# Bayesian II Assignment 02

Colin Yip & Joshua Damm

## Introduction

Penicillin production is dependent on corn steep liquor, which is highly variable in quality. These differences significantly impact the final yield of penicillin. In addition, there are numerous processes involving corn steep liquor to produce penicillin. This study seeks to assess the impact of blends and treatments on final production yield by implementing a randomized complete block design (RCBD). 5 blends of corn steep liquor were set up as blocks for 4 randomly ordered production processes. Doing this removed differences between blends and allows for statistically valid analysis.

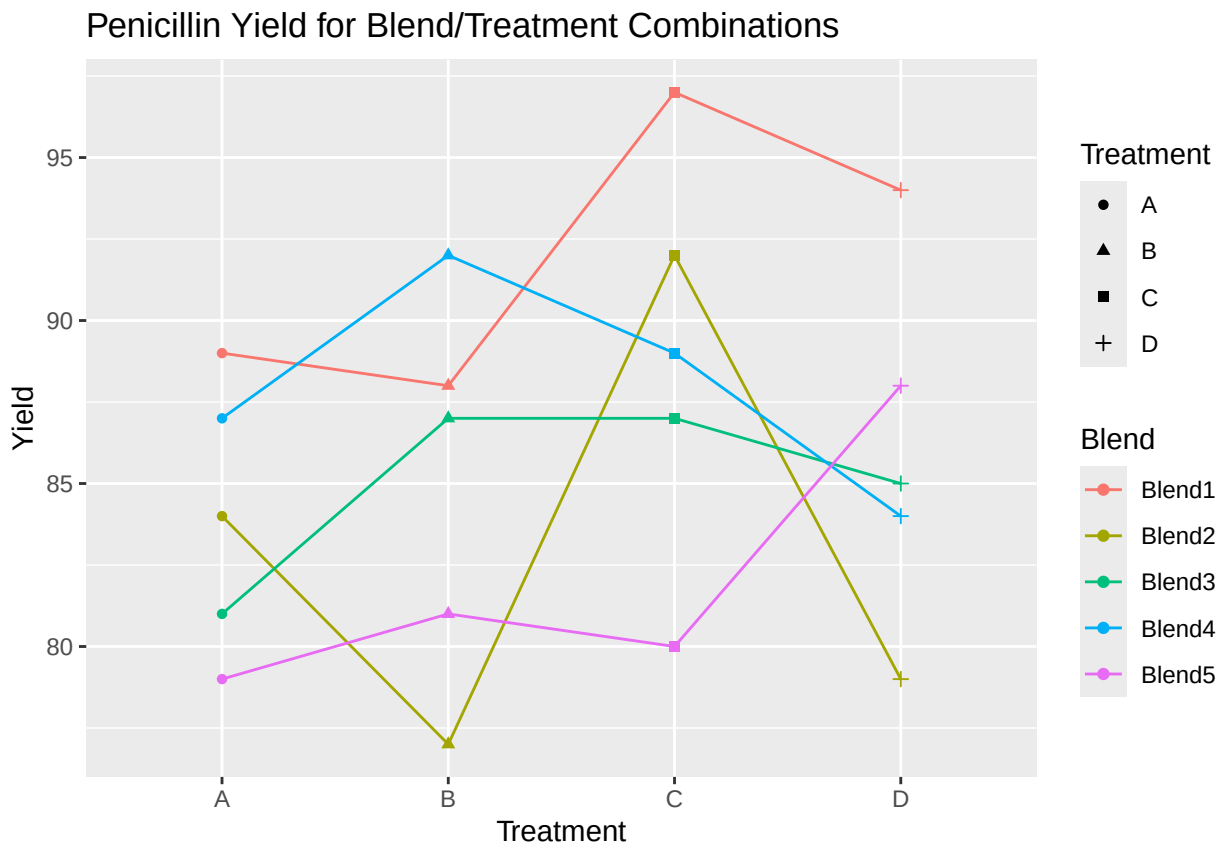
## Data Import and Exploratory Data Analysis

```
set.seed(12345678)

data(penicillin)
summary(penicillin)
```

```
##  treat    blend       yield
##  A:5   Blend1:4   Min.   :77
##  B:5   Blend2:4   1st Qu.:81
##  C:5   Blend3:4   Median :87
##  D:5   Blend4:4   Mean   :86
##        Blend5:4   3rd Qu.:89
##                   Max.   :97
```

We have 20 data points comprising of the combinations of production processes/treatments (`treat`) and the blended liquors (`blend`), which are the blocks in the RCBD design. These two variables are categorical, with the final yield (`yield`) being a continuous variable.

```
ggplot(penicillin, aes(y = yield, x = treat, shape = treat, color = blend)) +
  geom_point() +
  geom_line(penicillin, mapping=aes(y=yield, x=treat, group=blend)) +
  labs(x = "Treatment",
       y = "Yield",
       title = "Penicillin Yield for Blend/Treatment Combinations",
       shape = "Treatment",
       color = "Blend")
```




There are visually evident differences between the treatment levels. The variance in treatment A is considerably less than the spread in treatments B and C. Additionally, C exhibits a higher average yield than the other groups. Blend 1 clearly produces a higher yield than the other blends on average, with blend 5 potentially producing the least. It is also notable that blend 2 is highly variable between the various treatments. We validate these visual differences below by aggregating by both treatment and blend, respectively.

```
penicillin |>
  group_by(treat) |>
  summarise(treat_mu = mean(yield),
            treat_sigma = sd(yield))
```

```
## # A tibble: 4 x 3
##   treat treat_mu treat_sigma
##   <fct>    <dbl>       <dbl>
## 1 A           84        4.12
## 2 B           85        5.96
## 3 C           89        6.28
## 4 D           86        5.52
```

```
penicillin |>
  group_by(blend) |>
  summarise(blend_mu = mean(yield),
            blend_sigma = sd(yield))
```

```
## # A tibble: 5 x 3
```

```
##   blend  blend_mu blend_sigma
##   <fct>     <dbl>       <dbl>
## 1 Blend1       92        4.24
## 2 Blend2       83        6.68
## 3 Blend3       85        2.83
## 4 Blend4       88        3.37
## 5 Blend5       82        4.08
```

Based on the study design and the data, we will perform analysis using a Bayesian hierarchical model that accounts for differences both within blocks and between blocks. We specify the model as follows $y_{i,j} = \mu_j + \beta x + \epsilon_{i,j}, \epsilon_{i,j} \sim N(0, \sigma_\epsilon)$, for treatments $j = 1, \ldots, 4$ corresponding to treatments A through D, respectively. The intercept $\mu_j$ can be further decomposed into a cornerstone mean and the difference for a specific treatment mean, $\mu_j = \alpha + \alpha_j$. We can take $\beta$ to be the same across all blocks, as random ordering of the 4 treatments within each block nullifies the block differences. Furthermore, we assume no interaction between the block and the treatment as there is no evidence to support interaction.

