Multivariate analysis of variance (MANOVA)

Self-test answers

- Use *ggplot2* to plot a scatterplot of the number of obsession-related actions (*x*-axis) against obsession-related thoughts (*y*-axis) for each treatment group (as separate panels).

```
ocdScatter <- ggplot(ocdData, aes(Actions, Thoughts))
```

```
ocdScatter + geom_point() + geom_smooth(method = "lm")+ labs(x = "Number of Obsession-Related Behaviours", y = "Number of Obsession-Related Thoughts") + facet_wrap(~Group, ncol = 3)
```

- Use *ggplot2* to plot a bar graph (with error bars) of the treatment group on the *x*-axis and different-coloured bars to represent the mean number of obsession-related thoughts and behaviours.

First, we need to restructure the data into long format:

```
ocdMelt<-melt(ocdData, id = c("Group"), measured = c("Actions", "Thoughts"))
```

```
names(ocdMelt)<-c("Group", "Outcome_Measure", "Frequency")
```

We can now do the bar chart using this long format data:

```
ocdBar <- ggplot(ocdMelt, aes(Group, Frequency, fill = Outcome_Measure))
```

```
ocdBar + stat_summary(fun.y = mean, geom = "bar", position = "dodge") +
stat_summary(fun.data = mean_cl_boot, geom = "errorbar",
position=position_dodge(width=0.90), width = 0.2) + labs(x = "Treatment Group", y = "Number of Thoughts/Actions", fill = "Outcome Measure") + scale_y_continuous(breaks = seq(0, 20, by = 2))
```

- Use *ggplot2* to plot boxplots of treatment group on the *x*-axis and obsession-related thoughts and actions displayed on the *y*-axis (in different colours).

```
ocdBoxplot <- ggplot(ocdMelt, aes(Group, Frequency, colour = Outcome_Measure))
```

```
ocdBoxplot + geom_boxplot() + labs(x = "Treatment Group", y = "Number of Thoughts/Actions", colour = "Outcome Measure") + scale_y_continuous(breaks = seq(0, 20, by = 2))
```

- Delete case 26 from the dataframe and redo the Shapiro test of multivariate normality.

To remove the outlier we can create a new dataframe called *ocdNoOutlier* that is based on the *ocdData* dataframe. Remembering that the square brackets mean we are selecting part of the dataframe and that anything before the comma selects rows, and anything after selects columns, the *[-26,]* in the following command deletes row 26 (the minus sign means 'delete') but retains all of the columns (there is nothing after the comma so everything is retained):

```
ocdNoOutlier<-ocdData[-26,]
```

The data look like this (note that case 26 is missing):

```
   Group Actions Thoughts
```

```
1     CBT        5        14
2     CBT        5        11
3     CBT        4        16
4     CBT        4        13
5     CBT        5        12
6     CBT        3        14
7     CBT        7        12
8     CBT        6        15
9     CBT        6        16
10    CBT        4        11
11     BT        4        14
12     BT        4        15
13     BT        1        13
14     BT        1        14
15     BT        4        15
16     BT        6        19
17     BT        5        13
18     BT        5        18
19     BT        2        14
20     BT        5        17
21     NT        4        13
22     NT        5        15
23     NT        5        14
24     NT        4        14
25     NT        6        13
27     NT        7        13
28     NT        4        16
29     NT        6        14
30     NT        5        18
```

We then need to extract the NT group data using the same command as in the book:

```
nt<-t(ocdNoOutlier[21:29, 2:3])
```

Note that the only difference compared to the book is that we have selected rows 21 to 29 rather than 21 to 30 because, of course, we have one row fewer than we had before.

We can then run the Shapiro-Wilk test on this variable:

```
mshapiro.test(nt)
```

The results show that removing the outlier does make the data multivariate normal because the *p*-value is not significant ($p = .208$):

```
        Shapiro-Wilk normality test

data:  Z
W = 0.8918, p-value = 0.208
```

- Why might the univariate tests be non-significant when the multivariate tests were significant?

Although the issue of power is complicated in MANOVA (see the book chapter), one reason why the multivariate statistics might be significant when the univariate tests are not is because the multivariate tests take account of the correlations between dependent variables, whereas the univariate tests do not. It's also worth remembering that the univariate and multivariate tests look at different things: the multivariate tests tell us whether groups can be discriminated based on a linear combination of the dependent variables, whereas the univariate tests tell us whether the groups can be discriminated by a single variable.

Oliver Twisted

Please Sir, can I have some more … maths?

Calculation of E^{-1}

$$E = \begin{pmatrix} 51 & 13 \\ 13 & 122 \end{pmatrix}$$

$$\text{determinant of } E, |E| = (51 \times 122) - (13 \times 13) = 6053$$

$$\text{matrix of minors for } E = \begin{pmatrix} 122 & 13 \\ 13 & 51 \end{pmatrix}$$

$$\text{pattern of signs for } 2 \times 2 \text{ matrix} = \begin{pmatrix} + & - \\ - & + \end{pmatrix}$$

$$\text{matrix of cofactors} = \begin{pmatrix} 122 & -13 \\ -13 & 51 \end{pmatrix}$$

The inverse of a matrix is obtained by dividing the matrix of cofactors for *E* by $|E|$, the determinant of *E*:

$$E^{-1} = \begin{pmatrix} \frac{122}{6053} & \frac{-13}{6053} \\ \frac{-13}{6053} & \frac{51}{6053} \end{pmatrix} = \begin{pmatrix} 0.0202 & -0.0021 \\ -0.0021 & 0.0084 \end{pmatrix}$$

Calculation of HE^{-1}:

$$\begin{aligned} HE^{-1} &= \begin{pmatrix} 10.47 & -7.53 \\ -7.53 & 19.47 \end{pmatrix} \begin{pmatrix} 0.0202 & -0.0021 \\ -0.0021 & 0.0084 \end{pmatrix} \\ &= \begin{pmatrix} [(10.47 \times 0.0202) + (-7.53 \times -0.0021)] & [(10.47 \times -0.0021) + (-7.53 \times 0.0084)] \\ [(-7.53 \times 0.0202) + (19.47 \times -0.0021)] & [(-7.53 \times -0.0021) + (19.47 \times 0.0084)] \end{pmatrix} \\ &= \begin{pmatrix} 0.2273 & -0.0852 \\ -0.1930 & 0.1794 \end{pmatrix} \end{aligned}$$

Calculation of eigenvalues

The eigenvalues or roots of any square matrix are the solutions to the determinantal equation $|A - \lambda I| = 0$, in which *A* is the square matrix in question and *I* is an identity matrix of the same size as *A*. The number of eigenvalues will equal the number of rows (or columns) of the square matrix. In this case the square matrix of interest is HE^{-1}:

$$\begin{aligned} \left|HE^{-1} - \lambda I\right| &= \left| \begin{pmatrix} 0.2273 & -0.0852 \\ -0.1930 & 0.1794 \end{pmatrix} - \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix} \right| \\ &= \left| \begin{pmatrix} (0.2273 - \lambda) & -0.0852 \\ -0.1930 & (0.1794 - \lambda) \end{pmatrix} \right| \\ &= [(0.2273 - \lambda)(0.1794 - \lambda) - (-0.1930 \times -0.0852)] \\ &= \lambda^2 - 0.2273\lambda - 0.1794\lambda + 0.0407 - 0.0164 \\ &= \lambda^2 - 0.4067\lambda + 0.0243 \end{aligned}$$

Therefore the equation $|HE^{-1} - \lambda I| = 0$ can be expressed as:

$$\lambda^2 - 0.4067\lambda + 0.0243 = 0$$

To solve the roots of any quadratic equation of the general form $a\lambda^2 + b\lambda + c = 0$ we can apply the following formula:

$$\lambda_i = \frac{-b \pm \sqrt{(b^2 - 4ac)}}{2a}$$

For the quadratic equation obtained, $a = 1$, $b = -0.4067$, $c = 0.0243$. If we replace these values into the formula for discovering roots, we get:

$$\begin{aligned}\lambda_i &= \frac{-b \pm \sqrt{(b^2 - 4ac)}}{2a} \\ &= \frac{0.4067 \pm \sqrt{[(-0.4067)^2 - 0.0972]}}{2} \\ &= \frac{0.4067 \pm 0.2612}{2} \\ &= \frac{0.6679}{2} \text{ or } \frac{0.1455}{2} \\ &= 0.334 \text{ or } 0.073\end{aligned}$$

Hence, the eigenvalues are 0.334 and 0.073.

Labcoat Leni's real research

A lot of hot air

Problem

Marzillier, S. L., & Davey, G. C. L. (2005). *Cognition and Emotion*, *19*, 729–750.

Have you ever wondered what researchers do in their spare time? Well, some of them spend it tracking down the sounds of people burping and farting! It has long been established that anxiety and disgust are linked. Anxious people are, typically, easily disgusted. Throughout this book I have talked about how you cannot infer causality from relationships between variables. This has been a bit of a conundrum for anxiety researchers: does anxiety cause feelings of digust or does a low threshold for being disgusted cause anxiety? Two colleagues of mine at Sussex addressed this in an unusual study in which they induced feelings of anxiety, feelings of disgust, or a neutral mood, and they looked at the effect that these induced moods had on feelings of anxiety, sadness, happiness, anger, disgust and contempt. To induce these moods, they used three different types of manipulation: vignettes (e.g. 'You're swimming in a dark lake and something brushes your leg' for anxiety, and 'You go into a public toilet and find it has not been flushed. The bowl of the toilet is full of diarrhoea' for disgust), music (e.g. some scary music for anxiety, and a tape of burps, farts and vomitting for disgust), videos (e.g. a clip from *Silence of the Lambs* for anxiety and a scene from *Pink Flamingos* in which Divine eats dog faeces for disgust) and memory (remembering events from the past that had made the person anxious, disgusted or neutral).

Different people underwent anxious, disgust and neutral mood inductions. Within these groups, the induction was done using either vignettes and music, videos, or memory recall and music for different people. The outcome variables were the change (from before to after the induction) in six moods: anxiety, sadness, happiness, anger, disgust and contempt.

The data are in the file **Marzillier and Davey (2005).dat**. Draw an error bar graph of the changes in moods in the different conditions, then conduct a 3 (Mood: anxiety, disgust, neutral) × 3 (Induction: vignettes + music, videos, memory recall + music) MANOVA on these data. Whatever you do, don't imagine what their fart tape sounded like while you do the analysis!

