Comparing two means

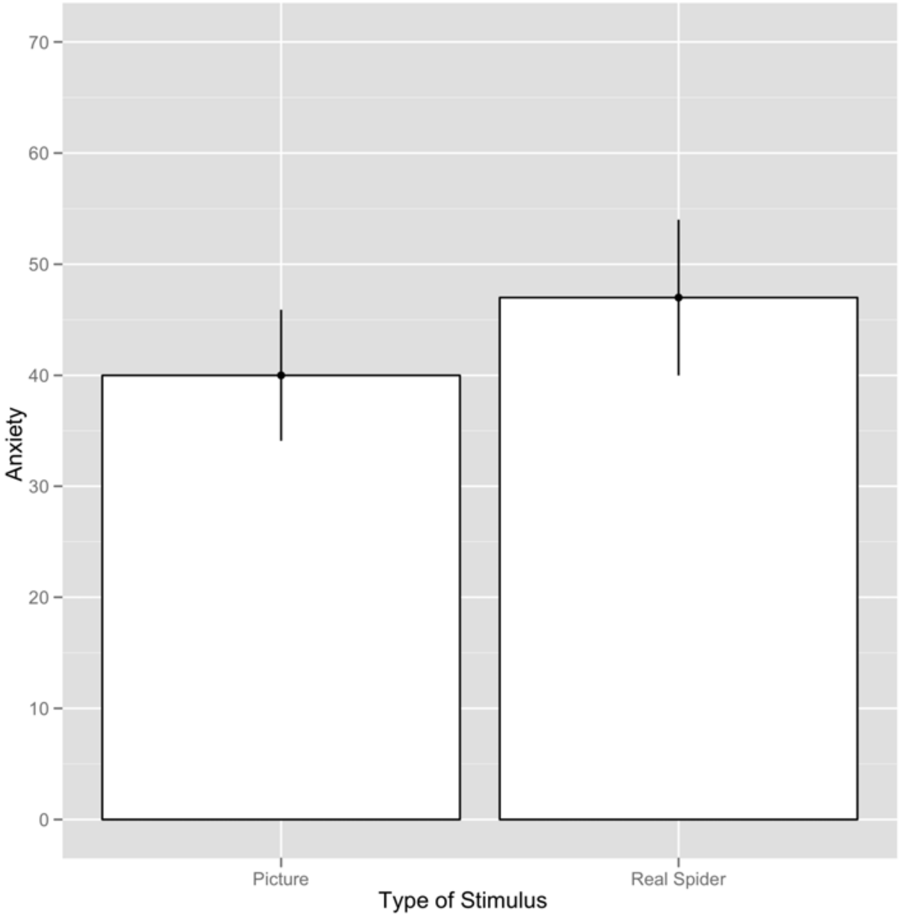
Self-test answers

- Enter these data into a dataframe called *spiderLong*. Plot an error bar graph of the spider data.

```
Group<-gl(2, 12, labels = c( "Picture", "Real Spider"))

Anxiety<-c(30, 35, 45, 40, 50, 35, 55, 25, 30, 45, 40, 50, 40, 35, 50, 55, 65, 55, 50,
35, 30, 50, 60, 39)

spiderLong<-data.frame(Group, Anxiety)
```

You can check your data entry against the file **spiderLong.dat**. To create the graph execute:

```
bar <- ggplot(spiderLong, aes(Group, Anxiety))

bar + stat_summary(fun.y = mean, geom = "bar", fill = "White", colour = "Black") +
stat_summary(fun.data = mean_cl_normal, geom = "pointrange") + labs(x = "Type of
Stimulus", y = "Anxiety") + scale_y_continuous(limits = c(0, 70), breaks = seq(from =
0, to = 70, by = 10))
```



- Enter these data into a dataframe called *spiderWide*.

```
picture<-c(30, 35, 45, 40, 50, 35, 55, 25, 30, 45, 40, 50)
real<-c(40, 35, 50, 55, 65, 55, 50, 35, 30, 50, 60, 39)
spiderWide<-data.frame(picture, real)
```

You can check your data entry against the file **spiderWide.dat**.

- Create an error bar chart of the mean of the adjusted values that you have just made (**real_adj** and **picture_adj**).

