610 R12 **Prof Colleen F. Moore**

Analysis of variance for Unbalanced Between Groups designs in R

For Psychology 610 University of Wisconsin--Madison

R is very touchy about unbalanced designs, partly because it includes several ways of calculating the SS. The Type I method is the default in the 'aov' module. Type II and Type III sums of squares can be tested using 'Anova' (note capital A) in the 'car' package. Install the 'car' package in R, then activate it for your session using the 'library(car)' command.

When using R for unbalanced designs, *the contrast coding of your factors is absolutely critical*. I illustrate how dramatically this can alter your results below. Other programs are more idiot-proof for unbalanced designs.

Contents of this tutorial:

I. Bring in data and arrange for analysis. This section includes some data transformation tricks for recoding variables, and constructing centered dummy codes for main effects and interactions.

II. Anova with Type I, Type II and Type III SS's.

III. Same analyses but conducted with multiple regression using the codes laboriously constructed in Part I.

IV. Non-orthogonality of the design illustrated by correlating the main effect contrast codes across all observations. Shows how to construct a correlation matrix.

Summary of R steps:

Step 1) set options for contrasts to 'sum' or 'poly',

Step 2) run anova in 'aov',

Step 3) use 'Anova' (capital A) in the 'car' package to extract Type II or Type III SS's from the 'aov' run.

Summary of R code:

```
> options(contrasts=c("contr.sum","contr.poly")) # set the contrast type R will use as a default
> model = aov(dv ~ A*B*C)  # make sure A, B and C are 'factors'
> library(car)  # activate the 'car' package
> Anova(model,type=c("III"))
```

Warnings: 1) Type III SS's are sensitive to the specific contrasts used. 2) make sure you make your independent variables into factors.

An Example in detail

An Unbalanced Between Groups anova is illustrated here with data from a 2(sex of animal) x 2(prenatal alcohol exposed or not) x 2(prenatal stress exposed or not). The dependent variable is fallypride uptake in the striatum, a measure of dopamine system function in that area of the brain.

I. Bring in data and set up the grouping variable codes. The original data set had sex coded as 1 and 2, and the animal's condition was coded 1=alc only, 2=control, 3=stress only, and 5=alc+stress. These codes needed to be turned into two separate variables for alcohol and stress. I did this in a very plodding manner to make sure I didn't make mistakes. Notice that I have entered 'NaN' for missing data.

```
> pet1.data=read.table(pipe("pbpaste"),header=T) # I pasted the data to the clipboard
> pet1.data
     ID sex condition      fal      fmt
1  AR54   2         1 12.88000 6.32000
2  AR56   2         1 14.80000 5.38000
3  AR58   1         1 11.46000 6.64000
4  AR61   1         2 13.48800 6.25700
5  AR66   2         3 17.34000 4.73000
6  AR67   1         3 12.05000 5.78000
7  AR69   1         2 10.15000 4.93600
8  AR70   2         5 18.94000 7.22000
9  AR71   2         3 16.67000 4.74000
10 AR72   1         5 14.39000 5.55000
11 AR76   2         2 14.85000 7.55000
12 AR77   2         2  9.37000 5.59000
13 AR78   2         3 11.91330 5.20512
14 AR80   1         2 14.61000 8.34000
15 AR82   1         1 12.41000 6.76000
16 AR87   1         1 13.92000 5.63000
17 AR88   2         3 17.10000 7.39000
18 AR89   1         2 11.35000 5.91000
19 AR95   2         2 10.53000 5.78000
21 AS03   1         2      NaN      NaN
22 AS06   2         5 23.07000 5.90000
24 AS47   1         1 14.49000 4.96000
25 AS48   1         5 19.26000 5.26000
26 AS51   1         2 15.72000 5.89500
29 AS59   1         2 18.73730 4.28064
30 AS61   1         1 12.85910 4.38298
31 AS62   1         2 11.79000 6.07000
32 AS64   1         1 13.72260 4.79091
34 AS80   2         1  8.98263 5.33459
36 AS89   1         2 14.53000 6.85000
37 AT35   1         5 14.69000 5.25000
40 AT83   1         3 17.04100 3.76925
41 AT85   2         3 19.03000 5.36000
42 AT86   1         1 12.04270 4.39087
43 AT94   2         5 11.70000 6.51000
44 AU03   2         5 18.74000 6.77000
45 AU08   2         3 14.09000 4.89000
46 AU09   2         5  9.67000 5.91000
47 AU11   1         5 15.03000 4.63000
```

A. Create codes for the prenatal stress variable, –1 and 1 for no stress and stress exposed. I call it 'tress' so it won't have the first letter 's', and I won't confuse it with sex of animal.

