Analysis of Split-Plot / Repeated Measures Designs

Dean Billheimer
dean.billheimer@arizona.edu

October 18, 2011

Summary

Split-plot designs in agriculture, also called repeated measures designs in other disciplines, are characterized by having treatments randomly assigned in *differently sized experimental units*. This note builds up the split-plot design and analysis starting with the paired t-test, and showing the relation with randomized complete block designs (RCB). We also show R code useful in performing the analysis

Paired t-test

Suppose we have 20 subjects, each treated with "A" and "B", independently, to left and right sides of their body (e.g., to their ears). We assume random assignment of treatment to left/right side.

For treatment "A" we observe

```
 [1]  78.04663 101.31718  99.64849 113.11044 128.81602 111.99808  94.21503
 [8] 127.40002 120.01163  89.20466 102.24282 100.29838  80.94072 106.66220
[15]  82.03891  92.58303  97.70150 117.64467 143.19236 125.18019
```

and for "B" for the same subjects

```
 [1]  86.71073 114.02964 108.85856 122.88292 140.97127 114.28859 109.10115
 [8] 131.03462 126.57009  94.69270 111.53536  99.13773  94.25874 107.64587
[15]  95.73387 102.30761 104.77701 126.66449 148.77435 130.92348
```

The usual independent sample t-test and paired-t-test results in the following output.

```
> t.test(A <- Ymat[, 1], B <- Ymat[, 2])

        Welch Two Sample t-test

data:  A <- Ymat[, 1] and B <- Ymat[, 2]
t = -1.4614, df = 37.883, p-value = 0.1521
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -18.921190   3.056607
sample estimates:
mean of x mean of y
 105.6126  113.5449

> t.test(A <- Ymat[, 1], B <- Ymat[, 2], paired = T)
```

```
        Paired t-test

data:  A <- Ymat[, 1] and B <- Ymat[, 2]
t = -8.1133, df = 19, p-value = 1.359e-07
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -9.978610 -5.885973
sample estimates:
mean of the differences
              -7.932292
```

Randomized Complete Block

Equivalently, we can analyze these data as a RCB, with subjects as "blocks".

```
> library(nlme)
> options(contrasts = c("contr.sum", "contr.poly"))
> anova(lme(Y ~ treat, random = ~1 | subject.id))

            numDF denDF  F-value p-value
(Intercept)     1    19 828.6219  <.0001
treat           1    19  65.8263  <.0001

> summary(lme(Y ~ treat, random = ~1 | subject.id))

Linear mixed-effects model fit by REML
 Data: NULL
       AIC      BIC    logLik
  286.9943 293.5446 -139.4972

Random effects:
 Formula: ~1 | subject.id
        (Intercept) Residual
StdDev:    16.88311 3.091712

Fixed effects: Y ~ treat
                Value Std.Error DF   t-value p-value
(Intercept) 109.57879  3.806697 19 28.785793       0
treat1       -3.96615  0.488843 19 -8.113338       0
 Correlation:
       (Intr)
treat1 0

Standardized Within-Group Residuals:
       Min         Q1        Med         Q3        Max
-1.5231308 -0.3882613 -0.0677661  0.4829766  1.4179390

Number of Observations: 40
Number of Groups: 20
```