Solution

First of all make sure you have set your working directory to where the data file is located and then read in the data:

```
marazillierData<-read.delim("Marzillier & Davey (2005).dat", header = TRUE)
```

Next we want to make sure that the categorical variables **Induction** and **Mood** are set to be factors and that the levels of each factor are in the correct order:

```
marazillierData$Induction<-factor(marazillierData$Induction, levels = c("Vignettes +
 Music", "Videos", "Memory Recall + Music"))

marazillierData$Mood<-factor(marazillierData$Mood, levels = c("Anxious", "Disgust",
 "Neutral"))
```

To do the graph we have to convert the dataframe, which is currently in the wide format, to the long format. We can do this by using the *melt()* function:

```
moodMelt<-melt(marazillierData, id = c("Induction", "Mood"), measured =
 c("Anxiety.Change", "Sad.Change", "Happy.Change", "Angry.Change", "Disgust.Change",
 "Contempt.Change"))
```

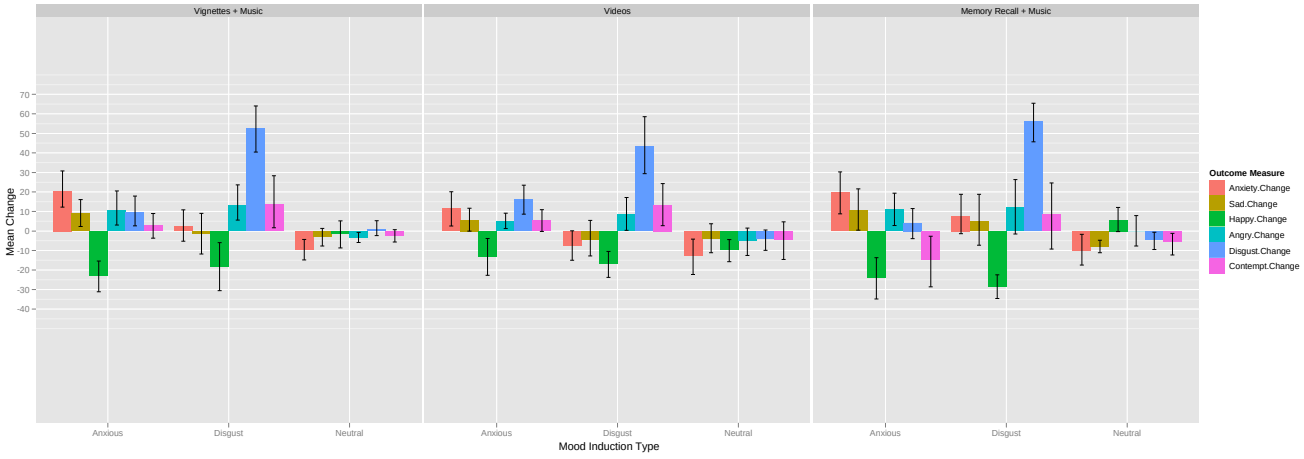
We can then specify names for the new variables by executing:

```
names(moodMelt)<-c("Induction", "Mood", "Outcome_Measure", "Frequency")
```

Now that we have the data in the long format we can plot the error bar graph by executing:

```
moodBar <- ggplot(moodMelt, aes(Mood, Frequency, fill = Outcome_Measure))
moodBar + stat_summary(fun.y = mean, geom = "bar", position = "dodge") +
 stat_summary(fun.data = mean_cl_boot, geom = "errorbar",
 position=position_dodge(width=0.90), width = 0.2) + labs(x = "Mood Induction Type", y
 = "Mean Change", fill = "Outcome Measure")+ facet_wrap(~Induction, ncol = 3) +
 scale_y_continuous(breaks = seq(-40, 70, by = 10))
```

The completed graph will look like that below. This shows that the neutral mood induction (regardless of the way in which it was induced) didn't really affect mood too much (the changes are all quite small). For the disgust mood induction, disgust always increased quite a lot regardless of how disgust was induced. Similarly, the anxiety induction raised anxiety (predominantly). Happiness decreased for both anxiety and disgust mood inductions.



Next we can set some contrasts. If we first look at the **Mood** variable, it makes sense for us to compare some form of mood induction (anxious and disgust) with a neutral mood induction, and also compare anxious with disgust. We can set these two contrasts by executing:

```
Mood_vs_Neut<-c(1, 1, -2)
anx_vs_disg <-c(1, -1, 0)
contrasts(marazillierData$Mood)<-cbind(Mood_vs_None, anx_vs_disg)
```

If we next look at the variable **Induction**, it makes some sense to compare the two methods of mood induction including music (vignettes and memory recall) with the video method, which does not include music. We could then compare the two methods that included music, vignettes and memory. We can set these contrasts by executing:

```
Music_vs_None<-c(1, -2, 1)
vig_vs_mem <-c(1, 0, -1)
contrasts(marazillierData$Induction)<-cbind(Music_vs_None, vig_vs_mem)
```

Main analysis

To do the main analysis, we specify a model in the function of the form *outcome ~predictor(s)* as we have done with most of the models in this book. However, if you remember from the chapter, because there are multiple outcomes in a MANOVA we have to first bind the variables together into a single entity using the *cbind()* function. In the current example we want to combine **Anxiety.Change**, **Sad.Change**, **Happy.Change**, **Angry.Change**, **Disgust.Change** and **Contempt.Change**, and we can create a single outcome object by executing:

```
outcome<-cbind(marazillierData$Anxiety.Change, marazillierData$Sad.Change,
 marazillierData$Happy.Change, marazillierData$Angry.Change,
 marazillierData$Disgust.Change, marazillierData$Contempt.Change)
```

This command creates an object called *outcome*, which contains the outcome variables of the *marazillierData* dataframe pasted together in columns. Therefore, for this example, we could estimate the model by executing:

```
marzillierModel<-manova(outcome ~ Induction*Mood, data = marazillierData)
```

To see the output of the model we use the *summary* command; by default, **R** produces Pillai's trace, which is a sensible choice:

```
summary(marzillierModel, intercept = TRUE)
```

```
                Df  Pillai   approx F num Df  den Df  Pr(>F)
(Intercept)      1 0.56756   36.311      6    166   <2e-16 ***
Induction        2 0.07760    1.123     12    334   0.3397
Mood             2 0.88092   21.910     12    334   <2e-16 ***
Induction:Mood   4 0.16615    1.221     24    676   0.2148
Residuals      171
```

The main multivariate statistics are shown above. A main effect of mood was found, $F(12, 334) = 21.91$, $p < .001$, showing that the changes for some mood inductions were bigger than for others overall (looking at the graph this finding probably reflects that the disgust mood induction had the greatest effect overall – mainly because it produced such huge changes in disgust). There was no significant main effect of the type of mood induction, $F(12, 334) = 1.12$, $p > .05$, showing that whether videos, memory, tapes, etc., were used did not affect the changes in mood. The type of mood × type of induction interaction was also non-significant, $F(24, 676) = 1.22$, $p > .05$, showing that the type of induction did not influence the main effect of mood. In other words, the fact that the disgust induction seemed to have the biggest effect on mood (overall) was not influenced by how disgust was induced.

If we want to follow up the analysis with univariate analysis of the individual outcome measures, then we can simply execute:

```
summary.aov(marzillierModel)
```

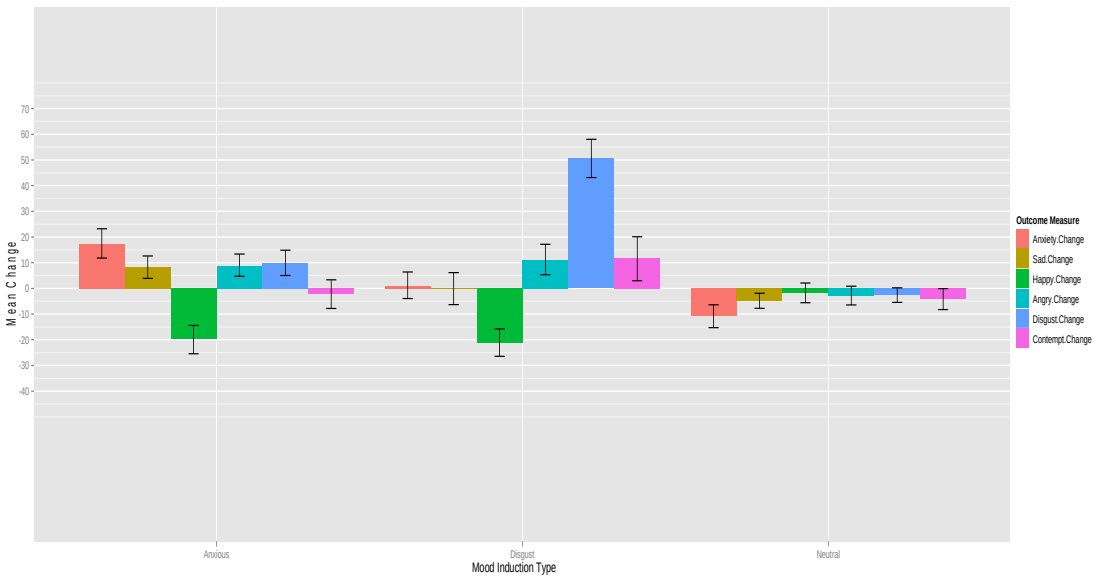
This produces the output below, which shows the ANOVA summary table for the dependent variables. The table labelled *Response 1* is for the **Anxiety.Change** variable, *Response 2* indicates the table for the **Sad.Change** variable, etc. The univariate effects for type of mood (which was the only significant multivariate effect) show that the effect of the type of mood induction was significant for all six moods (in other words, for all six moods there were significant differences across the anxiety, disgust and neutral conditions).

```
Response 1 :
                Df Sum Sq Mean Sq F value    Pr(>F)
Induction        2   2578  1289.1  3.1725   0.04438 *
Mood             2  23826 11913.0 29.3174 1.128e-11 ***
Induction:Mood   4    837   209.3  0.5150   0.72477
Residuals      171  69485   406.3
```

```
 Response 2 :
               Df Sum Sq Mean Sq F value   Pr(>F)
Induction       2    374  187.12  0.4819 0.618444
Mood            2   5501 2750.72  7.0842 0.001107 **
Induction:Mood  4   1048  261.91  0.6745 0.610509
Residuals     171  66397  388.29

 Response 3 :
               Df Sum Sq Mean Sq F value   Pr(>F)
Induction       2    189    94.5  0.2553  0.77495
Mood            2  14194  7096.8 19.1652 3.09e-08 ***
Induction:Mood  4   5077  1269.2  3.4276  0.01005 *
Residuals     171  63321   370.3

 Response 4 :
               Df Sum Sq Mean Sq F value    Pr(>F)
Induction       2    836   418.2   1.133 0.3244780
Mood            2   6677  3338.3   9.044 0.0001847 ***
Induction:Mood  4    131    32.9   0.089 0.9857879
Residuals     171  63120   369.1

 Response 5 :
               Df Sum Sq Mean Sq  F value Pr(>F)
Induction       2    217     108   0.2499 0.7792
Mood            2  92254   46127 106.2883 <2e-16 ***
Induction:Mood  4   3250     812   1.8721 0.1175
Residuals     171  74211     434

 Response 6 :
               Df Sum Sq Mean Sq  F value  Pr(>F)
Induction       2   3010  1505.2  2.4814 0.0866363 .
Mood            2   8882  4440.8  7.3210 0.0008899 ***
Induction:Mood  4   2093   523.2  0.8625 0.4877492
Residuals     171 103726   606.6
```

Below is a graph that collapses across the way that mood was induced (video, music, etc.) because this effect was not significant. We should do more tests, but just looking at the graph shows that changes in anxiety (red bars) are higher over the three mood conditions (they go up after the anxiety induction, stay positive for the disgust induction, and go down for the neutral induction). Similarly, for disgust, the change is biggest after the disgust induction, it increases a little after the anxiety induction and doesn't really change after the neutral. Finally, for happiness, this goes down after both anxiety and disgust inductions, but doesn't change for neutral.

```
moodBar <- ggplot(moodMelt, aes(Mood, Frequency, fill = Outcome_Measure))
moodBar + stat_summary(fun.y = mean, geom = "bar", position = "dodge") +
stat_summary(fun.data = mean_cl_boot, geom = "errorbar",
position=position_dodge(width=0.90), width = 0.2) + labs(x = "Mood Induction Type", y
= "Mean Change", fill = "Outcome Measure") + scale_y_continuous(breaks = seq(-40, 70,
by = 10))
```



Smart Alex's solutions

Task 1

- A clinical psychologist noticed that several of his manic psychotic patients did chicken impersonations in public. He wondered whether this behaviour could be used to diagnose this disorder and so decided to compare his patients against a normal sample. He observed 10 of his patients as they went through a normal day. He also needed to observe 10 of the most normal people he could find: naturally he chose to observe lecturers at the University of Sussex. He measured how many chicken impersonations they did in the streets of Brighton over the course of a day, and how good their impersonations were (as scored out of 10 by an independent farmyard noise expert). The data are in the file **chicken.dat**. Use MANOVA and DFA to find out whether these variables could be used to distinguish manic psychotic patients from those without the disorder.

