

Parameter Estimation

Bret Larget

Departments of Botany and of Statistics
University of Wisconsin—Madison

April 12, 2007

Kiwi Shade Example

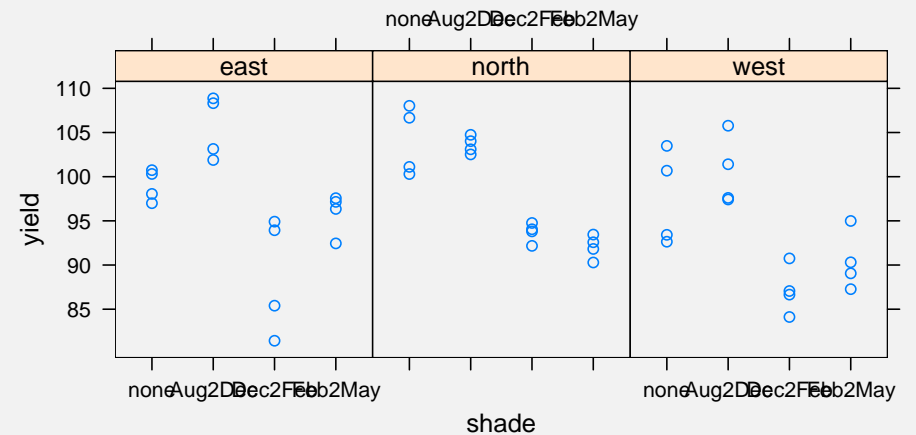
- Continue the kiwi shade example.
- Estimate the shade effects from Model 1.

Data

```
> library(DAAG)
> data(kiwishade)
> attach(kiwishade)
> str(kiwishade)

'data.frame': 48 obs. of 4 variables:
 $ yield: num 101.1 108.0 106.7 100.3 92.6 ...
 $ block: Factor w/ 3 levels "east","north",...: 2 2 2 2 3 3 3 3 1 1 ...
 $ shade: Factor w/ 4 levels "none","Aug2Dec",...: 1 1 1 1 1 1 1 1 1 1 ...
 $ plot : Factor w/ 12 levels "east.Aug2Dec",...: 8 8 8 8 12 12 12 12 4 4
```

Plots



Model 1

```
> library(lme4)
> kiwi1.lmer = lmer(yield ~ shade + (1 | block) + (1 | block:shade))
```

- Treat `block` as a random effect, as in the text.
- Treat `shade` as a fixed effect.
- We are interested in all of the comparisons between shade levels.

Model 1 Summary

```
> summary(kiwi1.lmer)

Linear mixed-effects model fit by REML
Formula: yield ~ shade + (1 | block) + (1 | block:shade)
   AIC   BIC logLik MLdeviance REMLdeviance
264.0 275.2 -126.0   262.8       252.0

Random effects:
Groups   Name              Variance Std.Dev.
block:shade (Intercept)  2.1857  1.4784
block      (Intercept)  4.0361  2.0090
Residual                    12.1878  3.4911
number of obs: 48, groups: block:shade, 12; block, 3

Fixed effects:
              Estimate Std. Error t value
(Intercept)   100.203    1.758    57.01
shadeAug2Dec     3.031     1.868     1.62
shadeDec2Feb   -10.282     1.868    -5.50
shadeFeb2May    -7.428     1.868    -3.98

Correlation of Fixed Effects:
      (Intr) shdA2D shdD2F
shadeAug2Dc -0.531
shadeDec2Fb -0.531  0.500
shadeFeb2My -0.531  0.500  0.500
```

- For parameter estimation, REML is preferable to ML.
- With the standard parameterization, we have parameters for the mean yield with shade level none and differences between none and other levels.
- Lets find 95% intervals for all possible pairwise differences.
- This will be somewhat similar to Fisher LSD tests.

Fixed Effects

```
> fe = fixef(kiwi1.lmer)
> fe
(Intercept) shadeAug2Dec shadeDec2Feb shadeFeb2May
100.202500    3.030833   -10.281667   -7.428333

> mu = c(fe[1], fe[1] + fe[2:4])
> names(mu)[1] = "none"
> mu
      none shadeAug2Dec shadeDec2Feb shadeFeb2May
100.20250    103.23333    89.92083    92.77417
```

- R code shows how to find the means for each treatment level.
- Are the pairwise differences significant?

A Parametric Approach

- In this balanced experiment, all of the pairwise differences in shade effects will have the same SE.
- Here it is calculated as 1.868.
- The most appropriate degrees of freedom for t distribution inference is the degrees of freedom associated with plot level.
- We can use the nested factor diagram to see that the best choice for degrees of freedom is 6.
- There are 12 levels in plot and 6 df in variables in which it is nested (1 for the intercept, 3 for shade, 2 for block).

