

Analysis of Split-Plot / Repeated Measures Designs

Dean Billheimer
<dean.billheimer@arizona.edu>

Julia Fisher
<julia@statlab.bio5.org>

October 11, 2017

Introduction

Split-plot designs in agriculture, also called repeated measures designs in other disciplines, are characterized by having treatments randomly assigned to *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.

First, let's generate some data:

```
set.seed(20100310)
subject <- c(rnorm(10, 100, 15), rnorm(10, 120, 15))
group <- factor(c(rep("M", 10), rep("F", 10)), levels=c("M", "F"))
permute <- order(sample(1:20))
subject <- subject[permute]
group <- group[permute]
treat <- factor(rep(c("A", "B", "C"), each = 20))
Y = c(rnorm(20, subject - 5, 5),
      rnorm(20, subject, 5),
      rnorm(20, subject + 5, 5))
subject.id <- factor(rep(1:20, 3))
df <- data.frame(subject.id = subject.id,
                 group = rep(group, 3),
                 treat = treat,
                 Y = Y)
```

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. (We’re going to ignore treatment “C” for the moment.)

For treatment “A”, we observe the following:

```
df[which(df$treat == "A"), ]
```

	subject.id	group	treat	Y
1	1	M	A	78.04663
2	2	F	A	101.31718
3	3	M	A	99.64849
4	4	F	A	113.11044
5	5	F	A	128.81602
6	6	M	A	111.99808
7	7	M	A	94.21503
8	8	F	A	127.40002
9	9	M	A	120.01163
10	10	F	A	89.20466
11	11	M	A	102.24282
12	12	F	A	100.29838
13	13	M	A	80.94072
14	14	F	A	106.66220
15	15	M	A	82.03891
16	16	F	A	92.58303
17	17	M	A	97.70150
18	18	F	A	117.64467
19	19	F	A	143.19236
20	20	M	A	125.18019

For treatment “B” (for the same subjects!), we observe the following:

```
df[which(df$treat == "B"), ]
```

	subject.id	group	treat	Y
21	1	M	B	86.71073
22	2	F	B	114.02964
23	3	M	B	108.85856
24	4	F	B	122.88292
25	5	F	B	140.97127
26	6	M	B	114.28859
27	7	M	B	109.10115
28	8	F	B	131.03462
29	9	M	B	126.57009
30	10	F	B	94.69270
31	11	M	B	111.53536
32	12	F	B	99.13773
33	13	M	B	94.25874
34	14	F	B	107.64587
35	15	M	B	95.73387
36	16	F	B	102.30761
37	17	M	B	104.77701
38	18	F	B	126.66449
39	19	F	B	148.77435
40	20	M	B	130.92348

Let’s compare the distributions of the “A” and “B” observations:

