Lab 10 - Two Way ANOVA

April 17, 2017

In this lab we go over Two-Way Balanced and Unbalanced ANOVA in R.

The data come from a study on hospitalization of patients with kidney failure. These patients are commonly treated on dialysis machines that filter toxic substances from the blood. The appropriate "dose"; for effective treatment depends, among other things, on duration of dialysis treatment and weight gain between treatments as a result of fluid buildup. Duration was categorized into two groups, short (less than four hours) and long (four or more hours) coded as 1 and 2 respectively. Average weight gain was categorized into slight, moderate, and severe coded as 1, 2, and 3 respectively. The response variable of interest is the number of days hospitalized due to subsequent kidney disease. Prior analysis indicated that transforming the number of hospital days to LOG=LOG(DAYS+1) made the data more normal, and equalized the variance.

To study the effects of duration of dialysis treatment and weight gain on the number of days hospitalized due to disease, a random sample of 10 patients per group who had undergone treatment at a large dialysis facility was obtained. This full data set has equal numbers of observations in each cell, so the design is balanced. The unbalanced design is obtained by dropping the last two patients in the long duration and severe weight gain category, and is done purely for illustrative reasons. In a real study, experimental units might be missing (due to causes unrelated to treatment, such as moving out of the area), or might be dropped because they are outliers. In an observational study such as this one, unequal sample sizes may also occur because not enough experimental units are available in some treatment groups.

a) Fit an interaction Two-Way ANOVA model. Check to see that for both the balanced and unbalanced designs, the fitted values are just the cell means.

First load in the data, and get unbalanced data as described above.

```r
#balanced data
library(readr)
l10datBAL = read_csv("KidneyDatSp17.csv")
```

```
## Parsed with column specification:
## cols(
##   PatientID = col_integer(),
##   DaysHosp = col_integer(),
##   Duration = col_integer(),
##   WeightGain = col_integer()
## )
```

```r
l10datBAL$Duration = as.factor(l10datBAL$Duration)
l10datBAL$WeightGain = as.factor(l10datBAL$WeightGain)
l10datBAL$logDays = log(l10datBAL$DaysHosp + 1)

#unbalanced data
l10datUNB = l10datBAL[-c(59, 60),]
```

We can get cell means (i.e. the mean for each combination of duration and weight gain) as follows

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
```

```
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
group_by(l10datBAL, Duration, WeightGain) %>% summarise(count = n(), mean = mean(logDays))
```

```
## Source: local data frame [6 x 4]
## Groups: Duration [?]
##
##   Duration WeightGain count      mean
##     <fctr>     <fctr> <int>     <dbl>
## 1        1          1    10 1.0211560
## 2        1          2    10 1.8650191
## 3        1          3    10 2.5482706
## 4        2          1    10 0.9169518
## 5        2          2    10 1.3377192
## 6        2          3    10 1.9949474
```

```
group_by(l10datUNB, Duration, WeightGain) %>% summarise(count = n(), mean = mean(logDays))
```

```
## Source: local data frame [6 x 4]
## Groups: Duration [?]
##
##   Duration WeightGain count      mean
##     <fctr>     <fctr> <int>     <dbl>
## 1        1          1    10 1.0211560
## 2        1          2    10 1.8650191
## 3        1          3    10 2.5482706
## 4        2          1    10 0.9169518
## 5        2          2    10 1.3377192
## 6        2          3     8 2.1638021
```

dplyr is a useful package made by Hadley Wickam. It, along with ggplot2, can be very useful packages, although they can also be a course unto themself. Specifically, do not worry too much about the %>% symbol – you will only need to change variable names for this code.

Now we do the ANOVA fits. If we save the fitted values in our dataset, then we can obtain the cell means with similar code to what was used above.