## Model Specification and Prior Selection

We specify the model as follows, as `blend` is the blocking/grouping of the data, and `treat` is the covariate stratified across the blocks. As mentioned above, we take a cornerstone representation given the categorical nature of `treat`. This also aids with interpretability of the model later on.

```
model_formula <- bf(yield ~ 1 + (1 | blend) + treat)
```

We check the default priors as given by `brms`.

```
get_prior(formula = model_formula, data = penicillin)
```

```
##                  prior     class      coef group resp dpar nlpar lb ub
##                 (flat)         b                                       
##                 (flat)         b    treatB                             
##                 (flat)         b    treatC                             
##                 (flat)         b    treatD                             
##  student_t(3, 87, 5.9) Intercept                                       
##   student_t(3, 0, 5.9)        sd                                     0
##   student_t(3, 0, 5.9)        sd           blend                     0
##   student_t(3, 0, 5.9)        sd Intercept blend                     0
##   student_t(3, 0, 5.9)     sigma                                     0
##       source
##      default
##  (vectorized)
##  (vectorized)
##  (vectorized)
##      default
##      default
##  (vectorized)
##  (vectorized)
##      default
```

Clearly, the flat priors for the treatment groups are non-informative and illogical for a finitely bounded variable. While we have a smaller sample size, we do not see evidence to support the need for large tails on the intercept as the treatments are all quite close together in mean. As such, a normal prior will suffice for the

intercept, centered on the mean of the response and with a $\sigma$ capturing the spread of the data. Assuming $N(0, 10)$ for the difference between treatment levels ensures that the maximum difference of 5 between treatment means is captured. For the between group variance, we use the Student$(3, 0, \lambda_\mu)$ distribution with $\lambda_\mu = 10$ to fully capture the mean differences between blocks/groups. Finally, it is standard to assume an $Exp(\sigma_\epsilon^{-1})$ for the prior of the deviation.

```
mu_y <- mean(penicillin$yield)
std_y <- sd(penicillin$yield)

mu_y # Prior for intercept
```

```
## [1] 86
```

```
std_y # Prior for sigma_E
```

```
## [1] 5.428967
```

```
priors <- c(set_prior("normal(0, 10)", class = "b", coef = "treatB"), # Prior for the fixed effects (al
            set_prior("normal(0, 10)", class = "b", coef = "treatC"),
            set_prior("normal(0, 10)", class = "b", coef = "treatD"),
            set_prior("normal(86, 10)", class = "Intercept"), # Prior for the intercept (alpha)
            set_prior("student_t(3, 0, 10)", # prior for between-group variance(sigma_mu)
                      class = "sd",
                      group = "blend",
                      lb = 0),
            set_prior("exponential(1.0 / 5.43)", class = "sigma", lb = 0) # Prior for the random effect
            )
```

## Model Fitting and Summary

Having specified our priors, we now fit the model using `brms`.

```
bayesian_LMM <- brm(formula = model_formula,
                    data = penicillin,
                    prior = priors,
                    control = list(adapt_delta = 0.9),
                    cores = 4)
```

```
## Compiling Stan program...
```

```
## Start sampling
```

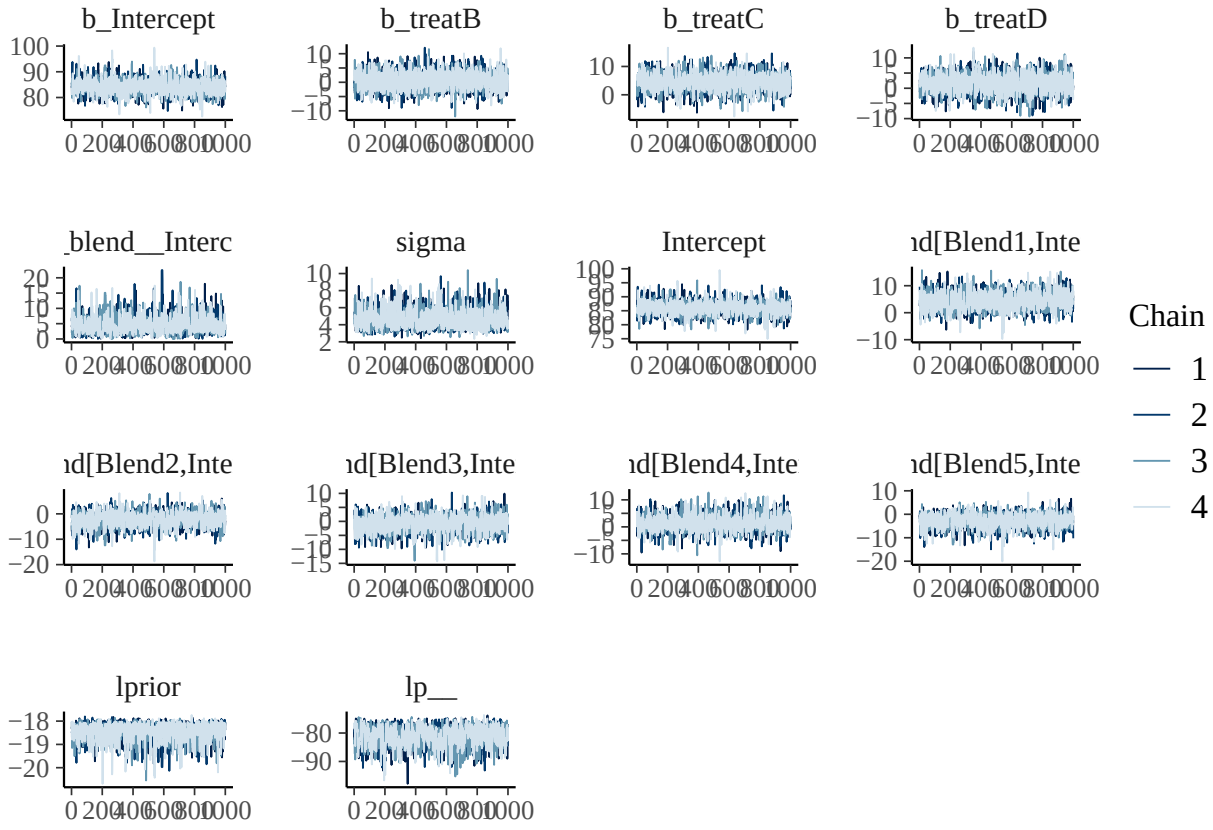
```
summary(bayesian_LMM)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: yield ~ 1 + (1 | blend) + treat
##    Data: penicillin (Number of observations: 20)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
```

```
##
## Multilevel Hyperparameters:
## ~blend (Number of levels: 5)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     4.35      2.68     0.59    11.03 1.00      988      913
##
## Regression Coefficients:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept    84.31      2.84    78.86    90.18 1.00     1650     1538
## treatB        0.64      2.84    -4.93     6.23 1.00     2707     2927
## treatC        4.45      2.84    -1.34    10.12 1.00     2847     2449
## treatD        1.50      2.90    -4.30     7.04 1.00     2652     2326
##
## Further Distributional Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     4.70      1.02     3.16     7.06 1.00     1525     2220
##
## Draws were sampled 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).
```