Fixed Model

```
> kiwi1.fixed = lm(yield ~ block * shade)
> anova(kiwi1.fixed)
```

Analysis of Variance Table

```
Response: yield
      Df Sum Sq Mean Sq F value    Pr(>F)
block   2  172.35   86.17   7.0734 0.002565 **
shade   3 1394.51  464.84 38.1553 2.836e-11 ***
block:shade  6  125.57   20.93  1.7178 0.145020
Residuals 36  438.58   12.18
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Fit a fixed effects model just to verify the df.
- See the 6 df for the interaction term.

R Functions for Pairwise Comparisons

```
> source("pairwise.R")
> pairwiseDiff

function (x)
{
  n = length(x)
  nm = names(x)
  np = n * (n - 1)/2
  d = rep(0, np)
  dnames = rep("A", np)
  k = 0
  for (i in 1:(n - 1)) {
    for (j in (i + 1):n) {
      k = k + 1
      d[k] = mu[i] - mu[j]
      dnames[k] = paste(nm[i], "-", nm[j], sep = "")
    }
  }
  names(d) = dnames
  d
}
```

R Functions for Pairwise Comparisons

```
> pairwiseCI

function (d, se, df, conf.level = 0.95)
{
  low = (1 - conf.level)/2
  high = 1 - low
  tmult = qt(high, df)
  a = d - tmult * se
  b = d + tmult * se
  tstat = d/se
  p = 2 * pt(-abs(tstat), df)
  out = data.frame(Diff = d, Low = a, High = b, t = tstat,
    Pvalue = p)
  cat(paste(100 * conf.level, "%", " Confidence Intervals for Pairwise Differences\n",
    sep = "\n"))
  out
}
```

Confidence Intervals

```
> d = pairwiseDiff(mu)
> ci = round(pairwiseCI(d, 1.868, 6), 4)

95% Confidence Intervals for Pairwise Differences

> ci
```

	Diff	Low	High	t	Pvalue
none-shadeAug2Dec	-3.0308	-7.6017	1.5400	-1.6225	0.1558
none-shadeDec2Feb	10.2817	5.7108	14.8525	5.5041	0.0015
none-shadeFeb2May	7.4283	2.8575	11.9992	3.9766	0.0073
shadeAug2Dec-shadeDec2Feb	13.3125	8.7417	17.8833	7.1266	0.0004
shadeAug2Dec-shadeFeb2May	10.4592	5.8883	15.0300	5.5991	0.0014
shadeDec2Feb-shadeFeb2May	-2.8533	-7.4242	1.7175	-1.5275	0.1775

MCMC Approach

```

> set.seed(324)
> kiwi1.mcmc = mcmcsmpl(kiwi1.lmer, 10000)
> kiwi1.mcmc[1, ]

      (Intercept)  shadeAug2Dec  shadeDec2Feb  shadeFeb2May  log(sigma^2)
100.5304568      3.5369449      -9.9892995      -8.5594243      2.3960431
log(blck.(In)) log(blck.(In))
-0.4282659      3.5135838

> out = matrix(0, 6, 2)
> out[1, ] = quantile(-kiwi1.mcmc[, 2], c(0.025, 0.975))
> out[2, ] = quantile(-kiwi1.mcmc[, 3], c(0.025, 0.975))
> out[3, ] = quantile(-kiwi1.mcmc[, 4], c(0.025, 0.975))
> out[4, ] = quantile(kiwi1.mcmc[, 2] - kiwi1.mcmc[, 3], c(0.025,
+ 0.975))
> out[5, ] = quantile(kiwi1.mcmc[, 2] - kiwi1.mcmc[, 4], c(0.025,
+ 0.975))
> out[6, ] = quantile(kiwi1.mcmc[, 3] - kiwi1.mcmc[, 4], c(0.025,
+ 0.975))
> round(cbind(out, ci[, 1:3]), 4)

      1      2  Diff  Low  High
none-shadeAug2Dec -6.3564 0.2383 -3.0308 -7.6017 1.5400
none-shadeDec2Feb  6.9811 13.5775 10.2817  5.7108 14.8525
none-shadeFeb2May  4.2025 10.6412  7.4283  2.8575 11.9992
shadeAug2Dec-shadeDec2Feb 10.0411 16.5880 13.3125  8.7417 17.8833
shadeAug2Dec-shadeFeb2May  7.2195 13.7467 10.4592  5.8883 15.0300
shadeDec2Feb-shadeFeb2May -6.1151  0.3674 -2.8533 -7.4242  1.7175

```

Histogram (1-4)

