

Factorial ANOVA

Self-test answers



- The file **GogglesRegression.dat** contains the dummy variables used in this example. Use this file and run a multiple regression on the data.

First load the data (set your working directory first):

```
gogglesReg<-read.delim("GogglesRegression.dat", header = TRUE)
```

The data look like this:

	gender	alcohol	interaction	attractiveness
1	1	0	0	65
2	1	0	0	70
3	1	0	0	60
4	1	0	0	60
5	1	0	0	60
6	1	0	0	55
7	1	0	0	60
8	1	0	0	55
9	1	1	1	55
10	1	1	1	65
11	1	1	1	70
12	1	1	1	55
13	1	1	1	55
14	1	1	1	60
15	1	1	1	50
16	1	1	1	50
17	0	0	0	50
18	0	0	0	55
19	0	0	0	80
20	0	0	0	65
21	0	0	0	70
22	0	0	0	75
23	0	0	0	75
24	0	0	0	65
25	0	1	0	30
26	0	1	0	30
27	0	1	0	30
28	0	1	0	55
29	0	1	0	35
30	0	1	0	20
31	0	1	0	45
32	0	1	0	40

Note how the predictors **gender** and **alcohol** have been dummy-coded, and that the variable **interaction** is these two variables multiplied together. To create the model (with **attractiveness** as the outcome and the other three variables as predictors) and see the output we can execute:

```
gogglesRegModel<-lm(attractiveness ~ gender + alcohol + interaction, data =
gogglesReg)

summary(gogglesRegModel)

summary.lm(gogglesRegModel)
```



- Use *ggplot2* to plot a line graph (with error bars) of the attractiveness of the date with alcohol consumption on the x-axis and different-coloured lines to represent males and females.

To do a multiple line chart we need to execute these commands:

```
line <- ggplot(gogglesData, aes(alcohol, attractiveness, colour = gender))

line + stat_summary(fun.y = mean, geom = "point") + stat_summary(fun.y = mean, geom =
"line", aes(group= gender)) + stat_summary(fun.data = mean_cl_boot, geom = "errorbar",
width = 0.2) + labs(x = "Alcohol Consumption", y = "Mean Attractiveness of Date (%)",
colour = "Gender")
```

The resulting graph can be found in the book chapter.



- Use *ggplot2* to plot boxplots of the attractiveness of the date at each level of alcohol consumption on the x-axis and different panels to represent males and females.

```
boxplot <- ggplot(gogglesData, aes(alcohol, attractiveness))
boxplot + geom_boxplot() + facet_wrap(~gender) + labs(x = "Alcohol Consumption", y =
"Mean Attractiveness of Date (%)")
```

The resulting graph can be found in the book chapter.



- Plot error bar graphs of the main effects of alcohol and gender.

To plot these error bar for the main effect of **gender** execute:

```
bar <- ggplot(gogglesData, aes(gender, attractiveness))
bar + stat_summary(fun.y = mean, geom = "bar", fill = "White", colour = "Black") +
stat_summary(fun.data = mean_cl_normal, geom = "pointrange") + labs(x = "Gender", y =
"Mean Attractiveness of Date (%)") + scale_y_continuous(breaks=seq(0,80, by = 10))
```

To plot these error bar for the main effect of **alcohol** execute:

```
bar <- ggplot(gogglesData, aes(alcohol, attractiveness))
bar + stat_summary(fun.y = mean, geom = "bar", fill = "White", colour = "Black") +
stat_summary(fun.data = mean_cl_normal, geom = "pointrange") + labs(x = "Alcohol
Consumption", y = "Mean Attractiveness of Date (%)") +
scale_y_continuous(breaks=seq(0, 80, by = 10))
```

Note that I've used *scale_y_continuous()* to override the defaults for the y-axis. Specifically, I have used the *breaks* option to specify the numbering along this axis: *breaks=seq(0, 80, by = 10)* uses the *seq()* function to create a sequence of numbers from 0 to 80 in steps of 10. Therefore, we get axis labels at 0, 10, 20, 30, 40, 50, 60, 70, 80 (the defaults were 0, 20, 40, 60, 80).

Oliver Twisted

Please Sir, can I have some more ... contrasts?



Another example of using contrasts

Imagine a clinical psychologist wanted to see the effects of a new antidepressant drug called Cheerup. He took 50 people suffering from clinical depression and randomly assigned them to one of five groups. The first group was a waiting list control group (i.e. they were people assigned to the waiting list who were not treated during the study), the second took a placebo tablet (i.e. they were told they were being given an antidepressant drug but actually the pills contained sugar and no active agents), the third group took a well-established selective serotonin reuptake inhibitor (SSRI) antidepressant called Seroxat (Paxil to American readers), the fourth group was given a well-established serotonin–norepinephrine reuptake inhibitor (SNRI) antidepressant called Effexor,¹ and the final group was given the new drug, Cheerup. Levels of depression were measured before and after two months on the various treatments, and ranged from 0 = as happy as a spring lamb to 20 = pass me the noose. The data are in the file **Depression.csv**.

Load the data into a dataframe called *depressionData* by executing:

¹ SSRIs work selectively to inhibit the reuptake of the neurotransmitter serotonin in the brain, whereas SNRIs, which are newer act not only on serotonin but on another neurotransmitter (from the same family), norepinephrine.

```
depressionData<-read.csv("Depression.csv", header = TRUE)
```

The variable **treat** contains strings representing the groups (e.g., Placebo to represent the placebo group). R will import this variable as a factor, but when it does it will order the levels of the factor alphabetically. Therefore, the order of factor levels for **treat** will be: Cheerup, Effexor, No Treatment, Placebo, Seroxat (Paxil). This is important to remember when entering the contrast codes.

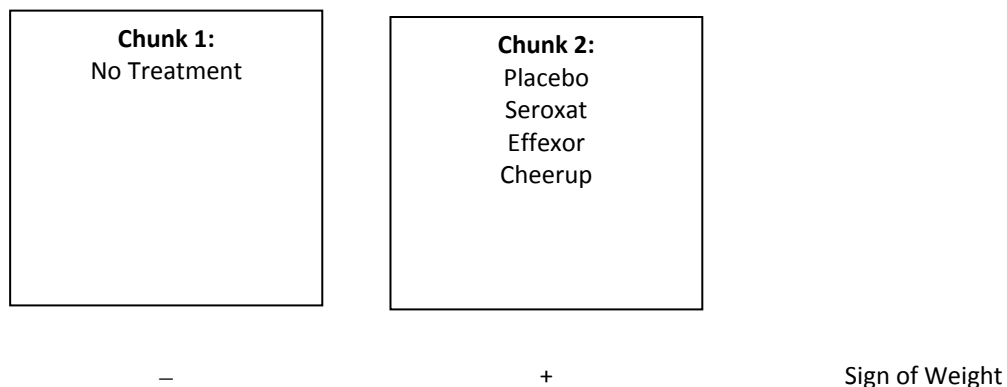
The design of this study is a two-way mixed design. There are two independent variables: treatment (no treatment, placebo, Seroxat, Effexor or Cheerup) and time (before or after treatment). Treatment is measured with different participants (and so is between-group) and time is, obviously, measured using the same participants (and so is repeated-measures). Hence, the ANOVA we want to use is a 5×2 two-way ANOVA. However, we also have a change score (**diff**) and because we haven't got to mixed ANOVA yet in the book we're just going to look at this change score, so the design reduces down to a one-way between group design: the treatment condition (**treat**) and the change in depression scores (**diff**).

Now, we want to do some contrasts. Imagine we have the following hypotheses:

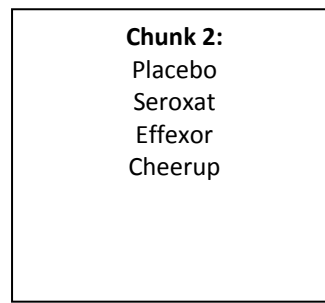
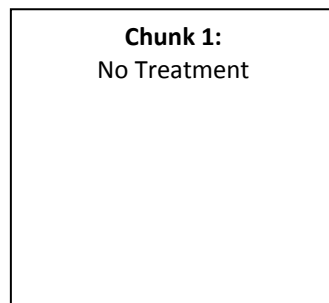
1. Any treatment will be better than no treatment.
2. Drug treatments will be better than the placebo.
3. Our new drug, Cheerup, will be better than old-style antidepressants.
4. The old-style antidepressants will not differ in their effectiveness.

We have to code these various hypotheses as we did in Chapter 10.

The first contrast involves comparing the no-treatment condition to all other groups. Therefore, the first step is to chunk these variables, and then assign a positive weight to one chunk and a negative weight to the other chunk.



Having done that, we need to assign a numeric value to the groups in each chunk. As I mentioned in Chapter 8, the easiest way to do this is just to assign a value equal to the number of groups in the *opposite* chunk. Therefore, the value for any group in chunk 1 will be the same as the number of groups in chunk 2 (in this case 4). Likewise, the value for any groups in chunk 2 will be the same as the number of groups in chunk 1 (in this case 1). So, we get the following codes:



–

+

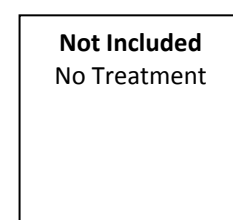
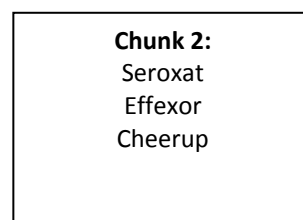
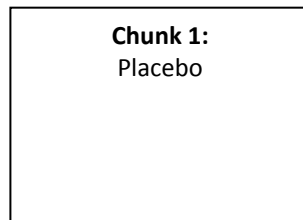
Sign of Weight

–4

+1

Value of Weight

The second contrast requires us to compare the placebo group to all of the drug groups. Again, we chunk our groups accordingly, assign one chunk a negative sign and the other a positive, and then assign a weight on the basis of the number of groups in the opposite chunk. We must also remember to give the no-treatment group a weight of 0 because they're not involved in the contrast.



–

+

Sign of Weight

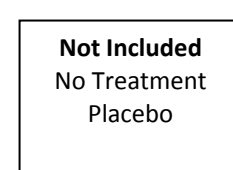
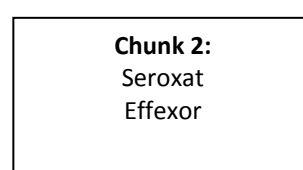
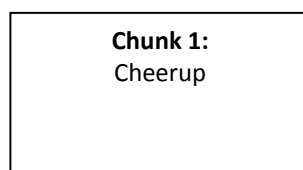
–3

+1

Value of Weight

0

The third contrast requires us to compare the new drug (Cheerup) to the old drugs (Seroxat and Effexor). Again, we chunk our groups accordingly, assign one chunk a negative sign and the other a positive, and then assign a weight on the basis of the number of groups in the opposite chunk. We must also remember to give the no-treatment and placebo groups a weight of 0 because they're not involved in the contrast.



–

+

Sign of Weight

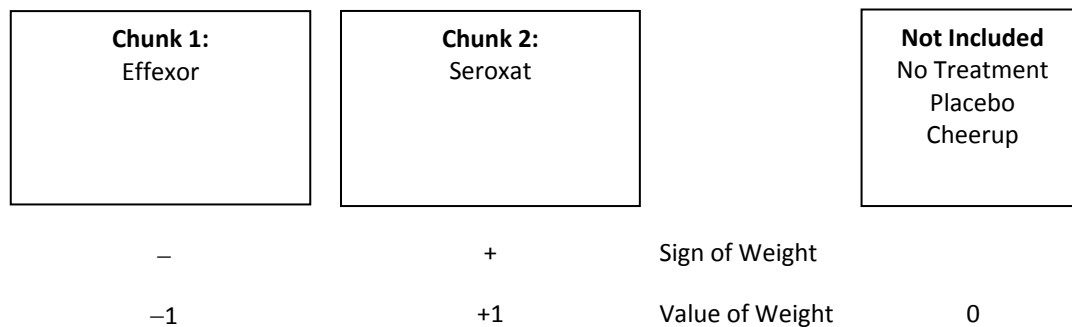
–2

+1

Value of Weight

0

The final contrast requires us to compare the two old drugs. Again, we chunk our groups accordingly, assign one chunk a negative sign and the other a positive, and then assign a weight on the basis of the number of groups in the opposite chunk. We must also give the no-treatment, placebo and Cheerup groups a weight of 0 because they're not involved in the contrast.



We can summarize these codes in the following table:

	No Treatment	Placebo	Seroxat	Effexor	Cheerup
Contrast 1	–4	1	1	1	1
Contrast 2	0	–3	1	1	1
Contrast 3	0	0	1	1	–2
Contrast 4	0	0	1	–1	0

Remember however, that the levels of treat are actually ordered alphabetically (because of how **R** imported the data), so, we need to enter the codes in this order:

	Cheerup	Effexor	No Treatment	Placebo	Seroxat
Contrast 1	1	1	–4	1	1
Contrast 2	1	1	0	–3	1
Contrast 3	–2	1	0	0	1
Contrast 4	0	–1	0	0	1

These are the codes that we need to enter into **R** to do the contrasts that we'd like to do.

