Mixed designs

Self-test answers

- Using what you learnt earlier in the chapter and the commands that we have just used to create **looks** and **personality**, can you work out how to enter the data into **R** directly?

If we wanted to enter the data directly into **R**, we would first need to create the variable that identifies participants by using the *gl()* function (Chapter 3). Remember that this function takes the general form:

```
factor<-gl(number of levels, cases in each level, total cases, labels = c("label1", "label2"…))
```

This function creates a factor variable called *factor*; you specify the number of levels or groups of the factor, how many cases are in each level/group, optionally the total number of cases (the default is to multiply the number of groups by the number of cases per group), and you can also use the *labels* option to list names for each level/group. For **participant**, we want nine scores for each of the 20 participants, so we can specify it as:

```
participant<-gl(20, 9, labels = c("P01", "P02", "P03", "P04", "P05", "P06", "P07", "P08", "P09", "P10", "P11", "P12", "P13", "P14", "P15", "P16", "P17", "P18", "P19", "P20" ))
```

The numbers in the function tell **R** that we had 20 sets of nine scores, the labels option then specifies the names to attach to these 20 sets, which correspond to their participant number. A quicker way to do this is to use the *paste()* function to create the labels for you. If we execute this command instead:

```
participant<-gl(20, 9, labels = c(paste("P", 1:20, sep = "_")))
```

The *paste()* function takes the things in brackets and pastes them together, the sep option specifies how to separate the bits that have been pasted together. So the *"P"* means that we begin with the letter P and then we paste a number after it separated by an underscore. The *1:20* creates a sequence of numbers from 1 to 20. Therefore, we create a sequence of text strings that are P then an underscore then a number, where the number starts at 1 and goes to 20. Therefore, we'll get a sequence of strings P_1, P_2, P_3, …, P_20. To see for yourself, just execute the *paste()* function as we have specified it above:

```
paste("P", 1:20, sep = "_")
```

The resulting sequence is:

```
"P_1"  "P_2"  "P_3"  "P_4"  "P_5"  "P_6"  "P_7"  "P_8"  "P_9"  "P_10" "P_11" "P_12"
"P_13" "P_14" "P_15" "P_16" "P_17" "P_18" "P_19" "P_20"
```

Therefore, by placing this paste command within the *gl()* function we automatically generate the labels for each person, which when you have a lot of participants is quicker then typing them all in.

To create the **gender** variable, we need to create two sets of scores that contain 90 rows each (because there are 10 males × 9 scores = 90 rows, followed by 10 females also × 9 scores each = 90 rows). Therefore, we execute this command:

```
gender<-gl(2, 90, labels = c("Male", "Female"))
```

The numbers in the function tell **R** that we had two sets of 90 scores, the labels option then specifies the names to attach to these 2 sets.

To create the **personality** variable we follow the same procedure as in the chapter. We currently have nine rows per person that we need to identify based on levels of **personality** and **looks**. Within each person, for each of the three levels of charisma (high, average and dull) there are three scores (one each for the attractive, avarege and ugly dates). Therefore, we want three groups that each contain three scores. This will create the codes within a person, and we need these codes to be repeated for all 20 people, and to do this we include the total number of cases (20 cases × 9 scores per case = 180 scores). Including this information in the *gl()* function we would execute:

```
personality<-gl(3, 3, 180, labels = c("Charismatic", "Average", "Dullard"))
```

This command creates a variable **personality**; the numbers in the function tell **R** that we had three sets of three scores, the labels option then specifies the names to attach to these three sets, which correspond to the levels of charisma. The *180* tells R to repeat this sequence for 180 cases. Essentially, this will create three rows with the label *Charismatic* then three labelled *Average*, then three labelled *Dullard*, and then repeats this sequence for 180 cases.

We also need a variable that tells us how attractive the date was. To do this we want three sets of one score (attractive, average, ugly). This will create three cases, or, put another way, it will create the codes for the first level (charismatic) of the **personality** variable. We want this pattern to be repeated for the remaining two levels of **personality** (i.e., average and dullard). We can do this by adding a third value to the function that is the total number of cases (i.e., 180). By specifying the total number of cases the *gl()* function will repeat the pattern of codes until it reaches this total number of cases

```
looks<-gl(3, 1, 180, labels = c("Attractive", "Average", "Ugly"))
```

We can add the ratings of the dates by creating a numeric variable in the usual way:

```
dateRating<-c(86, 84, 67, 88, 69, 50, 97, 48, 47, 91, 83, 53, 83, 74, 48, 86, 50, 46,
89, 88, 48, 99, 70, 48, 90, 45, 48, 89, 69, 58, 86, 77, 40, 87, 47, 53, 80, 81, 57, 88,
71, 50, 82, 50, 45, 80, 84, 51, 96, 63, 42, 92, 48, 43, 89, 85, 61, 87, 79, 44, 86, 50,
45, 100, 94, 56, 86, 71, 54, 84, 54, 47, 90, 74, 54, 92, 71, 58, 78, 38, 45, 89, 86,
63, 80, 73, 49, 91, 48, 39, 89, 91, 93, 88, 65, 54, 55, 48, 52, 84, 90, 85, 95, 70, 60,
50, 44, 45, 99, 100, 89, 80, 79, 53, 51, 48, 44, 86, 89, 83, 86, 74, 58, 52, 48, 47,
89, 87, 80, 83, 74, 43, 58, 50, 48, 80, 81, 79, 86, 59, 47, 51, 47, 40, 82, 92, 85, 81,
66, 47, 50, 45, 47, 97, 69, 87, 95, 72, 51, 45, 48, 46, 95, 92, 90, 98, 64, 53, 54, 53,
45, 95, 93, 96, 79, 66, 46, 52, 39, 47)
```

Finally, we can merge these variables into a dataframe called *speedData* by executing:

```
speedData<-data.frame(participant, gender, personality, looks, dateRating)
```

The data should look like this:

```
    participant gender personality      looks dateRating
1           P01   Male Charismatic Attractive         86
2           P01   Male Charismatic    Average         84
3           P01   Male Charismatic       Ugly         67
4           P01   Male     Average Attractive         88
5           P01   Male     Average    Average         69
6           P01   Male     Average       Ugly         50
7           P01   Male     Dullard Attractive         97
8           P01   Male     Dullard    Average         48
9           P01   Male     Dullard       Ugly         47
10          P02   Male Charismatic Attractive         91
11          P02   Male Charismatic    Average         83
12          P02   Male Charismatic       Ugly         53
13          P02   Male     Average Attractive         83
14          P02   Male     Average    Average         74
15          P02   Male     Average       Ugly         48
16          P02   Male     Dullard Attractive         86
17          P02   Male     Dullard    Average         50
18          P02   Male     Dullard       Ugly         46
19          P03   Male Charismatic Attractive         89
20          P03   Male Charismatic    Average         88
21          P03   Male Charismatic       Ugly         48
22          P03   Male     Average Attractive         99
23          P03   Male     Average    Average         70
24          P03   Male     Average       Ugly         48
25          P03   Male     Dullard Attractive         90
26          P03   Male     Dullard    Average         45
27          P03   Male     Dullard       Ugly         48
28          P04   Male Charismatic Attractive         89
29          P04   Male Charismatic    Average         69
30          P04   Male Charismatic       Ugly         58
31          P04   Male     Average Attractive         86
32          P04   Male     Average    Average         77
```

33	P04	Male	Average	Ugly	40
34	P04	Male	Dullard	Attractive	87
35	P04	Male	Dullard	Average	47
36	P04	Male	Dullard	Ugly	53
37	P05	Male	Charismatic	Attractive	80
38	P05	Male	Charismatic	Average	81
39	P05	Male	Charismatic	Ugly	57
40	P05	Male	Average	Attractive	88
41	P05	Male	Average	Average	71
42	P05	Male	Average	Ugly	50
43	P05	Male	Dullard	Attractive	82
44	P05	Male	Dullard	Average	50
45	P05	Male	Dullard	Ugly	45
46	P06	Male	Charismatic	Attractive	80
47	P06	Male	Charismatic	Average	84
48	P06	Male	Charismatic	Ugly	51
49	P06	Male	Average	Attractive	96
50	P06	Male	Average	Average	63
51	P06	Male	Average	Ugly	42
52	P06	Male	Dullard	Attractive	92
53	P06	Male	Dullard	Average	48
54	P06	Male	Dullard	Ugly	43
55	P07	Male	Charismatic	Attractive	89
56	P07	Male	Charismatic	Average	85
57	P07	Male	Charismatic	Ugly	61
58	P07	Male	Average	Attractive	87
59	P07	Male	Average	Average	79
60	P07	Male	Average	Ugly	44
61	P07	Male	Dullard	Attractive	86
62	P07	Male	Dullard	Average	50
63	P07	Male	Dullard	Ugly	45
64	P08	Male	Charismatic	Attractive	100
65	P08	Male	Charismatic	Average	94
66	P08	Male	Charismatic	Ugly	56
67	P08	Male	Average	Attractive	86
68	P08	Male	Average	Average	71
69	P08	Male	Average	Ugly	54
70	P08	Male	Dullard	Attractive	84
71	P08	Male	Dullard	Average	54
72	P08	Male	Dullard	Ugly	47
73	P09	Male	Charismatic	Attractive	90
74	P09	Male	Charismatic	Average	74
75	P09	Male	Charismatic	Ugly	54
76	P09	Male	Average	Attractive	92
77	P09	Male	Average	Average	71
78	P09	Male	Average	Ugly	58
79	P09	Male	Dullard	Attractive	78
80	P09	Male	Dullard	Average	38
81	P09	Male	Dullard	Ugly	45
82	P10	Male	Charismatic	Attractive	89
83	P10	Male	Charismatic	Average	86
84	P10	Male	Charismatic	Ugly	63
85	P10	Male	Average	Attractive	80
86	P10	Male	Average	Average	73
87	P10	Male	Average	Ugly	49
88	P10	Male	Dullard	Attractive	91
89	P10	Male	Dullard	Average	48
90	P10	Male	Dullard	Ugly	39
91	P11	Female	Charismatic	Attractive	89
92	P11	Female	Charismatic	Average	91
93	P11	Female	Charismatic	Ugly	93
94	P11	Female	Average	Attractive	88
95	P11	Female	Average	Average	65
96	P11	Female	Average	Ugly	54
97	P11	Female	Dullard	Attractive	55
98	P11	Female	Dullard	Average	48
99	P11	Female	Dullard	Ugly	52
100	P12	Female	Charismatic	Attractive	84
101	P12	Female	Charismatic	Average	90
102	P12	Female	Charismatic	Ugly	85
103	P12	Female	Average	Attractive	95
104	P12	Female	Average	Average	70
105	P12	Female	Average	Ugly	60
106	P12	Female	Dullard	Attractive	50
107	P12	Female	Dullard	Average	44
108	P12	Female	Dullard	Ugly	45
109	P13	Female	Charismatic	Attractive	99
110	P13	Female	Charismatic	Average	100
111	P13	Female	Charismatic	Ugly	89

```
112         P13 Female     Average Attractive         80
113         P13 Female     Average    Average         79
114         P13 Female     Average       Ugly         53
115         P13 Female     Dullard Attractive         51
116         P13 Female     Dullard    Average         48
117         P13 Female     Dullard       Ugly         44
118         P14 Female Charismatic Attractive         86
119         P14 Female Charismatic    Average         89
120         P14 Female Charismatic       Ugly         83
121         P14 Female     Average Attractive         86
122         P14 Female     Average    Average         74
123         P14 Female     Average       Ugly         58
124         P14 Female     Dullard Attractive         52
125         P14 Female     Dullard    Average         48
126         P14 Female     Dullard       Ugly         47
127         P15 Female Charismatic Attractive         89
128         P15 Female Charismatic    Average         87
129         P15 Female Charismatic       Ugly         80
130         P15 Female     Average Attractive         83
131         P15 Female     Average    Average         74
132         P15 Female     Average       Ugly         43
133         P15 Female     Dullard Attractive         58
134         P15 Female     Dullard    Average         50
135         P15 Female     Dullard       Ugly         48
136         P16 Female Charismatic Attractive         80
137         P16 Female Charismatic    Average         81
138         P16 Female Charismatic       Ugly         79
139         P16 Female     Average Attractive         86
140         P16 Female     Average    Average         59
141         P16 Female     Average       Ugly         47
142         P16 Female     Dullard Attractive         51
143         P16 Female     Dullard    Average         47
144         P16 Female     Dullard       Ugly         40
145         P17 Female Charismatic Attractive         82
146         P17 Female Charismatic    Average         92
147         P17 Female Charismatic       Ugly         85
148         P17 Female     Average Attractive         81
149         P17 Female     Average    Average         66
150         P17 Female     Average       Ugly         47
151         P17 Female     Dullard Attractive         50
152         P17 Female     Dullard    Average         45
153         P17 Female     Dullard       Ugly         47
154         P18 Female Charismatic Attractive         97
155         P18 Female Charismatic    Average         69
156         P18 Female Charismatic       Ugly         87
157         P18 Female     Average Attractive         95
158         P18 Female     Average    Average         72
159         P18 Female     Average       Ugly         51
160         P18 Female     Dullard Attractive         45
161         P18 Female     Dullard    Average         48
162         P18 Female     Dullard       Ugly         46
163         P19 Female Charismatic Attractive         95
164         P19 Female Charismatic    Average         92
165         P19 Female Charismatic       Ugly         90
166         P19 Female     Average Attractive         98
167         P19 Female     Average    Average         64
168         P19 Female     Average       Ugly         53
169         P19 Female     Dullard Attractive         54
170         P19 Female     Dullard    Average         53
171         P19 Female     Dullard       Ugly         45
172         P20 Female Charismatic Attractive         95
173         P20 Female Charismatic    Average         93
174         P20 Female Charismatic       Ugly         96
175         P20 Female     Average Attractive         79
176         P20 Female     Average    Average         66
177         P20 Female     Average       Ugly         46
178         P20 Female     Dullard Attractive         52
179         P20 Female     Dullard    Average         39
180         P20 Female     Dullard       Ugly         47
```

- Use *ggplot2* to plot boxplots of the rating of the dates according to their level of attractiveness (*x*-axis), and level of charisma (different colours) for men and women (different plots).

```
dateBoxplot <- ggplot(speedData, aes(looks, dateRating, colour = personality))
```

```
dateBoxplot + geom_boxplot() + labs(x = "Attractiveness", y = "Mean Rating of Date",
colour = "Charisma") + facet_wrap(~gender)
```

- What is the difference between a main effect and an interaction?

A main effect is the unique effect of a *predictor variable* (or *independent variable*) on an *outcome variable*. In this context it can be the effect of gender, charisma or looks on their own. So, in the case of gender, the main effect is the difference in the average score from men (irrespective of the type of date they were rating) to that of all women (irrespective of the type of date that they are rating). The main effect of looks would be the mean rating given to all attractive dates (irrespective of their charisma, or whether they were rated by a man or a woman), compared to the average rating given to all average-looking dates (irrespective of their charisma, or whether they were rated by a man or a woman) and the average rating of all ugly dates (irrespective of their charisma, or whether they were rated by a man or a woman). An interaction, on the other hand, looks at the *combined* effect of two or more variables: for example, were the average ratings of attractive, ugly and average-looking dates different in men and women?

- Using *ggplot2* and *stat.desc*, plot an error bar graph and get the means for the main effect of **gender**.

Descriptives:

```
by(speedData$dateRating, speedData$gender, stat.desc, basic = FALSE)
```

Graph:

```
genderBar <- ggplot(speedData, aes(gender, dateRating))

genderBar + stat_summary(fun.y = mean, geom = "bar", fill = "White", colour = "Black")
+ stat_summary(fun.data = mean_cl_boot, geom = "pointrange") + labs(x = "Gender", y =
"Mean Rating of Date")
```

- Using *ggplot2* and *stat.desc*, plot an error bar graph and get the means for the main effect of **looks**.

Descriptives:

```
by(speedData$dateRating, speedData$looks, stat.desc, basic = FALSE)
```

Graph:

```
looksBar <- ggplot(speedData, aes(looks, dateRating))

looksBar + stat_summary(fun.y = mean, geom = "bar", fill = "White", colour = "Black")
+ stat_summary(fun.data = mean_cl_boot, geom = "pointrange") + labs(x =
"Attractiveness", y = "Mean Rating of Date")
```

- Using *ggplot2* and *stat.desc*, plot an error bar graph and get the means for the main effect of **personality**.

Descriptives:

```
by(speedData$dateRating, speedData$personality, stat.desc, basic = FALSE)
```

Graph:

```
charismaBar <- ggplot(speedData, aes(personality, dateRating))
```

```
charismaBar + stat_summary(fun.y = mean, geom = "bar", fill = "White", colour =
"Black") + stat_summary(fun.data = mean_cl_boot, geom = "pointrange") + labs(x =
"Charisma", y = "Mean Rating of Date")
```

- Using *ggplot2* and *stat.desc*, plot a line graph and get the means for the **looks** × **gender** interaction.

Descriptives:

```
by(speedData$dateRating, list(speedData$looks, speedData$gender), stat.desc, basic =
FALSE)
```

Graph:

```
genderLooks <- ggplot(speedData, aes(looks, dateRating, colour = gender))

genderLooks + 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 = "Attractiveness", y = "Mean Rating of Date",
colour = "Gender") + scale_y_continuous(limits = c(0,100))
```

- Using *ggplot2* and *stat.desc*, plot a line graph and get the means for the **personality** × **gender** interaction.

Descriptives:

```
by(speedData$dateRating, list(speedData$personality, speedData$gender), stat.desc,
basic = FALSE)
```

Graph:

```
genderCharisma <- ggplot(speedData, aes(personality, dateRating, colour = gender))

genderCharisma + 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 = "Charisma", y = "Mean Rating of Date", colour =
"Gender") + scale_y_continuous(limits = c(0,100))
```

- Using *ggplot2* and *stat.desc*, plot a line graph and get the means for the **looks** × **personality** interaction.

