β_0 and Σ_β while for the
- **random** effects u, we assign a prior for Ω.

Therefore we have created a hierarchical model for the random effects and thus refer to this as a *hierarchical linear model.*

Summary

These models are referred to as

- mixed-effect models,
- hierarchical linear models, or
- multi-level models.

The parameters for the prior distribution for the

- fixed effects are not learned and
- random effects are learned.

This corresponds to a non-Bayesian analysis learning a variance parameter for random effects.

Seedling weight example

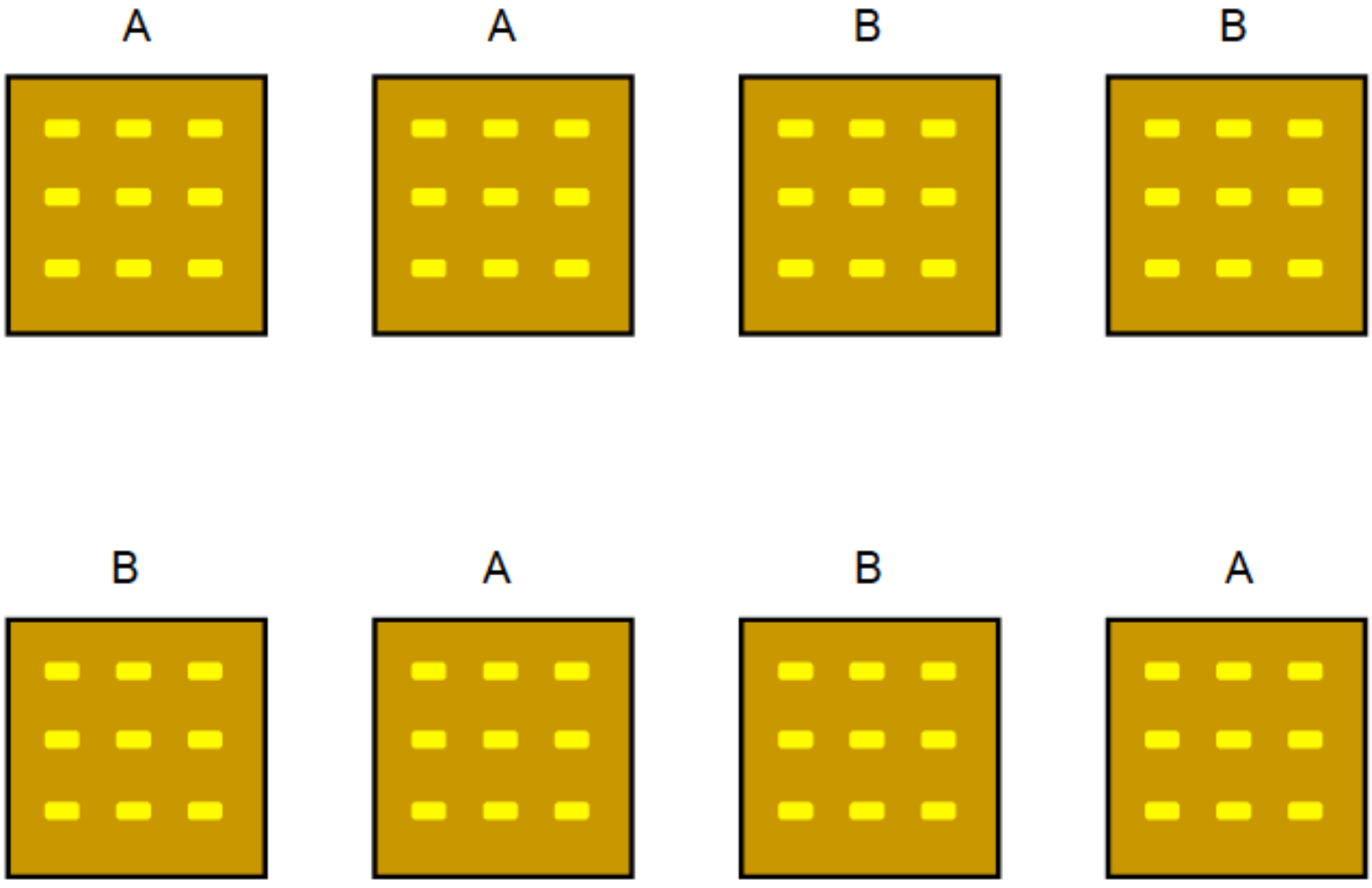
Example taken from Dan Nettleton:

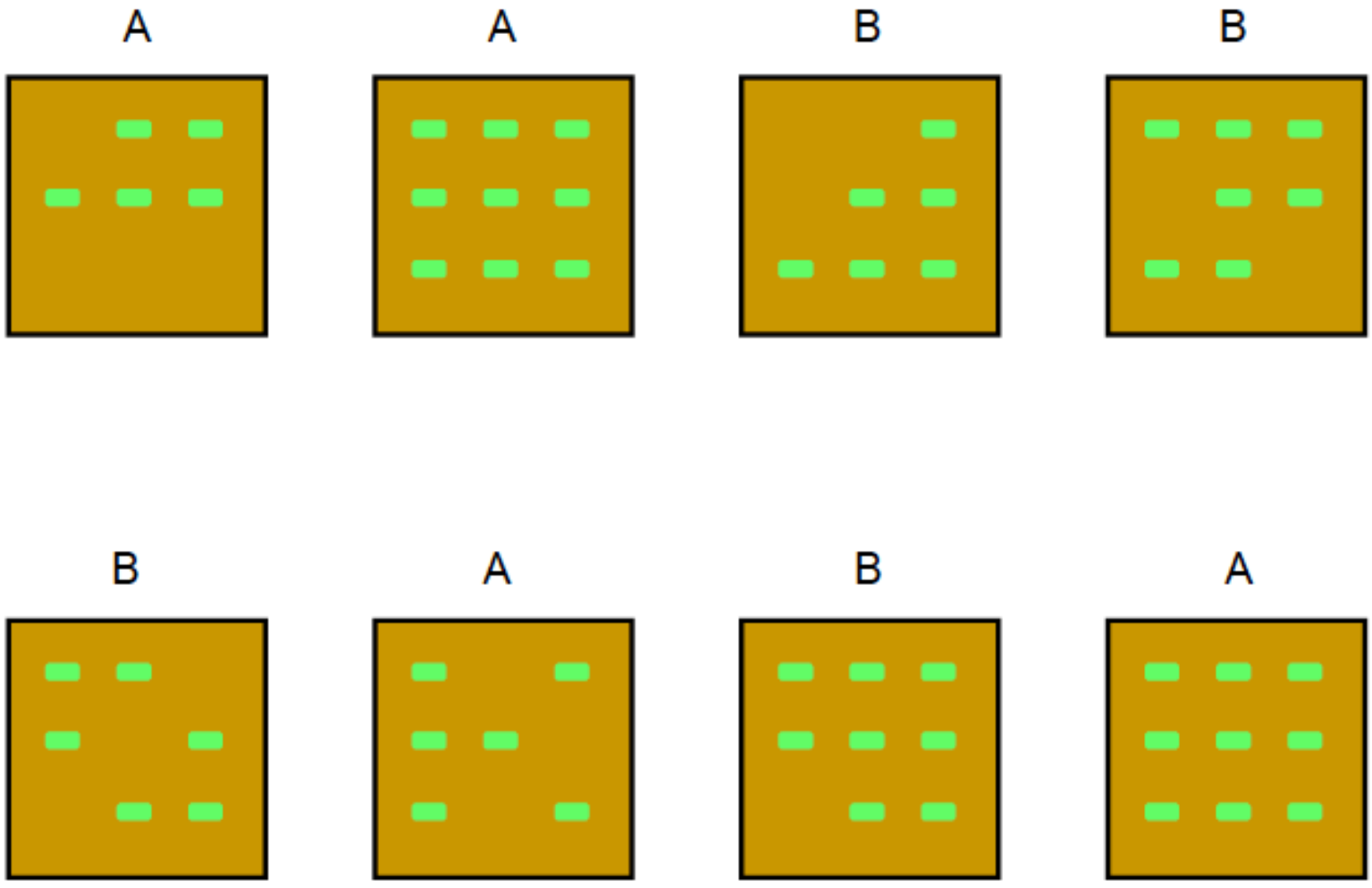
> *Researchers were interested in comparing the dry weight of maize seedlings from two different genotypes (A and B). For each genotype, nine seeds were planted in each of four trays. The eight trays in total were randomly positioned in a growth chamber. Three weeks after the emergence of the first seedling, emerged seedlings were harvested from each tray and, after drying, weighed.*

Assume the missing data (emergence) mechanism is ignorable.

