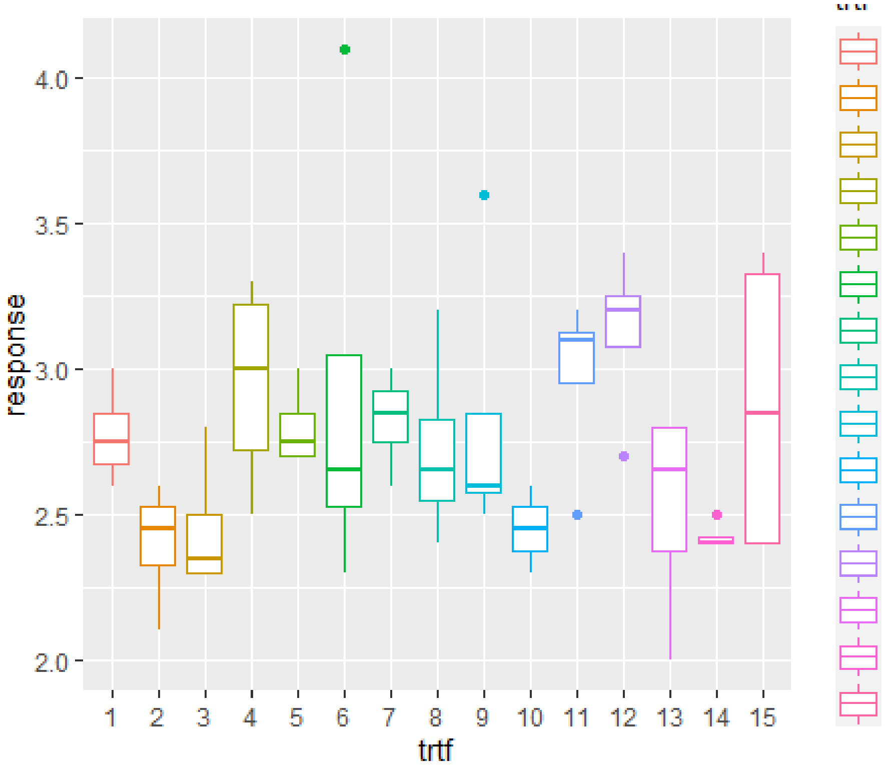
HW 7

Nathan Hawkins

3/11/2021

```
library(ggplot2)

library(dplyr)

library(brms)

library(R2jags)

pbib = read.table("pbib.dat", header = TRUE)


pbib$trtf <- as.factor(pbib$trt)
ggplot(data = pbib, mapping = aes(x = trtf, y = response, col = trtf)) +
  geom_boxplot()
```



1. Mixed Model

The effects for s2error and s2block are .098 and .075, respectively.

```
mdl <- "

model {

  for(i in 1:60){
    response[i] ~ dnorm(mu[i], 1/s2error)
    mu[i] <- beta[trt[i]] + u[blk[i]]
  }

  # Priors
  for(i in 1:15){
    beta[i] ~ dnorm(0, 0.01)
  }


  for(i in 1:15){
    u[i] ~ dnorm(0, 1/s2blk)
  }

  s2error ~ dgamma(1.1, 0.1)
  s2blk ~ dgamma(1.1, 0.1)

}
"

writeLines(mdl, 'dep.txt')

response <- pbib$response
trt <- pbib$trt
blk <- pbib$blk

data.jags <- c('response', 'trt', 'blk')
parms <- c('beta','s2error', 's2blk')

mixedmods.sim <- jags(data= data.jags, parameters.to.save = parms,
                  model.file = 'dep.txt', inits = NULL,
                  n.iter = 12000, n.thin = 5, n.burnin = 2000,
                  n.chains = 5)

## module glm loaded
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 60
##    Unobserved stochastic nodes: 32
##    Total graph size: 279
##
## Initializing model
```

