

Chapter 11

Factorial ANOVA**11.1 Numerical and Visual Summary of the Data, Including Means Plots**

In order to inspect the data let's look at the mean scores and standard deviations of the first story from the Obarow (2004) data (to follow along with me, import the Obarow.Story1.sav file as `obarrowStory1`), which are found in Table 11.1. I used the `numSummary()` command to give me these descriptive statistics.

```
numSummary(obarrowStory1[,c("gncs1.1", "gncs1.2")],
groups=obarrowStory1$Treatment1, statistics=c("mean", "sd"))
```

Table 11.1 Numerical Summary from Obarow (2004) Story 1

<i>Immediate Gains</i>		<i>N</i>	<i>Mean</i>	<i>SD</i>
No pictures	No music	15	.73	1.62
	Music	15	.93	1.80
Pictures	No music	17	1.47	1.01
	Music	17	1.29	1.72
<i>Delayed Gains</i>		<i>N</i>	<i>Mean</i>	<i>SD</i>
No pictures	No music	15	1.40	1.88
	Music	14	1.64	1.50
Pictures	No music	17	1.53	2.12
	Music	17	1.88	2.03

Considering that there were 20 vocabulary words tested from the story, we can see from the gain scores that most children did not learn very many words in any of the conditions. Actually the least number of words that any children knew in the first story was 11, so there were not actually 20 words to learn for any children. Also, it appears that in every category gains were larger when considered in the delayed post-test (one week after the treatment) than in the immediate post-test (the day after treatment ended). The numerical summary also shows that standard deviations are quite large, in most cases larger than the mean.

For the visual summary, since we are looking at group differences, boxplots would be an appropriate way of checking the distribution of data. Figure 11.1 shows boxplots of the four different groups, with the left side showing the immediate gain score and the right side showing the delayed gain score.

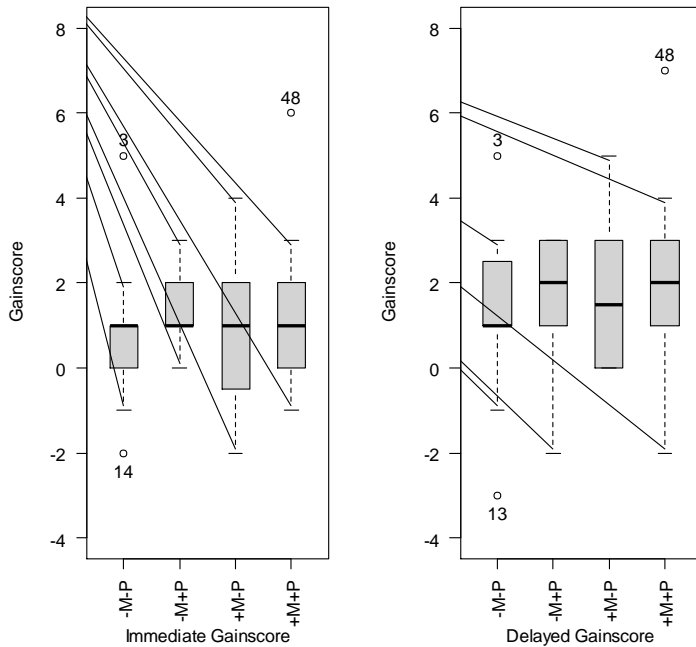


Figure 11.1 Boxplots of the Obarow Story 1 data.

The boxplots show a number of outliers for two of the four groups, meaning that the assumption of normality is already violated. None of the boxplots looks exactly symmetric, and some distributions have extreme skewness.

I'd like you to consider one more issue about the data in the obarowStory1 file. Notice which variables are factors (they show a count of the levels):

summary(obarowStory1)

```

gender      grade      rdglvl      PicturesT1      MusicT1
male :34   Min. :1.000   Min. :1.000   no pictures:30   no music:32
female:30  1st Qu.:1.000 1st Qu.:2.000 pictures :34     music  :32
          Median :2.000 Median :2.000
          Mean  :2.016 Mean  :2.094
          3rd Qu.:3.000 3rd Qu.:3.000
          Max.  :3.000 Max.  :3.000

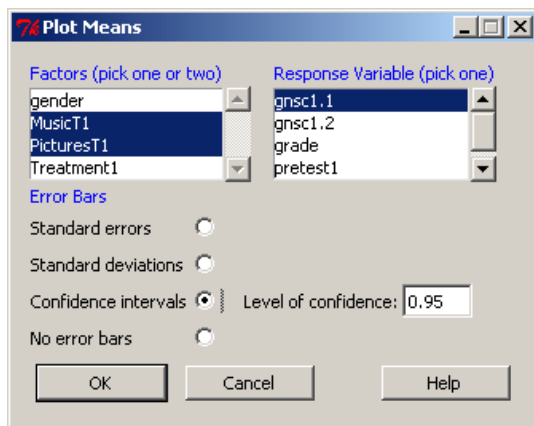
          Treatment1  pretest1      gnsc1.1
No Music No Pictures :17  Min. :11.00   Min.  : -2.000
No Music Yes Pictures :17  1st Qu.:15.00  1st Qu.: 0.000
Yes Music No Pictures :15  Median :16.00  Median : 1.000
Yes Music Yes Pictures:15  Mean  :15.59   Mean  : 1.125
          3rd Qu.:17.00  3rd Qu.: 2.000
          Max.  :17.00   Max.  : 6.000
    
```

Notice that `PicturesT1` is a factor, as is `MusicT1`. But so is `Treatment1`, which basically combines the factors of pictures and music. For my factorial ANOVA, if I want to see the effect of music and pictures separately, basically I'll need to have these as separate factors. But in some cases I want to have just one factor that combines both of these elements, and that is why I created the `Treatment1` factor as well. Make sure you have the factors whose interaction you want to check as separate factors in your data set. If you see a need later to combine them, as I did, you can create a new factor.

11.1.1 Means Plots

One type of graphic that is often given in ANOVA studies is a means plot. Although this graphic is useful for discerning trends in the data, it often provides only one piece of information per group—the mean. As a way of visualizing an interaction, means plots can be helpful, but they are graphically impoverished, since they lack much information. These types of plots will be appropriate only with two-way and higher ANOVAs (since a means plot with a one-way ANOVA would have only one line).

In R Commander, open `GRAPHS > PLOT OF MEANS`. In the dialogue box I choose two factors (see Figure 11.2), one displaying whether the treatment contained music or not (`MusicT1`) and the other displaying whether the treatment contained pictures or not (`PicturesT1`). Choose the categorical variables in the “Factors” box and the continuous dependent variable in the “Response Variable” box. This choice results in the means plot in Figure 11.3.



Note that you have several options for displaying error bars in R. I chose 95% confidence intervals.

Figure 11.2 Creating a means plot in R Commander.

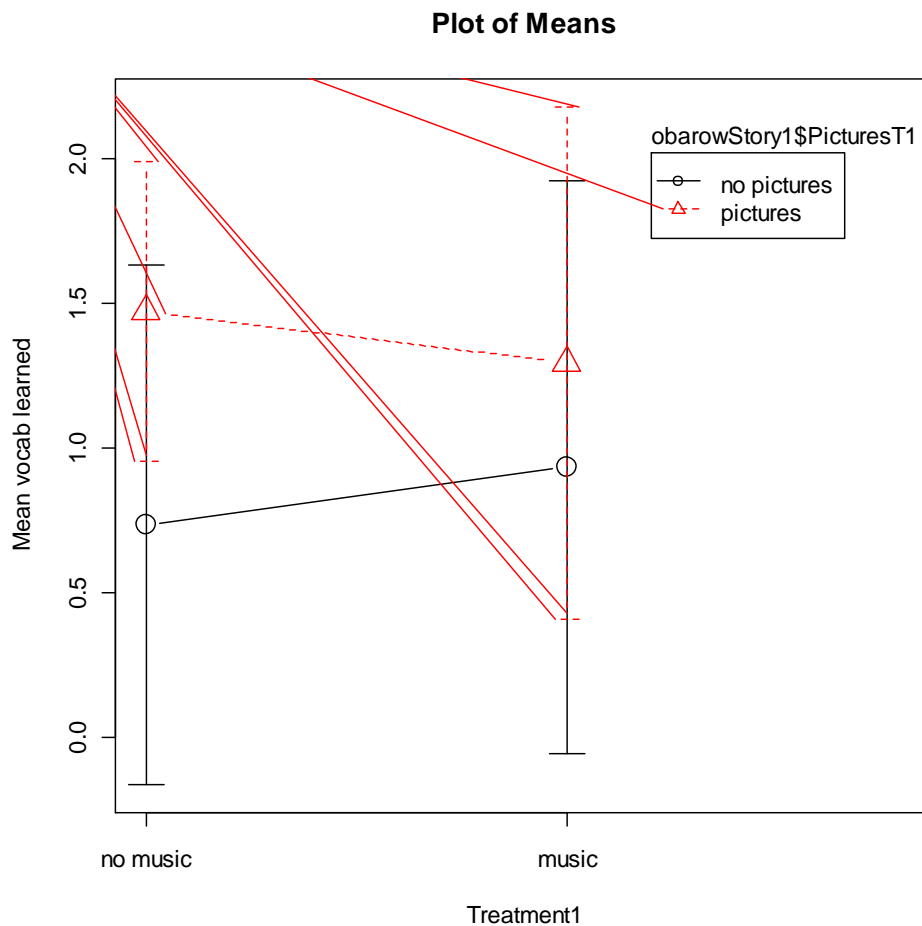


Figure 11.3 Means plot of first treatment in Obarow (2004) data with music and picture factors.

