# Homework 10
# Problem One - Crossed Random Effects

*The `Penicillin` data set in `library(lme4)` is derived from Table 6.6, page 144, of Davies and Goldsmith (1972), where it is described as coming from an investigation to*

> assess the variability between samples of penicillin by the B. subtilis method. In this test method a bulk-innoculated nutrient agar medium is poured into a Petri dish of approximately 90 mm. diameter, known as a plate. When the medium has set, six small hollow cylinders or pots (about 4 mm. in diameter) are cemented onto the surface at equally spaced intervals. A few drops of the penicillin solutions to be compared are placed in the respective cylinders, and the whole plate is placed in an incubator for a given time. Penicillin diffuses from the pots into the agar, and this produces a clear circular zone of inhibition of growth of the organisms, which can be readily measured. The diameter of the zone is related in a known way to the concentration of penicillin in the solution.

*The variables in the data set are*

- *diameter: the diameter of the zone of inhibition due to penicillin*
- *plate: the plate (petri dish) in which the diameters are measured*
- *sample: one of the six areas in the dish where the diameter of the zone of inhibition was measured. Each area corresponds to a different penicillin solution; the same six solutions are used in each plate.*

*Use `str(Penicillin)`, `View(Penicillin)`, etc. to familiarize yourself with the data.*

*Because the samples are associated with the same six penicillin solutions in each plate, we will treat the sample and plate factors as crossed: for each plate, the same six different solution samples are tried.*

First we familiarize ourselves with the data.

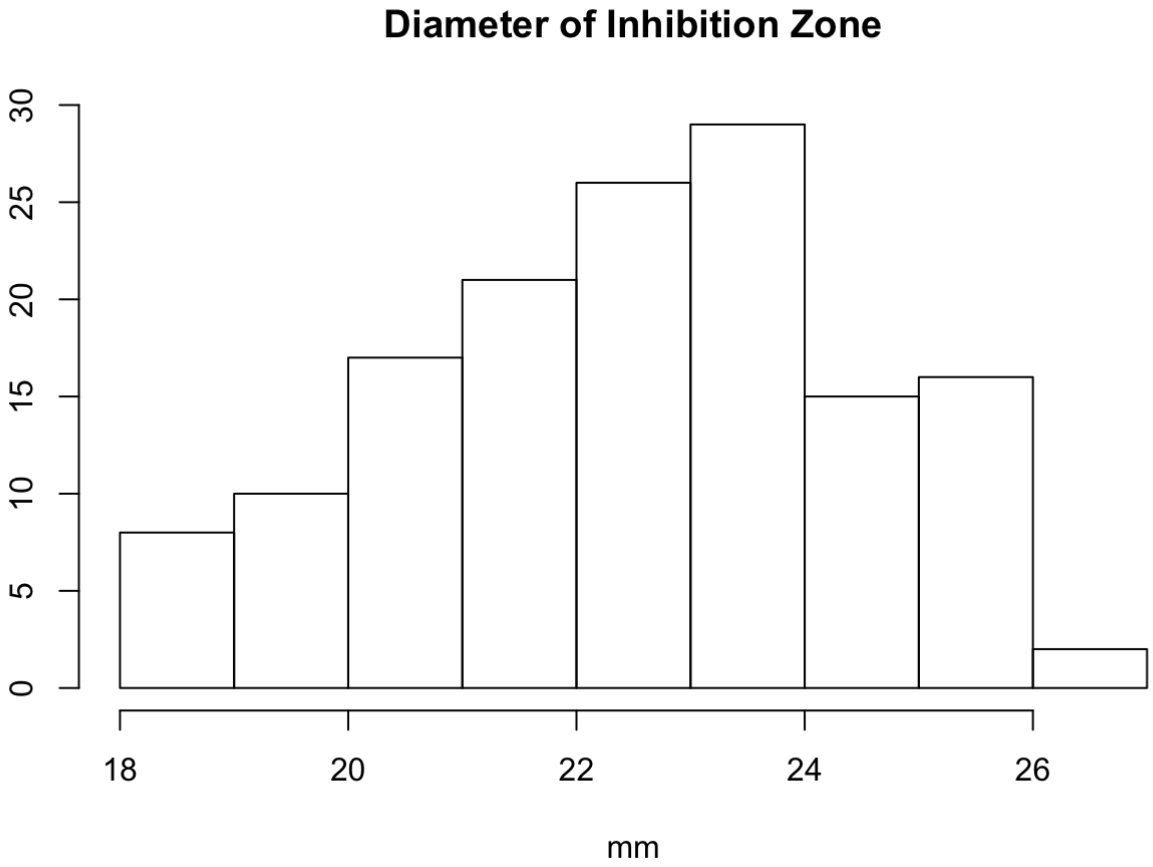
```
library(lme4)
```

```
## Loading required package: Matrix
## Loading required package: Rcpp
```

```
summary(Penicillin)
```

```
##     diameter      plate     sample
##  Min.   :18   a      :  6   A:24
##  1st Qu.:22   b      :  6   B:24
##  Median :23   c      :  6   C:24
##  Mean   :23   d      :  6   D:24
##  3rd Qu.:24   e      :  6   E:24
##  Max.   :27   f      :  6   F:24
##               (Other):108
```