Thus, we see that squaring the t statistic from the paired test yields the F statistic from the RCB
What if we treat blocks as fixed?

```
> anova(lm(Y ~ subject.id + treat))

Analysis of Variance Table

Response: Y
           Df  Sum Sq Mean Sq F value    Pr(>F)
subject.id 19 11013.1  579.64  60.640 4.141e-13 ***
treat       1   629.2  629.21  65.826 1.359e-07 ***
Residuals  19   181.6    9.56
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> summary(lm(Y ~ subject.id + treat))

Call:
lm(formula = Y ~ subject.id + treat)

Residuals:
   Min     1Q Median     3Q    Max
-4.546 -1.187  0.000  1.187  4.546

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  109.5788     0.4888 224.160  < 2e-16 ***
subject.id1  -27.2001     2.1308 -12.765 9.06e-11 ***
subject.id2   -1.9054     2.1308  -0.894 0.382400
subject.id3   -5.3253     2.1308  -2.499 0.021778 *
subject.id4    8.4179     2.1308   3.951 0.000858 ***
subject.id5   25.3148     2.1308  11.880 3.07e-10 ***
subject.id6    3.5645     2.1308   1.673 0.110742
subject.id7   -7.9207     2.1308  -3.717 0.001461 **
subject.id8   19.6385     2.1308   9.216 1.93e-08 ***
subject.id9   13.7121     2.1308   6.435 3.61e-06 ***
subject.id10 -17.6301     2.1308  -8.274 1.01e-07 ***
subject.id11  -2.6897     2.1308  -1.262 0.222120
subject.id12  -9.8607     2.1308  -4.628 0.000184 ***
subject.id13 -21.9791     2.1308 -10.315 3.19e-09 ***
subject.id14  -2.4248     2.1308  -1.138 0.269289
subject.id15 -20.6924     2.1308  -9.711 8.42e-09 ***
subject.id16 -12.1335     2.1308  -5.694 1.73e-05 ***
subject.id17  -8.3395     2.1308  -3.914 0.000933 ***
subject.id18  12.5758     2.1308   5.902 1.11e-05 ***
subject.id19  36.4046     2.1308  17.085 5.47e-13 ***
treat1        -3.9661     0.4888  -8.113 1.36e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.092 on 19 degrees of freedom
Multiple R-squared: 0.9846,        Adjusted R-squared: 0.9685
F-statistic:  60.9 on 20 and 19 DF,  p-value: 3.506e-13
```

Paired t-test with Groups

Now, suppose that the subjects from the analysis above may be segregated into two groups, say "M" and "F". Ideally, we would like to randomly assign "M" and "F" treatments to subjects. However, this is dependent on the experimental context.

How should we evaluate group (M/F) differences, and group-by-treatment (A/B) interaction?

Split-Plot Design

This is an example of a split plot design. We have an RCB, with subjects as "blocks"; treatment (A/B) as a "within-subject" factor (also called the split-plot factor), and group as a "between-subject" factor (whole plot factor).

A complicating issue is that the experimental units (EUs) differ in size for between and within group factors. The EU for group is subject, while the EU for treatment is ear-within-subject. The different sized EUs suggest different variances for hypothesis testing.

Degrees of Freedom Partitioning

Source	df	SS	MS	F	p
mean	1				
group	1				
subject	18				
treat	1				
group * treat	1				
residual	18				

To specify the model, we have group and treat as fixed effects, and subject.id as a random effect. Further, we specify that treatment experimental units are nested within subject.id. (That is, treat is the within-subject factor.)

```
> summary(lme(Y ~ group * treat, random = ~1 | subject.id/treat))

Linear mixed-effects model fit by REML
 Data: NULL
       AIC      BIC    logLik
  284.3037 295.3883 -135.1518

Random effects:
 Formula: ~1 | subject.id
        (Intercept)
StdDev:    16.22792

 Formula: ~1 | treat %in% subject.id
        (Intercept) Residual
StdDev:    2.379018 1.925264

Fixed effects: Y ~ group * treat
                  Value Std.Error DF   t-value p-value
(Intercept)   109.57879  3.660796 18 29.933049  0.0000
group1         -5.83971  3.660796 18 -1.595203  0.1281
treat1         -3.96615  0.483900 18 -8.196209  0.0000
group1:treat1  -0.57053  0.483900 18 -1.179031  0.2537
```

```
 Correlation:
              (Intr) group1 treat1
group1        0
treat1        0      0
group1:treat1 0      0      0

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-0.87364564 -0.28043288 -0.01170803  0.34001323  0.76086841

Number of Observations: 40
Number of Groups:
           subject.id treat %in% subject.id
                   20                    40
```

```
> anova(lme(Y ~ group * treat, random = ~1 | subject.id/treat))
```

```
            numDF denDF  F-value p-value
(Intercept)     1    18 895.9874  <.0001
group           1    18   2.5447  0.1281
treat           1    18  67.1778  <.0001
group:treat     1    18   1.3901  0.2537
```

The group effect F test contains subject-to-subject variability in the denominator of the F-statistic. Conversely, the treatment effect and group-by-treatment interaction use the ear-within-subject variance in the denominator. Because we have 2 groups (M/F) and 2 treatments (A/B), the denominator degrees of freedom (df) in the anova table are ambiguous about the test construction.

The example below used 3 within-subject treatments. This increases the ear-within-subject df, but leaves the subject df unchanged.