Figure 11.3 shows that those who saw pictures overall did better (the dotted line is higher than the straight line at all times). However, there does at least seem to be a trend toward interaction, because when music was present this was helpful for those who saw no pictures, while it seemed to be detrimental for those who saw pictures. We'll see in the later analysis, however, that this trend does not hold statistically, and we can note that the difference between the mean of pictures with no music (about 1.5) and the mean of pictures with music (about 1.3) is not a very large difference. Our graph makes this difference look larger. If the scale of your graph is small enough, the lines may not look parallel, but when you pull back to a larger scale that difference can fade away. Visually, no interaction would be shown by parallel lines, meaning that the effects of pictures were the same whether music was present or not.

Notice also that, because confidence intervals around the mean are included, the means plot has enriched information beyond just the means of each group. The fact that the confidence intervals overlap can be a clue that there may not be a statistical difference between groups as well, since the intervals overlap.

The R code for this plot is:

```
plotMeans(obarowStory1$gnsc1.1, obarowStory1$MusicT1,
          obarowStory1$PicturesT1, error.bars="conf.int")
```

<code>plotMeans(DepVar, factor 1, factor 2)</code>	Returns a means plot with one dependent and one or two independent variables.
<code>obarow\$gnsc1.1</code>	The response, or dependent variable.
<code>obarow\$MusicT1, obarow\$PicturesT1</code>	The factor, or independent variables.
<code>error.bars="conf.int"</code>	Plots error bars for the 95% confidence interval on the means plot; other choices include "sd" for standard deviation, "se" for standard error, and "none".

The command `interaction.plot()` can also be used but is a bit trickier, because usually you will need to reorder your data in order to run this command. Nevertheless, if you want to plot more than one dependent variable at a time, you could use `interaction.plot` in R (see the visual summary document for Chapter 12, "Repeated-Measures ANOVA," for an example of how to do this).

One problem you might run into when creating a means plot is that the factors levels are not in the order or set-up you want for the graph you want. With the Obarow data this wasn't an issue, as there were several choices for factors (the `Treatment1` factor coded for four different conditions, while the `MusicT1` coded just for \pm Music and the `PicturesT1` coded just for \pm Pictures). If I had used the `Treatment1` factor, which contains all four experimental conditions, I would have gotten the graph in Figure 11.4.

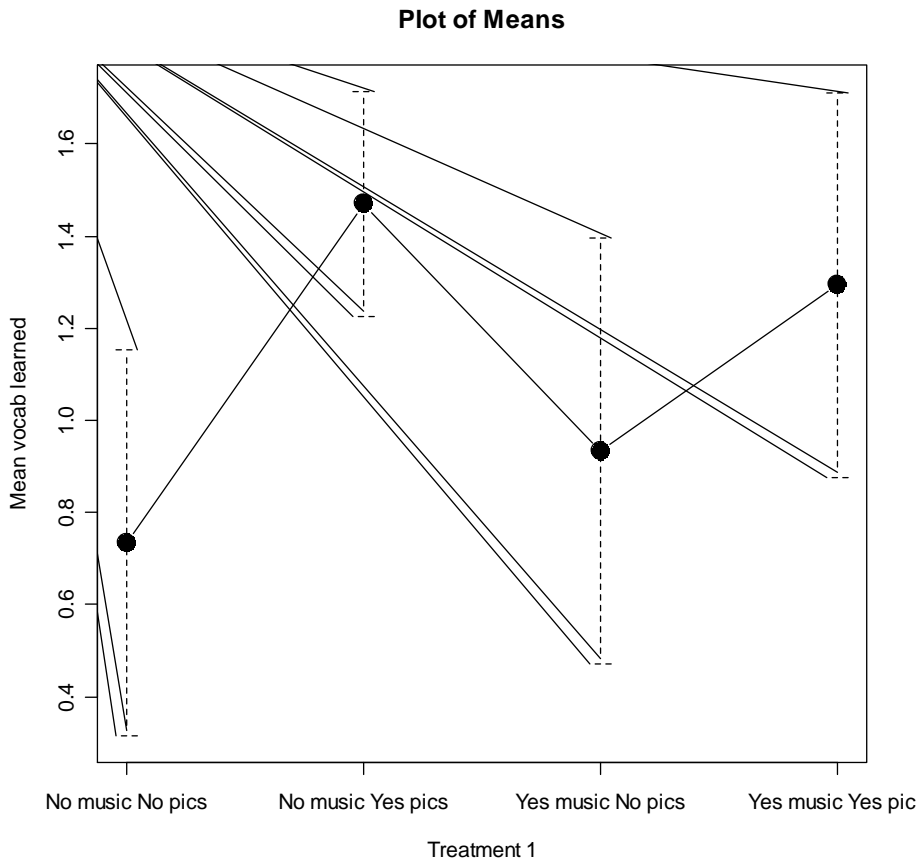


Figure 11.4 Means plot of first treatment in Obarow (2004) data with experimental factor.

Figure 11.4 is not as useful as Figure 11.3, because it doesn't show any interaction between the factors, so I wouldn't recommend using a means plot in this way. However, I will use this example of a factor with four levels to illustrate how to change the order of the factors. You can change the order of factors through the `ordered()` command. First, see what the order of factor levels is and the exact names by listing the variable:

```
obarrowStory1$Treatment1
4 Levels: -M-P -M+P +M-P +M+P
(note that -M means "No Music" and -P means "No Pictures")
```

Next, specify the order you want the levels to be in:

```
obarrowStory1$Treatment1=ordered(obarrowStory1$Treatment1, levels=
c("-M+P", "+M+P", "+M-P", "-M-P"))
```

Now run the means plot again.

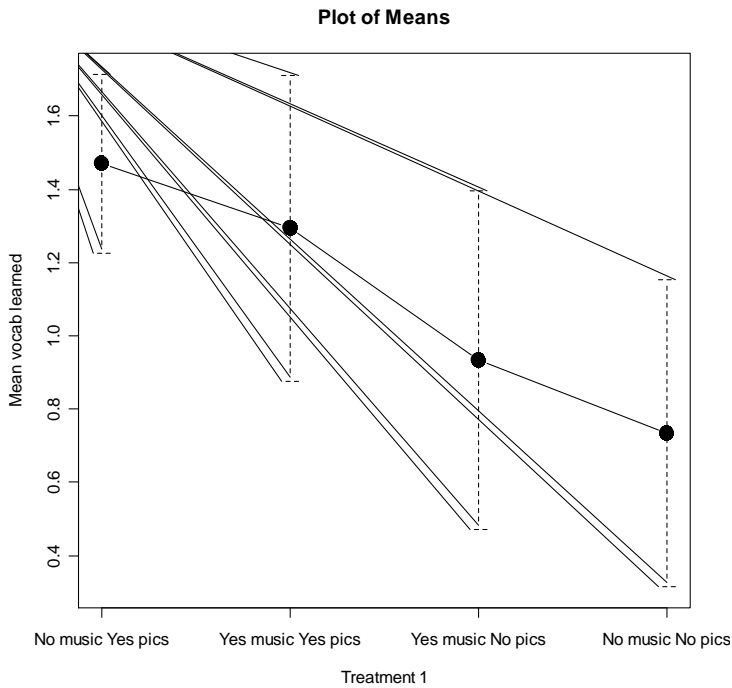


Figure 11.5 Obarow (2004) means plot with rearranged factors.

The result, shown in Figure 11.5, is rearranged groups showing a progression from highest number of vocabulary words learned to the lowest. Note that this works fine for creating a graph, but you will want to avoid using ordered factors in your ANOVA analysis. The reason is that, if factors are ordered, the results in R cannot be interpreted in the way I discuss in this chapter.

Creating a Means Plot in R

From the R Commander drop-down menu, open **GRAPHS > PLOT OF MEANS**. Pick one or two independent variables from the “Factors” box and one dependent variable from the “Response Variable” box. Decide what kind of error bars you’d like.

The simple command for this graph in R is:

```
plotMeans(obarowStory1$gnsc1.1, obarowStory1$MusicT1,
  obarowStory1$PicturesT1, error.bars="conf.int")
```

Formatted: Font: Italic

Deleted: "="

Deleted: ")

11.2 Putting Data in the Correct Format for a Factorial ANOVA

The Obarow data set provides a good example of the kind of manipulations one might need to perform to put the data into the correct format for an ANOVA. A condensation of Obarow's original data set is shown in Figure 11.6.

It can be seen in Figure 11.6 that Obarow had coded students for gender, grade level (this was a number from 1 to 3 given to them by their teachers), and then treatment group for the particular story. Obarow then recorded their score on the pre-test, as it was important to see how many of the 20 targeted words in the story the children already knew. As can be seen in the pre-test columns, most of the children already knew a large number of the words. Their scores on the same vocabulary test were recorded after they heard the story three times (immediate post-test) and then again one week later (delayed post-test). The vocabulary test was a four-item multiple choice format.

	gender	grade	rdgyl	trtmnt1	pretest1	posttest1	delpost1	trtmnt2	pretest2	posttest2
1	1	1	2	1	18	17	17	2	18	18
2	1	1	2	1	17	15	18	2	15	17
3	2	1	1	2	18	19	19	3	12	15
4	2	1	1	2	17	19	18	3	18	19
5	2	1	2	3	15	16	16	4	15	16
6	1	2	1	3	15	16	20	4	18	19
7	2	1	1	4	16	15	16	1	14	18

Figure 11.6 The original form of Obarow's data.

The format found here makes perfect sense for entering the data, but it is not the form we need in order to conduct the ANOVA. First of all, we want to look at gain scores, so we will need to calculate a new variable from the pre-test and post-test columns. In R Commander we can do this by using DATA > MANAGE VARIABLES IN ACTIVE DATA SET > COMPUTE NEW VARIABLE (note that I will not walk you through these steps here, but you can look at the online document "Understanding the R environment.Manipulating Variables_Advanced Topic" for more information on the topic).

