Chapter 9
T-Tests

9.1 Creating Boxplots

9.1.1 Boxplots for One Dependent Variable Separated by Groups (an Independent-Samples T-Test)

To look at the data from the Leow and Morgan-Short (2004) experiment, we will want to look at a boxplot that shows distributions on one variable for the two different groups. If you're following along with me, import the LeowMorganShort.sav file and save it as leow. To make a boxplot of one dependent variable split into two groups with R Commander, click GRAPHS > BOXPLOT to open up your dialogue box, as shown in Figure 9.1.

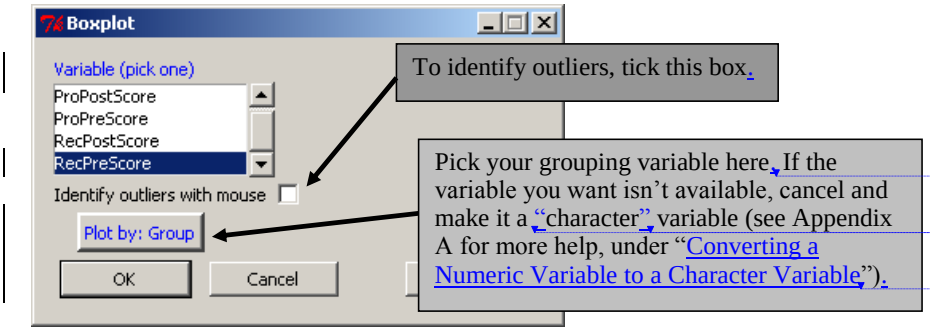

Figure 9.1 How to make a boxplot in R of one variable separated into groups.

From the boxplot for the pre-test recognition task in Figure 9.2, we can immediately see that the distribution of scores on this task is non-normal for both the think-aloud and non-think-aloud groups. Since this boxplot may look very strange to you, I will walk you through it.

Deleted: ;
Deleted: ‘
Deleted: ’
Deleted: Changing a numeric vector into a character vector

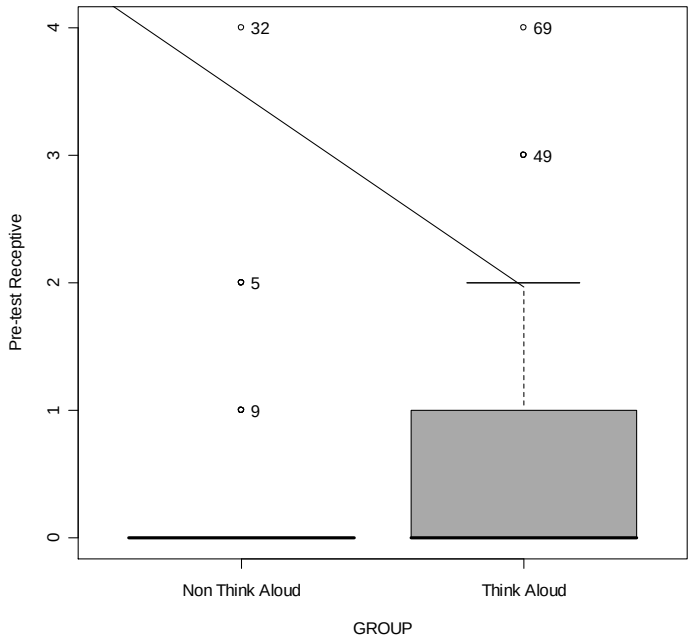

Figure 9.2 Leow and Morgan-Short's (2004) pre-experimental receptive measurement divided into groups in a boxplot.

For the non-think-aloud group in Figure 9.2, almost all of the scores are concentrated on zero, which indicates that this group knew very few of the four imperative forms they were shown. This was good, because the authors wanted to include people who had only a limited knowledge of imperative forms before the treatment. There were 39 participants in the non-think-aloud group, and we can see that 36 of them received a zero on this task. Three are then labeled as outliers because they got one, two, or four points on the pre-test.

For the think-aloud group in Figure 9.2, we can tell that a majority of the 38 participants received a zero, because the thick black median line lies on zero, and the median marks where half of the participants are below and half are above (in such a discrete-point test as this one, we will get a line only on the points themselves). The box of the boxplot contains 75% of the participants, so we know that at least 28 participants scored either one or zero ($38 \times .75=28.5$). Finally, the whisker of the boxplot for the think-aloud group extends to 2, which is categorized as the maximum value of the distribution. Two participants, one who scored three and the other who scored four, are classified as outliers to the distribution.