Descriptives:

```
by(speedData$dateRating, list(speedData$looks, speedData$personality), stat.desc,
basic = FALSE)
```

Graph:

```
looksCharisma <- ggplot(speedData, aes(looks, dateRating, colour = personality))

looksCharisma + stat_summary(fun.y = mean, geom = "point") + stat_summary(fun.y = mean,
geom = "line", aes(group= personality)) + stat_summary(fun.data = mean_cl_boot, geom =
"errorbar", width = 0.2) + labs(x = "Charisma", y = "Mean Rating of Date", colour =
"Charisma") + scale_y_continuous(limits = c(0,100))
```

- Using *ggplot2* and *stat.desc*, plot a line graph and get the means for the **looks** × **personality** × **gender** interaction.

Descriptives:

```
by(speedData$dateRating, list(speedData$looks, speedData$personality, speedData$gender), stat.desc, basic = FALSE)
```

Graph:

```
looksCharismaGender <- ggplot(speedData, aes(looks, dateRating, colour = personality))
```

```
looksCharismaGender + stat_summary(fun.y = mean, geom = "point") + stat_summary(fun.y = mean, geom = "line", aes(group= personality)) + stat_summary(fun.data = mean_cl_boot, geom = "errorbar", width = 0.2) + labs(x = "Attractiveness", y = "Mean Rating of Date", colour = "Charisma") + scale_y_continuous(limits = c(0,100)) + facet_wrap(~gender)
```

- Using *ggplot2* and *stat.desc*, plot a line graph and get the means for the relationship status × profile picture interaction.

The data are curently in this format (I've edited out some cases):

```
   case relationship_status couple alone
1     1   In a Relationship      4     4
2     2   In a Relationship      4     6
3     3   In a Relationship      4     7
4     4   In a Relationship      3     5
…     …          …               …     …
36   36              Single      5    10
37   37              Single      4     8
38   38              Single      6     9
39   39              Single      7    10
40   40              Single      3     5
```

First let's rename the variables **couple** and **alone** so that we'll get nicer legends on the graph (i.e. capital letters, and perhaps 'With Man' is a better description than 'couple'). Execute this command to change the names:

```
names(pictureData)<-c("case", "relationship_status", "With Man", "Alone")
```

We need the data to be in long format. We can do this restructuring using the *melt()* function from the *reshape* package. Remember that in this function we differentiate variables that identify attributes of the scores (in this case, **case** and **relationship_status** all tell us about a given score, for example, that it was from the 'single' group) from the scores or measured variables themselves (in this case the columns labelled **couple** and **alone** both contain scores). Attributes are specified with the *id* option, and scores with the *measured* option. Therefore, we can create a molten dataframe called *profileMelt* by executing:

```
profileMelt<-melt(pictureData, id = c("case", "row", "relationship_status"), measured = c("couple", "alone"))
```

To plot the graph we use this molten dataframe as follows:

```
profileGraph <- ggplot(profileMelt, aes(relationship_status, friend_requests, colour = profile_picture))
```

```
profileGraph + stat_summary(fun.y = mean, geom = "point") + stat_summary(fun.y = mean, geom = "line", aes(group= profile_picture)) + stat_summary(fun.data = mean_cl_boot, geom = "errorbar", width = 0.2) + labs(x = "Relationship Status", y = "Number of Friend Requests", colour = "Contents of Profile Picture") + scale_y_continuous(limits = c(0,10))
```

To get the descriptive statistics execute:

```
by(profileMelt$friend_requests, list(profileMelt$profile_picture, profileMelt$relationship_status), stat.desc, basic = FALSE)
```

```
: couple
: In a Relationship
median      mean    SE.mean   CI.mean.0.95      var      std.dev     coef.var
3.0000000  3.2941176 0.2058824  0.4364511    0.7205882    0.8488747    0.2576941
-------------------------------------------------------------------------------------
: alone
: In a Relationship
median      mean    SE.mean   CI.mean.0.95      var      std.dev     coef.var
6.0000000  5.6470588 0.3314538  0.7026506    1.8676471    1.3666188    0.2420054
```

```
--------------------------------------------------------------------------------------
: couple
: Single
median      mean      SE.mean    CI.mean.0.95     var       std.dev      coef.var
4.0000000  3.9565217 0.2845026  0.5900223    1.8616601    1.3644266    0.3448551
--------------------------------------------------------------------------------------
: alone
median      mean      SE.mean    CI.mean.0.95     var       std.dev      coef.var
8.0000000  7.9130435 0.3971890  0.8237195    3.6284585    1.9048513    0.2407230
```

Labcoat Leni's real research

Keep the faith(ful)?

Problem

Schützwohl, A. (2008). *Personality and Individual Differences*, *44*, 633–644.

People can be jealous. People can be especially jealous when they think that their partner is being unfaithful. An evolutionary view of jealousy suggests that men and women have evolved distinctive types of jealousy because male and female reproductive success is threatened by different types of infidelity. Specifically, a woman's sexual infidelity deprives her mate of a reproductive opportunity and in some cases burdens him with years investing in a child that is not his. Conversely, a man's sexual infidelity does not burden his mate with unrelated children, but may divert his resources from his mate's progeny. This diversion of resources is signalled by emotional attachment to another female. Consequently, men's jealousy mechanism should have evolved to prevent a mate's *sexual* infidelity, whereas in women it has evolved to prevent *emotional* infidelity. If this is the case then men and women should divert their attentional resources towards different cues to infidelity: women should be 'on the look-out' for emotional infidelity, whereas men should be watching out for sexual infidelity.

Achim Schützwohl put this theory to the test in a unique study in which men and women saw sentences presented on a computer screen (Schützwohl, 2008). On each trial, participants saw a target sentence that was always affectively neutral (e.g. 'The gas station is at the other side of the street'). However, the trick was that before each of these targets, a distractor sentence was presented that could also be affectively neutral, or could indicate sexual infidelity (e.g. 'Your partner suddenly has difficulty becoming sexually aroused when he and you want to have sex') or emotional infidelity (e.g. 'Your partner doesn't say "I love you" to you anymore'). The idea was that if these distractor sentences grabbed a person's attention then (1) they would remember them, and (2) they would not remember the target sentence that came afterwards (because their attentional resources were still focused on the distractor). These effects should show up only in people currently in a relationship. The outcome was the number of sentences that a participant could remember (out of 6), and the predictors were whether the person had a partner or not (**Relationship**), whether the trial used a neutral distractor, an emotional infidelity distractor or a sexual infidelity distractor, and whether the sentence was a distractor or the target following the distractor. Schützwohl analysed men and women's data separately. The predictions are that women should remember more emotional infidelity sentences (distractors) but fewer of the targets that followed those sentences (target). For men, the same effect should be found but for sexual infidelity sentences.

The data from this study are in the file **Schützwohl(2008).dat**. Labcoat Leni wants you to carry out two three-way mixed ANOVAs (one for men and the other for women) to test these hypotheses.

Solution

First of all load in the data:

```
jealousData<-read.delim("Schutzwohl(2008).dat", header = TRUE)
```

Set **Relationship** and **Gender** to be factors:

```
jealousData$Relationship<-factor(jealousData$Relationship, levels = c(0:1), labels = c("Without Partner", "With Partner"))

jealousData$Gender<-factor(jealousData$Gender, levels = c(1:2), labels = c("Male", "Female"))
```

The data were originally in the wide format, but we need the data to be in the long format for these analyses. Therefore, we need to melt the dataframe by executing:

```
jealousLong<-melt(jealousData, id = c("Participant", "Relationship", "Age", "Distractor_Colour", "Gender"), measured = c("Distractor_Neutral", "Distractor_Emotional", "Distractor_Sexual", "Target_Neutral", "Target_Emotional", "Target_Sexual") )
names(jealousLong)<-c("Participant", "Relationship", "Age", "Distractor_Colour", "Gender", "RMVariables", "Sentences_Remembered")
```

The variable **RMVariables** is a mixture of our two repeated measures predictor variables (**Sentence Type** and **Distractor Type**). Note, for example, that the first 240 rows are scores for the distractor sentences and, within these 240 rows, the first 80 are the scores for the neutral distractors, the next 80 are the scores for the emotional distractor sentences and the final 80 are the scores for the sexual distractor sentences. We therefore need to create two variables that dissociate the type of sentence from the type of distractor; these two variables will be the two repeated measures predictors in our model.

First, let's create a variable called **Sentence_Type**, which specifies whether the sentence was a target or a distractor. We can do this using the *gl()* function and executing this command:

```
jealousLong$Sentence_Type<-gl(2, 240, labels = c("Distractor", "Target"))
```

This creates a variable **Sentence_Type** in the dataframe *jealousLong*. The numbers in the function tell **R** that we want to create two sets of 240 scores, the *labels* option then specifies the names to attach to these two sets, which correspond to the levels of sentence type. Essentially, this will create 240 rows labelled *Distractor* and then 240 rows labelled *Target*.

We also need a variable (called **Distractor_Type**) that tells us whether the distractor sentence was of a neutral, emotional or sexual nature. To do this we want three groups that each contain 80 scores. This will create 240 cases, or, put another way, it will create the codes for the first level (Distractor) of the **Sentence_Type** variable. We want this pattern to be repeated for the remaining level of **Sentence_Type** (i.e., target). We can do this by adding a third value to the function that is the total number of cases (i.e., 480). By specifying the total number of cases the *gl()* function will repeat the pattern of codes until it reaches this total number of cases:

```
jealousLong$Distractor_Type<-gl(3, 80, 480, labels = c("Neutral", "Emotional", "Sexual"))
```

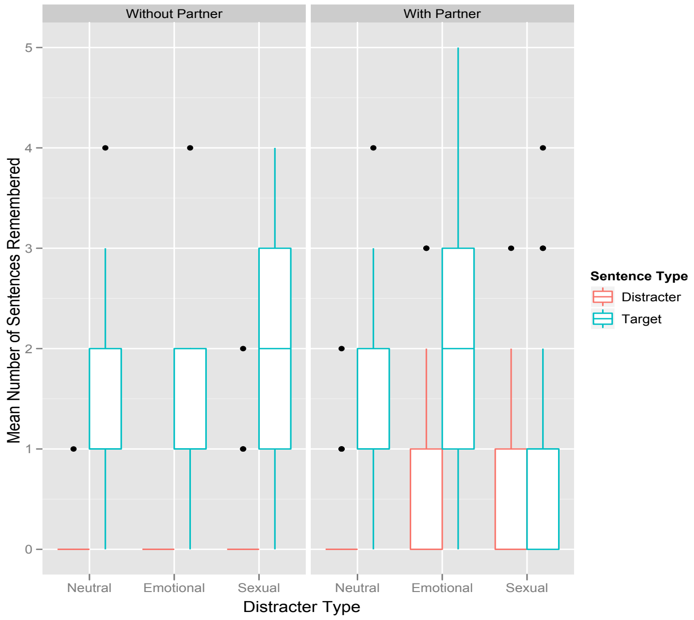
Now that the data are entered and are in the right format, we can get stuck into conducting the two three-way mixed ANOVAs. The question asks us to analyse the data separately for males and females as in the paper. Therefore, we need to create two new dataframes; one that keeps all of the variables but only the data for males, and one that keeps all of the variables but contains only the data for females. We can do this by executing the following commands (see Chapter 3):

```
malesOnly<-subset(jealousLong, Gender == "Male")

femalesOnly<-subset(jealousLong, Gender == "Female")
```

Executing the above commands produces two new dataframes, one called *malesOnly* that contains only the data for males, and one called *femalesOnly* that contains only the data for the females.

As ever, we'll look at some graphs first. To save space we'll look just at some boxplots. Let's look at the data for the men first. To create a boxplot of the mean number of distractor and target sentences remembered by males with and without a partner, we could execute the following command (remember to use the *malesOnly* dataframe):

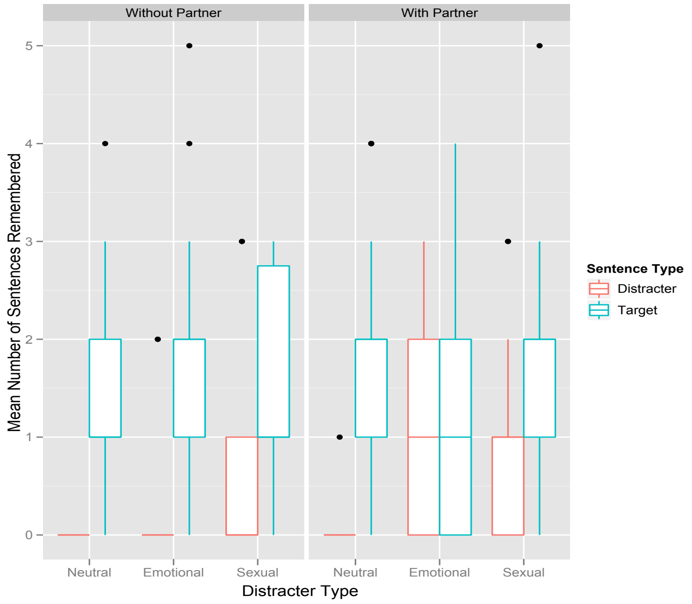
```
maleBoxplot <- ggplot(malesOnly, aes(Distractor_Type, Sentences_Remembered, colour =
 Sentence_Type))
maleBoxplot + geom_boxplot() + labs(x = "Distractor Type", y = "Mean Number of
 Sentences Remembered", colour = "Sentence Type") + facet_wrap(~Relationship)
```



The resulting plot above shows the pattern of scores for males with and without a partner. First off, in those without partners, they remember many more targets than they do distractors, and this is true for all types of trials. In other words, it doesn't matter whether the distractor is neutral, emotional or sexual; these people remember more targets than distractors. The same pattern is seen in those with partners *except* for distractors that indicate sexual infidelity. For these, the number of targets remembered is reduced. Put another way, they remember fewer targets that were preceded by a sexual-infidelity distractor. This supports the predictions of the author: men in relationships have an attentional bias such that their attention is consumed by cues indicative of sexual infidelity.

Next we want to look at the women. To create a boxplot of the mean number of distractor and target sentences remembered by females with and without a partner we could execute the following command (remember to use the *femalesOnly* dataframe):

```
femaleBoxplot <- ggplot(femalesOnly, aes(Distractor_Type, Sentences_Remembered, colour
 = Sentence_Type))
femaleBoxplot + geom_boxplot() + labs(x = "Distractor Type", y = "Mean Number of
 Sentences Remembered", colour = "Sentence Type") + facet_wrap(~Relationship)
```



As for the men, women without partners remember many more targets than they do distractors, and this is true for all types of trials (although it's less true for the sexual infidelity trials because this line has a shallower slope). The same pattern is seen in those with partners *except* for distractors that indicate emotional infidelity. For these, the number of targets remembered is reduced. They remember fewer targets that were preceded by an emotional-infidelity distractor, and these women also remember more of the emotional-infidelity distractors than women who are not in a relationship. This supports the predictions of the author: women in relationships have an attentional bias such that their attention is consumed by cues indicative of emotional infidelity.

Next, let's get some descriptive statistics for the male data using the *by()* function. To get descriptive statistics for the combined levels of **Distractor_Type**, **Sentence_Type**, and **Relationship** we execute:

```
by(malesOnly$Sentences_Remembered,list(malesOnly$Distractor_Type,
malesOnly$Sentence_Type, malesOnly$Relationship), stat.desc, basic = FALSE)
```

The resulting edited output below contains descriptive statistics for each of the six conditions split according to whether the male participants were in a relationship or not. These descriptive statistics are interesting because they show us the pattern of means across all experimental conditions (so we use these means to produce the graphs of the three-way interaction).

```
: Neutral
: Distractor
: Without Partner
median      mean          SE.mean     CI.mean.0.95      var          std.dev      coef.var
0.0000000  0.07692308   0.07692308   0.16760099   0.07692308   0.27735010   3.60555128
-------------------------------------------------------------------------------------
: Emotional
: Distractor
: Without Partner
median          mean      SE.mean  CI.mean.0.95      var       std.dev      coef.var
  0              0          0           0             0           0            NaN
-------------------------------------------------------------------------------------
: Sexual
: Distractor
: Without Partner
median         mean      SE.mean  CI.mean.0.95    var        std.dev      coef.var
0.0000000     0.3076923  0.1748485    0.3809621  0.3974359  0.6304252    2.0488818
-------------------------------------------------------------------------------------
: Neutral
: Target
```

```
: Without Partner
median      mean      SE.mean   CI.mean.0.95  var       std.dev   coef.var
1.0000000  1.4615385 0.3124630  0.6807984    1.2692308 1.1266014  0.7708326
--------------------------------------------------------------------------------------
: Emotional
: Target
: Without Partner
 median     mean       SE.mean   CI.mean.0.95  var       std.dev   coef.var
 1.0000000  1.3846154  0.3108809  0.6773513    1.2564103  1.1208971  0.8095368
--------------------------------------------------------------------------------------
: Sexual
: Target
: Without Partner
 median     mean       SE.mean   CI.mean.0.95 var       std.dev    coef.var
 2.0000000  2.0000000  0.3580574 0.7801401    1.6666667 1.2909944  0.6454972
--------------------------------------------------------------------------------------
: Neutral
: Distractor
: With Partner
median     mean       SE.mean     CI.mean.0.95  var        std.dev     coef.var
0.00000000 0.19230769 0.09638434  0.19850726    0.24153846 0.49146563  2.55562126
--------------------------------------------------------------------------------------
: Emotional
: Distractor
: With Partner
median     mean      SE.mean    CI.mean.0.95  var       std.dev    coef.var
0.0000000 0.5384615 0.1774240  0.3654117     0.8184615 0.9046886  1.9601587
--------------------------------------------------------------------------------------
: Sexual
: Distractor
: With Partner
median      mean      SE.mean    CI.mean.0.95  var       std.dev   coef.var
0.5000000   0.6923077 0.1780898  0.3667828     0.8246154 0.9080834 1.1805084
--------------------------------------------------------------------------------------
: Neutral
: Target
: With Partner
median       mean      SE.mean   CI.mean.0.95 var       std.dev    coef.var
   1.0000000 1.5384615 0.2017673 0.4155476    1.0584615  1.0288156  0.6687301
--------------------------------------------------------------------------------------
: Emotional
: Target
: With Partner
median      mean      SE.mean    CI.mean.0.95 var       std.dev    coef.var
2.0000000   1.9230769 0.2931163  0.6036843    2.2338462 1.4946057  0.7771950
--------------------------------------------------------------------------------------
: Sexual
: Target
: With Partner
median      mean      SE.mean   CI.mean.0.95  var       std.dev    coef.var
1.0000000   1.0000000 0.2287087 0.4710345     1.3600000 1.1661904  1.1661904
```