Raftery Lewis Diagnostics look good; they are around 0. So do effective sample sizes; they are all over 5,000.

```
sims <- as.mcmc(mixedmods.sim)
chains <- as.matrix(sims)
raftery.diag(chains)
```

```
##
## Quantile (q) = 0.025
## Accuracy (r) = +/- 0.005
## Probability (s) = 0.95
##
##           Burn-in  Total Lower bound  Dependence
##           (M)      (N)   (Nmin)       factor (I)
##  beta[1]  2        3931  3746         1.050
##  beta[10] 2        3865  3746         1.030
##  beta[11] 2        3620  3746         0.966
##  beta[12] 2        3802  3746         1.010
##  beta[13] 2        3710  3746         0.990
##  beta[14] 2        3620  3746         0.966
##  beta[15] 2        3696  3746         0.987
##  beta[2]  2        3771  3746         1.010
##  beta[3]  2        3865  3746         1.030
##  beta[4]  2        3802  3746         1.010
##  beta[5]  2        3802  3746         1.010
##  beta[6]  2        3710  3746         0.990
##  beta[7]  2        3710  3746         0.990
##  beta[8]  2        3834  3746         1.020
##  beta[9]  2        3710  3746         0.990
##  deviance 2        3897  3746         1.040
##  s2blk    7        7746  3746         2.070
##  s2error  2        3680  3746         0.982
```

```
effectiveSize(chains)
```

```
##   beta[1]  beta[10]  beta[11]  beta[12]  beta[13]  beta[14]  beta[15] 
beta[2]
## 10000.000 10368.471 10472.116 10000.000 10000.000 10000.000 10000.000 
10000.000
##   beta[3]   beta[4]   beta[5]   beta[6]   beta[7]   beta[8]   beta[9] 
deviance
```

```
## 10000.000  9375.009 10000.000 10000.000 10527.150 10000.000 10000.000
6583.315
##     s2blk   s2error
##  5029.704  7208.927
```

2)

The 5 strains with the highest responses are 12, 6, 9, 11, 15

3)

The top five are not statistically significant from each other. But there is a significant difference between strain 12 and strain 2.

```
quantile(chains[,8] - chains[,3], probs = c(.025, .975))

##        2.5%       97.5%
## -0.96786310 -0.04105539

quantile(chains[,9] - chains[,3], probs = c(.025, .975))

##        2.5%       97.5%
## -0.91940395  0.02106534
```