> pet2.data=transform(pet1.data, tress=ifelse(condition > 2, 1,-1)) # the ifelse statement sets the value of the named variable for a whole vector according whether the test in parentheses (condition > 2) is true or false.
> pet2.data # I want to see the results before I continue

```
ID sex condition      fal      fmt      tress
1  AR54   2         1 12.88000 6.32000    -1
2  AR56   2         1 14.80000 5.38000    -1
3  AR58   1         1 11.46000 6.64000    -1
4  AR61   1         2 13.48800 6.25700    -1
5  AR66   2         3 17.34000 4.73000     1
6  AR67   1         3 12.05000 5.78000     1
. . .
44 AU03   2         5 18.74000 6.77000     1
45 AU08   2         3 14.09000 4.89000     1
46 AU09   2         5  9.67000 5.91000     1
47 AU11   1         5 15.03000 4.63000     1
```

B. Create codes for the prenatal alcohol variable, -1 and 1 for no alc and alc exposed. This is trickier because the original conditions aren't in order. I folded the original condition in the middle by subtracting a constant and then taking the absolute value. This is a good trick to learn for recoding variables.

> pet3.data=transform(pet2.data,alc=ifelse(abs(condition - 3) >= 2,1,-1))
> pet3.data

```
     ID sex condition      fal      fmt tress alc
1  AR54   2         1 12.88000 6.32000    -1   1
2  AR56   2         1 14.80000 5.38000    -1   1
3  AR58   1         1 11.46000 6.64000    -1   1
4  AR61   1         2 13.48800 6.25700    -1  -1
5  AR66   2         3 17.34000 4.73000     1  -1
6  AR67   1         3 12.05000 5.78000     1  -1
. . .
44 AU03   2         5 18.74000 6.77000     1   1
45 AU08   2         3 14.09000 4.89000     1  -1
46 AU09   2         5  9.67000 5.91000     1   1
47 AU11   1         5 15.03000 4.63000     1   1
```

C. Create codes for sex of animal, -1 for females, 1 for males.

> pet4.data=transform(pet3.data,sexcode=ifelse(sex==1,-1,1))
> pet4.data

```
     ID sex condition      fal      fmt tress alc sexcode
1  AR54   2         1 12.88000 6.32000    -1   1       1
2  AR56   2         1 14.80000 5.38000    -1   1       1
3  AR58   1         1 11.46000 6.64000    -1   1      -1
4  AR61   1         2 13.48800 6.25700    -1  -1      -1
. . .
45 AU08   2         3 14.09000 4.89000     1  -1       1
46 AU09   2         5  9.67000 5.91000     1   1       1
47 AU11   1         5 15.03000 4.63000     1   1      -1
```

D. Create interaction codes. You don't need to create the next codes, which represent all the interactions by multiplying codes together. I am doing this solely for instructional purposes, so we can see how to calculate Type III SS's using multiple regression later.

```
> attach(pet4.data)  # attach the data set to simplify naming variables for calculations
> alcXtress=alc*tress
> alcXsex=alc*sexcode
> tressXsex=tress*sexcode
> way3=alc*tressXsex
```

E. Now combine the interaction codes with the data:

```
> pet5.data=cbind(pet4.data,alcXtress,alcXsex,tressXsex,way3)  # cbind means "column bind" variables together
> pet5.data
     ID sex condition      fal     fmt tress alc  sexcode alcXtress alcXsex tressXsex way3
1  AR54   2         1 12.88000 6.32000    -1   1        1        -1       1        -1   -1
2  AR56   2         1 14.80000 5.38000    -1   1        1        -1       1        -1   -1
3  AR58   1         1 11.46000 6.64000    -1   1       -1        -1      -1         1    1
4  AR61   1         2 13.48800 6.25700    -1  -1       -1         1       1         1   -1
5  AR66   2         3 17.34000 4.73000     1  -1        1        -1      -1         1   -1
. . .
43 AT94   2         5 11.70000 6.51000     1   1        1         1       1         1    1
44 AU03   2         5 18.74000 6.77000     1   1        1         1       1         1    1
45 AU08   2         3 14.09000 4.89000     1  -1        1        -1      -1         1   -1
46 AU09   2         5  9.67000 5.91000     1   1        1         1       1         1    1
47 AU11   1         5 15.03000 4.63000     1   1       -1         1      -1        -1   -1
```