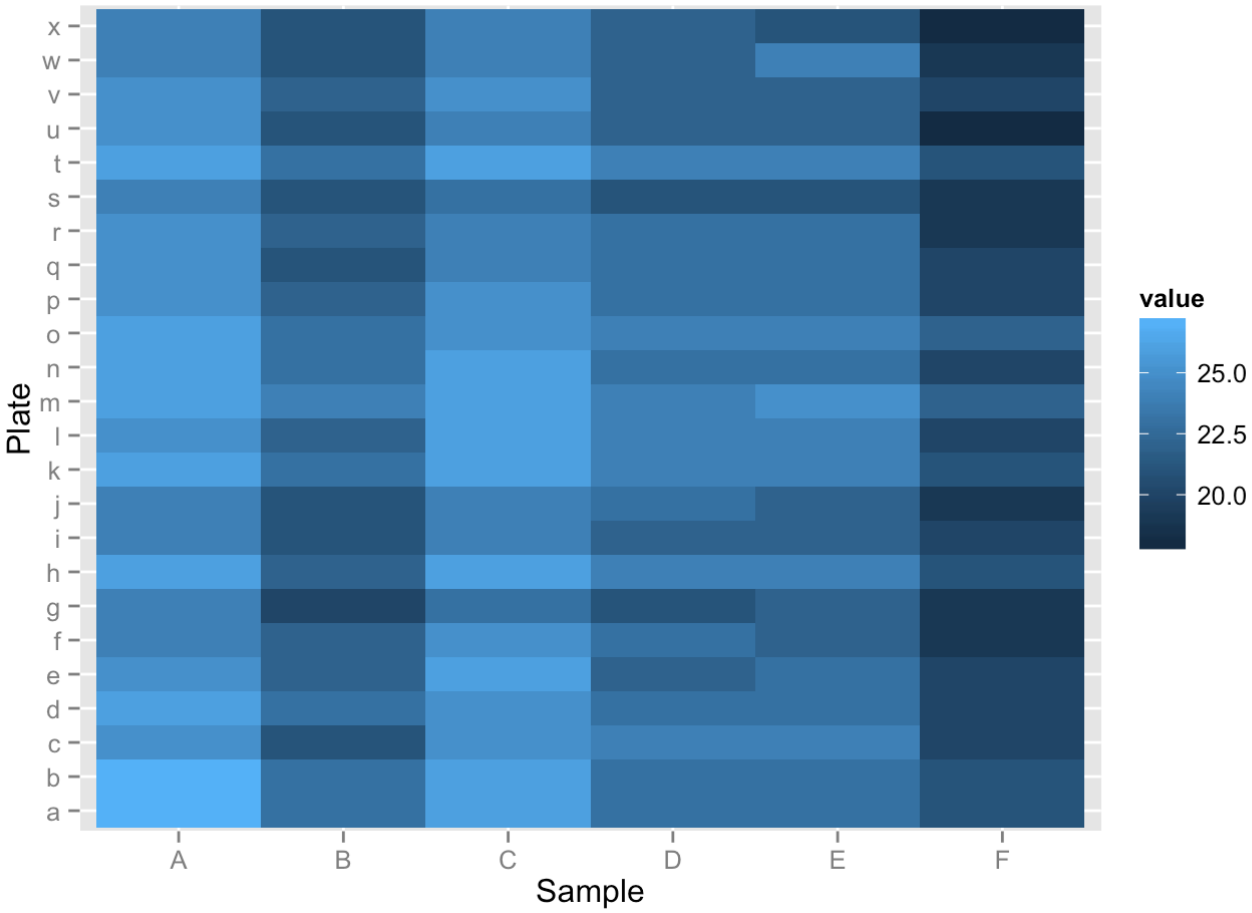
```
hist(Penicillin$diameter, main = "Diameter of Inhibition Zone",
      xlab = "mm", ylab = "")
```




# 1 (A)

*Devise a plot that shows the data nested within each unique pair of “plate” and “sample” levels for this completely crossed design. Provide your plot, and write a short paragraph explaining how to interpret this plot, in terms of the crossed factors “sample” and “plate”.*

```
library(reshape2)
library(ggplot2)

attach(Penicillin)
platesample_heatmap <- qplot(
    x = plate, y = sample,
    data = melt(diameter),
    fill = value, geom = "tile") +
    xlab("Plate") + ylab("Sample")
platesample_heatmap + coord_flip()
```




We made a heatmap showing diameters for each combination of plate and sample. The clearest features are 1. Sample looks like it matters more than plate. 2. Samples A and C seem to have unusually large diameters, while sample F has smaller diameters. 3. There is some variation across plates. Plate t has larger diameters than plate s, for example.

# 1 (B)

*Fit the additive linear model*

$$(diameter)_i = \beta_0 + \beta_{platej[i]} + \beta_{samplek[i]} + \epsilon_i, \epsilon_i \ N(0, \sigma^2)$$

*using* `Lm()` *in* `R`*, where the* $\beta$*'s are all fixed effects,* $\beta_{platej}$ *is the effect of plate* $j$ *and* $\beta_{samplek}$ *is the effect of sample* $k$*.*

*Provide a fitted model summary and one paragraph describing the model's fit and interpretation.*