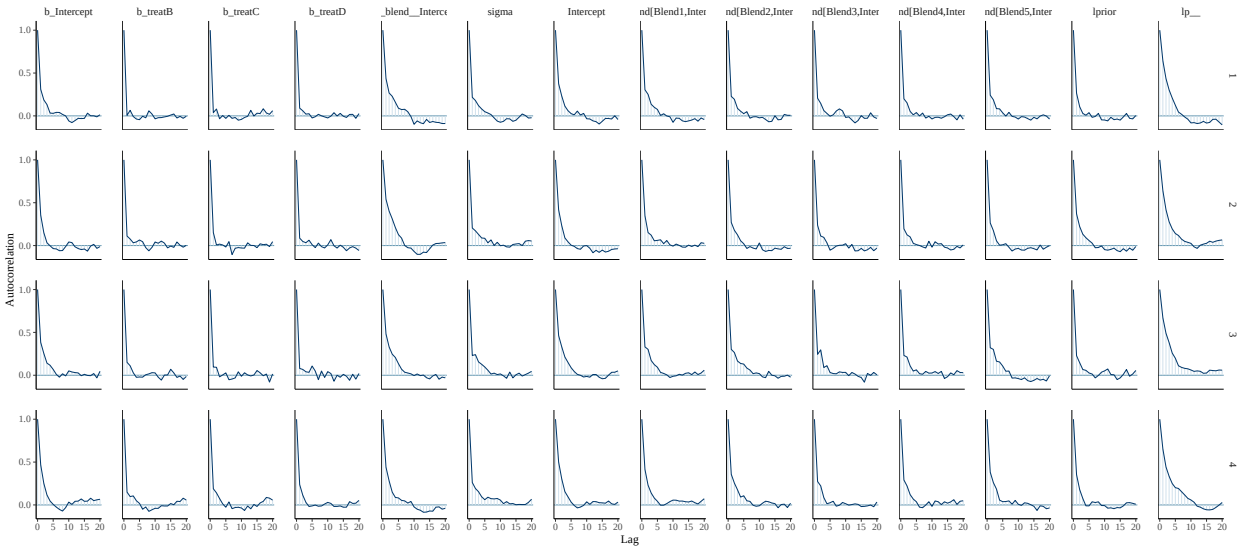
From the summary we can see that the chains have converged, as $\hat{R}$ is 1 across all parameters and the Effective Sample Sizes are large. We further validate this by checking the trace plots and autocorrelations of the chains.

```
mcmc_trace(bayesian_LMM)
```



Looking at the trace plots, we can see that the MCMC sampler has mixed well as the chains have neither experienced very large jumps nor remained stagnant in one spot for a large number of steps, indicating that the parameter space has been thoroughly explored.

```
mcmc_acf(bayesian_LMM)
```