Get rid of extraneous things, and then attach the final data set.

```
> rm(pet4.data)
> detach(pet4.data)
> attach(pet5.data)

        The following object(s) are masked _by_ .GlobalEnv :

         alcXsex alcXtress tressXsex way3
```

F. Save the constructed data set in your computer so you never have to go through this re-coding again.

```
> library(MASS)  # activate the 'MASS' package
> write.matrix(pet5.data,file="petwithcodes",sep=" ")  # I like a blank as a separator
```

because it can be read by some legacy statistical software I use, but many prefer a comma because files that are comma-delimited can be read by many many programs very easily.

II. Create factors and carry out anova using 'aov'

```
> Alc=factor(alc)
> Tress=factor(tress)
> Sex=factor(sexcode)
```

A. Run 'aov', and look at the default output from 'aov'. This gives ***Type I SS***. In Type I SS, the order of entry of the factors matters, and almost no one is interested in these. Here I show how order affects the results in this example.
> modfull=aov(fal~Alc*Tress*Sex) # this specifies the full model with all possible main effects and interactions of the 3 factors.
> summary(modfull,intercept=T) # ask for the summary. This gives **Type I** Sums of Squares, not the Type III we are used to in Psychology.

```
                Df Sum Sq Mean Sq  F value    Pr(>F)
(Intercept)      1 7771.1  7771.1 834.9437 < 2.2e-16 ***
Alc              1    0.2     0.2   0.0206  0.886908
Tress            1   81.0    81.0   8.7021  0.006111 **
Sex              1    1.4     1.4   0.1547  0.696840
Alc:Tress        1    1.7     1.7   0.1863  0.669088
Alc:Sex          1    0.8     0.8   0.0865  0.770642
Tress:Sex        1   11.4    11.4   1.2242  0.277330
Alc:Tress:Sex    1    2.7     2.7   0.2880  0.595437
Residuals       30  279.2     9.3
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
1 observation deleted due to missingness
```

Type I SS's are sensitive to the order in which the factors are entered. Here I reordered the factors in the 'aov' statement. The SS below are close, but notice that Alc is reduced in size, and Tress is a touch larger.
> modreorder=aov(fal~Tress*Alc*Sex)
> summary(modreorder,intercept=T)

```
                Df   Sum Sq  Mean Sq  F value    Pr(>F)
(Intercept)      1   7771.1   7771.1 834.9437 < 2.2e-16 ***
Tress            1     81.2     81.2   8.7225  0.006058 **
Alc              1 0.001548 0.001548   0.0002  0.989796
Sex              1      1.4      1.4   0.1547  0.696840
Tress:Alc        1      1.7      1.7   0.1863  0.669088
Tress:Sex        1     11.9     11.9   1.2803  0.266807
Alc:Sex          1      0.3      0.3   0.0305  0.862640
Tress:Alc:Sex    1      2.7      2.7   0.2880  0.595437
Residuals       30    279.2      9.3
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
1 observation deleted due to missingness
```

B. Means and standard errors.
> model.tables(modfull,"means",se=T) # R tells us that the design is unbalanced, and so we need to use another method to get the standard errors.

```
Design is unbalanced - use se.contrast() for se's
Tables of means
Grand mean

14.30044

 Alc
      -1     1
   14.23 14.37
rep 19.00 19.00
```

R gives *weighted means*, that is the marginals are weighted by the cell n.