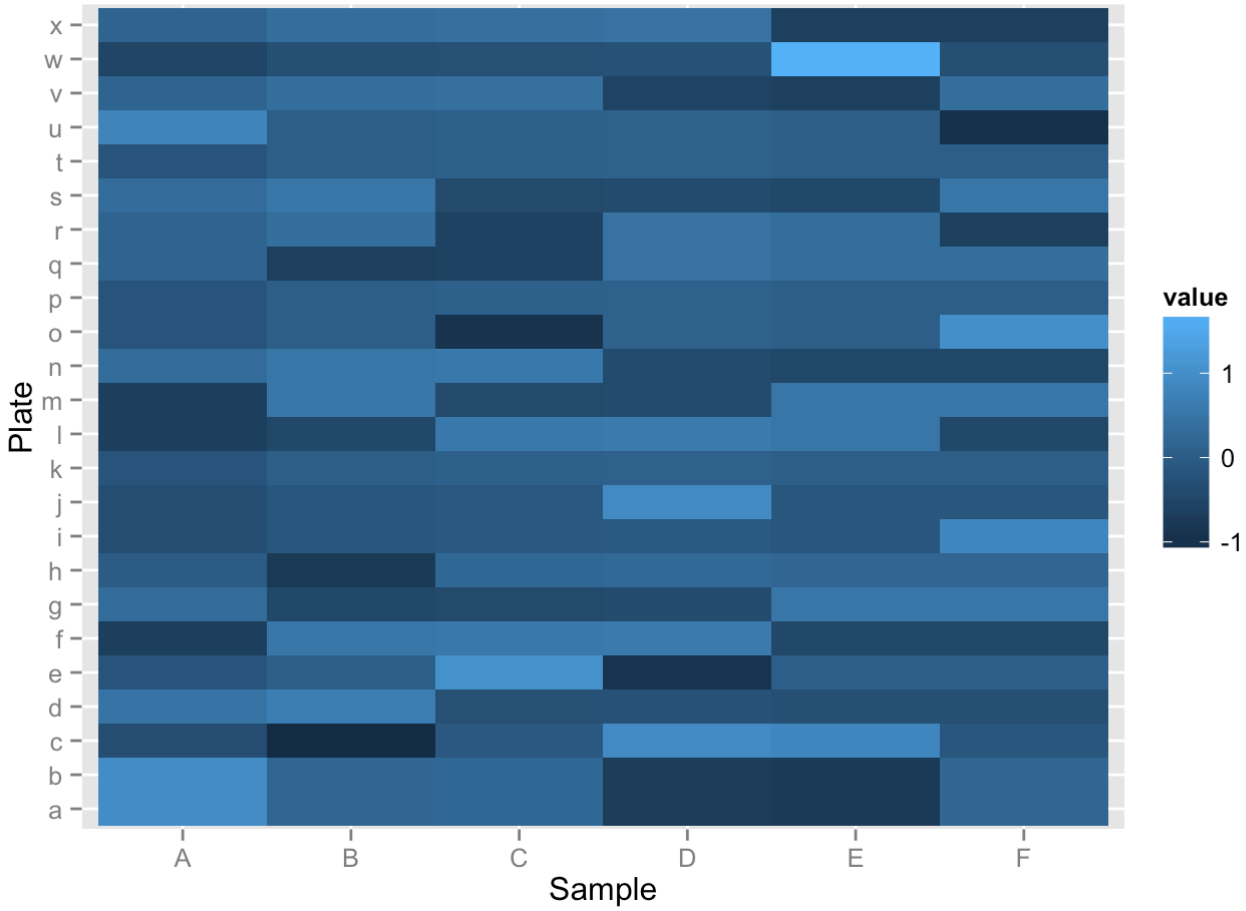
```
model1b <- lm(diameter ~ plate + sample)

summary(model1b)
```

```
##
## Call:
## lm(formula = diameter ~ plate + sample)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1528 -0.3715  0.0139  0.3472  1.6806
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.60e+01   2.47e-01  105.47  < 2e-16 ***
## plateb       5.50e-14   3.17e-01    0.00  1.00000
## platec      -6.67e-01   3.17e-01   -2.10  0.03794 *
## plated      -5.00e-01   3.17e-01   -1.57  0.11805
## platee      -8.33e-01   3.17e-01   -2.62  0.00985 **
## platef      -1.33e+00   3.17e-01   -4.20  5.3e-05 ***
## plateg      -2.33e+00   3.17e-01   -7.35  3.1e-11 ***
## plateh       5.01e-14   3.17e-01    0.00  1.00000
## platei      -1.67e+00   3.17e-01   -5.25  7.1e-07 ***
## platej      -1.67e+00   3.17e-01   -5.25  7.1e-07 ***
## platek       1.67e-01   3.17e-01    0.52  0.60064
## platel      -3.33e-01   3.17e-01   -1.05  0.29598
## platem       6.67e-01   3.17e-01    2.10  0.03794 *
## platen      -3.33e-01   3.17e-01   -1.05  0.29598
## plateo       1.67e-01   3.17e-01    0.52  0.60064
## platep      -8.33e-01   3.17e-01   -2.62  0.00985 **
## plateq      -1.17e+00   3.17e-01   -3.67  0.00036 ***
## plater      -1.17e+00   3.17e-01   -3.67  0.00036 ***
## plates      -2.33e+00   3.17e-01   -7.35  3.1e-11 ***
## platet       1.67e-01   3.17e-01    0.52  0.60064
## plateu      -1.83e+00   3.17e-01   -5.77  6.7e-08 ***
## platev      -1.17e+00   3.17e-01   -3.67  0.00036 ***
## platew      -1.50e+00   3.17e-01   -4.72  6.6e-06 ***
## platex      -2.17e+00   3.17e-01   -6.82  4.4e-10 ***
## sampleB     -3.21e+00   1.59e-01  -20.21  < 2e-16 ***
## sampleC     -2.50e-01   1.59e-01   -1.57  0.11805
## sampleD     -2.29e+00   1.59e-01  -14.44  < 2e-16 ***
## sampleE     -2.21e+00   1.59e-01  -13.91  < 2e-16 ***
## sampleF     -5.21e+00   1.59e-01  -32.81  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.55 on 115 degrees of freedom
## Multiple R-squared:  0.941,  Adjusted R-squared:  0.927
## F-statistic: 65.6 on 28 and 115 DF,  p-value: <2e-16
```

The model here treats plate $a$ and sample $A$ as the baseline. Every sample is significant other than C, which does look very much like our baseline A. Some plates are highly significant, some not at all, seems to match the plot.

We also look at a heatmap of the residuals. There's a bit of structure, but they seem reasonably random.

```
residuals_heatmap <- qplot(
    x = plate, y = sample,
    data = melt(model1b$residuals),
    fill = value, geom = "tile") +
    xlab("Plate") + ylab("Sample")
residuals_heatmap + coord_flip()
```




# One (C)

*In an experiment like this, the effect of* `plate` *is not really of interest, so we could model it as a random effect. We might also be more concerned about the variability among the samples, rather than the effect of each particular sample type, so we could model sample as a random effect also. Use* `lmer()` *to fit a random intercept model to the data, with random effects for plate and sample:´*

$$\textbf{Level1 :}$$
$$(diameter)_i = \beta_0 + \alpha_{platej[i]} + \alpha_{samplek[i]} + \epsilon_i, \qquad \epsilon_i \sim N(0, \sigma^2) i = samples$$
$$\textbf{Level2 :}$$
$$\alpha_{platej} = 0 + \eta_{platej}, \qquad \eta_{platej} \sim N(0, \tau^2_{plate}) j = plates$$
$$\alpha_{samplek} = 0 + \eta_{samplek}, \qquad \eta_{samplek} \sim N(0, \tau^2_{sample}) k = samples$$