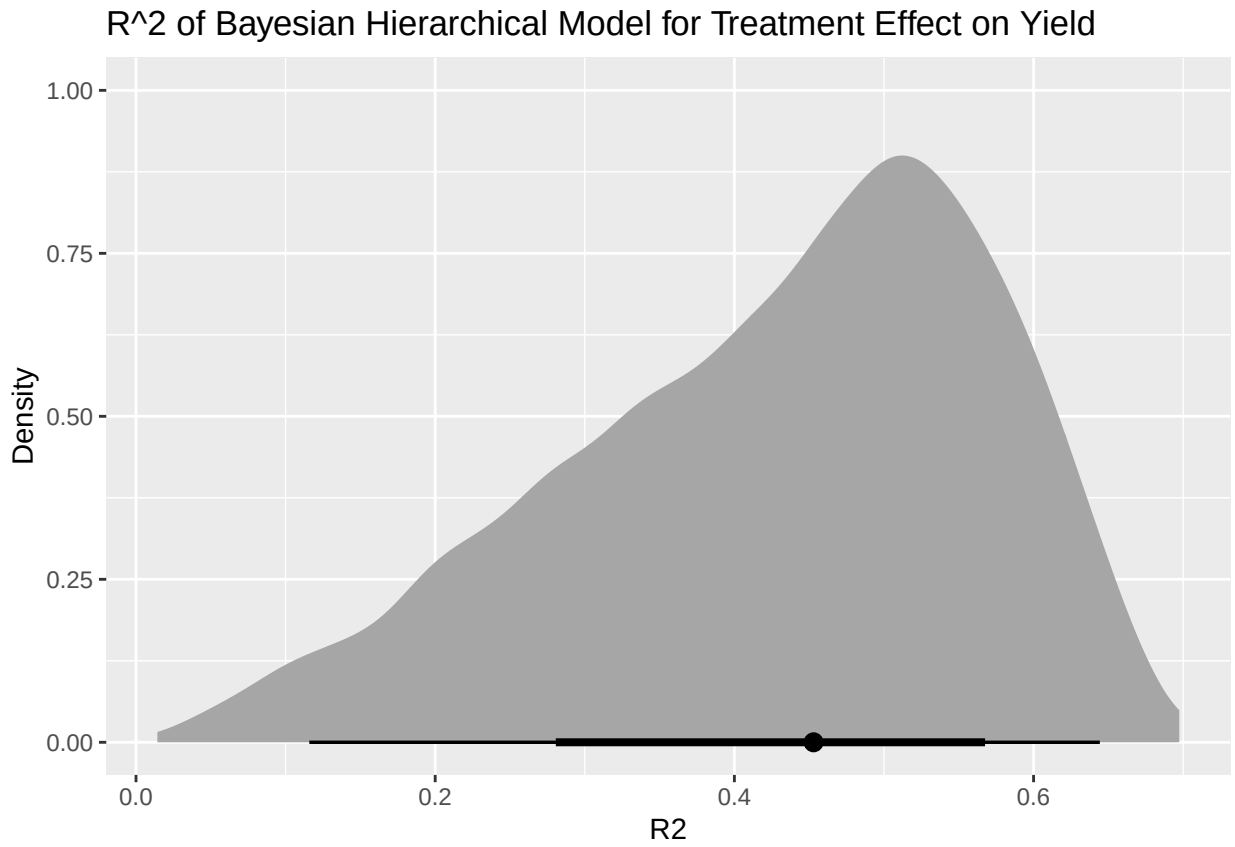

The autocorrelation plots indicate that the chains have completely checked the posterior, with no high autocorrelation remaining. This indicates that no chain has gotten stuck on a local value, and the chains have roughly found a optimal solution within the parameter space. As such, we can confirm that the Markov chains of the model have converged.

Looking at the model estimates, we can see that the impact of the between group/blend differences is significant with an estimated standard deviation of 4.42 and a 95% Credible Interval (CI) of $(0.63, 11.47)$. However, there is no significant difference of the treatments from the cornerstone treatment (treatment A) as all estimated coefficients have 95% CIs that include 0. We now look at the Bayesian $R^2$ to assess how much of the variance in the data the model can explain.

```
regression_R2 <- bayes_R2(bayesian_LMM, summary = FALSE)
regression_R2_df <- data.frame(R2 = regression_R2)
mean(regression_R2_df$R2)
```

```
## [1] 0.428846
```

```
regression_R2_df |>
  ggplot(aes(x = R2)) +
  stat_halfeye() +
  labs(title = "R^2 of Bayesian Hierarchical Model for Treatment Effect on Yield",
       y = "Density")
```




We see an average $R^2$ of about 0.43, which indicates that the model can explain less than half of the total variation within the data.

### Model Diagnostics and Assumption Checking

```
# Between vs within variation
icc(bayesian_LMM, by_group = TRUE)
```

```
## # ICC by Group
##
## Group |   ICC
## -------------
## blend | 0.485
```

```
# Some other performance metrics
model_performance(bayesian_LMM, metrics = "common", verbose = TRUE)
```

```
## Response residuals not available to calculate mean square error. (R)MSE
##   is probably not reliable.

## # Indices of model performance
##
## ELPD    | ELPD_SE |   LOOIC | LOOIC_SE |    WAIC |    R2 | R2 (marg.) |  RMSE
```

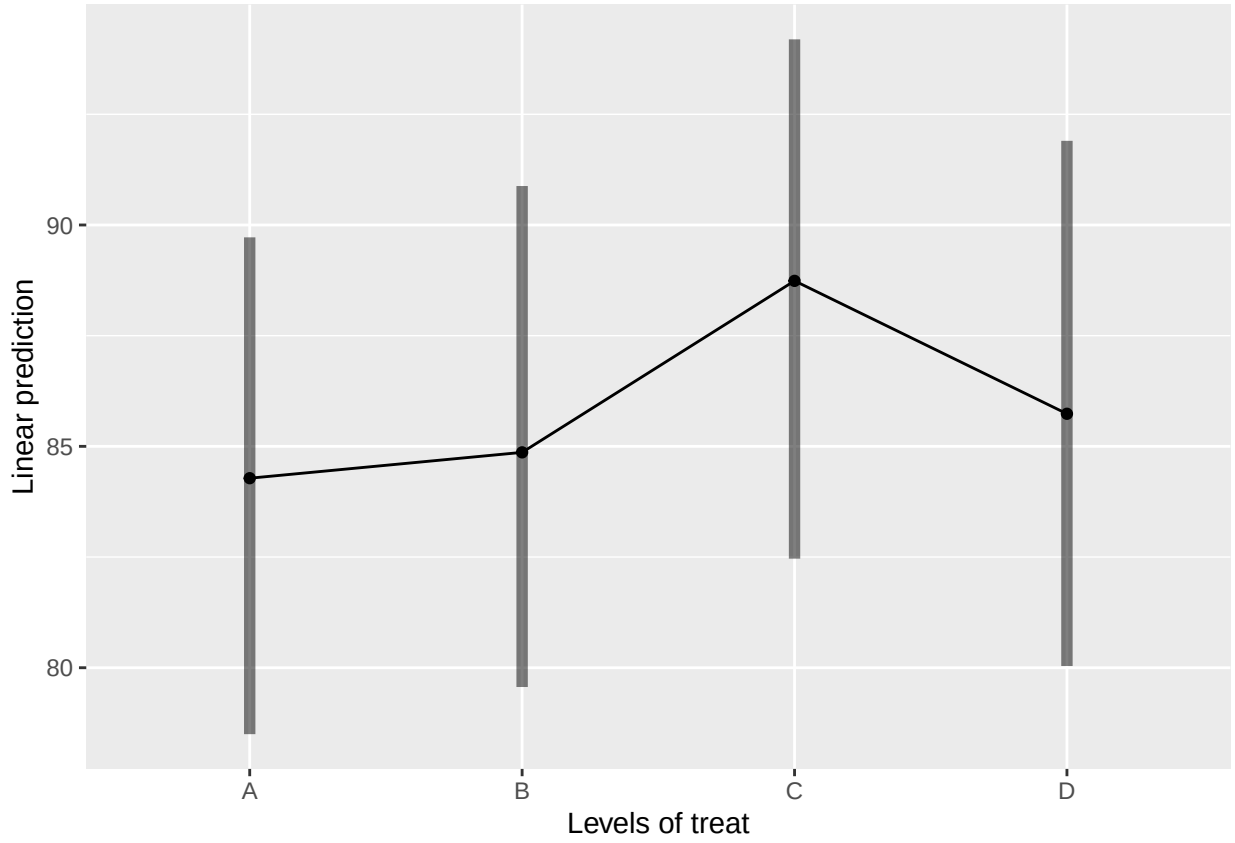
```
## ------------------------------------------------------------------------------
## -63.515 |   2.283 | 127.031 |    4.566 | 125.418 | 0.453 |      0.158 | 3.601
```