Then there is an issue we should think about before we start to rearrange the data. Although the vocabulary words were chosen to be unfamiliar to this age of children, there were some children who achieved a perfect score on the pre-test. Children with such high scores will clearly not be able to achieve any gains on a post-test. Therefore, it is worth considering whether there should be a cut-off point for children with certain scores on the pre-test. Obarow herself did not do this, but it seems to me that, unless the children scored about a 17 or below, there is really not much room for them to improve their vocabulary in any case. With such a cut-off level, for the first story there are still 64 cases out of an original 81, which is still quite a respectable size.

To cut out some rows in R, the simplest way is really to use the R Console:

```
new.obarow <- subset(obarow, subset=PRETEST1<18)
```

The next thing to rearrange in the original data file is the coding of the independent variables. Because music and pictures are going to be two separate independent variables, we will need two columns, one coded for the inclusion of music and one coded for the inclusion of illustrations in the treatment. The way Obarow originally coded the data, trtmnt1, was coded

as 1, 2, 3, or 4, which referred to different configurations of music and pictures. Using this column as it is, we would be able to conduct a one-way ANOVA (with `trtmnt1` as the IV), but we would not then be able to test for the presence of interaction between music and pictures. In order to have two separate factors, I could recode this in R like this for the music factor (and then also create a picture factor):

```
obarrow$MusicT1 <- recode(obarrow$trtmnt1, '1="no music"; 2="no music"; 3="yes music"; 4="yes music"; ', as.factor.result=TRUE)
```

After all of this manipulation, I am ready to get to work on my ANOVA analysis! If you are following along with me, we are going to use the SPSS file `Obarow.Story1`, imported as `obarrowStory1`. Notice in Figure 11.7 that I now have a gain score (`gnsc1.1`), and two columns coding for the presence of music (`MusicT1`) and pictures (`PicturesT1`). There are only 64 rows of data, since I excluded any participants whose pre-test score was 18 or above.

	gender	grade	rdgwl	MusicT1	PicturesT1	Treatment1	pretest1	gnsc1.1
1	1	1	1	1	1	1	14	1.00
2	2	3	3	1	1	1	14	1.00
3	1	1	2	1	2	2	13	3.00
4	2	2	2	1	2	2	13	2.00
5	1	1	3	2	1	3	13	4.00
6	2	1	2	2	1	3	13	2.00
7	1	1	3	2	2	4	11	6.00
8	1	1	3	2	2	4	13	-1.00

Figure 11.7 The “long form” of Obarow’s data.

11.3 Performing a Factorial ANOVA

11.3.1 Performing a Factorial ANOVA Using R Commander

To replicate the type of full factorial ANOVA done in most commercial statistics programs, you can use a factorial ANOVA command in R Commander. Choose `STATISTICS > MEANS > MULTI-WAY ANOVA`, as seen in Figure 11.8.

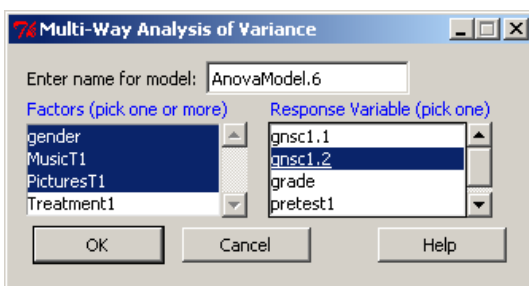


Figure 11.8 Conducting a three-way ANOVA in R Commander.

With the Obarow (2004) data set I am testing the effect of three independent variables (gender, presence of music, and presence of pictures), which I choose from the Factors box. The response variable I will use is the gain score for the immediate pre-test (gnsc1.1).

The R Commander output will give an ANOVA table for the full factorial model, which includes all main effects and all interaction effects:

```
y=Gender + Music + Pictures + Gender*Music + Gender*Pictures + Music*Pictures +
Gender*Music*Pictures
```

where “*” means an interaction.

```
Anova Table (Type II tests)
```

```
Response: gnsc1.1
```

	Sum Sq	Df	F value	Pr(>F)
gender	9.990	1	4.3504	0.04157 *
MusicT1	0.365	1	0.1588	0.69176
PicturesT1	2.388	1	1.0399	0.31223
gender:MusicT1	5.551	1	2.4174	0.12563
gender:PicturesT1	0.270	1	0.1175	0.73308
MusicT1:PicturesT1	1.669	1	0.7266	0.39763
gender:MusicT1:PicturesT1	1.369	1	0.5959	0.44338
Residuals	128.601	56		

```
---
```

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

Any effects which are statistical at the .05 level or lower are noted with asterisks. Here we see on the first line that the main effect of gender is statistical ($F_{1,56}=4.35$, $p=.04$). In other words, separating the numbers strictly by gender and not looking at any other factor, there is a difference between how males and females performed on the task.

Main effects may not be of much interest if any of the interaction effects are statistical, since if we knew that, for example, the three-way interaction were statistical, this would tell us that males and females performed differently depending on whether music was present *and* depending on whether pictures were present. In that case, our interest would be in how all three factors together affected scores, and we wouldn't really care about just the plain difference between males and females. However, for this data there are no interaction effects.

The last line, which is called Residuals, is the error line. This part is called error because it contains all of the variation we can't account for by the other terms of our equation. Thus, if the sum of squares for the error line is large relative to the sum of squares for the other terms (which it is here), this shows that the model does not account for very much of the variation.

Notice that the numbers you can obtain here may be different from the identical analysis run in a different statistical program such as SPSS. This is because R is using Type II sums of squares (and this is identified clearly in the output), while SPSS or other statistical programs may use Type III sums of squares (for more information on this topic, see section 11.4.3 of the SPSS book, *A Guide to Doing Statistics in Second Language Research Using SPSS*, pp. 311–313).

If you need to calculate mean squares (possibly because you are calculating effect sizes that ask for mean squares), these can be calculated by using the sums of squares and degrees of

freedom given in the R Commander output. The mean square of any row is equal to the sum of squares divided by the df of the row. Thus, the mean square of Gender is $9.99/1=9.99$.

The next part of the output from R Commander gives descriptive statistics (mean, sd, and count) for the permutation of the independent variables in the order they are placed in the regression. The first three lines of the ANOVA table above show the order in which the variables were placed: gender, then music, and then pictures. The last variable placed in the equation is **Pictures**, so the descriptive statistics show the number of vocabulary items used when, first of all, pictures are present versus absent, and splits that data up into variables of gender and music. The following output shows the mean scores for this situation:

```
, , PicturesT1 = no pictures

      MusicT1
gender  no music music
male   0.7500000  2.2
female 0.7142857  0.3

, , PicturesT1 = pictures

      MusicT1
gender  no music  music
male   1.5833333 1.777778
female 1.2000000 0.750000
```

If this wasn't the organization of the data that you wanted, you could just change the order of the factors in your ANOVA model (by hand in R, since there is no way to change the order in R Commander).

In reporting on a factorial ANOVA you will want to report descriptive statistics (which R Commander output gives, as shown here), information about each main effect and interaction in your model (shown in the ANOVA table), effect sizes for factors, and the overall effect size of multiple R^2 .

The partial eta-squared effect size for main effects and interactions (if all of the factors are fixed, not random) can be calculated by using the formula $_{\text{partial}} \hat{\eta}^2 = \frac{SS_{\text{effect}}}{SS_{\text{effect}} + SS_{\text{error}}}$ where

SS =sum of squares. For the case of the main effect of gender, then partial eta-squared= $9.99/(9.99+128.601)=.072$.

The multiple R squared is not found in the output that R Commander calls for. You can obtain it by doing a summary of the linear model that R Commander constructs. Here the model was automatically labeled as AnovaModel.6, as seen in Figure 11.8.

summary(AnovaModel.6))

[output cut here]

```
Residual standard error: 1.515 on 56 degrees of freedom
Multiple R-Squared: 0.1483,    Adjusted R-squared: 0.04188
F-statistic: 1.393 on 7 and 56 DF,  p-value: 0.2263
```

The summary first provides coefficient estimates for the default level of the factors compared to the other factors. The last three lines of the regression output are shown above. The first line, which is labeled “Residual standard error,” is the square root of the error variance from the ANOVA table above (128.601) with the error degrees of freedom. We want this number to be as small as possible, for this is the part of the data that we can’t explain. The second line gives the overall effect size for the full factorial regression, while the adjusted R squared gives the same thing but adjusted for positive bias. So this full factorial model accounts for 15% of the variance in scores or, adjusting for bias, the full model accounts for only 4.2% of the variance. The third line is a model that is rarely of interest, but it tests the hypothesis that “performance is a function of the full set of effects (the main effects and the interactions)” (Howell, 2002, p. 464).

Conducting a Factorial ANOVA Using R Commander

1. STATISTICS > MEANS > MULTI-WAY ANOVA.
2. Choose your independent variables under the “Factors (pick one or more)” box on the left. Highlight more than one by holding down your Ctrl button while you left-click the factors. Choose a dependent variable in the “Response Variable (pick one)” box. Click OK.
3. To obtain R^2 and adjusted R^2 , take the ANOVA model that R Commander creates and run a summary on it:

```
summary(AnovaModel)
```

Formatted: Font: Italic

Formatted: Font: Italic

Formatted: Font: Italic

Deleted:

11.3.2 Performing a Factorial ANOVA Using R

The code for the full factorial performed by R Commander was clearly apparent in the previous section. What I will demonstrate in this section is how to perform an ANOVA where we will not try to fit the (one) model to the data, but instead fit the data to the model by searching for the best fit of the data. This procedure is identical to what was seen in the section on Regression (in the online documents “Multiple Regression.Finding the best fit” and “Multiple Regression.Further steps in finding the best fit”), since ANOVA in R is modeled as a regression equation, and then commands like `Anova()` reformat the results into an ANOVA table.