To enter these contrasts we could create a variable for each contrast that contains the codes:

```
contrast1<-c(1, 1, -4, 1, 1)
contrast2<-c(1, 1, 0, -3, 1)
contrast3<-c(-2, 1, 0, 0, 1)
contrast4<-c(0, -1, 0, 0, 1)
```

We can then apply the *contrasts()* function to the variable **treat** and use *cbind()* to concatenate these variables in different columns:

```
contrasts(depressionData$treat)<-cbind(contrast1, contrast2, contrast3, contrast4)
```

If we look at **treat** we can see that the appropriate contrast codes have been attached to the variable:

```
attr(,"contrasts")
      contrast1 contrast2 contrast3 contrast4
Cheerup         1         1        -2         0
Effexor          1         1         1        -1
No Treatment    -4         0         0         0
Placebo          1        -3         0         0
Seroxat (Paxil)  1         1         1         1
Levels: Cheerup Effexor No Treatment Placebo Seroxat (Paxil)
```

We can then run the ANOVA in the usual way by executing:

```
depressionModel<-aov(diff~treat, data = depressionData)
```

Output from the contrasts

First we can look at the overall effect of treatment by executing:

```
summary(depressionModel)
```

The resulting output shows that treatment had a significant effect on the change in depression scores $F(4, 45) = 4.42, p < .01$:

```

      Df Sum Sq Mean Sq F value    Pr(>F)
treat    4   251.8   62.950    4.4213 0.004238 **
Residuals 45   640.7   14.238
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

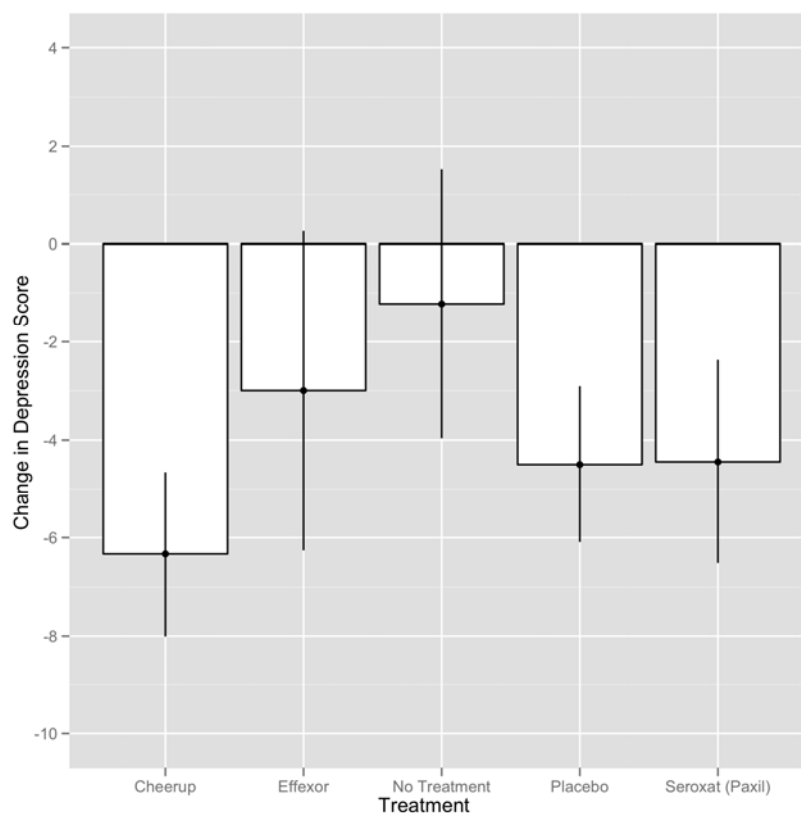
We can plot a graph of this main effect by executing:

```

bar <- ggplot(depressionData, aes(treat, diff))
bar + stat_summary(fun.y = mean, geom = "bar", fill = "White", colour = "Black") +
  stat_summary(fun.data = mean_cl_normal, geom = "pointrange") + labs(x = "Treatment", y =
    "Change in Depression Score") + scale_y_continuous(breaks=seq(-10,4, by = 2), limits
    = c(-10, 4))

```

Note that I've used the *scale_y_continuous()* to over-ride the defaults for the y-axis. Specifically, I have used the *breaks* option to specify the numbering along this axis: *breaks=seq(-10, 4, by = 2)* uses the *seq()* function to create a sequence of numbers from -10 to 4 in steps of 2. The *limits* option sets the limits of the scale to be -10 and 4. The resulting graph is:



It looks as though Cheerup creates the biggest decrease in depression scores followed by the placebo and seroxat. The least effective treatment is the no-treatment condition (which shows a very small decline in depression levels).

We can see the contrast estimates by executing:

```
summary.lm(depressionModel)
```

```

Coefficients:
(Intercept)      -3.50000    0.53362   -6.559 4.61e-08 ***
treatcontrast1   -0.80000    0.26681   -2.998 0.00441 **
treatcontrast2    0.06667    0.34445    0.194 0.84740
treatcontrast3    1.38333    0.48713    2.840 0.00675 **
treatcontrast4   -0.65000    0.84374   -0.770 0.44510

```

```

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

Residual standard error: 3.773 on 45 degrees of freedom
Multiple R-squared:  0.2821,    Adjusted R-squared:  0.2183 
F-statistic: 4.421 on 4 and 45 DF,  p-value: 0.004238

```

- *treatcontrast1*: This is our first contrast (no-treatment vs. the rest) and, as you can see, this is significant (p is below 0.05). This tells us that the change in depression levels in the no-treatment group was significantly different from the average change in all other groups, $t = -2.998$, $p < .01$. As you can see in the graph, there is little change in depression in the no-treatment group, but in all other groups there is a fall in depression. Therefore, this contrast reflects the fact that there is little change in the no-treatment group, but there is a decrease in depression levels in all other groups.
- *treatcontrast2*: The second contrast (placebo vs. Seroxat, Effexor and Cheerup) is very non-significant, $t = 0.194$, $p = .85$. This shows that the decrease in depression levels seen in the placebo group is comparable to the average decrease in depression levels seen in the Seroxat, Effexor and Cheerup conditions. In other words, the combined effect of the drugs on depression is no better than a placebo.
- *treatcontrast3*: The third contrast (Cheerup vs. Effexor and Seroxat), is highly significant, $t = 2.84$, $p < .01$. This shows that the decrease in depression levels seen in the Cheerup group is significantly bigger than the decrease seen in the Effexor and Seroxat groups combined. Put another way, Cheerup has a significantly bigger effect than other established antidepressants.
- *treatcontrast4*: Our last contrast (Seroxat vs. Effexor) is very non-significant, $t = -0.77$, $p = 0.45$. This tells us that the decrease in depression levels seen in the Seroxat group is comparable to the decrease in depression levels seen in the Effexor group. Put another way, Effexor and Seroxat seem to have similar effects on depression.

Please Sir, can I have some more ... simple effects?



Calculating Simple Effects

A simple main effect (usually called a simple effect) is just the effect of one variable at levels of another variable. Chapter 12 gives an example in which we measured the attractiveness of dates after no alcohol, 2 pints and 4 pints in both men and women. Therefore, we have two independent variables: alcohol (none, 2 pints, 4 pints) and gender (male and female). One simple effects analysis we could do would be to look at the effect of gender (i.e. compare male and female scores) at the three levels of alcohol. Let's look how we'd do this. We're partitioning the model sum of squares, and we saw in Chapter 10 that we calculate model sums of squares using this equation:

$$SS_M = \sum n_k (\bar{x}_k - \bar{x}_{\text{grand}})^2$$

For simple effects, we calculate the model sum of squares for the effect of gender at each level of alcohol. So, we'd begin with when there was no alcohol, and calculate the model sum of squares. Thus the grand mean becomes the mean for when there was no alcohol, and the group means are the means for men (when there was no alcohol) and women (when there was no alcohol). So, we group the data by the amount of alcohol drunk. Within each of these three groups, we calculate the overall mean and also the mean of the male and female scores separately. These mean scores are all we really need. Pictorially, you can think of the data as displayed pictorially below.

We can then apply the same equation for the model sum of squares that we used for the overall model sum of squares, but we use the grand mean of the no-alcohol data (63.75) and the means of males (66.875) and females (60.625) within this group:

No Alcohol	
Female	Male

2 Pints	
Female	Male

4 Pints	
Female	Male

65	50
70	55
60	80
60	65
60	70
55	75
60	75
55	65
60.625	66.875

**Mean None =
63.75**

70	45
65	60
60	85
70	65
65	70
60	70
60	80
50	60
62.50	66.875

**Mean 2 Pints =
64.6875**

55	30
65	30
70	30
55	55
55	35
60	20
50	45
50	40
57.500	35.625

**Mean 4 Pints =
46.5625**

$$\begin{aligned}
 SS_{\text{Gender(No Alcohol)}} &= \sum n_k (\bar{x}_k - \bar{x}_{\text{grand}})^2 \\
 &= 8(60.625 - 63.75)^2 + 8(66.875 - 63.75)^2 \\
 &= 156.25
 \end{aligned}$$

The degrees of freedom for this effect are calculated the same way as for any model sum of squares ; that is, they are one less than the number of conditions being compared ($k - 1$), which in this case when we're comparing only two conditions will be 1.

The next step is to do the same but for the 2-pints data. Now we use the grand mean of the 2-pints data (64.6875) and the means of males (66.875) and females (62.50) within this group. The equation, however, stays the same:

$$\begin{aligned}
 SS_{\text{Gender(2 Pints)}} &= \sum n_k (\bar{x}_k - \bar{x}_{\text{grand}})^2 \\
 &= 8(62.50 - 64.6875)^2 + 8(66.875 - 64.6875)^2 \\
 &= 76.56
 \end{aligned}$$

The degrees of freedom are the same as in the previous simple effect, namely $k - 1$, which is 1 for these data.

The next step is to do the same but for the 4-pints data. Now we use the grand mean of the 4-pints data (46.5625) and the means of females (57.500) and males (35.625) within this group. The equation, however, stays the same:

$$\begin{aligned}
 SS_{\text{Gender(4 Pints)}} &= \sum n_k (\bar{x}_k - \bar{x}_{\text{grand}})^2 \\
 &= 8(57.50 - 46.5625)^2 + 8(35.625 - 46.5625)^2 \\
 &= 1914.06
 \end{aligned}$$

Again, the degrees of freedom are 1 (because we've compared two groups).

As with any ANOVA, we need to convert these sums of squares to mean squares by dividing by the degrees of freedom. However, because all of these sums of squares have 1 degree of freedom, so we're dividing by 1, the mean squares will be the same as the sum of squares. So, the final stage is to calculate an F -ratio for each simple effect. As ever, the F -ratio is just the mean squares for the model divided by the residual mean squares. So, you might well ask, what do we use for the residual mean squares? When conducting simple effects we use the residual mean squares for the original ANOVA (the residual mean squares for the entire model). In doing so we are merely partitioning the model sums of squares and so keep control of the Type I error rate. For these data, the residual sum of squares was 83.036 (see section 10.2.6). Therefore, we get:

$$F_{\text{Gender (No Alcohol)}} = \frac{MS_{\text{Gender (No Alcohol)}}}{MS_R} = \frac{156.25}{83.036} = 1.88$$

$$F_{\text{Gender (2 Pints)}} = \frac{MS_{\text{Gender (2 Pints)}}}{MS_R} = \frac{76.56}{83.036} = 0.92$$

$$F_{\text{Gender (4 Pints)}} = \frac{MS_{\text{Gender (4 Pints)}}}{MS_R} = \frac{1914.06}{83.036} = 23.05$$

We can evaluate these F -values in the usual way (they will have 1 and 42 degrees of freedom for these data). However, for the 2-pints data we can be sure there is not a significant effect of gender because the F -ratio is less than 1.

We can also convert them to t -values (to compare against the output from **R**) by remembering that F is simply the square root of t . Therefore:

$$t_{\text{Gender (No Alcohol)}} = \sqrt{1.88} = 1.37$$

$$t_{\text{Gender (2 Pints)}} = \sqrt{0.92} = 0.96$$

$$t_{\text{Gender (4 Pints)}} = \sqrt{23.05} = 4.80$$

Please Sir, can I have some more ... robust methods?



Robust three -way independent ANOVA

Smart Alex's Task 5 involves a three-way ANOVA: hospitals were reporting an increase in injuries related to playing Nintendo Wii (<http://www.telegraph.co.uk/news/uknews/1576244/Spate-of-injuries-blamed-on-Nintendo-Wii.html>). These injuries were attributed mainly to muscle and tendon strains. A researcher was interested to see whether these injuries could be prevented. She hypothesized that a stretching warm-up before playing Wii would help lower injuries, and that athletes would be less susceptible to injuries because their regular activity should make them more flexible. She took 60 athletes and 60 non-athletes (**athlete**), half of them played Wii and half watched others playing as a control (**wii**), and within these groups half did a 5-minute stretch routine before playing/watching whereas the other half did not (**stretch**). The outcome was a pain score out of 10 (where 0 is no pain, and 10 is severe pain) after playing for 4 hours (**injury**). The data are in the file **Wii.dat**. Conduct a three-way ANOVA to test whether athletes are less prone to injury, and whether the prevention programme worked.

Some graphs and how to obtain them in **R** are in the answers to the Smart Alex task, and you should also consult this section to see how the non-robust ANOVA was run, and how to interpret the results and so on.

This example is a three-way ANOVA because there are three independent variables (**athlete**, **stretch** and **wii**) each with two levels. The only function for conducting a three-way robust ANOVA where groups are independent is `t3way()`, which performs a three-way independent ANOVA on trimmed means. To access this function we need to load the *WRS* package (see the chapter).

The first problem we have is that, as with the functions discussed in the book chapter, this function needs the data to be in wide format rather than long (see Chapter 3). Essentially we want levels of our three factors to be represented in different columns. Therefore, rather than a dataframe with 4 columns and 120 rows, we want one with 8 columns and 15 rows.

Like the example in the chapter, we can use `melt()` and `cast()` to do the restructuring for us. To get the restructuring to work, we need to add a variable to our dataframe that identifies the rows in the wide format. The easiest approach is to create a variable (called **row**) that identifies within each of the eight groups (made up of the combination of the three variables) the row number of a given score. In other words, it will be a value from 1 to 20 telling us whether the score is the first, second, third etc. score within the chunk. We can add this variable to the dataframe by executing:

```
athleteData$row<-rep(1:15, 8)
```

This command uses the *rep()* function to create a variable *row* in the dataframe *athleteData*, that is, the numbers 1 to 15 repeated eight times (*rep(1:15, 8)*). The dataframe now looks like this (edited):

	athlete	stretch	wii	injury	row
1	Athlete	Stretching	Playing	Wii	2
2	Athlete	Stretching	Playing	Wii	2
3	Athlete	Stretching	Playing	Wii	1
4	Athlete	Stretching	Playing	Wii	2
5	Athlete	Stretching	Playing	Wii	0
6	Athlete	Stretching	Playing	Wii	1
7	Athlete	Stretching	Playing	Wii	2
8	Athlete	Stretching	Playing	Wii	0
9	Athlete	Stretching	Playing	Wii	2
10	Athlete	Stretching	Playing	Wii	2
11	Athlete	Stretching	Playing	Wii	2
12	Athlete	Stretching	Playing	Wii	1
13	Athlete	Stretching	Playing	Wii	4
14	Athlete	Stretching	Playing	Wii	2
15	Athlete	Stretching	Playing	Wii	2
16	Athlete	Stretching	Watching	Wii	0
17	Athlete	Stretching	Watching	Wii	0
18	Athlete	Stretching	Watching	Wii	3
19	Athlete	Stretching	Watching	Wii	3
20	Athlete	Stretching	Watching	Wii	3
21	Athlete	Stretching	Watching	Wii	2
22	Athlete	Stretching	Watching	Wii	1
23	Athlete	Stretching	Watching	Wii	0
24	Athlete	Stretching	Watching	Wii	2
25	Athlete	Stretching	Watching	Wii	2
26	Athlete	Stretching	Watching	Wii	3
27	Athlete	Stretching	Watching	Wii	2
28	Athlete	Stretching	Watching	Wii	2
29	Athlete	Stretching	Watching	Wii	3
30	Athlete	Stretching	Watching	Wii	1
31	Athlete	No Stretching	Playing	Wii	2
32	Athlete	No Stretching	Playing	Wii	4
33	Athlete	No Stretching	Playing	Wii	1
34	Athlete	No Stretching	Playing	Wii	2
35	Athlete	No Stretching	Playing	Wii	2
36	Athlete	No Stretching	Playing	Wii	2
37	Athlete	No Stretching	Playing	Wii	1
38	Athlete	No Stretching	Playing	Wii	4
39	Athlete	No Stretching	Playing	Wii	4
40	Athlete	No Stretching	Playing	Wii	1
41	Athlete	No Stretching	Playing	Wii	2
42	Athlete	No Stretching	Playing	Wii	3
43	Athlete	No Stretching	Playing	Wii	3
44	Athlete	No Stretching	Playing	Wii	3
45	Athlete	No Stretching	Playing	Wii	3
46	Athlete	No Stretching	Watching	Wii	2
47	Athlete	No Stretching	Watching	Wii	3
48	Athlete	No Stretching	Watching	Wii	2
49	Athlete	No Stretching	Watching	Wii	2
50	Athlete	No Stretching	Watching	Wii	2
51	Athlete	No Stretching	Watching	Wii	1
52	Athlete	No Stretching	Watching	Wii	0
53	Athlete	No Stretching	Watching	Wii	3
54	Athlete	No Stretching	Watching	Wii	3
55	Athlete	No Stretching	Watching	Wii	2
56	Athlete	No Stretching	Watching	Wii	1
57	Athlete	No Stretching	Watching	Wii	2
58	Athlete	No Stretching	Watching	Wii	4
59	Athlete	No Stretching	Watching	Wii	1
60	Athlete	No Stretching	Watching	Wii	2
61	Non-Athlete	Stretching	Playing	Wii	5
62	Non-Athlete	Stretching	Playing	Wii	5
63	Non-Athlete	Stretching	Playing	Wii	3
64	Non-Athlete	Stretching	Playing	Wii	6
65	Non-Athlete	Stretching	Playing	Wii	4
66	Non-Athlete	Stretching	Playing	Wii	3
67	Non-Athlete	Stretching	Playing	Wii	4
68	Non-Athlete	Stretching	Playing	Wii	5
69	Non-Athlete	Stretching	Playing	Wii	5
70	Non-Athlete	Stretching	Playing	Wii	2
71	Non-Athlete	Stretching	Playing	Wii	6
72	Non-Athlete	Stretching	Playing	Wii	4
73	Non-Athlete	Stretching	Playing	Wii	4
74	Non-Athlete	Stretching	Playing	Wii	4
75	Non-Athlete	Stretching	Playing	Wii	3
76	Non-Athlete	Stretching	Watching	Wii	4
77	Non-Athlete	Stretching	Watching	Wii	3
78	Non-Athlete	Stretching	Watching	Wii	2
79	Non-Athlete	Stretching	Watching	Wii	1
80	Non-Athlete	Stretching	Watching	Wii	4
81	Non-Athlete	Stretching	Watching	Wii	3
82	Non-Athlete	Stretching	Watching	Wii	2