The intraclass correlation (ICC) suggests that there is more variability within-groups than between groups, meaning that the yield is more similar across blends/blocks than within blends/blocks. Mechanically, the calculation of explained differences between groups is driven by the following calculation: $\frac{\sigma_\mu^2}{\sigma_\mu^2+\sigma_\epsilon^2}$. In the case of the model above, this is thus calculated for values $\sigma_\mu = 4.42$ and $\sigma_\epsilon = 4.64$. Being below 0.5 indicates that the differences between groups explains a minority of variation in the yield.

```
## Plot estimates

bayesian_LMM_emm <- emmeans(bayesian_LMM, ~ treat)
bayesian_LMM_emm
```

```
##  treat emmean lower.HPD upper.HPD
##  A       84.3      78.5      89.7
##  B       84.9      79.6      90.9
##  C       88.7      82.5      94.2
##  D       85.7      80.0      91.9
##
## Point estimate displayed: median
## HPD interval probability: 0.95
```

```
emmip(bayesian_LMM, ~ treat, CI = TRUE)
```



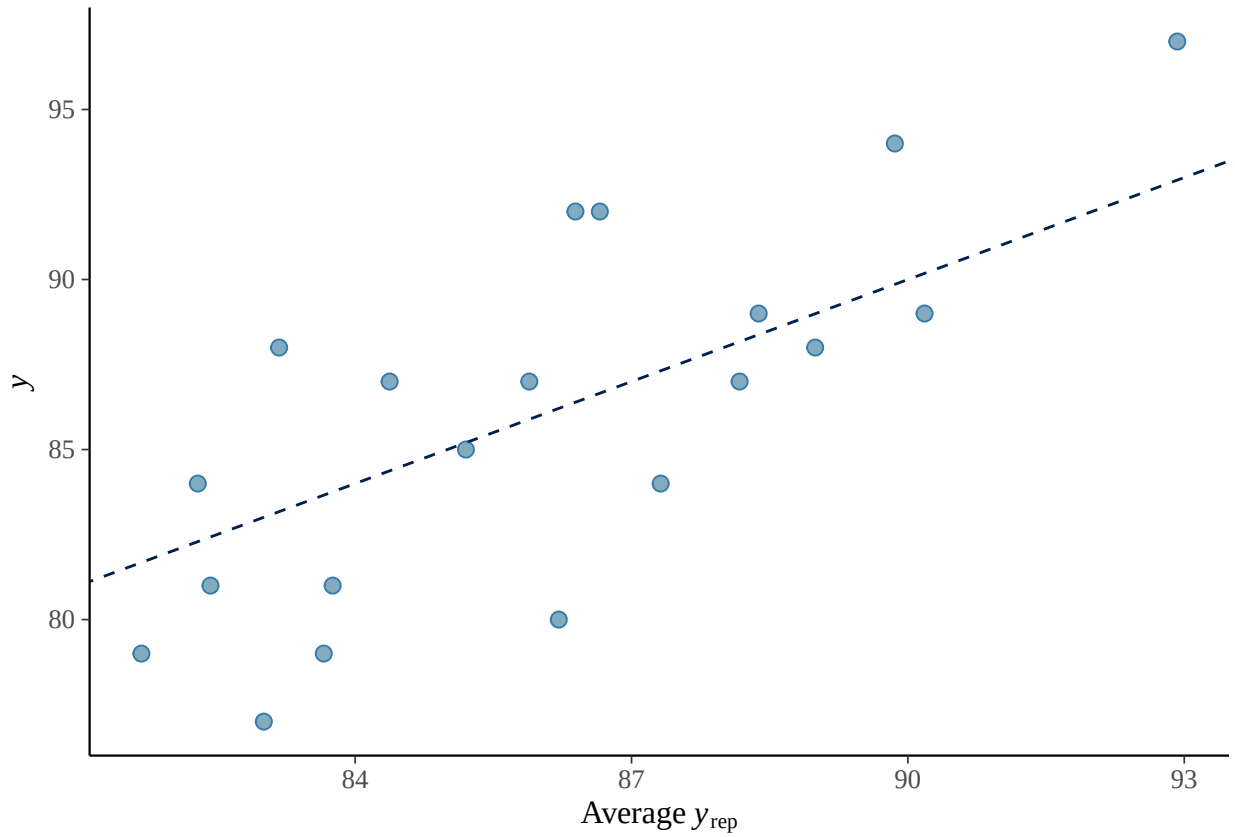
```
pairs(bayesian_LMM_emm)
```

```
##  contrast estimate lower.HPD upper.HPD
##  A - B      -0.660     -6.18      4.98
##  A - C      -4.441    -10.22      1.17
##  A - D      -1.536     -7.10      4.12
##  B - C      -3.794    -10.16      1.62
##  B - D      -0.862     -6.67      5.40
##  C - D       2.902     -3.01      8.82
##
## Point estimate displayed: median
## HPD interval probability: 0.95
```

Looking at the posterior means, one can see that treatment C shows the highest yield. However, when looking at the pairwise differences, each HPD interval includes 0, suggesting no significant difference between the treatment means. This can be visually validated as well given the overlap between the CI bars of each level of treatment. As such, the treatments/processes are not significantly different.

