Module 7 Lab Submission

Ben Tankus

The `Resin` data set and the following data description come from Gary Oehlert's book *A First Course in Design and Analysis of Experiments*:

"The oleo resin of trees is obtained by cutting a tapping gash in the bark and removing the resin that collects there. Acid treatments can also improve collection. In this experiment, four trees (dipterocarpus kerrii) will be tapped seven times each. Each of the tappings will be treated with a difference strength of sulfuric acid (0, 2.5, 5, 10, 15, 25, and 50% strength), and the resin collected from each tapping is the response (in grams, data from Bin Jantan, Bin Ahmad, and Bin Admad 1987)."

First we read in the data, and then convert it to a `groupedData` object using the function `groupedData()`, which is part of the `nlme` package. Note that we will be treating the acid strength as a categorical (factor) variable rather than as a quantitative variable. We indicate this with the `as.factor()` function applied to the `strength` variable.

```
Resin <- read.csv("Resin.csv")
Resin_group <- groupedData(yield ~ as.factor(strength) | tree, data=Resin,
                            order.groups=FALSE)
```

1. **Plot the grouped data object `Resin_group`.**

```
plot(Resin_group)
```

0   5   15   50
2.5   10   25

tree

4

3

2

1

0   100   200   300

yield
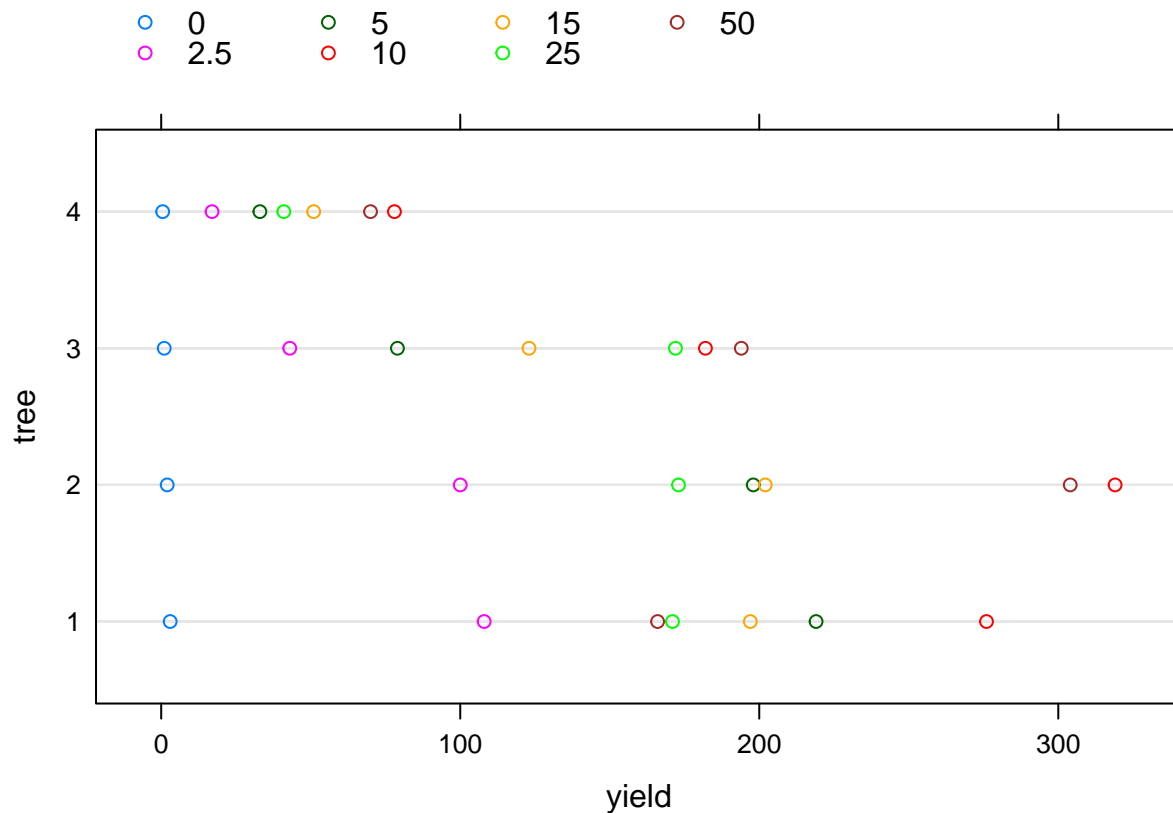
2. Use the `lme()` function to fit a mixed effects model with `strength` as a fixed effect and `tree` as a random effect. Be sure to treat `strength` as a categorical variable by using the `factor()` function around the `strength` variable in your formula.

```
lmeFit <- lme(yield ~ as.factor(strength) , data = Resin_group, random = ~ 1 | tree)
```

3. Use the `summary()` function to summarize the mixed effects model you fit above.

```
summary(lmeFit)
```

```
## Linear mixed-effects model fit by REML
##  Data: Resin_group
##        AIC      BIC    logLik
##   254.5835 263.9842 -118.2918
##
## Random effects:
##  Formula: ~1 | tree
##         (Intercept) Residual
## StdDev:    61.4961 44.35169
##
## Fixed effects: yield ~ as.factor(strength)
##                        Value Std.Error DF  t-value p-value
## (Intercept)            1.625  37.91056 18 0.042864  0.9663
## as.factor(strength)2.5 65.375  31.36138 18 2.084570  0.0516
```

```
## as.factor(strength)5    130.625   31.36138 18 4.165155   0.0006
## as.factor(strength)10   212.125   31.36138 18 6.763892   0.0000
## as.factor(strength)15   141.625   31.36138 18 4.515904   0.0003
## as.factor(strength)25   137.625   31.36138 18 4.388359   0.0004
## as.factor(strength)50   181.875   31.36138 18 5.799330   0.0000
##  Correlation:
##                          (Intr) a.()2. as.()5 a.()10 a.()15 a.()25
## as.factor(strength)2.5 -0.414
## as.factor(strength)5    -0.414  0.500
## as.factor(strength)10  -0.414  0.500  0.500
## as.factor(strength)15  -0.414  0.500  0.500  0.500
## as.factor(strength)25  -0.414  0.500  0.500  0.500  0.500
## as.factor(strength)50  -0.414  0.500  0.500  0.500  0.500  0.500
##
## Standardized Within-Group Residuals:
##        Min         Q1        Med         Q3        Max
## -1.2914379 -0.4946467 -0.1293257  0.5288214  1.7439594
##
## Number of Observations: 28
## Number of Groups: 4
```

4. **Use the `intervals()` function to construct confidence intervals for the fixed effect coefficient parameters and the random effect standard deviation parameter.**

```
intervals(lmeFit)
```

```
## Approximate 95% confidence intervals
##
##  Fixed effects:
##                            lower     est.      upper
## (Intercept)           -78.0221312    1.625   81.27213
## as.factor(strength)2.5  -0.5128163   65.375  131.26282
## as.factor(strength)5    64.7371837  130.625  196.51282
## as.factor(strength)10  146.2371837  212.125  278.01282
## as.factor(strength)15   75.7371837  141.625  207.51282
## as.factor(strength)25   71.7371837  137.625  203.51282
## as.factor(strength)50  115.9871837  181.875  247.76282
## attr(,"label")
## [1] "Fixed effects:"
##
##  Random Effects:
##   Level: tree
##                 lower     est.     upper
## sd((Intercept)) 26.02413 61.4961 145.3178
##
##  Within-group standard error:
##    lower      est.     upper
## 31.99215 44.35169 61.48610
```

3