First of all – yes, you've guessed it – we need to read in the data!

```
chickenData<-read.delim("chicken.dat", header = TRUE)
```

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

```
chickenData$group<-factor(chickenData$group, levels = c(1:2), labels = c("Manic
Psychosis", "Sussex Lecturers"))
```

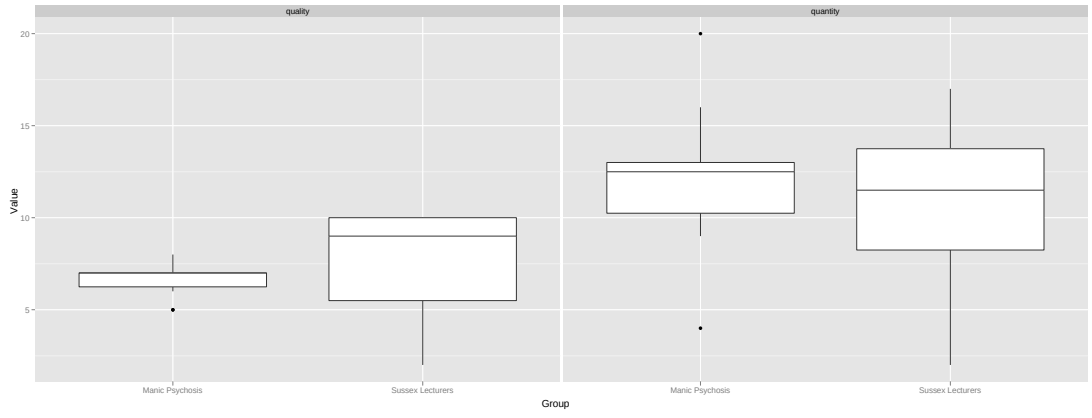
Let's explore the data with some graphs. To be able to plot a boxplot and an error bar graph, we need the data to be in a long format, however at the moment they are in the wide format. Not to worry, though, because as we have seen many times, we can convert the data into a long format using the *melt()* function:

```
chickenMelt<-melt(chickenData, id = c("group"), measured = c("quality", "quantity"))
names(chickenMelt)<-c("group","Outcome_Measure", "Value")
```

The above command produces a new dataframe called *chickenMelt*, which is *chickenData* converted into a long format (it is not a chicken and cheese panini, which being a vegetarian is a relief to me ☺).

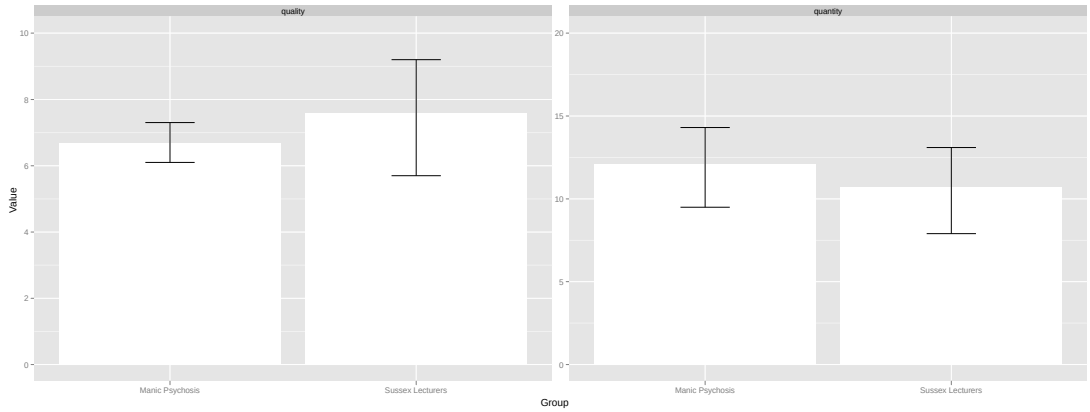
We can now use the *chickenMelt* dataframe to plot a boxplot:

```
Boxplot <- ggplot(chickenMelt, aes(group, Value))
Boxplot + geom_boxplot() + labs(x = "Group", y = "Value") + facet_wrap(~
 Outcome_Measure)
```



and an error bar graph:

```
Bar <- ggplot(chickenMelt, aes(group, Value))
Bar + stat_summary(fun.y = mean, geom = "bar", position = "dodge", fill = "white") +
 stat_summary(fun.data = mean_cl_boot, geom = "errorbar",
 position=position_dodge(width=0.90), width = 0.2) + labs(x = "Group", y = "Value") +
 facet_wrap(~ Outcome_Measure, ncol = 2, scales = "free_y")
```



Both of the resulting graphs above show that manic psychotics and Sussex lecturers do pretty similar numbers of chicken impersonations (lecturers do slightly fewer actually, but they are of a higher quality).

We can also have a look at some descriptive statistics for both outcome variables using the *by()* command:

```
by(chickenData$quality, chickenData$group, stat.desc, basic = FALSE)
```

```
chickenData$group: Manic Psychosis
      median         mean      SE.mean   CI.mean.0.95       var      std.dev     coef.var
       7.000        6.700        0.335        0.758        1.122        1.059        0.158
-------------------------------------------------------------------------------------------
chickenData$group: Sussex Lecturers
      median         mean      SE.mean   CI.mean.0.95       var      std.dev     coef.var
       9.000        7.600        0.945        2.138        8.933        2.989        0.393
```

```
by(chickenData$quantity, chickenData$group, stat.desc, basic = FALSE)
```

```
chickenData$group: Manic Psychosis
      median         mean      SE.mean   CI.mean.0.95       var      std.dev     coef.var
      12.500       12.100        1.337        3.025       17.878        4.228        0.349
-------------------------------------------------------------------------------------------
chickenData$group: Sussex Lecturers
      median         mean      SE.mean   CI.mean.0.95       var      std.dev     coef.var
      11.500       10.700        1.383        3.128       19.122        4.373        0.409
```

The tables above contain the group means, standard deviations, standard errors etc. for each dependent variable in turn. These again show that manic psychotics and Sussex lecturers do pretty similar numbers of chicken impersonations (lecturers do slightly fewer, but they are of a higher quality).

Having looked at the data in summary form, we can start to look at assumptions. To check the homogeneity of covariance matrices we can again use the *by()* function but in combination with the *cov()* function, which can be used to print the covariance matrix to the console:

```
by(chickenData[, 2:3], chickenData$group, cov)
```

The above command takes columns 2 and 3 of the *chickenData* dataframe, which means that we're selecting the columns that contain the variables **quality** and **quantity**. The command then applies the function *cov()* to these columns, but splits the output by the variable **group**.

The output below shows the variance–covariance matrices for each group. The diagonal elements represent the variances for each outcome measure and the off-diagonals are the covariances (i.e., the relationship between quality and quantity). The variances for quality are quite different across groups

(1.12 and 8.93), with the largest variance being nearly nine times bigger than the smallest. The variances for quantity are really quite similar (17.88 and 19.12), with a variance ratio of about 1.5, which is below the threshold of 2. Looking at the covariances, these are very different (4.03 and 12.42), reflecting the different relationships between quality and quantity across the two groups. On balance, there is evidence to suggest that the matrices are different across groups; however, given the group sizes are equal, we probably don't need to worry too much about these differences.

```
chickenData$group: Manic Psychosis
          quality  quantity
quality  1.122222  4.033333
quantity 4.033333 17.877778
------------------------------------
chickenData$group: Sussex Lecturers
          quality quantity
quality   8.933333 12.42222
quantity 12.422222 19.12222
```

The final assumption that we need to test is multivariate normality. We can do this using the *mshapiro.test()* function. We need to apply this test to the groups individually, so the first thing to do is to extract the data for each group and transpose the rows and columns using the transpose function *t()* so that the data are in the correct format for *mshapiro.test()*.

```
ManicPsychosis<-t(chickenData[1:10, 2:3])
SussexLecturers<-t(chickenData[11:20, 2:3])
```

To apply the test, we simply execute the function on each of the two variables that we have just created:

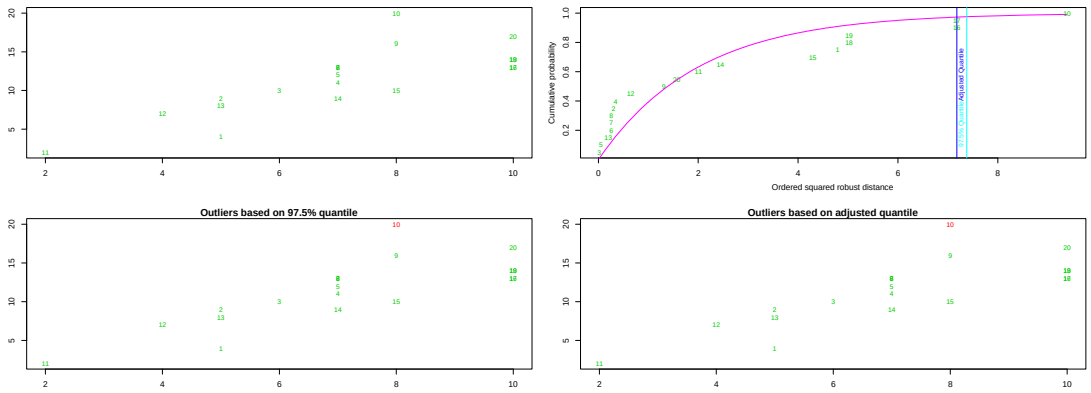
```
mshapiro.test(ManicPsychosis)
mshapiro.test(SussexLecturers)
```

The output below shows the results of the two tests. It is clear that for both the manic psychosis (p = .168), and Sussex lecturers (p = .147) there is no problem because both results are non-significant.

```
Shapiro-Wilk normality test

data:  Z
W = 0.8897, p-value = 0.1683


Shapiro-Wilk normality test

data:  Z
W = 0.8844, p-value = 0.1465
```

We can also look for multivariate outliers using the *aq.plot()* function:

```
aq.plot(chickenData[, 2:3])
```



The resulting plots above show the case numbers (i.e. the row number in the dataframe), and you need to be looking for values in red in all but the top right graph. You can see that row 10 might be an outlier. In the top right plot, you are looking for any cases that fall to the right of the vertical line.