```

83 Non-Athlete Stretching Watching Wii 2 8
84 Non-Athlete Stretching Watching Wii 1 9
85 Non-Athlete Stretching Watching Wii 3 10
86 Non-Athlete Stretching Watching Wii 1 11
87 Non-Athlete Stretching Watching Wii 1 12
88 Non-Athlete Stretching Watching Wii 3 13
89 Non-Athlete Stretching Watching Wii 4 14
90 Non-Athlete Stretching Watching Wii 2 15
91 Non-Athlete No Stretching Playing Wii 7 1
92 Non-Athlete No Stretching Playing Wii 8 2
93 Non-Athlete No Stretching Playing Wii 6 3
94 Non-Athlete No Stretching Playing Wii 9 4
95 Non-Athlete No Stretching Playing Wii 4 5
96 Non-Athlete No Stretching Playing Wii 7 6
97 Non-Athlete No Stretching Playing Wii 5 7
98 Non-Athlete No Stretching Playing Wii 9 8
99 Non-Athlete No Stretching Playing Wii 6 9
100 Non-Athlete No Stretching Playing Wii 4 10
101 Non-Athlete No Stretching Playing Wii 8 11
102 Non-Athlete No Stretching Playing Wii 5 12
103 Non-Athlete No Stretching Playing Wii 4 13
104 Non-Athlete No Stretching Playing Wii 7 14
105 Non-Athlete No Stretching Playing Wii 10 15
106 Non-Athlete No Stretching Watching Wii 1 1
107 Non-Athlete No Stretching Watching Wii 3 2
108 Non-Athlete No Stretching Watching Wii 2 3
109 Non-Athlete No Stretching Watching Wii 1 4
110 Non-Athlete No Stretching Watching Wii 3 5
111 Non-Athlete No Stretching Watching Wii 3 6
112 Non-Athlete No Stretching Watching Wii 2 7
113 Non-Athlete No Stretching Watching Wii 3 8
114 Non-Athlete No Stretching Watching Wii 4 9
115 Non-Athlete No Stretching Watching Wii 2 10
116 Non-Athlete No Stretching Watching Wii 0 11
117 Non-Athlete No Stretching Watching Wii 1 12
118 Non-Athlete No Stretching Watching Wii 3 13
119 Non-Athlete No Stretching Watching Wii 2 14
120 Non-Athlete No Stretching Watching Wii 0 15

```

Note that the structure is the same as before, it's just that we have a new variable called **row** that identifies the scores within each combination of **athlete**, **stretch** and **wii** as a value from 1 to 15.

Now that we have changed the data set, we need to make it molten so that we can cast the data into the wide format. To do this we use the `melt()` function. We need to differentiate variables that identify attributes of the scores (in this case, **athlete**, **stretch**, **wii**, and **row**) from the scores or measured variables themselves (in this case **injury**). Attributes are specified with the *id* option, and scores with the *measured* option. Therefore, we can create a molten dataframe called *athleteMelt* by executing:

```
athleteMelt<-melt(athleteData, id = c("row", "athlete", "stretch", "wii"), measured = c("injury"))
```

Having melted the data we want to cast it in the wide format using `cast()`. To do this we use a formula in the form: *variables specifying the rows ~ variables specifying the columns*. In this case, **row** tells us in which row to place a score, and we want the alcohol and gender variables split across different columns, so we'd use the formula: *row ~ athlete + stretch + wii*. Therefore, we can make a wide dataframe called *athleteWide* by executing:

```
athleteWide<-cast(athleteMelt, row ~ athlete + stretch + wii)
```

Note that we have applied this command to the molten data set (*athleteMelt*). The result is that the data have been transformed from the long format to the wide format. However, because we added the variable **row** to the data frame, our new dataframe also contains this variable, and for the analysis we want only the independent variables, therefore, we want to remove **row**. We can do this by executing:

```
athleteWide$row<-NULL
```

which basically zaps the variable **row** into oblivion. If you look at the dataframe you'll see a lovely wide format set of data:

Athlete	Athlete				Non-Athlete			
Stretch	No Stretch		Stretch		No Stretch		Stretch	
Wii	Play	Watch	Play	Watch	Play	Watch	Play	Watch

	2	2	2	0	7	1	5	4
	4	3	2	0	8	3	5	3
	1	2	1	3	6	2	3	2
	2	2	2	3	9	1	6	1
	2	2	0	3	4	3	4	4
	2	1	1	2	7	3	3	3
	1	0	2	1	5	2	4	2
	4	3	0	0	9	3	5	2
	4	3	2	2	6	4	5	1
	1	2	2	2	4	2	2	3
	2	1	2	3	8	0	6	1
	3	2	1	2	5	1	4	1
	3	4	4	2	4	3	4	3
	3	1	2	3	7	2	4	4
	3	2	2	1	10	0	3	2

It's important to note the order of the columns because this affects how we specify the robust analysis. In this case, the hierarchy of the independent variables is **athlete** followed by **stretch** followed by **wii**. We would say that **athlete** is factor A, **stretch** is factor B and **wii** is factor C. As such, the order of the columns reflects a $2 \times 2 \times 2$ design (two levels of athlete divided up into two levels of stretching, divided into two levels of Wii activity).

The function `t3way()` takes the general form:

```
t3way(levels of factor A, levels of factor B, levels of factor C, data, tr = .2, alpha = .05)
```

As with other functions we've encountered, the level of trimming is by default 20% ($tr = .2$), but can be changed by including the $tr =$ option. Also, the default alpha level is .05 but can be changed by including the $alpha =$ option. Assuming we are happy with the default level of trimming, we need only specify the dataframe (*athleteWide*) and the levels of factor A (2 in this case, as explained above), factor B (2 in this case) and factor C (again 2). Therefore, we can do a robust three-way factorial ANOVA based on trimmed means by executing:

```
t3way(2,2,2, athleteWide)
```

The output of this command is shown below. We are given a test statistic for factor A ($\$Qa$), factor B ($\Qb), and factor C ($\$Qc$) and all of their interactions ($\$Qab$) as well as the corresponding p -value ($\$A.p.value$, $\$B.p.value$, and $\$C.p.value$ etc.). If the p -value is less than .05 then we say the effect was significant.

```
$Qa
[1] 44.17362

$Qa.crit
[1] 4.037312

$A.p.value
[1] 1e-04

$Qb
[1] 6.956038

$Qb.crit
[1] 4.037312

$B.p.value
[1] 0.012

$Qc
[1] 38.74235

$Qc.crit
[1] 12.17617

$C.p.value
[1] 0.001
```

```

$Qab
[1] 1.880913

$Qab.crit
[1] 1.876811

$AB.p.value
[1] 0.177

$Qac
[1] 33.66722

$Qac.crit
[1] 12.17617

$AC.p.value
[1] 0.001

$Qbc
[1] 9.360045

$Qbc.crit
[1] 9.093685

$BC.p.value
[1] 0.004

$Qabc
[1] 3.216472

$Qabc.crit
[1] 3.196099

$ABC.p.value
[1] 0.08

```

Remember that factor A was **athlete**, factor B **stretch** and factor C **wii**. We could tabulate these effects as follows (which makes them a little easier to understand). To sum up, all of the main effects are significant, and the **athlete** × **wii** and **stretch** × **wii** interactions also. Most important, the three-way interaction is not significant (which differs from what you should find when you run the non-robust ANOVA). Compare these results with the answers to smart Alex's task, and also see there for how to interpret the significant effects.

Effect	Label	Value	Critical Value	p-value	Significant?
Athlete	\$Qa	44.17362	4.037312	1e-04	Y
Stretch	\$Qb	6.956038	4.037312	0.012	Y
Wii	\$Qc	38.74235	12.17617	0.001	Y
Athlete × Stretch	\$Qab	1.880913	1.876811	0.177	N
Athlete × Wii	\$Qac	33.66722	12.17617	0.001	Y
Stretch × Wii	\$Qbc	9.360045	9.093685	0.004	Y
Athlete × Stretch × Wii	\$Qabc	3.216472	3.196099	0.08	N

There is not a specific procedure for *post hoc* tests in three-way robust ANOVA. However, if the three-way interaction is significant you could split the data by one of the independent variables and run separate two-way robust ANOVAs. For example, here we might run separate **stretch** × **wii** ANOVAs for athletes and non-athletes. Having done this, you can use the *post hoc* procedures described in the book chapter.

Labcoat Leni's real research

Don't forget your toothbrush?

Davey, G. C. L., et al. (2003). *Journal of Behavior Therapy & Experimental Psychiatry*, 34, 141–160.



We have all experienced that feeling after we have left the house of wondering whether we locked the door, or closed the window, or whether we remembered to remove the bodies from the fridge in case the police turn up. This behaviour is normal; however, people with obsessive compulsive disorder (OCD) tend to check things excessively. They might, for example, check whether they have locked the door so often that it takes them an hour to leave their house. It is a very debilitating problem.

One theory of this checking behaviour in OCD suggests that it is caused by a combination of the mood you are in (positive or negative) interacting with the rules you use to decide when to stop a task (do you continue until you feel like stopping, or until you have done the task as best as you can?). Davey, Startup, Zara, MacDonald, and Field (2003) tested this hypothesis by inducing a negative, positive or no mood in different people and then asking them to imagine that they were going on holiday and to generate as many things as they could that they should check before they went away. Within each mood group, half of the participants were instructed to generate as many items as they could (known as an ‘as many as can’ stop rule), whereas the remainder were asked to generate items for as long as they felt like continuing the task (known as a ‘feel like continuing’ stop rule). The data are in the file **Davey2003.dat**.

Davey et al. hypothesized that people in negative moods, using an ‘as many as can’ stop rule, would generate more items than those using a ‘feel like continuing’ stop rule. Conversely, people in a positive mood would generate more items when using a ‘feel like continuing’ stop rule compared to an ‘as many as can’ stop rule. Finally, in neutral moods, the stop rule used shouldn’t affect the number of items generated. Draw an error bar chart of the data and then conduct the appropriate analysis to test Davey et al.’s hypotheses.

Solution

First of all load in the data (presuming that you have set your working directory to be the location of the **Davey2003.dat** file):

```
daveyData<-read.delim("Davey2003.dat", header = TRUE)
```

Next, we should set the variables **Mood** and **Stop_Rule** to be factors:

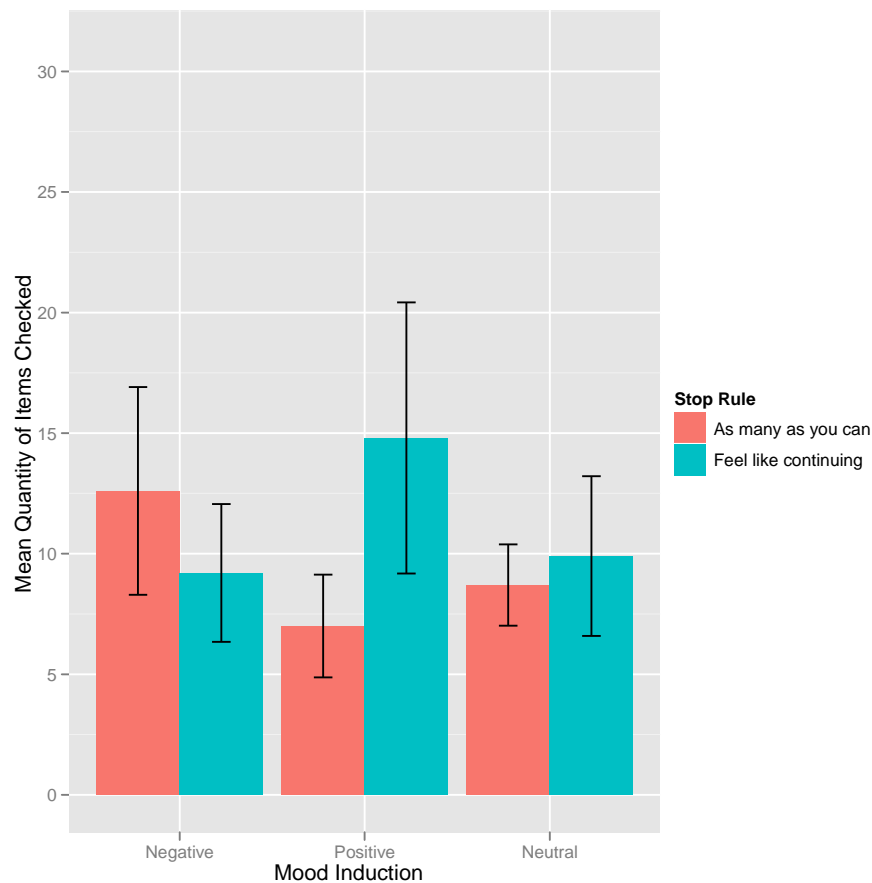
```
daveyData$Mood<-factor(daveyData$Mood, levels = c(1:3), labels = c("Negative",  
"Positive", "Neutral"))
```

```
daveyData$Stop_Rule<-factor(daveyData$Stop_Rule, levels = c(1:2), labels = c("As many  
as you can", "Feel like continuing"))
```

Next we can draw the error bar graph with different colours to represent different stop rules:

```
bar <- ggplot(daveyData, aes(Mood, Checks, fill = Stop_Rule))  
bar + stat_summary(fun.y = mean, geom = "bar", position="dodge") +  
  stat_summary(fun.data = mean_cl_normal, geom = "errorbar",  
  position=position_dodge(width=0.90), width = 0.2) + labs(x = "Mood Induction", y =  
  "Mean Quantity of Items Checked", fill = "Stop Rule")
```

The resulting graph should look like this:



To get some descriptive statistics of the interaction between **Mood** and **Stop_Rule**, we can execute:

```
by(daveyData$Checks, list(daveyData$Stop_Rule, daveyData$Mood), stat.desc)
```