Data: {`http://www.public.iastate.edu/~dnett/S511/SeedlingDryWeight2.txt`}

A picture




A
A
B
B
B
A
B
A

A mixed effect model for seedling weight

Let y_{ijk} be the seedling weight of the

- i^{th} genotype with $i = 1, 2$,
- j^{th} tray $j = 1, 2, 3, 4$ of the i^{th} genotype, and
- k^{th} seedling with $k = 1, \ldots, n_{ij}$.

Then, we assume

$$y_{ijk} = \gamma_i + \tau_{ij} + e_{ijk}$$

where

- $\tau_{ij} \overset{ind}{\sim} N(0, \sigma_\tau^2)$ and, independently,
- $e_{ijk} \overset{ind}{\sim} N(0, \sigma_e^2)$.

The main quantity of interest is the difference in mean seedling weight: $\gamma_2 - \gamma_1$.

As a general mixed effects model

Let X have the following 2 columns

- col1: all ones (intercept) $[\gamma_1]$
- col2: ones if genotype B and zeros otherwise $[\gamma_2 - \gamma_1]$

Let Z have the following 8 columns

- col1: ones if genotype 1, tray 1 and zeros otherwise $[\tau_{11}]$
- col2: ones if genotype 1, tray 2 and zeros otherwise $[\tau_{12}]$
- $\vdots$
- col8: ones if genotype 2, tray 4 and zeros otherwise $[\tau_{24}]$

Then

$$y = X\beta + Zu + e$$

with $u \sim N(0, \sigma_\tau^2 I)$ and, independently, $e \sim N(0, \sigma_e^2 I)$.

Seedling weight data

```
d = structure(list(Genotype = structure(c(1L, 1L, 1L, 1L, 1L, 1L,
1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L,
1L, 1L, 1L, 1L, 1L, 1L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L,
2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L,
2L, 2L), .Label = c("A", "B"), class = "factor"), Tray = c(1L,
1L, 1L, 1L, 1L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 3L, 3L, 3L,
3L, 3L, 3L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 5L, 5L, 5L, 5L,
5L, 5L, 6L, 6L, 6L, 6L, 6L, 6L, 6L, 7L, 7L, 7L, 7L, 7L, 7L, 8L,
8L, 8L, 8L, 8L, 8L, 8L, 8L), Seedling = c(1L, 2L, 3L, 4L, 5L,
1L, 2L, 3L, 4L, 5L, 6L, 7L, 8L, 9L, 1L, 2L, 3L, 4L, 5L, 6L, 1L,
2L, 3L, 4L, 5L, 6L, 7L, 8L, 9L, 1L, 2L, 3L, 4L, 5L, 6L, 1L, 2L,
3L, 4L, 5L, 6L, 7L, 1L, 2L, 3L, 4L, 5L, 6L, 1L, 2L, 3L, 4L, 5L,
6L, 7L, 8L), SeedlingWeight = c(8L, 9L, 11L, 12L, 10L, 17L, 17L,
16L, 15L, 19L, 18L, 18L, 18L, 24L, 12L, 12L, 16L, 15L, 15L, 14L,
17L, 20L, 20L, 19L, 19L, 18L, 20L, 19L, 19L, 9L, 12L, 13L, 16L,
14L, 14L, 10L, 10L, 9L, 8L, 13L, 9L, 11L, 12L, 16L, 17L, 15L,
15L, 15L, 9L, 6L, 8L, 8L, 13L, 9L, 9L, 10L)), .Names = c("Genotype",
"Tray", "Seedling", "SeedlingWeight"), class = "data.frame", row.names
-56L))
d$Seedling = NULL
```

```r
head(d,3)
```

```
  Genotype Tray SeedlingWeight
1        A    1              8
2        A    1              9
3        A    1             11
```

```r
summary(d)
```

```
 Genotype      Tray        SeedlingWeight
 A:29     Min.   :1.000   Min.   : 6.00
 B:27     1st Qu.:2.750   1st Qu.:10.00
          Median :4.000   Median :14.00
          Mean   :4.554   Mean   :13.88
          3rd Qu.:6.250   3rd Qu.:17.00
          Max.   :8.000   Max.   :24.00
```

```r
with(d, table(Genotype, Tray))
```

```
        Tray
Genotype 1 2 3 4 5 6 7 8
       A 5 9 6 9 0 0 0 0
       B 0 0 0 0 6 7 6 8
```