To calculate u*nweighted means*, you just average together the cell means that go into the marginal you want (i.e., take the mean of the means). There might be a function to calculate unweighted means in R, but I haven't found it yet.

```
 Tress
       -1     1
    12.99 15.92
rep 21.00 17.00

 Sex
       -1    1
    14.46 14.1
rep 21.00 17.0

 Alc:Tress
     Tress
Alc   -1     1
  -1  13.173 15.682
  rep 11.000  8.000
  1   12.779 16.141
  rep 10.000  9.000

 Alc:Sex
     Sex
Alc   -1     1
  -1  14.522 13.905
  rep 10.000  9.000
  1   14.417 14.309
  rep 11.000  8.000

 Tress:Sex
     Sex
Tress -1     1
  -1  13.427 11.894
  rep 15.000  6.000
  1   15.379 16.217
  rep  6.000 11.000

 Alc:Tress:Sex
, , Sex = -1

     Tress
Alc   -1     1
  -1  13.797 14.546
  rep  8.000  2.000
  1   12.986 15.843
  rep  7.000  4.000

, , Sex = 1

     Tress
Alc   -1     1
  -1  11.583 16.024
  rep  3.000  6.000
  1   12.221 16.424
  rep  3.000  5.000
```

C. Use 'Anova' (with capital 'A') in 'car' package to get Type II and/or Type III SS
. This is where things are tricky. Type III SS solutions depend on the contrasts that R uses internally. We used the factors, not our contrast codes, in the 'aov' run. R made its own default contrast coding. The best thing to do is to set the options in R:

> options(contrasts=c("contr.sum","contr.poly")) # you MUST specify this before you run 'aov'
> modfull=aov(fal~ Alc*Tress*Sex)
> library(car) # activate the 'car' package
> Anova(modfull,type=c("III")) # ask for type III SS from the aov run we did just above.

```
Anova Table (Type III tests)

Response: fal
               Sum Sq Df  F value    Pr(>F)    
(Intercept)    6271.9  1 673.8665 < 2.2e-16 ***
Alc               1.1  1   0.1217  0.729664    
Tress            73.1  1   7.8583  0.008783 ** 
Sex               0.4  1   0.0443  0.834804    
Alc:Tress         1.7  1   0.1832  0.671697    
Alc:Sex           0.1  1   0.0159  0.900448    
Tress:Sex        12.4  1   1.3300  0.257915    
Alc:Tress:Sex     2.7  1   0.2880  0.595437    
Residuals       279.2 30                       
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

***** If you run the 'aov' analysis without specifying the contrasts 'options'** given above, you'll get pretty strange results for Type III SS's. Here's an example of what I got without the options statement set properly. Whoops, where'd my significant effect of Tress go? And all the SS's are quite different, except the residual and the 3-way interaction, which do match.

Incorrect analysis shown here for illustration:

```
Anova Table (Type III tests)

Response: fal
                Sum Sq Df  F value    Pr(>F)    
(Intercept)    1522.84  1 163.6171 1.113e-13 ***
Alc               2.45  1   0.2635    0.6115    
Tress             0.90  1   0.0963    0.7584    
Sex              10.69  1   1.1486    0.2924    
Alc:Tress         4.36  1   0.4689    0.4988    
Alc:Sex           2.24  1   0.2411    0.6270    
Tress:Sex        12.12  1   1.3018    0.2629    
Alc:Tress:Sex     2.68  1   0.2880    0.5954    
Residuals       279.22 30                       
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Type II SS's are not as sensitive to the contrasts as Type III:
> Anova(modfull) # this will give type II SS tests, the default for 'Anova' (with a capital A). Now the significant Tress effect is back, but which anova table is "correct"?? The SS don't match up, though they are in the ballpark. Actually, this issue is 'which do you *prefer* to answer your research question'.

```
Anova Table (Type II tests)

Response: fal
                Sum Sq Df F value   Pr(>F)
```

```
Alc             0.061  1  0.0065 0.936201
Tress          79.366  1  8.5272 0.006582 **
Sex             1.071  1  0.1150 0.736854
Alc:Tress       1.738  1  0.1867 0.668775
Alc:Sex         0.283  1  0.0305 0.862640
Tress:Sex      11.394  1  1.2242 0.277330
Alc:Tress:Sex   2.681  1  0.2880 0.595437
Residuals     279.220 30
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

D. Re-do the anova in 'aov' but using our codes instead of having R create factors
> modfull2=aov(fal~alc+tress+sexcode+alcXtress+alcXsex+tressXsex+way3) # we list all the variables that we want in the analysis. Remember that we created the interaction codes and named them with the 'X' as part of the name earlier. The 'X' isn't an operator. Also, these codes are contrast codes, so we use them directly without making factors out of them.

> summary(modfull2,intercept=T) # ask for the results, these will be Type I SS. These results match the first results exactly. 'aov' gives the same results (Type I SS) whether you use contrast codes you created yourself, or make factors and let R use those. Order o entry still matters for Type I SS.