Notice that, although the median scores do not seem to be that different between the think-aloud and non-think-aloud groups, the distributions are clearly not normal, because the boxes and whiskers of the boxplot are not symmetrical around the median lines. Both are positively skewed (meaning the tail extends to the right if we were looking at a histogram, or if we turned the boxplot on its side), with the majority of scores concentrated toward the lower end of the scale. In the think-aloud group, the median line is clearly non-symmetric in relation to

the box. The whiskers are also non-symmetric, meaning the distribution is not normally distributed. The presence of outliers also means that the distribution is not normal.

The analysis of this command in R is:

```
boxplot(RecPreScore~Group, ylab="RecPreScore", xlab="Group",
data=leow)
identify(leow$Group, leow$RecPreScore)
```

boxplot(x~y)	Makes a boxplot modeling x as a function of y.
RecPreScore ~ Group,	Scores on the pre-test recognition task are modeled as a function of group.
ylab="RecPreScore" xlab="Group"	Gives custom names to x- and y-axis labels.
data= leow	Gives the data set a name.
identify(x)	This command will allow identification of points on the plot with the mouse.
leow$Group, leow$ RecPreScore	Names of the variables in the plot.

Creating a Boxplot in R for One Variable Split into Groups

1. On the R Commander drop-down menu, choose GRAPHS > BOXPLOT. Pick your dependent variable. To split the variable into groups, use the "Plot by Groups" button. To identify outliers, tick the box labeled "identify outliers with mouse."

The basic R code for this command is:

```
boxplot(RecPreScore~Group, ylab="RecPreScore", xlab="Group", data=leow)
```

Formatted: Font: Not Bold, Italic
Formatted: Font: Not Bold, Italic
Formatted: Font: Not Bold, Italic
Formatted: Font: Not Bold
Deleted: .
Deleted: "
Deleted: "
Deleted: "
Deleted: "

9.1.2 Boxplots for a Series of Dependent Variables (Paired-Samples T-Test)

To look at the data from the French and O'Brien (2008) study, we will want to look at four variables—scores on the ENWR and ANWR at Time 1 and Time 2. If you are following along with me, import the SPSS file French&O'BrienGrammar.sav and call it french. To look at a series of boxplots of dependent variables side by side in R we will just use the boxplot command from the previous section and add more variables (so our only choice for this is the R Console, not R Commander).

```
boxplot(ANWR_1,ANWR_2,ENWR_1,ENWR_2, ylab="Score on test out of 40",
names=c("Arabic Time 1", "Arabic Time 2", "English Time 1", "English Time 2"),
las=1,notch=TRUE,col="grey", boxwex=.5, ylim=range(c(1,40)),medcol="white")
```

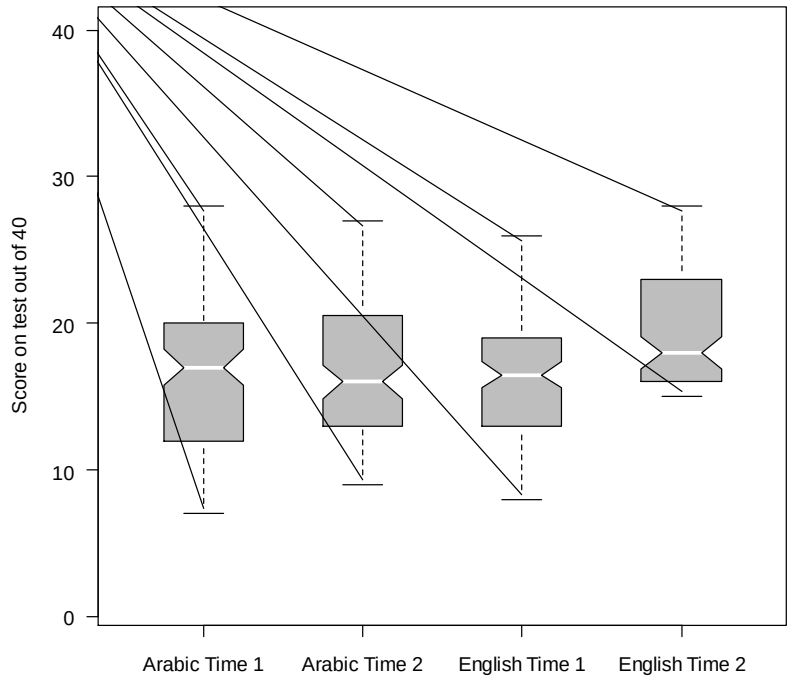

Figure 9.3 A boxplot of French and O'Brien's (2008) phonological memory measures.

The resulting boxplot in Figure 9.3 shows the distribution of all four variables that measure phonological memory in this experiment. Notice that all of the distributions except for the English nonwords at Time 2 look fairly normally distributed. There are no outliers identified. The medians for the Arabic nonword test are quite similar at Time 1 and Time 2, although the range of the distribution at Time 2 is slightly smaller than at Time 1. For the English nonword test the median at Time 2 is definitely higher than at Time 1, and the range of distribution at Time 2 is also smaller.