As with regression, it makes more sense to find the minimally adequate model for the data and only later examine plots which evaluate the suitability of assumptions such as homogeneity of variances, normality of distribution, etc. I will assume that my reader has already looked at the section on finding the minimally adequate model in the online document “Multiple Regression.Finding the best fit,” but, as a review, the steps I will follow to find the minimally adequate model are (following Crawley, 2007):

1. Create a full factorial model.
2. Examine the output for statistical terms.
3. Create a new model that deletes unstatistical entries, beginning with largest terms first and working backwards to simpler terms.
4. Compare the two models, and retain the new, simpler model if it does not cause a statistical increase in deviance.

Using the Obarow (2004) data contained in `Obarow.Story.1` (imported as `obarrow` into R), I will first create the full factorial model. Remember that, instead of writing out the entire seven-parameter model (with 1 three-way interaction, 3 two-way interactions, and 3 main effects), using the asterisk (*) notation creates the full model.

```
attach(obarrow)
model=aov(gnsc1.1~gender*MusicT1*PicturesT1)
summary(model)
```

The `summary()` command produces an ANOVA table, as was seen above in the previous section with R Commander. This summary shows that there is only one statistical entry, the main effect of gender. The choice of `aov()` instead of `lm()` for the regression model means that the `summary()` command will return an ANOVA table instead of regression output.

Although only one variable is statistical, just as we saw with regression modeling, the best way to go about our ANOVA analysis is to perform stepwise deletion, working backwards by removing the largest interactions first, and testing each model along the way. We could simply stop after this first pass and proclaim that gender is the only variable that is statistical, but that would not be finding the minimally adequate model. I suggested in the chapter on regression and I will also suggest here that finding the minimally adequate model is a much more informative analysis than a simple full factorial model with reporting on which main effects and interactions are statistical. What we will find is that, by performing the minimally adequate model, the stepwise removal of variables may change the analysis along the way, since order matters in a data set where the number of persons in groups is not strictly equal (as is true in this case).

Now we will simplify the model by removing the least significant terms, starting with the highest-order terms first (in our case, we'll remove the third-order interaction first).

```
model2=update(model,~.-gender:MusicT1:PicturesT1)
```

Remember, the syntax here must be exactly right! The syntax in the parentheses that comes after the name of the model to update (here, called just "model") is "comma tilde period minus." This syntax means that `model2` will take the first argument (`model`) as it is and then subtract the three-way interaction. Now we compare the updated model to the original model using `anova()`.

```
anova(model,model2)
```

```
Analysis of Variance Table

Model 1: gnsc1.1 ~ gender * MusicT1 * PicturesT1
Model 2: gnsc1.1 ~ gender + MusicT1 + PicturesT1 + gender:MusicT1 + gender:PicturesT1 +
  MusicT1:PicturesT1
  Res.Df    RSS Df Sum of Sq    F Pr(>F)
1     56 128.60
2     57 129.97 -1    -1.3685 0.5959 0.4434
```

The fact that the ANOVA is not statistical tells us that our newer `model2` does not have statistically higher deviance than the original `model`, and thus it is not worse at explaining what is going on than the original model. We prefer `model2` because it is simpler. We can now look at a summary of `model2` and decide on the next least statistical argument to take out from among the two-way interactions:

```
summary(model2)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
gender	1	11.863	11.8627	5.2026	0.02631 *
MusicT1	1	0.434	0.4340	0.1903	0.66428
PicturesT1	1	2.438	2.4382	1.0693	0.30547
gender:MusicT1	1	4.558	4.5579	1.9989	0.16285
gender:PicturesT1	1	0.069	0.0693	0.0304	0.86223
MusicT1:PicturesT1	1	1.669	1.6685	0.7318	0.39590
Residuals	57	129.969	2.2802		

The least statistical of the two-way interactions is `gender:PicturesT1` (it has the highest p -value), so we will remove that in the next model.

```
model3=update(model2,~. gender:PicturesT1)
anova(model2,model3)
```

Analysis of Variance Table

```
Model 1: gnsc1.1 ~ gender + MusicT1 + PicturesT1 + gender:MusicT1 + gender:PicturesT1 +
  MusicT1:PicturesT1
Model 2: gnsc1.1 ~ gender + MusicT1 + PicturesT1 + gender:MusicT1 + MusicT1:PicturesT1
Res.Df  RSS Df Sum of Sq    F Pr(>F)
1      57 129.97
2      58 130.24 -1  -0.26975 0.1183 0.7321
```

Again, there is no difference in deviance between the models, so we will leave this interaction term out. Examining the summary of `model3`:

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
gender	1	11.863	11.8627	5.2829	0.02516 *
MusicT1	1	0.434	0.4340	0.1933	0.66184
PicturesT1	1	2.438	2.4382	1.0858	0.30172
gender:MusicT1	1	4.558	4.5579	2.0298	0.15960
MusicT1:PicturesT1	1	1.468	1.4681	0.6538	0.42207
Residuals	58	130.239	2.2455		

we see that the next interaction term with the highest p -value is `MusicT1:PicturesT1`.

I will leave it to the reader to verify that, by continuing this procedure, you will reach `model7`, which contains only the variable of `gender`. At this point, we should compare a model with only `gender` to the null model, which contains only the overall average score and total deviance. The way to represent the null model is simply to put the number “1” as the explanatory variable after the tilde. If there is no difference between these two models, then what we have found is that none of our variables do a good job of explaining what is going on!

```
model8=aov(gnsc1.1~1)
anova(model7,model8)
```

Analysis of Variance Table

```

Model 1: gnsc1.1 ~ gender
Model 2: gnsc1.1 ~ 1
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1      62 139.14
2      63 151.00 -1   -11.863 5.2861 0.02488 *

```

Fortunately, here we find a statistical difference in the amount of deviance each model explains, so we can stop and claim that the best model for the data is one with only one explanatory variable, that of gender.

`summary(model7)`

```

          Df Sum Sq Mean Sq F value Pr(>F)
gender     1  11.863  11.8627   5.2861 0.02488 *
Residuals 62 139.137   2.2441

```

The regression output for the same model will give more information:

`summary.lm(model7)`

[output cut here]

```

Residual standard error: 1.498 on 62 degrees of freedom
Multiple R-squared:  0.07856,    Adjusted R-squared:  0.0637
F-statistic: 5.286 on 1 and 62 DF,  p-value: 0.02488

```

The multiple R^2 value shows that the variable of gender accounted for about 8% of the variance in scores, a fairly small amount. Notice that the F-statistic and p -value given in the regression output are exactly the same as in the ANOVA table, but this is only because there is only one term in the equation.

A shortcut to doing the stepwise deletion by hand is the command `boot.stepAIC` from the `bootStepAIC` library with the original full factorial model (as mentioned in Chapter 7, this function is much more accurate and parsimonious than the `step` command found in R's base library):

`boot.stepAIC(model,data=obarrow) #need to specify data set even though attached!`

```

Initial Model:
gnsc1.1 ~ gender * MusicT1 * PicturesT1

Final Model:
gnsc1.1 ~ gender + MusicT1 + gender:MusicT1

```

	Step	Df	Deviance	Resid. Df	Resid. Dev	AIC
1				56	128.6008	60.66111
2	- gender:MusicT1:PicturesT1	1	1.3685372	57	129.9693	59.33859
3	- gender:PicturesT1	1	0.2697513	58	130.2391	57.47128
4	- MusicT1:PicturesT1	1	1.4680660	59	131.7071	56.18866
5	- PicturesT1	1	2.3880898	60	134.0952	55.33870

The full model has an **AIC** (Akaike information criterion) score of 60.66. As long as this measure of fit goes *down*, the step procedure will continue to accept the more simplified model, as we did by hand. In other words, the lower the AIC, the better (the AIC is a measure of the tradeoff between degrees of freedom and the fit of the model). The second line of the steps indicated on the print-out shows that a model that has removed the three-way interaction has a lower AIC than the full model (59.34 as opposed to 60.66). The step procedure's last step is to stop when there are still three terms in the equation.

Let's check out the difference between this model that `boot.stepAIC` kept (I'll create a `model9` for this) and the model we retained by hand (`model7`).

```
model9=aov(gnsc1.1~gender+MusicT1+gender:MusicT1)
summary.lm(model9)
```

data deleted . . .

```
Residual standard error: 1.495 on 60 degrees of freedom
Multiple R-squared: 0.112, Adjusted R-squared: 0.06755
F-statistic: 2.521 on 3 and 60 DF, p-value: 0.06631
```

We find that `model9` has a slightly smaller residual standard error on fewer degrees of freedom than `model7` (1.495 for `model9`, 1.498 for `model7`), and a higher R^2 value (11.2% for `model9`, 7.9% for `model8`), but a model with more parameters will always have a higher R^2 (there are 64 participants in the Obarow study, and if we had 64 parameters the R^2 fit would be 100%), so this is not surprising.

An ANOVA table shows that neither the `MusicT1` nor the interaction `gender:MusicT1` is statistical in this three-parameter model.

```
summary(model9)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
gender	1	11.863	11.8627	5.3079	0.02471 *
MusicT1	1	0.434	0.4340	0.1942	0.66103
gender:MusicT1	1	4.608	4.6080	2.0618	0.15622
Residuals	60	134.095	2.2349		

Just to make sure what we should do, let's compare `model7` to `model9` using `anova()`.

```
anova(model7,model9)
```

```
Analysis of Variance Table

Model 1: gnsc1.1 ~ gender
Model 2: gnsc1.1 ~ gender + MusicT1 + gender:MusicT1
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1     62 139.14
2     60 134.09  2     5.042 1.128 0.3304
```

An ANOVA finds no statistical difference between these models. All other things being equal, we prefer the model that is simpler, so in the end I will retain the model with only the main effect of gender.

The message here is that, although `boot.stepAIC` can be a useful tool, you will also want to understand and know how to hand-calculate a stepwise deletion procedure. Crawley (2007) notes that step procedures can be generous and leave in terms which are not statistical, and you will want to be able to check by hand and take out any terms which are not statistical. Another situation where a stepping command is not useful is when you have too many parameters relative to your sample size, called overparameterization (see the online section “Multiple Regression. Further steps in finding the best fit” for more information on this). In this case, you should work by hand and test only as many parameters as your data size will allow at a time.