To get the descriptives for the females we execute:

```
by(femalesOnly$Sentences_Remembered, list(femalesOnly$Distractor_Type,
femalesOnly$Sentence_Type, femalesOnly$Relationship), stat.desc, basic = FALSE)
```

```
: Neutral
: Distractor
: Without Partner
median    mean   SE.mean CI.mean.0.95   var    std.dev    coef.var
0         0      0            0         0      0           NaN
--------------------------------------------------------------------------------------
: Emotional
: Distractor
: Without Partner
median       mean       SE.mean     CI.mean.0.95    var       std.dev     coef.var
0.0000000    0.2857143    0.1941046  0.4193376    0.5274725  0.7262730    2.5419556
--------------------------------------------------------------------------------------
: Sexual
: Distractor
: Without Partner
median      mean      SE.mean    CI.mean.0.95    var        std.dev     coef.var
0.0000000 0.7142857    0.2857143 0.6172482    1.1428571    1.0690450    1.4966630
--------------------------------------------------------------------------------------
: Neutral
: Target
: Without Partner
```

```
median      mean      SE.mean   CI.mean.0.95 var       std.dev    coef.var
1.0000000  1.5000000  0.3273268 0.7071466    1.5000000 1.2247449  0.8164966
------------------------------------------------------------------------------------
: Emotional
: Target
: Without Partner
median      mean      SE.mean   CI.mean.0.95  var       std.dev    coef.var
2.0000000  2.0000000  0.3476767 0.7511099    1.6923077  1.3008873  0.6504436
------------------------------------------------------------------------------------
: Sexual
: Target
: Without Partner
median      mean      SE.mean   CI.mean.0.95  var       std.dev     coef.var
1.0000000  1.5000000 0.2918301 0.6304607      1.1923077 1.0919284   0.7279523
------------------------------------------------------------------------------------
: Neutral
: Distractor
: With Partner
median       mean       SE.mean      CI.mean.0.95  var         std.dev    coef.var
0.00000000  0.03703704  0.03703704  0.07613072    0.03703704  0.19245009 5.19615242
------------------------------------------------------------------------------------
: Emotional
: Distractor
: With Partner
median      mean      SE.mean   CI.mean.0.95  var        std.dev     coef.var
1.0000000   1.1851852 0.2137512 0.4393718     1.2336182  1.1106837   0.9371394
------------------------------------------------------------------------------------
: Sexual
: Distractor
: With Partner
median     mean      SE.mean   CI.mean.0.95 var       std.dev   coef.var
1.0000000 0.8148148 0.1773261  0.3644990    0.8490028 0.9214135 1.1308257
------------------------------------------------------------------------------------
: Neutral
: Target
: With Partner
median     mean      SE.mean   CI.mean.0.95 var       std.dev     coef.var
2.0000000 1.8888889  0.2222222 0.4567843    1.3333333 1.1547005    0.6113120
------------------------------------------------------------------------------------
: Emotional
: Target
: With Partner
median         mean      SE.mean CI.mean.0.95         var       std.dev     coef.var
1.0000000    1.3703704    0.2335661    0.4801020    1.4729345    1.2136451
0.8856329
------------------------------------------------------------------------------------
: Sexual
: Target
: With Partner
median      mean       SE.mean    CI.mean.0.95  var       std.dev    coef.var
2.0000000   1.6296296  0.2144904  0.4408913     1.2421652 1.1145247  0.6839129
```

OK, now to the mixed ANOVA! We will run the analysis for males using *ezANOVA()* first. We want to use Type III sums of squares, so we have to set some orthogonal contrasts for our predictor variables. For **Distractor_Type** we could consider the neutral category as a useful control condition. Therefore, we could create one contrast that compares emotional and sexual distractors to neutral distractors, and then one that compares the sexual to the emotional distractors. With regard to the **Sentence_Type** and **Relationship** variables, we don't really need to worry about setting contrasts because these variables only have two conditions (Target vs. Distractor and With Partner vs. Without Partner, respectively) and so **R** will set orthogonal contrasts by default – because there is only one way of comparing two variables!

To set the orthogonal contrasts we can first create variables representing each contrast and then bind these variables together and set them as the contrast for **Distractor_Type**:

```
NeutralvsEmotionalandSexual<-c(-2, 1, 1)
EmotionalvsSexual<-c(0, -1, 1)

contrasts(malesOnly$Distractor_Type)<-cbind(NeutralvsEmotionalandSexual,
 EmotionalvsSexual)
```

We can then run the analysis by executing the following command:

```
malesModel<-ezANOVA(data = malesOnly, dv = .(Sentences_Remembered), wid
 = .(Participant),  between = .(Relationship), within = .(Distractor_Type,
 Sentence_Type), type = 3, detailed = TRUE)

malesModel
```

```
$ANOVA
                                      Effect DFn DFd    SSn  SSd       F        p p<.05        ges
1                                (Intercept)   1  37 208.722 55.5 139.164 4.22e-14     * 0.48420
2                               Relationship   1  37   0.618 55.5   0.412 5.25e-01       0.00277
3                            Distractor_Type   2  74   1.855 58.6   1.171 3.16e-01       0.00827
4               Relationship:Distractor_Type   2  74   6.209 58.6   3.920 2.41e-02     * 0.02717
5                              Sentence_Type   1  37  80.209 55.7  53.282 1.14e-08     * 0.26511
6                 Relationship:Sentence_Type   1  37   2.925 55.7   1.943 1.72e-01       0.01299
7              Distractor_Type:Sentence_Type   2  74   4.726 52.6   3.328 4.13e-02     * 0.02081
8 Relationship:Distractor_Type:Sentence_Type   2  74   5.389 52.6   3.794 2.70e-02     * 0.02366

$`Mauchly's Test for Sphericity`
                                      Effect     W     p p<.05
3                            Distractor_Type 0.956 0.449
4               Relationship:Distractor_Type 0.956 0.449
7              Distractor_Type:Sentence_Type 0.997 0.940
8 Relationship:Distractor_Type:Sentence_Type 0.997 0.940

$`Sphericity Corrections`
                                      Effect   GGe  p[GG]  p[GG]<.05  HFe  p[HF] p[HF]<.05
3                            Distractor_Type 0.958 0.3145            1.01 0.3157
4               Relationship:Distractor_Type 0.958 0.0258          * 1.01 0.0241         *
7              Distractor_Type:Sentence_Type 0.997 0.0415          * 1.05 0.0413         *
8 Relationship:Distractor_Type:Sentence_Type 0.997 0.0271          * 1.05 0.0270         *
```

The output above shows the results of Mauchly's sphericity test for **Distractor_Type** (as it has three conditions) and the interaction of **Distractor_Type** with **Sentence_Type** and **Relationship**. None of the effects violate the assumption of sphericity because all of the values in the column labeled *p* are above .05. Therefore, we can assume sphericity when we look at our *F*-statistics.

Next, we can look at the summary table (labeled *$*ANOVA) of the effects in the ANOVA. We could report these effects as follows. A three-way ANOVA with current relationship status as the between-subject factor and men's recall of sentence type (targets vs. distractors) and distractor type (neutral, emotional infidelity and sexual infidelity) as the within-subjects factors yielded a significant main effect of sentence type, $F(1, 37) = 53.28$, $p < .001$, a significant interaction between distractor type and sentence type, $F(2, 74) = 3.33$, $p = .041$, and a significant interaction between current relationship status and distractor content, $F(2, 74) = 3.92$, $p = .024$. More important, the three-way interaction was also significant, $F(2, 74) = 3.79$, $p = .027$. The remaining main effects and interactions were not significant, *F*s < 2, *p*s > .17.

Now let's run the same analysis for the females. Again, we want to use Type III sums of squares so we have to set some orthogonal contrasts for our predictor variables. We can set the same contrasts as we set for the males – just remember to specify that we want to use the *femalesOnly* dataframe:

```
NeutralvsEmotionalandSexual<-c(-2, 1, 1)
EmotionalvsSexual<-c(0, -1, 1)

contrasts(femalesOnly$Distractor_Type)<-cbind(NeutralvsEmotionalandSexual,
 EmotionalvsSexual)
```

We can then run the analysis as before by executing the following command:

```
femalesModel<-ezANOVA(data = femalesOnly, dv = .(Sentences_Remembered), wid
 = .(Participant),  between = .(Relationship), within = .(Distractor_Type,
 Sentence_Type), type = 3, detailed = TRUE)

femalesModel
```

```
$ANOVA
                       Effect DFn DFd      SSn  SSd        F        p p<.05      ges
1                 (Intercept)   1  39 298.5407 49.6 234.5411 4.36e-18     * 0.555295
2                Relationship   1  39   1.3174 49.6   1.0350 3.15e-01       0.005480
3             Distractor_Type   2  78   5.5935 50.3   4.3362 1.64e-02     * 0.022861
```

```
4              Relationship:Distractor_Type   2  78   0.0987 50.3   0.0765 9.26e-01   0.000413
5                             Sentence_Type   1  39  71.9065 70.9  39.5530 2.05e-07 * 0.231217
6                Relationship:Sentence_Type   1  39   2.0256 70.9   1.1142 2.98e-01   0.008401
7               Distractor_Type:Sentence_Type 2  78  13.1057 68.2   7.4908 1.06e-03 * 0.051967
8 Relationship:Distractor_Type:Sentence_Type  2  78   9.3273 68.2   5.3312 6.76e-03 * 0.037548

$`Mauchly's Test for Sphericity`
                                      Effect     W     p p<.05
3                            Distractor_Type 0.968 0.540
4               Relationship:Distractor_Type 0.968 0.540
7              Distractor_Type:Sentence_Type 0.945 0.343
8 Relationship:Distractor_Type:Sentence_Type 0.945 0.343

$`Sphericity Corrections`
                                      Effect   GGe   p[GG] p[GG]<.05   HFe   p[HF] p[HF]<.05
3                            Distractor_Type 0.969 0.01741         * 1.019 0.01638         *
4               Relationship:Distractor_Type 0.969 0.92158           1.019 0.92638
7              Distractor_Type:Sentence_Type 0.948 0.00132         * 0.995 0.00108         *
8 Relationship:Distractor_Type:Sentence_Type 0.948 0.00778         * 0.995 0.00685         *
```

The output above shows the results of Mauchly's sphericity test for **Distractor_Type** (as it has 3 conditions) and the interaction of **Distractor_Type** with **Sentence_Type** and **Relationship**. None of the effects violate the assumption of sphericity because all of the values in the column labeled *p* are above .05. Therefore, we can assume sphericity when we look at our *F*-statistics for the females.

We could report these effects as follows. A three-way ANOVA with current relationship status as the between-subject factor and women's recall of sentence type (targets vs. distractors) and distractor type (neutral, emotional infidelity and sexual infidelity) as the within-subject factors yielded a significant main effect of sentence type, $F(1, 39) = 39.55$, $p < .001$, and distractor type, $F(2, 78) = 4.34$, $p = .016$. Additionally, significant interactions were found between sentence type and distractor type, $F(2, 78) = 7.49$, $p < .01$, and, most important, sentence type × distractor type × relationship, $F(2, 78) = 5.33$, $p = .007$. The remaining main effect and interactions were not significant, *F*s < 1.2, *p*s > .29.

Using lme()

Let's run the analysis again for men and women separately but this time using *lme()* – I know you want to! Let's look at the males first.

Before we build the model we need to set some contrasts. Although we already set some contrasts for using *ezANOVA*, those contrasts we set were simply so that we could get Type III sums of squares, and we were constrained to use orthogonal contrasts. However, if we use a multilevel model we don't have to worry about orthogonal contrasts because we don't have to concern ourselves with types of sums of squares in the same way that we do for ANOVA. As such, we can set some non-orthogonal contrasts. In the variable **Distractor_Type** there were three conditions – neutral, emotional and sexual – and it makes sense to compare the emotional and sexual conditions to the neutral condition. We can do this by setting 'neutral' to zero in both contrasts, and in one contrast set 'emotional' to be 1 and in the other contrast set 'sexual' to be 1 (I have given the contrasts very short names to save space in the output table):

```
EvsN<-c(0, 1, 0)
SvsN<-c(0, 0, 1)

contrasts(malesOnly$Distractor_Type)<-cbind(EvsN, SvsN)
```

With regard to the variables **Relationship** and **Sentence_Type**, there were only two levels in each and therefore the only contrasts that we are able to set are comparing level 1 to level 2:

```
contrasts(malesOnly$Relationship)<-c(1, 0)
contrasts(malesOnly$Sentence_Type)<-c(1, 0)
```

Now let's build our model. We want to look at the overall main effects and interactions so we will build up the model a predictor at a time from a baseline that includes no predictors other than the intercept. We can specify the baseline model by executing:

```
baseline<-lme(Sentences_Remembered ~ 1, random =
```

```
~1|Participant/Distractor_Type/Sentence_Type, data = malesOnly, method = "ML")
```

To see the overall effect of each main effect and interaction we need to add them to the model one at a time. To do this we could execute:

```
SentenceM<-update(baseline, .~. + Sentence_Type)

DistractorM<-update(SentenceM, .~. + Distractor_Type)

RelationshipM<-update(DistractorM, .~. + Relationship)

Sentence_Relationship<-update(RelationshipM, .~. + Sentence_Type:Relationship)

Distractor_Relationship<-update(Sentence_Relationship, .~. +
Distractor_Type:Relationship)

Sentence_Distractor<-update(Distractor_Relationship, .~. +
Sentence_Type:Distractor_Type)

malejealousModel<-update(Sentence_Distractor, .~. +
Sentence_Type:Distractor_Type:Relationship)
```

Executing the final command above creates a model called *malejealousModel*, which contains all main effects and interactions. This is the final model. To compare these models we can list them in the order in which we want them compared in the *anova()* function:

```
anova(baseline, SentenceM, DistractorM, RelationshipM, Sentence_Relationship,
Distractor_Relationship, Sentence_Distractor, malejealousModel)
```

Executing the above command produces:

```
                        Model df      AIC      BIC    logLik   Test  L.Ratio p-value
baseline                    1  5 750.3791 767.6557 -370.1896
SentenceM                   2  6 683.0773 703.8092 -335.5386 1 vs 2 69.30184  <.0001
DistractorM                 3  8 685.1435 712.7861 -334.5718 2 vs 3  1.93375  0.3803
RelationshipM               4  9 686.7119 717.8098 -334.3560 3 vs 4  0.43159  0.5112
Sentence_Relationship       5 10 685.6225 720.1757 -332.8113 4 vs 5  3.08941  0.0788
Distractor_Relationship     6 12 682.8977 724.3616 -329.4489 5 vs 6  6.72480  0.0347
Sentence_Distractor         7 14 681.6188 729.9933 -326.8094 6 vs 7  5.27891  0.0714
malejealousModel            8 16 679.4204 734.7055 -323.7102 7 vs 8  6.19841  0.0451
```

The output above first compares the effect of **Sentence_Type** to the baseline. By adding **Sentence_type** as a predictor we significantly improve the model. In other words, the type of sentence (target vs. distractor) had a significant effect on the number of sentences remembered by males, $\chi^2(6) = 69.30$, $p < .0001$. The model (*Distractor_Relationship*) shows that the **Distractor_Type** × **Relationship** interaction is also significant, $\chi^2(12) = 6.72$, $p = .035$. This significant interaction means that the way in which the number of sentences remembered was affected by distractor type (whether the sentence was neutral, emotional or sexual) was different for males with and without a partner. The final model (*jealousModel*) shows that the **Sentence_Type** × **Distractor_Type** × **Relationship** interaction is also significant, $\chi^2(16) = 6.20$, $p = .045$, meaning that the **Sentence_Type** × **Distractor_Type** interaction was significantly different in males with a partner and males without a partner. The remaining main effects and interactions were not significant, *p*s > .29. We can see the model parameters by executing:

```
summary(malejealousModel)
```

```
Fixed effects: Sentences_Remembered ~ Sentence_Type + Distractor_Type + Relationship +
Sentence_Type:Relationship + Distractor_Type:Relationship + Sentence_Type:Distractor_Type +
Sentence_Type:Distractor_Type:Relationship

                                      Value Std.Error  DF   t-value p-value
(Intercept)                       1.5384615 0.1962689 111   7.838540  0.0000
Sentence_Type                    -1.3461538 0.2633960 111  -5.110760  0.0000
Distractor_TypeEvsN               0.3846154 0.2633960  74   1.460217  0.1485
Distractor_TypeSvsN              -0.5384615 0.2633960  74  -2.044304  0.0445
Relationship                     -0.0769231 0.3399477  37 - 0.226279  0.8222
Sentence_Type:Relationship       -0.0384615 0.4562153 111  -0.084306  0.9330
Distractor_TypeEvsN:Relationship -0.4615385 0.4562153  74  -1.011668  0.3150
```

```
Distractor_TypeSvsN:Relationship                    1.0769231 0.4562153  74    2.360559  0.0209
Sentence_Type:Distractor_TypeEvsN                  -0.0384615 0.3724983  111  -0.103253  0.9179
Sentence_Type:Distractor_TypeSvsN                   1.0384615 0.3724983  111   2.787829  0.0062
Sentence_Type:Distractor_TypeEvsN:Relationship      0.0384615 0.6451859  111   0.059613  0.9526
Sentence_Type:Distractor_TypeSvsN:Relationship     -1.3461538 0.6451859  111  -2.086459  0.0392
```

The output above shows the parameter estimates for the model (I've edited some of the names to save space), along with the contrasts that we requested. The most important effects are the two three-way interactions at the bottom of the table. This table tells us that, as predicted, the effect of whether or not you are in a relationship and whether you were remembering a distractor or target was significantly different in trials in which a sexual-infidelity distractor was used compared to when a neutral distractor was used, b = –1.35, $t(111)$ = –2.09, p = .039 However, there was not a significant difference in trials in which an emotional infidelity distractor was used compared to those in which a neutral distractor was used, b = 0.04, $t(111)$ = 0.06, p = .953.