```
:As many as you can
:Negative
  median    mean    SE.mean  CI.mean.0.95    var    std.dev    coef.var
 12.000000 12.600000  1.9043809  4.3080089    36.2666667  6.0221812  0.4779509
-----
:Feel like continuing
:Negative
  median    mean    SE.mean  CI.mean.0.95    var    std.dev    coef.var
   8.500000   9.200000  1.2631530  2.8574507    15.9555556  3.9944406  0.4341783
-----
:As many as you can
:Positive
  median    mean    SE.mean  CI.mean.0.95    var    std.dev    coef.var
```

```

7.0000000 7.0000000 0.9428090 2.1327822 8.8888889 2.9814240 0.4259177
-----
:Feel like continuing
:Positive
  median    mean    SE.mean    CI.mean.0.95    var    std.dev    coef.var
12.0000000 14.8000000 2.4846194 5.6205995 61.7333333 7.8570563 0.5308822
-----
:As many as you can
:Neutral
  median    mean    SE.mean    CI.mean.0.95    var    std.dev    coef.var
9.5000000 8.7000000 0.7461010 1.6877977 5.5666667 2.3593784 0.2711929
-----
:Feel like continuing
:Neutral
  median    mean    SE.mean    CI.mean.0.95    var    std.dev    coef.var
9.0000000 9.9000000 1.4640128 3.3118269 21.4333333 4.6296148 0.4676379
-----

```

In the resulting (edited) output above, we can see that in the ‘as many as you can’ condition, people induced with a negative mood typically checked around 13 items, people induced with a positive mood typically checked around 7 items, and those induced with a neutral mood checked around 9 items. In the ‘feel like continuing’ condition, people induced with a negative mood typically checked around 9 items, people induced with a positive mood typically checked around 15 items, and those induced with a neutral mood checked around 10 items.

The final thing to do at this stage is to compute Levene’s test to see whether the variances differ across all six groups (not just the two stop rule groups and three mood groups). To do this we can add the *interaction()* option to the Levene’s test function, which will compute Levene’s test across any combination of groups for the variables specified within *interaction()*:

```
leveneTest(daveyData$Checks, interaction(daveyData$Mood, daveyData$Stop_Rule), center = median)
```

```

Levene's Test for Homogeneity of Variance (center = median)
      Df    F value    Pr(>F)
group  5     1.7291     0.1436
      54

```

Levene’s test is non-significant, indicating that there is no problem with homogeneity of variance.

Now we can do the factorial ANOVA. Remember to set some orthogonal contrasts, otherwise the Type III sums of squares will not compute properly. It makes sense for the first mood contrast to compare a neutral mood (the control condition) against both positive and negative moods; then, in the second mood contrast, we could compare positive mood with negative mood. With regard to stop rule, there are only two levels and so we can only set one contrast to compare ‘as many as you can’ with ‘feel like continuing’:

```

contrasts(daveyData$Mood)<-cbind(c(1, 1, -2), c(-1, 1, 0))
contrasts(daveyData$Stop_Rule)<-c(-1, 1)
daveyModel<-aov(Checks ~ Stop_Rule+Mood + Stop_Rule:Mood, data = daveyData)
Anova(daveyModel, type="III")

```

Anova Table (Type III tests)

```

Response: Checks
      Sum Sq Df    F value    Pr(>F)
(Intercept) 6448.1 1 258.1904 < 2.2e-16 ***
Stop_Rule    52.3 1  2.0928  0.153771
Mood         34.1 2  0.6834  0.509222
Stop_Rule:Mood 316.9 2  6.3452  0.003349 **
Residuals   1348.6 54

```

NB: If you get a different set of results than me, make sure that you have set some orthogonal contrasts. Even if you think setting contrasts is pointless (i.e. all your variables have only two levels) you still need to set some orthogonal contrasts before running the ANOVA for the Type III sums of squares to compute properly.

The main effect of mood was not significant, $F(2, 54) = 0.68$, $p > .05$, indicating that the number of checks (when we ignore the stop rule adopted) was roughly the same regardless of whether the person was in a positive, negative or neutral mood. Similarly, the main effect of stop rule was not significant, $F(1, 54) = 2.09$, $p > .05$, indicating that the number of checks (when we ignore the mood induced) was roughly the same regardless of whether the person used an ‘as many as can’ or a ‘feel like continuing’ stop rule. The mood \times stop rule interaction was significant, $F(2, 54) = 6.35$, $p < .01$, indicating that the mood combined with the stop rule affected checking behaviour. Looking at the

graph, a negative mood in combination with an ‘as many as can’ stop rule increased checking as did the combination of a ‘feel like continuing’ stop rule and a positive mood, just as Davey et al. predicted.

To view the output of the orthogonal contrasts that we set earlier execute:

```
summary.lm(daveyModel)
```

Coefficients:

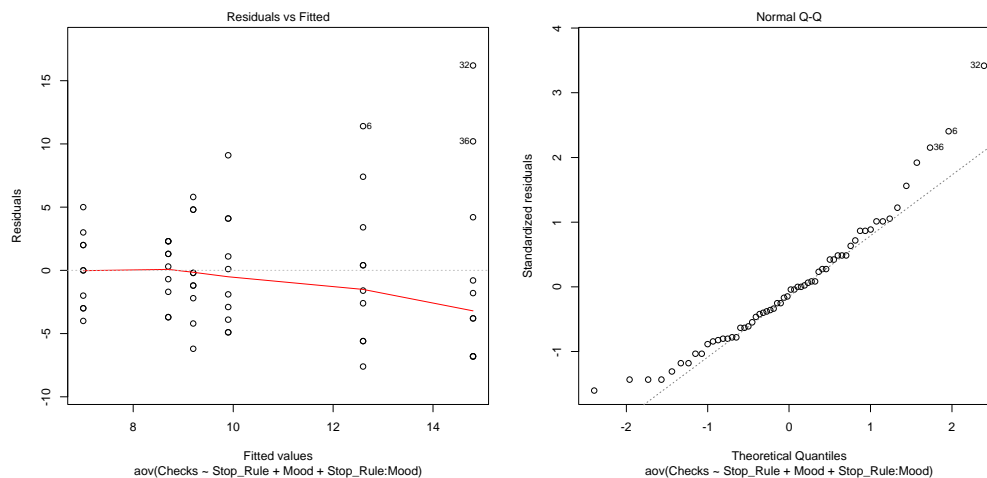
	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	1.037e+01	6.452e-01	16.068	< 2e-16	***
Stop_Rule1	9.333e-01	6.452e-01	1.447	0.153771	
Mood1	5.333e-01	4.562e-01	1.169	0.247506	
Mood2	-1.597e-16	7.902e-01	0.000	1.000000	
Stop_Rule1:Mood1	1.667e-01	4.562e-01	0.365	0.716286	
Stop_Rule1:Mood2	-2.800e+00	7.902e-01	-3.544	0.000823	***

- **Stop_Rule1**: This is the contrast for the main effect of stop rule; because stop rule has only two groups, this is the same as the effect of **Stop_Rule** from the ANOVA output.
- **Mood1**: This contrast compares the neutral mood group to the positive and negative mood groups. This tests whether the mean of the neutral mood group is different from the mean of the positive and negative mood groups combined. The p -value is .25, which is larger than .05, indicating a non-significant difference. So we could conclude that the effect of positive or negative mood induction on the number of checks a person makes before leaving the house is not significantly different from the effect of neutral mood induction on the number of checks.
- **Mood2**: This contrast tests whether the mean of the positive mood group is different from the mean of the negative mood group. The p -value is 1, which is larger than .05, and therefore indicates a non-significant difference between the groups. We can conclude that a positive mood induction vs. a negative mood induction did not significantly affect the number of checks generated.
- **Stop_Rule1:Mood1**: This contrast tests whether the effect of **Mood1** described above is different in the different stop rule groups, ‘as many as can’ and ‘feel like continuing’. It answers the question: is the combined effect of positive and negative mood induction, compared to neutral mood induction, on the number of checks generated comparable in the two stop rule groups? The p -value is .72, which is non-significant, so the answer is ‘yes, the extent to which positive and negative mood induction vs. neutral mood induction has an effect on number of checks is similar in the two stop rule groups’.
- **Stop_Rule1:Mood2**: This contrast tests whether the effect of **Mood2** described above is different in the two stop rule groups. It answers the question: is the effect of positive mood induction, compared to negative mood induction, on the number of checks comparable when participants are asked to generate as many items as they could (‘as many as can’ stop rule) to when they are asked to generate items for as long as they felt like continuing the task (‘feel like continuing’ stop rule). The p -value is less than .001, which is significant, so the answer is ‘no, the extent to which positive and negative mood induction has an effect on the number of checks is different in the two stop rule groups’. We can see from the graph above that people in negative moods, using an ‘as many as can’ stop rule, generated more items than those using a ‘feel like continuing’ stop rule. Conversely, people in a positive mood generated more items when using a ‘feel like continuing’ stop rule compared to an ‘as many as can’ stop rule. This again is in line with what Davey et al. predicted.

The `aov()` function automatically generates some plots that we can use to test the assumptions. We can see these graphs by executing:

```
plot(daveyModel)
```

The first graph below (on the left) can be used for testing homogeneity of variance: if it has a funnel shape then we’re in trouble. The plot we have does show funnelling, which implies that the residuals might be heteroscedastic. The second plot is a Q-Q plot, which tells about the normality of residuals in the model. We want our residuals to be normally distributed, which means that the dots on the graph should hover around the diagonal line. On ours, there is only some deviation from the line, suggesting that we can assume normality of our residuals/errors.



Smart Alex's solutions

Task 1

- People's musical tastes change as they get older (my parents, for example, after years of listening to relatively cool music when I was a kid, subsequently hit their mid-forties and developed a worrying obsession with country and western music). This worries me immensely because the future seems bleak if it is spent listening to Garth Brooks and thinking 'oh boy, did I underestimate Garth's immense talent when I was in my 20s'. So, I did some imaginary research to find out whether my fate really was sealed, or whether it's possible to be old and like good music too. First, I got two groups of people (45 people in each group): one group contained young people (which I arbitrarily decided was under 40 years of age) and the other group contained more mature individuals (above 40 years of age). This is my first independent variable, **age**. I then split each of these groups of 45 into three smaller groups of 15 and assigned them to listen to Fugazi (who everyone knows are the coolest band on the planet), ABBA or Barf Grooks (a less well-known country and western musician not to be confused with anyone real who produces music that makes me want to barf). This is my second independent variable, **music**. After listening to the music I got each person to rate it on a scale ranging from -100 (please poke a pencil through my eardrum so I don't have to listen any more) through 0 (I am completely indifferent) to +100 (I love this music so much, it gives me a tingle down my spine). This variable is called **liking**. The data are in the file **fugazi.dat**. Conduct a two-way independent ANOVA on them.

Read in the data:

```
fugaziData<-read.delim("fugazi.dat", header = TRUE)
```

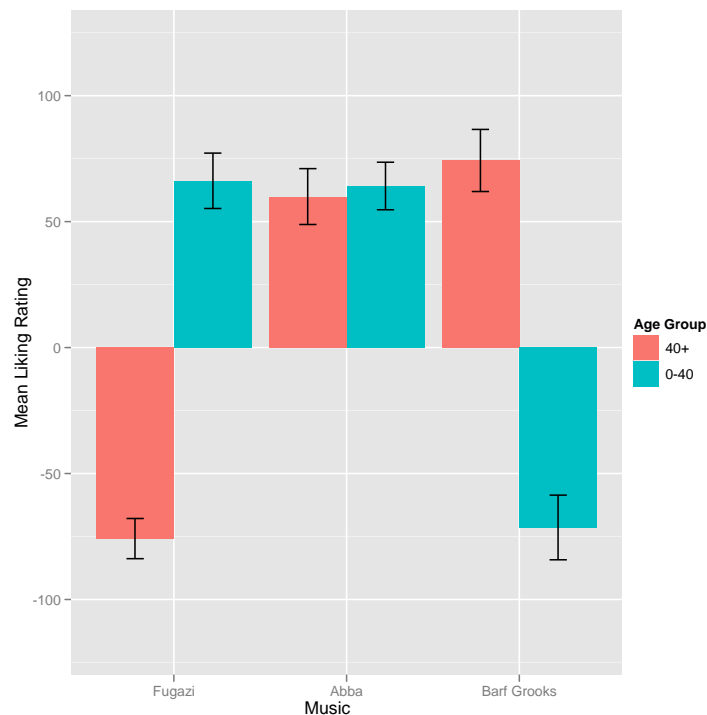
Set music and age to be factors:

```
fugaziData$music<-factor(fugaziData$music, levels = c(1:3), labels = c("Fugazi",  
"Abba", "Barf Grooks"))
```

```
fugaziData$age<-factor(fugaziData$age, levels = c(1:2), labels = c("40+", "0-40"))
```

Now produce an error bar graph:

```
bar <- ggplot(fugaziData, aes(music, liking, fill = age))  
bar + stat_summary(fun.y = mean, geom = "bar", position="dodge") +  
stat_summary(fun.data = mean_cl_normal, geom = "errorbar",  
position=position_dodge(width=0.90), width = 0.2) + labs(x = "Music", y = "Mean  
Liking Rating", fill = "Age Group")
```



The error bar chart of the music data shows the mean rating of the music played to each group. It's clear from this chart that when people listened to Fugazi the two age groups were divided: the older ages rated it very low, but the younger people rated it very highly. A reverse trend is found if you look at the ratings for Barf Grooks: the youngsters give it low ratings while the wrinkly ones love it. For ABBA the groups agreed: both old and young rated them highly.

We can produce some descriptive statistics:

```
by(fugaziData$liking, list(fugaziData$age, fugaziData$music), stat.desc)
```

```

: 40+
: Fugazi
mean      SE.mean      CI.mean.0.95      var      std.dev      coef.var
-75.8666667  3.7108165  7.9589097  206.5523810  14.3719303  -0.1894367
-----
: 0-40
: Fugazi
mean      SE.mean      CI.mean.0.95      var      std.dev      coef.var
66.2000000  5.1392051  11.0224986  396.1714286  19.9040556  0.3006655
-----
: 40+
: Abba
mean      SE.mean      CI.mean.0.95      var      std.dev      coef.var
59.9333333  5.1597957  11.0666612  399.3523810  19.9838030  0.3334339
-----
: 0-40
: Abba
mean      SE.mean      CI.mean.0.95      var      std.dev      coef.var
64.1333333  4.3881514  9.4116488  288.8380952  16.9952374  0.2649985
-----
: 40+
: Barf Grooks
mean      SE.mean      CI.mean.0.95      var      std.dev      coef.var
74.2666667  5.7565422  12.3465550  497.0666667  22.2949920  0.3002019
-----
: 0-40
: Barf Grooks
mean      SE.mean      CI.mean.0.95      var      std.dev      coef.var
-71.4666667  5.9847956  12.8361098  537.2666667  23.1790135  -0.3243332

```

The above (edited) descriptive statistics output will be useful for interpreting the significant effects later on.

We want to do Levene's test:

```
leveneTest(fugaziData$liking, interaction(fugaziData$music, fugaziData$age),
```

```
center = median)
```

The following output shows the result. For these data the significance value is 0.383, which is greater than the criterion of .05. This means that the variances in the different experimental groups are roughly equal (i.e. not significantly different), and that the assumption has been met.