First we need to make the data long. Add a variable to identify the participant (i.e. an ID variable):

```
spiderWide$id<-gl(12, 1, labels = c(paste("P", 1:12, sep = "_")))
```

The data now look like this:

```
   picture real pMean   adj picture_adj real_adj pMean2   id
1       30   40  35.0   8.5        38.5     48.5   43.5  P_1
2       35   35  35.0   8.5        43.5     43.5   43.5  P_2
3       45   50  47.5  -4.0        41.0     46.0   43.5  P_3
4       40   55  47.5  -4.0        36.0     51.0   43.5  P_4
5       50   65  57.5 -14.0        36.0     51.0   43.5  P_5
6       35   55  45.0  -1.5        33.5     53.5   43.5  P_6
7       55   50  52.5  -9.0        46.0     41.0   43.5  P_7
8       25   35  30.0  13.5        38.5     48.5   43.5  P_8
9       30   30  30.0  13.5        43.5     43.5   43.5  P_9
10      45   50  47.5  -4.0        41.0     46.0   43.5 P_10
11      40   60  50.0  -6.5        33.5     53.5   43.5 P_11
12      50   39  44.5  -1.0        49.0     38.0   43.5 P_12
```

Next we need to use *melt()* to make the data molten/long:

```
adjustedData<-melt(spiderWide, id = c("id", "picture", "real", "pMean", "adj", 
"pMean2"), measured = c("picture_adj", "real_adj"))
```

(Note that if you haven’t got the same variables in the original dataframe as me, you might need to remove some of the variable names from the command above.) The data now look like this:

```
     id picture real pMean   adj pMean2    variable value
1   P_1      30   40  35.0   8.5   43.5 picture_adj  38.5
2   P_2      35   35  35.0   8.5   43.5 picture_adj  43.5
3   P_3      45   50  47.5  -4.0   43.5 picture_adj  41.0
4   P_4      40   55  47.5  -4.0   43.5 picture_adj  36.0
5   P_5      50   65  57.5 -14.0   43.5 picture_adj  36.0
6   P_6      35   55  45.0  -1.5   43.5 picture_adj  33.5
7   P_7      55   50  52.5  -9.0   43.5 picture_adj  46.0
8   P_8      25   35  30.0  13.5   43.5 picture_adj  38.5
9   P_9      30   30  30.0  13.5   43.5 picture_adj  43.5
10 P_10      45   50  47.5  -4.0   43.5 picture_adj  41.0
11 P_11      40   60  50.0  -6.5   43.5 picture_adj  33.5
12 P_12      50   39  44.5  -1.0   43.5 picture_adj  49.0
13  P_1      30   40  35.0   8.5   43.5    real_adj  48.5
14  P_2      35   35  35.0   8.5   43.5    real_adj  43.5
15  P_3      45   50  47.5  -4.0   43.5    real_adj  46.0
16  P_4      40   55  47.5  -4.0   43.5    real_adj  51.0
17  P_5      50   65  57.5 -14.0   43.5    real_adj  51.0
18  P_6      35   55  45.0  -1.5   43.5    real_adj  53.5
19  P_7      55   50  52.5  -9.0   43.5    real_adj  41.0
20  P_8      25   35  30.0  13.5   43.5    real_adj  48.5
21  P_9      30   30  30.0  13.5   43.5    real_adj  43.5
22 P_10      45   50  47.5  -4.0   43.5    real_adj  46.0
23 P_11      40   60  50.0  -6.5   43.5    real_adj  53.5
24 P_12      50   39  44.5  -1.0   43.5    real_adj  38.0
```

We don’t want the variables in columns 2 to 6, so let’s delete them by executing:

```
adjustedData<-adjustedData[, -c(2:6)]
```

This command re-creates the *adjustedData* dataframe from itself, but *[, -c(2:6)]* selects out part of the dataframe: the fact there is nothing before the comma means we retain all of the rows, and *-c(2:6)* after the comma means we delete columns 2 to 6 (the minus sign means delete).