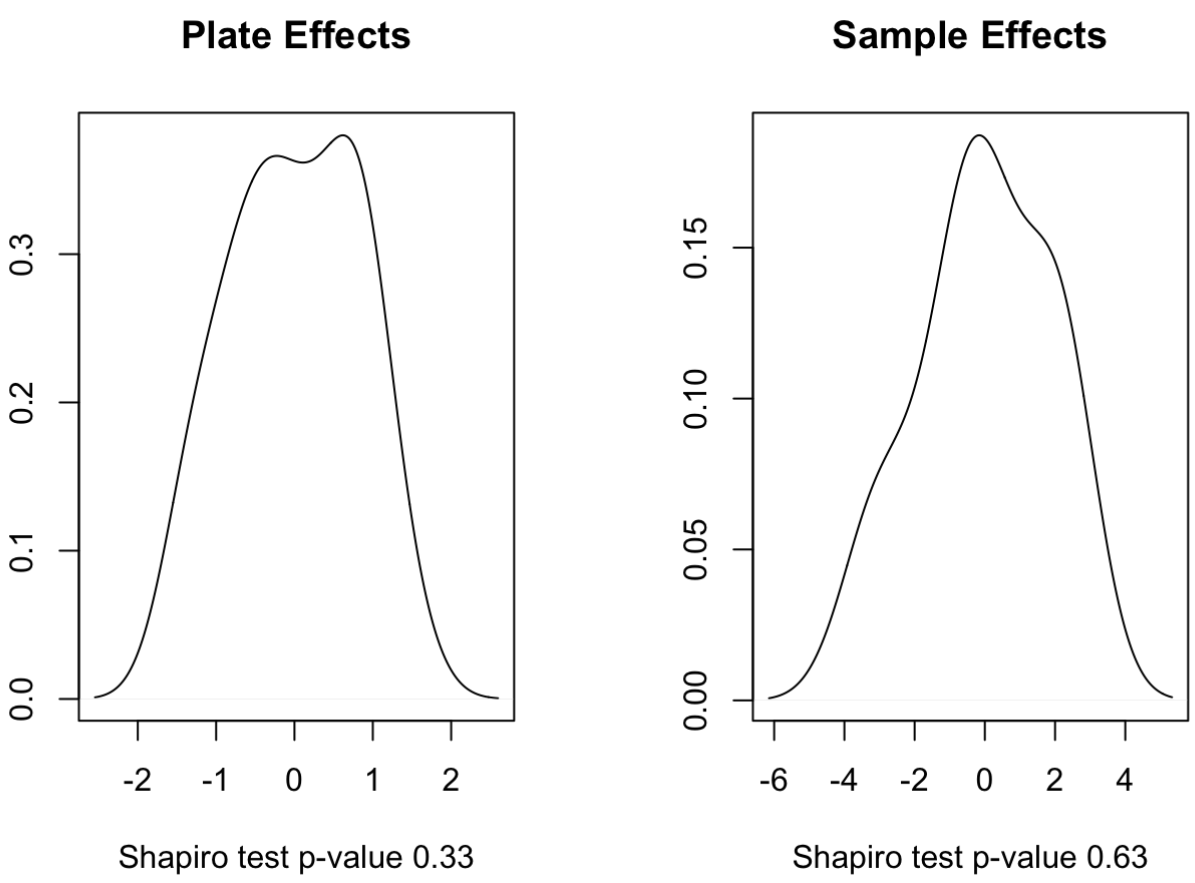
*where* $\beta_0$ *is a fixed effect, and the random effects* $\alpha_{platej}$ *and* $\alpha samplek$ *are all centered at zero. Since the factors that define the random effects are crossed, this is a* crossed random effects *model.*

*Provide a fitted model summary and one paragraph describing the model's fit and interpretation.*

```
model1c <- lmer(diameter ~ (1|plate) + (1|sample))

summary(model1c)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: diameter ~ (1 | plate) + (1 | sample)
##
## REML criterion at convergence: 330.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.0792 -0.6714  0.0629  0.5838  2.9796
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  plate    (Intercept) 0.717    0.847
##  sample   (Intercept) 3.731    1.932
##  Residual             0.302    0.550
## Number of obs: 144, groups:  plate, 24; sample, 6
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   22.972      0.809    28.4
```

```
#  Calculate ICC for plate and sample
tau2plate <- as.numeric(summary(model1c)$varcor)[1]
tau2sample <- as.numeric(summary(model1c)$varcor)[2]
sigma2 <- summary(model1c)$sigma^2

icc_plate <- tau2plate / (tau2plate + tau2sample + sigma2)
icc_sample <- tau2sample / (tau2plate + tau2sample + sigma2)

#  Check if random effects are normally distributed
plate_effects <- unlist(ranef(model1c)$plate)
sample_effects <- unlist(ranef(model1c)$sample)

normpval_plate <- shapiro.test(plate_effects)$p.value
normpval_sample <- shapiro.test(sample_effects)$p.value

platelab <- paste("Shapiro test p-value",
                  round(normpval_plate, 2))
samplelab <- paste("Shapiro test p-value",
                   round(normpval_sample, 2))

par(mfrow = c(1,2))
plot(density(plate_effects), main = "Plate Effects",
      xlab = platelab, ylab = "")
plot(density(sample_effects), main = "Sample Effects",
     xlab = samplelab, ylab = "")
```



The `sample` component has by far the largest influence. We can calculate an intraclass correlation of 0.1509 for the plates and a very high 0.7854 for the samples.

Surprisingly, the random effects actually do appear reasonably normally distributed.

# One (D)

*Use AIC and BIC to assess whether each of the random effects is needed in the mixed effects model.*

```
emptymodel <- lm(diameter ~ 1)
justplate <- lmer(diameter ~ (1|plate))
justsample <- lmer(diameter ~ (1|sample))

#  Adding plate first
anova(justplate, model1c)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data:
## Models:
## justplate: diameter ~ (1 | plate)
## model1c: diameter ~ (1 | plate) + (1 | sample)
##           Df AIC BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## justplate  3 618 627   -306      612
## model1c    4 340 352   -166      332   279      1     <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#  Adding sample first
anova(justsample, model1c)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data:
## Models:
## justsample: diameter ~ (1 | sample)
## model1c: diameter ~ (1 | plate) + (1 | sample)
##            Df AIC BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## justsample  3 443 452   -219      437
## model1c     4 340 352   -166      332   105      1     <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Starting with either random effect, ANOVA tells us to add the other. We can also check AIC and BIC for all models.

```
modelslist <- list(emptymodel, justplate, justsample, model1c)

aics <- sapply(modelslist, AIC)
bics <- sapply(modelslist, BIC)