Now let's look at the females. We can do exactly the same as we did for males but this time using the *femalesOnly* dataframe that we created earlier. Set the same contrasts:

```
EvsN<-c(0, 1, 0)
SvsN<-c(0, 0, 1)

contrasts(femalesOnly$Distractor_Type)<-cbind(EvsN, SvsN)
contrasts(femalesOnly$Relationship)<-c(1, 0)
contrasts(femalesOnly$Sentence_Type)<-c(1, 0)
```

Now let's build our model a predictor at a time from a baseline:

```
baseline<-lme(Sentences_Remembered ~ 1, random =
~1|Participant/Distractor_Type/Sentence_Type, data = femalesOnly, method = "ML")
```

To see the overall effect of each main effect and interaction we need to add them to the model one at a time. To do this we could execute:

```
SentenceM<-update(baseline, .~. + Sentence_Type)

DistractorM<-update(SentenceM, .~. + Distractor_Type)

RelationshipM<-update(DistractorM, .~. + Relationship)

Sentence_Relationship<-update(RelationshipM, .~. + Sentence_Type:Relationship)

Distractor_Relationship<-update(Sentence_Relationship, .~. +
Distractor_Type:Relationship)

Sentence_Distractor<-update(Distractor_Relationship, .~. +
Sentence_Type:Distractor_Type)

femalejealousModel<-update(Sentence_Distractor, .~. +
Sentence_Type:Distractor_Type:Relationship)
```

Executing the final command above creates a model called fe*malejealousModel*, which contains all main effects and interactions of the female data. This is the final model. To compare these models we can list them in the order in which we want them compared in the *anova()* function:

```
anova(baseline, SentenceM, DistractorM, RelationshipM, Sentence_Relationship,
Distractor_Relationship, Sentence_Distractor, femalejealousModel)
```

Executing the above command produces:

```
                      Model df      AIC      BIC    logLik   Test  L.Ratio p-value
baseline                  1  5 789.4998 807.0265 -389.7499
SentenceM                 2  6 733.1309 754.1629 -360.5655 1 vs 2 58.36890  <.0001
DistractorM               3  8 731.8415 759.8842 -357.9207 2 vs 3  5.28943  0.0710
RelationshipM             4  9 732.7676 764.3156 -357.3838 3 vs 4  1.07385  0.3001
Sentence_Relationship     5 10 732.8180 767.8713 -356.4090 4 vs 5  1.94965  0.1626
```

```
Distractor_Relationship     6 12 736.7225 778.7865 -356.3612 5 vs 6  0.09552  0.9534
Sentence_Distractor         7 14 727.6330 776.7076 -349.8165 6 vs 7 13.08953  0.0014
femalejealousModel          8 16 721.7803 777.8656 -344.8901 7 vs 8  9.85265  0.0073
```

The output above first compares the effect of **Sentence_Type** to the baseline. By adding **Sentence_type** as a predictor we significantly improve the model. In other words, the type of sentence (target vs. distractor) had a significant effect on the number of sentences remembered by females, $\chi^2(6) = 58.37$, $p < .0001$. The model (*Sentence_Distractor*) shows that the **Sentence_Type** × **Distractor_Type** interaction is also significant, $\chi^2(14) = 13.09$, $p < .01$. This significant interaction means that, although the number of sentences remembered was affected by the type of sentence (distractor or a target), the way in which the number of sentences remembered was affected by sentence type was different for the different types of distractor sentences (neutral, emotional and sexual). The final model (*jealousModel*) shows that the **Sentence_Type** × **Distractor_Type** × **Relationship** interaction is also significant, $\chi^2(16) = 9.85$, $p < .01$, meaning that the **Sentence_Type** × **Distractor_Type** interaction was significantly different in males with a partner and males without a partner. The remaining main effects and interactions were not significant, *p*s > .29. We can see the model parameters by executing:

```
summary(femalejealousModel)
```

```
Fixed effects: Sentences_Remembered ~ Sentence_Type + Distractor_Type + Relationship +
Sentence_Type:Relationship + Distractor_Type:Relationship + Sentence_Type:Distractor_Type +
Sentence_Type:Distractor_Type:Relationship

                                              Value Std.Error  DF   t-value p-value
(Intercept)                               1.8888889 0.1945297 117  9.710026  0.0000
Sentence_Type1                           -1.8518519 0.2682594 117 -6.903214  0.0000
Distractor_TypeEvsN                      -0.5185185 0.2682594  78 -1.932900  0.0569
Distractor_TypeSvsN                      -0.2592593 0.2682594  78 -0.966450  0.3368
Relationship1                            -0.3888889 0.3329001  39 -1.168185  0.2498
Sentence_Type1:Relationship1              0.3518519 0.4590741 117  0.766438  0.4450
Distractor_TypeEvsN:Relationship1         1.0185185 0.4590741  78  2.218636  0.0294
Distractor_TypeSvsN:Relationship1         0.2592593 0.4590741  78  0.564744  0.5739
Sentence_Type1:Distractor_TypeEvsN        1.6666667 0.3793760 117  4.393179  0.0000
Sentence_Type1:Distractor_TypeSvsN        1.0370370 0.3793760 117  2.733533  0.0072
Sentence_Type1:Distractor_TypeEvsN:Relationship1 -1.8809524 0.6492288 117 -2.897210  0.0045
Sentence_Type1:Distractor_TypeSvsN:Relationship1 -0.3227513 0.6492288 117 -0.497130  0.6200
```

The output above shows the parameter estimates for the model (I've edited some of the names to save space), along with the contrasts that we requested. The most important effects are the two three-way interactions at the bottom of the table. This table tells us that as predicted, the effect of whether or not you are in a relationship and whether you were remembering a distractor or target was significantly different in trials in which an emotional infidelity distractor was used compared to when a neutral distractor was used, $b = -1.88$, $t(117) = -2.90$, $p < .01$. However, there was not a significant difference in trials in which a sexual-infidelity distractor was used compared to those in which a neutral distractor was used, $b = -0.32$, $t(117) = -0.50$, $p = .62$.

These results support the predictions of the author: men in relationships have an attentional bias such that their attention is consumed by cues indicative of sexual infidelity, whereas women in relationships have an attentional bias such that their attention is consumed by cues indicative of emotional infidelity.

Smart Alex's solutions

Task 1

- I am going to extend the example from the previous chapter (advertising and different imagery) by adding a between-group variable into the design.[1] To recap, participants viewed a total of nine mock adverts over three sessions. In these adverts there were three products (a brand of beer, a brand of wine, and a brand of water). These could

[1] Previously the example contained two repeated-measures variables (drink type and imagery type), but now it will include three variables (two repeated measures and one between-group).

be presented alongside positive, negative or neutral imagery. Over the three sessions and nine adverts, each type of product was paired with each type of imagery (read the previous chapter if you need more detail). After each advert participants rated the drinks on a scale ranging from −100 (dislike very much) through 0 (neutral) to 100 (like very much). The design, thus far, has two independent variables: the type of drink (beer, wine or water) and the type of imagery used (positive, negative or neutral). I also took note of each person's gender. It occurred to me that men and women might respond differently to the products (because, in keeping with stereotypes, men might mostly drink lager whereas women might drink wine). Therefore, I wanted to analyse the data taking this additional variable into account. Now, gender is a between-group variable because a participant can be only male or female: they cannot participate as a male and then change into a female and participate again! The data are the same as in the previous chapter (Table 13.4) and can be found in the file **MixedAttitude.dat**. Run a mixed ANOVA on these data.

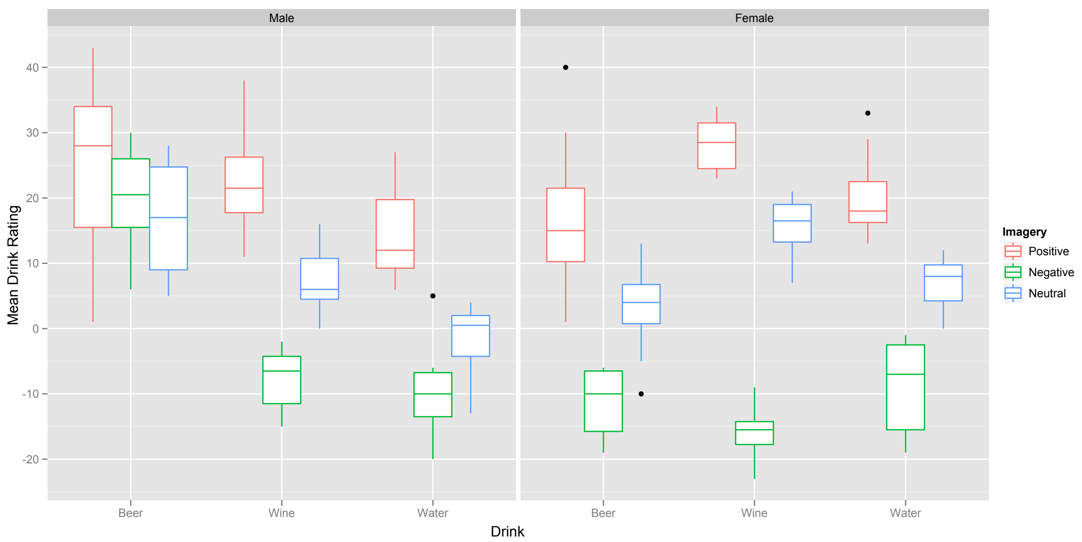
Read in the data:

```
mixedAttitude<-read.delim("MixedAttitude.dat", header = TRUE)
```

Set gender to be a factor:

```
mixedAttitude$gender<-factor(mixedAttitude$gender, levels = c(1:2), labels = c("Male", "Female"))

attitudeLong<-melt(mixedAttitude, id = c("Participant", "gender"), measured = c("beerpos", "beerneg", "beerneut", "winepos", "wineneg", "wineneut", "waterpos", "waterneg", "waterneu") )

names(attitudeLong)<-c("Participant", "Gender", "Groups", "Drink_Rating")

attitudeLong$Drink<-gl(3, 60, labels = c("Beer", "Wine", "Water"))
attitudeLong$Imagery<-gl(3, 20, 180, labels = c("Positive", "Negative", "Neutral"))
```

```
attitudeBoxplot <- ggplot(attitudeLong, aes(Drink, Drink_Rating, colour = Imagery))
attitudeBoxplot + geom_boxplot() + labs(x = "Drink", y = "Mean Drink Rating", colour = "Imagery") + facet_wrap(~Gender)
```



This boxplot shows us the three-way interaction between the variables. We can see that the variability among scores was greatest when beer was used as a product, and that when a corpse image was used the ratings given to the products were negative (as expected) for all conditions except the men in the beer condition. Likewise, ratings of products were very positive when a sexy person was used as the imagery irrespective of the gender of the participant, or the product being advertised.

```
by(attitudeLong$Drink_Rating, list(attitudeLong$Drink, attitudeLong$Imagery,
 attitudeLong$Gender), stat.desc, basic = FALSE)
```

```
: Beer
: Positive
: Male
      median         mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
  28.0000000   24.8000000    4.4291961   10.0195376  196.1777778   14.0063478    0.5647721
------------------------------------------------------------------------------------------------
: Wine
: Positive
: Male
      median         mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
  21.5000000   22.3000000    2.4131584    5.4589435   58.2333333    7.6310768    0.3422008
------------------------------------------------------------------------------------------------
: Water
: Positive
: Male
      median         mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
  12.0000000   14.5000000    2.1460558    4.8547155   46.0555556    6.7864244    0.4680293
------------------------------------------------------------------------------------------------
: Beer
: Negative
: Male
      median         mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
  20.5000000   20.1000000    2.4785749    5.6069259   61.4333333    7.8379419    0.3899474
------------------------------------------------------------------------------------------------
: Wine
: Negative
: Male
      median         mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
  -6.5000000   -7.8000000    1.5620499    3.5336024   24.4000000    4.9396356   -0.6332866
------------------------------------------------------------------------------------------------
: Water
: Negative
: Male
      median         mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
 -10.0000000   -9.8000000    2.1437247    4.8494422   45.9555556    6.7790527   -0.6917401
------------------------------------------------------------------------------------------------
: Beer
: Neutral
: Male
      median         mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
  17.0000000   16.9000000    2.7016456    6.1115469   72.9888889    8.5433535    0.5055239
------------------------------------------------------------------------------------------------
: Wine
: Neutral
: Male
      median         mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
   6.0000000    7.5000000    1.5723302    3.5568580   24.7222222    4.9721446    0.6629526
------------------------------------------------------------------------------------------------
: Water
: Neutral
: Male
      median         mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
    0.500000    -2.100000     1.991370     4.504793    39.655556     6.297266    -2.998698
------------------------------------------------------------------------------------------------
: Beer
: Positive
: Female
      median         mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
  15.0000000   17.3000000    3.6026225    8.1496983  129.7888889   11.3924927    0.6585256
------------------------------------------------------------------------------------------------
: Wine
: Positive
: Female
      median         mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
  28.5000000   28.4000000    1.3012814    2.9437031   16.9333333    4.1150132    0.1448948
------------------------------------------------------------------------------------------------
: Water
: Positive
: Female
      median         mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
  18.0000000   20.3000000    2.0223748    4.5749297   40.9000000    6.3953108    0.3150399
------------------------------------------------------------------------------------------------
: Beer
: Negative
: Female
```

```
     median        mean     SE.mean CI.mean.0.95         var     std.dev    coef.var 
-10.0000000 -11.2000000   1.6248077    3.6755703  26.4000000   5.1380930  -0.4587583 
-------------------------------------------------------------------------------------------------
: Wine
: Negative
: Female
     median        mean     SE.mean CI.mean.0.95         var     std.dev    coef.var 
-15.5000000 -16.2000000   1.3063945    2.9552697  17.0666667   4.1311822  -0.2550112 
-------------------------------------------------------------------------------------------------
: Water
: Negative
: Female
     median        mean     SE.mean CI.mean.0.95         var     std.dev    coef.var 
 -7.0000000  -8.6000000   2.2568415    5.1053301  50.9333333   7.1367593  -0.8298557 
-------------------------------------------------------------------------------------------------
: Beer
: Neutral
: Female
     median        mean     SE.mean CI.mean.0.95         var     std.dev    coef.var 
   4.000000    3.100000    2.121058     4.798168   44.988889    6.707376    2.163670 
-------------------------------------------------------------------------------------------------
: Wine
: Neutral
: Female
     median        mean     SE.mean CI.mean.0.95         var     std.dev    coef.var 
 16.5000000  15.8000000   1.3888444    3.1417844  19.2888889   4.3919118   0.2779691 
-------------------------------------------------------------------------------------------------
: Water
: Neutral
: Female
     median        mean     SE.mean CI.mean.0.95         var     std.dev    coef.var 
  8.0000000   6.8000000   1.2274635    2.7767154  15.0666667   3.8815804   0.5708207 
```

The output table above contains the descriptive statistics (mean and standard deviation) for each of the nine conditions split according to whether participants were male or female. These descriptive statistics are interesting because they show us the pattern of means across all experimental conditions (so, we use these means to produce the graphs of the three-way interaction). Again, like we saw from the interaction graph, we can see that the variability among scores was greatest when beer was used as a product, and that when a corpse image was used the ratings given to the products were negative (as expected) for all conditions except the men in the beer condition. Likewise, ratings of products were very positive when a sexy person was used as the imagery irrespective of the gender of the participant, or the product being advertised.

Using *ezANOVA* produces the following output:

```
NeutvsPosandNeg<-c(1, 1, -2)
PosvsNeg<-c(1, -1, 0)

contrasts(attitudeLong$Imagery)<-cbind(NeutvsPosandNeg, PosvsNeg)

WatervsBeerandWine<-c(1, 1, -2)
BeervsWine<-c(1, -1, 0)

contrasts(attitudeLong$Drink)<-cbind(WatervsBeerandWine, BeervsWine)


attitudeModel<-ezANOVA(data = attitudeLong, dv = .(Drink_Rating), wid = .(Participant),
 between = .(Gender), within = .(Drink, Imagery), type = 3, detailed = TRUE)

attitudeModel
```

```
$ANOVA
                Effect DFn DFd        SSn      SSd          F            p p<.05        ges
1          (Intercept)   1  18 11218.0056 1396.500 144.592982 4.882273e-10     * 0.57243760
2               Gender   1  18   523.6056 1396.500   6.748944 1.818048e-02     * 0.05881553
3                Drink   2  36  2092.3444 3216.867  11.707728 1.211310e-04     * 0.19981813
4         Gender:Drink   2  36  4569.0111 3216.867  25.565934 1.231002e-07     * 0.35287631
5              Imagery   2  36 21628.6778 1354.533 287.417216 7.359051e-23     * 0.72077386
6       Gender:Imagery   2  36  1998.3444 1354.533  26.555419 8.216664e-08     * 0.19256985
7        Drink:Imagery   4  72  2624.4222 2411.000  19.593364 6.048650e-11     * 0.23851180
8 Gender:Drink:Imagery   4  72   495.6889 2411.000   3.700705 8.519272e-03     * 0.05585486

$`Mauchly's Test for Sphericity`
                Effect         W           p p<.05
3                Drink 0.5723497 0.008712091     *
4         Gender:Drink 0.5723497 0.008712091     *
5              Imagery 0.9646213 0.736263076
```

```
6       Gender:Imagery 0.9646213 0.736263076
7        Drink:Imagery 0.6085813 0.521119714
8 Gender:Drink:Imagery 0.6085813 0.521119714

$`Sphericity Corrections`
                Effect    GGe      p[GG]         p[GG]<.05   HFe        p[HF]          p[HF]<.05
3                Drink 0.7004516 8.280562e-04         * 0.7415211 6.352807e-04          *
4         Gender:Drink 0.7004516 5.979446e-06         * 0.7415211 3.503910e-06          *
5              Imagery 0.9658301 3.734096e-22         * 1.0798112 7.359051e-23          *
6       Gender:Imagery 0.9658301 1.294718e-07         * 1.0798112 8.216664e-08          *
7        Drink:Imagery 0.8128306 2.780779e-09         * 1.0132286 6.048650e-11          *
8 Gender:Drink:Imagery 0.8128306 1.429430e-02         * 1.0132286 8.519272e-03          *
```

The results of Mauchly's sphericity test are different from the example in Chapter 13, because the between-group factor is now being accounted for by the test. The main effect of drink still significantly violates the sphericity assumption (W = 0.572, p < .01) but the main effect of imagery no longer does. Therefore, the F-value for the main effect of drink (and its interaction with the between-group variable **gender**) needs to be corrected for this violation.