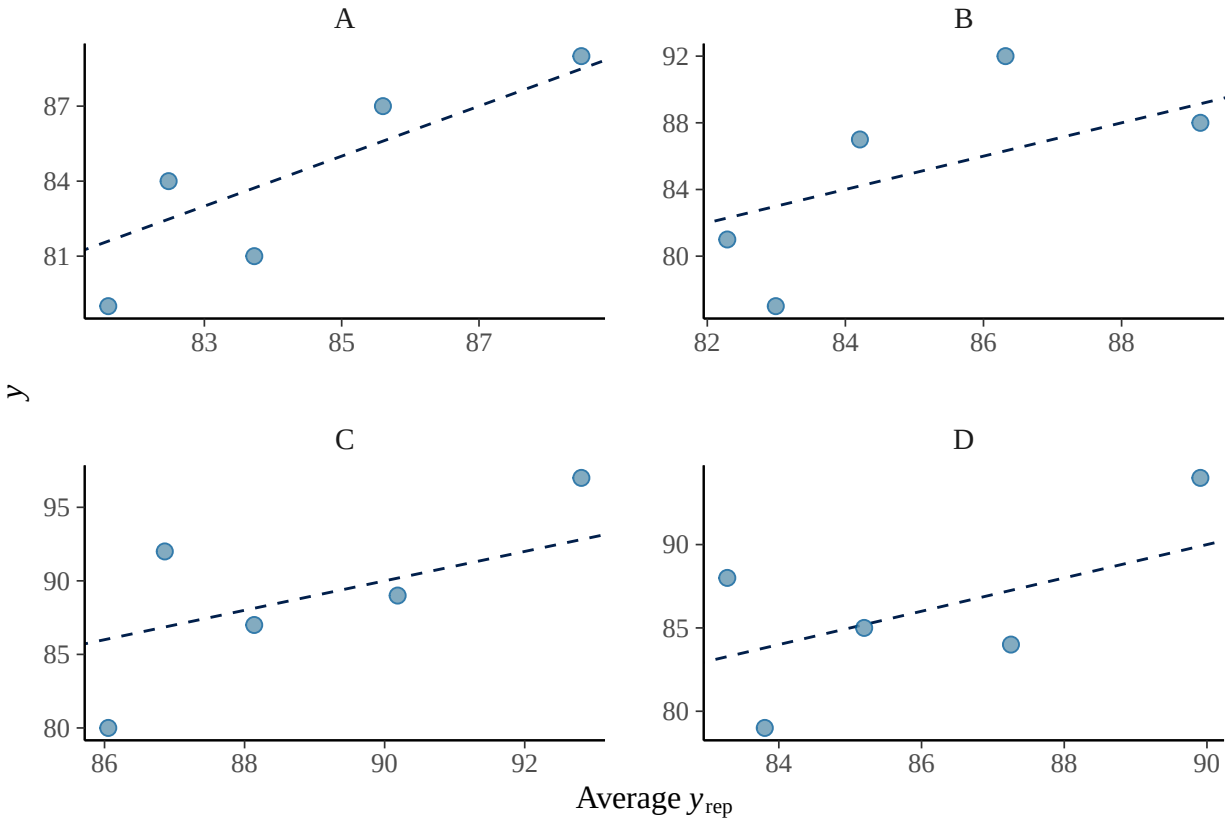
```
# PPC
pp_check(bayesian_LMM, type = "scatter_avg")
```

```
## Using all posterior draws for ppc type 'scatter_avg' by default.
```



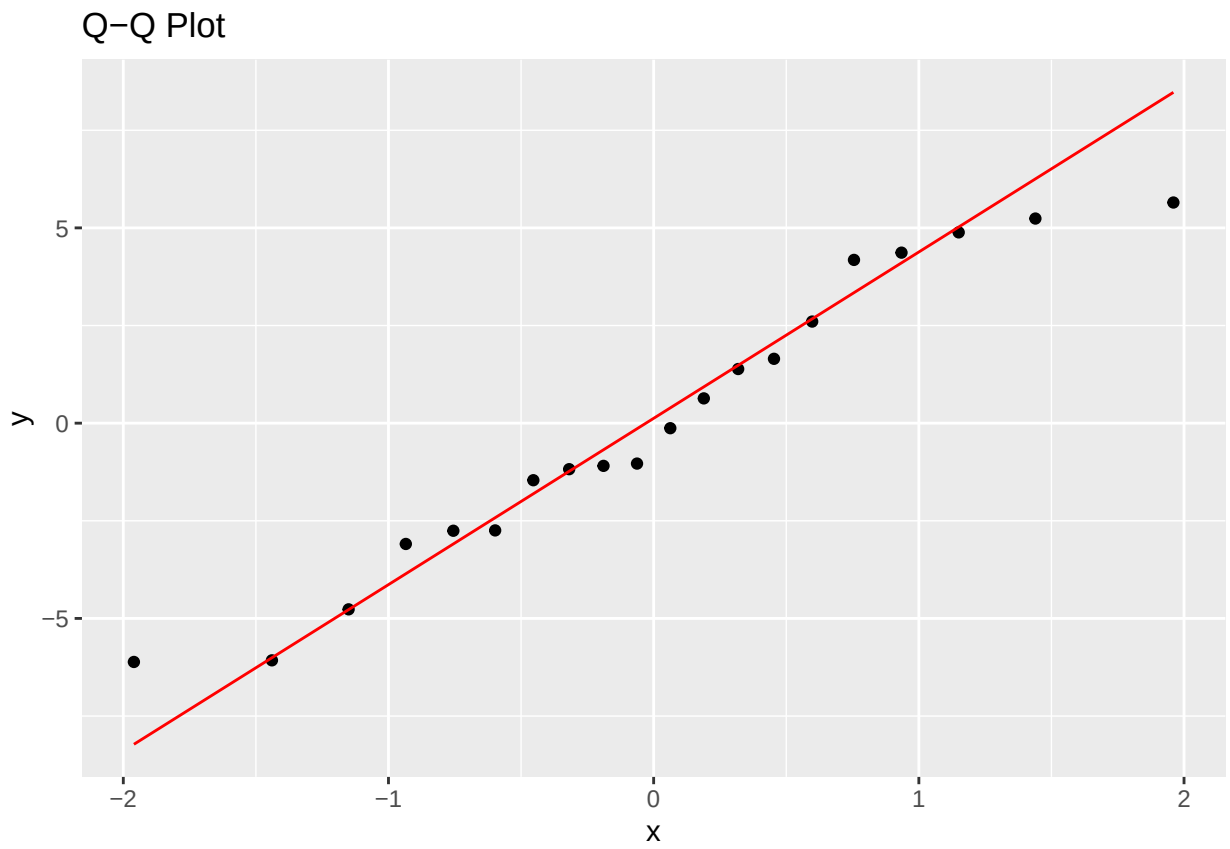
```
pp_check(bayesian_LMM, type = "scatter_avg_grouped", group = "treat")
```

```
## Using all posterior draws for ppc type 'scatter_avg_grouped' by default.
```