Non-Bayesian analysis

```
# library(lme4) function lmer : linear mixed effects model
m1 = lmer(SeedlingWeight ~ Genotype + (1|Tray), d); summary(m1)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: SeedlingWeight ~ Genotype + (1 | Tray)
   Data: d

REML criterion at convergence: 247.1

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.0928 -0.5697  0.0470  0.5146  3.2347

Random effects:
 Groups   Name        Variance Std.Dev.
 Tray     (Intercept) 11.661   3.415
 Residual              3.543   1.882
Number of obs: 56, groups:  Tray, 8

Fixed effects:
            Estimate Std. Error t value
(Intercept)   15.289      1.745   8.761
GenotypeB     -3.550      2.469  -1.438

Correlation of Fixed Effects:
          (Intr)
GenotypeB -0.707
```

Why no p-values?

Explanation by package author, D. Bates):

- `lmer` provides estimates of the fixed-effects parameters, standard errors for these parameters and a t-ratio but no p-values
- recall: the t-statistic in the fixed-effects part of the model is the square root of an F-statistic with 1 numerator d.f.
- however, the estimates computed by `lmer` are MLE (or REML); not based on observed and expected mean squares
- this is important because it allows flexibility in the random effect modeling (e.g. can have unbalanced designs)
- many applications for the fixed-effects specification in a mixed model begin with the assumption that the test statistics will have an F distribution with a known numerator degrees of freedom and the only purpose of the research is to decide how to obtain an approximate denominator degrees of freedom; not the right way to think about it
- recommendation: ther is a function to compute p-values in the package; see details

```r
confint(m1, method="profile")
```

```
                 2.5 %    97.5 %
.sig01        1.837050  5.379221
.sigma        1.560415  2.332764
(Intercept) 11.926526 18.637543
GenotypeB   -8.287734  1.204894
```

```r
confint(m1, method="Wald")
```

```
                 2.5 %    97.5 %
.sig01              NA        NA
.sigma              NA        NA
(Intercept) 11.868527 18.709148
GenotypeB   -8.388448  1.288046
```

```r
confint(m1, method="boot")
```

```
                 2.5 %    97.5 %
.sig01        1.452736  5.493312
.sigma        1.532352  2.237814
(Intercept) 11.829765 18.667237
GenotypeB   -8.279851  1.231284
```

Bayesian model

An alternative notation convenient for programming in Stan is

- y_i is the weight for seedling i with $i = 1, \ldots, n$
- $g[i] \in \{1, 2\}$ is the genotype for seedling i
- $t[i] \in \{1, 2, \ldots, 8\}$ is the **unique** tray id for seedling i

Then the model is

$$y_i = \gamma_{g[i]} + \tau_{t[i]} + e_i$$

with $e_i \overset{ind}{\sim} N(0, \sigma_e^2)$ and, independently, $\tau_t \overset{ind}{\sim} N(0, \sigma_\tau^2)$ with $t = 1, \ldots, 8$.

Prior: $p(\gamma_1, \gamma_2, \sigma_e, \sigma_u) \propto Ca^+(\sigma_e; 0, 10) Ca^+(\sigma_u; 0, 10)$.

Folded Cauchy Density for Hierarchical Variance Parameters

Gelman calls this a 'weakly informative' prior.