Now that the best fit of the data (so far!) has been determined, it is time to examine ANOVA (regression) assumptions by calling for diagnostic plots:

```
plot(model7)
detach(obarow)
```

Plotting the model will return four different diagnostic plots, shown in Figure 11.9.

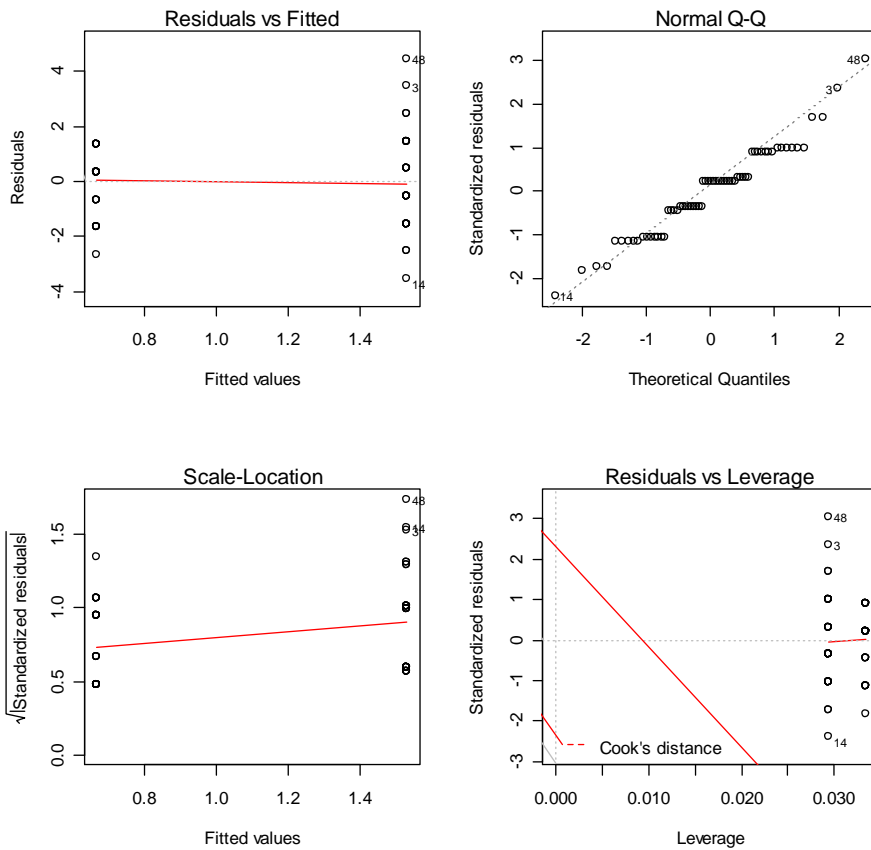


Figure 11.9 Diagnostic plots for a minimally adequate model of Obarow (2004).

The first plot, shown in the upper left-hand corner of Figure 11.9, is a residual plot, and if the data have equal variances it should show only a random scattering of data. Since the data values are so discrete we don't really see a scattering at all, but the outliers on the right-hand side of the plot (we know they are outliers because they are numbered) may indicate that the residuals are increasing as the fitted values get larger, indicating a problem with heteroscedasticity. The second plot, shown in the upper right-hand corner, tests the data against the assumption of a normal distribution. The data should be reasonably linear, and, although the discreteness means that the middle points form little parallel lines instead of following the diagonal, there also does not seem to be any clear pattern of non-normality (such as an S-shaped or J-shaped pattern in the Q-Q plot). The third plot, shown in the lower left-hand corner, is, like the first plot, meant to detect heteroscedasticity, and if there were problems the data would show a pattern and not be randomly scattered. The last plot, in the lower right-hand corner, is meant to test for influential points, and these are identified as 3, 14, and 48 in the Obarow (2004) data set. Crawley (2002, p. 240) says the two biggest problems for regression/ANOVA are heterogeneity of variances (non-constant variance across the x-axis) and non-normal errors. There seem to be problems in this model with both non-normal points (some outliers with influence) and possibly heterogeneity of variance.

One way of dealing with non-normality would be to run the analysis again and leave out the influential points, and report the results of both ways to the reader (although by now you probably know this type of analysis would not be my recommendation). If you were to do this properly, you would start again with the full model, but just to show how this could be accomplished in R, here is the code for our minimally adequate model with the three influential points identified in plot 4 removed:

```
model.adj=aov(gnsc1.1[-c(3,14,48)]~gender[-c(3,14,48)],data=obarrow)
summary.lm(model.adj)
```

The problem is that there are still outliers, and new influential points pop up when we remove the previous three points. Thus, my preference would be to use a robust ANOVA order to address the issues of both non-normality and possible heterogeneity of variances.

The final summary that I would make of the Obarow (2004) data is that there was no difference (at least, in the immediate gain score) for the effect of music and pictures, but there was an effect for gender. One idea to take away from the Obarow data is that, when expected results fail to materialize, it often pays to explore your data a bit to look for other explanatory variables. Obarow (2004) looked at her data using only the music and pictures variables and found no statistical results.

Conducting a Factorial ANOVA Using R

There is no simple formula for conducting a factorial ANOVA in R, but there are steps to take:

1. Create a full factorial model of your data using either `aov()` or `lm()`.

```
model1=aov(gnsc1.1~gender*MusicT1*PicturesT1, data=obarrow)
```

2. Begin to look for the best model that fits the data. You might start with the step function from the `bootStepAIC` library:

```
boot.stepAIC (model1)
```

3. Confirm `boot.stepAIC`'s choices with your own stepwise deletion procedure:

- a. Look at ANOVA table of model (use `anova()`) and decide which one term to delete in the next model.
- b. Choose the highest-order term to delete first. If there is more than one, choose the one with the highest *p*-value. If an interaction term is statistical, always retain its component main effects (for example, if `gender*MusicT1` is statistical, you must keep both `gender` and `MusicT1` main effects).
- c. Create a new model by using `update`:

```
model2=update(model1, ~.-gender:MusicT1:PicturesT1, data=obarrow)
```

- d. Compare the new and old model using `anova`:

```
anova(model1, model2)
```

- e. If there is no statistical difference between models, retain the newer model and continue deleting more terms. If you find a statistical difference, keep the older model and you are finished.
- f. Finally, you may want to compare your model with the null model, just to make sure your model has more explanative value than just the overall mean score:

```
model.null=lm(gnsc1.1~1,data=obarrow)
```

4. When you have found the best fit, check the model's assumptions of normality and constant variances (homoscedasticity):

```
plot(maximalModel)
```

Formatted: Font: Italic

Formatted: Font: Italic

Formatted: Font: Italic

Deleted: ()

Formatted: Font: (Default) Arial

Formatted: Indent: Left: 0", First line: 0"

Deleted: ()

Deleted:

Formatted: Font: Italic

Formatted: Indent: First line: 0"

Formatted: Indent: Left: 0", First line: 0"

Formatted: Font: (Default) Arial

Formatted: Indent: Left: 0", First line: 0"

The results of a factorial ANOVA should include information about the type of ANOVA you used, and by this I mean information about the regression equation that was used in your analysis. Your reader should know whether you just used a full factorial model (as many commercial statistical programs like SPSS do) or whether you searched for the minimal adequate model, as I have recommended. Reports of these two different types of analyses will be quite different. If you do a minimal adequate model you should report on the steps you took in the search for the minimal adequate model and report the change in deviance and

possibly also AIC scores for the models (check the deviance and AIC of models by hand by using the `deviance()` and `AIC()` commands).

Results should also include information about how you checked regression assumptions of the data, along with descriptive statistics such as the N, means, and standard deviations. Besides F-values and *p*-values for terms in the regression equation, you should provide effect sizes. For post-hoc tests or planned comparisons, report *p*-values.

Here is an example of a report you might write for the Obarow (2004) data if you just looked at the full model:

A 2×2×2 full factorial ANOVA examining the effects of music, pictures, and gender on the increase in participants' vocabulary size found a statistical effect for the main effect of gender only ($F_{1,56} = 4.486$, $p = .039$, partial $\eta^2 = .07$). None of the effect sizes for any other terms of the ANOVA were above partial $\eta^2 = .03$. Females overall learned more vocabulary after hearing the story three times ($M = 1.53$, $sd = 1.76$, $n = 33$) than males did ($M = .67$, $sd = 1.12$, $n = 11$). The effect size shows that this factor accounted for $R^2 = 7\%$ of the variance in the data, which is a small effect. None of the other main effects or interactions were found to be statistical. I did not check any of the assumptions of ANOVA.

Here is an example of a report you might write for the Obarow (2004) data if you searched for the minimal adequate model:

A search for a minimal adequate model was conducted starting with a full factorial model with three independent variables of gender, use of music, and use of pictures. The ANOVA model began with all three main effects plus all two-way interactions and the three-way interaction between terms. Deleting the terms and then checking for differences between models, my minimal model was a model that retained only the main effect of gender ($F_{1,62} = 5.3$, $p = .02$). This model explained $R^2 = 8\%$ of the variance in scores on the vocabulary test. The table below gives the steps of my model search and the change in deviance and AIC between models (where increase in deviance is desirable but decrease in AIC indicates better fit):

<i>Model</i>	<i>Terms</i>	<i>Deviance</i>	Δ <i>Deviance</i>	<i>AIC</i>
Model1	Gender*Music*Pictures	128.60		244.29
Model2	-Gender:Music:Pictures	129.97	1.37	242.96
Model3	-Gender:Pictures	130.24	0.27	241.10
Model4	-Music:Pictures	131.71	1.47	239.81
Model5	-Gender:Music	136.27	4.56	239.99
Model6	-Music	136.63	0.36	238.16
Model7	-Pictures	139.14	2.51	237.33

In checking model assumptions, this model showed heteroscedasticity and non-normal distribution of errors.