Again, row 10 of the dataframe has been identified. These plots, therefore, suggest that row 10 might be an outlier. We could delete this case and see whether it makes any difference to the multivariate normality, although because we do not have a problem with multivariate normality in this example, we could also keep it in.

Before running the main analysis we need to set some contrasts … well, actually in this case we don't really because our categorical variable **group** has only two levels, but here is the code anyway:

```
contrasts(chickenData$group)<-c(-1, 1)
```

To run the main analysis, first we need to combine our two outcome variables, **quality** and **quantity** into a single outcome object. We can do this by executing:

```
outcome<-cbind(chickenData$quality, chickenData$quantity)
```

This command creates an object called *outcome*, which contains the outcome variables of the *chickenData* dataframe pasted together in columns. Therefore, for this example, we could estimate the model by executing:

```
chickenModel<-manova(outcome ~ group, data = chickenData)
```

To see the output of the model we use the summary command; by default, **R** produces Pillai's trace, which is a sensible choice:

```
summary(chickenModel, intercept = TRUE)
```

```
            Df  Pillai approx F num Df den Df    Pr(>F)
(Intercept)  1 0.91882   96.201      2     17 5.376e-10 ***
group        1 0.33334    4.250      2     17   0.03185 *
 Residuals 18
```

The main multivariate statistics are shown above. The column of real interest is the one containing the significance values of the *F*-ratio for **group**. For these data, the test statistics for **group** is significant, $p = .032$ (which is less than .05). From this result we should probably conclude that the groups do indeed differ in terms of the quality and quantity of their chicken impersonations; however, this effect needs to be broken down to find out exactly what's going on.

To follow up the analysis with univariate analysis of the individual outcome measures, we can simply execute:

```
summary.aov(chickenModel)
```

This produces the output below, which shows the ANOVA summary table for the dependent variables. The table labeled *Response 1* is for the **quality** variable and *Response 2* indicates the table for the **quantity** variable.

The row labelled *group* contains an ANOVA summary table for quality and quantity of chicken impersonations, respectively. The values of *p* indicate that there was a non-significant difference between groups in terms of both (both *p*s are greater than .05). The multivariate test statistics led us to conclude that the groups *did* differ significantly in the quality and quantity of their chicken impressions, yet the univariate results contradict this!

```
Response 1 :
            Df Sum Sq Mean Sq F value Pr(>F)
group        1   4.05  4.0500  0.8055 0.3813
Residuals   18  90.50  5.0278

 Response 2 :
            Df Sum Sq Mean Sq F value Pr(>F)
group        1    9.8     9.8  0.5297 0.4761
Residuals   18  333.0    18.5
```

We don't need to look at the contrasts because the univariate tests were non-significant (and in any case there were only two groups and so no further comparisons would be necessary), and instead, to

see how the dependent variables interact, we need to carry out a discriminant function analysis. To carry out discriminant function analysis for the current data, we would execute:

```
chickenDFA<-lda(group ~ quality + quantity, data = chickenData)
```

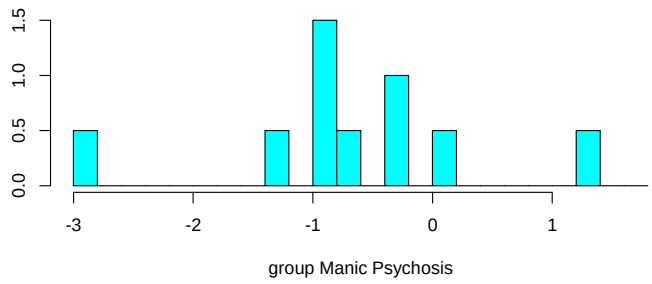
This creates a model called *chickenDFA*. To see this model, execute the name of the model:

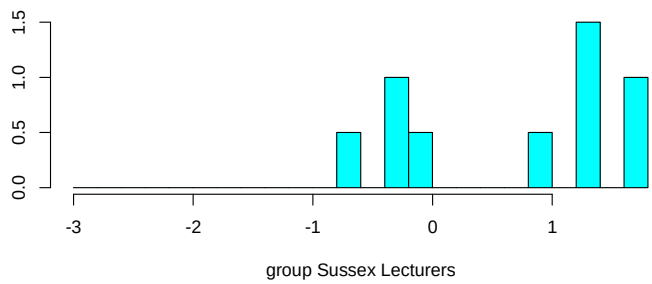
```
 chickenDFA

Call:
lda(group ~ quality + quantity, data = chickenData)

Prior probabilities of groups:
 Manic Psychosis Sussex Lecturers
             0.5               0.5

Group means:
                  quality quantity
Manic Psychosis       6.7     12.1
Sussex Lecturers      7.6     10.7

Coefficients of linear discriminants:
                 LD1
quality    0.8292832
quantity -0.4252235
```

The main part of the output above tells us the coefficients of the linear discriminants. We can see that there was only one variate (because there are only two groups), therefore, the group differences shown by the MANOVA can be explained in terms of *one* underlying dimension. The coefficients of the linear discriminants also tell us the relative contribution of each variable to the variates. Both quality and quantity of impersonations have quite large coefficients, indicating that they both have a strong influence in discriminating the groups. However, they have the opposite sign, which suggests that that group differences are explained by the difference between the quality and quantity of impersonations.

We can have a look at a plot of the scores broken down by group membership by executing:

```
plot(chickenDFA)
```





The above graphs for each group confirm that variate 1 discriminates the two groups because the manic psychotics have a negative coefficient and the Sussex lecturers have a positive one. There won't be a combined-groups plot because there is only one variate.

Overall we could conclude that manic psychotics are distinguished from Sussex lecturers in terms of the difference between the pattern of results for quantity of impersonations compared to quality of them. If we look at the means we can see that manic psychotics produce slightly more impersonations than Sussex lecturers (but remember from the non-significant univariate tests that this isn't sufficient, alone, to differentiate the groups), but the lecturers produce impersonations of a higher quality (but again remember that quality alone is not enough to differentiate the groups). Therefore, although the manic psychotics and Sussex lecturers produce similar numbers of impersonations of similar quality (see univariate tests) if we combine the quality and quantity we can differentiate the groups.

Task 2

I was intrigued by a news story claiming that children who lie would become successful citizens (http://bit.ly/ammQNT). I was particularly intrigued because although the article cited a lot of well-conducted work by Dr Khang Lee that shows that children lie, I couldn't find anything at all that made the rather fabulous jump of logic to these children becoming successful citizens. If we wanted to test this hypothesis, we could imagine a Huxleyesque parallel universe in which the government is stupid enough to believe this newspaper story and decides to implement a systematic programme of infant conditioning. Some infants were trained not to lie, others were brought up as normal, and a final group were trained in the art of lying. Thirty years later, they collected data on how successful these children were as adults. They measured their **salary**, and two indices of how successful they were in their **family** and **work** life, on a 0–10 scale (10 = very successful, 0 = very unsuccessful). The data are in **lying.dat**. Use MANOVA and DFA to find out whether lying really does make you a better citizen.

Let's begin by reading in the data:

```
lyingData<-read.delim("lying.dat", header = TRUE)
```

Next, we need to set **lying** to be a factor with the levels labelled in the correct order:

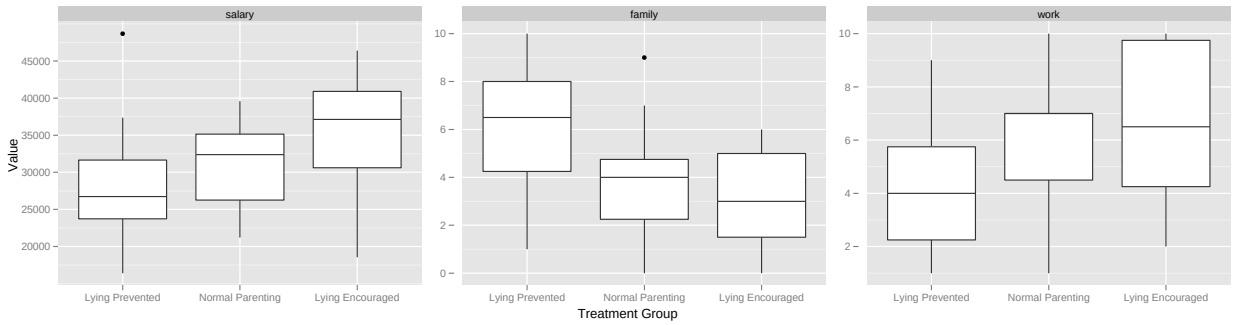
```
lyingData$lying<-factor(lyingData$lying, levels = c("Lying Prevented", "Normal
 Parenting", "Lying Encouraged"))
```

We then want to explore the data with some graphs. You probably know by now that to do this, we need the data to be in the long format. The data have been entered in the wide format and so we need to convert the data to the long format using the *melt()* function:

```
lyingMelt<-melt(lyingData, id = c("lying", "row"), measured = c("salary", "family",
 "work"))
names(lyingMelt)<-c("lying", "row", "Outcome_Measure", "Value")
```

To plot a boxplot we could execute:

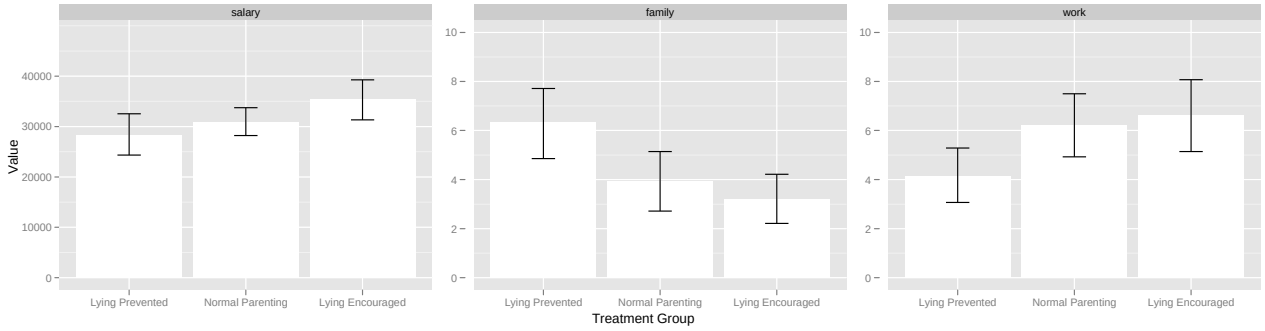
```
Boxplot <- ggplot(lyingMelt, aes(lying, Value))
Boxplot + geom_boxplot() + labs(x = "Treatment Group", y = "Value") + facet_wrap(~
 Outcome_Measure, ncol = 3, scales = "free_y")
```



The resulting boxplot above suggests that children who were encouraged to lie had the highest salaries and were the most successful at work; however, they were the least successful in their family lives. On the other hand, children who were trained not to lie had the lowest salaries and the least success in their work; however, they had the most success in their family life. The final group of children who received normal parenting was OK in all areas, although their family success seems quite low compared to children who were trained not to lie. Therefore, it seems that to be successful at work and to earn a lot of money you need to be a good liar but to have a successful family life you need to be honest and not lie.