We should just have three variables left in the dataframe:

```
     id    variable value
1   P_1 picture_adj  38.5
2   P_2 picture_adj  43.5
3   P_3 picture_adj  41.0
4   P_4 picture_adj  36.0
5   P_5 picture_adj  36.0
6   P_6 picture_adj  33.5
7   P_7 picture_adj  46.0
8   P_8 picture_adj  38.5
9   P_9 picture_adj  43.5
10 P_10 picture_adj  41.0
11 P_11 picture_adj  33.5
12 P_12 picture_adj  49.0
13  P_1    real_adj  48.5
14  P_2    real_adj  43.5
15  P_3    real_adj  46.0
16  P_4    real_adj  51.0
17  P_5    real_adj  51.0
18  P_6    real_adj  53.5
19  P_7    real_adj  41.0
20  P_8    real_adj  48.5
21  P_9    real_adj  43.5
22 P_10    real_adj  46.0
23 P_11    real_adj  53.5
24 P_12    real_adj  38.0
```

Let’s give the variables some useful names:

```
names(adjustedData)<-c("id", "Group", "Anxiety_Adj")
```

and rename the factor levels of **Group**:

```
adjustedData$Group<-factor(adjustedData$Group, labels = c("Spider Picture", "Real
Spider"))
```

The dataframe now looks like this:

```
     id          Group Anxiety_Adj
1   P_1 Spider Picture        38.5
2   P_2 Spider Picture        43.5
3   P_3 Spider Picture        41.0
4   P_4 Spider Picture        36.0
5   P_5 Spider Picture        36.0
6   P_6 Spider Picture        33.5
7   P_7 Spider Picture        46.0
8   P_8 Spider Picture        38.5
9   P_9 Spider Picture        43.5
10 P_10 Spider Picture        41.0
11 P_11 Spider Picture        33.5
12 P_12 Spider Picture        49.0
13  P_1    Real Spider        48.5
14  P_2    Real Spider        43.5
15  P_3    Real Spider        46.0
16  P_4    Real Spider        51.0
17  P_5    Real Spider        51.0
18  P_6    Real Spider        53.5
19  P_7    Real Spider        41.0
20  P_8    Real Spider        48.5
21  P_9    Real Spider        43.5
22 P_10    Real Spider        46.0
23 P_11    Real Spider        53.5
24 P_12    Real Spider        38.0
```

To plot the graph execute:

```
bar <- ggplot(adjustedData, aes(Group, Anxiety_Adj))
```

```
bar + stat_summary(fun.y = mean, geom = "bar", fill = "White", colour = "Black") +
stat_summary(fun.data = mean_cl_normal, geom = "pointrange") + labs(x = "Type of
Stimulus", y = "Anxiety") + scale_y_continuous(limits = c(0, 70), breaks = seq(from =
0, to = 70, by = 10))
```

- Using the *lm()* function, un a regression on the data in **spiderLong.dat** with **Goup** as the predictor and **Anxiety** as the outcome.

Load the data:

```
spiderLong<-read.delim("SpiderLong.dat", header = TRUE)
```

Create and display the regression model:

```
t.test.GLM<-lm(Anxiety ~ Group, data = spiderLong)
summary(t.test.GLM)
```

- Use *ggplot2* to produce a boxplot and bar chart with error bars showing confidence intervals for the spider data.

Boxplot:

```
spiderBoxplot <- ggplot(spiderLong, aes(Group, Anxiety))

spiderBoxplot + geom_boxplot() + labs(x = "Type of Stimulus", y = "Anxiety") +
scale_y_continuous(limits = c(0, 100), breaks = seq(from = 0, to = 100, by = 10))
```

Erro bar chart:

```
bar <- ggplot(spiderLong, aes(Group, Anxiety))

bar + stat_summary(fun.y = mean, geom = "bar", fill = "White", colour = "Black") +
stat_summary(fun.data = mean_cl_normal, geom = "pointrange") + labs(x = "Type of
Stimulus", y = "Anxiety") + scale_y_continuous(limits = c(0, 60), breaks = seq(from =
0, to = 60, by = 10))
```

- Using the **spiderWide.dat** data, compute the differences between the picture and real condition and check the assumption of normality for these differences.

First compute the differences:

```
spiderWide$diff<-spiderWide$real-spiderWide$picture
```

Executing this command creates a new variable **diff** in the dataframe that contains the values of the picture condition subtracted from the real condition. The dataframe now looks like this:

```
   picture real diff
1       30   40   10
2       35   35    0
3       45   50    5
4       40   55   15
5       50   65   15
6       35   55   20
7       55   50   -5
8       25   35   10
9       30   30    0
10      45   50    5
11      40   60   20
12      50   39  -11
```

We can see, for example, that the difference for the first participant was 10. Now do some normality tests using the *stat.desc()* function:

```
stat.desc(spiderWide$diff, basic = FALSE, desc = FALSE, norm = TRUE)
```

Note that I have used *desc = FALSE* to switch off the descriptive statistics, which means that the output will contain only the normality tests (everything else is set to FALSE).

```
  skewness   skew.2SE   kurtosis   kurt.2SE normtest.W normtest.p
-0.2464810 -0.1933785 -1.2342159 -0.5007991  0.9557904  0.7224811
```

The output shows that the distribution of differences is not significantly different from normal, *W* = 0.956, *p* > .05. As such, it looks as though we can assume that our differences are normal and that, therefore, the sampling distribution of these differences is normal too. Happy days!