aics
```

```
## [1] 615.7 619.3 441.9 338.9
```

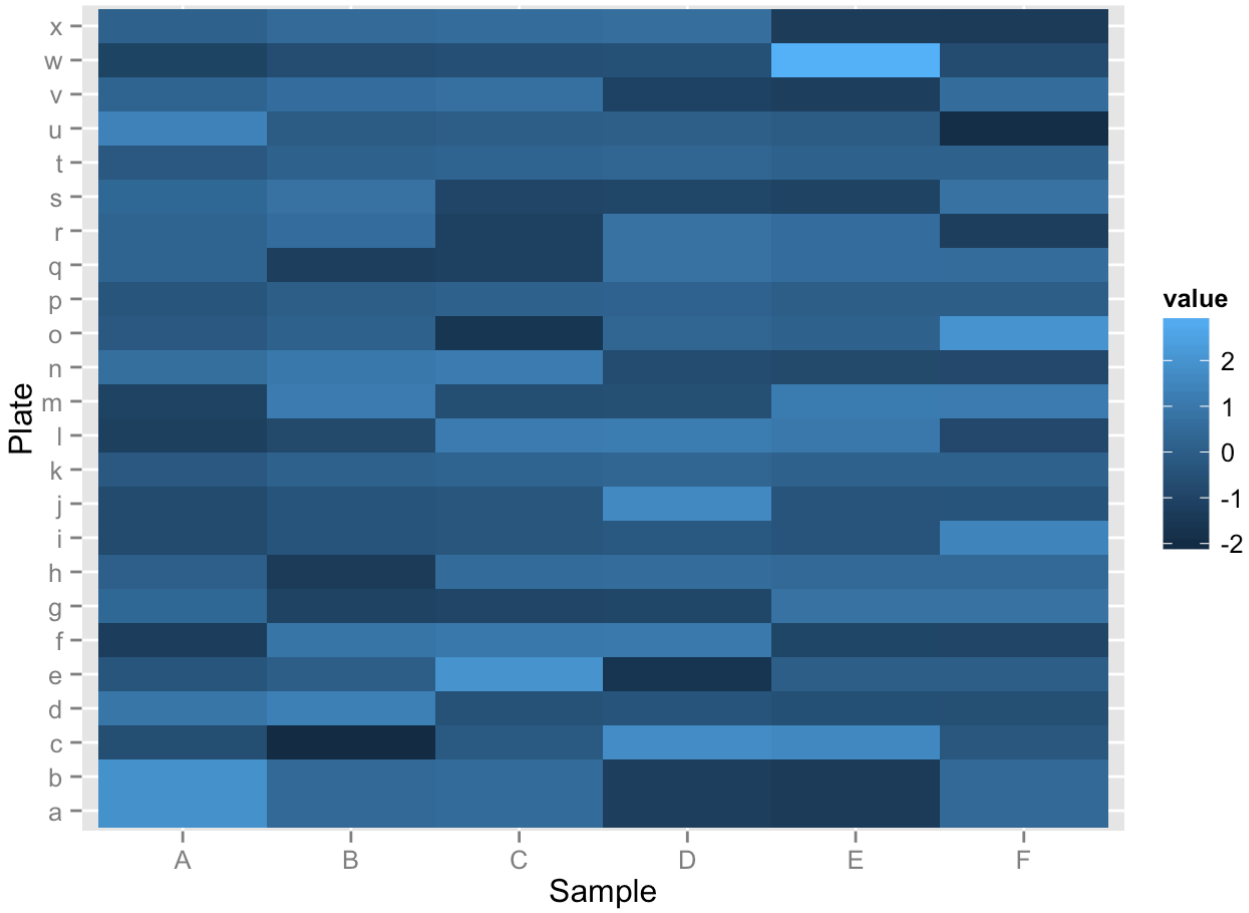
```
bics
```

```
## [1] 621.7 628.2 450.8 350.7
```

Both AIC and BIC would say there is not enough variation in the plates themselves to warrant including them as a random effect, when looking at no effects vs including the plates.

But both prefer a model with sample over a model without sample, and when you consider adding a plate effect to a model already haveing a sample effect, both criteria would tell you to add it. So AIC and BIC support including both random effects.

```
residuals_heatmap1c <- qplot(
    x = plate, y = sample,
    data = melt(summary(model1c)$residuals),
    fill = value, geom = "tile") +
    xlab("Plate") + ylab("Sample")
residuals_heatmap1c + coord_flip()
```




```
detach(Penicillin)
```

# Problem Two - *Nested Random Effects*

*The `Pastes` data set in the `Lme4` package also comes from Davies and Goldsmith (1972 table 6.5 page 138). They describe the data as coming from*

> deliveries of a chemical paste product contained in casks where, in addition to sampling and testing errors, there are variations in quality between deliveries. . . As a routine, three casks selected at random from each delivery were sampled and the samples were kept for reference. . . Ten of the delivery batches were sampled at random and two analytical tests carried out on each of the 30 samples.

*(That is to say two identical tests were performed on each sample.) The variables in the data set are:*