To plot an error bar graph we could execute:

```
Bar <- ggplot(lyingMelt, aes(lying, Value))
Bar + stat_summary(fun.y = mean, geom = "bar", position = "dodge", fill = "white") +
 stat_summary(fun.data = mean_cl_boot, geom = "errorbar",
 position=position_dodge(width=0.90), width = 0.2) + labs(x = "Treatment Group", y =
 "Value") + facet_wrap(~ Outcome_Measure, ncol = 3, scales = "free_y")
```



The resulting graph above displays much the same information as the boxplot, just in a different way.

Next it would be a good idea to have a look at the descriptive statistics for each outcome variable by executing:

```
by(lyingData$salary, lyingData$lying, stat.desc, basic = FALSE)
```

```
lyingData$lying: Lying Prevented
      median          mean       SE.mean  CI.mean.0.95           var       std.dev      coef.var
    2.67e+04      2.83e+04      2.23e+03      4.82e+03      6.97e+07      8.35e+03      2.95e-01
-------------------------------------------------------------------------------------------------
lyingData$lying: Normal Parenting
      median          mean       SE.mean  CI.mean.0.95           var       std.dev      coef.var
    3.24e+04      3.09e+04      1.53e+03      3.30e+03      3.27e+07      5.72e+03      1.85e-01
-------------------------------------------------------------------------------------------------
lyingData$lying: Lying Encouraged
      median          mean       SE.mean  CI.mean.0.95           var       std.dev      coef.var
    3.71e+04      3.54e+04      2.17e+03      4.68e+03      6.58e+07      8.11e+03      2.29e-01
```

```
by(lyingData$family, lyingData$lying, stat.desc, basic = FALSE)
```

```
lyingData$lying: Lying Prevented
      median          mean       SE.mean  CI.mean.0.95           var       std.dev      coef.var
       6.500         6.357         0.775         1.674         8.401         2.898         0.456
-------------------------------------------------------------------------------------------------
lyingData$lying: Normal Parenting
      median          mean       SE.mean  CI.mean.0.95           var       std.dev      coef.var
       4.000         3.929         0.633         1.368         5.610         2.369         0.603
-------------------------------------------------------------------------------------------------
lyingData$lying: Lying Encouraged
```

```
      median         mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
       3.000        3.214        0.526        1.136        3.874        1.968        0.612
```

```
by(lyingData$work, lyingData$lying, stat.desc, basic = FALSE)
```

```
lyingData$lying: Lying Prevented
      median         mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
       4.000        4.143        0.582        1.258        4.747        2.179        0.526
------------------------------------------------------------------------------------------------------------
lyingData$lying: Normal Parenting
      median         mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
       7.000        6.214        0.720        1.556        7.258        2.694        0.434
------------------------------------------------------------------------------------------------------------
lyingData$lying: Lying Encouraged
      median         mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
       6.500        6.643        0.789        1.704        8.709        2.951        0.444
```

The tables above contain the group means, standard deviations, standard errors etc. for each dependent variable in turn. These again show that children encouraged to lie won the best and highest-paid jobs, but had the worst family success compared to the other two groups. Children who were trained not to lie had great family lives but not so great jobs compared to children who were brought up to lie and children who experienced normal parenting. Finally, children who were in the normal parenting group (if that exists!) were pretty middle-of-the-road compared to the other two groups.

Having looked at the data in summary form, we can start to look at assumptions. To check the homogeneity of covariance matrices we can again use the *by()* function in combination with the *cov()* function, which can be used to print the covariance matrix to the console:

```
by(lyingData[, 2:4], lyingData$lying, cov)
```

The above command takes columns 2 to 4 of the *lyingData* dataframe, which means that we're selecting the columns that contain the variables **salary**, **family** and **work**. The command then applies the function *cov()* to these columns, but splits the output by the variable **lying**.

The output below shows the variance–covariance matrices for each group. The diagonal elements represent the variances for each outcome measure and the off-diagonals are the covariances (i.e., the relationship between salary, family and work). The variances for salary are quite similar across groups (69717086.84, 32709144.90 and 65788080.73), with a variance ratio of about 1.8, which is just below the threshold of 2. However, the variances for family are not that similar (8.40, 5.61 and 3.87), nor are the variances for work (4.75, 7.26 and 8.71), although variance ratios are close to the threshold of 2. Looking at the covariances, these are fairly different in most cases, reflecting the different relationships between salary, work and family across the three groups. On balance, there is evidence to suggest that the matrices are different across groups; however, given the group sizes are equal, we probably don't need to worry too much about these differences.

```
lyingData$lying: Lying Prevented
             salary      family        work
salary 69717086.835 2268.659341 2263.956044
family     2268.659    8.401099    3.714286
work       2263.956    3.714286    4.747253
------------------------------------------------
lyingData$lying: Normal Parenting
             salary      family        work
salary 32709144.901 -2999.24176 1597.109890
family    -2999.242     5.60989    2.170330
work       1597.110     2.17033    7.258242
------------------------------------------------
lyingData$lying: Lying Encouraged
             salary       family         work
salary 65788080.725 7075.0219780 7575.0659341
family     7075.022    3.8736264   -0.8406593
work       7575.066   -0.8406593    8.7087912
```

The final assumption that we need to test is multivariate normality. We can do this using the *mshapiro.test()* function. We need to apply this test to the groups individually, so the first thing to do is to extract the data for each group and transpose the rows and columns using the transpose function *t()* so that the data are in the correct format for *mshapiro.test()*.

```
lp<-t(lyingData[1:14, 2:4])
np<-t(lyingData[15:28, 2:4])
le<-t(lyingData[29:42, 2:4])
```

To apply the test, we simply execute the function on each of the two variables that we have just created:

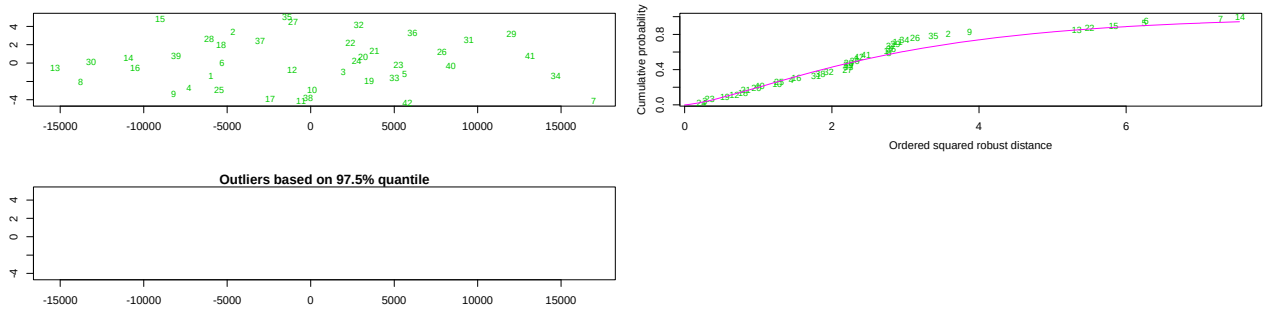
```
mshapiro.test(lp)
mshapiro.test(np)
mshapiro.test(le)
```

The output below shows the results of the two tests. It is clear that, for all three groups, lying prevented (p = .114), normal parenting (p = .605) and lying encouraged (p = .161), there is no problem because all results are non-significant.

```
> mshapiro.test(lp)

        Shapiro-Wilk normality test

data:  Z
W = 0.9003, p-value = 0.114

> mshapiro.test(np)

        Shapiro-Wilk normality test

data:  Z
W = 0.9528, p-value = 0.605

> mshapiro.test(le)

        Shapiro-Wilk normality test

data:  Z
W = 0.9106, p-value = 0.1608
```

We can also look for multivariate outliers using the *aq.plot()* function:

```
aq.plot(lyingData[, 2:4])
```



The resulting plots above show the case numbers (i.e. the row number in the dataframe), and you need to be looking for values in red in all but the top right graph. In this case it seems that there are no outliers – hooray!

Main analysis

Before running the main analysis, we need to set some contrasts for the **lying** variable. When thinking about the best contrasts to set, it makes sense for the first contrast to compare children who were encouraged to lie (lying encouraged) with those who were not encouraged to tell lies (normal parenting and lying prevented combined). It then makes sense for the second contrast to compare children who had normal parenting with those who were trained not to lie. We can set these contrasts by executing:

```
Encouraged_vs_Not<-c(1, 1, -2)
normal_vs_prevented <-c(1, -1, 0)
 contrasts(lyingData$lying)<-cbind(Encouraged_vs_Not, normal_vs_prevented)
```

To run the main analysis, first we need to combine our three outcome variables, **salary**, **family** and **work** into a single outcome object. We can do this by executing:

```
outcome<-cbind(lyingData$salary, lyingData$family, lyingData$work)
```

This command creates an object called *outcome*, which contains the outcome variables of the *lyingData* dataframe pasted together in columns. Therefore, for this example, we could estimate the model by executing:

```
lyingModel<-manova(outcome ~ lying, data = lyingData)
```

To see the output of the model we use the *summary* command; by default, **R** produces Pillai's trace, which is a sensible choice:

```
summary(lyingModel, intercept = TRUE)
```

```
            Df  Pillai approx F  num Df  den Df    Pr(>F)
(Intercept)  1 0.95711  275.252       3      37 < 2.2e-16 ***
lying        2 0.47811    3.979       6      76  0.001626 **
Residuals   39
```

The main multivariate statistics are shown above. The column of real interest is the one containing the significance values of the *F*-ratio for **lying**. For these data, the test statistic for **lying** is significant, $p < .01$ (which is less than .05). From this result we should probably conclude that the groups do indeed differ in terms of salary, family and work as a result of their lying ability; however, this effect needs to be broken down to find out exactly what's going on.

To follow up the analysis with univariate analysis of the individual outcome measures, we can simply execute:

```
summary.aov(lyingModel)
```

This produces the output below, which shows the ANOVA summary table for the dependent variables. The table labelled *Response 1* is for the **salary** variable, *Response 2* indicates the table for the **family** variable and *Response 3* is for the **work** variable.

```
Response 1 :
            Df     Sum Sq   Mean Sq F value  Pr(>F)
lying        2  366202116 183101058  3.2655 0.04884 *
Residuals   39 2186786062  56071437
---

 Response 2 :
            Df Sum Sq Mean Sq F value   Pr(>F)
lying        2   76.0  38.000  6.3742 0.004025 **
Residuals   39  232.5   5.962
---

 Response 3 :
            Df  Sum Sq Mean Sq F value  Pr(>F)
lying        2  50.048 25.0238  3.6241 0.03601 *
Residuals   39 269.286  6.9048
```

The row labelled *lying* contains an ANOVA summary table for success in the three areas: salary, family and work, respectively. The values of *p* indicate that there was a significant difference between groups in terms of all three dependent variables (all *p*s are less than .05).