```
#run the two way ANOVA
aovIBAL = aov(logDays~WeightGain * Duration, data = l10datBAL)
aovIUNB = aov(logDays~WeightGain * Duration, data = l10datUNB)

#save the fitted values
l10datBAL$fittedI = aovIBAL$fitted.values
l10datUNB$fittedI = aovIUNB$fitted.values

#get cells
group_by(l10datBAL, Duration, WeightGain) %>% summarise(count = n(), mean = mean(fittedI))
```

```
## Source: local data frame [6 x 4]
## Groups: Duration [?]
##
##   Duration WeightGain count      mean
##     <fctr>     <fctr> <int>     <dbl>
## 1        1          1    10 1.0211560
## 2        1          2    10 1.8650191
## 3        1          3    10 2.5482706
## 4        2          1    10 0.9169518
## 5        2          2    10 1.3377192
## 6        2          3    10 1.9949474
```

```
group_by(l10datUNB, Duration, WeightGain) %>% summarise(count = n(), mean = mean(fittedI))
```

```
## Source: local data frame [6 x 4]
## Groups: Duration [?]
##
##   Duration WeightGain count      mean
##     <fctr>     <fctr> <int>     <dbl>
## 1        1          1    10 1.0211560
## 2        1          2    10 1.8650191
## 3        1          3    10 2.5482706
## 4        2          1    10 0.9169518
## 5        2          2    10 1.3377192
## 6        2          3     8 2.1638021
```

Comparing this to the table above, we see that the fitted values are just the cell means.

b) Now fit just the main effects model. Notice that for the balanced design, the type I and type III sums of squares are the same, but for the unbalanced design, they differ. Why is this?

We fit the main effects model and obtain the Type I and Type III sum of squares for both models

```
#fit models
aovBAL = aov(logDays~WeightGain + Duration, data = l10datBAL)
aovUNB = aov(logDays~WeightGain + Duration, data = l10datUNB)

#Type I vs Type III Balanced
library(car) #needed for Anova function
```

```
##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode
```

```
anova(aovBAL)
```

```
## Analysis of Variance Table
##
## Response: logDays
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## WeightGain  2 16.9713  8.4856 16.0405 3.109e-06 ***
## Duration    1  2.3397  2.3397  4.4227   0.03997 *
## Residuals  56 29.6249  0.5290
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Anova(aovBAL, type = 'III')
```

```
## Anova Table (Type III tests)
##
## Response: logDays
##              Sum Sq Df F value    Pr(>F)
## (Intercept) 20.4117  1 38.5844 6.865e-08 ***
## WeightGain  16.9713  2 16.0405 3.109e-06 ***
## Duration     2.3397  1  4.4227   0.03997 *
## Residuals   29.6249 56
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Type I vs Type III Unbalanced
anova(aovUNB)
```

```
## Analysis of Variance Table
##
## Response: logDays
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## WeightGain  2 18.8062  9.4031 18.4468 7.838e-07 ***
## Duration    1  1.6394  1.6394  3.2162   0.07851 .
## Residuals  54 27.5261  0.5097
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Anova(aovUNB, type = 'III')
```

```
## Anova Table (Type III tests)
##
## Response: logDays
##              Sum Sq Df F value    Pr(>F)
## (Intercept) 19.2238  1 37.7128 1.012e-07 ***
## WeightGain  18.2652  2 17.9161 1.076e-06 ***
## Duration     1.6394  1  3.2162   0.07851 .
## Residuals   27.5261 54
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In a balanced design, the treatments in a two way Anova are orthogonal (i.e. knowing the category for one treatment doesn't tell you anything about the other treatment. In this case, there knowing the WeightGain group of a patient does not tell me anything about the duration of dyalysis). Thus, we see the same factoring out as was seen in the contrasts for HW6.

c) In the balanced data, which effects are significant? What is the appropriate reduced model for this data? Do this by first testing for an interaction, and then testing for individual effects if no interaction is present. Since any of these results would be scientifically significant, be sure to test for multiple tests.