```
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  5  1.0707 0.3825
      84
```

Next we can conduct the two-way ANOVA. Remember that for the Type III sums of squares to work, we have to set orthogonal contrasts. I am going to set the first **music** contrast to compare ABBA with both Fugazi and Barf Grooks, and the second **music** contrast to compare Fugazi with Barf Grooks. For **age** there are only two levels (0–40 and 40+) and so I will set the **age** contrast to compare these two age groups.

```
contrasts(fugaziData$music)<-cbind(c(1, -2, 1), c(1, 0, -1))
contrasts(fugaziData$age)<-c(-1, 1)
fugaziModel<-aov(liking ~ music*age, data = fugaziData)
Anova(fugaziModel, type="III")
```

```
Anova Table (Type III tests)
```

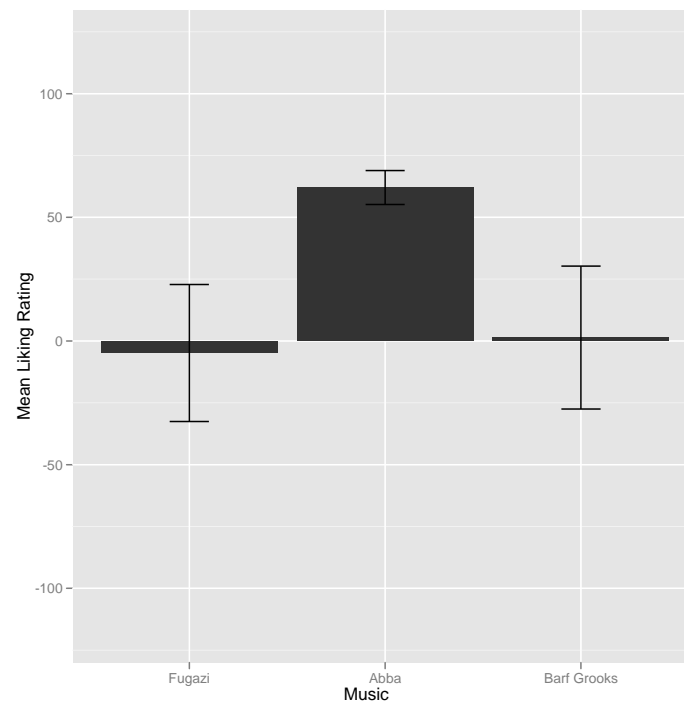
```
Response: liking
      Sum Sq Df F value    Pr(>F)
(Intercept) 34340 1  88.6089 8.744e-15 ***
music        81864 2 105.6198 < 2.2e-16 ***
age           1 1  0.0018  0.9659
music:age    310790 2 400.9769 < 2.2e-16 ***
Residuals   32553 84
```

NB: If you get a different set of results than me, make sure that you have set some orthogonal contrasts. Even if you think setting contrasts is pointless (i.e. all your variables have only two levels) you still need to set some orthogonal contrasts before running the ANOVA for the Type III sums of squares to compute properly.

The main effect of music is shown by the *F*-ratio in the row labeled **music**; in this case the significance is much lower than the usual cut-off point of .05. Hence, we can say that there was a significant effect of the type of music on the ratings. To understand what this actually means, we need to look at the mean ratings for each type of music when we ignore whether the person giving the rating was old or young:

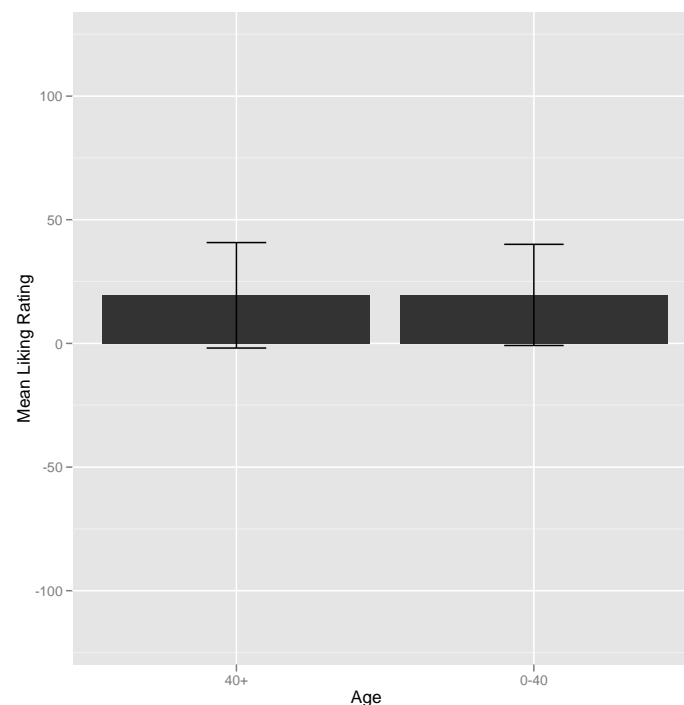
```
bar <- ggplot(fugaziData, aes(music, liking))

bar + stat_summary(fun.y = mean, geom = "bar", position="dodge") +
  stat_summary(fun.data = mean_cl_normal, geom = "errorbar",
    position=position_dodge(width=0.90), width = 0.2) + labs(x = "Music", y = "Mean
    Liking Rating")
```



What this graph shows is that the significant main effect of music is likely to reflect the fact that ABBA were rated (overall) much more positively than the other two artists.

The main effect of age is shown by the F -ratio in the row labelled *age*; the probability associated with this F -ratio is 0.966, which is so close to 1 that it means that it is a virtual certainty that this F could occur by chance alone. Again, to interpret the effect we need to look at the mean ratings for the two age groups ignoring the type of music to which they listened.



This graph shows that when you ignore the type of music that was being rated, older people, on average, gave almost identical ratings to younger people (i.e. the mean ratings in the two groups are virtually the same).

The interaction effect is shown by the F -ratio in the row labeled **age:music**; the associated significance value is again much less than the criterion of .05. Therefore, we can say that there is a

significant interaction between age and the type of music rated. To interpret this effect we need to look at the mean ratings in all conditions, and these means were originally plotted at the beginning of this answer. The fact there is a significant interaction tells us that for certain types of music the different age groups gave different ratings. In this case, although they agree on ABBA, there are large disagreements in ratings of Fugazi and Barf Grooks.

To view the results of the orthogonal contrasts that we specified earlier, execute the following command:

```
summary.lm(fugaziModel)
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  19.53333    2.07509   9.413 8.74e-15 ***
music1       -21.25000    1.46731 -14.482 < 2e-16 ***
music2       -3.11667    2.54146  -1.226  0.224
age1         0.08889     2.07509   0.043  0.966
music1:age1  -1.00556    1.46731  -0.685  0.495
music2:age1  71.95000    2.54146  28.310 < 2e-16 ***
```

- *age1*: This is the contrast for the main effect of age; because age has only two groups this is the same as the effect of **age** from the ANOVA output.
- *music1*: This contrast compares ABBA to Fugazi and Barf Grooks. This tests whether the mean of the ABBA group is different to the mean of the Fugazi and Barf Grooks groups combined. The *p*-value is much smaller than .05, indicating a significant difference. So we could conclude that liking is significantly different for ABBA compared to Fugazi and Barf Grooks. Looking at the error bar graph, we can see that liking is significantly higher for ABBA than for Fugazi and Barf Grooks.
- *music2*: This contrast tests whether the mean liking of Fugazi is different to the mean liking of Barf Grooks. The *p*-value is .22, which is larger than .05, and therefore indicates a non-significant difference between the two music groups. We can conclude that liking is similar for Fugazi and Barf Grooks.
- *music1:age1*: This contrast tests whether the effect of **music1** described above is different in the different age groups, 0–40 and 40+. It answers the question: ‘is the combined effect of Fugazi and Barf Grooks, compared to ABBA, on liking comparable in the different age groups?’ The *p*-value is .50, which is non-significant, so the answer is ‘yes, the extent to which Fugazi and Barf Grooks vs. ABBA has an effect on liking is not significantly different in people who are under 40 and those who are over 40’.
- *music2:age1*: This contrast tests whether the effect of **music2** described above is different in the two age groups. It answers the question: ‘is the effect of Fugazi compared to Barf Grooks on liking ratings comparable in the two age groups?’ The *p*-value is significant, so the answer is ‘no, the extent to which Fugazi and Barf Grooks has an effect on liking rating is different in two age groups’. We can see from the graph above that when people listened to Fugazi the older ages rated it very low, but the younger people rated it very highly. A reverse trend is found for the ratings for Barf Grooks: the under 40’s give it low ratings while the over 40’s gave it comparably high ratings.

Because we have conducted some contrasts, we do not need to look at *post hoc* tests as they should just tell us what we have already found out from the contrasts above. However, I will show you how you conduct some *post hoc* tests anyway for the sake of practice.

Given that we found a main effect of music, and of the interaction between music and age, we can look at some of the *post hoc* tests to establish where the difference lies. Let’s conduct a Bonferroni *post hoc* test. We can do this by executing the following command:

```
pairwise.t.test(fugaziData$liking, fugaziData$music, p.adjust.method = "bonferroni")

              Fugazi   Abba
Abba          0.00026    -
Barf Grooks  1.00000  0.00099
```

The output shows the result of the Bonferroni *post hoc* tests. First, ratings of Fugazi are compared to ABBA, which reveals a significant difference (the significance value is less than .05), and then Barf

Grooks, which reveals no difference (the significance value is greater than .05). In the next column, ratings of ABBA are compared to Barf Grooks, which reveals a significant difference (the significance value is below .05).

Calculating effect sizes

First, make sure that you have executed the function from Chapter 12. After you have done this once, you do not need to execute it again. Having executed this function we can use it to calculate omega squared in the current data by using values of n (15 people per group), a (levels of age = 2), b (levels of music = 3) and the four sums of squares from the ANOVA output above:

```
omega_factorial(15,2,3,1,81864,310790,32553)

[1] "Omega-Squared A: -0.000908223141102699"
[1] "Omega-Squared B: 0.19053049613252"
[1] "Omega-Squared AB: 0.728426175925751"
```

For the main effect of age we get, $\omega_{\text{age}}^2 = -.00$; for the main effect of music we get, $\omega_{\text{music}}^2 = .19$; and for the interaction, $\omega_{\text{age} \times \text{music}}^2 = \omega^2 = .73$.

Interpreting and writing the result

As with the other ANOVAs we've encountered, we have to report the details of the F -ratio and the degrees of freedom from which it was calculated. For the various effects in these data the F -ratios will be based on different degrees of freedom: they were derived from dividing the mean squares for the effect by the mean squares for the residual. For the effects of music and the **music \times age** interaction, the model degrees of freedom were 2 ($df_M = 2$), but for the effect of age the degrees of freedom were only 1 ($df_M = 1$). For all effects, the degrees of freedom for the residuals were 84 ($df_R = 84$). We can, therefore, report the three effects from this analysis as follows:

- ✓ The main effect of age on the ratings of the music was non-significant ($F(1, 84) = .00, p = 1, \omega^2 = -.00$).
- ✓ The results show that the main effect of the type of music listened to significantly affected the ratings of that music ($F(2, 84) = 105.62, p < .001, \omega^2 = .19$). The Bonferroni *post hoc* test revealed that ABBA were rated significantly higher than both Fugazi and Barf Grooks (both $ps < .01$).
- ✓ The interaction between music and age was significant ($F(2, 84) = 400.98, p < .001, \omega^2 = .73$), indicating that different types of music were rated differently by the two age groups. Specifically, Fugazi were rated more positively by the young group ($M = 66.20, SD = 19.90$) than the old ($M = -75.87, SD = 14.37$); ABBA were rated fairly equally in the young ($M = 64.13, SD = 16.99$) and old groups ($M = 59.93, SD = 19.98$); Barf Grooks was rated less positively by the young group ($M = -71.47, SD = 23.17$) compared to the old ($M = 74.27, SD = 22.29$). These findings indicate that there is no hope for me — the minute I hit 40 I will suddenly start to love country and western music and will burn all of my Fugazi CDs (it will never happen ... arghhhh!!!).

Task 2

- In Chapter 3 we used some data that related to men and women's arousal levels when watching either *Bridget Jones's Diary* or *Memento* (**ChickFlick.dat**). Analyse these data to see whether men and women differ in their reactions to different types of films.

First of all, load in the data:

```
chickFlick<-read.delim("ChickFlick.dat", header = TRUE)
```

If you execute:

```
chickFlick
```

You will be able to view the contents of the dataframe. If the data have loaded in as text then you do not need to set the categorical variables **gender** and **film** to be factors. However, if the data have loaded as numbers then you do need to set these variables to be factors. In my case, the data have loaded as text and so I do not need to set **gender** and **film** to be factors, as **R** has already assumed that they are factors.

Next, let's run Levene's test:

```
leveneTest(chickFlick$arousal, interaction(chickFlick$film, chickFlick$gender), center = median)
```

```
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  3  0.8311 0.4856
      36
```

The output shows that for these data the significance value is 0.486, which is greater than the criterion of .05. This means that the variances in the different experimental groups are roughly equal (i.e. not significantly different), and that the assumption has been met.

Next we can conduct the ANOVA, remembering to set some orthogonal contrasts. We need to set orthogonal contrasts even though both of the variables (gender and film) have only two levels, otherwise the Type III sums of squares will not compute properly:

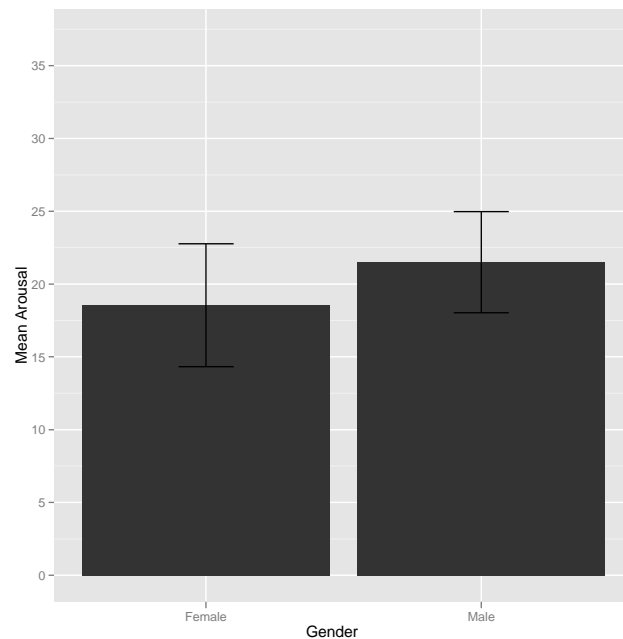
```
contrasts(chickFlick$film)<-c(-1, 1)
contrasts(chickFlick$gender)<-c(-1, 1)
chickFlick<-aov(arousal ~ gender*film, data = chickFlick)
Anova(chickFlick, type="III")
Anova Table (Type III tests)
```

```
Response: arousal
      Sum Sq Df F value    Pr(>F)
(Intercept) 16040.0 1 393.4325 < 2.2e-16 ***
gender         87.0 1  2.1346  0.1527
film        1092.0 1 26.7854 8.78e-06 ***
gender:film    34.2 1  0.8395  0.3656
Residuals    1467.7 36
```

The main effect of gender is shown by the *F*-ratio in the row labelled *gender*; in this case the significance is .15, which is greater than the usual cut-off point of .05. Hence, we can say that there was not a significant effect of gender on arousal during the films. To understand what this actually means, we need to look at the mean arousal levels for men and women (when we ignore which film they watched). We can look at this by creating an error bar graph:

```
bar <- ggplot(chickFlick, aes(gender, arousal))

bar + stat_summary(fun.y = mean, geom = "bar", position="dodge") +
  stat_summary(fun.data = mean_cl_normal, geom = "errorbar",
    position=position_dodge(width=0.90), width = 0.2) + labs(x = "Gender", y = "Mean Arousal")
```

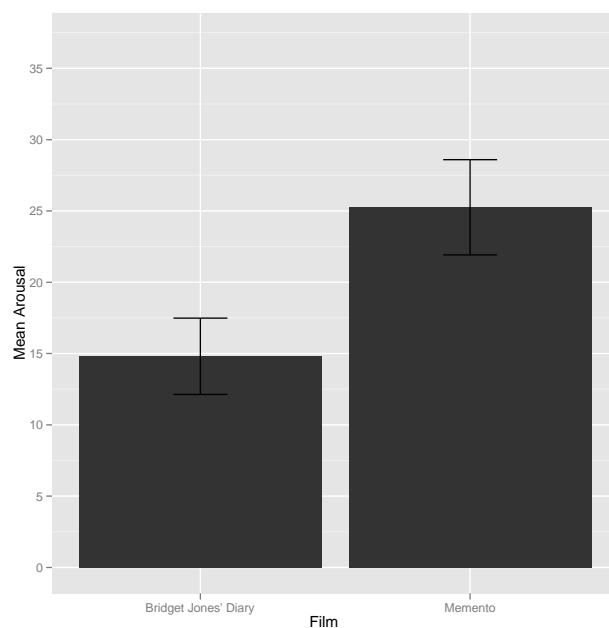



What this graph shows is that arousal levels were quite similar for men and women in general; this is why the main effect of gender was non-significant.

The main effect of film is shown by the F -ratio in the row labelled *film*; the probability associated with this F -ratio is much less than the critical value of .05, hence we can say that arousal levels were significantly different in the two films. Again, to interpret the effect we need to look at the mean arousal levels but this time comparing the two films (and ignoring whether the person was male or female).