11.4 Performing Comparisons in a Factorial ANOVA

This section will explain what you should do in a factorial ANOVA if any of your variables have more than two levels. The Obarow (2004) data that was examined in the online document "Factorial ANOVA. Factorial ANOVA test" had only two levels per variable, and there was thus no need to do any more analysis, because mean scores showed which group performed better. I will be using a data set adapted from the R data set `ChickWeight`. The

data provides the complex analysis that I want to show here, but I have renamed it in order to better help readers to understand the types of analyses that we would do in second language acquisition. I call this data set *Writing*, and we will pretend that this data describes an experiment which investigated the role of L1 background and experimental condition on scores on a writing sample (if you want to follow along with me, import *Writing.txt* file into R Commander, choosing “white space” as the field separator, and naming it *Writing*).

The dependent variable in this data set is the score on the writing assessment, which ranges from 35 to 373 (we might pretend this is an aggregated score from four separate judges who each rated the writing samples on a 100-point score). The independent variables are first language (four L1s: Arabic, Japanese, Russian, and Spanish) and condition. There were three conditions that students were asked to write their essays in: “correctAll,” which means they were told their teachers would correct all of their errors; “correctTarget,” which means the writers were told only specific targeted errors would be corrected; and “noCorrect,” in which nothing about correction was mentioned to the students.

First, a barplot will help in visually getting a feel for the multivariate data (Figure 11.10).

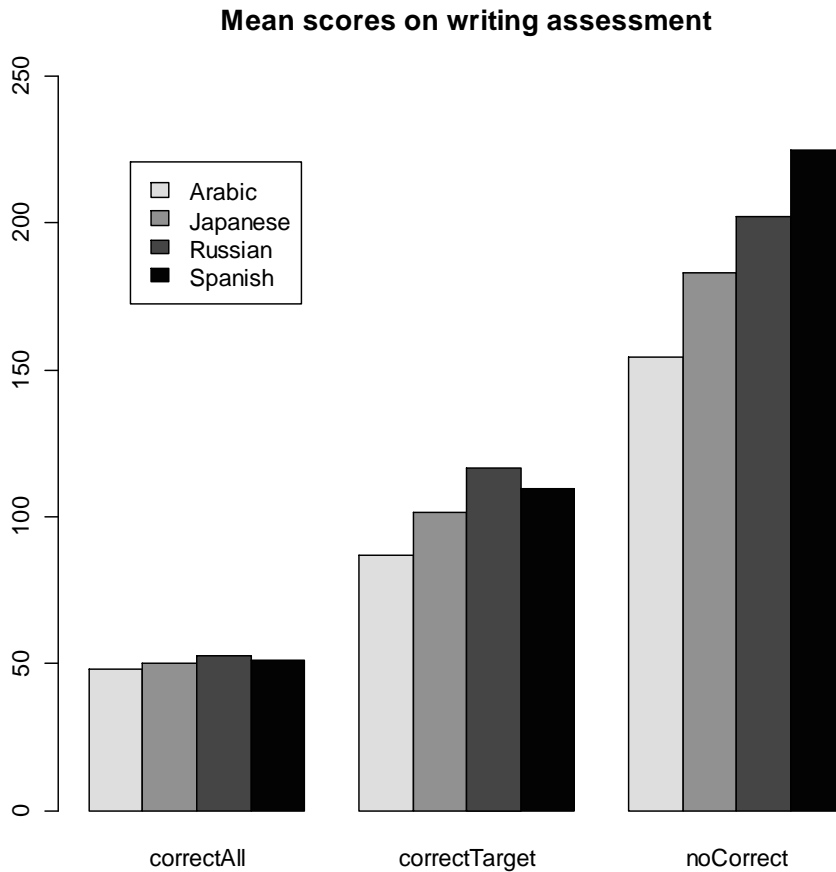


Figure 11.10 Barplot of Writing data set (adapted from ChickWeight in core R).

The barplot in Figure 11.10 shows that those who were not told anything about correction got the highest overall scores. Those who were told everything would be corrected got the most depressed scores. Within the condition of “correctAll,” there doesn’t seem to be much difference depending on L1, but within the “noCorrect” condition the L1 Spanish writers seemed to have scored much higher. To get the mean values, we use `tapply()`:

```
attach(Writing)
tapply(score,list(L1,condition),mean)

      correctAll correctTarget noCorrect
Arabic    48.23729    87.01316  154.3882
Japanese  49.96667   101.72500  182.9200
Russian   52.43333   116.72500  202.4792
Spanish   51.13333   109.45000  224.8400
```

Now we will begin to model. We can use either `aov()` or `lm()` commands; it makes no difference in the end. Using the summary call for `aov()` will result in an ANOVA table, while summary for `lm()` will result in parameter estimates and standard errors (what I call regression output). However, you can easily switch to the other form by using either `summary.lm()` or `summary.aov()`. In order to proceed with the method of level comparison that Crawley (2007) recommends, which is model simplification, we will be focusing on the regression output, so I will model with `lm()`.

An initial full factor ANOVA is called with the following call:

```
write=lm(score~L1*condition)
Anova(write)
```

```
Anova Table (Type II tests)
```

```
Response: score
      Sum Sq Df F value    Pr(>F)
L1      131252  3  27.1899 2.220e-16 ***
condition 1777365  2 552.2920 < 2.2e-16 ***
L1:condition  70588  6   7.3114 1.533e-07 ***
Residuals  910740 566
```

The ANOVA shows that the main effect of first language, the main effect of condition, and the interaction of L1 and condition are statistical. Remember, however, that when interactions are statistical we are usually more interested in what is happening in the interaction and the main effects are not as important. Thus, we now know that scores are affected by the combination of both first language and condition, but we still need to know in more detail how L1 and condition affect scores. This is when we will need to perform comparisons.

We'll start by looking at a summary of the model (notice that this is the regression output summary, because I modeled with `lm()` instead of `aov()`).

```
summary(write)
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)      48.237      5.222   9.237 < 2e-16 ***
L1[T.Japanese]    1.729      8.995   0.192 0.847606
L1[T.Russian]     4.196      8.995   0.466 0.641044
L1[T.Spanish]     2.896      8.995   0.322 0.747598
condition[T.correctTarget] 38.776      6.960   5.571 3.92e-08 ***
condition[T.noCorrect] 106.151      6.797  15.617 < 2e-16 ***
L1[T.Japanese]:condition[T.correctTarget] 12.982     11.929   1.088 0.276932
L1[T.Russian]:condition[T.correctTarget] 25.516     11.929   2.139 0.032869 *
L1[T.Spanish]:condition[T.correctTarget] 19.541     11.929   1.638 0.101966
L1[T.Japanese]:condition[T.noCorrect] 26.802     11.490   2.333 0.020015 *
L1[T.Russian]:condition[T.noCorrect] 43.895     11.548   3.801 0.000160 ***
L1[T.Spanish]:condition[T.noCorrect] 67.556     11.490   5.880 7.04e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 40.11 on 566 degrees of freedom
Multiple R-squared:  0.6875,    Adjusted R-squared:  0.6814
F-statistic: 113.2 on 11 and 566 DF,  p-value: < 2.2e-16
```

This output is considerably more complex than the ANOVA tables we've seen before, so let's go through what this output means. In the parameter estimate (regression) version of the output, the factor levels are compared to the first level of the factor, which will be whatever comes first alphabetically (unless you have specifically ordered the levels). To see what the first level is, examine it with the `levels` command:

```
levels(L1)
[1] "Arabic" "Japanese" "Russian" "Spanish"
```

Thus, the row labeled `L1[T.Japanese]` compares the Arabic group to the Japanese group because it compares the first level of `L1` (Arabic) to the level indicated. The first level of condition is `allCorrect`. The Intercept estimate shows that the overall mean score on the writing sample for Arabic speakers in the `allCorrect` condition is 48.2 points, and Japanese learners in the `allCorrect` condition score, on average, 1.7 points above the Arabic speakers (that is what the Estimate for `L1[T.Japanese]` shows). The first three rows after the intercept show, then, that no group is statistically different from the Arabic speakers in the `allCorrect` condition (which we can verify visually in Figure 11.10). The next two rows, with "condition" at the beginning, are comparing the "correctTarget" and "noCorrect" condition each with the "allCorrect" condition, and both of these comparisons are statistical. The last six rows of the output compare various `L1` groups in either the "correctTarget" condition or the "noCorrect" condition with the Arabic speakers in the "allCorrect" condition (remember, this is the default level). Thus, the last six lines tell us that there is a difference between the Russian speakers in the "correctTarget" condition and the Arabic speakers in the "allCorrect" condition, and also that all `L1` speakers are statistically different in the "noCorrect" condition from the Arabic speakers in the "allCorrect" condition.

At this point, you may be thinking this is really complicated and you really didn't want comparisons only with Arabic or only in the "allCorrect" condition. What can you do to get the type of post-hoc tests you would find in a commercial statistical program like SPSS which can compare all levels? The `pairwise.t.test()` command can perform all possible pairwise comparisons if that is what you would like.