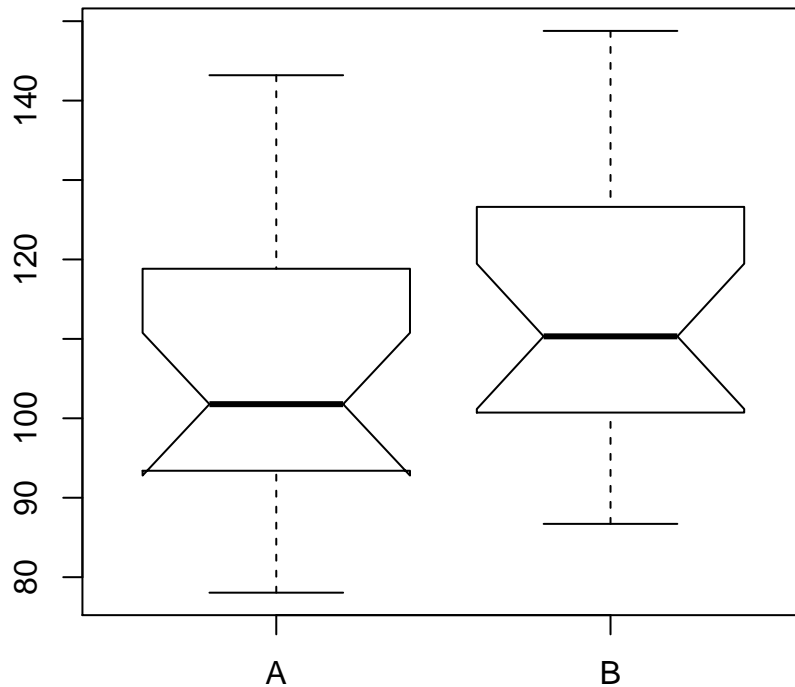


Figure 1: Boxplot of A and B groups.

```

boxplot(df[which(df$treat == "A"), "Y"], df[which(df$treat == "B"), "Y"],
        notch = T,
        names = c("A", "B"))

```

Let's look at the relationship between each "A" and "B" observation within an individual:

```

plot(c(rep(1, 20), rep(2, 20)),
     df[which(df$treat %in% c('A', 'B')), 'Y'],
     xlim = c(0.5, 2.5),
     xaxt = 'n',
     ylab = "Response",
     xlab = "Treatment")
segments(rep(1, 20), df[which(df$treat == 'A'), 'Y'],
         rep(2, 20), df[which(df$treat == 'B'), 'Y'])
axis(1, at = c(1,2), labels = c("A", "B"))

```

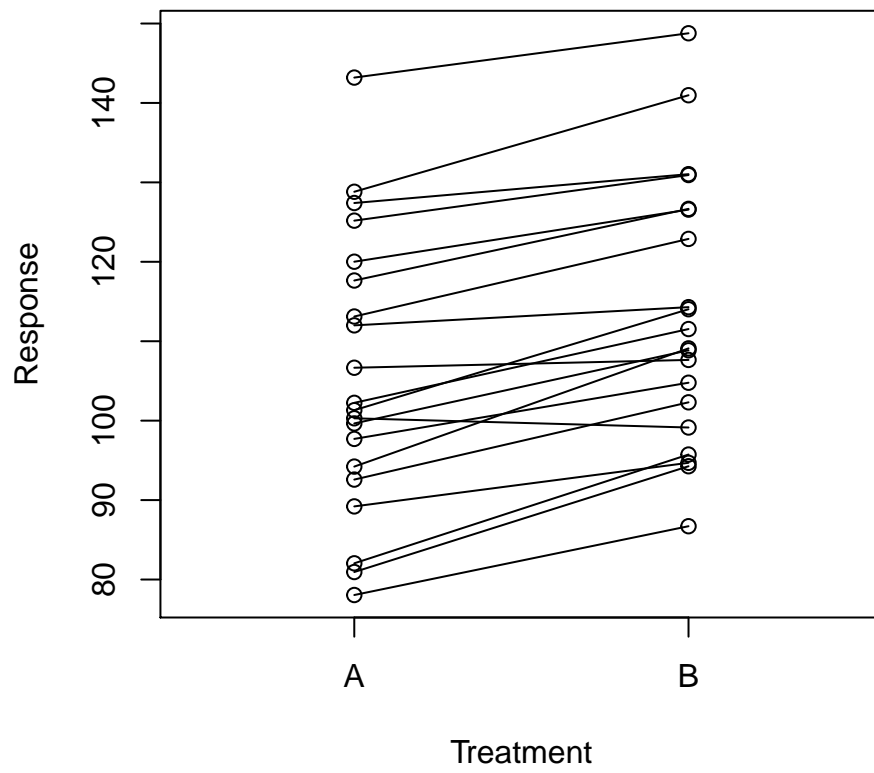


Figure 2: Paired plot of A and B groups.