- uniform prior is too weak (too uninformative); inferencese not precise enough
- folded Cauchy has peak at 0, single scale parameter, say A
- as $A \to \infty$, the variance tends to infinity (like uniform prior)

This is helpful when there is a small number of groups, as inference will be sensitive to prior information.

Stan model

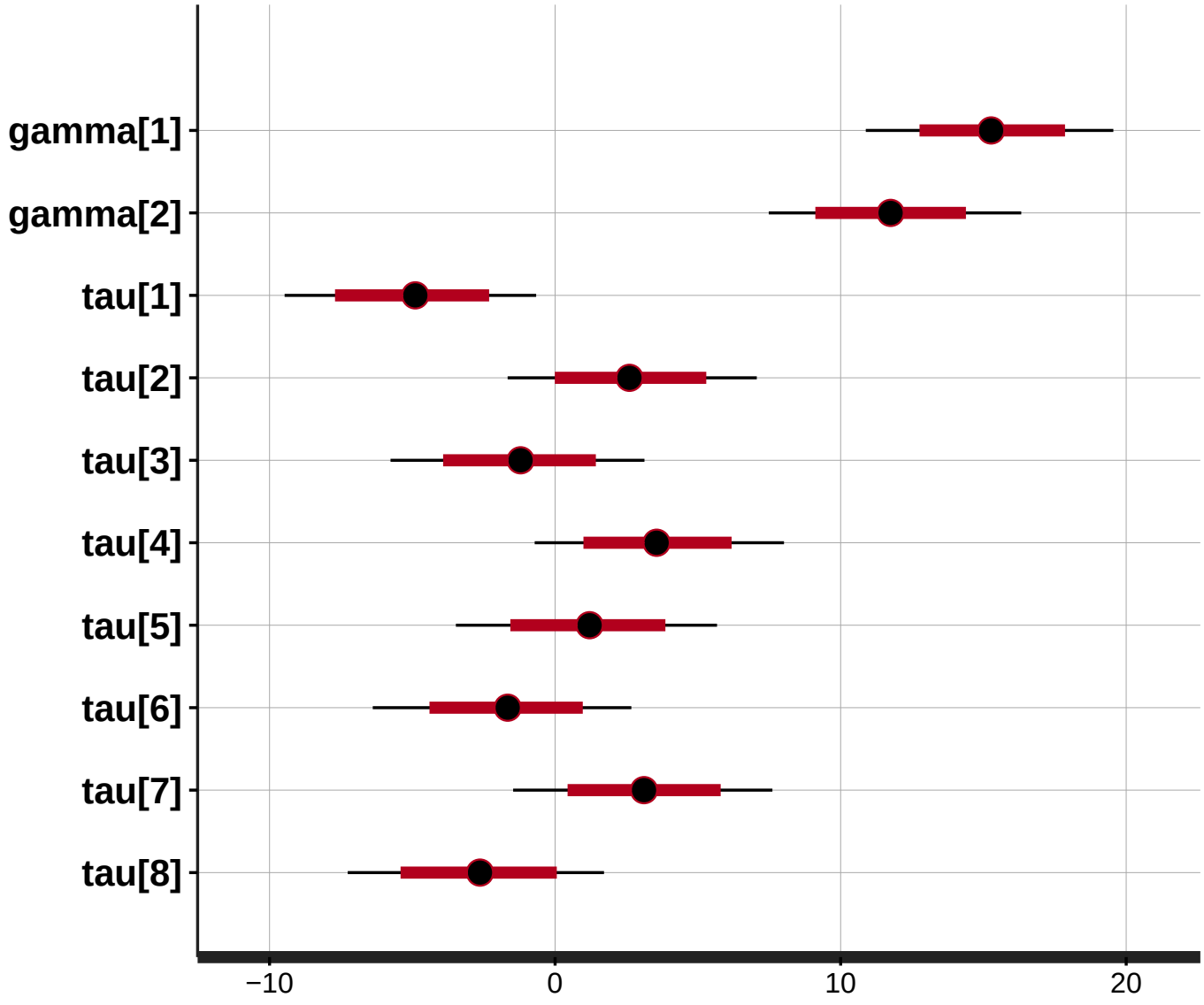
```
stan_model = "
data {
  int<lower=1> n;
  int<lower=1> n_genotypes;
  int<lower=1> n_trays;

  real y[n];
  int genotype[n];
  int tray[n];
}
parameters {
  real gamma[n_genotypes]; // Implicit prior over whole real line
  real tau[n_trays];
  real<lower=0> sigma_e;   // Implicit prior over positive reals
  real<lower=0> sigma_tau; // Implicit prior over positive reals
}

model {
  sigma_e   ~ cauchy(0,10);
  sigma_tau ~ cauchy(0,10);

  tau ~ normal(0,sigma_tau);

  for (i in 1:n) y[i] ~ normal(gamma[genotype[i]]+tau[tray[i]], sigma_e);
}

generated quantities {
  real delta;
  delta <- gamma[2] - gamma[1];
}
"
```

```
m = stan_model(model_code=stan_model)

r = sampling(m,
             list(n = nrow(d),
        n_genotypes = nlevels(d$Genotype),
        n_trays     = max(d$Tray),
        genotype    = as.numeric(d$Genotype),
        tray        = d$Tray,
        y           = d$SeedlingWeight),
        c("gamma","tau","sigma_e","sigma_tau","delta"))
```

```
SAMPLING FOR MODEL 'e78d6569de296a37b353a54065c1bb27' NOW (C