4) Now do it all in brm

Effective sample sizes look good. The top 5 strains are 12, 6, 9, 11, 15. These are the same as before

```
fit2 <- brm(formula = response ~ -1 + trtf + (1|blk), data = pbib,
            family = "gaussian",
            prior = c(set_prior("normal(70,100)", class = "b"),
                      set_prior("gamma(1.1,0.5)", class = "sd")),
            warmup = 1000, iter = 3000, chains = 4,
            control = list(adapt_delta = 0.98),
            save_pars = save_pars(all = TRUE))

## Compiling Stan program...

## Start sampling

##
## SAMPLING FOR MODEL '7df9fdf436814d3fa65f6b2f2e9ce312' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would
take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 3000 [  0%]  (Warmup)
## Chain 1: Iteration:  300 / 3000 [ 10%]  (Warmup)
## Chain 1: Iteration:  600 / 3000 [ 20%]  (Warmup)
## Chain 1: Iteration:  900 / 3000 [ 30%]  (Warmup)
## Chain 1: Iteration: 1001 / 3000 [ 33%]  (Sampling)
## Chain 1: Iteration: 1300 / 3000 [ 43%]  (Sampling)
## Chain 1: Iteration: 1600 / 3000 [ 53%]  (Sampling)
## Chain 1: Iteration: 1900 / 3000 [ 63%]  (Sampling)
## Chain 1: Iteration: 2200 / 3000 [ 73%]  (Sampling)
## Chain 1: Iteration: 2500 / 3000 [ 83%]  (Sampling)
## Chain 1: Iteration: 2800 / 3000 [ 93%]  (Sampling)
## Chain 1: Iteration: 3000 / 3000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 1.069 seconds (Warm-up)
## Chain 1:                1.051 seconds (Sampling)
## Chain 1:                2.12 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL '7df9fdf436814d3fa65f6b2f2e9ce312' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would
take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 3000 [  0%]  (Warmup)
## Chain 2: Iteration:  300 / 3000 [ 10%]  (Warmup)
## Chain 2: Iteration:  600 / 3000 [ 20%]  (Warmup)
## Chain 2: Iteration:  900 / 3000 [ 30%]  (Warmup)
## Chain 2: Iteration: 1001 / 3000 [ 33%]  (Sampling)
## Chain 2: Iteration: 1300 / 3000 [ 43%]  (Sampling)
## Chain 2: Iteration: 1600 / 3000 [ 53%]  (Sampling)
## Chain 2: Iteration: 1900 / 3000 [ 63%]  (Sampling)
## Chain 2: Iteration: 2200 / 3000 [ 73%]  (Sampling)
## Chain 2: Iteration: 2500 / 3000 [ 83%]  (Sampling)
## Chain 2: Iteration: 2800 / 3000 [ 93%]  (Sampling)
## Chain 2: Iteration: 3000 / 3000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.75 seconds (Warm-up)
## Chain 2:                1.304 seconds (Sampling)
## Chain 2:                2.054 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL '7df9fdf436814d3fa65f6b2f2e9ce312' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would
```

```
take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 3000 [  0%]  (Warmup)
## Chain 3: Iteration:  300 / 3000 [ 10%]  (Warmup)
## Chain 3: Iteration:  600 / 3000 [ 20%]  (Warmup)
## Chain 3: Iteration:  900 / 3000 [ 30%]  (Warmup)
## Chain 3: Iteration: 1001 / 3000 [ 33%]  (Sampling)
## Chain 3: Iteration: 1300 / 3000 [ 43%]  (Sampling)
## Chain 3: Iteration: 1600 / 3000 [ 53%]  (Sampling)
## Chain 3: Iteration: 1900 / 3000 [ 63%]  (Sampling)
## Chain 3: Iteration: 2200 / 3000 [ 73%]  (Sampling)
## Chain 3: Iteration: 2500 / 3000 [ 83%]  (Sampling)
## Chain 3: Iteration: 2800 / 3000 [ 93%]  (Sampling)
## Chain 3: Iteration: 3000 / 3000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.676 seconds (Warm-up)
## Chain 3:                1.275 seconds (Sampling)
## Chain 3:                1.951 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL '7df9fdf436814d3fa65f6b2f2e9ce312' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would
take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 3000 [  0%]  (Warmup)
## Chain 4: Iteration:  300 / 3000 [ 10%]  (Warmup)
## Chain 4: Iteration:  600 / 3000 [ 20%]  (Warmup)
## Chain 4: Iteration:  900 / 3000 [ 30%]  (Warmup)
## Chain 4: Iteration: 1001 / 3000 [ 33%]  (Sampling)
## Chain 4: Iteration: 1300 / 3000 [ 43%]  (Sampling)
## Chain 4: Iteration: 1600 / 3000 [ 53%]  (Sampling)
## Chain 4: Iteration: 1900 / 3000 [ 63%]  (Sampling)
## Chain 4: Iteration: 2200 / 3000 [ 73%]  (Sampling)
## Chain 4: Iteration: 2500 / 3000 [ 83%]  (Sampling)
## Chain 4: Iteration: 2800 / 3000 [ 93%]  (Sampling)
## Chain 4: Iteration: 3000 / 3000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.728 seconds (Warm-up)
## Chain 4:                0.873 seconds (Sampling)
## Chain 4:                1.601 seconds (Total)
## Chain 4:
```

```r
summary(fit2)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: response ~ -1 + trtf + (1 | blk)
##    Data: pbib (Number of observations: 60)
## Samples: 4 chains, each with iter = 3000; warmup = 1000; thin = 1;
##          total post-warmup samples = 8000
##
## Group-Level Effects:
## ~blk (Number of levels: 15)
##                  Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.23      0.08     0.07     0.39 1.00     1818     1959
##
## Population-Level Effects:
##        Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## trtf1      2.82      0.18     2.47     3.16 1.00     6083     5555
## trtf2      2.41      0.18     2.06     2.76 1.00     6449     5905
## trtf3      2.45      0.18     2.10     2.80 1.00     6454     5926
## trtf4      2.80      0.18     2.44     3.17 1.00     5736     5476
## trtf5      2.81      0.18     2.46     3.17 1.00     5690     5337
## trtf6      2.92      0.18     2.57     3.27 1.00     6166     5604
## trtf7      2.79      0.17     2.45     3.14 1.00     6186     5852
## trtf8      2.78      0.18     2.43     3.13 1.00     6549     5382
## trtf9      2.89      0.18     2.53     3.24 1.00     6173     5241
## trtf10     2.49      0.18     2.14     2.84 1.00     6558     5933
## trtf11     2.91      0.18     2.56     3.26 1.00     7068     5826
## trtf12     3.06      0.18     2.71     3.42 1.00     6374     4902
## trtf13     2.62      0.18     2.27     2.97 1.00     5447     6048
## trtf14     2.49      0.18     2.12     2.84 1.00     5798     5272
## trtf15     2.87      0.18     2.52     3.21 1.00     6294     5722
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.31      0.04     0.24     0.40 1.00     2843     4572
##
## 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).
```

p_loo is 23.1 for this analysis.

```
library(loo)

## Warning: package 'loo' was built under R version 4.0.3

## This is loo version 2.4.1

## - Online documentation and vignettes at mc-stan.org/loo

## - As of v2.0.0 loo defaults to 1 core but we recommend using as many as
possible. Use the 'cores' argument or set options(mc.cores = NUM_CORES) for
an entire session.
```

```
## - Windows 10 users: loo may be very slow if 'mc.cores' is set in your
.Rprofile file (see https://github.com/stan-dev/loo/issues/94).
```

```
loo1 <- loo(fit2)
```

```
## Warning: Found 7 observations with a pareto_k > 0.7 in model 'fit2'. It is
## recommended to set 'moment_match = TRUE' in order to perform moment
matching for
## problematic observations.
```