Use Type III SS as we are interested in the effect of each variable *in the presence of* the other variables.

```
Anova(aovIBAL, type = 'III')
```

```
## Anova Table (Type III tests)
##
## Response: logDays
##                      Sum Sq Df F value    Pr(>F)
## (Intercept)         10.4276  1 19.4241 5.012e-05 ***
## WeightGain          11.7034  2 10.9003 0.0001056 ***
## Duration             0.0543  1  0.1011 0.7516994
## WeightGain:Duration  0.6357  2  0.5920 0.5567479
## Residuals           28.9892 54
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Here we see that the interaction is not significant, so we will look at the significance of Duration and Weight gain in a main effects model.

In the main effects model we have

```
Anova(aovBAL, type = 'III')
```

```
## Anova Table (Type III tests)
##
## Response: logDays
##              Sum Sq Df F value    Pr(>F)
## (Intercept) 20.4117  1 38.5844 6.865e-08 ***
## WeightGain  16.9713  2 16.0405 3.109e-06 ***
## Duration     2.3397  1  4.4227   0.03997 *
## Residuals   29.6249 56
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In this case both WeightGain is significant at the 0.05/3 significance level (note the Bonferonni correction), however Duration is not significant at this level.

Since Duration is fairly close to being significant, an appropriate model for this data would be `logDays ~ WeightGain + Duration`, however we cannot yet conclude that Duration is significant.

d) Regardless of the model chosen in part C, continue with the interaction model. Estimate main effects and interactions for the balanced model for each factor level and each treatment. How would you get confidence intervals or perform tests for the indvidual main effects and interaction effects?

The `aov` function we are using is an ANOVA specific version of the `lm` function. If we try to get the coefficient and standard error estimates in our usual way, we get

```
summary(aovIBAL)
```

```
##                     Df Sum Sq Mean Sq F value   Pr(>F)
## WeightGain           2 16.971   8.486  15.807 3.94e-06 ***
## Duration             1  2.340   2.340   4.358   0.0416 *
## WeightGain:Duration  2  0.636   0.318   0.592   0.5567
## Residuals           54 28.989   0.537
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Which is not what we want. In order to get the summary information from the `aov` object, we do

```
summary.lm(aovIBAL)
```

```
##
## Call:
## aov(formula = logDays ~ WeightGain * Duration, data = l10datBAL)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.33772 -0.51121  0.06302  0.62926  1.17950
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)              1.0212     0.2317   4.407 5.01e-05 ***
## WeightGain2              0.8439     0.3277   2.575   0.0128 *
## WeightGain3              1.5271     0.3277   4.661 2.10e-05 ***
## Duration2               -0.1042     0.3277  -0.318   0.7517
## WeightGain2:Duration2  -0.4231     0.4634  -0.913   0.3653
## WeightGain3:Duration2  -0.4491     0.4634  -0.969   0.3368
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7327 on 54 degrees of freedom
## Multiple R-squared:  0.4076, Adjusted R-squared:  0.3528
## F-statistic: 7.431 on 5 and 54 DF,  p-value: 2.301e-05
```

From here, obtaining confidence intervals and performing tests on individual parameters is completed as usual.

e) Construct the ANOVA table for the reduced model.

```
Anova(aovBAL, type = 'III')
```

```
## Anova Table (Type III tests)
##
## Response: logDays
##              Sum Sq Df F value    Pr(>F)
## (Intercept) 20.4117  1 38.5844 6.865e-08 ***
## WeightGain  16.9713  2 16.0405 3.109e-06 ***
## Duration     2.3397  1  4.4227   0.03997 *
## Residuals   29.6249 56
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

g) Contrasts can be completed as they are done in One-Way Anova. Create a contrast for testing Slight WeightGain against Moderate to Severe WeightGain, and Moderate WeightGain against Severe WeightGain.

```
Con1 = c(-1, 1/2, 1/2)
Con2 = c(0, 1/2, -1/2)
Cons = cbind(Con1, Con2)
aovCONBAL = aov(logDays~C(WeightGain, Con1, 1) + C(WeightGain, Con2, 1) + Duration, data = l10datBAL)
summary(aovCONBAL)
```

```
##                         Df Sum Sq Mean Sq F value   Pr(>F)
## C(WeightGain, Con1, 1)  1 12.479  12.479  23.589 9.98e-06 ***
## C(WeightGain, Con2, 1)  1  4.492   4.492   8.492  0.00512 **
## Duration                1  2.340   2.340   4.423  0.03997 *
## Residuals              56 29.625   0.529
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```