We now need to look at the contrasts to see where these differences occur. Now, if you remember from the chapter, the contrasts are not part of the main MANOVA model and so to generate the output for them you have to create separate linear models for each outcome measure. This is basically the same as doing a one-way ANOVA on each outcome measure. So, for **salary**, **family** and **work** we would create the following models using the *lm()* function:

```
salaryModel<-lm(salary ~ lying, data = lyingData)
familyModel<-lm(family ~ lying, data = lyingData)
workModel<-lm(work ~ lying, data = lyingData)
```

The first command creates a model, *salaryModel*, based on predicting the variable **salary** from lying and the second and third commands do pretty much the same but predicting **family** and **work**, respectively. We can get the contrast parameters by using *summary.lm()*:

```
summary.lm(salaryModel)
```

```
Coefficients:
```

```
                          Estimate Std. Error t value  Pr(>|t|)
(Intercept)                  31529       1155  27.287   <2e-16 ***
lyingEncouraged_vs_Not       -1941        817  -2.376   0.0225 *
lyingnormal_vs_prevented     -1331       1415  -0.941   0.3526
```

```
summary.lm(familyModel)
```

```
Coefficients:
                          Estimate Std. Error t value  Pr(>|t|)
(Intercept)                 4.5000     0.3768  11.944   1.33e-14 ***
lyingEncouraged_vs_Not      0.6429     0.2664   2.413   0.0206 *
lyingnormal_vs_prevented    1.2143     0.4614   2.632   0.0121 *
```

```
summary.lm(workModel)
```

```
                          Estimate Std. Error t value   Pr(>|t|)
(Intercept)                 5.6667     0.4055  13.976   <2e-16 ***
lyingEncouraged_vs_Not     -0.4881     0.2867  -1.702   0.0966 .
lyingnormal_vs_prevented   -1.0357     0.4966  -2.086   0.0436 *
```

Looking at the *Pr(>|t|))* columns in the output tables above, we can see that when we compare people who were encouraged to lie as children with those who were not, there are significant differences in salary (p = .023) and family success (p = .021), but not in work success (p = .097). Looking back at the graphs and means, we can see that people who are trained to lie as children earn significantly higher salaries but have significantly less successful family lives when compared to those not trained to lie.

When we compare people who experienced normal parenting and those who were prevented from lying as children, there were significant differences in both family (p = .012) and work success (p = .044) but not in salary (p = .353). Again, if we look back at the boxplot and error bar graph, we can see that people prevented from lying as children have significantly more successful family lives but are significantly less successful at work when compared with those who experienced normal parenting.

For these data, we wouldn't normally run a robust analysis because there were no problems with normality. However, for the purposes of giving you an example of how you would run a robust analysis, I am going to run one anyway.

Robust analysis

The robust functions need the data to be in wide format rather than long. Essentially we want levels of the independent variable (**lying**) and outcome measures (**salary**, **family** and **work**) to be represented in different columns. The outcome measures are already spread across different columns (**salary**, **family** and **work**), but the lying group is differentiated by different rows of data. Therefore, we need to take the rows representing people who were in the lying prevented, normal parenting and lying encouraged groups and shift them into columns alongside the columns currently labelled **salary**, **family** and **work**.

We can do this restructuring using the *melt()* and *cast()* functions. To get the restructuring to work, we need to add a variable to our dataframe that identifies the rows in the wide format. We can add this variable to the dataframe by executing:

```
lyingData$row<-rep(1:14, 3)
```

Executing this command creates a variable **row** in the dataframe *lyingData*, that is, the numbers 1 to 14 repeated three times. The structure of the data will be the same as before – it's just that we have a new variable called **row** that identifies the scores within each lying group.

Next we need to make the data molten so that we can cast the data into the wide format:

```
lyingMelt<-melt(lyingData, id = c("lying", "row"), measured = c("salary", "family",
 "work"))
```

We can then take the *lyingMelt* dataframe and assign informative names to each column:

```
names(lyingMelt)<-c("lying", "row", "Outcome_Measure", "value")
```

Finally, we want to cast our data into the wide format using *cast()* and then remove the variable **row** from the dataframe as we do not want it for the robust analysis:

```
lyingRobust<-cast(lyingMelt, row ~ lying + Outcome_Measure, value = "value")

lyingRobust$row<-NULL
```

You can view the dataframe by executing its name:

```
 lyingRobust
```

To save space, I'm not going to paste the data here, but you should have a look at it because it is important to note the order of the columns: the hierarchy of the independent variables is **lying** followed by **Outcome_Measure**. It is important that you specify the variables in this order because this arranges the data correctly for Wilcox's functions.

We need only specify the dataframe (*lyingRobust*) and then the number of groups (three in this case) and the number of outcome measures (again there are three in this case). Therefore, we can do a robust MANOVA based on ranks by executing:

```
mulrank(3, 3, lyingRobust)

cmanova(3, 3, lyingRobust)
```

Mulrank()

```
$test.stat
[1] 4.359817


$nu1
[1] 5.248292


$p.value
             [,1]
[1,] 0.000444425


$N
[1] 42


$q.hat
          [,1]      [,2]      [,3]
[1,] 0.3707483 0.6862245 0.3350340
[2,] 0.4795918 0.4447279 0.5688776
[3,] 0.6496599 0.3690476 0.5960884
```

cmanova()

```
$test.stat
[1] 28.94646

$df
[1] 6

$p.value
                  [,1]
[1,] 6.227267e-05
```

The output for both these commands is shown above. For *Mulrank()* we are given a test statistic for the type of lying group (*$test.stat*) as well as the corresponding *p*-value (*$p.value*). We could conclude that there was a significant main effect of the type of lying condition on outcome measures of success, $F = 4.36$, $p < .001$. The numbers under *$q.hat* tell us the relative effects (i.e., the typical ranks across the combinations of groups in the rows and outcome measures in the columns).

We could relabel this grid as:

```
     [salary]   [family]   [work]
[lp] 0.3707483  0.6862245  0.3350340
[np] 0.4795918  0.4447279  0.5688776
[le] 0.6496599  0.3690476  0.5960884
```

This shows that for the lying prevented group (*lp*), the ranks were fairly similar for salary and work success (0.37 and 0.34) but were higher for family success (0.69). This indicates that lying prevention

affects family success more than salary and work success. For the normal parenting (*np*) group, all the ranks were fairly similar (0.48, 0.44 and 0.57), although they were slightly higher for work (0.57), suggesting that normal parenting affects work more than the other two outcomes. Finally, for the lying encouraged group (*le*), the ranks were highest for salary and work (0.65 and 0.60), suggesting that lying encouragement affected salary and work success more than family success.

The output of *cmanova()* tells us much the same things: we get a test statistic (*$test.stat*), the degrees of freedom (*$df*) and an associated *p*-value (*$p.value*). We could conclude that there was a significant main effect of the type of lying condition on outcome measures of success, $H(6) = 28.95$, $p < .001$.

Reporting Results

- ✓ Using Pillai's trace, there was a significant effect of lying on future success, $V = 0.48$, $F(6, 76) = 3.98$, $p < .01$. Separate univariate ANOVAs on the outcome variables revealed significant lying effects on salary $F(2, 39) = 3.27$, $p < .05$, family, $F(2, 39) = 6.37$, $p < .01$ and work, $F(2, 39) = 3.62$, $p < .05$. Contrasts revealed that: (1) people who were encouraged to lie as children and those who were not did not differ significantly with regard to work success, $b = –0.49$, $t(0.29) = –1.70$, $p > .05$. However, people who were encouraged to lie as children grew up to have significantly larger salaries, $b = –1941.0$, $t(817) = –2.38$, $p < .05$, and significantly less successful family lives, $b = 0.64$, $t(0.27) = 2.41$, $p < .05$, than those who were not encouraged to lie; (2) people who were prevented from lying as children and those who experienced normal parenting did not differ significantly in terms of salary, $b = –1331.0$, $t(1415) = –0.94$, $p > .05$; however, they were significantly more successful in their family lives, $b = 1.21$, $t(0.46) = 2.63$, $p < .05$, but significantly less successful in their work lives, $b = –1.03$, $t(0.50) = –2.09$, $p < .05$.

If you have used a robust MANOVA then you might report this as:

- ✓ A MANOVA was conducted on the ranked data using Munzel and Brunner's (2000) method implemented in **R** using the *mulrank()* function (Wilcox, 2005). There was a significant main effect of the type of treatment on outcomes of success, $F = 4.36$, $p < .001$.
- ✓ A MANOVA was conducted on the ranked data using Choi and Marden's (1997) method, implemented in **R** using the *cmanova()* function (Wilcox, 2005). There was a significant main effect of lying on outcomes of success, $H(6) = 28.95$, $p < .001$.

Task 3

- I was interested in whether students' knowledge of different aspects of psychology improved throughout their degree. I took a sample of first years, second years and third years and gave them five tests (scored out of 15) representing different aspects of psychology: **exper** (experimental psychology such as cognitive and neuropsychology); **stats** (statistics); **social** (social psychology); **develop** (developmental psychology); **person** (personality). Your task is to: (1) carry out an appropriate general analysis to determine whether there are overall group differences along these five measures; (2) look at the scale-by-scale analyses of group differences produced in the output and interpret the results accordingly; (3) select contrasts that test the hypothesis that second and third years will score higher than first years on all scales; (4) select tests that compare all groups to each other and briefly compare these results with the contrasts; and (5) carry out a separate analysis in which you test whether a combination of the measures can successfully discriminate the groups (comment only briefly on this analysis). Include only those scales that revealed group differences for the contrasts. How do the results help you to explain the findings of your initial analysis? The data are in the file **psychology.dat**.