There are a large number of parameters that you can experiment with to make your boxplots look really nice. Below is my analysis of the command I used for the boxplots in Figure 9.3.

```
attach(french)
boxplot(ANWR_1,ANWR_2,ENWR_1,ENWR_2, ylab="Score on test out of 40",
names=c("Arabic Time 1", "Arabic Time 2", "English Time 1", "English Time 2"),
las=1,notch=TRUE,col="grey", boxwex=.5, ylim=range(c(1,40)),medcol="white")
```

attach(french)	Attaching this means I don't have to type attach$ANWR_1 before every variable (just remember to detach!).

boxplot()	The command to call for a boxplot.
ANWR_1,ANWR_2, etc.	List as many variables as you want here, separated by commas.
ylab="Score on test out of 40"	Gives custom name to the y-axis label.
names=c("Arabic Time 1", etc.)	Gives x-axis labels to the variables; otherwise they appear only as numbers.
las=1	Orients the variable labels horizontal to the axis (las=2 would orient them perpendicular to the axis).
notch=TRUE	Makes notched boxplots, where the 95% confidence interval is indicated by the notches.
col="grey"	Fills the boxplots with a color.
boxwex=.5	Boxwex is a scale factor for all boxes. It makes the boxes thinner than normal.
ylim=range(c(1,40))	Specifies that y-axis limits should range from 1 to 40.
medcol="white"	Makes the median line the indicated color.

Tip: If you can't remember the names of your variables, you can go back to R Commander and pull up the Data Editor with the Edit data set button. Alternatively, type the command names(french) in R Console.

The following table gives additional commands that could be useful for customizing any boxplot that you make (for even more, type help(boxplot) or see Paul Murrell's 2006 book *R graphics*).

varwidth=TRUE	Draws boxplots with width proportional to number of observations in group.
outline=FALSE	Won't draw in outliers if set to FALSE.
names=FALSE	Won't print labels under each boxplot if set to F.
horizontal=TRUE	Draws boxplots horizontally if set to T.
par(mfrow=c(1,2))	Sets the parameter so that two graphs can be displayed side by side in the Graphics Device; the first number in c() gives the number of rows, and the second number gives the number of columns.

Creating a Boxplot for Several Variables Side by Side

The basic R code for this type of boxplot is:

boxplot(ANWR_1,ANWR_2,ENWR_1,ENWR_2)

I have also included commands in this section for many ways to customize boxplots.

Formatted: Font: Not Bold, Italic
Formatted: Font: Not Bold, Italic
Formatted: Font: Not Bold, Italic
Formatted: Font: Not Bold, Italic
Formatted: Font: Not Bold
Formatted: Font: (Default) Arial

9.1.3 Boxplots of a Series of Dependent Variables Split into Groups

There's one more boxplot that might be useful to look at. This is the case where we have more than one dependent variable, but the scores are split into groups. I'll use the Leow and Morgan-Short (2004) data on the productive pre-test and post-test measures (leow). The data will be split by the think-aloud and non-think-aloud groups.

The best way to get such a boxplot in R is to set the par(mfrow) command to include the number of plots you want, set side by side. I will examine only two variables.

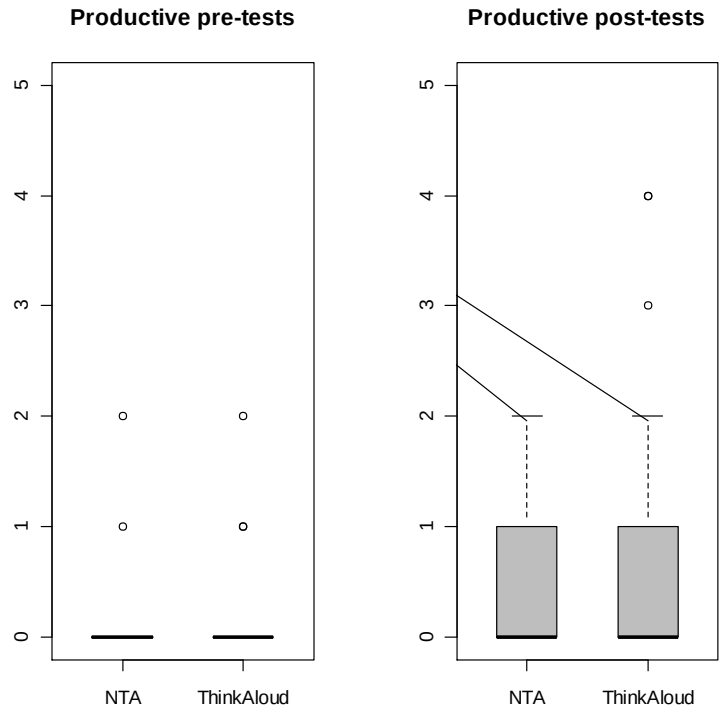

Figure 9.4 Leow and Morgan-Short's (2004) variables split by groups using multiple boxplots.