Labcoat Leni's real research

You don't have to be mad to work here, but it helps

Problem

Board, B. J., & Fritzon, K. (2005). *Psychology, Crime & Law*, *11*, 17–32.

In the UK you often see the 'humorous' slogan 'You don't have to be mad to work here, but it helps' displayed in work places. Well, Board and Fritzon (2005) took this a step further by measuring whether 39 senior business managers and chief executives from leading UK companies were mad (well, had personality disorders). They tested them with the Minnesota Multiphasic Personality Inventory Scales for DSM III Personality Disorders (MMPI-PD), which is a well-validated measure of 11 personality disorders: Histrionic, Narcissistic, Antisocial, Borderline, Dependent, Compulsive, Passive-aggressive, Paranoid, Schizotypal, Schizoid and Avoidant. They needed a comparison group, and what better one to choose than 317 legally classified psychopaths at Broadmoor Hospital (a famous high-security psychiatric hospital in the UK).

The authors report the means and standard deviations for these two groups in Table 2 of their paper. Using these values we can run *t*-tests on these means. The data from Board and Fritzon's Table 2 are in the file **Board and Fritzon 2005.dat**. Use this file to run *t*-tests to see whether managers score higher on personality disorder questionnaires than legally classified psychopaths. Report these results. What do you conclude?

First of all we need to load in the data (I am going to rename the data *madData* as it will be quicker to type than *Board&Fritzon(2005)*) the data by executing:

```
 madData<-read.delim("Board&Fritzon(2005).dat", header = TRUE)
```

We can view the dataframe by executing the name of the dataframe:

```
 madData
```

The data look like this:

```
             Outcome    x1    x2  sd1  sd2 n1  n2
1         Histrionic 13.33  8.88 3.48 3.67 39 317
2       Narcissistic 15.58 14.54 3.72 4.42 39 317
3         Compulsive  7.35  6.92 2.43 3.38 39 317
4         Antisocial  8.64 12.43 3.75 4.33 39 317
5         Borderline  9.23 15.77 4.09 3.82 39 317
6          Dependent  5.92 12.06 2.98 3.77 39 317
7  Passive-aggressive  5.56  7.87 2.78 3.64 39 317
8           Paranoid  5.82 13.79 2.86 5.61 39 317
9        Schizotypal  9.17 22.85 5.09 7.73 39 317
10          Schizoid  6.61 12.82 3.48 4.58 39 317
11          Avoidant 12.79 21.93 7.06 8.70 39 317
```

The columns represent the following:
Outcome: a string variable that tells us which personality disorder the numbers in each row relate to.
X1: the mean of the managers group.
X2: the mean of the psychopaths group.
sd1: the standard deviation of the managers group.
sd2: the standard deviation of the psychopaths group.
n1: the number of managers tested.
n2: the number of psychopaths tested.

OK, now we want to compute *t*-tests from the means and standard deviations as we do not have the raw scores. To do this, first make sure that you have executed the function from the chapter (you only need to execute this function once):

```
ttestfromMeans<-function(x1, x2, sd1, sd2, n1, n2)
{     df<-n1 + n2 - 2
      poolvar <- (((n1-1)*sd1^2)+((n2-1)*sd2^2))/df
      t <- (x1-x2)/sqrt(poolvar*((1/n1)+(1/n2)))
      sig <- 2*(1-(pt(abs(t),df)))
      paste("t(df = ", df, ") = ", t, ", p = ", sig, sep = "")

}
```

Once you have executed the above function you can plug in the numbers into:

```
ttestfromMeans(x1, x2, sd1, sd2, n1, n2)
```

For the current dataframe (*madData*) this would be:

```
ttestfromMeans(madData$x1, madData$x2, madData$sd1, madData$sd2,
madData$n1,madData$n2)
```