The summary output of the repeated-measures effects in the ANOVA

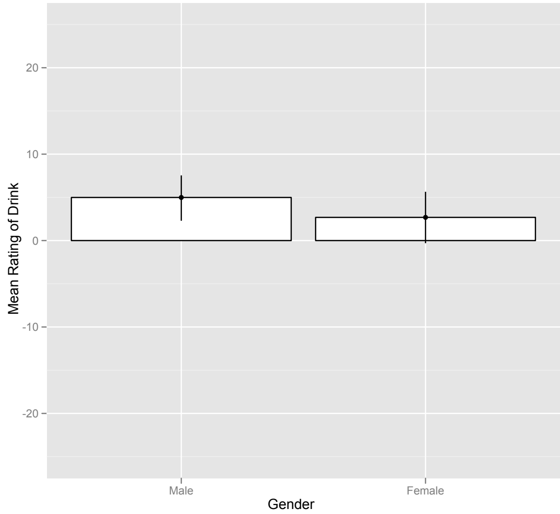
The table format is the same as for the previous example, except that the interactions between gender and the repeated-measures effects are included also. We would expect to still find the effects that were previously present (in a balanced design, the inclusion of an extra variable should not affect these effects). By looking at the significance values it is clear that this prediction is true: there are still significant effects of the type of drink used, the type of imagery used, and the interaction of these two variables.

In addition to the effects already described, we find that gender interacts significantly with the type of drink used (men and women respond differently to beer, wine and water regardless of the context of the advert). There is also a significant interaction of gender and imagery (so, men and women respond differently to positive, negative and neutral imagery regardless of the drink being advertised). Finally, the three-way interaction between gender, imagery and drink is significant, indicating that the way in which imagery affects responses to different types of drinks depends on whether the subject is male or female. The effects of the repeated-measures variables have been outlined in Chapter 13 and the pattern of these responses will not have changed, so rather than repeat myself, I will concentrate on the new effects and the forgetful reader should look back at Chapter 13!

The effect of gender

We can report that there was a significant main effect of gender, $F(1, 18) = 6.75$, p < .05. This effect tells us that if we ignore all other variables, male subjects' ratings were significantly different than those of females. If you look back at the descriptives output above, it is clear from the means that men's ratings were generally significantly more positive than women's. Therefore, men gave more positive ratings than women regardless of the drink being advertised and the type of imagery used in the advert. We could also draw a bar graph of the mean ratings of men and women by executing:

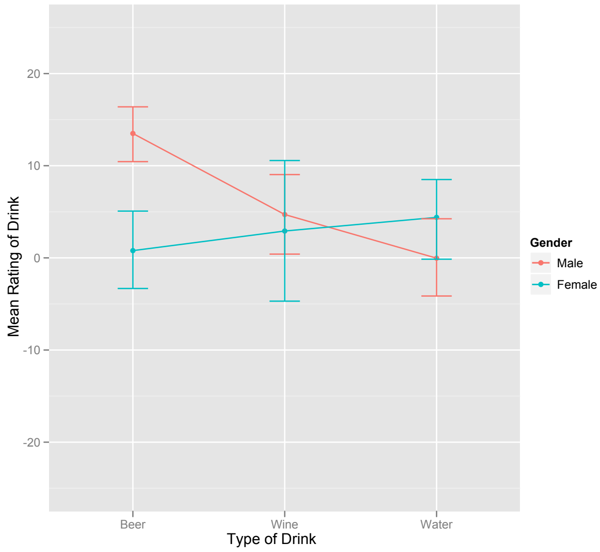
```
genderBar <- ggplot(attitudeLong, aes(Gender, Drink_Rating))
 genderBar + stat_summary(fun.y = mean, geom = "bar", fill = "White", colour =
"Black") + stat_summary(fun.data = mean_cl_boot, geom = "pointrange") + labs(x =
"Gender", y = "Mean Rating of Drink") + scale_y_continuous(limits = c(-25,25))
```



The interaction between gender and drink

Gender interacted in some way with the type of drink used as a stimulus. Remembering that the effect of drink violated sphericity, we must report the Greenhouse–Geisser-corrected value (ε = .57) for this interaction with the between-group factor. From the summary table we should report that there was a significant interaction between the type of drink used and the gender of the subject $F(1.14, 20.59) = 25.57$, $p < .001$. This effect tells us that the type of drink being advertised had a different effect on men and women. We can draw an interaction graph by executing:

```
GenderDrink <- ggplot(attitudeLong, aes(Drink, Drink_Rating, colour = Gender))
GenderDrink + 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 = "Type of Drink", y = "Mean Rating of Drink",
 colour = "Gender") + scale_y_continuous(limits = c(-25,25))
```



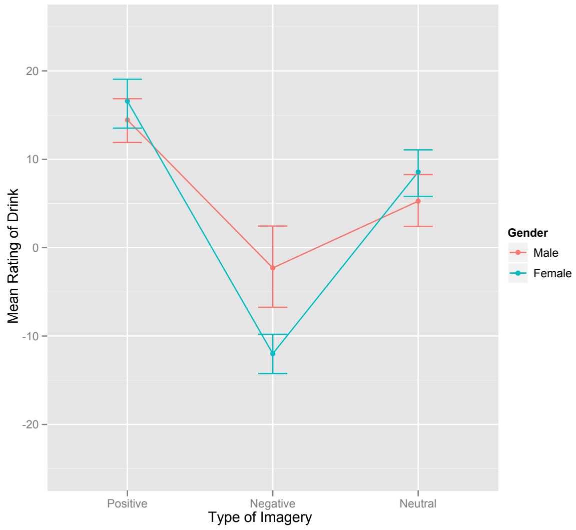
The graph clearly shows that male and female ratings are very similar for wine and water, but men seem to rate beer more highly than women — regardless of the type of imagery used. We could interpret this interaction as meaning that the type of drink being advertised influenced ratings differently in men and women. Specifically, ratings were similar for wine and water, but males rated beer higher than women.

The interaction between gender and imagery

Gender interacted in some way with the type of imagery used as a stimulus. The effect of imagery did not violate sphericity, so we can report the uncorrected *F*-value. From the summary table we should report that there was a significant interaction between the type of imagery used and the gender of the subject ($F(2, 36) = 26.55$, $p < .001$). We can plot an interaction graph by executing:

```
GenderImagery <- ggplot(attitudeLong, aes(Imagery, Drink_Rating, colour = Gender))
GenderImagery + 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 = "Type of Imagery", y = "Mean Rating of Drink",
colour = "Gender") + scale_y_continuous(limits = c(-25,25))
```

The resulting graph clearly shows that male and female ratings are very similar for positive and neutral imagery, but men seem to be less affected by negative imagery than women — regardless of the drink in the advert. To interpret this finding more fully, we should consult the contrasts for this interaction.



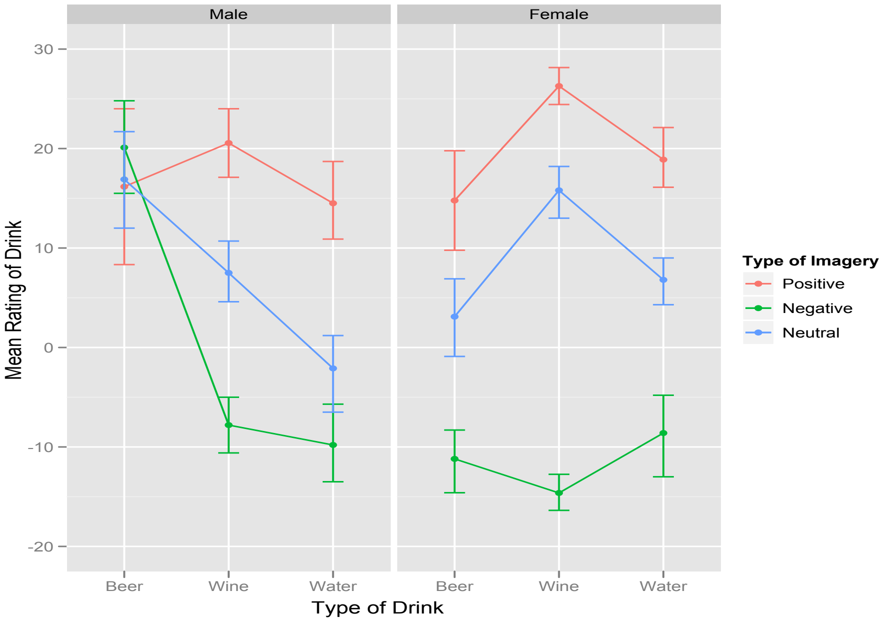
The interaction between drink and imagery

The interpretation of this interaction between drink and imagery is the same as for the two-way ANOVA (see Chapter 13). You may remember that the interaction reflected the fact that negative imagery has a different effect to both positive and neutral imagery (because it decreased ratings rather than increasing them).

The interaction between gender, drink and imagery

The three-way interaction between gender, drink and imagery tells us whether the drink by imagery interaction is the same for men and women (i.e. whether the combined effect of the type of drink and the imagery used is the same for male subjects as for female subjects). We can conclude that there is a significant three-way drink × imagery × gender interaction, $F(4, 72) = 3.70$, $p < .01$. We can plot a three-way interaction line graph by executing (we have already produced a boxplot of the three-way interaction earlier, but we could plot a line graph too):

```
GenderDrinkImagery <- ggplot(attitudeLong, aes(Drink, Drink_Rating, colour = Imagery))
GenderDrinkImagery + stat_summary(fun.y = mean, geom = "point") + stat_summary(fun.y =
mean, geom = "line", aes(group= Imagery)) + stat_summary(fun.data = mean_cl_boot, geom
= "errorbar", width = 0.2) + labs(x = "Type of Drink", y = "Mean Rating of Drink",
colour = "Type of Imagery") + scale_y_continuous(limits = c(-20,30)) +
facet_wrap(~Gender)
```



The resulting graph above shows the imagery by drink interaction for men and women separately. The male graph shows that when positive imagery is used, men generally rated all three drinks positively. This pattern is true of women also (the line representing positive imagery is above the other two lines). When neutral imagery is used, men rate beer very highly, but rate wine and water fairly neutrally. Women, on the other hand rate beer and water neutrally, but rate wine more positively (in fact, the pattern of the positive and neutral imagery lines show that women generally rate wine slightly more positively than water and beer). So, for neutral imagery men still rate beer positively, and women still rate wine positively. For the negative imagery, the men still rate beer very highly, but give low ratings to the other two types of drink. So, regardless of the type of imagery used, men rate beer very positively (if you look at the graph you'll note that ratings for beer are virtually identical for the three types of imagery). Women, however, rate all three drinks very negatively when negative imagery is used. The three-way interaction is, therefore, likely to reflect these sex differences in the interaction between drink and imagery. Specifically, men seem fairly immune to the effects of imagery when beer is being used as a stimulus, whereas women are not. The contrasts will show up exactly what this interaction represents; we will have a look at these below when we run the same analysis using *lme()*.

Using a multilevel model

Before we build the model we need to set some contrasts. If we look at the first variable, **Imagery**, there were three conditions: Positive, Negative and Neutral. In many ways it makes sense to compare the positive and negative conditions to the neutral condition. To do this we need to code the baseline category (neutral) as 0 for all contrasts (that's how **R** knows it is the baseline). Then for one of the contrasts we assign a 1 to positive and in the other we assign a 1 to negative:

```
PosvsNeut<-c(1, 0, 0)
NegvsNeut<-c(0, 1, 0)

contrasts(attitudeLong$Imagery)<-cbind(PosvsNeut, NegvsNeut)
```

If we look at the second variable, **drink**, there were three conditions: Beer, Wine and Water. it makes sense to compare the beer and wine conditions to the water condition, which acts as a neutral condition. To do this we need to code the baseline category (water) as 0 for all contrasts. Then for one of the contrasts we assign a 1 to Beer and in the other we assign a 1 to Wine:

```
BeervsWater<-c(1, 0, 0)
WinevsWater<-c(0, 1, 0)
```

```
contrasts(attitudeLong$Drink)<-cbind(BeervsWater, WinevsWater)
```

Since **Gender** only has two conditions, there is only one way in which we can compare them:

```
contrasts(attitudeLong$Gender)<-c(1, 0)
```

Now let's build the model starting with a baseline and adding one predictor at a time. We can do this by executing the following commands:

```
baseline<-lme(Drink_Rating ~ 1, random = ~1|Participant/Drink/Imagery, data =
 attitudeLong, method = "ML")
ImageryM<-update(baseline, .~. + Imagery)
DrinkM<-update(ImageryM, .~. + Drink)
GenderM<-update(DrinkM, .~. + Gender)
Imagery_Gender<-update(GenderM, .~. + Imagery:Gender)
Drink_Gender<-update(Imagery_Gender, .~. + Drink:Gender)
Imagery_Drink<-update(Drink_Gender, .~. + Imagery:Drink)
attitudeModel<-update(Imagery_Drink, .~. + Imagery:Drink:Gender)
```

To compare these models we can list them in the order in which we want them compared in the *anoca()* function:

```
anova(baseline, ImageryM, DrinkM, GenderM, Imagery_Gender, Drink_Gender, Imagery_Drink,
 attitudeModel)
```

Executing the above command produces:

```
               Model df      AIC      BIC    logLik   Test   L.Ratio p-value
baseline           1  5 1503.590 1519.555 -746.7950
ImageryM           2  7 1358.242 1380.592 -672.1208 1 vs 2 149.34836  <.0001
DrinkM             3  9 1350.529 1379.265 -666.2644 2 vs 3  11.71292  0.0029
GenderM            4 10 1349.201 1381.131 -664.6007 3 vs 4   3.32738  0.0681
Imagery_Gender     5 12 1322.624 1360.939 -649.3120 4 vs 5  30.57743  <.0001
Drink_Gender       6 14 1285.324 1330.026 -628.6621 5 vs 6  41.29970  <.0001
Imagery_Drink      7 18 1235.738 1293.212 -599.8692 6 vs 7  57.58592  <.0001
attitudeModel      8 22 1228.898 1299.143 -592.4492 7 vs 8  14.83998  0.0050
```

The output above shows similar results as those gained from using *ezANOVA* above. All the effects are significant except for the main effect of **Gender** which was non-significant, $p > .05$. This non-significant result suggests that when ignoring imagery and drink type, men and women did not differ in their ratings of the drinks; however, this effect was significant in the *ezANOVA* and it was found that on the whole men gave significantly higher ratings than women. If we look closely at the significance values of the main effect of **Gender** for *ezANOVA* and the *lme*, they are actually very close and almost the same! Using *ezANOVA* we get .0588, and for *lme()* we get .0681. In fact if we round up .0588 to two decimal places we get .06. This just shows that significance testing is not really that accurate.

We can see the parameter estimates of the model and the contrasts we requested by executing:

```
summary(attitudeModel)
```

```
Fixed effects: Drink_Rating ~ Imagery + Drink + Gender + Imagery:Gender + Drink:Gender +
Imagery:Drink + Imagery:Drink:Gender

                                  Value Std.Error  DF   t-value p-value
(Intercept)                         6.8  2.274230 108  2.990023  0.0035
ImageryPosvsNeut                   13.5  2.640698 108  5.112284  0.0000
ImageryNegvsNeut                  -15.4  2.640698 108 -5.831791  0.0000
DrinkBeervsWater                   -3.7  3.216247  36 -1.150409  0.2576
DrinkWinevsWater                    9.0  3.216247  36  2.798292  0.0082
Gender1                            -8.9  3.216247  18 -2.767200  0.0127
PosvsNeut:Gender                    3.1  3.734511 108  0.830095  0.4083
NegvsNeut:Gender                    7.7  3.734511 108  2.061849  0.0416
BeervsWater:Gender                 22.7  4.548460  36  4.990700  0.0000
WinevsWater:Gender                  0.6  4.548460  36  0.131913  0.8958
PosvsNeut:DrinkBeervsWater          0.7  3.734511 108  0.187441  0.8517
NegvsNeut:DrinkBeervsWater          1.1  3.734511 108  0.294550  0.7689
```

```
PosvsNeut:DrinkWinevsWater              -0.9  3.734511 108 -0.240995  0.8100
NegvsNeut:DrinkWinevsWater             -16.6  3.734511 108 -4.445026  0.0000
PosvsNeut:DrinkBeervsWater:Gender       -9.4  5.281396 108 -1.779832  0.0779
NegvsNeut:DrinkBeervsWater:Gender        9.8  5.281396 108  1.855570  0.0662
PosvsNeut:DrinkWinevsWater:Gender       -0.9  5.281396 108 -0.170409  0.8650
NegvsNeut:DrinkWinevsWater:Gender        9.0  5.281396 108  1.704095  0.0912
```

Contrasts for repeated-measures variables

We requested simple contrasts for the **drink** variable (for which water was used as the control category) and for the **imagery** category (for which neutral imagery was used as the control category). The table above is the same as for the previous example, except that the added effects of **Gender** and its interaction with other variables are now included. For the **imagery** main effect, the first contrast compares positive to the baseline category (neutral) and reveals a significant effect, b = 13.5 , t(108) = 5.11, p < .001. The second contrast reveals a significant difference for the negative imagery condition compared to the neutral, b = –15.4, t(108) = –5.83, p < .001. For the main effect of **drink**, the first contrast compares beer against the base category (in this case the last category, water); this result is non-significant, b = –3.7, t(36) = –1.15, p > .05. The next contrast compares wine with the base category (water) and confirms the significant difference found when gender was not included as a variable in the analysis, b = 9.0, t(36) = 362.80, p < .01. No contrast was specified for gender.