```
            Df Sum Sq Mean Sq  F value    Pr(>F)
(Intercept)  1 7771.1  7771.1 834.9437 < 2.2e-16 ***
alc          1    0.2     0.2   0.0206  0.886908
tress        1   81.0    81.0   8.7021  0.006111 **
sexcode      1    1.4     1.4   0.1547  0.696840
alcXtress    1    1.7     1.7   0.1863  0.669088
alcXsex      1    0.8     0.8   0.0865  0.770642
tressXsex    1   11.4    11.4   1.2242  0.277330
way3         1    2.7     2.7   0.2880  0.595437
Residuals   30  279.2     9.3
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
1 observation deleted due to missingness
```

Next, use 'Anova' to get Type II and Type III SS solutions from the run using the our own coded variables. The Type III SS are now in the same ballpark as the Type I and Type II SS. Not only that, but the Type III SS here will match those from SPSS, SAS, or my legacy DOS software, BMDP. But now the Type II SS don't match the Type II SS from before; Type II and Type III SS match each other now.

Main point: the coding of your predictor variables is ***critical*** when you ask for Type III SS in R !!! But we still face the question of which is better for our purposes.
> Anova(modfull2,type=c("III"))

```
Anova Table (Type III tests)

Response: fal
            Sum Sq Df  F value    Pr(>F)
(Intercept) 6271.9  1 673.8665 < 2.2e-16 ***
alc            1.1  1   0.1217  0.729664
```

```
tress          73.1  1   7.8583  0.008783 **
sexcode         0.4  1   0.0443  0.834804
alcXtress       1.7  1   0.1832  0.671697
alcXsex         0.1  1   0.0159  0.900448
tressXsex      12.4  1   1.3300  0.257915
way3            2.7  1   0.2880  0.595437
Residuals     279.2 30
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

> Anova(modfull2)

```
Anova Table (Type II tests)

Response: fal
           Sum Sq Df F value    Pr(>F)
alc         1.132  1  0.1217 0.729664
tress      73.140  1  7.8583 0.008783 **
sexcode     0.412  1  0.0443 0.834804
alcXtress   1.705  1  0.1832 0.671697
alcXsex     0.148  1  0.0159 0.900448
tressXsex  12.379  1  1.3300 0.257915
way3        2.681  1  0.2880 0.595437
Residuals 279.220 30
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

III. Re-do all the analyses using multiple regression 'lm' instead of 'aov'.

Type III SS find the variance accounted for by each variable *after* the other variables are entered in the model. With all these interactions, this will be a nuisance, and we'll be grateful for the Type III option in the 'car' package.

A. First run the full model in 'lm', and get the R-squared of the regression run. You can also get the analysis of variance table of the regression by using 'anova' (with a small 'a').

> full=lm(fal~alc+tress+sexcode+alcXtress+alcXsex+tressXsex+way3)
> summary(full)

```
Call:
lm(formula = fal ~ alc + tress + sexcode + alcXtress + alcXsex +
    tressXsex + way3)

Residuals:
    Min      1Q  Median      3Q     Max
-6.7540 -1.8320  0.2594  1.8182  6.6460

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  14.1779     0.5462  25.959  < 2e-16 ***
alc           0.1905     0.5462   0.349  0.72966
tress         1.5311     0.5462   2.803  0.00878 **
sexcode      -0.1149     0.5462  -0.210  0.83480
alcXtress     0.2338     0.5462   0.428  0.67170
alcXsex       0.0689     0.5462   0.126  0.90045
```

```
tressXsex     0.6299     0.5462   1.153  0.25791
way3         -0.2931     0.5462  -0.537  0.59544
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.051 on 30 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared: 0.2622,      Adjusted R-squared: 0.09007
F-statistic: 1.523 on 7 and 30 DF,  p-value: 0.1974
```

B. Now run the regression again, but **omitting** in turn each main effect term.

> alcmain=lm(fal~tress+sexcode+alcXtress+alcXsex+tressXsex+way3)
> summary(alcmain)