Paired t -Test

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

Independent Sample t -Test:

```
t.test(df[which(df$treat == 'A'), 'Y'],
       df[which(df$treat == 'B'), 'Y'])

Welch Two Sample t-test

data: df[which(df$treat == "A"), "Y"] and df[which(df$treat == "B"), "Y"]
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
```

Dependent Sample t -Test:

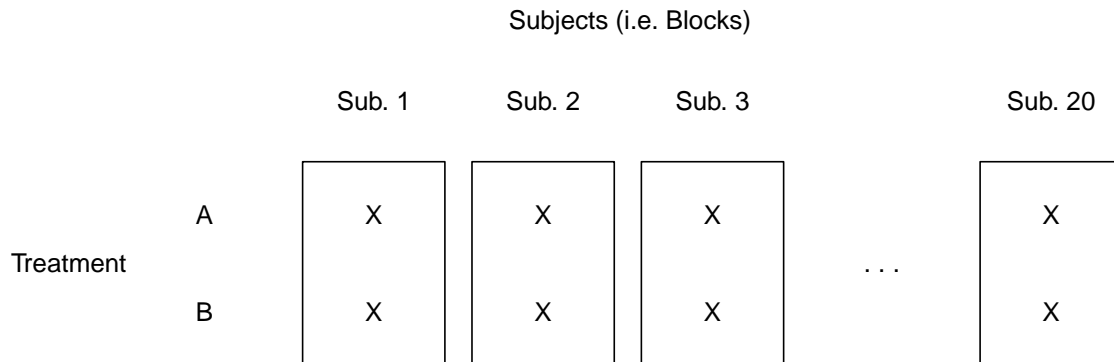
```
t.test(df[which(df$treat == 'A'), 'Y'],
       df[which(df$treat == 'B'), 'Y'],
       paired = TRUE)

Paired t-test

data: df[which(df$treat == "A"), "Y"] and df[which(df$treat == "B"), "Y"]
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 randomized complete block design (RCB), with subjects as “blocks”.



There are two ways to approach this. The first uses a mixed-effects model, with subjects (blocks) as random:

```
library(nlme)
options(contrasts=c("contr.sum", "contr.poly")) ## not required
mdl.mixed <- lme(Y ~ treat,
  random = ~ 1 | subject.id,
  data = df[which(df$treat %in% c("A", "B")), ])
anova(mdl.mixed)

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

summary(mdl.mixed)
Linear mixed-effects model fit by REML
Data: df[which(df$treat %in% c("A", "B")), ]
      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
```

Are we getting the same analysis as the paired t -test? Yes! The t statistic in the summary of the mixed-effects model is the same as the t -statistic in the paired t -test. Also, squaring that t statistic yields the F statistic from the RCB.

We can also use a fixed-effects-only model (i.e. a normal multiple regression model):

```
mdl.fixed <- lm(Y ~ subject.id + treat, data = df[which(df$treat %in% c("A", "B")), ])
anova(mdl.fixed)
```

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(mdl.fixed)
```

Call:

```
lm(formula = Y ~ subject.id + treat, data = df[which(df$treat %in%
  c("A", "B")), ])
```

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

```

What changed?

- Did the estimated effect (and t statistic) for treatment change? No!
- We do get estimates for each subject. Actually, we also get these in the mixed model. They're just not so obvious. They're not identical to the estimates from the fixed-effects model, but they're really close.