Imagery × Gender interaction 1: positive vs. neutral, male vs. female

The first interaction term looks at positive imagery compared to neutral imagery, comparing male and female scores. This contrast is not significant b = 3.1, t(108) = 0.83, p > .05. This result tells us that ratings of drinks presented with positive imagery (relative to those presented with neutral imagery) were equivalent for males and females. This finding represents the fact that in the earlier graph of this interaction the squares and circles for both the positive and neutral conditions overlap (therefore male and female responses were the same).

Imagery × Gender interaction 2: negative vs. neutral, male vs. female

The second interaction term looks at negative imagery compared to neutral imagery, comparing male and female scores. This contrast is significant, b = 7.7, t(108) = 2.06, p < .05. This result tells us that the difference between ratings of drinks paired with negative imagery compared to neutral was different for men and women. Looking at the earlier graph of this interaction, this finding represents the fact that, for men, ratings of drinks paired with negative imagery were relatively similar to ratings of drinks paired with neutral. However, looking at the female ratings, drinks were rated much less favourably when presented with negative imagery than when presented with neutral imagery. Therefore, overall, the imagery × gender interaction has shown up a difference between males and females in terms of their ratings to drinks presented with negative imagery compared to neutral; specifically, men seem less affected by negative imagery.

Drink × Gender interaction 1: beer vs. water, male vs. female

The first interaction term looks at beer compared to water comparing male and female scores. This contrast is highly significant, b = 22.7, t(36) = 4.99, p < .001. This result tells us that the increased ratings of beer compared to water found for men are not found for women. So we can conclude that male ratings of beer (compared to water) were significantly greater than women's ratings of beer (compared to water).

Drink × gender interaction 2: wine vs. water, male vs. female

The second interaction term compares wine to water, contrasting male and female scores. There is no significant difference for this contrast, b = 0.6, t(36) = 0.13, p > .05, which tells us that the difference between ratings of wine compared to water in males is roughly the same as in females. Therefore, overall, the drink × gender interaction has shown up a difference between males and females in how they rate beer (regardless of the type of imagery used).

Imagery × drink × Gender interaction 1: positive vs. neutral imagery, beer vs. water, male vs. female

The first interaction term compares beer to water, when positive imagery is used compared to neutral in males compared to females, b = –9.4, t(108) = –1.78, p > .05. The non-significance of this contrast tells us that the difference in ratings when positive imagery is used compared to neutral imagery is roughly equal when beer is used as a stimulus and when water is used, and these differences are equivalent in male and female subjects.

Imagery × drink × Gender interaction 2: beer vs. water, negative vs. neutral imagery, male vs. female

The second interaction term looks at beer compared to water, when negative imagery is used compared to neutral. This contrast is just non-significant, b = 9.8, $t(108)$ = 1.86, p = .067. This result tells us that the difference in ratings between beer and water when negative imagery is used (compared to neutral imagery) is similar between men and women. If we look back at the plot of the three-way interaction, we see that ratings after negative imagery are always lower than ratings for neutral imagery except for men's ratings of beer, which are actually higher after negative imagery. As such, this contrast tells us that the interaction effect reflects a difference in the way males rate beer compared to females when negative imagery is used compared to neutral. Males and females are similar in their pattern of ratings for water but different in the way they rate beer.

Imagery × drink × Gender Interaction 3: positive vs. neutral imagery, wine vs. water, male vs. female

The third interaction term looks at wine compared to water, when positive imagery is used compared to neutral in males compared to females. This contrast is non-significant, b = –0.9, $t(108)$ = –0.17, p > .05. This result tells us that the difference in ratings when positive imagery is used compared to neutral imagery is roughly equal when wine is used as a stimulus and when water is used, and these differences are equivalent in male and female subjects.

Imagery × drink × Gender interaction 4: negative vs. neutral imagery, wine vs. water, male vs. female

The final interaction term looks at wine compared to water, when negative imagery is used compared to neutral. This contrast is very close to significance, b = 9.0, $t(108)$ = 1.70, p = .09. This result tells us that the difference in ratings between wine and water when negative imagery is used (compared to neutral imagery) is different between men and women (although this difference has not quite reached significance). If we look back at the three-way interaction graph, we see that ratings after negative imagery are always lower than ratings for neutral imagery, but for women rating wine the change is much more dramatic (the line is steeper). As such, this contrast tells us that the interaction effect reflects a difference in the way in which females rate wine differently than males when neutral imagery is used compared to when negative imagery is used. Males and females are similar in their pattern of ratings for water but different in the way they rate wine. It is noteworthy that this contrast was not significant using the usual .05 level; however, it is worth remembering that this cut-off point was set in a fairly arbitrary way, and so it is worth reporting these close effects and letting your reader decide whether they are meaningful or not. There is also a growing trend towards reporting effect sizes in preference to using significance levels.

Effect sizes

To calculate the effect sizes, we first execute the command from the chapter:

```
rcontrast<-function(t, df)
{r<-sqrt(t^2/(t^2 + df))
	print(paste("r = ", r))
	}
```

and then we can get the effect sizes by simply executing:

```
rcontrast(t, df)
```

We should really only quantify the highest-order interaction because other effects are not interesting given that the three-way interaction is significant. Therefore, we can get the effect sizes by executing *rcontrast()* for each of the four contrasts for the three-way interaction:

```
rcontrast(-1.779832, 108)
rcontrast(1.855570, 108)
```

```
rcontrast(-0.170409, 108)
rcontrast(-1.704095, 108)
```

```
> rcontrast(-1.779832, 108)
[1] "r =  0.168806630613793"
> rcontrast(1.855570, 108)
[1] "r =  0.175772395813633"
> rcontrast(-0.170409, 108)
[1] "r =  0.0163954096054545"
> rcontrast(-1.704095, 108)
[1] "r =  0.161815572836376"
```

In other words, we get:

- $r_{\text{Positive vs. Neutral, Beer vs. Water, Male vs. Female}} = .17$
- $r_{\text{Negative vs. Neutral, Beer vs. Water, Male vs. Female}} = .18$
- $r_{\text{Positive vs. Neutral, Wine vs. Water, Male vs. Female}} = .02$
- $r_{\text{Negative vs. Neutral, Wine vs. Water, Male vs. Female}} = .16$

We could report these results as follows:

✓ There were significant main effects of the type of imagery used, $\chi^2(2) = 149.35$, $p < .0001$, and the type of drink that was being rated, $\chi^2(2) = 11.71$, $p < .01$, on drink rating. However, the ratings from male and female participants were, in general, the same, $\chi^2(1) = 3.33$, $p > .05$.

✓ There were significant interaction effects of the type of imagery used and the gender of the participant, $\chi^2(2) = 30.58$, $p < .0001$, the type of drink being rated and the gender of the participant, $\chi^2(2) = 41.30$, $p < .0001$, and the type of imagery used and the type of drink being rated $\chi^2(4) = 57.59$, $p < .0001$.

✓ Most important, the drink × imagery × gender interaction was significant, $\chi^2(4) = 14.84$, $p < .01$. This indicates that the drink × imagery interaction described previously was different in male and female participants. Contrasts were used to break down this interaction; these compared male and females scores at each level of imagery compared to the baseline category of 'neutral' across each level of drink compared to the category of water. The first contrast revealed a non-significant difference between male and female responses when comparing beer to water, when positive imagery is used compared to neutral imagery, $b = –9.4$, $t(108) = –1.78$, $p > .05$, $r = .17$, and tells us that the difference in ratings when positive imagery is used compared to neutral imagery is roughly equal when beer is used as a stimulus and when water is used, and these differences are equivalent in male and female subjects. The second contrast looked for differences between males and females when comparing beer to water, when negative imagery is used compared to neutral. This contrast is just non-significant, $b = 9.8$, $t(108) = 1.86$, $p = .067$, $r = .18$, and tells us that the difference in ratings between beer and water when negative imagery is used (compared to neutral imagery) is similar between men and women. As such, this contrast tells us that the interaction effect reflects a difference in the way in which males rate beer compared to females when negative imagery is used compared to neutral. Males and females are similar in their pattern of ratings for water but different in the way in which they rate beer. The third contrast investigated differences between males and females when comparing wine to water, when positive imagery is used compared to neutral imagery. This contrast is non-significant, $b = –0.9$, $t(108) = –0.17$, $p > .05$, $r = .02$. This result tells us that the difference in ratings when positive imagery is used compared to neutral imagery is roughly equal when wine is used as a stimulus and when water is used, and these differences are equivalent in male and female subjects. The final contrast looked for differences between men and women when comparing wine to water, when negative imagery is used compared to neutral. This contrast is very close to significance, $b = 9.0$, $t(108) = 1.70$, $p = .09$, $r = .16$, and tells us that the interaction effect reflects a difference in the way in which females rate wine differently to males when neutral imagery is used compared to when negative imagery is used. Males and females are similar in their pattern of ratings for water but different in the way in which they rate wine.

Summary

These contrasts again tell us nothing about the differences between the beer and wine conditions (or the positive and negative conditions), and different contrasts would have to be run to find out more. However, what is clear so far is that differences exist between men and women in terms of their ratings of beer and wine. It seems as though men are relatively unaffected by negative imagery when it comes to beer. Likewise, women seem more willing to rate wine positively when neutral imagery is used than men do. What should be clear from this is that complex ANOVA in which several independent variables are used results in complex interaction effects that require a great deal of concentration to interpret (imagine interpreting a four-way interaction!). Therefore, it is essential to take a systematic approach to interpretation, and plotting graphs is a particularly useful way to proceed. It is also advisable to think carefully about the appropriate contrasts to use to answer the questions you have about your data. It is these contrasts that will help you to interpret interactions, so make sure you select sensible ones!

Task 2

- Text messaging is very popular among mobile phone owners, to the point that books have been published on how to write in text speak (BTW, hope u no wat I mean by txt spk). One concern is that children may use this form of communication so much that it will hinder their ability to learn correct written English. One concerned researcher conducted an experiment in which one group of children was encouraged to send text messages on their mobile phones over a six-month period. A second group was forbidden from sending text messages for the same period. To ensure that kids in this latter group didn't use their phones, this group were given armbands that administered painful shocks in the presence of microwaves (like those emitted from phones). There were 50 different participants: 25 were encouraged to send text messages, and 25 were forbidden. The outcome was a score on a grammatical test (as a percentage) that was measured both before and after the experiment. The first independent variable was, therefore, text message use (text messagers versus controls) and the second independent variable was the time at which grammatical ability was assessed (before or after the experiment). The data are in the file **TextMessages.dat**.

First of all, read in the data:

```
textMessages<-read.delim("Textmessages.dat", header = TRUE)
```

Next we need to set the categorical variable **Group** to be a factor:

```
textMessages$Group<-factor(textMessages$Group, levels = c(1:2), labels = c("Text Messagers", "Controls"))
```

The next thing we need to do is convert the data from the 'wide' format into a 'long' format so that we can analyse the data using *ezANOVA()* and *lme()*. We can convert the data to the long format using the *melt()* function:

```
textLong<-melt(textMessages, id = c("Participant", "Group"), measured = c("Baseline", "Six_months") )
```

Let's give the new variables some appropriate names:

```
names(textLong)<-c("Participant", "Group", "Time", "Grammar_Score")
```

If we now execute:

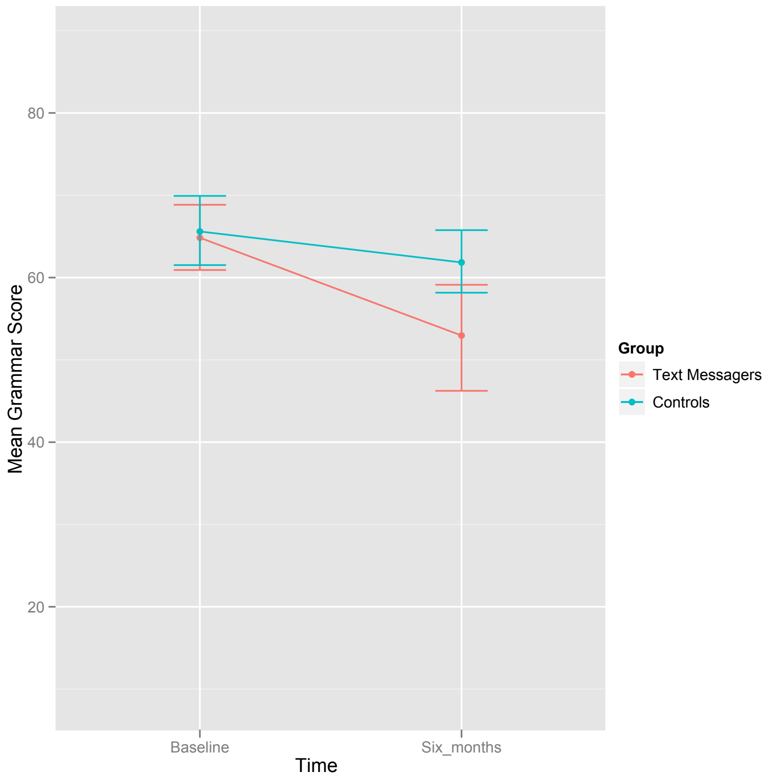
```
textLong
```

We can see how the new dataframe *textLong* looks (I have only included a small section of the data to save space).

```
Participant          Group       Time        Grammar_Score
1              1 Text Messagers   Baseline            52
51             1 Text Messagers Six_months            32
2              2 Text Messagers   Baseline            68
52             2 Text Messagers Six_months            48
3              3 Text Messagers   Baseline            85
53             3 Text Messagers Six_months            62
4              4 Text Messagers   Baseline            47
54             4 Text Messagers Six_months            16
5              5 Text Messagers   Baseline            73
55             5 Text Messagers Six_months            63
```

Next we need to explore the data. We can draw an interaction graph by executing:

```
textLine <- ggplot(textLong, aes(Time, Grammar_Score, colour = Group))
textLine + stat_summary(fun.y = mean, geom = "point") + stat_summary(fun.y = mean,
geom = "line", aes(group= Group)) + stat_summary(fun.data = mean_cl_boot, geom =
"errorbar", width = 0.2) + labs(x = "Time", y = "Mean Grammar Score", colour =
"Group")
```



Line chart (with error bars showing the standard error of the mean) of the mean grammar scores before and after the experiment for text messagers and controls

The resulting line chart (with error bars) shows the grammar data. The means before and after the experiment are connected by a line for the two groups (text message group and controls) separately. It's clear from this chart that in the text message group grammar scores went down dramatically over the six-month period in which they used their mobile phone. For the controls, their grammar scores also fell but much less dramatically.

Let's also have a look at the descriptive statistics by executing:

```
by(textLong$Grammar_Score, list(textLong$Time, textLong$Group), stat.desc, basic =
FALSE)
```

```
Baseline
```

```
: Text Messagers
      median         mean      SE.mean CI.mean.0.95          var      std.dev 
   64.000000    64.840000     2.135946     4.408377   114.056667    10.679732 
    coef.var 
    0.164709 
------------------------------------------------------------ 
: Six_months
: Text Messagers
      median         mean      SE.mean CI.mean.0.95          var      std.dev 
  58.0000000   52.9600000    3.2662313    6.7411700  266.7066667   16.3311563 
    coef.var 
   0.3083678 
------------------------------------------------------------ 
: Baseline
: Controls
      median         mean      SE.mean CI.mean.0.95          var      std.dev 
  65.0000000   65.6000000    2.1671794    4.4728385  117.4166667   10.8358971 
    coef.var 
   0.1651814 
------------------------------------------------------------ 
: Six_months
: Controls
      median         mean      SE.mean CI.mean.0.95          var      std.dev 
  62.0000000   61.8400000    1.8820910    3.8844450   88.5566667    9.4104552 
    coef.var 
   0.1521742
```

The output above shows the table of descriptive statistics from the two-way mixed ANOVA; the table has means split according to whether the people were in the text messaging group or the control group at baseline and six months. These means correspond to those plotted above.

Using ezANOVA()

Normally we would begin by setting some orthogonal contrasts so that we could use Type III sums of squares. However, because both of our independent variables, **Time** and **Group**, have only two levels we do not need to do this – there is only one way of comparing two groups and so orthogonal contrasts will be set automatically. We can therefore run the ANOVA by executing:

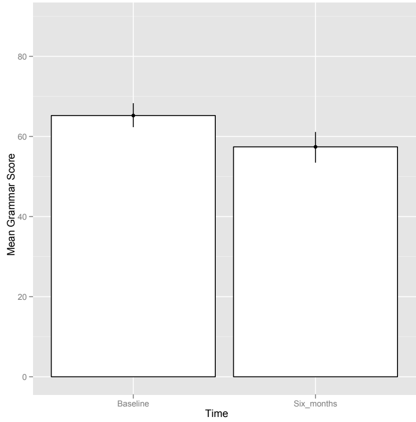
```
textModel<-ezANOVA(data = textLong, dv = .(Grammar_Score), wid = .(Participant), 
between = .(Group), within = .(Time), type = 3, detailed = TRUE)

textModel
```

```
$ANOVA
       Effect DFn DFd    SSn     SSd           F            p p<.05        ges
1 (Intercept)   1  48 375891.61 9334.08 1933.002211 1.944103e-40     * 0.96389066
2       Group   1  48    580.81 9334.08    2.986784 9.037565e-02       0.03961196
3        Time   1  48   1528.81 4747.60   15.456837 2.705478e-04     * 0.09793479
4  Group:Time   1  48    412.09 4747.60    4.166383 4.675237e-02     * 0.02843222
```

We know that when we use repeated measures we have to check the assumption of sphericity. In this case, we have only two levels of the repeated measure so the assumption of sphericity does not apply.