For example, we can look at comparisons only between the `L1`s:

```
pairwise.t.test(score,L1, p.adjust.method="fdr")
```

	Arabic	Japanese	Russian
Japanese	0.02279	-	-
Russian	0.00013	0.19195	-
Spanish	2.5e-06	0.03519	0.39274

This shows that, overall, Arabic `L1` speakers are different from all the other groups; the Japanese `L1` speakers are different from the Spanish `L1` speakers, but not the Russian `L1` speakers. Lastly, the Russian speakers are not different from the Spanish speakers. We can also look at the comparisons between conditions:

```
pairwise.t.test(score,condition, p.adjust.method="fdr")
```

	correctAll	correctTarget
correctTarget	<2e-16	-
noCorrect	<2e-16	<2e-16

Here we see very low p -values for all conditions, meaning all conditions are different from each other. Lastly, we can look at comparisons on the interaction:

```
pairwise.t.test(score,L1:condition, p.adjust.method="fdr")
```

	Arabic:correctAll	Arabic:correctTarget	Arabic:noCorrect
Arabic:correctTarget	6.0e-08	-	-
Arabic:noCorrect	< 2e-16	< 2e-16	-
Japanese:correctAll	0.87409	2.9e-05	< 2e-16
Japanese:correctTarget	2.9e-10	0.07059	3.8e-11
Japanese:noCorrect	< 2e-16	< 2e-16	9.5e-05
Russian:correctAll	0.69359	9.4e-05	< 2e-16
Russian:correctTarget	1.3e-15	0.00021	1.7e-06
Russian:noCorrect	< 2e-16	< 2e-16	1.3e-10
Spanish:correctAll	0.79583	5.1e-05	< 2e-16
Spanish:correctTarget	7.2e-13	0.00531	1.4e-08
Spanish:noCorrect	< 2e-16	< 2e-16	< 2e-16

This is only the beginning of a very long chart . . .

Going down the first column, it's clear that none of the groups are different from the Arabic L1 speakers in the "correctAll" condition (comparing Arabic to Japanese $p=.87$, Arabic to Russian $p=.69$, Arabic to Spanish $p=.80$). That, of course, doesn't tell us everything; for example, we don't know whether the Japanese speakers are different from the Russian and Spanish speakers in the "correctAll" condition, but given our barplot in Figure 11.10 we might assume they will be when we check. Looking over the entire chart we get a full picture of the nature of the interaction: no groups are different from each other in the "correctAll" condition; in the "correctTarget" condition, only the Russian and Arabic and the Spanish and Arabic speakers differ statistically from each other; in the "noCorrect" condition, all groups perform differently from one another.

Here is the analysis of the post-hoc command in R:

<code>pairwise.t.test(score,L1:condition, p.adjust.method="fdr")</code>	
<code>pairwise.t.test(score, . . .</code>	Put in the dependent (response) variable as the first argument.
<code>L1:condition</code>	The second argument is the variable containing the levels you want compared; here, I didn't care about the main effects of L1 or condition, just the interaction between them, so I put the interaction as the second argument.
<code>p.adjust.method="fdr"</code>	Specifies the method of adjusting p -values for multiple tests; you can choose <code>hochberg</code> , <code>hommel</code> , <code>bonferroni</code> , <code>BH</code> , <code>BY</code> , and <code>none</code> (equal to <code>LSD</code>); I chose <code>fdr</code> because it gives more power while still controlling for Type I error.

We've been able to answer our question about how the levels of the groups differ from one another, but we have had to perform a large number of post-hoc tests. If we didn't really care about the answers to all the questions, we have wasted some power by performing so many. One way to avoid such a large number of post-hoc tests would be to have planned out your question in advance and to run a priori planned comparisons. I do not know of any way to do this in one step for that crucial interaction post-hoc, but one approach would be to run the interaction post-hoc with no adjustment of p -values, and then pick out just the post-hoc tests

you are interested in and put the p -values into the FDR calculation (of course, since all of these post-hocs had low p -values to start with, doing this calculation for this example will not change anything, but in other cases it could certainly make a difference!). For example, let's say you only cared about which groups were different from each other in "noCorrect" condition. You can run the post-hoc with no adjustment of p -values:

```
pairwise.t.test(score,L1:condition, p.adjust.method="none")
```

Picking out the p -values for only the "noCorrect" condition, they are:

	<i>Arabic</i>	<i>Japanese</i>	<i>Russian</i>
Japanese	7.4e-05		
Russian	7.4e-11	.016	
Spanish	<2e-16	2.4e-07	.00599

Now put these in the FDR calculation (also found in Appendix B):

```
pvalue<-c(7.4e-05,7.4e-11,2e-16,.016,2.4e-07,.00599)
sorted.pvalue<-sort(pvalue)
j.alpha<-(1:6)*(.05/6)
dif<-sorted.pvalue-j.alpha
neg.dif<-dif[dif<0]
pos.dif<-neg.dif[length(neg.dif)]
index<-dif==pos.dif
p.cutoff<-sorted.pvalue[index]
p.cutoff
[1] 0.016
```

The `p.cutoff` value is the value under which the p -values will be statistical. Obviously, all of the values in this case are statistical.

The final conclusion we can make is that in the "noCorrect" condition L1 was very important. Every L1 group scored differently in this condition, with the mean scores showing that, from the best performers to the worst, we have Spanish L1 > Russian L1 > Japanese L1 > Arabic L1. In the "correctTarget" condition the Russian L1 and Spanish L1 speakers performed better than the Arabic L1 speakers, and the Japanese were somewhere in the middle, neither statistically different from the Arabic speakers nor statistically different from the Russian and Spanish speakers. For the "correctAll" condition L1 made absolutely no difference. Everyone performed very poorly when they thought all of their mistakes were going to be graded. This is the story that I would tell if reporting on this study.

Tip: Sometimes logic doesn't work with multiple comparisons! For example, you may at times find that statistically Group A is different from Group B, and Group B is different from Group C, but statistically Group A is not different from Group C. As Howell (2002) points out, logically this seems impossible. If $A \neq B$ and $B \neq C$, then logically $A \neq C$. However, logic is not statistics and sometimes there is not enough power to find a difference which logically should be there. Howell says sometimes we just have to live with uncertainty!

Performing Comparisons after a Factorial ANOVA with R

1. If the comparison you want to make is with just the one factor level that R uses as a default, the output of the `summary.lm()` (if you have modeled with `aov()`) or `summary()` (if you have modeled with `lm()`) can provide enough information to make comparisons.

2. You can target only certain main effects or certain interactions by using the syntax:

```
pairwise.t.test(Writing$score, Writing$L1, p.adjust.method="fdr") #for main effect
pairwise.t.test(Writing$score:Writing$L1, p.adjust.method="fdr") #for interaction
where the first argument is the response variable and the second is the explanatory variable.
```

3. To simulate planned comparisons, use the `pairwise.t.test()` with no adjustment for *p*-values (`p.adjust.method="none"`); then pick out the comparisons you want and use the FDR method (Appendix B) to adjust the *p*-values and calculate a cut-off value to substitute for $\alpha=.05$.

- Formatted: Font: Italic
- Formatted: Font: Italic
- Formatted: Font: Italic
- Deleted: ()
- Deleted: ()
- Deleted: ()
- Deleted: ()
- Deleted: ()
- Deleted: ()
- Deleted: ="
- Deleted: "
- Deleted: "
- Deleted: "
- Deleted: ()
- Deleted: ()
- Formatted: Font: Italic
- Deleted: "
- Deleted: "
- Deleted: ,
- Formatted: Font: Italic

11.5 Application Activities with Factorial ANOVA

1. Obarow (2004) data. In the online document “Factorial ANOVA.Factorial ANOVA test” I performed a factorial ANOVA on the gain score for Obarow’s Treatment 1. Import the SPSS file Obarow.Story2.sav and call it `obarrow2`. You will need to prepare this data for analysis as outlined in the online document “Factorial ANOVA.Putting data in correct format for factorial ANOVA.” You will need to decide whether to select cases. You will need to change the `trtmnt2` variable into two different IVs, music and pictures (use the values for `trtmnt2` to help guide you). Then calculate a gain score. Once you have configured the data to a form ready to perform an ANOVA, visually and numerically examine the data. Last, perform a factorial ANOVA to find out what effect gender, music, and pictures had on the gain score for the second treatment/story.

2. Larson-Hall and Connell (2005). Import the `LarsonHall.Forgotten.sav` data set as `forget`. Use the `SentenceAccent` variable to investigate whether the independent variables of gender (`sex`) and (`immersion`) status could help explain variance in what kind of accent rating participants received. You will need to use post-hoc tests with this data, because there are three levels for the status variable. Be sure to comment on assumptions.

3. Import the `Eysenck.Howell13.sav` data set as `eysenck`. This data comes from a study by Eysenck (1974), but the data are given in Howell (2002), Chapter 13. Eysenck wanted to see whether age group (`young=18–30` years, `old=55–60` years) would interact with task condition in how well the participants could recall words from a word list. There were five conditions, four of which ascertained whether the participants would learn the words incidentally, but some tasks required more semantic processing than others. The tasks were letter counting (`Counting`), rhyming the words in the list (`Rhyming`), finding an adjective to modify the noun in the list (`Adjective`), forming an image of the word (`Imagery`), and control (`Intention`; intentionally trying to learn the word). You will first need to rearrange the data from the wide form to the long form. Perform a 2 (`AgeGroup`) × 5 (`Condition`) factorial ANOVA on the data.

11.6 Performing a Robust ANOVA

In this section I will provide an explanation of some of Wilcox’s (2005) robust methods for two-way and three-way ANOVAs. Before using these commands you will need to load

Wilcox's library, *WRS*, into R (see the online document "Using Wilcox's R library" to get started with this). The command `t2way()` uses trimmed means for a two-way ANOVA, the command `t3way()` uses trimmed means for a three-way ANOVA, and the command `pbad2way()` uses M-measures of location (not means) in a two-way design. For all of these tests, the data need to be stored in a matrix or list. Wilcox (2005, p. 280) gives the table found in Table 11.2, which shows which column contains which data (where `x[[1]]` means the data in column 1 of a matrix or the first vector of a list).

Table 11.2 Data Arranged for a Two-Way ANOVA Command from Wilcox

	Factor B			
Factor A	<code>x[[1]]</code>	<code>x[[2]]</code>	<code>x[[3]]</code>	<code>x[[4]]</code>
	<code>x[[5]]</code>	<code>x[[6]]</code>	<code>x[[7]]</code>	<code>x[[8]]</code>

Thus, the data for level 1 of both Factors A and B is contained in the first vector of the list (`x[[1]]`), the data for level 1 of Factor A and level 2 of Factor B is contained in the second vector of the list (`x[[2]]`), and so on.