```
bar <- ggplot(chickFlick, aes(film, arousal))

bar + stat_summary(fun.y = mean, geom = "bar", position="dodge") +
  stat_summary(fun.data = mean_cl_normal, geom = "errorbar",
    position=position_dodge(width=0.90), width = 0.2) + labs(x = "Film", y = "Mean Arousal")
```

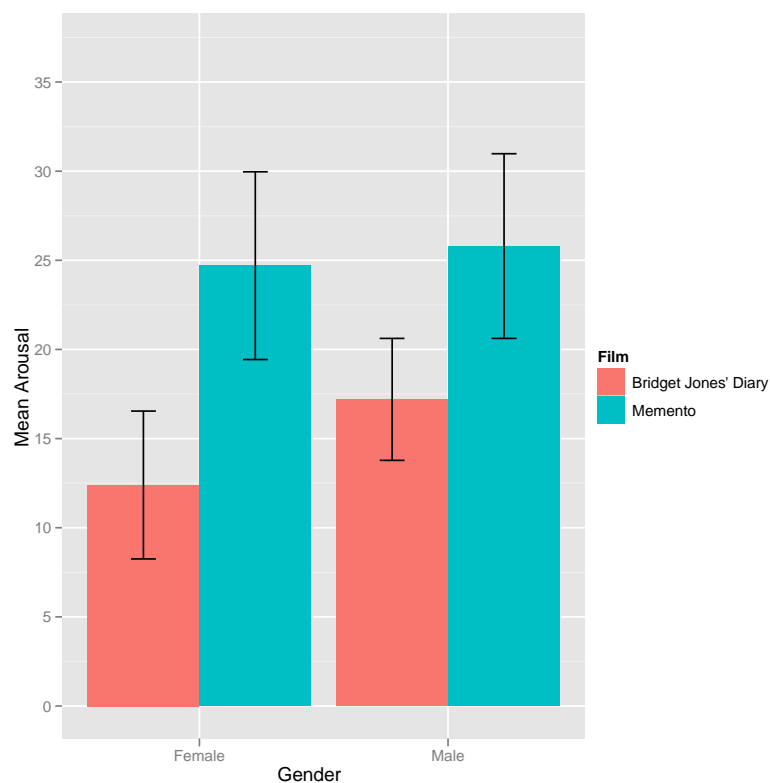


This graph shows that when you ignore the gender of the person, arousal levels were significantly higher for *Memento* than for *Bridget Jones's Diary*.

The interaction effect is shown by the *F*-ratio in the row labelled *gender:film*; the associated significance value is 0.366, which is greater than the criterion of .05. Therefore, we can say that there is not a significant interaction between gender and the type of film watched. To interpret this effect we need to look at the mean arousal in all conditions.

```
bar <- ggplot(chickFlick, aes(gender, arousal, fill = film))

bar + stat_summary(fun.y = mean, geom = "bar", position="dodge") +
  stat_summary(fun.data = mean_cl_normal, geom = "errorbar",
    position=position_dodge(width=0.90), width = 0.2) + labs(x = "Gender", y = "Mean
  Arousal", fill = "Film")
```



This graph shows the non-significant interaction: arousal levels are higher for *Memento* than for *Bridget Jones's Diary* in both men and women (i.e. the difference between the green and blue bars is more or less the same for men and women).

We could view the orthogonal contrasts that we set earlier by executing:

```
summary.lm(chickFlick)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	20.025	1.010	19.835	< 2e-16 ***
gender1	1.475	1.010	1.461	0.153
film1	5.225	1.010	5.175	8.78e-06 ***
gender1:film1	-0.925	1.010	-0.916	0.366

Looking at the output for the contrasts above we can see that it is the same as the ANOVA output. This is because gender and film both have only two levels.

Calculating effect sizes

If you have already created the function from Chapter 12 then you do not need to execute it again. If not, go back to Chapter 12 and execute the function.

Once you have executed the function, you can use it to calculate omega squared in the current data by using values of n (10 people per group), a (levels of gender = 2), b (levels of film = 2) and the four sums of squares from the ANOVA output above:

```
omega_factorial(10,2,2,87,1092,34.2,1467.7)

[1] "Omega-Squared A: 0.0169861022799528"
[1] "Omega-Squared B: 0.386244757864097"
[1] "Omega-Squared AB: -0.00241375544625899"
```

For the main effect of gender we get, $\omega_{\text{gender}}^2 = 0.02$; for the main effect of film we get, $\omega_{\text{film}}^2 = .39$; and for the interaction, $\omega_{\text{gender} \times \text{film}}^2 = -0.00$.

Interpreting and writing the result

We can report the three effects from this analysis as follows:

- The results show that the main effect of the type of film significantly affected arousal during that film, $F(1, 36) = 26.79$, $p < .001$, $\omega^2 = .39$; arousal levels were significantly higher during *Memento* compared to *Bridget Jones's Diary*.
- The main effect of gender on arousal levels during the films was non-significant, $F(1, 36) = 2.83$, $p > .05$, $\omega^2 = .02$.
- The gender \times film interaction was non-significant, $F(1, 36) = .84$, $p > .05$, $\omega^2 = -.00$. This showed that arousal levels were higher for *Memento* compared to *Bridget Jones's Diary* in both men and women.

Task 3

- At the start of this chapter I described a way of empirically researching whether I wrote better songs than my old band mate Malcolm, and whether this depended on the type of song (a symphony or song about flies). The outcome variable would be the number of screams elicited by audience members during the songs. These data are in the file **Escape From Inside.dat**. Draw an error bar graph (lines) and analyse and interpret these data.

Read in the data:

```
escapeData<-read.delim("Escape From Inside.dat", header = TRUE)
```

Next, set **Song Type** and **Songwriter** to be factors, which is necessary because R has loaded the data into the dataframe as numbers rather than text:

```
escapeData$Song_Type<-factor(escapeData$Song_Type, levels = c(0:1), labels =
c("Symphony", "Fly Song"))

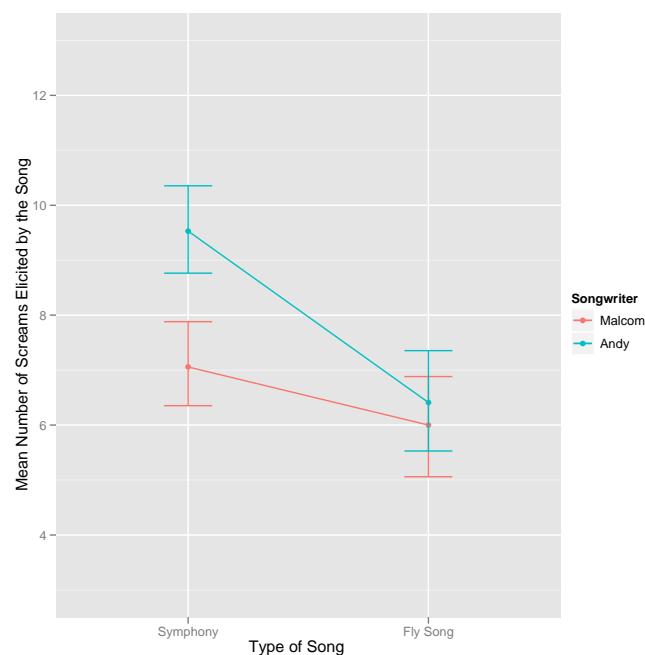
escapeData$Songwriter<-factor(escapeData$Songwriter, levels = c(0:1), labels =
c("Malcom", "Andy"))
```

To draw an error line graph, execute the following commands:

```
line <- ggplot(escapeData, aes(Song_Type, Screams, colour = Songwriter))

line + stat_summary(fun.y = mean, geom = "point") + stat_summary(fun.y = mean, geom =
"line", aes(group= Songwriter)) + stat_summary(fun.data = mean_cl_boot, geom =
"errorbar", width = 0.2) + labs(x = "Type of Song", y = "Mean Number of Screams
Elicited by the Song", colour = "Songwriter")
```

The resulting graph looks like this:



Next we can conduct Levene's test by executing the following command:

```
leveneTest(escapeData$Screams, interaction(escapeData$Song_Type,
  escapeData$Songwriter), center = median)
```

The following output shows that for these data, the significance value is 0.857, which is greater than the criterion of .05. This means that the variances in the different experimental groups are roughly equal (i.e. not significantly different), and that the assumption has been met.

```
Levene's Test for Homogeneity of Variance (center = median)
  Df F value Pr(>F)
group 3  0.2563 0.8566
  64
```

Next we can conduct the two-way independent ANOVA and set some orthogonal contrasts.

```
contrasts(escapeData$Song_Type)<-c(-1, 1)
contrasts(escapeData$Songwriter)<-c(1, -1)
escapeModel<-aov(Screams ~ Song_Type*Songwriter, data = escapeData)
Anova(escapeModel, type="III")
```

Anova Table (Type III tests)

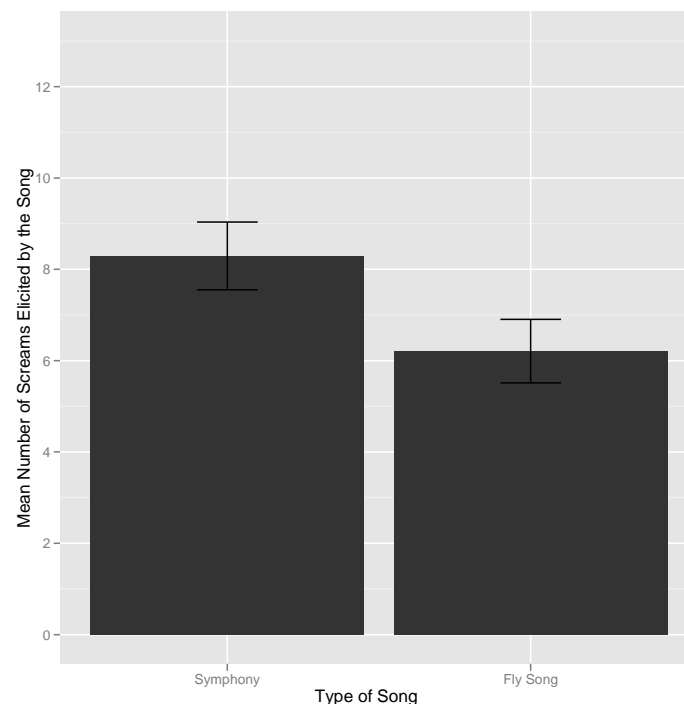
```
Response: Screams
          Sum Sq Df    F value    Pr(>F)
(Intercept) 3574.3  1 1006.4141 < 2.2e-16 ***
Song_Type    74.1  1  20.8737 2.293e-05 ***
Songwriter    35.3  1   9.9420 0.002460 **
Song_Type:Songwriter 18.0  1   5.0725 0.027747 *
```

In the ANOVA output above, the main effect of the type of song is shown by the F -ratio in the row labelled *Song_Type*; in this case the significance is much smaller than the usual cut-off point of .05. Hence, we can say that there was a significant effect of the type of song on the number of screams elicited while it was played. We can draw an error bar graph by executing the following commands:

```
bar <- ggplot(escapeData, aes(Song_Type, Screams))
```

```
bar + stat_summary(fun.y = mean, geom = "bar", position="dodge") +
stat_summary(fun.data = mean_cl_normal, geom = "errorbar",
position=position_dodge(width=0.90), width = 0.2) + labs(x = "Type of Song", y = "Mean
Number of Screams Elicited by the Song")
```

The resulting graph shows that the two symphonies elicited significantly more screams of agony than the two songs about flies.

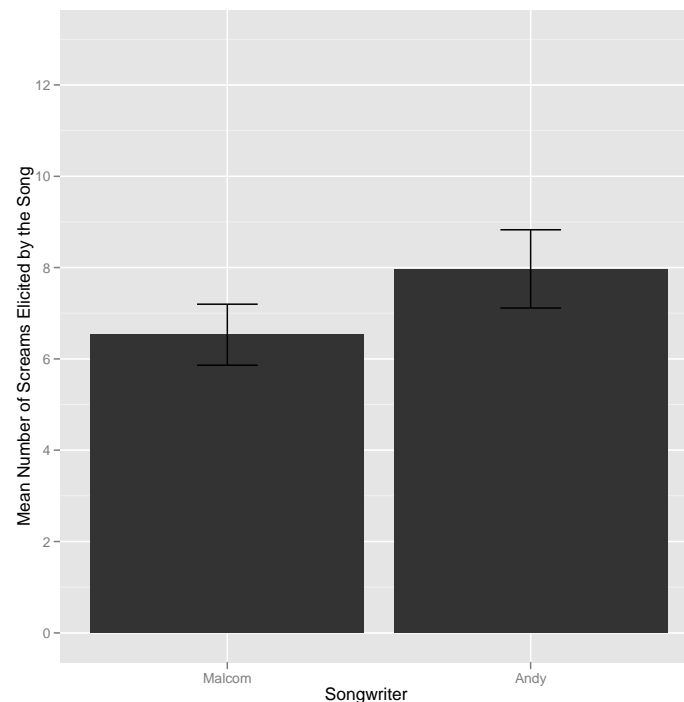


The main effect of the songwriter was significant because the significance of the F -ratio for this effect is .002, which is less than the critical value of .05, hence we can say that Andy and Malcolm differed in the reactions to their songs. We can draw an error bar graph by executing the following commands:

```
bar <- ggplot(escapeData, aes(Songwriter, Screams))
```

```
bar + stat_summary(fun.y = mean, geom = "bar", position="dodge") +
stat_summary(fun.data = mean_cl_normal, geom = "errorbar",
position=position_dodge(width=0.90), width = 0.2) + labs(x = "Songwriter", y = "Mean
Number of Screams Elicited by the Song")
```

The resulting graph tells us that Andy's songs elicited significantly more screams of torment from the audience than Malcolm's songs.