Your output should look like this:

```
[1]  "t(df = 354) = 7.18447030378608,  p = 4.01678690309382e-12"
[2]  "t(df = 354) = 1.4088189871865,   p = 0.159766433758804"
[3]  "t(df = 354) = 0.769932839467559, p = 0.441853202748396"
[4]  "t(df = 354) = -5.22870588198729, p = 2.92575248028726e-07"
[5]  "t(df = 354) = -10.0107431874896, p = 0"
[6]  "t(df = 354) = -9.79693901591482, p = 0"
[7]  "t(df = 354) = -3.82635047509956, p = 0.000153717979781698"
[8]  "t(df = 354) = -8.72584921731946, p = 0"
[9]  "t(df = 354) = -10.7613018106178, p = 0"
[10] "t(df = 354) = -8.17797383133495, p = 5.32907051820075e-15"
[11] "t(df = 354) = -6.30773035698329, p = 8.4501161623507e-10"
```

Let’s compute effect sizes by executing:

```
df<-ttestfromMeans$df
t<-ttestfromMeans$t
r <- sqrt(t^2/(t^2+df))
round(r, 3)
```

```
0.357 0.075 0.041 0.268 0.470 0.462 0.199 0.421 0.496 0.399 0.318
```

We can report that managers scored significantly higher than psychopaths on histrionic personality disorder, *t*(354) = 7.18, *p* < .001, *r* = .36. There were no significant differences between groups on narcissistic personality disorder, *t*(354) = 1.41, *p* > .05, *r* = .08 , or compulsive personality disorder, *t*(354) = 0.77, *p* > .05, *r* = .04. On all other measures, psychopaths scored significantly higher than managers: antisocial personality disorder, *t*(354) = −5.23, *p* < .001, *r* = .27; borderline personality disorder, *t*(354) = −10.01, *p* < .001, *r* = .47; dependent personality disorder, *t*(354) = −9.80, *p* < .001, *r* = .46; passive-aggressive personality disorder, *t*(354) = −3.83, *p* < .001, *r* = .20; paranoid personality disorder, *t*(354) = −8.73, *p* < .001, *r* = .42; schizotypal personality disorder, *t*(354) = −10.76, *p* < .001, *r* = .50; schizoid personality disorder, *t*(354) = −8.18, *p* < .001, *r* = .40; avoidant personality disorder, *t*(354) = −6.31, *p* < .001, *r* = .32.

The results show the presence of elements of personality disorder in the senior business manager sample, especially those most associated with psychopathic personality disorder. The senior business manager group showed significantly higher levels of traits associated with histrionic personality disorder than psychopaths. They also did not significantly differ from psychopaths in narcissistic and compulsive personality disorder traits. These findings could be an issue of power (effects were not detected but are present). The effect sizes (*r*) can help us out here and these are quite small (.08 and .04), which can give us confidence that there really isn’t a difference between psychopaths and managers on these traits. Broad and Fritzon (2005) conclude that: ‘At a descriptive level this translates to: superficial charm, insincerity, egocentricity, manipulativeness (histrionic), grandiosity, lack of empathy, exploitativeness, independence (narcissistic), perfectionism, excessive devotion to work, rigidity, stubbornness, and dictatorial tendencies (compulsive). Conversely, the senior business manager group is less likely to demonstrate physical aggression, consistent irresponsibility with work and finances, lack of remorse (antisocial), impulsivity, suicidal gestures, affective instability (borderline), mistrust (paranoid), and hostile defiance alternated with contrition (passive/aggressive).’ And these people are in charge of large companies like Sage Publications Ltd. Hmm, suddenly a lot of things make sense.

Smart Alex’s solutions

Task 1

- One of my pet hates is ‘pop psychology’ books. Along with banishing Freud from all bookshops, it is my avowed ambition to rid the world of these rancid putrefaction-ridden wastes of trees. Not only do they give psychology a very bad name by stating the bloody obvious and charging people for the privilege, but they are also considerably less enjoyable to look at than the trees killed to produce them (admittedly the same could be said for the turgid tripe that I produce in the name of education, but let’s not go there just for now!). Anyway, as part of my plan to rid the world of popular psychology I did a little experiment. I took two groups of people who were in relationships and randomly assigned them to one of two conditions. One group read the famous popular psychology book *Women Are from Bras, Men Are from Penis*, whereas another group read *Marie Claire*. I tested only 10 people in each of these groups, and the dependent variable was an objective measure of their happiness with their relationship after reading the book. I didn't make any specific prediction about which reading material would improve relationship happiness. The data are in the file **Penis.dat**. Analyse them with the appropriate *t*-test.