```
Call:
lm(formula = fal ~ tress + sexcode + alcXtress + alcXsex + tressXsex +
    way3)

Residuals:
    Min      1Q  Median      3Q     Max
-6.6054 -1.8983  0.2506  1.7751  6.7946

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 14.19638    0.53584  26.494  < 2e-16 ***
tress        1.55283    0.53484   2.903  0.00675 **
sexcode     -0.13955    0.53385  -0.261  0.79551
alcXtress    0.21685    0.53625   0.404  0.68871
alcXsex      0.06747    0.53836   0.125  0.90108
tressXsex    0.60189    0.53254   1.130  0.26705
way3        -0.22048    0.49770  -0.443  0.66084
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.007 on 31 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared: 0.2592,      Adjusted R-squared: 0.1159
F-statistic: 1.808 on 6 and 31 DF,  p-value: 0.1300
```

> tressmain=lm(fal~alc+sexcode+alcXtress+alcXsex+tressXsex+way3) # omit 'tress'
> summary(tressmain)

```
Call:
lm(formula = fal ~ alc + sexcode + alcXtress + alcXsex + tressXsex +
    way3)

Residuals:
    Min      1Q  Median      3Q     Max
-5.5597 -2.2337  0.0535  1.5501  7.8403

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  14.0420     0.6012  23.357   <2e-16 ***
alc           0.3656     0.5996   0.610    0.547
sexcode       0.4689     0.5580   0.840    0.407
alcXtress     0.3822     0.6007   0.636    0.529
alcXsex      -0.1559     0.5970  -0.261    0.796
tressXsex     0.6183     0.6035   1.024    0.314
way3         -0.4913     0.5985  -0.821    0.418
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.371 on 31 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared: 0.06896,     Adjusted R-squared: -0.1112
F-statistic: 0.3827 on 6 and 31 DF,  p-value: 0.8844
```

> sexmain=lm(fal~alc+tress+alcXtress+alcXsex+tressXsex+way3) # omit sex
> summary(sexmain)

```
Call:
lm(formula = fal ~ alc + tress + alcXtress + alcXsex + tressXsex +
    way3)

Residuals:
    Min      1Q  Median      3Q     Max
-6.8436 -1.8288  0.2233  1.8762  6.5564

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 14.17879    0.53767  26.371   <2e-16 ***
alc          0.20538    0.53316   0.385   0.7027
tress        1.48724    0.49706   2.992   0.0054 **
alcXtress    0.25064    0.53185   0.471   0.6408
alcXsex      0.05777    0.53515   0.108   0.9147
tressXsex    0.64007    0.53556   1.195   0.2411
way3        -0.30626    0.53416  -0.573   0.5705
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.003 on 31 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared: 0.2611,      Adjusted R-squared: 0.1181
F-statistic: 1.826 on 6 and 31 DF,  p-value: 0.1264
```

C. Subtract R-squared's by hand to get the contribution of each variable after all the other variables are included, **or use the 'anova' (small 'a')** function to have R calculate the significance of the R-squared difference for you. Notice that we only calculated the R-squared values that we need for the 3 main effects here. In order to calculate the interaction SS's you would have to omit each interaction term and obtain the R-sq value.

> anova(full,alcmain) # by listing two models, R will compare them. The SS difference is negative, but it doesn't matter. List them in the opposite order in the 'anova' statement

```
Analysis of Variance Table

Model 1: fal ~ alc + tress + sexcode + alcXtress + alcXsex + tressXsex +
    way3
Model 2: fal ~ tress + sexcode + alcXtress + alcXsex + tressXsex + way3
  Res.Df     RSS Df Sum of Sq      F Pr(>F)
1     30 279.220
2     31 280.352 -1    -1.132 0.1217 0.7297
```

> anova(full,sexmain)

```
Analysis of Variance Table

Model 1: fal ~ alc + tress + sexcode + alcXtress + alcXsex + tressXsex +
    way3
Model 2: fal ~ alc + tress + alcXtress + alcXsex + tressXsex + way3
```