Chain 1, Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 1, Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 1, Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 1, Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 1, Iteration:  800 / 2000 [ 40%]  (Warmup)
```

```r
Inference for Stan model: e78d6569de296a37b353a54065c1bb27.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

          mean se_mean   sd   2.5%   25%   50%   75% 97.5% n_eff Rhat
gamma[1]  15.29    0.07 2.12  10.88 13.97 15.27 16.59 19.56   827    1
gamma[2]  11.76    0.07 2.17   7.48 10.47 11.75 12.98 16.32   985    1
tau[1]    -4.98    0.07 2.20  -9.47 -6.30 -4.90 -3.64 -0.66   905    1
tau[2]     2.63    0.07 2.18  -1.67  1.27  2.60  4.02  7.06   880    1
tau[3]    -1.23    0.07 2.20  -5.78 -2.60 -1.21  0.16  3.13   872    1
tau[4]     3.59    0.07 2.16  -0.72  2.26  3.56  4.91  8.02   871    1
tau[5]     1.18    0.07 2.24  -3.47 -0.09  1.21  2.52  5.67  1040    1
tau[6]    -1.71    0.07 2.21  -6.39 -3.00 -1.66 -0.36  2.67  1040    1
tau[7]     3.10    0.07 2.23  -1.47  1.77  3.11  4.42  7.61  1051    1
tau[8]    -2.67    0.07 2.23  -7.26 -3.98 -2.63 -1.32  1.71  1007    1
sigma_e    1.93    0.01 0.20   1.59  1.79  1.91  2.06  2.38  1237    1
sigma_tau  4.11    0.05 1.44   2.20  3.12  3.80  4.77  7.82   998    1
delta     -3.53    0.10 3.09  -9.76 -5.42 -3.53 -1.68  2.95   912    1
lp__     -76.85    0.09 2.71 -83.13 -78.41 -76.51 -74.89 -72.67   997    1

Samples were drawn using NUTS(diag_e) at Mon Apr 18 14:31:54 2016.
For each parameter, n_eff is a crude measure of effective sample size,
```

```
plot(r)
```