```

# Subject Effects in Mixed-Effects Model:
df2 <- data.frame(random_effects = ranef mdl.mixed)[, '(Intercept)'],
                  fixed_effects = c(coef(mdl.fixed)[2:20], -sum(coef(mdl.fixed)[2:20])))
rownames(df2)[20] <- "subject.id20"
print(df2)

```

	random_effects	fixed_effects
subject.id1	-26.751561	-27.200113
subject.id2	-1.873964	-1.905385
subject.id3	-5.237451	-5.325269
subject.id4	8.279072	8.417890
subject.id5	24.897388	25.314850
subject.id6	3.505757	3.564539
subject.id7	-7.790083	-7.920701
subject.id8	19.314672	19.638527
subject.id9	13.485941	13.712064
subject.id10	-17.339382	-17.630117
subject.id11	-2.645345	-2.689701
subject.id12	-9.698129	-9.860740
subject.id13	-21.616615	-21.979067
subject.id14	-2.384768	-2.424754
subject.id15	-20.351170	-20.692404
subject.id16	-11.933382	-12.133473
subject.id17	-8.202012	-8.339538
subject.id18	12.368405	12.575790
subject.id19	35.804220	36.404560
subject.id20	18.168406	18.473041

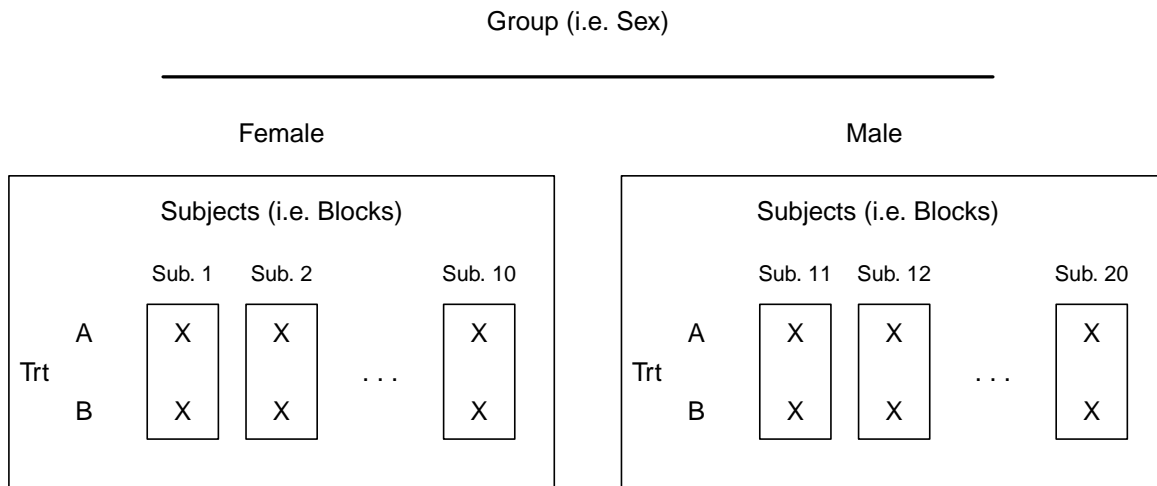
Paired t -test with Groups

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 the 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). See the diagram below:



Compare this diagram to the RCB diagram before. Notice that within each whole-plot treatment level, we have an RCB!

But wait! What are our experimental units? Subjects are the experimental units for the whole plot factor (sex). Ears-within-subjects are the experimental units for the split-plot treatment (A/B).

A complicating issue is that the experimental units (EUs) differ in size for between and within group factors. The different sized EUs suggest different variances for hypothesis testing. More on this in a second.

Degrees of Freedom Partitioning

Let’s think about how to partition the degrees of freedom in a split plot design.

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

Note: When thinking about the degrees of freedom in a split-plot design, note two things:

- The degrees of freedom from the whole-plot section of the table (including the whole plot error) must add to the number of whole-plot experimental units.
- The total degrees of freedom in the table must add up to the total number of data points (i.e. the total number of split-plot units).

Now it's your turn! Suppose we're running an experiment on men and women of three different language backgrounds: English, Spanish, and Japanese. We present them with pairs of sounds. In each pair, the first sound may either repeat or be different. There are seven different initial sounds, and each of those seven sounds has a unique second sound that will get played in the "different" condition. The subjects' task is to press a button as quickly as possible indicating whether or not the two sounds are the same. We measure their reaction time (i.e. how quickly it takes them to respond). Each subject sees each language condition once. We have 10 men and 10 women from each language background participating in the task. Suppose we want to model reaction time with fixed factors sex, language background, initial sound, same/different, and all possible interactions of those factors. What are the degrees of freedom associated with each factor? What are the error degrees of freedom at the whole and split plot levels?

Now back to our group by treatment problem. To specify the model, we have group and treatment as fixed effects, and subject.id as a random effect.