The interaction effect was significant too because the associated significance value is 0.028, which is less than the criterion of .05. Therefore, we can say that there is a significant interaction effect between the type of song and who wrote it on people's appreciation of the song. The line graph that you drew earlier on tells us that although reactions to Malcolm's and Andy's were fairly similar for the flies song, they differed quite a bit for the symphony: Andy's symphony elicited more screams of torment than Malcolm's. We can conclude that in general Malcolm was a better songwriter than Andy, but the interaction tells us that this effect is true mainly for symphonies.

Calculating effect sizes

Assuming that you have written and executed the function in Chapter 12, we can calculate omega squared in the current data by using values of n (17 people per group), a (levels of Song_Type = 2), b (levels of Songwriter = 2) and the four sums of squares from the ANOVA output above:

```
omega_factorial(17,2,2,74.1,35.3,18,227.3)

[1] "Omega-Squared A: 0.196924298131986"
[1] "Omega-Squared B: 0.0886205136928049"
[1] "Omega-Squared AB: 0.0403304242392523"
```

For the main effect of song type we get $\omega^2_{\text{Song_Type}} = .20$; for the main effect of songwriter we get $\omega^2_{\text{Songwriter}} = .09$; and for the interaction, $\omega^2_{\text{Song_Type} \times \text{Songwriter}} = .04$.

Interpreting and writing the result

We can, report the three effects from this analysis as follows:

- ✓ The results show that the main effect of the type of song significantly affected screams elicited during that song, $F(1, 64) = 20.87$, $p < .001$, $\omega^2 = .20$; the two symphonies elicited significantly more screams of agony than the two songs about flies.
- ✓ The main effect of the songwriter significantly affected screams elicited during that song, $F(1, 64) = 9.94$, $p < .01$, $\omega^2 = .09$; Andy's songs elicited significantly more screams of torment from the audience than Malcolm's songs.
- ✓ The song type \times songwriter interaction was significant, $F(1, 64) = 5.07$, $p < .05$, $\omega^2 = .04$. Although reactions to Malcolm's and Andy's were fairly similar for the flies song, they differed quite a bit for the symphony: Andy's symphony elicited more screams of torment than Malcolm's.

Task 4

- Using R's Souls' Tip 12.2, conduct a simple effects analysis of the effect of alcohol at different levels of gender (which is the opposite to the example in the chapter).

As always, the first thing to do is load in the data:

```
gogglesData<-read.csv("goggles.csv", header = TRUE)
```

Next we need to create a variable in the dataframe that merges the variables of interest into a single factor:

```
gogglesData$simple<-gl(6,8)
```

We can then use the *factor()* function to specify labels for these six groups:

```
gogglesData$simple<-factor(gogglesData$simple, levels = c(1:6), labels =  
c("Male_0", "Male_2", "Male_4", "Female_0", "Female_2", "Female_4"))
```

Next we create contrasts that break these six groups up using the standard rules for planned contrasts. We want to break the groups up into five contrasts to do simple effects analysis of **alcohol**. The first contrast compares men to women (*genderEffect1*). Remember that these two chunks of variation are made up of the different alcohol groups and so need to be broken down further. The second contrast (*MaleEffect1*) compares men who drank no alcohol to men who drank some alcohol (2 pints and 4 pints combined). The third contrast (*MaleEffect2*) breaks down the 'alcohol' chunk in men to compare 2 pints to 4 pints in men. The fourth contrast (*FemaleEffect1*) compares women who had no alcohol to women who drank some alcohol (2 pints and 4 pints combined). Finally, the fifth contrast (*FemaleEffect2*) breaks down the female 'alcohol' chunk to compare 2 pints to 4 pints in women.

```
genderEffect1<-c(1, 1, 1, -1, -1, -1)  
MaleEffect1<-c(0, 0, 0, -2, 1, 1)  
MaleEffect2<-c(0, 0, 0, 0, 1, -1)  
FemaleEffect1<-c(-2, 1, 1, 0, 0, 0)  
FemaleEffect2<-c(0, 1, -1, 0, 0, 0)  
simpleEff<-cbind(genderEffect1, MaleEffect1, MaleEffect2, FemaleEffect1,  
FemaleEffect2)
```

```
contrasts(gogglesData$simple)<-simpleEff  
simpleEffectModel<-aov(attractiveness ~ simple, data = gogglesData)  
summary.lm(simpleEffectModel)
```

```
Coefficients:  
              Estimate Std. Error t value Pr(>|t|)  
(Intercept)    58.3333     1.3153  44.351 < 2e-16 ***  
simplegenderEffect1    1.8750     1.3153   1.426 0.161382  
simpleMaleEffect1     -5.2083     1.3153  -3.960 0.000284 ***  
simpleMaleEffect2     15.6250     2.2781   6.859 2.31e-08 ***  
simpleFemaleEffect1    -0.2083     1.3153  -0.158 0.874903  
simpleFemaleEffect2     2.5000     2.2781   1.097 0.278716
```

MaleEffect1 is significant, indicating a significant effect of alcohol for men: men who consumed some alcohol (2 pints or 4 pints) chose significantly less attractive dates than men who consumed no alcohol. *MaleEffect2* was also significant, suggesting that the effect of drinking 4 pints is significantly different from drinking 2 pints in men. Think back to the chapter – this reflects the fact that men choose very unattractive dates after 4 pints. However, there is no significant effect of alcohol for females (*FemaleEffect1* and *FemaleEffect2* are both non significant). This tells us that women are not affected by the beer-goggles effect: the attractiveness of their dates does not change as they drink more.

Task 5

- Back in 2008, hospitals were reporting an increase in injuries related to playing Nintendo Wii (<http://www.telegraph.co.uk/news/uknews/1576244/Spate-of-injuries-blamed-on-Nintendo-Wii.html>). These injuries were attributed mainly to muscle and tendon strains. A researcher was interested to see whether these injuries could be prevented. She hypothesized that a stretching warm-up before playing Wii would help lower injuries, and that athletes would be less susceptible to injuries because their regular activity should make them more flexible. She took 60 athletes and 60 non-athletes (**athlete**), half of them played Wii and half watched others playing as a control (**wii**), and within these groups half did a 5-minute stretch routine before playing/watching whereas the other half did not (**stretch**). The outcome was a pain score out of 10 (where 0 is no pain, and 10 is severe pain) after playing for 4 hours (**injury**). The data are in the file **Wii.dat**. Conduct a three-way ANOVA to test whether athletes are less prone to injury, and whether the prevention programme worked.

First of all, load the data into a dataframe called *athleteData*:

```
athleteData<-read.delim("Wii.dat", header = TRUE)
```

Next, we need to test for homogeneity of variance using the Levene's test:

```
leveneTest(athleteData$injury, interaction(athleteData$athlete, athleteData$stretch,
athleteData$wii), center = median)
```

```
Levene's Test for Homogeneity of Variance (center = median)
      Df F value    Pr(>F)
group   7  2.1378 0.04523 *
112
```

The test is significant, $F(7, 112) = 2.14$, $p < .05$, suggesting that the assumption of homogeneity of variance has been violated. This means that we need to conduct a robust ANOVA on these data. I will first of all go through how you would conduct a normal ANOVA on these data, followed by how you would conduct a robust ANOVA.

To run the normal three-way ANOVA we would execute the following commands (remember you *must* set orthogonal contrasts if using Type III sums of squares):

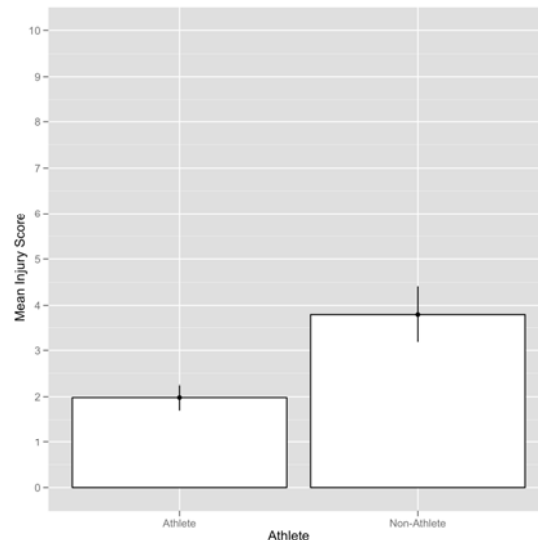
```
contrasts(athleteData$athlete)<-c(-1, 1)
contrasts(athleteData$stretch)<-c(-1, 1)
contrasts(athleteData$wii)<-c(-1, 1)

athleteModel<-aov(injury ~ athlete*stretch*wii, data = athleteData)
Anova(athleteModel, type="III")
```

```
Response: injury
              Sum Sq Df F value    Pr(>F)
(Intercept) 1003.41  1 656.9470 < 2.2e-16 ***
athlete      99.01   1  64.8223 9.595e-13 ***
stretch      16.87   1  11.0483 0.0012001 **
wii          85.01   1  55.6563 1.982e-11 ***
athlete:stretch  1.88   1   1.2276 0.2702496
athlete:wii     69.01   1  45.1808 7.869e-10 ***
stretch:wii     21.68   1  14.1910 0.0002651 ***
athlete:stretch:wii  9.08   1   5.9415 0.0163612 *
Residuals    171.07 112
```

The results show that there was a significant main effect of **athlete**, $F(1, 112) = 64.82$, $p < .001$. To help us interpret this significant effect we could plot an error bar chart by executing the following commands:

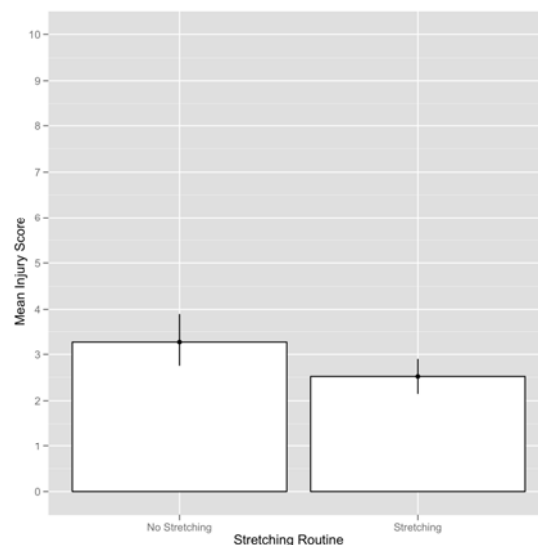
```
athleteME <- ggplot(athleteData, aes(athlete, injury))
athleteME + stat_summary(fun.y = mean, geom = "bar", fill = "White", colour = "Black")
+ stat_summary(fun.data = mean_cl_boot, geom = "pointrange") + labs(x = "Athlete", y =
"Mean Injury Score") + scale_y_continuous(breaks=seq(0,10, by = 1), limits = c(0, 10))
```

The resulting graph shows that on average, athletes had significantly lower injury scores than non-athletes.

There was also a significant main effect of **stretch**, $F(1, 112) = 11.05, p < .01$. We can plot an error bar graph to help us to interpret the effect of **stretch** by executing:

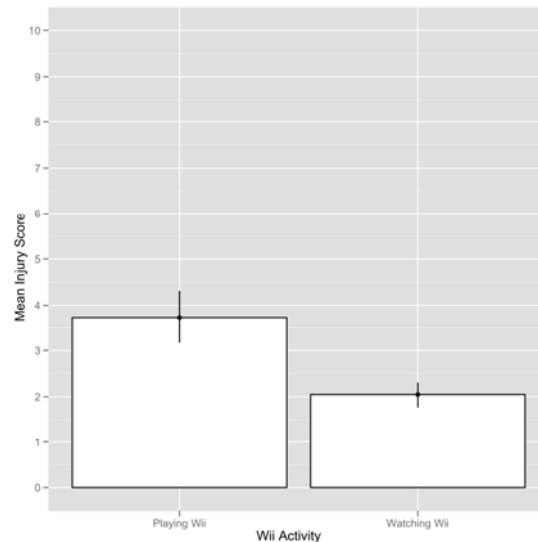
```
stretchME <- ggplot(athleteData, aes(stretch, injury))
stretchME + stat_summary(fun.y = mean, geom = "bar", fill = "White", colour = "Black")
+ stat_summary(fun.data = mean_cl_boot, geom = "pointrange") + labs(x = "Stretching
Routine", y = "Mean Injury Score") + scale_y_continuous(breaks=seq(0,10, by = 1),
limits = c(0, 10))
```



Looking at the error bar graph, we can see that stretching significantly decreased injury score compared to no stretching. However, the interaction graph (see below) tells us that this is true only for athletes and non-athletes who played on the Wii, not for those in the control group. This is an example of how main effects can sometimes be misleading.

There was also a significant main effect of **wii**, $F(1, 112) = 55.66, p < .001$. Lets again plot an error bar graph to help us to interpret the significant effect:

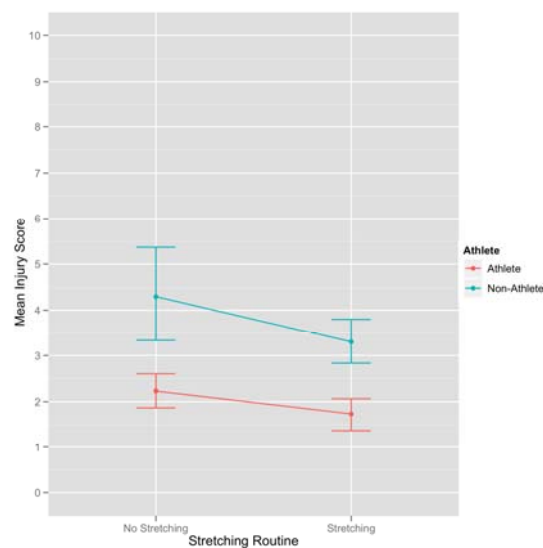
```
wiiME <- ggplot(athleteData, aes(wii, injury))
wiiME + stat_summary(fun.y = mean, geom = "bar", fill = "White", colour = "Black") +
stat_summary(fun.data = mean_cl_boot, geom = "pointrange") + labs(x = "Wii Activity",
y = "Mean Injury Score") + scale_y_continuous(breaks=seq(0,10, by = 1), limits = c(0,
10))
```



The resulting graph tells us (not surprisingly) that playing on the Wii resulted in a significantly higher injury score compared to watching other people playing on the Wii (control).