The output above shows the main ANOVA summary tables. Like any two-way ANOVA, we still have three effects to find: two main effects (one for each independent variable) and one interaction term. The main effect of **Time** is significant, so we can conclude that grammar scores were significantly affected by the time at which they were measured. The exact nature of this effect is easily determined because there were only two points in time (and so this main effect is comparing only two means). We can plot a graph of the main effect of **Time** by executing:

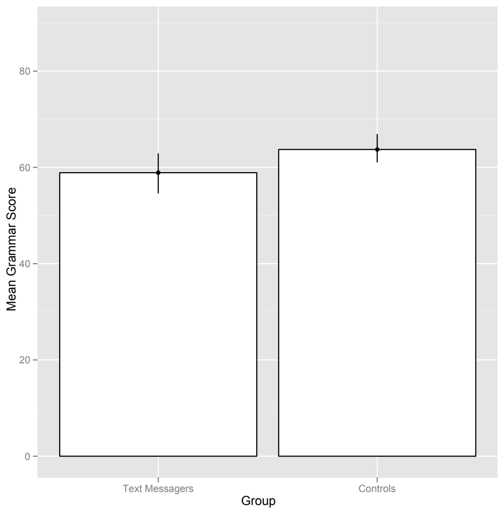
```
TimeBar <- ggplot(textLong, aes(Time, Grammar_Score))
 TimeBar + stat_summary(fun.y = mean, geom = "bar", fill = "White", colour = "Black") 
+ stat_summary(fun.data = mean_cl_boot, geom = "pointrange") + labs(x = "Time", y = 
"Mean Grammar Score")
```



The resulting graph shows that, before the experimental manipulation, grammar scores were higher before the experiment than after, meaning that the manipulation had the net effect of significantly reducing grammar scores. This main effect seems rather interesting until you consider that these means include both text messagers and controls. There are three possible reasons for the drop in grammar scores: (1) the text messagers got worse and are dragging down the mean after the experiment; (2) the controls somehow got worse; or (3) the whole group just got worse and it had nothing to do with whether the children text messaged or not. Until we examine the interaction, we won't see which of these is true.

The main effect of **Group** is shown by the *F*-ratio in the second table above. The probability associated with this *F*-ratio is .09, which is just above the critical value of .05. Therefore, we must conclude that there was no significant main effect on grammar scores of whether children text messaged or not. Again, this effect seems interesting enough and mobile phone companies might certainly chose to cite it as evidence that text messaging does not affect your grammatical ability. However, remember that this main effect ignores the time at which grammatical ability is measured. It just means that if we took the average grammar score for text messagers (that's including their score both before and after they started using their phone), and compared this to the mean of the controls (again including scores before and after) then these means would not be significantly different. We can plot a error bar graph of the main effect of **Group** by executing:

```
GroupBar <- ggplot(textLong, aes(Group, Grammar_Score))
 GroupBar + stat_summary(fun.y = mean, geom = "bar", fill = "White", colour = "Black")
+ stat_summary(fun.data = mean_cl_boot, geom = "pointrange") + labs(x = "Group", y =
"Mean Grammar Score")
```



The resulting graph shows that when you ignore the time at which grammar was measured, the controls have slightly better grammar than the text messagers, but not significantly so.

Main effects are not always that interesting and should certainly be viewed in the context of any interaction effects. The interaction effect in this example is shown by the *F*-ratio in the row labelled *Group:Time*, and because the probability of obtaining a value this big by chance is .047, which is just less than the criterion of .05, we can say that there is a significant interaction between the time at which grammar was measured and whether or not children were allowed to text message within that time. The mean ratings in all conditions help us to interpret this effect. The significant interaction tells us that the change in grammar scores was significantly different in text messagers compared to controls. Looking at the interaction graph that we plotted earlier we can see that although grammar scores fell in controls, the drop was much more marked in the text messagers; so, text messaging does seem to ruin your ability at grammar compared to controls.[2]

Writing the result

We can report the effects as follows:

- ✓ The results show that the grammar ratings at the end of the experiment were significantly lower than those at the beginning of the experiment, $F(1, 48) = 15.46$, $p < .001$.
- ✓ The main effect of group on the grammar scores was non-significant, $F(1, 48) = 2.99$, *ns*. This indicated that when the time at which grammar was measured is ignored, the grammar ability in the text message group was not significantly different than the controls.
- ✓ The time × group interaction was significant, $F(1, 48) = 4.17$, $p < .05$, indicating that the change in grammar ability in the text message group was significantly different from the change in the control groups. These findings indicate that although there was a natural decay of grammatical ability over time (as shown by the controls) there was a much stronger effect when participants were encouraged to use text messages. This shows that using text messages accelerates the inevitable decline in grammatical ability.

Using lme()

Before we build the model we would normally set some contrasts. However, in this example it would be pointless since both the independent variables have only two levels, therefore we do not need to worry about setting any contrasts.

Now let's build the model starting with a baseline and adding one predictor at a time. We can do this by executing the following commands:

```
baseline<-lme(Grammar_Score ~ 1, random = ~1|Participant/Time/Group, data = textLong, 
method = "ML")
TimeM<-update(baseline, .~. + Time)
GroupM<-update(TimeM, .~. + Group)
textModel<-update(GroupM, .~. + Time:Group)

anova(baseline, TimeM, GroupM, textModel)
```

We obtain the following output:

```
          Model df      AIC      BIC    logLik   Test   L.Ratio  p-value
baseline      1  5 803.0823 816.1081 -396.5411
TimeM         2  6 792.1066 807.7377 -390.0533 1 vs 2 12.975657  0.0003
GroupM        3  7 791.0884 809.3246 -388.5442 2 vs 3  3.018272  0.0823
textModel     4  8 788.9265 809.7679 -386.4633 3 vs 4  4.161864  0.0413
```

which shows similar results to those gained from using *ezANOVA* above. All the effects are significant, except for the main effect of **Group**. This non-significant result suggests that when

---

[2] It's interesting that the control group means dropped too. This could be because the control group were undisciplined and still used their mobile phones, or it could just be that the education system in this country is so underfunded that there is no one to teach English anymore!

ignoring the time at which the grammar test was taken, text messagers and controls did not differ in their grammar scores.

We can see the parameter estimates of the model by executing:

```
summary(textModel)
```

```
                              Value  Std.Error DF   t-value  p-value
(Intercept)                   64.84  2.422265 48 26.768337  0.0000
Time                         -11.88  2.812946 48 -4.223330  0.0001
Group                          0.76  3.425600 48  0.221859  0.8254
Time:Group                     8.12  3.978107 48  2.041172  0.0468
```

The above output is not that interesting. This is because each independent variable had only two levels and so the parameter estimates do not provide any additional information to the output from the ANOVA above. However, we can use the degrees of freedom and *t*-value to calculate the effect size of the two-way interaction between **Group** and **Time**.

To calculate the effect sizes, you first need to execute the command from the book chapter (but only if you haven't done this already) and then simply execute:

```
rcontrast(2.041172, 48)
```

```
> rcontrast(2.041172  , 48)
[1] "r =  0.282607861573644"
```

In other words, we get:

- $r_{\text{Baseline vs. Six months, Text messagers vs. non text messagers}}$ = .28

Reporting results of a multilevel model

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

- ✓ The results show that the grammar ratings at the end of the experiment were significantly lower than those at the beginning of the experiment, $\chi^2(1)$ = 12.98, $p < .001$.
- ✓ The main effect of group on the grammar scores was non-significant, $\chi^2(1)$ = 3.02, $p > .05$. This indicated that when the time at which grammar was measured is ignored, the grammar ability in the text message group was not significantly different than the controls.
- ✓ The time × group interaction was significant, $\chi^2(1)$ = 4.16, $p < .05$, r = .28, indicating that the change in grammar ability in the text message group was significantly different from the change in the control groups. These findings indicate that although there was a natural decay of grammatical ability over time (as shown by the controls), there was a much stronger effect when participants were encouraged to use text messages. This shows that text messaging accelerates the inevitable decline in grammatical ability.

Task 3

- A researcher was interested in the effects on people's mental health of participating in *Big Brother* (see Chapter 1 if you don't know what *Big Brother* is). The researcher hypothesized that that they start off with personality disorders that are exacerbated by being forced to live with people as attention seeking as themselves. To test this hypothesis, she gave eight contestants a questionnaire measuring personality disorders before they entered the house, and again when they left the house. A second group of eight people acted as a waiting list control. These people were short-listed to go into the house, but never actually made it. They too were given the questionnaire at the same points in time as the contestants. The data are in **BigBrother.dat**. Conduct a mixed ANOVA on the data.

First of all, as always, read in the data:

```
bigBrother<-read.delim("BigBrother.dat", header = TRUE)
```

Then set **bb** to be a factor:

```
bigBrother$bb<-factor(bigBrother$bb, levels = c(0:1), labels = c("No Treatment Control", "Big Brother Contestant"))
```

Next, we need to convert the dataframe from the wide format to a long format:

```
brotherLong<-melt(bigBrother, id = c("Participant", "bb"), measured = c("time1", "time2") )
```

We can then give our newly created variables appropriate names by executing:

```
names(brotherLong)<-c("Participant", "Group", "Time", "Personality_Score")
```

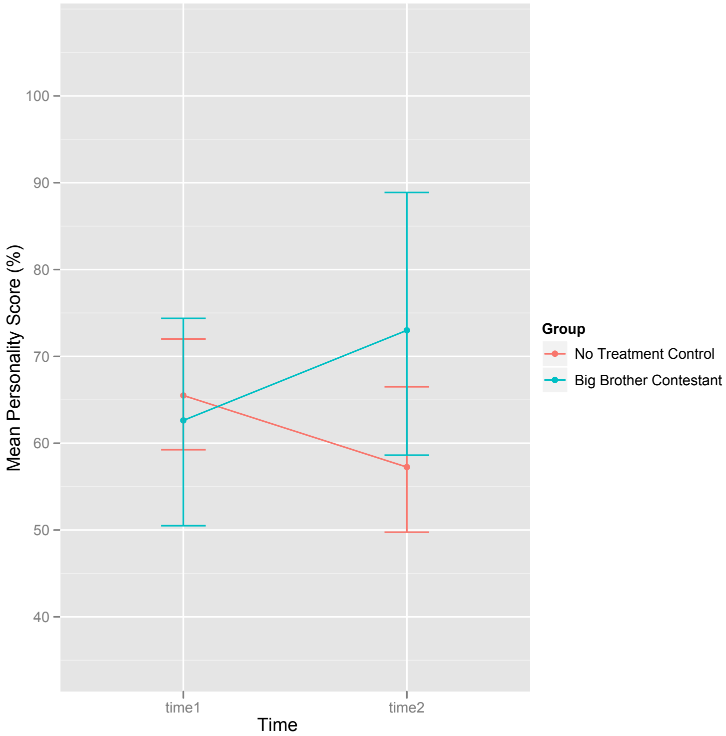
The resulting dataframe *brotherLong* should look like this (I have only included a small section to save space):

```
Participant       Group              Time       Personality_Score
1               No Treatment    Control time1       65
1               No Treatment    Control time2       50
2               No Treatment    Control time1       74
2               No Treatment    Control time2       47
3               No Treatment    Control time1       60
3               No Treatment    Control time2       52
4               No Treatment    Control time1       63
4               No Treatment    Control time2       57
5               No Treatment    Control time1       66
5               No Treatment    Control time2       51
6               No Treatment    Control time1       84
```

Exploring data

Let's draw a line graph of the two-way interaction between **Group** and **Time**:

```
PersonalityTime <- ggplot(brotherLong, aes(Time, Personality_Score, colour = Group))
PersonalityTime + stat_summary(fun.y = mean, geom = "point") + stat_summary(fun.y = mean, geom = "line", aes(group= Group)) + stat_summary(fun.data = mean_cl_boot, geom = "errorbar", width = 0.2) + labs(x = "Time", y = "Mean Personality Score (%)", colour = "Group")
```



Error line chart of the mean personality disorder score before entering and after leaving the *Big Brother* house

Looking at the line graph above, it seems that before entering the *Big Brother* house (time 1), *Big Brother* contestants and controls have similar borderline personality disorder (BPD) scores. However, at time 2 (after being in the *Big Brother* house) the BPD score of the *Big Brother* contestants has increased, whereas the BPD score of the no treatment controls has decreased. This suggests a significant interaction may exist between **Group** and **Time.**

We can also have a look at the descriptive statistics using the *by()* function:

```
by(brotherLong$Personality_Score, list(brotherLong$Time, brotherLong$Group),
stat.desc, basic = FALSE)
```

```
: time1
: No Treatment Control
      median         mean      SE.mean CI.mean.0.95          var      std.dev
  64.0000000   65.5000000    3.6105006    8.5374772  104.2857143   10.2120377
    coef.var
   0.1559090
------------------------------------------------------------
: time2
: No Treatment Control
      median         mean      SE.mean CI.mean.0.95          var      std.dev
  51.5000000   57.2500000    4.5503140   10.7597827  165.6428571   12.8702314
    coef.var
   0.2248075
------------------------------------------------------------
: time1
: Big Brother Contestant
      median         mean      SE.mean CI.mean.0.95          var      std.dev
  66.0000000   62.6250000    6.7000466   15.8430928  359.1250000   18.9505937
    coef.var
   0.3026043
------------------------------------------------------------
: time2
: Big Brother Contestant
      median         mean      SE.mean CI.mean.0.95          var      std.dev
  68.5000000   73.0000000    8.3430895   19.7282718  556.8571429   23.5978207
    coef.var
   0.3232578
```

The output above shows the table of descriptive statistics from the two-way mixed ANOVA; these means correspond to those plotted above.

We know that when we use repeated measures we have to check the assumption of sphericity. However, we also know that for sphericity to be an issue we need at least three conditions. We have only two conditions here, so sphericity does not need to be tested.

Using ezANOVA()

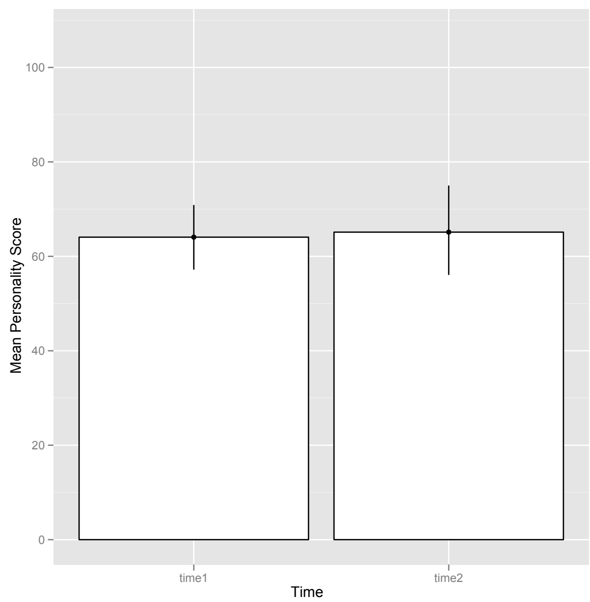
Normally we would begin by setting some orthogonal contrasts so that we could use Type III sums of squares. However, because both of our independent variables, **Time** and **Group**, have only two levels we do not need to do this – there is only one way of comparing two groups and so orthogonal contrasts will be set automatically. We can therefore run the ANOVA by executing:

```
brotherModel<-ezANOVA(data = brotherLong, dv = .(Personality_Score), wid =
.(Participant),  between = .(Group), within = .(Time), type = 3, detailed = TRUE)

 brotherModel
```

```
$ANOVA
       Effect DFn DFd          SSn      SSd            F            p p<.05
1 (Intercept)   1  14 133515.28125 6942.688 269.23492344 1.543839e-10     *
2       Group   1  14    331.53125 6942.688   0.66853614 4.272593e-01      
3        Time   1  14      9.03125 1358.688   0.09305856 7.648124e-01      
4  Group:Time   1  14    693.78125 1358.688   7.14876489 1.816739e-02     *
        ges
1 0.94146403
2 0.03840320
3 0.00108674
4 0.07712832
```

The output above shows the main ANOVA summary tables. Like any two-way ANOVA, we still have three effects to find: two main effects (one for each independent variable) and one interaction term. The main effect of time is not significant so we can conclude that BPD scores were significantly affected by the time at which they were measured. The exact nature of this effect is easily determined because there were only two points in time (and so this main effect is comparing only two means). We can plot an error bar graph of the main effect of **Time** by executing:

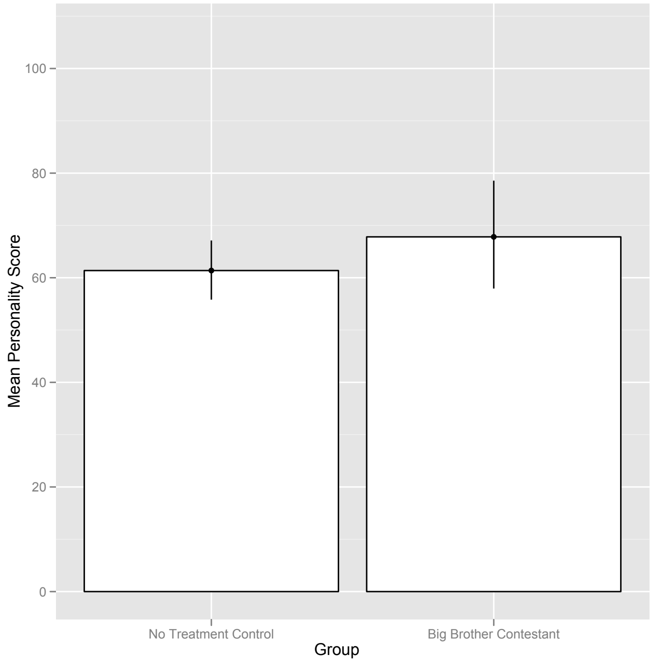
```
TimeBar <- ggplot(brotherLong, aes(Time, Personality_Score))
TimeBar + stat_summary(fun.y = mean, geom = "bar", fill = "White", colour = "Black") +
stat_summary(fun.data = mean_cl_boot, geom = "pointrange") + labs(x = "Time", y =
"Mean Personality Score")
```



The resulting graph shows that BPD scores were not significantly different after leaving the *Big Brother* house compared to before entering it.

The main effect of **Group** is shown by the *F*-ratio in the output table above. The probability associated with this *F*-ratio is .43, which is above the critical value of .05. Therefore, we must conclude that there was no significant main effect on BPD scores of whether the person was a *Big Brother* contestant or not. The graph shows that when you ignore the time at which BPD was measured, the contestants and controls are not significantly different:

```
GroupBar <- ggplot(brotherLong, aes(Group, Personality_Score))
 GroupBar + stat_summary(fun.y = mean, geom = "bar", fill = "White", colour = "Black")
+ stat_summary(fun.data = mean_cl_boot, geom = "pointrange") + labs(x = "Group", y =
"Mean Personality Score")
```



The interaction effect in this example is shown by the *F*-ratio in the row labelled *Group:Time*, and because the probability of obtaining a value this big is .018, which is less than the criterion of .05, we can say that there is a significant interaction between the time at which BPD was measured and whether or not the person was a contestant or not. The mean ratings in all conditions (and on the interaction graph) help us to interpret this effect. The significant interaction seems to indicate that for controls BPD scores went down (slightly) from before entering the house to after leaving it but for contestants these opposite is true: BPD scores increased over time.