```
mdl.mixed2 <- lme(Y ~ group*treat,
                random = ~ 1 | subject.id,
                data = df[which(df$treat %in% c("A", "B")), ])
anova(mdl.mixed2)

              numDF denDF  F-value p-value
(Intercept)      1     18 895.9889 <.0001
group             1     18  2.5447 0.1281
treat            1     18 67.1777 <.0001
group:treat      1     18  1.3901 0.2537

summary(mdl.mixed2)

Linear mixed-effects model fit by REML
Data: df[which(df$treat %in% c("A", "B")), ]
      AIC      BIC    logLik
282.3037 291.8048 -135.1518

Random effects:
Formula: ~1 | subject.id
      (Intercept) Residual
StdDev:    16.22791  3.060455
```

```

Fixed effects: Y ~ group * treat
              Value Std.Error DF   t-value p-value
(Intercept)  109.57879  3.660793 18 29.933073  0.0000
group1       -5.83971  3.660793 18 -1.595205  0.1281
treat1      -3.96615  0.483900 18 -8.196203  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
-1.38877002 -0.44578316 -0.01861136  0.54049416  1.20949582

Number of Observations: 40
Number of Groups: 20

```

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 uses 3 within-subject treatments. This increases the ear-within-subject degrees of freedom, but leaves the subject degrees of freedom unchanged.

Three Within-Subject Treatments

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

```

mdl.mixed3 <- lme(Y ~ group*treat,
                 random = ~ 1 | subject.id,
                 data = df)
anova(mdl.mixed3)

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

summary(mdl.mixed3)

Linear mixed-effects model fit by REML
Data: df
      AIC      BIC    logLik
408.5897 424.5016 -196.2948

Random effects:

```

```

Formula: ~1 | subject.id
      (Intercept) Residual
StdDev:   15.89512  3.86136

Fixed effects: Y ~ group * treat
              Value Std.Error DF   t-value p-value
(Intercept)  108.69810  3.589044 36 30.286086  0.0000
group1       -6.66421  3.589044 18 -1.856822  0.0798
treat1      -3.08545  0.704985 36 -4.376619  0.0001
treat2       4.84684  0.704985 36  6.875103  0.0000
group1:treat1  0.25397  0.704985 36  0.360244  0.7208
group1:treat2  1.39503  0.704985 36  1.978814  0.0555
Correlation:
(Intr) group1 treat1 treat2 grp1:1
group1      0.0
treat1      0.0   0.0
treat2      0.0   0.0  -0.5
group1:treat1 0.0   0.0   0.0   0.0
group1:treat2 0.0   0.0   0.0   0.0  -0.5

Standardized Within-Group Residuals:
      Min      Q1      Med      Q3      Max
-1.85455461 -0.44900709 -0.06855952  0.49814044  1.86062940

Number of Observations: 60
Number of Groups: 20

```

Testing Linear Contrasts in Split Plot Designs

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?

Contrasts on Cell Means

We can estimate contrasts on the “cell means”.

```
library(lsmeans)

# Get the predicted marginal means for the male and female cells.
mdl.mixed3.lsmeans1 <- lsmeans(mdl.mixed3, ~ group)
print(mdl.mixed3.lsmeans1)

  group   lsmean      SE df  lower.CL upper.CL
  M     102.0339  5.075675  18   91.37029 112.6975
  F     115.3623  5.075675  18  104.69872 126.0259

Results are averaged over the levels of: treat
Confidence level used: 0.95

# Estimate the difference between male and female groups:
contrast(mdl.mixed3.lsmeans1, list("M - F" = c(1, -1)),
         by = NULL)

  contrast  estimate      SE df t.ratio p.value
  M - F     -13.32843  7.178088  18  -1.857  0.0798

Results are averaged over the levels of: treat

# Get the predicted marginal means for the group x treatment cells.
mdl.mixed3.lsmeans2 <- lsmeans(mdl.mixed3, ~ treat | group)
print(mdl.mixed3.lsmeans2)

group = M:
  treat   lsmean      SE df  lower.CL upper.CL
  A      99.20240  5.172667  18   88.33503 110.0698
  B     108.27576  5.172667  18   97.40839 119.1431
  C      98.62349  5.172667  18   87.75612 109.4909

group = F:
  treat   lsmean      SE df  lower.CL upper.CL
  A     112.02290  5.172667  18  101.15553 122.8903
  B     118.81412  5.172667  18  107.94675 129.6815
  C     115.24992  5.172667  18  104.38255 126.1173