Let's begin by reading in the data:

```
psychologyData<-read.delim("psychology.dat", header = TRUE)
```

Next, we need to set **group** to be a factor with the levels labeled in the correct order:

```
psychologyData$group<-factor(psychologyData$group, levels = c(0:2), labels = c("Yr_1", "Yr_2", "Yr_3"))
```

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

```
by(psychologyData$exper, list(psychologyData$group), stat.desc, basic = FALSE)
by(psychologyData$stats, list(psychologyData$group), stat.desc, basic = FALSE)
by(psychologyData$social, list(psychologyData$group), stat.desc, basic = FALSE)
by(psychologyData$develop, list(psychologyData$group), stat.desc, basic = FALSE)
by(psychologyData$person, list(psychologyData$group), stat.desc, basic = FALSE)
```

```
exper
: Yr_1
      median         mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
   6.0000000    5.6363636    0.6504925    1.4493876    4.6545455    2.1574396    0.3827715
-------------------------------------------------------------------------------------------
: Yr_2
      median         mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
   5.5000000    5.5000000    0.3979112    0.8481277    2.5333333    1.5916449    0.2893900
-------------------------------------------------------------------------------------------
: Yr_3
      median         mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
   6.0000000    7.0000000    0.5883484    1.2819011    4.5000000    2.1213203    0.3030458


stats
: Yr_1
      median         mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
    7.000000     7.545455     1.073343     2.391558    12.672727     3.559877     0.471791
-------------------------------------------------------------------------------------------
: Yr_2
      median         mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
   9.0000000    8.6875000    0.5966486    1.2717264    5.6958333    2.3865945    0.2747159
-------------------------------------------------------------------------------------------
: Yr_3
      median         mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
  11.0000000   10.4615385    0.8594526    1.8725864    9.6025641    3.0988004    0.2962089


social
: Yr_1
      median         mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
  10.0000000   10.3636364    0.8232168    1.8342414    7.4545455    2.7303013    0.2634501
-------------------------------------------------------------------------------------------
: Yr_2
      median         mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
   8.0000000    8.5625000    0.7010037    1.4941541    7.8625000    2.8040150    0.3274762
-------------------------------------------------------------------------------------------
: Yr_3
      median         mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
   8.0000000    8.7692308    0.4550831    0.9915408    2.6923077    1.6408253    0.1871117


develop
: Yr_1
      median         mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
  10.0000000   11.0000000    0.7977240    1.7774399    7.0000000    2.6457513    0.2405228
-------------------------------------------------------------------------------------------
: Yr_2
      median         mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
   8.5000000    8.8750000    0.4269563    0.9100358    2.9166667    1.7078251    0.1924310
-------------------------------------------------------------------------------------------
: Yr_3
      median         mean      SE.mean  CI.mean.0.95          var      std.dev     coef.var
   9.0000000    8.7692308    0.8408927    1.8321479    9.1923077    3.0318819    0.3457409


person
: Yr_1
      median         mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
  11.0000000   10.6363636    1.0024763    2.2336563   11.0545455    3.3248377    0.3125916
-------------------------------------------------------------------------------------------
: Yr_2
      median         mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
   8.0000000    8.4375000    0.4997395    1.0651696    3.9958333    1.9989581    0.2369135
-------------------------------------------------------------------------------------------
: Yr_3
      median         mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
   9.0000000    8.3846154    0.6654328    1.4498535    5.7564103    2.3992520    0.2861493
```

The output above shows the tables of descriptive statistics of the group means and standard deviations etc. of each year for each dependent variable in turn. I have edited the tables a little to make them clearer.

Next we need to check the assumptions. To check the homogeneity of covariance matrices we can again use the *by()* function in combination with the *cov()* function, which can be used to print the covariance matrix to the console:

```
by(psychologyData[, 2:6], psychologyData$group, cov)
```

The above command takes columns 2 to 6 of the *psychologyData* dataframe, which means that we're selecting the columns that contain the variables **exper**, **stats**, **social**, **develop** and **person**. The command then applies the function *cov()* to these columns, but splits the output by the variable **group**.

The output below shows the variance–covariance matrices for each group. The diagonal elements represent the variances for each outcome measure and the off-diagonals are the covariances (i.e., the relationship between each outcome measure). The variances for all of the outcome variables are quite different and, looking at the covariances, these are fairly different in most cases too, reflecting the different relationships between the different outcome variables across the three groups. On balance, there is evidence to suggest that the matrices are different across groups; additionally, given the group sizes are unequal, we probably should carry out a robust MANOVA on these data.

```
psychologyData$group: Yr_1
          exper     stats   social develop    person
exper   4.654545  5.018182 2.245455     1.1  1.654545
stats   5.018182 12.672727 6.581818     4.5  4.118182
social  2.245455  6.581818 7.454545     6.2  6.445455
develop 1.100000  4.500000 6.200000     7.0  7.200000
person  1.654545  4.118182 6.445455     7.2 11.054545
------------------------------------------------------------
psychologyData$group: Yr_2
           exper    stats    social  develop   person
exper   2.5333333 1.500000 0.1666667 1.200000 1.033333
stats   1.5000000 5.695833 4.6541667 2.891667 3.412500
social  0.1666667 4.654167 7.8625000 3.675000 3.070833
develop 1.2000000 2.891667 3.6750000 2.916667 2.058333
person  1.0333333 3.412500 3.0708333 2.058333 3.995833
------------------------------------------------------------
psychologyData$group: Yr_3
           exper       stats    social  develop     person
exper    4.500000 -1.00000000 1.5000000 1.916667 1.16666667
stats   -1.000000  9.60256410 0.8653846 4.032051 0.05769231
social   1.500000  0.86538462 2.6923077 2.525641 1.26282051
develop  1.916667  4.03205128 2.5256410 9.192308 3.01282051
person   1.166667  0.05769231 1.2628205 3.012821 5.75641026
```

The final assumption that we need to test is multivariate normality. We can do this using the *mshapiro.test()* function. We need to apply this test to the groups individually, so the first thing to do is to extract the data for each group and transpose the rows and columns using the transpose function *t()* so that the data are in the correct format for *mshapiro.test()*.

```
Yr_1<-t(psychologyData[1:11, 2:6])
Yr_2<-t(psychologyData[12:27, 2:6])
 Yr_3<-t(psychologyData[28:40, 2:6])
```

To apply the test, we simply execute the function on each of the two variables that we have just created:

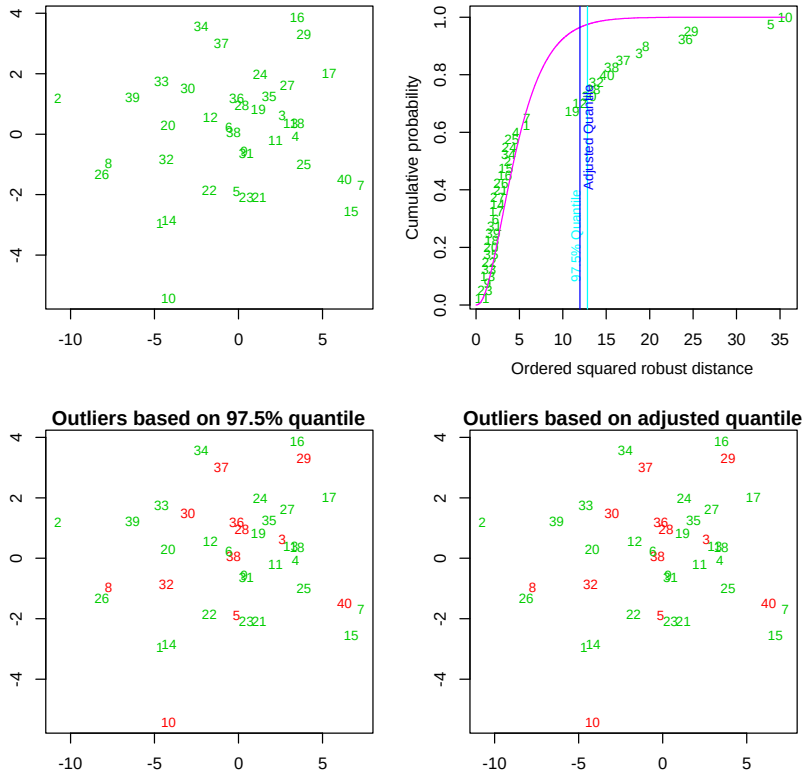
```
mshapiro.test(Yr_1)
mshapiro.test(Yr_2)
mshapiro.test(Yr_3)
```

The output below shows the results of the three tests. It is clear that for years 1 and 2, the data deviate significantly from multivariate normality, both *p*s < .05; however, for year 3 there is no problem because the result of the Shapiro–Wilk test is non-significant, $p > .05$.

```
mshapiro.test(Yr_1)

        Shapiro-Wilk normality test

data:  Z
W = 0.7116, p-value = 0.0006605

> mshapiro.test(Yr_2)

        Shapiro-Wilk normality test

data:  Z
W = 0.8832, p-value = 0.04346

> mshapiro.test(Yr_3)

        Shapiro-Wilk normality test

 data:  Z
W = 0.886, p-value = 0.08593
```

We can also look for multivariate outliers using the *aq.plot()* function:

```
aq.plot(psychologyData[, 2:6])
```



The resulting plots above show the case numbers (i.e. the row number in the dataframe), and you need to be looking for values in red in all but the top right graph. In this case it seems that there are likely to be many outliers! In the top right plot, you are looking for any cases that fall to the right of the vertical line labeled *97.5% Quantile*. Again, it seems that there are likely to be a lot of outliers. We could consider deleting all the outliers and see if it makes the data multivariate normal, but because there are so many likely outliers it makes more sense to leave all the cases in and conduct a robust MANOVA to combat the effects of the outliers.

Main analysis

Before running the main analysis we need to set some contrasts for the **group** variable. When thinking about the best contrasts to set, it makes some sense for the first contrast to compare year 1 with the other two years combined and then for the second contrast to compare year 2 with year 3. We can set these contrasts by executing:

```
Yr1_vs_Yr2&3<-c(-2, 1, -1)
Yr2_vs_Yr3 <-c(0, -1, 1)
contrasts(psychologyData$group)<-cbind(Yr1_vs_others, Yr2_vs_Yr3)
```

To run the main analysis, first need to combine our five outcome variables, **exper**, **stats**, **social**, **develop** and **personal**, into a single outcome object. We can do this by executing:

```
outcome<-cbind(psychologyData$exper, psychologyData$stats, psychologyData$social,
 psychologyData$develop, psychologyData$person)
```

This command creates an object called *outcome*, which contains the outcome variables of the *psychologyData* dataframe pasted together in columns. Therefore, for this example, we could estimate the model by executing:

```
psychologyModel<-manova(outcome ~ group, data = psychologyData)
```

To see the output of the model we use the *summary* command; by default, **R** produces Pillai's trace, which is a sensible choice:

```
summary(psychologyModel, intercept = TRUE)
```

```
Df Pillai approx F num Df den Df  Pr(>F)
(Intercept)  1 0.9604   160.09      5     33 < 2e-16 ***
group        2 0.5104     2.33     10     68 0.01995 *
Residuals   37
```

The main multivariate statistics are shown above. The column of real interest is the one containing the significance values of the *F*-ratio for **group**. For these data, the test statistic for **group** is significant, $p < .05$. From this result we should probably conclude that the profile of knowledge across different areas of psychology does indeed change across the three years of the degree. The nature of this effect is not clear from the multivariate test statistic.

To follow up the analysis with univariate analysis of the individual outcome measures, we can simply execute:

```
summary.aov(psychologyModel)
```

This produces the output below, which shows the ANOVA summary table for the dependent variables. The table labeled *Response 1* is for the **exper** variable, *Response 2* is for the **stats** variable, *Response 3* is for the **social** variable, *Response 4* is for the **develop** variable and *Response 5* is for the **person** variable.

```
Response 1 :
            Df Sum Sq Mean Sq F value  Pr(>F)
group        2  18.43  9.2148  2.4609 0.09922 .
Residuals   37 138.54  3.7445

 Response 2 :
            Df Sum Sq Mean Sq F value  Pr(>F)
group        2   52.5 26.2522  2.9668 0.06382 .
Residuals   37  327.4  8.8485

Response 3 :
            Df  Sum Sq Mean Sq F value Pr(>F)
group        2  23.584 11.7922   1.941 0.1579
Residuals   37 224.791  6.0754

 Response 4 :
            Df  Sum Sq Mean Sq F value  Pr(>F)
group        2  37.717 18.8587  3.1142 0.05623 .
Residuals   37 224.058  6.0556

Response 5 :
            Df  Sum Sq Mean Sq F value  Pr(>F)
group        2  39.415 19.7076  3.0438 0.05973 .
Residuals   37 239.560  6.4746
```

The row labelled *group* contains an ANOVA summary table for test scores in the five subject areas; experimental, statistics, social, developmental and personality, respectively. The values of *p* indicate that there was a non-significant difference between student groups in terms of all areas of psychology (all *p*s are greater than .05). The multivariate test statistics led us to conclude that the student groups *did* differ significantly across the types of psychology yet the univariate results contradict this (again ... I really should stop making up data sets that do this!).