```
- *stength: strength of the paste sample in each anlytical test*
- *batch: which delivery batch the paste sample same from*
- *cask: which cask within the delivery batch the paste sample came from*
- *sample: another identifier for the cask and batch that each paste sample came from*
```

*Familiarize yourself with the data.*

*Note that the casks are* nested *within the delivery batches; each sample can be in one and only one cask, and each cask can be in one and only one delivery batch. Also, the cask labels are not unique; they are repeated within each batch.* If we did not know that cask was nested within batch, this might lead us to belive that the cask and batch factors

were crossed instead of nested.

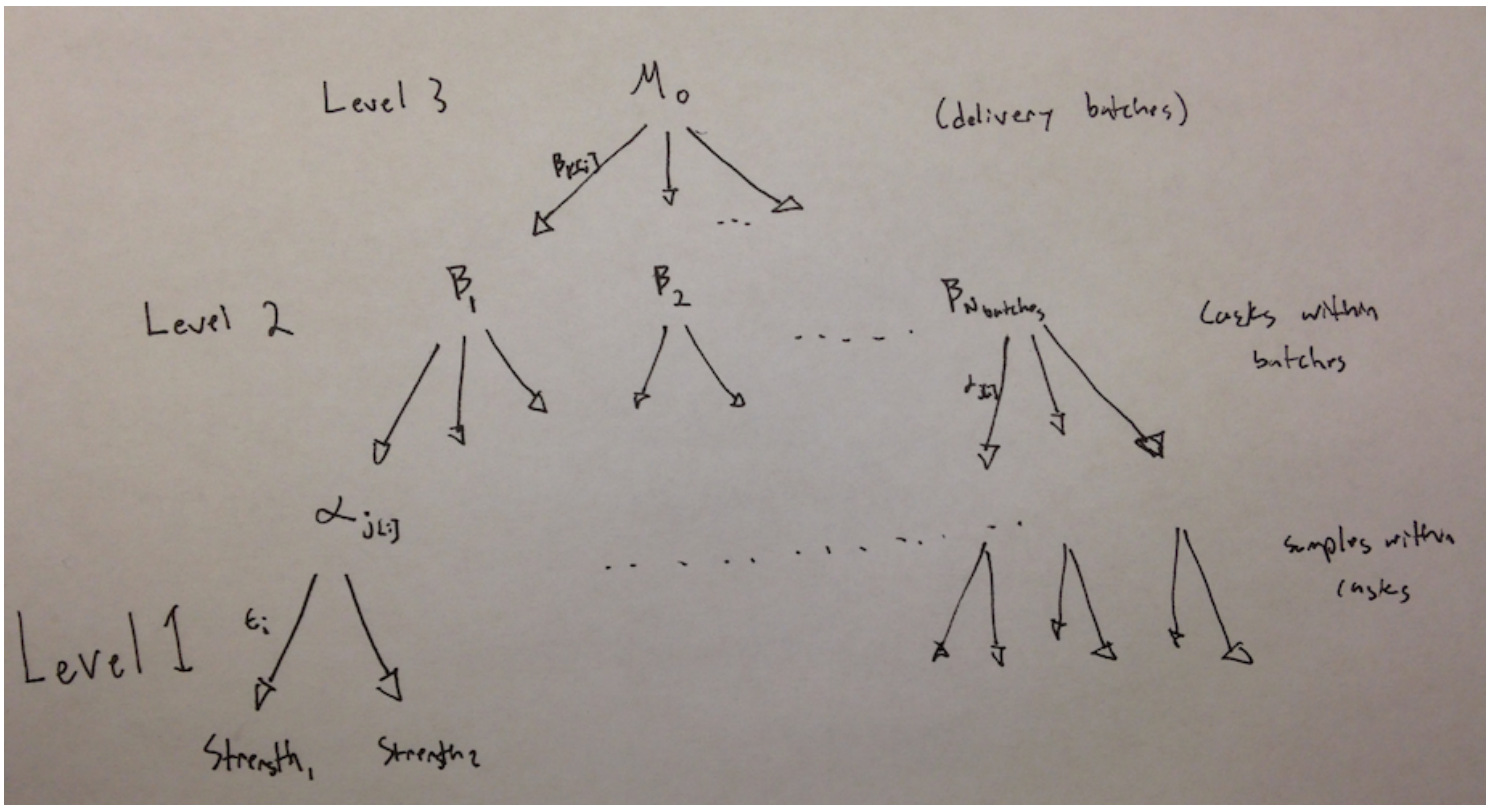
*A random intercepts model for this data with casks nested within batches might look like this:*

$$
\begin{aligned}
&\textbf{Level1 :} \\
(\text{strength})_i &= \alpha_{j[i]} + \epsilon_i, \qquad \epsilon_i \sim N(0, \sigma^2) i = \text{samples within cask} \\
&\textbf{Level2 :} \\
\alpha_j &= \beta_{k[j]} + \eta_{2j}, \qquad \eta_{2j} \sim N(0, \tau^2) j = \text{cask within delivery batch} \\
&\textbf{Level3 :} \\
\beta_k &= \mu_0 + \eta_{3k}, \qquad \eta_{3k} \sim N(0, \omega^2) k = \text{delivery batch}
\end{aligned}
$$

*Here, the $\alpha$s are random effects at level 1, the $\beta$s are random effects at level 2, and $\mu_0$ is a fixed effect (the grand mean).*

# Two (A)

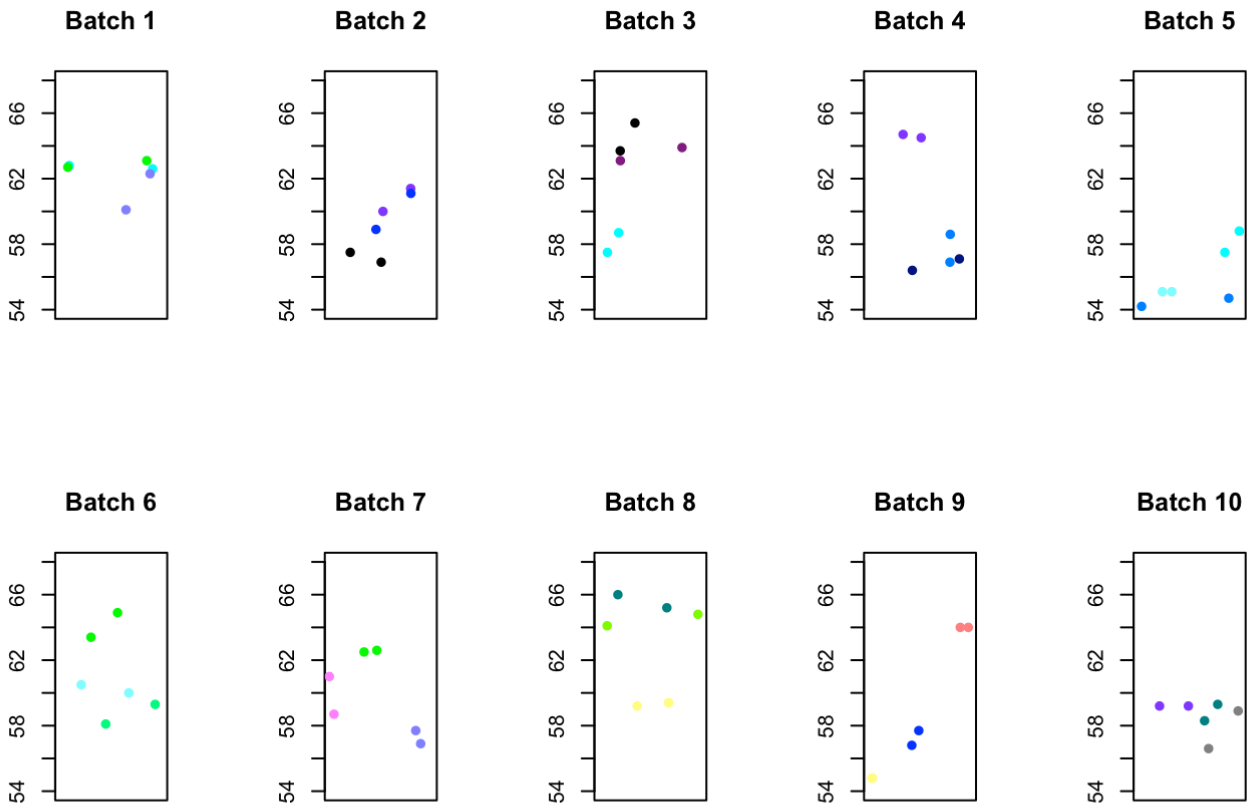
*Make a tree diagram showing how each pair of observations is nested within each cask, and the casks are nested within each delivery batch. Label the branches of the tree with the appropriate parameters from the multi-level model above.*




# Two (B)

*Make a plot showing the raw data in each cask, and each cask within each sample.*

```
attach(Pastes)