Confidence level used: 0.95
```

```

# Estimate the difference between male and female groups by first averaging
# over treatment within each level of group:
contrast(mdl.mixed3.lsmmeans2, list("M - F" = c(1/3, 1/3, 1/3, -1/3, -1/3, -1/3)),
        by = NULL)

contrast estimate          SE df t.ratio p.value
M - F      -13.32843 7.178088 18  -1.857  0.0798

# Estimate the difference between male and female groups at just
# level A of treatmentL
contrast(mdl.mixed3.lsmmeans2, list("M - F | A" = c(1, 0, 0, -1, 0, 0)),
        by = NULL)

contrast estimate          SE df t.ratio p.value
M - F | A  -12.8205 7.315255 18  -1.753  0.0967

# Let's estimate the difference between treatment C and A.
contrast(mdl.mixed3.lsmmeans2, list("C - A" = c(-1/2, 0, 1/2, -1/2, 0, 1/2)),
        by = NULL)

contrast estimate          SE df t.ratio p.value
C - A       1.324056 1.221069 36   1.084  0.2854

```

Model Parameterization

Now, let's think about our model parameterization — the relationship between “cell means” and treatment effects in the model. That parameterization is given below:

Mean	Int	group1	treat1	treat2	grp1:trt1	grp1:trt2
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

As an example, this says that the estimated cell mean for female participants receiving treatment B is the intercept - group1 + treat2 - grp1:trt2.

These are sum contrasts. (Remember, we set that option just before our first model!) In sum contrasts, the intercept is the global mean. Then, for each factor, the estimated effects for the levels of that factor sum to zero. What gets displayed in R are the estimated effects for the first $l - 1$ treatment levels. To get the effect for the last level, sum the effects at the first $l - 1$ levels and multiply by -1.

If we wanted to check the lsmeans estimates of cell means, we can do that as follows:

```

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) )

estimated.cell.means <- model.mat %*% fixef mdl.mixed3)
print(estimated.cell.means)

      [,1]
MA  99.20240
MB 108.27576
MC  98.62349
FA 112.02290
FB 118.81412
FC 115.24992

# We can also use the estimable command from the gmodels package.
# This command estimates the expected value and standard error
# for each linear combination in the given matrix.
library(gmodels)
estimable(mdl.mixed3, 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

```

Contrasts on Model Parameters

Moreover, we can use this knowledge of the relationship between model parameters and cell means to conduct all the contrasts above on model parameters. It's a different way of viewing the same estimates. For example,

$$\begin{aligned}
M - F &= (MA + MB + MC)/3 - (FA + FB + FC)/3 \\
&= (\text{int} + \text{group1} + \text{treat1} + \text{grp1:trt1} + \\
&\quad \text{int} + \text{group1} + \text{treat2} + \text{grp1:trt2} + \\
&\quad \text{int} + \text{group1} - \text{treat1} - \text{treat2} - \text{grp1:trt1} - \text{grp1:trt2})/3 \\
&\quad - (\text{int} - \text{group1} + \text{treat1} - \text{grp1:trt1} + \\
&\quad \text{int} - \text{group1} + \text{treat2} - \text{grp1:trt2} + \\
&\quad \text{int} - \text{group1} - \text{treat1} - \text{treat2} + \text{grp1:trt1} + \text{grp1:trt2})/3 \\
&= 2 \times \text{group1}
\end{aligned}$$

```

# M - F effect across all treatments:
estimable(mdl.mixed3, rbind("M - F"= c(0, 2, 0, 0, 0, 0)))

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

```

```

# M - F effect only for treatment A:
estimable(mdl.mixed3, 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

# C - A:
estimable(mdl.mixed3,
          rbind("C - A"= c(1/2*(model.mat[3,] + model.mat[6,])
                        - 1/2*(model.mat[1,] + model.mat[4,])))

      Estimate Std. Error  t value DF   Pr(>|t|)
C - A  1.324056    1.221069  1.084342 36 0.2854221

```


Check lme4

The package lme4 is a newer package for mixed effects models by Doug Bates, one of the authors of the nlme package. It uses a slightly different model specification; some people think this is easier.