We don't need to look at contrasts because the univariate tests were non-significant; instead, to see how the dependent variables interact, we need to carry out a discriminant function analysis.

To carry out discriminant function analysis we would use the *ida()* function. You may have already noticed that the *psychologyData* data set has unequal group sizes (yr1 = 11, yr2 = 16, yr3 = 13) and, if you remember from the chapter, when you have unequal group sizes it is a good idea to base the prior probabilities on the sample size of the group. We can do this using the *prior* option of *ida()*. and execute:

```
psychologyDFA<-lda(group ~ exper + stats + social + develop + person, data =
 psychologyData, prior = c(11, 16, 13)/40)
```

This creates a model called *psychologyDFA*. To see this model execute the name of the model:

```
psychologyDFA
```

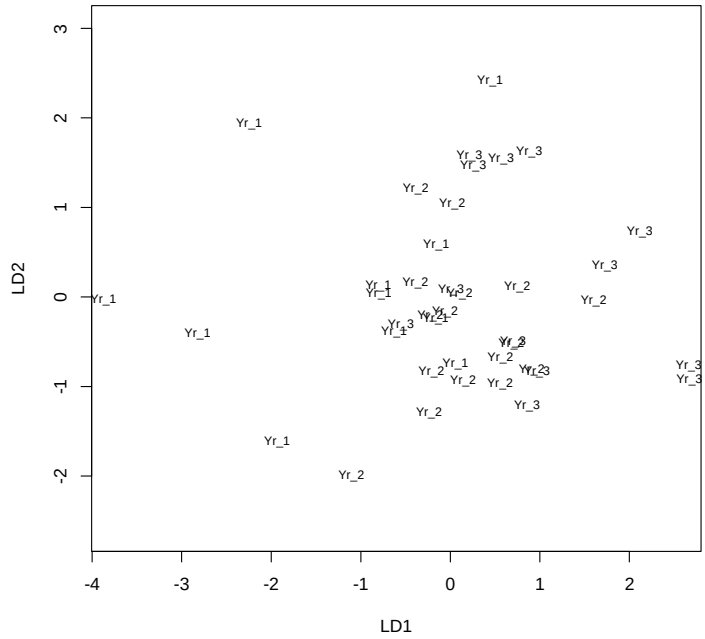
```
Prior probabilities of groups:
 Yr_1  Yr_2  Yr_3
0.275 0.400 0.325

Group means:
        exper     stats    social   develop    person
Yr_1 5.636364  7.545455 10.363636 11.000000 10.636364
Yr_2 5.500000  8.687500  8.562500  8.875000  8.437500
Yr_3 7.000000 10.461538  8.769231  8.769231  8.384615

Coefficients of linear discriminants:
               LD1          LD2
exper    0.1896714  0.407918384
stats    0.3097717 -0.027315706
social  -0.1433943  0.129480761
develop -0.2510129  0.005388236
person  -0.1022320  0.084955363

Proportion of trace:
   LD1    LD2
0.9057 0.0943
```

The main part of the output above tells us the coefficients of the linear discriminants. We can see that there were two variates. Looking at the first variate, it's clear that statistics has the greatest contribution to this. Most interesting is that on the first variate, statistics and experimental psychology have positive weights, whereas social, developmental and personality have negative weights. This suggests that the group differences are explained by the difference between experimental psychology and statistics compared to other areas of psychology. On the second variate, all variables have positive weights except for statistics, which has a negative weight. This suggests that the group differences are explained by the difference between statistics and all the other areas of psychology, although the variables don't have very strong relationships with the second variate, suggesting that perhaps the group differences shown by the MANOVA can be explained in terms of *one* underlying dimension only.

Finally, the proportion trace shows us that the first variate accounts for 90.57% of the variance compared to the second variate, which only accounts for 9.43%, again suggesting that the MANOVA can be explained in terms of *one* underlying dimension only.

It is also useful to have a look at a plot of the discriminant scores broken down by group membership. This can be obtained by using the *plot()* function on our model:

```
plot(psychologyDFA)
```



The above plot for these data indicates that variate 1 discriminates the first years from subsequent years (look at the horizontal distance between these centroids).

Overall, we could conclude that different years are discriminated by different areas of psychology. In particular, it seems as though statistics and aspects of experimentation (compared to other areas of psychology) discriminate between first-year undergraduates and subsequent years. From the means, we could interpret this as first years struggling with statistics and experimental psychology (compared to other areas of psychology), but their ability improves across the three years. However, for other areas of psychology, first years are relatively good but their abilities decline over the three years. Put another way, psychology degrees improve only your knowledge of statistics and experimentation.☺

Reporting results from discriminant analysis

- ✓ The MANOVA was followed up with discriminant analysis, which revealed two discriminant functions. The first explained 90.57% of the variance, whereas the second explained only 9.43%. The coefficients of the discriminant functions revealed that function 1 differentiated experimental psychology (b = 0.19) and statistics (b = 0.31) from the other areas of psychology: social (b = –0.14), developmental (b = –0.25) and personality (b = –0.10). For the second variate, the coefficients of the discriminant functions differentiated statistics (b = –0.03) from all other areas of psychology; experimental (b = 0.41), social (b = 0.13), developmental (b = 0.01) and personality (b = 0.08). The discriminant function plot showed that the first function discriminated year 1 students from the other two years, and the second function differentiated the year 2 and year 3 students.

Robust analysis

If you remember back to when we tested the assumptions in this example, the data were found to deviate from normal. Therefore, it would be appropriate to run a robust MANOVA on these data.

The robust functions need the data to be in wide format rather than long. Essentially we want levels of the independent variable (**group**) and outcome measures (**exper**, **stats**, **social**, **develop** and **person**) to be represented in different columns. The outcome measures are already spread across different columns (**exper**, **stats**, **social**, **develop** and **person**), but the **group** variable is differentiated by different rows of data. Therefore, we need to take the rows representing people who were in years 1, 2 and 3 and shift them into columns alongside the columns currently labelled **exper**, **stats**, **social**,

develop and **person**. We can do this restructuring using the *melt()* and *cast()* functions. To get the restructuring to work, we need to add a variable to our dataframe that identifies the rows in the wide format. This is a little bit trickier for this particular example because our data set contains unequal group sizes: year 1 has 11 students, year 2 has 16 students and year 3 has 13 students. Therefore, we need to tell **R** that we want the variable **row** to list 1–11 for year 1, 1–16 for year 2 and 1–13 for year 3. We can do this by executing:

```
psychologyData$row<-rep(c(1:11, 1:16, 1:13))
```

Executing this command creates a variable **row** in the dataframe *psychologyData*, that is, the numbers 1 to 11 then the numbers 1 to 16 and finally the numbers 1 to 16. The structure of the data will be the same as before – it's just that we have a new variable called **row** that identifies the scores within each year group.

Next we need to make the data molten so that we can cast the data into the wide format:

```
psychologyMelt<-melt(psychologyData, id = c("group", "row"), measured = c("exper",
"stats", "social", "develop", "person"))
```

We can then take the dataframe *psychologymelt* and assign informative names to each column:

```
names(psychologyMelt)<-c("group", "row", "Outcome_Measure", "value")
```

Finally, we want to cast our data into the wide format using *cast()* and then remove the variable **row** from the dataframe as we do not want it for the robust analysis:

```
psychologyRobust<-cast(psychologyMelt, row ~ group + Outcome_Measure, value = "value")

psychologyRobust$row<-NULL
```

You can view the dataframe by executing its name:

```
 psychologyRobust
```

To save space, I'm not going to paste the data here, but you should have a look at it because it is important to note the order of the columns: the hierarchy of the independent variables is **group** followed by **Outcome_Measure**. It is important that you specify the variables in this order because this arranges the data correctly for Wilcox's functions.

We need only specify the dataframe (*psychologyRobust*) and then the number of groups (three in this case) and the number of outcome measures (five in this case). Therefore, we can do a robust MANOVA based on ranks by executing:

```
mulrank(3, 5, psychologyRobust)
```

```
$test.stat
[1] 2.286823

$nu1
[1] 5.163434

$p.value
           [,1]
[1,] 0.04148764

$N
[1] 40

$q.hat
       [,1]        [,2]        [,3]       [,4]        [,5]
[1,] 0.4431818  0.3602273  0.6295455  0.6681818  0.6443182
[2,] 0.4352273  0.5238636  0.4454545  0.4159091  0.4170455
[3,] 0.6181818  0.6897727  0.5284091  0.4750000  0.4204545
```

The output for the robust MANOVA is shown above. We are given a test statistic for the year group (*$test.stat*) as well as the corresponding *p*-value (*$p.value*). We could conclude that there was a significant main effect of year of study on psychology ability, $F = 2.29$, $p < .05$. The numbers under

$q.hat tell us the relative effects (i.e., the typical ranks across the combinations of groups in the rows and outcome measures in the columns).

We could relabel this grid as:

```
            [exper]          [stats]       [social]        [develop]       [person]
[Year1]   0.4431818       0.3602273       0.6295455       0.6681818       0.6443182
[Year2,]  0.4352273       0.5238636       0.4454545       0.4159091       0.4170455
[Year3,]  0.6181818       0.6897727       0.5284091       0.4750000       0.4204545
```

This shows that for year 1, the ranks were highest and fairly similar for social, developmental and personality ability (0.63, 0.67 and 0.64). This indicates that in year 1 students are better at social, developmental and personality psychology than they are at experimental and statistics. For year 2, all the ranks were fairly similar (0.44, 0.52, 0.45, 0.42 and 0.42), although they were slightly higher for stats (0.52), suggesting that in year 2 student's ability is evenly spread across the five areas of psychology. Finally, for year 3 the ranks were highest for experimental and statistics (0.62 and 0.69) suggesting that by year 3 students are better at statistics and experimental psychology than they are in the other areas of psychology. This reconfirms what we found in the discriminant analysis earlier, that different years are discriminated by different areas of psychology. In particular, it seems as though statistics and aspects of experimentation (compared to other areas of psychology) discriminate between first year undergraduates and subsequent years. It seems that first years struggle with statistics and experimental psychology (compared to other areas of psychology) but their ability improves across the three years. However, for other areas of psychology, first years are relatively good but their abilities decline over the three years.

Reporting results of robust MANOVA

- ✓ A MANOVA was conducted on the ranked data using Munzel and Brunner's (2000) method implemented in **R** using the *mulrank()* function (Wilcox, 2005). There was a significant main effect of the type of treatment on outcomes of success, $F = 2.29$, $p < .05$.