There was not a significant **athlete** × **stretch** interaction $F(1, 112) = 1.23, p > .05$. Let's plot an interaction graph to help us to interpret this non-significant effect:

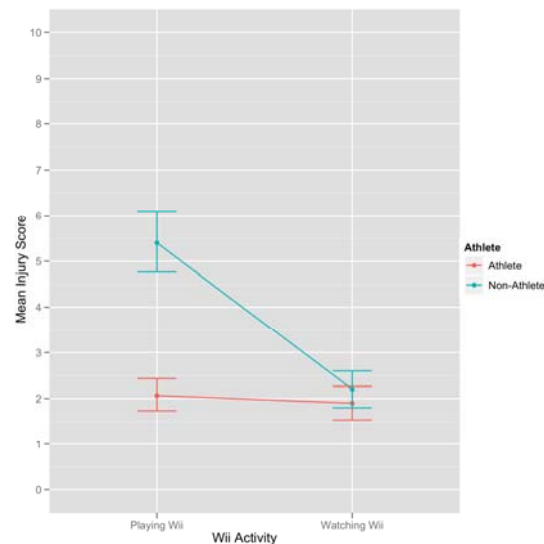
```
athletestretchInt <- ggplot(athleteData, aes(stretch, injury, colour = athlete))
athletestretchInt + stat_summary(fun.y = mean, geom = "point") + stat_summary(fun.y =
mean, geom = "line", aes(group= athlete)) + stat_summary(fun.data = mean_cl_boot, geom
= "errorbar", width = 0.2) + labs(x = "Stretching Routine", y = "Mean Injury Score",
colour = "Athlete") + scale_y_continuous(breaks=seq(0,10, by = 1), limits = c(0, 10))
```



The resulting graph shows that (not taking playing/watching the Wii into account) while non-athletes had higher injury scores than athletes overall, stretching decreased the number of injuries in both athletes and non-athletes by roughly the same amount. Parallel lines usually indicate a non-significant interaction effect and so it is not surprising, looking at this graph, that the interaction between stretch and athlete was non-significant.

There was a significant **athlete** × **wii** interaction, $F(1, 112) = 45.18, p < .001$. To plot an interaction graph, we can execute:

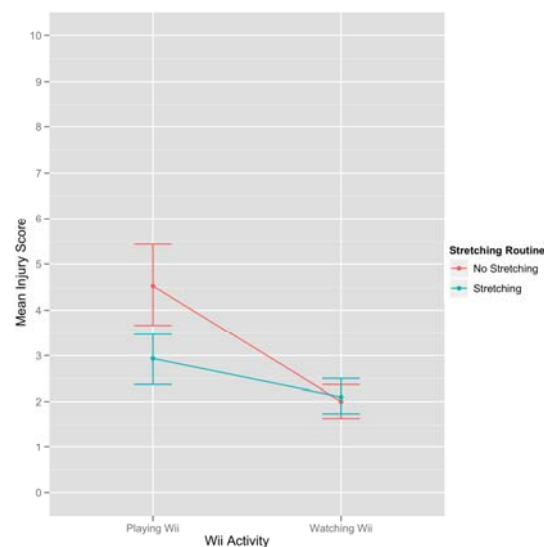
```
athleteWiiInt <- ggplot(athleteData, aes(wii, injury, colour = athlete))
athleteWiiInt + stat_summary(fun.y = mean, geom = "point") + stat_summary(fun.y =
mean, geom = "line", aes(group= athlete)) + stat_summary(fun.data = mean_cl_boot, geom
= "errorbar", width = 0.2) + labs(x = "Wii Activity", y = "Mean Injury Score", colour
= "Athlete") + scale_y_continuous(breaks=seq(0,10, by = 1), limits = c(0, 10))
```



The resulting graph shows that (not taking stretching into account) when playing on the Wii, non-athletes suffered significantly higher injury scores than athletes. However, when watching other people playing on the Wii, athletes and non-athletes had very similar injury scores.

There was a significant **stretch** \times **wii** interaction, $F(1, 112) = 14.19, p < .001$. To plot an interaction graph to help us to interpret this result, we can execute:

```
stretcheWiiInt <- ggplot(athleteData, aes(wii, injury, colour = stretch))
stretcheWiiInt + stat_summary(fun.y = mean, geom = "point") + stat_summary(fun.y =
mean, geom = "line", aes(group= stretch)) + stat_summary(fun.data = mean_cl_boot, geom
= "errorbar", width = 0.2) + labs(x = "Wii Activity", y = "Mean Injury Score", colour
= "Stretching Routine") + scale_y_continuous(breaks=seq(0,10, by = 1), limits = c(0,
10))
```

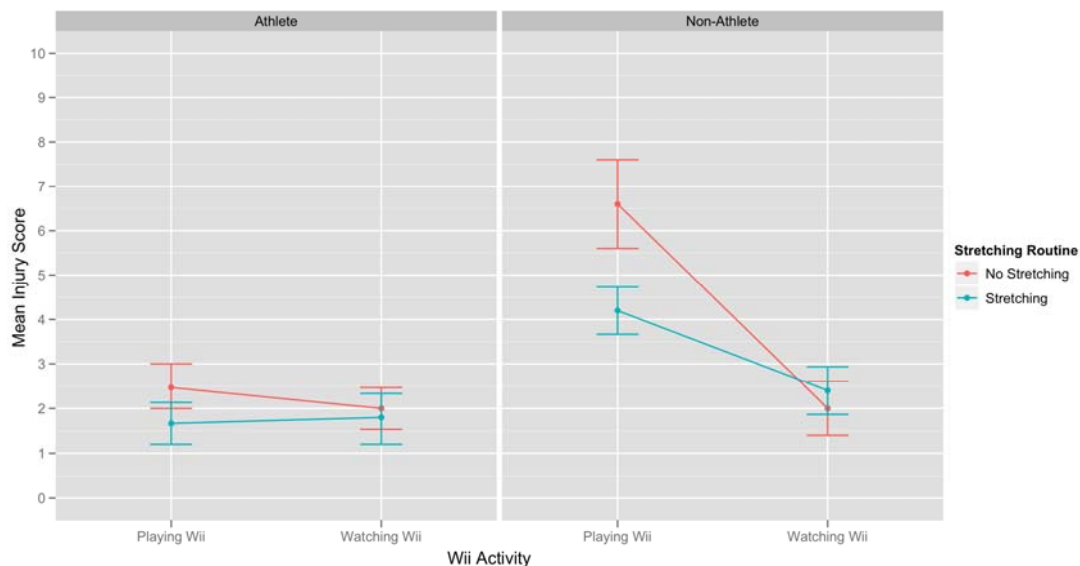


The resulting graph shows that (not taking athlete into account) stretching before playing on the Wii significantly decreased injury scores, but stretching before watching other people playing on the Wii did not significantly reduce injury scores. This is not surprising as watching other people playing on the Wii is unlikely to result in sports injury!

There was a significant **athlete** \times **stretch** \times **wii** interaction, $F(1, 112) = 5.94, p < .05$. What this actually means is that the effect of stretching and playing on the Wii on injury score was different for athletes than it was for non-athletes. In the presence of this significant interaction it makes no sense to interpret the main effects. Let's plot an interaction graph for this effect by executing:

```
threeWayInt <- ggplot(athleteData, aes(wii, injury, colour = stretch))
```

```
threeWayInt + stat_summary(fun.y = mean, geom = "point") + stat_summary(fun.y = mean,
geom = "line", aes(group= stretch)) + stat_summary(fun.data = mean_cl_boot, geom =
"errorbar", width = 0.2) + labs(x = "Wii Activity", y = "Mean Injury Score", colour =
"Stretching Routine") + facet_wrap(~athlete) + scale_y_continuous(breaks=seq(0,10, by
= 1), limits = c(0, 10))
```



The resulting graph clearly shows that for athletes, stretching and playing on the Wii has very little effect: their mean injury score is quite stable across the two conditions (whether they played on the Wii or watched other people playing on the Wii, stretched or did no stretching). However, for the non-athletes, watching other people play on the Wii compared to not stretching and playing on the Wii rapidly declines their mean injury score. The interaction tells us that stretching and watching rather than playing on the Wii both result in a lower injury score and that this is true only for non-athletes. In short, the results show that athletes are able to minimize their injury level regardless of whether they stretch before exercise or not, whereas non-athletes only have to bend slightly and they get injured! Although I wonder if we would get the same results using the Arsenal football team ...☺

To view the output of the contrasts that we specified, execute:

```
summary.lm(athleteModel)
```

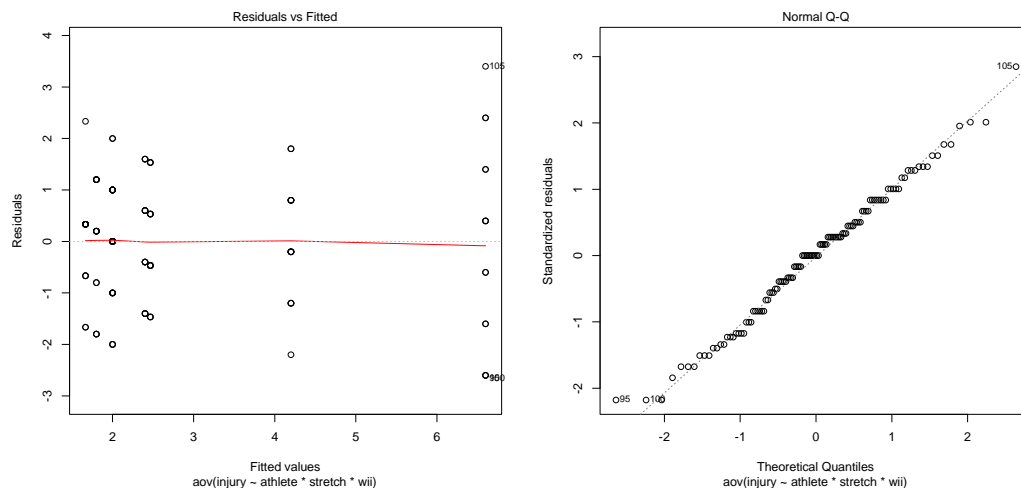
Doing so will display the parameter estimates for the model:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	2.8917	0.1128	25.631	< 2e-16	***
athlete1	0.9083	0.1128	8.051	9.60e-13	***
stretch1	-0.3750	0.1128	-3.324	0.001200	**
wii1	-0.8417	0.1128	-7.460	1.98e-11	***
athlete1:stretch1	-0.1250	0.1128	-1.108	0.270250	
athlete1:wii1	-0.7583	0.1128	-6.722	7.87e-10	***
stretch1:wii1	0.4250	0.1128	3.767	0.000265	***
athlete1:stretch1:wii1	0.2750	0.1128	2.438	0.016361	*

However, because **athlete**, **stretch** and **wii** all have only two groups, all these effects are the same as in the ANOVA output above!

To view the plots that we can use to test the assumptions of ANOVA, we can execute:

```
plot(athleteModel)
```



The graph on the left can be used for testing homogeneity of variance (which we already know has been violated): if it has a funnel shape then we're in trouble. The plot we have does show funnelling, which implies that the residuals might be heteroscedastic. The plot on the right is a Q-Q plot, which tells us about the normality of residuals in the model. We want our residuals to be normally distributed, which means that the dots on the graph should hover around the diagonal line. In ours, this is the case suggesting that we can assume normality of our residuals/errors.

You may remember that, earlier on in the analysis, we obtained a significant result for Levene's test, therefore in reality we would need to conduct a robust analysis on these data. To do this we need to first transform our data into the wide format rather than long. Essentially, we want levels of our three factors to be represented in different columns. Therefore, rather than a dataframe with 4 columns and 120 rows, we want one with 8 columns and 15 rows.

We are going to use `melt()` and `cast()` to do the restructuring for us. To get the restructuring to work, we need to add a new variable to our dataframe that identifies the rows in the wide format. It will be a value from 1 to 15 telling us whether the score is in the first, second, third, etc. score within the chunk. At the moment the chunks are stacked on top of each other, so we want a variable that is the sequence of numbers 1 to 15 repeated for each of the eight chunks. We can add this variable (which I will call **row**) by executing:

```
athleteData$row<-rep(1:15, 8)
```

Now that we have changed the data set we need to make it molten so that we can cast the data into the wide format. To do this we use the `melt()` function. Remember that in this function we differentiate variables that identify attributes of the scores (in this case, **athlete**, **stretch**, **wii** and **row** all tell us about a given injury score) from the scores or measured variables themselves. Attributes are specified with the *id* option, and scores with the *measured* option. Therefore, we can create a molten dataframe called *athleteMelt* by executing:

```
athleteMelt<-melt(athleteData, id = c("row", "athlete", "stretch", "wii"), measured =  
c( "injury" ))
```

Having melted the data, we want to cast it in the wide format using `cast()`. To do this we use a formula in the form: *variables specifying the rows ~ variables specifying the columns*. In this case, **row** tells us which row to place a score, and we want the **athlete**, **stretch** and **wii** variables split across different columns, so we'd use the formula: *row ~ athlete + stretch + wii*. Therefore, we can make a wide dataframe called *athleteWide* by executing:

```
athleteWide<-cast(athleteMelt, row ~ athlete + stretch + wii)
```

We have now transformed our data into the wide format! We can now remove the variable **row** (as we no longer need it) by executing:

```
athleteWide$row<-NULL
```

If you look at the dataframe you'll see a lovely wide format set of data:

```
athleteWide
```

It is important to view the data set and note the order of the columns because this affects how we specify the robust analysis (well ... not really in this case because all three variables have two levels and so it doesn't really matter). In this case the hierarchy of the independent variables is **athlete**, followed by **stretching**, followed by **wii**. We would say that **athlete** is factor A, **stretching** is factor B and **wii** is factor C.

To perform a three-way independent ANOVA on trimmed means, we can adapt the function `t2way()` by replacing 2 with 3 so it becomes `t3way()`. The function `t3way()` will take the general form (assuming we are happy with the default level of trimming):

```
t3way(levels of factor A, levels of factor B, levels of factor C, dataframe)
```

Therefore, we can do a robust three-way factorial ANOVA based on trimmed means by executing:

```
t3way(2,2,2, athleteWide)
```

```
$Qa
[1] 44.17362

$Qa.crit
[1] 4.037312

$A.p.value
[1] 1e-04

$Qb
[1] 6.956038

$Qb.crit
[1] 4.037312

$B.p.value
[1] 0.012

$Qc
[1] 38.74235

$Qc.crit
[1] 12.17617

$C.p.value
[1] 0.001

$Qab
[1] 1.880913

$Qab.crit
[1] 1.876811

$AB.p.value
[1] 0.177

$Qac
[1] 33.66722

$Qac.crit
[1] 12.17617

$AC.p.value
[1] 0.001

$Qbc
[1] 9.360045

$Qbc.crit
[1] 9.093685

$BC.p.value
[1] 0.004

$Qabc
[1] 3.216472

$Qabc.crit
[1] 3.196099
```

```
$ABC.p.value
[1] 0.08
```

In the output above, we are given a test statistic for factor A (Q_a), factor B (Q_b), factor C (Q_c), the interaction between factors A and B (Q_{ab}), the interaction between factors A and C (Q_{ac}), the interaction between factors B and C (Q_{bc}) and the three way interaction (Q_{abc}) as well as their corresponding p -value ($A.p.value$, $B.p.value$, $C.p.value$, $AB.p.value$, $AC.p.value$, $BC.p.value$ and $ABC.p.value$, respectively). Remember that factor A was **athlete**, factor B was **stretch**, factor C was **wii**. Therefore, we could conclude that there was a significant main effect of **athlete**, $Q = 44.17$, $p < .001$, a significant main effect of **stretch**, $Q = 6.96$, $p < .05$, a significant main effect of **wii**, $Q = 38.74$, $p < .01$, a non-significant **athlete** \times **stretch** interaction, $Q = 1.88$, $p = .18$, a significant **athlete** \times **wii** interaction effect, $Q = 33.67$, $p < .01$, a significant **stretch** \times **wii** interaction, $Q = 9.36$, $p < .01$, and a non-significant three-way interaction between **athlete** \times **stretch** \times **wii**, $Q = 3.22$, $p = .08$.