```
require('lme4')
mdl.lmer <- lmer(Y ~ group*treat + (1 |subject.id), data = df)
anova(mdl.lmer)
```

Analysis of Variance Table

	Df	Sum Sq	Mean Sq	F value
group	1	51.41	51.41	3.4478
treat	2	722.29	361.14	24.2214
group:treat	2	94.60	47.30	3.1722

```
summary(mdl.lmer)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: Y ~ group * treat + (1 | subject.id)
Data: df
```

REML criterion at convergence: 392.6

Scaled residuals:

Min	1Q	Median	3Q	Max
-1.85455	-0.44901	-0.06856	0.49814	1.86063

Random effects:

Groups	Name	Variance	Std.Dev.
subject.id	(Intercept)	252.65	15.895
	Residual	14.91	3.861

Number of obs: 60, groups: subject.id, 20

Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	108.698	3.589	30.286
group1	-6.664	3.589	-1.857
treat1	-3.085	0.705	-4.377
treat2	4.847	0.705	6.875
group1:treat1	0.254	0.705	0.360
group1:treat2	1.395	0.705	1.979

Correlation of Fixed Effects:

	(Intr)	group1	treat1	treat2	grp1:1
group1	0.000				
treat1	0.000	0.000			
treat2	0.000	0.000	-0.500		
group1:trt1	0.000	0.000	0.000	0.000	
group1:trt2	0.000	0.000	0.000	0.000	-0.500

Answer to Degrees of Freedom Problem

Source	df	SS	MS	F	p
mean	1				
sex	1				
language background	2				
sex * language background	2				
subject	54				
initial sound	6				
same/different	1				
initial sound * same/different	6				
initial sound * sex	6				
initial sound * language background	12				
same/different * sex	1				
same/different * language background	2				
initial sound * same/different * sex	6				
initial sound * same/different * language background	12				
initial sound * sex * language background	12				
same/different * sex * language background	2				
initial sound * same/different * sex * language background	12				
residual	702				

```

library(dplyr)
library(tidyr)

subject.id <- rep(1:60, 14)
sex <- rep(rep(c(rep("F", 10), rep("M", 10)), 3), 14)
lang.back <- rep(c(rep("Japanese", 20), rep("Spanish", 20), rep("English", 20)), 14)
initial.sound <- rep(c("n", "l", "b", "t", "r", "v", "w"), each = 120)
same.diff <- rep(c(rep("same", 60), rep("different", 60)), 7)
reaction.time <- rnorm(840, 0, 5)

lang.task.df <- data.frame(subject.id = factor(subject.id),
                          sex = sex,
                          lang.back = lang.back,
                          initial.sound = initial.sound,
                          same.diff = same.diff,
                          reaction.time = reaction.time) %>%
  arrange(subject.id, sex, lang.back, initial.sound, same.diff)

lang.task.mdl <- lme(reaction.time ~ sex*lang.back*initial.sound*same.diff,
                    random = ~ 1 | subject.id, lang.task.df)
# summary(lang.task.mdl)
anova(lang.task.mdl)


```

	numDF	denDF	F-value	p-value
(Intercept)	1	702	1.359029	0.2441
sex	1	54	0.095861	0.7580
lang.back	2	54	2.324575	0.1075
initial.sound	6	702	0.353977	0.9077
same.diff	1	702	0.502860	0.4785

sex:lang.back	2	54	2.302430	0.1098
sex:initial.sound	6	702	0.618446	0.7157
lang.back:initial.sound	12	702	0.719620	0.7330
sex:same.diff	1	702	1.393623	0.2382
lang.back:same.diff	2	702	0.501911	0.6056
initial.sound:same.diff	6	702	0.535470	0.7815
sex:lang.back:initial.sound	12	702	0.753397	0.6989
sex:lang.back:same.diff	2	702	4.548101	0.0109
sex:initial.sound:same.diff	6	702	0.911830	0.4857
lang.back:initial.sound:same.diff	12	702	1.886860	0.0328
sex:lang.back:initial.sound:same.diff	12	702	0.861578	0.5865