First of all, load in the data:

```
 penisData<-read.delim("Penis.dat", header = TRUE)
```

If we then execute the name of the dataframe:

```
 penisData
```

we can see the data look like this:

```
    book  happy
1     1    19
2     1    28
3     1    23
4     1    20
5     1    22
6     1    22
7     1    18
8     1    19
9     1    16
10    1    13
11    2    27
12    2    15
13    2    20
14    2    30
15    2    28
16    2    23
17    2    28
18    2    27
19    2    20
20    2    24
```

These data were originally entered into SPSS and as you can see by looking at the categorical variable **book**, the values have been exported as numbers rather than as text. Therefore, we need to tell **R** that **book** is a factor by executing:

```
penisData$book<-factor(penisData$book, levels = c(1:2), labels = c("Women are from
Bras, Men are from Penis", "Marie Claire"))
```

If we then execute:

```
penisData
```

we can see that the dataframe now looks like this:

```
                                   book      happy
1  Women are from Bras, Men are from Penis    19
2  Women are from Bras, Men are from Penis    28
3  Women are from Bras, Men are from Penis    23
4  Women are from Bras, Men are from Penis    20
5  Women are from Bras, Men are from Penis    22
6  Women are from Bras, Men are from Penis    22
7  Women are from Bras, Men are from Penis    18
8  Women are from Bras, Men are from Penis    19
9  Women are from Bras, Men are from Penis    16
10 Women are from Bras, Men are from Penis    13
11                            Marie Claire    27
12                            Marie Claire    15
13                            Marie Claire    20
14                            Marie Claire    30
15                            Marie Claire    28
16                            Marie Claire    23
17                            Marie Claire    28
18                            Marie Claire    27
19                            Marie Claire    20
20                            Marie Claire    24
```

OK, now we need to run an independent *t*-test. The reason for running an independent *t*-test rather than a dependent *t*-test on these data is that a different group of participants were used in each of the two conditions.

We could run a *t*-test to compare the mean happiness after reading *Women are from Bras, Men Are from Penis* to the mean happiness after reading *Marie Claire* by executing:

```
penis.t.test<-t.test(happy ~ book, data = penisData)
 penis.t.test
```

R output for the independent *t*-test:

```
Welch Two Sample t-test

data:  happy by book
t = -2.1249, df = 17.676, p-value = 0.04796
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -8.35799532 -0.04200468
sample estimates:
mean in group Women are from Bras, Men are from Penis   mean in group Marie Claire
                   20.0                                                 24.2
```

Calculating the effect size

We know the value of *t* and the *df* from the **R** output and so we can compute *r* by executing:

```
t<-penis.t.test$statistic[[1]]
df<-penis.t.test$parameter[[1]]
r <- sqrt(t^2/(t^2+df))
 round(r, 3)

[1] 0.451
```

If you think back to our benchmarks for effect sizes, this represents a fairly large effect (it is just below .5, the threshold for a large effect). Therefore, as well as being statistically significant, this effect is large and so represents a substantive finding.

Reporting the results

When you report any statistical test you usually state the finding to which the test relates, and then, in brackets, report the test statistic (usually with its degrees of freedom), the probability value of that test statistic, and more recently the American Psychological Association is, quite rightly, requesting an estimate of the effect size. To get you into good habits early, we'll start thinking about effect sizes now, before you get too fixated on Fisher's magic .05. In this example we know that the value of *t* was −2.12, that the degrees of freedom on which this was based were 18, and that it was significant at *p* = .048. This can all be obtained from the **R** output above. We can also see the means for each group. Based on what we learnt about reporting means, we could now write something like:

- ✓ On average, the reported relationship happiness after reading *Marie Claire* (*M* = 24.20, *SE* = 1.49), was significantly higher than after reading *Women Are from Bras, Men Are from Penis* (*M* = 20.00, *SE* = 1.30), *t*(18) = −2.12, *p* < .05, *r* = .45.

Task 2

- Imagine Twaddle and Sons, the publishers of *Women Are from Bras, Men Are from Penis*, were upset about my claims that their book was about as useful as a paper umbrella. They decided to take me to task and design their own experiment in which participants read their book and one of my books (Field and Hole) at different times. Relationship happiness was measured after reading each book. To maximize their chances of finding a difference they used a sample of 500 participants, but got each participant to take part in both conditions (they read both books). The order in which books were read was counterbalanced and there was a delay of 6 months between reading the books. They predicted that reading their wonderful contribution to popular psychology would lead to greater relationship happiness than reading some dull and tedious book about experiments. The data are in **Field&Hole.dat**. Analyse them using the appropriate *t*-test.