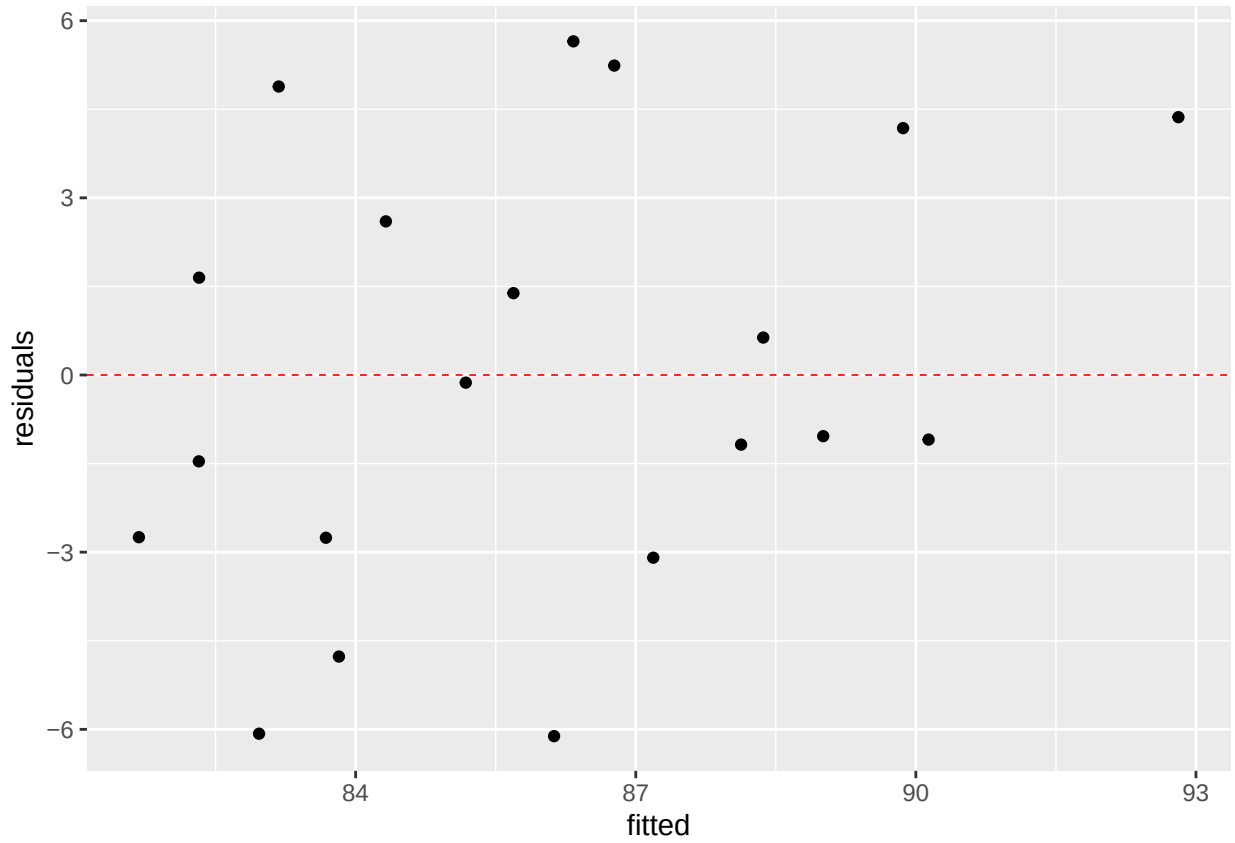

Overall, when comparing the predicted values from our posterior distribution to the observed values for yield we can see that the model generally captures the overall trend across the treatment groups and also within the treatment groups. However, the residuals are substantially large, so there is a lot of variance in the data that could not be captured by the model, as also indicated by the low $R^2$ and rather low ICC.

```
## Diagnostics

diagnostics_df <- penicillin

diagnostics_df$fitted <- fitted(bayesian_LMM)[, 1]
diagnostics_df$residuals <- residuals(bayesian_LMM)[, 1]

# Q-Q plot

diagnostics_df %>%
  ggplot(aes(sample = residuals)) +
  geom_qq() +
  geom_qq_line(colour = "red") +
  labs(title = "Q-Q Plot")
```