gamma[1]
gamma[2]
tau[1]
tau[2]
tau[3]
tau[4]
tau[5]
tau[6]
tau[7]
tau[8]
−10
0
10
20

Probability that genotype B has greater *mean* seedling weight than genotype A.

Given our prior, i.e.

$$p(\gamma_1, \gamma_2, \sigma_e, \sigma_u) \propto Ca^+(\sigma_e; 0, 10)Ca^+(\sigma_u; 0, 10),$$

Our posterior probability that genotype B has greater mean seedling weight than genotype A is

$$P(\gamma_2 > \gamma_1|y) = P(\delta > 0|y) = E[I(\delta > 0)|y] = E[I(\gamma_2 > \gamma_1)|y].$$

If $\delta^{(k)}$ are MCMC samples from $p(\delta|y)$, then

$$\frac{1}{K}\sum_{k=1}^{K} I(\delta^{(k)} > 0) \stackrel{a.s.}{\to} P(\gamma_2 > \gamma_1|y)$$

and (if the regularity conditions hold)

$$\frac{1}{K}\sum_{k=1}^{K} I(\delta^{(k)} > 0) \stackrel{d}{\to} N(P(\gamma_2 > \gamma_1|y), \sigma^2/K).$$

Probability that genotype B has greater mean seedling weight than genotype A

The probability is estimated to be

```
library(mcmcse)
delta = extract(r, "delta")$delta
as.data.frame(mcse(delta>0))
```

```
      est          se
1 0.10575 0.005153977
```

```
# A point estimate (posterior median) and a
# 95\% credible interval are calculated below:
ddply(dd <- data.frame(q=c(.025,.5,.975)), .(q),
      function(x) as.data.frame(mcse.q(delta, x$q)))
```

```
      q       est          se
1 0.025 -9.769312 0.17688237
2 0.500 -3.535104 0.05451168
3 0.975  2.946557 0.25801394
```

Prediction for a new comparison

The real question is whether this idea generalizes, i.e. is true for other representatives of these genotypes. Let $\tilde{y}_A$ and $\tilde{y}_B$ be some future observation of seedling weight (on the same tray) for genotype A and B, respectively. We might be interested in

$$P(\tilde{y}_B > \tilde{y}_A|y) = P(\tilde{\delta} > 0|y) = E[I(\tilde{\delta} > 0)|y]$$

where $\tilde{\delta} = \tilde{y}_B - \tilde{y}_A$. If $\tilde{\delta}^{(k)} = \tilde{y}_B^{(k)} - \tilde{y}_A^{(k)}$ is a sample from the posterior predictive distribution, then we can estimate this probability via

$$\frac{1}{K}\sum_{k=1}^{K} I(\tilde{\delta}^{(k)} > 0)$$

and have a similar LLN and CLT (if regularity conditions hold).

Prediction for a new comparison

Assuming $\tilde{y}_A^{(k)}$ and $\tilde{y}_B^{(k)}$ are independent conditional on γ_1, γ_2, and σ_e, then

$$\tilde{\delta} = \tilde{y}_B - \tilde{y}_A \sim N(\gamma_2 - \gamma_1, 2\sigma_e^2)$$

and

$$p(\tilde{\delta}|y) \int N(\tilde{\delta}; \gamma_2 - \gamma_1, 2\sigma_e^2) p(\gamma_1, \gamma_2, \sigma_e|y) d\gamma_1 d\gamma_2 d\sigma_e$$

```
samps = extract(r, c("gamma","sigma_e"))
gamma1 = samps['gamma']$gamma[,1]
gamma2 = samps['gamma']$gamma[,2]
sigmae = samps['sigma_e']$sigma_e
tilde_delta = rnorm(length(gamma1), gamma2-gamma1,
                    sqrt(2)*sigmae)
as.data.frame(mcse(tilde_delta>0))
```

```
      est          se
1 0.19125 0.006461277
```

```
ddply(data.frame(q=c(.025,.5,.975)), .(q),
  function(x) as.data.frame(mcse.q(tilde_delta, q=x$q)))
```

```
      q         est          se
1 0.025 -11.711201 0.20564243
2 0.500  -3.511492 0.07984841
3 0.975   5.008898 0.26784198
```

Extensions

Consider the model

$$y_i = \gamma_{g[i]} + \tau_{t[i]} + e_i$$

and the following modeling assumptions:

- $\gamma_g \overset{ind}{\sim} N(\mu, \sigma_\gamma^2)$ and learn μ, σ_γ
- $\tau_t \overset{ind}{\sim} La(0, \sigma_\tau^2)$
- $\gamma_g \overset{ind}{\sim} La(\mu, \sigma_\gamma^2)$
- $e_i \overset{ind}{\sim} La(0, \sigma_e^2)$
- $e_i \overset{ind}{\sim} t_\nu(0, \sigma_e^2)$

From a Bayesian perspective these changes do not affect the approach to inference.