Figure 9.4 shows that in the pre-test, where participants had to produce the imperative form, basically everyone in both groups scored zero. The boxplot labels all those who scored above zero as outliers. In the post-test, both groups have the same distribution, although there are more outliers in the think-aloud group than the non-think-aloud one (at least in this portion of the data; it actually goes to 15, but I have limited it, since the distribution is so close to zero in both cases). Still, in the productive post-test, at least half of the participants scored zero, as evidenced by the median line on zero.

The code for Figure 9.4 is:

```
par(mfrow=c(1,2))
```

```
levels(leow$Group)=c("NTA", "ThinkAloud") #Make labels shorter so they'll print
boxplot(ProPreScore~Group,data=leow,ylim=range(c(0,5)),col="gray",
main="Productive pre-tests",boxwex=.5)
boxplot(ProPostScore~Group,data=leow,ylim=range(c(0,5)),col="gray",
main="Productive post-tests",boxwex=.5)
```

I won't analyze this command because it is not substantially different from the boxplot() command I analyzed in the previous section. However, I will point out that, to split by groups, you model the dependent variable (ProPreScore) by the independent variable (Group) by using the tilde syntax (ProPreScore~Group). Also notice that it is especially important to use the ylim argument here so graphs which are being compared are using the same scale.

Creating a Boxplot in R for Multiple Variables Split into Groups

The way to do this in R is to concatenate a series of boxplots split by groups. The way to tell R to put boxplots together is to use the following command:

```
par(mfrow=c(1,2)) #the first entry tells # of rows, second tells # of columns
```

Now put in as many split-group boxplots as you planned for:

```
boxplot(ProPreScore~Group,data=leow,ylim=range(c(0,5)))
boxplot(ProPostScore~Group,data=leow,ylim=range(c(0,5)))
```

Formatted: Font: Not Bold, Italic

Formatted: Font: Not Bold, Italic

Formatted: Font: Not Bold, Italic

Formatted: Font: Not Bold, Italic

Note: Here's the code I used for the boxplots in Figure 9.6 of the SPSS book (*A Guide to Doing Statistics in Second Language Research Using SPSS*, p. 255):

```
boxplot(RecPostScore~Group,data=leow,ylim=range(c(0,17)),col="gray",
main="Receptive Task",boxwex=.5)
boxplot(ProPostScore~Group,data=leow,ylim=range(c(0,17)),col="gray",
main="Productive Task",boxwex=.5)
```

9.2 Application Activities with Creating Boxplots

1. Use Leow and Morgan-Short's (2004) data (import LeowMorganShort.sav and call it leow). Create two new variables (gainscores) by subtracting the pre-test score from the post-test score in both the receptive and productive conditions (the receptive conditions are preceded by the prefix "rec-" and the productive by the prefix "pro-"). Plot these two new variables (I called them "recfinal" and "profinal"; do *not* divide the boxplots by experimental group right now). Did the participants seem to perform differently in the receptive versus productive condition? Would you say these distributions have the same size of box (the interquartile range)? Are there any outliers? Do these groups have normal distribution?

2. Using the same data as in activity 1, make a boxplot of the variable you calculated in activity 1 (the gain score in the receptive condition, which I called "recfinal") but this time divide this boxplot into the think-aloud and non-think-aloud groups. Are there any outliers? Which group improved the most on the receptive measure? Which group has more spread? What do you think it means if one boxplot is larger (has more spread) than another?

3. Use the Yates2003.sav data set (call it yates). This is data from an MA thesis by Yates (2003) which examined whether pronunciation practice that emphasized suprasegmentals (by

having participants mimic the actors in *Seinfeld*) was more effective than laboratory segmental practice in improving English learners' accent. Create a series of boxplots of the four variables. Did the lab group seem to improve over the semester? What about the mimicry group? Are there any outliers? Which group has more spread?

4. Use the Inagaki and Long (1999) t-test data (InagakiLong1999.Ttest.sav, import as inagaki). The authors tested the hypothesis that learners of Japanese who heard recasts of target L2 structures would have a greater ability to produce those structures than learners who heard models of the structures. These data were for adjectives, and the authors compared the differences between the recast and model groups. Plot a boxplot of the gain score divided by groups. Did the groups seem to perform differently? Would you say these distributions have the same size of box (the interquartile range)? Are there any outliers? Do these groups have normal distribution?