Three Within-Subject Treatments

Expanding to 3 within-subject treatments (A/B/C) demonstrates that we are geting the appropriate testing terms for between and within factors.

```
> anova(lme(Y.3 ~ group.3 * treat.3, random = ~1 | subject.3/treat.3))
```

```
                numDF denDF  F-value p-value
(Intercept)         1    36 917.2470  <.0001
group.3             1    18   3.4478  0.0798
treat.3             2    36  24.2214  <.0001
group.3:treat.3     2    36   3.1722  0.0538
```

```
> summary(lme(Y.3 ~ group.3 * treat.3, random = ~1 | subject.3/treat.3))
```

```
Linear mixed-effects model fit by REML
 Data: NULL
       AIC      BIC    logLik
  410.5897 428.4906 -196.2948

Random effects:
 Formula: ~1 | subject.3
        (Intercept)
```

```
StdDev:    15.89512

 Formula: ~1 | treat.3 %in% subject.3
        (Intercept) Residual
StdDev:    3.153954 2.227706

Fixed effects: Y.3 ~ group.3 * treat.3
                     Value Std.Error DF   t-value p-value
(Intercept)       108.69810  3.589044 36 30.286086  0.0000
group.31           -6.66421  3.589044 18 -1.856822  0.0798
treat.31           -3.08545  0.704985 36 -4.376619  0.0001
treat.32            4.84684  0.704985 36  6.875103  0.0000
group.31:treat.31   0.25397  0.704985 36  0.360244  0.7208
group.31:treat.32   1.39503  0.704985 36  1.978814  0.0555
 Correlation:
                  (Intr) grp.31 trt.31 trt.32 g.31:.31
group.31           0.0
treat.31           0.0    0.0
treat.32           0.0    0.0   -0.5
group.31:treat.31  0.0    0.0    0.0    0.0
group.31:treat.32  0.0    0.0    0.0    0.0   -0.5

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-1.06993442 -0.25904233 -0.03955354  0.28738847  1.07343910

Number of Observations: 60
Number of Groups:
             subject.3 treat.3 %in% subject.3
                    20                     60
```

Testing Linear Contrasts

Suppose we want to estimate the mean difference "M - F", and test it is equal to zero?
To get the test statistic, we could look at the overall anova test from above.
Or?

```
> library(gmodels)
> fit.3 <- lme(Y.3 ~ group.3 + treat.3 + group.3:treat.3, random = ~1 |
+     subject.3/treat.3)
> summary(fit.3)

Linear mixed-effects model fit by REML
 Data: NULL
       AIC      BIC    logLik
  410.5897 428.4906 -196.2948

Random effects:
 Formula: ~1 | subject.3
        (Intercept)
StdDev:    15.89512

 Formula: ~1 | treat.3 %in% subject.3
        (Intercept) Residual
StdDev:    3.153954 2.227706

Fixed effects: Y.3 ~ group.3 + treat.3 + group.3:treat.3
                     Value Std.Error DF   t-value p-value
(Intercept)       108.69810  3.589044 36 30.286086  0.0000
group.31           -6.66421  3.589044 18 -1.856822  0.0798
treat.31           -3.08545  0.704985 36 -4.376619  0.0001
treat.32            4.84684  0.704985 36  6.875103  0.0000
group.31:treat.31   0.25397  0.704985 36  0.360244  0.7208
group.31:treat.32   1.39503  0.704985 36  1.978814  0.0555
 Correlation:
                  (Intr) grp.31 trt.31 trt.32 g.31:.31
group.31           0.0
treat.31           0.0    0.0
treat.32           0.0    0.0   -0.5
group.31:treat.31  0.0    0.0    0.0    0.0
group.31:treat.32  0.0    0.0    0.0    0.0   -0.5

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-1.06993442 -0.25904233 -0.03955354  0.28738847  1.07343910

Number of Observations: 60
Number of Groups:
             subject.3 treat.3 %in% subject.3
                    20                    60

> fit.contrast(fit.3, "group.3", c(-1, 1))

                   Estimate Std. Error  t-value  Pr(>|t|)
group.3 c=( -1 1 ) 13.32843   7.178088 1.856822 0.0797791
```

```
> fit.contrast(fit.3, "group.3", c(-1, 1), showall = T)

                              Estimate Std. Error    t-value     Pr(>|t|)
(Intercept)                 108.6980971  3.5890440 30.2860862 0.000000e+00
group.3 c=( -1 1 )           13.3284297  7.1780881  1.8568217 7.977910e-02
treat.31                     -3.0854492  0.7049847 -4.3766189 9.907540e-05
treat.32                      4.8468423  0.7049847  6.8751032 4.789961e-08
group.3 c=( -1 1 ):treat.31  -0.5079333  1.4099693 -0.3602442 7.207696e-01
group.3 c=( -1 1 ):treat.32  -2.7900667  1.4099693 -1.9788138 5.552618e-02

> fit.contrast(fit.3, "treat.3", c(-1, 0, 1))

                    Estimate Std. Error  t-value  Pr(>|t|)
treat.3 c=( -1 0 1 ) 1.324056   1.221069 1.084342 0.2854221

> fit.contrast(fit.3, "treat.3", c(-1, 0, 1), showall = T)

                               Estimate Std. Error   t-value     Pr(>|t|)
(Intercept)                  108.698097  3.5890440 30.286086 0.000000e+00
group.31                      -6.664215  3.5890440 -1.856822 7.977910e-02
treat.3 c=( -1 0 1 )           1.324056  1.2210693  1.084342 2.854221e-01
treat.3 C2                    -5.936145  0.8634263 -6.875103 4.789961e-08
group.31:treat.3 c=( -1 0 1 ) -1.902967  1.2210693 -1.558443 1.278771e-01
group.31:treat.3 C2           -1.708560  0.8634263 -1.978814 5.552618e-02

> fit.3$contr

$group.3
  [,1]
M    1
F   -1

$treat.3
  [,1] [,2]
A    1    0
B    0    1
C   -1   -1
```