If you have your data already in the wide format, there is no need to put it into a list. Instead, you can tell the command what order to take the columns in by specifying for `grp`, like this:

```
grp<-c(2,3,5,8,4,1,6,7)
```

where column 2 (as denoted by `c(2)`) is the data for level 1 of Factors A and B, column 3 is the data for level 1 of Factor A and level 2 of Factor B, etc.

For Obarow's data let's look at a two-way ANOVA with music and pictures only. There are two levels to each of these factors. I will call "No music No pictures" level 1 of Factor Music and Factor Pictures, "No music Yes pictures" level 1 of Music and level 2 of Pictures, "Yes music No Pictures" level 2 of Music and level 1 of Pictures, and "Yes music Yes pictures" level 2 of both factors. I will create a list from the `Obarow.Story1.sav` file, which I previously imported as `ObarowStory1` and subset the variables for the immediate gain score into separate vectors in the list.

```
levels(obarowStory1$Treatment1) #find out exactly what names to use to subset
[1] "No Music No Pictures" "No Music Yes Pictures" "Yes Music No Pictures"
[4] "Yes Music Yes Pictures"
O2way=list()
O2way[[1]]=subset(obarowStory1, subset=Treatment1=="No Music No Pictures",
select=c(gnsc1.1))
O2way[[2]]=subset(obarowStory1, subset=Treatment1=="No Music Yes Pictures",
select=c(gnsc1.1))
O2way[[3]]=subset(obarowStory1, subset=Treatment1=="Yes Music No Pictures",
select=c(gnsc1.1))
O2way[[4]]=subset(obarowStory1, subset=Treatment1=="Yes Music Yes Pictures",
select=c(gnsc1.1))
```

Now I can run the `t2way()` test:

```
library(WRS)
t2way(2,2,O2way,tr=.2,grp=c(1,2,3,4)) #the first two numbers indicate the number of
levels in the first variable and the second variable, respectively
```

```

$Qa
[1] 1.355270

$A.p.value
[1] 0.256

$Qb
[1] 0.4108072

$B.p.value
[1] 0.528

$Qab
[1] 0.004277459

$AB.p.value
[1] 0.949

$means
      [,1]      [,2]
[1,] 1.3636364 1.0909091
[2,] 0.8888889 0.6666667

```

The output from this test returns the test statistic Q for Factor A (which is **Music** here) in the $\$Qa$ line, and the p -value for this test in the $\$A.p.value$ line. So the factor of **Music** is not statistical, $p=.26$. Neither is the factor of **Pictures** ($p=.53$), nor the interaction between the two ($p=.95$).

If there had been statistical effects for groups, and the groups had more than two levels, you would want to perform multiple comparisons to find out which levels were statistically different for one another. For this, the command `mcppb20()`, which uses means trimming and the percentile bootstrap, would be the appropriate command. This command was explained in the online document “One way ANOVA. A robust one-way ANOVA test” and could be used with the `O2way` list we just created.

For a two-way analysis using percentile bootstrapping and M-estimators, the command `pbad2way()` can be used. The data need to be arranged in exactly the same way as for the previous command, so we are ready to try this command out:

```
pbad2way(2,2,O2way,est=mom, conall=T, grp=c(1,2,3,4))
```

The term `conall=T` means that all possible pairs are tested, while using the term `conall=F` would provide an alternative strategy that might affect power (see Wilcox, 2005, p. 318 for more details). This term can be used with the modified one-step M-estimator (MOM), as shown here, or the median (`est=median`). The output will simply return significance levels for the two variables and the interaction.

Obviously a three-way interaction will be more complicated. The command `t3way()` also requires data to be set up in a specific way. The easiest way to do this again will be to create a list. Now we need to arrange the data so that, for level 1 of Factor A, we will find the first vector of the list contains level 1 for Factor B and level 1 for Factor C. The second vector will contain level 1 for Factor A, level 1 for Factor B, and level 2 for Factor C. See Table 11.3

(taken from Wilcox, 2005, p. 286) for more details on how a design of 2×2×4 would be set up.

Table 11.3 Data Arranged for a Three-Way ANOVA Command from Wilcox

Level 1 of Factor A	Factor C			
Factor B	x[[1]]	x[[2]]	x[[3]]	x[[4]]
	x[[5]]	x[[6]]	x[[7]]	x[[8]]

Level 2 of Factor A	Factor C			
Factor B	x[[9]]	x[[10]]	x[[11]]	x[[12]]
	x[[13]]	x[[14]]	x[[15]]	x[[16]]

For the Obarow data then, using a 2×2×2 design with Gender included as Factor C, we would have the data arranged as indicated in Table 11.4.

Table 11.4 Obarow Data Arranged for a Three-Way ANOVA

Music: No		Gender:	
		Male	Female
Pictures:	No	x[[1]]	x[[2]]
	Yes	x[[3]]	x[[4]]

Music: Yes		Gender	
		Male	Female
Pictures:	No	x[[5]]	x[[6]]
	Yes	x[[7]]	x[[8]]

I don't know how to subset with two variables at a time, so I will first create two data sets, one with males and one with females, before I subset once more for the treatment categories:

```

obarrowStory1Male <- subset(obarrowStory1, subset=gender=="male")
obarrowStory1Female <- subset(obarrowStory1, subset=gender=="female")
O3way=list()
O3way[[1]]=subset(obarrowStory1Male, subset=Treatment1=="No Music No Pictures", select=c(gnsc1.1))
O3way[[2]]=subset(obarrowStory1Female, subset=Treatment1=="No Music No Pictures", select=c(gnsc1.1))
O3way[[3]]=subset(obarrowStory1Male, subset=Treatment1=="No Music Yes Pictures", select=c(gnsc1.1))
O3way[[4]]=subset(obarrowStory1Female, subset=Treatment1=="No Music Yes Pictures", select=c(gnsc1.1))
O3way[[5]]=subset(obarrowStory1Male, subset=Treatment1=="Yes Music No Pictures", select=c(gnsc1.1))
O3way[[6]]=subset(obarrowStory1Female, subset=Treatment1=="Yes Music No Pictures", select=c(gnsc1.1))
O3way[[7]]=subset(obarrowStory1Male, subset=Treatment1=="Yes Music Yes Pictures", select=c(gnsc1.1))
O3way[[8]]=subset(obarrowStory1Female, subset=Treatment1=="Yes Music Yes Pictures", select=c(gnsc1.1))
    
```

Now I am ready to perform the three-way analysis using trimmed means:

```
t3way(2,2,2, O3way, tr=.2)
Error in `[.data.frame`(x, order(x, na.last = na.last, decreasing = decreasing)) :
undefined columns selected
```

For some reason I get an error message here. I wondered if it was because the length of the lists is different, but the two-way command seemed to work just fine with different-sized lists. So I went back and typed in the data (looking at what I had obtained from my work above):

```
O3way[[1]]=c(3,3,1,3,0,1,3,1,0,1,2,1)
O3way[[2]]=c(2,1,1,1,1)
O3way[[3]]=c(6,-1,0,3,2,3,1,0,2)
O3way[[4]]=c(1,0,0,1,2,-2,1,2)
O3way[[5]]=c(4,1,4,2,0)
O3way[[6]]=c(2,2,1,1,-1,-1,0,-2,-1,2)
O3way[[7]]=c(1,5,-1,0,2,0,1,-2)
O3way[[8]]=c(1,2,1,0,1,-1,1)
t3way(2,2,2,O3way)
```

Now the test worked fine . . . ! It started with the values for the main variables, with the main effect for Factor C (gender) last:

```
$Qc
[1] 2.960803

$Qc.crit
[1] 2.947086

$C.p.value
[1] 0.12

$Qab
[1] 0.4147462

$Qab.crit
[1] 0.4128055

$AB.p.value
[1] 0.538

$Qac
[1] 0.03581812

$Qac.crit
[1] 0.03560799
```

Since it is a three-way ANOVA, it also returns the results of all of the two-way interactions, such as **Music:Treatment** (ab) and **Music:Gender** (ac), as well as the three-way interaction. The test statistic *Q* is returned along with a critical value for *Q* and the *p*-value. You can see in the output that the interaction between music and gender is statistical here (a different result from the parametric test using untrimmed means). Wilcox (2005) does not provide any

methods for three-way analyses using bootstrapping. I'm afraid I don't see that these robust tests are as useful, though, as previous ones, since there seems to be no way of searching for the minimal adequate model.

Performing Robust ANOVAs Using Wilcox's (2005) Commands

1. For a two-way ANOVA, rearrange data in list form so that data in the first vector of the list, `x[[1]]`, contains the information for level 1 of Factor A and level 1 of Factor B, `x[[2]]` contains the information for level 1 of Factor A and level 2 of Factor B, and so on. For a three-way ANOVA, separate first by levels of Factor A, then `x[[1]]` contains information for level 1 of Factor B and level 1 of Factor C, `x[[2]]` contains information for level 1 of Factor B and level 2 of Factor C, and so on (this is fairly confusing, but examples are given in [the](#) text).

2. Use one of Wilcox's (2005) commands after opening the WRS library:

```
t2way(J, K, O3way, tr=.2) #trimmed means; J=# of levels for Factor A, K=# of levels for Factor B
```

```
t3way(J, K, L, O3way, tr=.2) #trimmed means; J=# of levels for Factor A, K=# of levels for Factor B, L=# of levels for Factor C
```

```
pbad2way(J, K, O3way, est=mom, conall=T) #uses bootstrapping and MOM estimator (or median also possible); J=# of levels for Factor A, K=# of levels for Factor B
```

Formatted: Font: Italic

Deleted: .