9.3 Performing an Independent-Samples T-Test

Perform an independent-samples t-test in R Commander by clicking on STATISTICS > MEANS > INDEPENDENT SAMPLES T-TEST (see Figure 9.5).

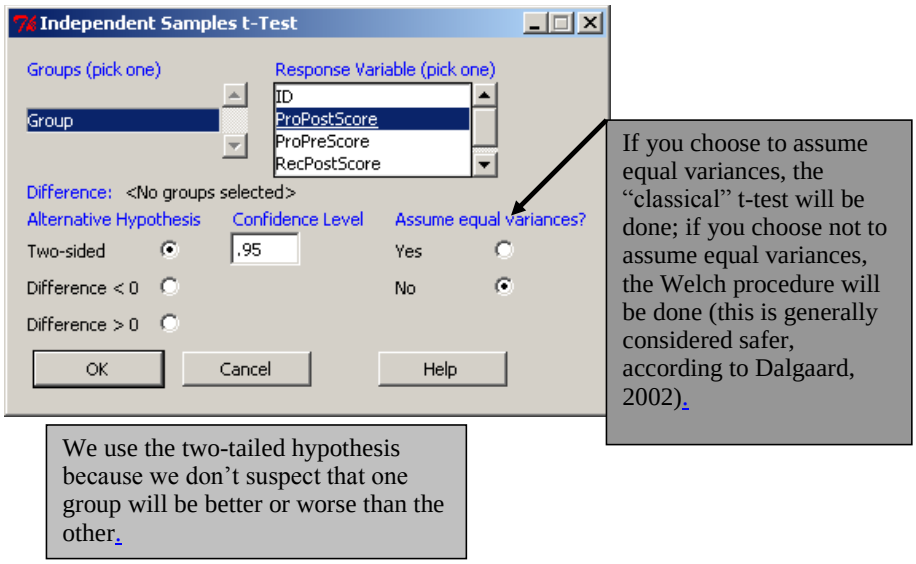

Figure 9.5 Performing an independent-samples t-test in R Commander.

The R code for the independent-samples t-test is:

```
t.test(ProPostScore~Group, alternative='two.sided', conf.level=.95,
  var.equal=FALSE, data=leow)
```

Here is the output:

```
        Welch Two Sample t-test

data:  ProPostScore by Group
t = -0.2513, df = 73.747, p-value = 0.8022
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.680795  1.304277
sample estimates:
       mean in group NTA mean in group ThinkAloud
                1.153846                 1.342105
```

We see from the last line that the means for the post-test production task in the two groups are different numbers (think-aloud group=1.34, non-think-aloud group=1.15), but our question is whether they are different enough to say they come from two different populations (the actual difference in mean scores between our groups is .19). To answer this question, look at the 95% confidence interval. The interval in which we expect, with 95% confidence, to find the true difference between the means is quite wide, from −1.68 to 1.30. This confidence interval covers zero, which means we cannot reject the null hypothesis. The CI is also much wider than the actual mean difference (only .19), indicating that we are not very confident about the actual difference between groups. However, with repeated sampling we would expect to find the mean difference in our CI range. When you report the results of a t-test, you'll also want to report the t-statistic (notice it is very small, much smaller than 2) and the degrees of freedom (df). Notice that the df is not an integer—this happens when the Welch procedure is used. If you report the CI you don't actually need to report the *p*-value, but you can.

The R code for this test is:

```
t.test(ProPostScore~Group, alternative='two.sided', conf.level=.95,
var.equal=FALSE, data=leow)
```

t.test (x, . . .)	Calls a t-test.
ProPostScore~Group	Syntax for modeling the dependent variable by the independent variable (Group).
alternative="two.sided"	This default calls for a two-sided hypothesis test; other alternatives: "less", "greater".
conf.level=.95	Sets the confidence level for the mean difference.
var.equal=FALSE	Calls for the Welch procedure, which does not assume equal variances.
data=leow	Specifies the data set.

Tip: The only variables that will appear in the Groups list for the independent-samples t-test in R Commander are those that have *only* two dimensions, as the Leow and Morgan-Short data set has. Thus, if you have three groups that you would like to compare using three different t-tests, you should subset your original data set to contain just two groups at a time. For example, with my data set (SPSS file called LarsonHall.Forgotten.sav, imported as forget) that had three groups, here is a command to subset that will exclude the group "Non":

```
forgetNoNon <- subset(forget, subset=Status!= "Non")
#note that the "!=" symbol means "does not equal"
```

Performing an Independent-Samples T-Test

On the R Commander drop-down menu, choose STATISTICS > MEANS > INDEPENDENT SAMPLES T-TEST. Pick your group variable (the independent variable) and the "Response Variable" (the dependent variable). Unless otherwise strongly compelled, leave the "Assume equal variances" button at its default of "No."