```
  Res.Df     RSS Df Sum of Sq       F Pr(>F)
1     30 279.220
2     31 279.632 -1    -0.412 0.0443 0.8348
> anova(full,tressmain)
Analysis of Variance Table

Model 1: fal ~ alc + tress + sexcode + alcXtress + alcXsex + tressXsex +
    way3
Model 2: fal ~ alc + sexcode + alcXtress + alcXsex + tressXsex + way3
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     30 279.22
2     31 352.36 -1    -73.14 7.8583 0.008783 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This laborious regression method creates Type III SS's and F's that match those from the Type III SS's that we calculated using our orthogonal contrast codes and 'Anova' (capital A).

IV. How non-orthogonal is this design?

There are two ways that I think about non-orthogonality.

First, do the SS add up the SS total? When a design is unbalanced, they ∑SS > SS total. This is because some of the SS is overlapping. **Second**, do the predictor variables correlate with each other?

Second, we can look at the correlation between the predictor variables that we created. We constructed our codes for the main effects as –1 vs 1 contrasts, and all the interactions were constructed by multiplying the main effect contrast codes together. Two contrasts are not orthogonal is $\sum cj * ck \neq 0$. The cross product of two variables is the numerator of the correlation coefficient, so let's make a correlation table of the data.

The result shows that tress and sexcode are correlated 0.37. Also, tress and alc are correlated only .07. And alc correlates with the 'way3' the 3 way interaction codes. It would be nicer to have an orthogonal design wouldn't it?? Too bad we can't prenatally randomly assign animals to turn out to be male or female.

```
> cor(pet5.data, use="pairwise.complete.obs") # ask for correlation matrix
           ID         sex   condition          fal          fmt       tress         alc
ID         NA          NA          NA           NA           NA          NA          NA
sex        NA  1.00000000  0.25179072  0.110157454  0.188496160  0.37433155 -0.02917864
condition  NA  0.25179072  1.00000000  0.433514073  0.064119421  0.85445225  0.16953595
fal        NA  0.11015745  0.43351407  1.000000000  0.000745151  0.46315103  0.02249309
fmt        NA  0.18849616  0.06411942  0.000745151  1.000000000 -0.11583400 -0.04590049
tress      NA  0.37433155  0.85445225  0.463151029 -0.115834002  1.00000000  0.07427291
alc        NA -0.02917864  0.16953595  0.022493085 -0.045900495  0.07427291  1.00000000
sexcode    NA  1.00000000  0.25179072  0.110157454  0.188496160  0.37433155 -0.02917864
alcXtress  NA -0.11968254  0.47740149  0.076931948  0.333756000 -0.01595767 -0.12664952
alcXsex    NA -0.02917864  0.06402687  0.014046055  0.248399594 -0.13263019 -0.12894737
tressXsex  NA -0.08618128 -0.12598329  0.138699767  0.131278312 -0.08618128 -0.12824729
way3       NA  0.06149733 -0.03181471 -0.104682687  0.068917230 -0.04278075  0.38462756
              sexcode   alcXtress     alcXsex   tressXsex        way3
ID                 NA          NA          NA          NA          NA
sex        1.00000000 -0.11968254 -0.02917864 -0.08618128  0.06149733
condition  0.25179072  0.47740149  0.06402687 -0.12598329 -0.03181471
fal        0.11015745  0.07693195  0.01404605  0.13869977 -0.10468269
fmt        0.18849616  0.33375600  0.24839959  0.13127831  0.06891723
tress      0.37433155 -0.01595767 -0.13263019 -0.08618128 -0.04278075
alc       -0.02917864 -0.12664952 -0.12894737 -0.12824729  0.38462756
```

```
sexcode    1.00000000 -0.11968254 -0.02917864 -0.08618128  0.06149733
alcXtress -0.11968254  1.00000000  0.38786415 -0.06000686 -0.11968254
alcXsex   -0.02917864  0.38786415  1.00000000  0.09404802 -0.13263019
tressXsex -0.08618128 -0.06000686  0.09404802  1.00000000  0.02585438
way3       0.06149733 -0.11968254 -0.13263019  0.02585438  1.00000000
Warning message:
In cor(pet5.data, use = "pairwise.complete.obs") :
  NAs introduced by coercion
```