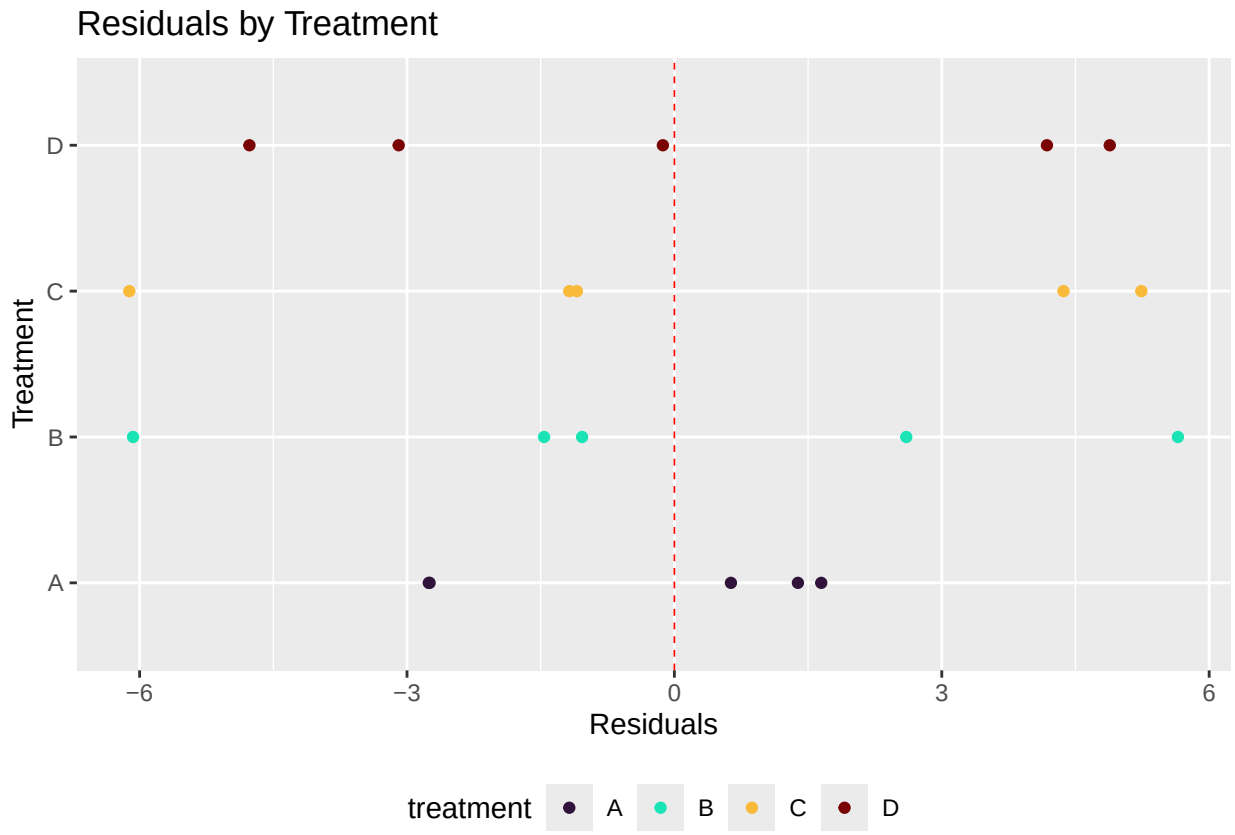



```
# Residuals vs fitted

diagnostics_df %>%
  ggplot() +
  geom_point(aes(x = fitted, y = residuals)) +
  geom_hline(yintercept=0, colour='red', linetype='dashed', size=0.3) +
  labs("Residuals vs. Fitted Values")
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```




```
# Residuals
diagnostics_df %>%
  ggplot() +
  geom_point(aes(x = residuals, y = treat, colour = treat)) +
  theme(legend.position = "bottom")  +
  scale_colour_viridis_d(option = "turbo") +
  guides(colour = guide_legend(title = "treatment")) +
  geom_vline(xintercept=0, colour='red', linetype='dashed', size=0.3) +
  labs(title = "Residuals by Treatment",
       x = "Residuals",
       y  = "Treatment")
```




When considering model diagnostic checks, the QQ-plot of the residuals of our model suggests that the errors are normally distributed, as we are looking for. Furthermore, the pattern in the distribution of the predicted values plotted against the residuals is random. However, when looking at the distribution of residuals within each treatment group, we observe that the spread of the errors in treatment A is considerably lower than in treatments B, C, and D. In conjunction with the PPC checks and initial EDA above, we surmise that this is driven by treatment A having an inherently lower within-group variance.

## Conclusions

We have used a Bayesian hierarchical model to assess the differences in impacts of 4 different processes on penicillin yield while controlling for differences in corn steep liquor quality. The experiment was completed using a randomized complete block design (RBCD), and so by blending enough blends of liquor to assign to each process within each block, the study design controls for differences in liquor blend quality, thus isolating for the differences in treatment effect. Exploratory data analysis showed that there was overlap in production between the treatments, but that some of the blends appeared to outperform others. The priors selected were based on summary statistics of the individual treatments, which were assumed to be normal, and of the blend blocks, with a Student's t-distribution used to capture all the between-group variance. As is standard, we assumed the within-group variance, $\sigma_\epsilon$, to be exponentially distributed, with rate parameter as $1/\hat{\sigma}$, $\hat{\sigma}$ being the sample standard deviation of yield.

The model was fit using `brms`, and there were no convergence issues observed, as the ESS values of each parameter were large and the $\hat{R}$ values were all 1. The behavior of the Markov chains was further validated using traceplots and autocorrelation plots, which indicated that the parameter space was thoroughly explored with no stagnation or multi-modal jumps from the sampler and sufficient decreases in autocorrelation within chains. This satisfied our requirements for MC chain convergence. Looking at the model summary, there is significant between-group differences with respect to the blend blocks, with an estimated standard deviation

of 4.42 and a 95% CI of $(0.63, 11.47)$. In contrast, there was no observed significant difference between the treatments relative to the cornerstone treatment (A), as all CIs included 0. This lack of explanatory power of the model contributed to a low $R^2$ value of roughly 0.43, indicating that the majority of variation seen in the yield cannot be explained by the model. Looking at model diagnostics, we used a posterior predictive check to validate that the model does capture the trend of the data, albeit with large residuals. These residuals were assessed to be normal and relatively homoskedastistic using a QQ plot and by plotting the residuals against the predicted values, both for the entire dataset and for individual treatment groups. This indicates that the model was set up properly and the model met the required assumptions for using a hierarchical model.

The analysis thus shows that there is no significant difference in penicillin yield between various treatment types relative to the baseline treatment A, of 84.06 units of penicillin, and rather that the differences in corn steep liquor blend drive these differences instead. The standard deviation between these blend groups was estimated to be 4.42 units of penicillin, although the lower 95% CI bound, at 0.63, is quite close to 0. This leads to the second conclusion, which is that the model has limited predictive value, and that there are most likely other factors impacting the yield of penicillin beyond the corn steep liquor blend and the production process chosen. This is supported by the large $\sigma_\epsilon$ value of 4.64, which is larger than $\sigma_\mu$, so more overall variance is explained from within-group factors as opposed to between group factors. As such, additional studies should investigate other factors potentially impacting final yield.