Basic R code for this command is:

```
t.test(ProPostScore~Group, var.equal=FALSE, data=leow)
```

9.4 Performing a Robust Independent-Samples T-Test

We have seen repeatedly that real data sets rarely exactly fulfill the requirement of having a normal distribution. We can perform a robust t-test on our data and not have to worry that the data is not normal or that there may be outliers in the data. Wilcox (2003) states that using 20% trimmed means and the non-parametric (percentile) bootstrap procedure is the best overall approach for a robust test of data. You could adjust the amount of means trimming to be larger or smaller if you had reason to, but for general purposes stick to 20%. Wilcox has an R library, but at the moment it cannot be retrieved from the drop-down list method in R Console. Instead, type this line directly into R:

```
install.packages("WRS",repos="http://R-Forge.R-project.org")
```

If you have any trouble installing this, go to the website http://r-forge.r-project.org/R/?group_id=468 for more information.

Once the library is downloaded, open it:

```
library(WRS)
```

The command we'll want to use is trimpb2(). This command calls for the data to be subsetted so that the vector contains only the data for one group. Here is how I subsetted for the Leow and Morgan-Short productive post-test data (this could also be done in R Commander by going to DATA > ACTIVE DATA SET > SUBSET ACTIVE DATA SET):

```
ProPostScoreNTA <- subset(leow, subset=Group=="NTA", select=c(ProPostScore))
#n=30
```


Deleted: Independent Sampl
Deleted: *only*
Deleted: 3
Deleted: "
Deleted: "
Deleted: "
Formatted: Font: Not Bold, Italic
Formatted: Font: Not Bold, Italic
Formatted: Font: Not Bold

```
ProPostScoreTA <- subset(leow, subset=Group=="ThinkAloud",
select=c(ProPostScore)) #n=37
```

Now I'm ready to run the robust command:

```
trimpb2 (ProPostScoreNTA, ProPostScoreTA, tr=.2, alpha=.05, nboot=2000, win=F,
plotit=T)
```

Note that R may require a little bit of time to run this calculation. The output looks like this:

```
$p.value
[1] 0.743

$ci
[1] -0.59  0.36

$est.dif
[1] -0.05
```

The first thing we will look at is the confidence interval for the difference in means between the groups, which we see runs through zero [−.59, .36]. The interval is very wide and runs through zero, so we conclude there is no difference between groups. The $est.dif value is the difference between the two groups using the 20% trimmed means, not the normally calculated means. You can calculate this yourself:

```
mean(ProPostScoreNTA,tr=.2) #untrimmed mean is 1.15
 0.2
mean(ProPostScoreTA,tr=.2) #untrimmed mean is 1.34
 0.25
```

Therefore, the estimated 20% trimmed mean difference between groups is .05!

Notice that the CI for the bootstrapped test is different from the one for the parametric test (which was [−1.68, 1.30]). It is no wonder the two are different, since the original data were not normally distributed at all. The robust CI will more likely approximate the actual 20% trimmed mean differences we would find with future testing. The *p*-value tests the null hypothesis that the difference between groups is zero.

Here is an analysis of Wilcox's robust command:

trimpb2 (ProPostScoreNTA, ProPostScoreTA, tr=.2, alpha=.05, nboot=2000, win=F)

trimpb2(x, y, . . .)	Command stands for "trimmed percentile bootstrap."
x, y	The data sets used in the trimpb() command are found in the variables x and y; these must be single vectors of data containing the data of only one group.
tr=.2	Specifies 20% means trimming; if you have many outliers you might want to set this higher.
alpha=.05	Default alpha level.
nboot=2000	Specifies the number of times to bootstrap; Wilcox (2003, p. 224) says that 500 samples probably suffice, but says there are arguments for using 2,000.
WIN=F	If set to TRUE, this will Winsorize the data, an alternative to

	means trimming which I will not use (see Wilcox, 2003 for more information)
win=.1	If Winsorizing, specifies the amount.

I want to note that the way you order your two variables does matter slightly. It doesn't matter in the sense that the outcome will change, but it does matter in the sense that your CI will change depending on which variable is being subtracted from the other. So in a case where you have a CI that uses negative numbers given one ordering of the variables (say, for example, it is [−10.75, −1.28] in one case), it will use positive numbers given the opposite ordering of the variables (now the CI is [1.36, 10.57] in the opposite order). This of course is of no consequence, as the difference between the groups is the same.