#  Set up ten plots, one for each batch.
par(mfrow = c(2,5))

#  For each plot, give the casks a random color and plot the strengths.
batches <- unique(batch)
casks <- unique(cask)

library(colorRamps)

for (i in 1:10) {
  curbatch = batches[i]
  curdata <- Pastes[Pastes$batch == curbatch, ]

  plot(c(0,1), c(0,0), xlim = c(0,1), ylim = c(54, 68),
        xlab = "", ylab = "", main = paste("Batch", i),
       xaxt = "n")
  threecolors <- sample(primary.colors(), 3, replace = FALSE)

  #  Use random colors and a random x location.
  for (caskj in 1:3) {
    curcask = casks[caskj]
    cursamples <- curdata[curdata$cask == curcask, ]
    points(runif(2), cursamples$strength,
         col = threecolors[caskj], pch = 16)

  }

}
```




This plot shows every batch seperately, and then plots each cask as a distinct random color.

# Two (C)

*Build a random intercept model for this data. Use the term* `(1 | cask:batch)` *to force* `lmer()` *to treat* `cask` *as nested within* `batch`.

```
model2c <- lmer(strength ~ (1|batch) + (1|cask:batch))
summary(model2c)
```

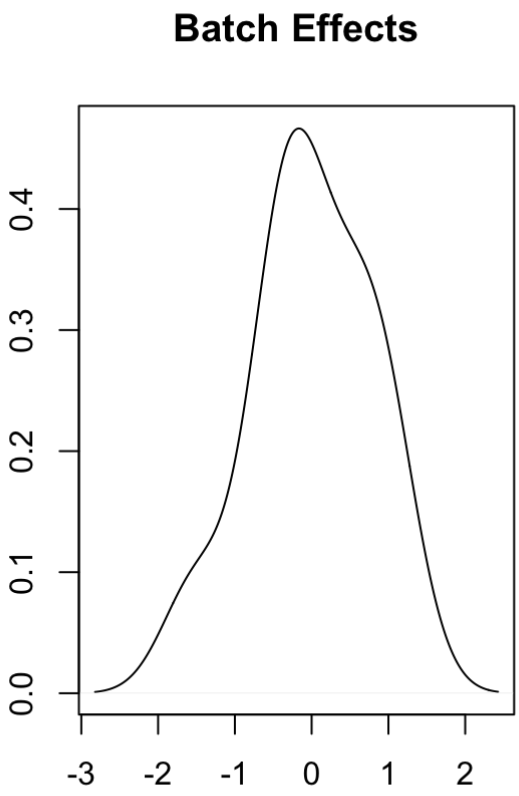
```
## Linear mixed model fit by REML ['lmerMod']
## Formula: strength ~ (1 | batch) + (1 | cask:batch)
##
## REML criterion at convergence: 247
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.4798 -0.5156  0.0095  0.4720  1.3897
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  cask:batch (Intercept) 8.434    2.904
##  batch      (Intercept) 1.657    1.287
##  Residual               0.678    0.823
## Number of obs: 60, groups:  cask:batch, 30; batch, 10
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   60.053      0.677    88.7
```

It looks like the seperate casks are the source of most of the variation in strength. We'll also check the normality of our random effects.

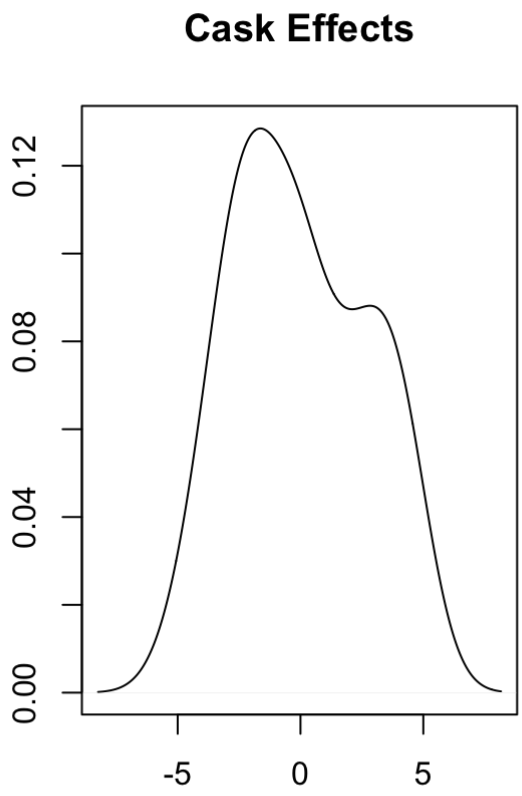
```
batchtau2 <- as.numeric(summary(model2c)$varcor)[2]
casktau2 <- as.numeric(summary(model2c)$varcor)[1]
sigma2 <- summary(model2c)$sigma^2

icc_batch <- batchtau2 / (batchtau2 + casktau2 + sigma2)
icc_cask <- casktau2 / (batchtau2 + casktau2 + sigma2)
```

We find an ICC in batches of 0.1539, and a higher ICC for casks, 0.7831.

```
batcheffects <- unlist(ranef(model2c)$batch)
caskeffects <- unlist(ranef(model2c)$"cask:batch")

batch_normpval <- shapiro.test(batcheffects)$p.value
cask_normpval <- shapiro.test(caskeffects)$p.value

batch_label <- paste("Shapiro test p-value",
                     round(batch_normpval, 2))
cask_label <- paste("Shapiro test p-vaue",
                    round(cask_normpval, 2))


par(mfrow = c(1,2))
plot(density(batcheffects), main = "Batch Effects",
     xlab = batch_label, ylab = "")
plot(density(caskeffects), main = "Cask Effects",
     xlab = cask_label, ylab = "")
```




Shapiro test p-value 0.8




Shapiro test p-vaue 0.15

# Two (D)

*The `sample` variable in the `Pastes` dataset provides unique labels for each cask within each batch. So the model `lmer(strength ~ 1 + (1|sample) + (1|batch))` should produce the same fit and parameter estimates as your model in part 2c. Fit this model and verify that it agrees with the previous model.*

```
model2d <- lmer(strength ~ (1|batch) + (1|sample))
summary(model2d)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: strength ~ (1 | batch) + (1 | sample)
##
## REML criterion at convergence: 247
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.4798 -0.5156  0.0095  0.4720  1.3897
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  sample   (Intercept) 8.434    2.904
##  batch    (Intercept) 1.657    1.287
##  Residual             0.678    0.823
## Number of obs: 60, groups:  sample, 30; batch, 10
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   60.053      0.677    88.7
```

The summary output looks exactly the same.

# Two (E)

*Use AIC and BIC to assess whether each of the random effects is needed in the mixed effects model.*