Model Parameterization

The matrix below shows the model parameterization – the relationship between "cell means" and treatment effects in the model.

Mean	Int	group.3	treat.31	treat.32	grp3:trt31	grp3:trt32
MA	1	1	1	0	1	0
MB	1	1	0	1	0	1
MC	1	1	-1	-1	-1	-1
FA	1	-1	1	0	-1	0
FB	1	-1	0	1	0	-1
FC	1	-1	-1	-1	1	1

We encode this in the model matrix below.

```
> model.mat <- rbind(MA = c(1, 1, 1, 0, 1, 0), MB = c(1, 1, 0,
+     1, 0, 1), MC = c(1, 1, -1, -1, -1, -1), FA = c(1, -1, 1,
+     0, -1, 0), FB = c(1, -1, 0, 1, 0, -1), FC = c(1, -1, -1,
+     -1, 1, 1))
```

The `estimable` command estimates the expected value and standard error for each linear combibation. Here we compute the cell means.

```
> estimable(fit.3, model.mat)

    Estimate Std. Error  t value DF     Pr(>|t|)
MA  99.20240   5.172667 19.17819 18 1.985079e-13
MB 108.27576   5.172667 20.93229 18 4.396483e-14
MC  98.62349   5.172667 19.06628 18 2.193801e-13
FA 112.02290   5.172667 21.65670 18 2.442491e-14
FB 118.81412   5.172667 22.96961 18 8.881784e-15
FC 115.24992   5.172667 22.28056 18 1.465494e-14
```

Next, we can estiamte the "M-F" effect only among the "A" level of treatment.

```
> estimable(fit.3, rbind(`M-F|A` = c(model.mat[1, ] - model.mat[4,
+     ])))

      Estimate Std. Error  t value DF   Pr(>|t|)
M-F|A -12.8205   7.315255 -1.75257 36 0.08818987
```

Similarly, we can test "M-F" across all levels of the second factor.

```
> estimable(fit.3, rbind(`M-F Effect` = c(0, -2, 0, 0, 0, 0)))

           Estimate Std. Error  t value DF   Pr(>|t|)
M-F Effect 13.32843   7.178088 1.856822 36 0.07153401
```

Note that this result is identical to what we saw before using `fit.contrast`.

```
> fit.contrast(fit.3, "group.3", c(-1, 1))

                  Estimate Std. Error  t-value  Pr(>|t|)
group.3 c=( -1 1 ) 13.32843   7.178088 1.856822 0.0797791
```

Finally, we also check the ■A-B■ effect averaged across M and F groups.

```
> estimable(fit.3, rbind(`A-B Effect` = c(0, 0, 1, -1, 0, 0)))
```

```
            Estimate Std. Error   t value DF     Pr(>|t|)
A-B Effect -7.932292   1.221069 -6.496185 36 1.517473e-07
```

Same, using fit.contrast.

```
> fit.contrast(fit.3, "treat.3", c(1, -1, 0))

                      Estimate Std. Error   t-value     Pr(>|t|)
treat.3 c=( 1 -1 0 ) -7.932292   1.221069 -6.496185 1.517473e-07
```