Writing the results

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

- ✓ The main effect of group was not significant, $F(1, 14) = 0.67$, $p = .43$, indicating that across both time points BPD scores were similar in *Big Brother* contestants and controls.
- ✓ The main effect of time was not significant, $F(1, 14) = 0.09$, $p = .77$, indicating that across all participants BPD scores were similar before entering the house and after leaving it.
- ✓ The time × group interaction was significant, $F(1, 14) = 7.15$, $p < .05$, indicating that although BPD scores decreased for controls from before entering the house to after leaving it, scores *increased* for the contestants.

Using lme()

Before we build the model we would normally set some contrasts. However, in this example it would be pointless since both independent variables (**Group** and **Time**) have only two levels, therefore we do not need to worry about setting any contrasts.

Now let's build the model starting with a baseline and adding one predictor at a time. We can do this by executing the following commands:

```
baseline<-lme(Personality_Score ~ 1, random = ~1|Participant/Time/Group, data =
brotherLong, method = "ML")
TimeM<-update(baseline, .~. + Time)
GroupM<-update(TimeM, .~. + Group)
brotherModel<-update(GroupM, .~. + Time:Group)


anova(baseline, TimeM, GroupM, brotherModel)
```

We obtain the following output:

```
             Model df  AIC      BIC      logLik   Test  L.Ratio    p-value
baseline         1  5 276.4617 283.7904 -133.2309
TimeM            2  6 278.3915 287.1859 -133.1957 1 vs 2 0.070249  0.7910
GroupM           3  7 279.6451 289.9053 -132.8226 2 vs 3 0.746360  0.3876
brotherModel     4  8 275.0447 286.7706 -129.5224 3 vs 4 6.600387  0.0102
```

which shows similar results to those gained from using *ezANOVA* above.

We can see the parameter estimates of the model by executing:

```
summary(brotherModel)
                                           Value Std.Error DF   t-value p-value
(Intercept)                               65.500  6.087669 14 10.759455  0.0000
Time                                      -8.250  4.925675 14 -1.674897  0.1161
Group                                     -2.875  8.609264 14 -0.333943  0.7434
Time:Group                                18.625  6.965956 14  2.673718  0.0182
```

The above output is not that interesting, because each independent variable had only two levels and so the parameter estimates do not provide any additional information to the output from the ANOVA above. However, we can use the degrees of freedom and *t*-value to calculate the effect size of the significant two-way interaction between **Group** and **Time**.

To calculate the effect size, you first need to execute the command from the book chapter (but only if you haven't done this already) and then simply execute:

```
rcontrast(2.673718, 14)

> rcontrast(2.673718, 48)
[1] "r =  0.360037462825541"
```

In other words, we get:

- $r_{\text{Time1 vs. Time2, } \textit{Big Brother} \text{ Contestants vs. Controls}}$ = .36

Reporting results of a multilevel model

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

- ✓ The main effect of time was not significant, $\chi^2(1)$ = 0.07, p > .05, indicating that across all participants BPD scores were similar before entering the house and after leaving it.
- ✓ The main effect of group was not significant, $\chi^2(1)$ = 0.74, p > .05, indicating that across both time points BPD scores were similar in *Big Brother* contestants and controls.
- ✓ The time × group interaction was significant, $\chi^2(1)$ = 6.60, p < .05, r = .36, indicating that although BPD scores decreased for controls from before entering the house to after leaving it, scores *increased* for the contestants.

Task 4

- In this chapter we did a robust analysis on some data about how people's profile pictures on social networking sites affect their friend requests. Reanalyse these data using non-robust analysis.

First of all, as always, read in the data:

```
pictureData<-read.delim("ProfilePicture.dat", header = TRUE)
```

Next, we need to convert the dataframe from the wide format to a long format:

```
pictureLong<-melt(pictureData, id = c("case", "relationship_status"), measured =
c("couple", "alone") )
```

We can then give our newly created variables appropriate names by executing:

```
names(pictureLong)<-c("Case", "Relationship_Status", "Photo", "Friend_Requests")
```

If we order the dataframe by **Case** we can check that we have created the long dataframe correctly:

```
pictureLong<-pictureLong[order(pictureLong$Case),]
```

After executing the above command execute:

```
pictureLong
```

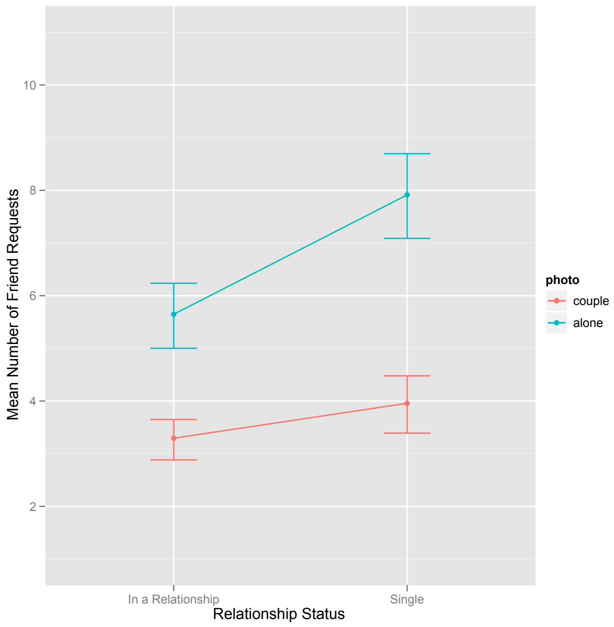
which produces (I have only included a small section of the dataframe to save space):

```
     Case Relationship_Status  Photo Friend_Requests
1      1   In a Relationship couple               4
41     1   In a Relationship  alone               4
2      2   In a Relationship couple               4
42     2   In a Relationship  alone               6
3      3   In a Relationship couple               4
43     3   In a Relationship  alone               7
4      4   In a Relationship couple               3
44     4   In a Relationship  alone               5
5      5   In a Relationship couple               4
45     5   In a Relationship  alone               3
6      6   In a Relationship couple               2
46     6   In a Relationship  alone               5
```

Exploring data

Let's draw a line graph of the two-way interaction between **Relationship_Status** and **Photo**:

```
profileLine <- ggplot(pictureLong, aes(Relationship_Status, Friend_Requests, colour =
Photo))
profileLine + stat_summary(fun.y = mean, geom = "point") + stat_summary(fun.y = mean,
geom = "line", aes(group= Photo)) + stat_summary(fun.data = mean_cl_boot, geom =
"errorbar", width = 0.2) + labs(x = "Relationship Status", y = "Mean Number of Friend
Requests", colour = "photo")
```



Error line chart of the mean number of friends of single women and women in a relationship when displaying a photo of themselves alone and with a partner.

Looking at the line graph above, it seems that on the whole, single women receive more friend requests than women who are in a relationship. When displaying a photo of themselves alone rather than with a partner, the number of friend requests increases in both women in a relationship and single women. However, for single women this increase is greater than for women who are in a relationship.

We can also have a look at some descriptive statistics using the *by()* function:

```
by(pictureLong$Friend_Requests, list(pictureLong$Relationship_Status,
pictureLong$Photo), stat.desc, basic = FALSE)
```

```
: In a Relationship
: couple
      median         mean      SE.mean CI.mean.0.95          var      std.dev
   3.0000000    3.2941176    0.2058824    0.4364511    0.7205882    0.8488747
    coef.var
   0.2576941
-----------------------------------------------------------
: Single
: couple
      median         mean      SE.mean CI.mean.0.95          var      std.dev
   4.0000000    3.9565217    0.2845026    0.5900223    1.8616601    1.3644266
    coef.var
   0.3448551
-----------------------------------------------------------
: In a Relationship
: alone
      median         mean      SE.mean CI.mean.0.95          var      std.dev
   6.0000000    5.6470588    0.3314538    0.7026506    1.8676471    1.3666188
    coef.var
   0.2420054
-----------------------------------------------------------
: Single
: alone
      median         mean      SE.mean CI.mean.0.95          var      std.dev
   8.0000000    7.9130435    0.3971890    0.8237195    3.6284585    1.9048513
    coef.var
   0.2407230
```

The output above shows the table of descriptive statistics from the two-way mixed ANOVA; these means correspond to those plotted above.

We know that when we use repeated measures we have to check the assumption of sphericity. However, we also know that for sphericity to be an issue we need at least three conditions. We have only two conditions here, so sphericity does not need to be tested.

Using *ezANOVA()*

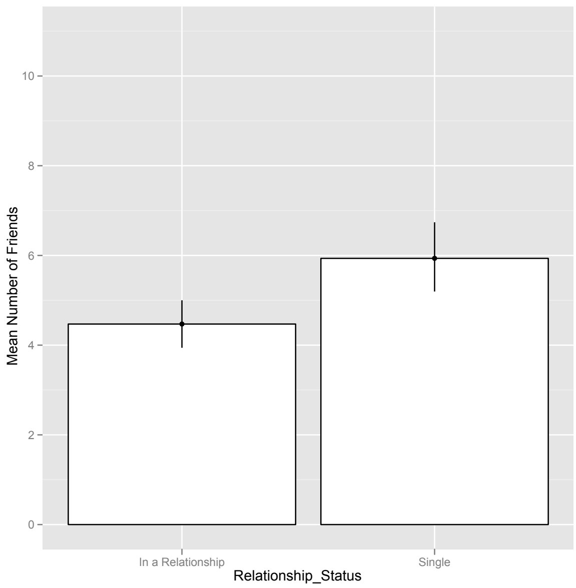
Normally we would begin by setting some orthogonal contrasts so that we could use Type III sums of squares. However, because both of our independent variables, **Photo** and **Relationship_Status**, have only two levels we do not need to do this – there is only one way of comparing two groups and so orthogonal contrasts will be set automatically. We can therefore run the ANOVA by executing:

```
pictureModel<-ezANOVA(data = pictureLong, dv = .(Friend_Requests), wid = .(Case),
between = .(Relationship_Status), within = .(Photo), type = 3, detailed = TRUE)

pictureModel
```

```
$ANOVA
                   Effect DFn DFd        SSn      SSd         F            p
1             (Intercept)   1  38 2257.81250 97.77494 877.49354 7.283934e-28
2     Relationship_Status   1  38   41.91256 97.77494  16.28922 2.537511e-04
3                   Photo   1  38  214.51250 64.41944 126.53751 1.174531e-13
4 Relationship_Status:Photo   1  38   12.56806 64.41944   7.41370 9.720279e-03
  p<.05        ges
1     * 0.93297772
2     * 0.20534610
3     * 0.56944143
4     * 0.07191513
```

The output above shows the main ANOVA summary tables. Like any two-way ANOVA, we still have three effects to find: two main effects (one for each independent variable) and one interaction term. The main effect of **Relationship_Status** is significant, so we can conclude that the number of friend requests were significantly affected by the relationship status of the woman. The exact nature of this effect is easily determined because there were only two levels of relationship status (and so this main effect is comparing only two means). We can plot an error bar graph of the main effect of **Relationship_Status** by executing:

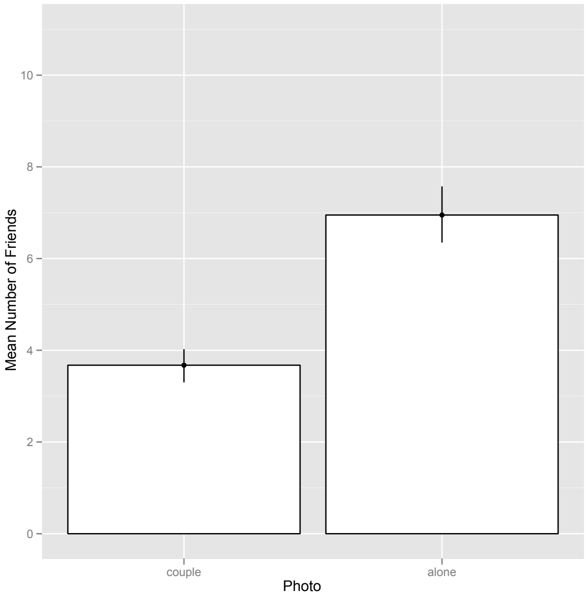
```
RelationshipBar <- ggplot(pictureLong, aes(Relationship_Status, Friend_Requests))
RelationshipBar + stat_summary(fun.y = mean, geom = "bar", fill = "White", colour =
"Black") + stat_summary(fun.data = mean_cl_boot, geom = "pointrange") + labs(x =
"Relationship_Status", y = "Mean Number of Friends")
```



The resulting graph shows that the number of friend requests were significantly higher for single women compared to women who were in a relationship.

The main effect of **Photo** is shown by the *F*-ratio in the output table above. The probability associated with this *F*-ratio is much smaller than the critical value of .05. Therefore, we must conclude that there was a significant main effect on the number of friend requests of whether the person was alone in their profile picture or was with a partner. We can plot an error bar graph of the main effect of **Photo** by executing:

```
PhotoBar <- ggplot(pictureLong, aes(Photo, Friend_Requests))
PhotoBar + stat_summary(fun.y = mean, geom = "bar", fill = "White", colour = "Black")
+ stat_summary(fun.data = mean_cl_boot, geom = "pointrange") + labs(x = "Photo", y =
"Mean Number of Friends")
```



The resulting graph shows that when ignoring relationship status, the number of friend requests were significantly higher when the women were alone in their profile picture than when they were with a partner.

The interaction effect in this example is shown by the *F*-ratio in the row labelled *Relationship_Status:Photo*, and because the probability of obtaining a value this big is .0097, which is less than the criterion of .05, we can say that there is a significant interaction between the relationship status of women and whether they had a photo of themselves alone or with a partner. The mean ratings in all conditions (and on the interaction graph) help us to interpret this effect. The significant interaction seems to indicate that when displaying a photo of themselves alone rather than with a partner, the number of friend requests increases in both women in a relationship and single women. However, for single women this increase is greater than for women who are in a relationship.

Writing the results

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

- ✓ The main effect of relationship status was significant, $F(1, 38) = 16.29$, $p < .001$, indicating that single women received more friend requests than women who were in a relationship, regardless of their type of profile picture.
- ✓ The main effect of photo was significant, $F(1, 38) = 126.54$, $p < .001$, indicating that across all women, the number of friend requests was greater when displaying a photo alone rather than with a partner.
- ✓ The relationship status × photo interaction was significant, $F(1, 38) = 7.41$, $p < .01$, indicating that although number of friend requests increased in all women when they displayed a photo of themselves alone compared to when they displayed a photo of themselves with a partner, this increase was significantly greater for single women than for women who were in a relationship.

Using lme()

Before we build the model we would normally set some contrasts. However, in this example it would be pointless since both independent variables (**Relationship_Status** and **Photo**) have only two levels, therefore we do not need to worry about setting any contrasts.

Now let's build the model starting with a baseline and adding one predictor at a time. We can do this by executing the following commands:

```
baseline<-lme(Friend_Requests ~ 1, random = ~1|Case/Relationship_Status/Photo, data =
pictureLong, method = "ML")
RelationshipM<-update(baseline, .~. + Relationship_Status)
PhotoM<-update(RelationshipM, .~. + Photo)
pictureModel<-update(PhotoM, .~. + Relationship_Status:Photo)
```

```
anova(baseline, RelationshipM, PhotoM, pictureModel)
```

```
              Model df   AIC      BIC    logLik   Test  L.Ratio p-value
baseline          1  5 371.7915 383.7016 -180.8957
RelationshipM     2  6 365.6109 379.9031 -176.8055 1 vs 2  8.18057  0.0042
PhotoM            3  7 302.9723 319.6464 -144.4861 2 vs 3 64.63864  <.0001
pictureModel      4  8 297.8432 316.8994 -140.9216 3 vs 4  7.12911  0.0076
```

The output above shows similar results to those gained from using *ezANOVA* above.

We can see the parameter estimates of the model by executing:

```
summary(pictureModel)
```

```
Fixed effects: Friend_Requests ~ Relationship_Status + Photo +
Relationship_Status:Photo
                              Value Std.Error DF  t-value p-value
(Intercept)                3.294118 0.3543126 38 9.297206  0.0000
Relationship_Status        0.662404 0.4672537 38 1.417654  0.1644
Photo                      2.352941 0.4465882 38 5.268705  0.0000
Relationship_Status:Photo  1.603581 0.5889430 38 2.722811  0.0097
```

The above output is not that interesting, this is because each independent variable had only two levels and so the parameter estimates do not provide any additional information to the output from the ANOVA above. However, we can use the degrees of freedom and *t*-value to calculate the effect size of the significant two-way interaction between **Relationship_Status** and **Photo**.

To calculate the effect sizes, you first need to execute the command from the book chapter (but only if you haven't done this already) and then simply execute:

```
rcontrast(2.722811, 38)
```

```
rcontrast(2.722811, 38)
[1] "r =  0.404039713368767"
```

In other words, we get:

- $R_{\text{In a Relationship vs. Single, Couple vs. Alone}}$ = .40

Reporting results of a multilevel model

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

- ✓ The main effect of relationship status was significant, $\chi^2(1) = 8.18$, $p < .01$, indicating that across all participants the number of friend requests was greater for single women than for women who were in a relationship.
- ✓ The main effect of photo was also significant, $\chi^2(1) = 64.64$, $p < .0001$, indicating that across all participants the number of friend requests was greater for women who displayed a photo of themselves alone rather than with a partner.
- ✓ The time × group interaction was significant, $\chi^2(1) = 7.13$, $p < .01$, $r = .40$, indicating that although the number of friend requests increased in all women when they displayed a photo of themselves alone compared to when they displayed a photo of themselves with a partner, this increase was significantly greater for single women than for women who were in a relationship.