```
justbatch <- lmer(strength ~ (1|batch))
justcask <- lmer(strength ~ (1|sample))
noeffects <- lm(strength ~ 1)

models_list <- list(noeffects, justbatch, justcask, model2d)

aics <- sapply(models_list, AIC)
bics <- sapply(models_list, BIC)

aics
```

```
## [1] 314.3 307.6 253.6 255.0
```

```
bics
```

```
## [1] 318.5 313.9 259.9 263.4
```

The selection criteria agree that the best model is the one with random effects *only* for casks. The model with just a batch effect is preferred by both over the empty model, but this is likely just due to batch capturing variation that would otherwise be covered by cask IDs. Both models prefer just casks over casks and batches.

# Problem Three - *Fake Data Testing*

*In problem 2e, you used AIC and BIC to decide whether to keep the batch random effect. In this problem we will construct a fake data test to confirm or disconfirm the approximate inference you made with AIC and BIC.*

*Let $m_1, m_2, \ldots, m_B$ be the mean strength in each of the $B$ batches, let $\bar{m} = \frac{1}{B}\sum_{b=1}^{B} m_b$ be their mean, and let $T(Pastes) = \frac{1}{B-1}\sum_{b=1}^{B}(m_b - \bar{m})^2$, the sample variance of the batch means. This will be our test statistic - if it is small, we do not need the batch random effect; if it is large, we do need the batch random effet. For 1000 fake data sets simulated from the $H_0$ model (without the batch random effect) we will calculate $T(fake.Pastes) $ and compare the observed value $T(Pastes)$ to the distribution of simulated $T(fake.Pastes)$ values to determine if $T(pastes)$ is small or large.*

# 3 (A)

*Compute $T(Pastes)$.*

```
gett <- function(strengthlist) {
  #  I'm always assuming this is in the same
  #  order as in the original data frame.
  means <- aggregate(strengthlist, list(batch), mean)[,2]
  ng <- length(unique(batch))
  return ((ng/(ng-1)) * var(means))
}

tstat <- gett(strength)
```

We get $T(pastes)$ = 5.0906.

# 3 (B)

*Use the `sim()` function from `library(arm)` to generate 1000 sets of plausible parameter values from the model that omits the batch random effect.*

```
library(arm)
```

```
## Loading required package: MASS
##
## arm (Version 1.7-07, built: 2014-8-27)
##
## Working directory is /Users/patrickfoley/Box Sync/463 TAing
```

```
justcasks <- lmer(strength ~ 1 + (1|sample))
newparameters <- sim(justcasks, 1000)
```

# 3 (C)

*Use the `fitted()` function to generate 1000 fake data sets of 60 observations each from the 1000 sets of parameter values you produced.*

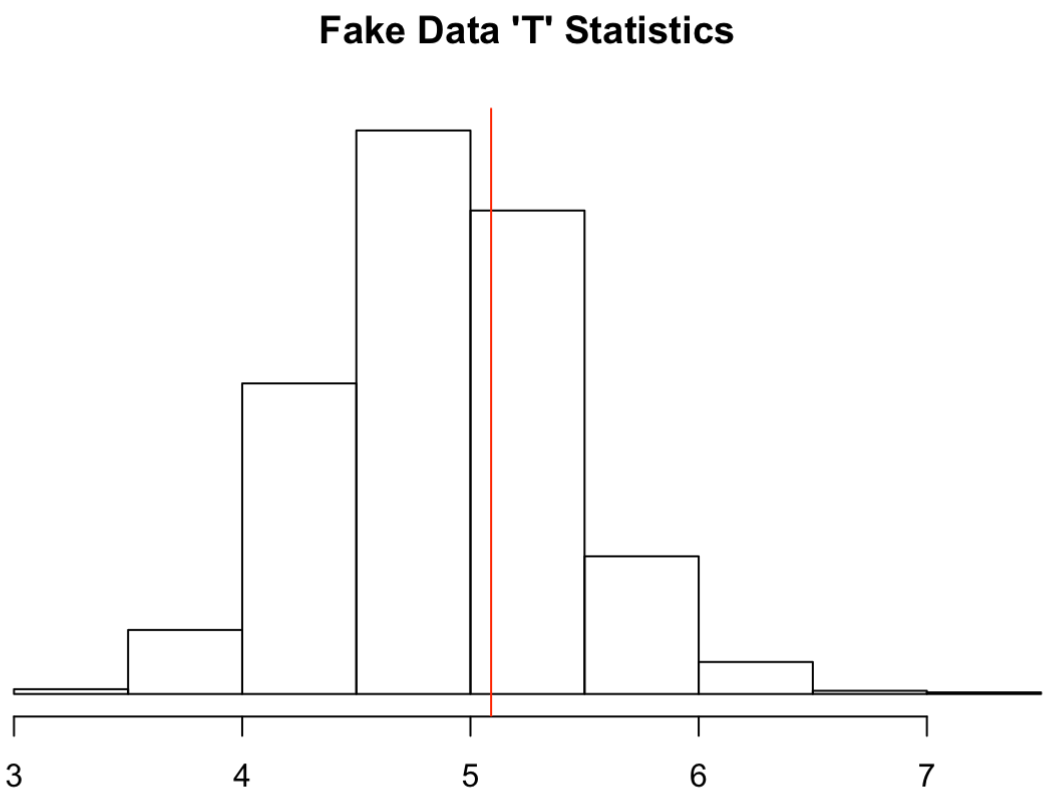
```
newdata <- fitted(newparameters,  justcasks)
dim(newdata)
```

```
## [1]   60 1000
```

# 3 (D)

*Find $T(`fake.data[,i]`)$ for each column $i$ of the matrix of simulted fake data sets, and make a histogram of the result. Plot $T(Pastes)$ sa a vertical red line on the histogram. Does the fake-data test reject $H_0$? Why? Is this consistent with the AIC and BIC results?*

```
# Using the function we defined earlier:

newts <- apply(newdata, 2, gett)

hist(newts, main = "Fake Data 'T' Statistics",
     xlab = "", ylab = "", yaxt = "n")
abline(v = tstat, col = "red")
```




```
quantiletstat <- mean(newts < tstat)
```

From the histogram, our real data's $T$ doesn't look very high at all. It's in the 0.653 quantile. We can't reject $H_0$ (that we do not need the `(1|batch)` effect) with any confidence.

This agrees with the AIC and BIC results.