As always, begin by loading in the data:

```
Field_Hole<-read.delim("Field&Hole.dat", header = TRUE)
```

Now, we need to run a dependent *t*-test on these data because the participants all read both of the books. In other words, participants took part in both conditions of the experiment. We can conduct the dependent *t*-test by executing:

```
fieldHole.t.test<-t.test(Field_Hole$women, Field_Hole$statbook, paired = TRUE)
fieldHole.t.test
```

```
Paired t-test

data:  Field_Hole$women and Field_Hole$statbook
t = 2.7056, df = 499, p-value = 0.00705
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.41843 2.63757
sample estimates:
mean of the differences
                  1.528
```

Calculating the effect size

We know the value of *t* and the *df* from the **R** output and so we can compute *r* as follows:

```
t<-fieldHole.t.test$statistic[[1]]
df<-fieldHole.t.test$parameter[[1]]
r <- sqrt(t^2/(t^2+df))
round(r, 3)
```

```
[1] 0.12
```

If you think back to our benchmarks for effect sizes, this represents a small effect (it is just above .1, the threshold for a small effect). Therefore, although this effect is highly statistically significant, the size of the effect is very small and so represents a trivial finding.

Interpreting and writing the results

In this example, it would be tempting for Twaddle and Sons to conclude that their book produced significantly greater relationship happiness than our book. In fact, many researchers would write conclusions like this:

- The results show that reading *Women Are from Bras, Men Are from Penis* produces significantly greater relationship happiness than that book by smelly old Field and Hole. This result is highly significant.

However, to reach such a conclusion is to confuse statistical significance with the importance of the effect. By calculating the effect size we've discovered that although the difference in happiness after reading the two books is statistically very different, the size of effect that this represents is very small indeed. So, the effect is actually not very significant in real terms. A more correct interpretation might be to say:

- The results show that reading *Women Are from Bras, Men Are from Penis* produces significantly greater relationship happiness than that book by smelly old Field and Hole. However, the effect size was small, revealing that this finding was not substantial in real terms.

Of course, this latter interpretation would be unpopular with Twaddle and Sons who would like to believe that their book had a huge effect on relationship happiness.

Robust analysis

We could conduct a robust analysis on these data very easily. There are three options to choose from: *yuend()*, *ydbt()* and *bootdpci()*. However, you would only need to run one of them.

To run a robust analysis using *yuend()* you would execute:

```
yuend(Field_Hole$women, Field_Hole$statbook, tr=.2, alpha=.05)
```

If you execute the above command you will see the output in the table below, which shows that based on this robust test reading *Women Are from Bras, Men Are from Penis* produces significantly greater relationship happiness than that book by smelly old Field and Hole, T_y(299) = 2.62, *p* = .009.

To run a robust analysis using *ydbt()* you would execute:

```
ydbt(Field_Hole$women, Field_Hole$statbook, tr=.2, alpha=.05, nboot = 2000)
CI
```

If you execute this command you will see in the output table below, which shows that based on this robust test there is again a significant difference (because the confidence interval does not cross zero) in happiness scores after reading the two books, Y_t = 1.46 (0.36, 2.55), *p* = .009.

yuend() Output	*ydbt()* Output
\$ci [1] 0.3618989 2.5514344 \$siglevel [1] 0.009282477 \$dif [1] 1.456667 \$se [1] 0.5563043 \$teststat [1] 2.618471 \$df [1] 299	[1] "Taking bootstrap samples. Please wait." \$ci [1] 0.3613113 2.5520220 \$dif [1] 1.456667 \$p.value [1] 0.009

To run a robust analysis using *bootdpci()* you would execute:

```
bootdpci(Field_Hole$women, Field_Hole$statbook, est=tmean, nboot=2000)

$output
     con.num psihat p.value p.crit  ci.lower ci.upper
[1,]       1   1.87   0.001   0.05 0.7733333 2.966667
```

If you execute the *bootdpci()* function with the default settings you will see the output above, which shows that based on this robust test there is a significant difference (because the confidence interval does not cross zero and *p* is less than .05) in happiness scores across the two different books, $\bar{\psi}$ = 1.87 (0.77, 2.97), *p* = .001.