Performing a Robust Independent-Samples T-Test in R

First, the Wilcox commands must be loaded or sourced into R (see Appendix C).

library(WRS)

The basic R code for this command is:

trimpb2 (ProPostScoreNTA, ProPostScoreTA, tr=.2, alpha=.05, nboot=2000, win=F)
where the variables are subsetted from the leow$ProPostScore variable

Formatted: Font: Not Bold, Italic
Deleted:
Formatted: Font: Not Bold, Italic
Formatted: Font: Not Bold

In reporting the results of a robust test, you will want to mention what robust method was used, the N, the mean for each group, and a confidence interval for the mean difference between the groups. Here is an example:

Using 20% trimmed means and a percentile bootstrapping method, an independent-samples t-test found no evidence of a difference between scores on the productive post-test task for the think-aloud group (20% trimmed mean=.25, N=37) and the non-think-aloud group (20% trimmed mean=.2, N=30). The 95% CI for the difference in means was [−.59,.36] (p=.74). The effect size for this comparison, calculated on the entire data set, was Cohen's d=−.06, a negligible effect size.

9.5 Application Activities for the Independent-Samples T-Test

1. Larson-Hall (2008) examined 200 Japanese college learners of English. They were divided into two groups, one called early learners (people who started studying English before junior high school) and the other later learners (people who started studying English only in junior high). Import the Larsonhall2008.sav file, name it larsonhall2008, and see whether the groups (divided by the variable erlyexp) were different in their language learning aptitude (aptscore), use of English in everyday life (useeng), scores on a grammaticality judgment test (GJT), and scores on a phonemic listening test (rlwscore). Be sure to explore whether the data are normally distributed and have equal variances by looking at boxplots. Report results for all variables regardless of meeting assumptions: 95% CIs, means, standard deviations, Ns, hypothesis testing results (t- and p-values), and effect sizes. Discuss what the numbers mean.

2. Use the Inagaki and Long (1999) t-test data (import InagakiLong1999.Ttest.sav as inagaki). Test the hypothesis that learners of Japanese who heard recasts of adjectives would

have a greater ability to produce this structure than learners who heard models of the structure. Be sure to explore whether the data are normally distributed and have equal variances by looking at boxplots. Report results regardless of meeting assumptions: 95% CIs, means, standard deviations, Ns, hypothesis testing results (*t*- and *p*-values), and effect sizes. Discuss what the numbers mean.

3. Practice doing a t-test with a data set that contains more than two groups at a time. Use the LarsonHall.Forgotten.sav file and import it as forget. There are three groups in this data, called "Non," "Early," and "Late" (I examined Japanese users of English who had never lived abroad, "Non," lived abroad as children, "Early," or lived abroad as adults, "Late"). First subset the data set into three smaller data sets that contain only two groups at a time. Then perform independent-samples t-tests on the variable of SentenceAccent to see if the groups are different in how well they pronounced entire sentences. Use the Status variable to divide the groups. Report effect sizes for the tests as well.

4. In activity 1 you examined several comparisons. The phonemic test gjtscore did not show a statistical difference between groups. Run a robust t-test with the two groups (erlyexp) and compare the results of the robust test to the parametric t-test.

5. Open the made-up data set called vocabulary (it is a .csv file). This file shows gain scores from each group on a vocabulary test done after one group experienced an experimental treatment to help them remember vocabulary better, and the other group did not. (Note that the data are not in the correct format needed to perform an independent-sample t-test; you must change from a wide format to a long format; see the online document "Factorial ANOVA.Putting data in correct format for factorial ANOVA" for help with this.) First perform a normal parametric t-test on the data. Then perform a robust t-test, first with 20% means trimming, and then 10% means trimming. What are the results?

9.6 Performing a Paired-Samples T-Test

To perform the paired-samples t-test in R, use R Commander and choose STATISTICS > MEANS >PAIRED T-TEST (see Figure 9.6).

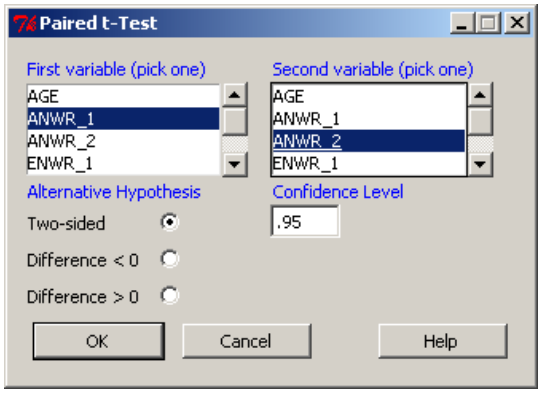

Figure 9.6 Performing a paired-samples t-test in R Commander.

The R code for the paired-samples t-test is:

```
t.test(french$ANWR_1, french$ANWR_2, alternative='two.sided',
```

```
conf.level=.95, paired=TRUE)
```

The only difference between this command and that for the independent-samples t-test explained in the online document "T-tests: The independent samples t-test" is the addition of the argument paired=TRUE. This argument simply specifies that a paired-samples test should be done.

The output looks like this:

```
        Paired t-test

data:  french$ANWR_1 and french$ANWR_2
t = -1.8319, df = 103, p-value = 0.06985
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.52065542  0.02065542
sample estimates:
mean of the differences
                  -0.25
```

Looking at the confidence intervals, we see that the 95% CI for the mean difference between the ANWR at Time 1 and Time 2 is [−.52, .02]. This just barely goes through zero, but means that the difference between mean scores could be as large as .52 or as small as zero (or go .02 in the other direction) with 95% confidence. For a 40-point test this is not a wide CI, but it is quite close to zero. If we just considered the *p*-value of p=0.07, we might argue that we could reject the null hypothesis that there was no difference between measurements, but the CI argues for the fact that the effect size will be quite small. In fact, the calculated effect size (for more information on how to do this, see the SPSS book, *A Guide to Doing Statistics in Second Language Research Using SPSS*, Section 9.5.2, p. 263) for this difference is d=−0.05, which is a negligible effect size.

On the other hand, a paired-samples t-test exploring the differences between mean scores for Time 1 and Time 2 for the ENWR reveals something different.

```
        Paired t-test

data:  french$ENWR_1 and french$ENWR_2
t = -14.2922, df = 103, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.536743 -2.674796
sample estimates:
mean of the differences
              -3.105769
```

The CI is −3.53, −2.68, meaning that the differences between groups might be as large as three and a half points or as small as about two and a half points, with 95% confidence. This is much further from zero than the difference between Time 1 and Time 2 on the ANWR, and we can conclude that the participants statistically improved in their performance on the ENWR over time (additionally, the effect size here is quite large, with d=.8). French and O'Brien concluded that scores on the ENWR improved because the participants were becoming more proficient at English, but the fact that scores did not improve statistically on the ANWR shows it is a language-independent measure of phonological memory (for people who do not know Arabic!).

Tip: If you use a directional one-tailed hypothesis for a t-test, you will have more power to find differences. If you use a one-tailed test, however, the CI will not specify an upper range. This makes sense if you think it through, because if you are testing the hypothesis that one group will be better than another then you are not testing for how much *greater* the difference could be; you simply want to test if it could range into the *lesser* range. Note that the *p*-value that will be given is already adjusted for a **one-tailed** (or **directional**) **hypothesis**.

Performing a Paired-Samples T-Test in R

On the R Commander drop-down menu, choose STATISTICS > MEANS > PAIRED T-TEST. Pick your two matching variables. The order will not be important unless you have a one-tailed hypothesis.

Basic R code for this command is:

```
t.test(french$ANWR_1, french$ANWR_2, alternative='two.sided',
conf.level=.95, paired=TRUE)
```

Deleted: ,
Deleted: ,
Formatted: Font: Italic
Formatted: Font: Italic
Formatted: Font: Italic
Deleted: '
Deleted: '

9.7 Performing a Robust Paired-Samples T-Test

We have seen repeatedly that real data sets rarely exactly fulfill the requirement of having a normal distribution. We can perform a robust t-test on our data and not have to worry that the data is not normal or that there may be outliers in the data. Wilcox (2003) states that using 20% trimmed means and the non-parametric (percentile) bootstrap procedure is the best overall approach for a robust test of data. You could adjust the amount of means trimming to be larger or smaller if you had reason to, but for general purposes stick to 20%. Wilcox has an R library, but at the moment it cannot be retrieved from the drop-down list method in R Console. Instead, type this line directly into R:

```
install.packages("WRS",repos="http://R-Forge.R-project.org")
```

If you have any trouble installing this, go to the website http://r-forge.r-project.org/R/?group_id=468 for more information.

Once the library is downloaded, open it:

```
library(WRS)
```

The command we'll want to use is rmmcppb(). Wilcox (2003) notes that, when using paired groups, there are two hypotheses that could be tested—that the difference between the groups is zero, or that the group means are equivalent. For the first choice, the hypothesis is:

H_0: $\mu_D = 0$

In this case, bootstrapping will be done by resampling with replacement the difference values for each pair. For the second choice, the hypothesis is:

H_0: $\mu_1 = \mu_2$

When we want to test this hypothesis, pairs of observations will be resampled. The difference in these two hypotheses is not merely academic. At times, it may affect the outcome of whether the null hypothesis is rejected or not. It should be noted that the paired t-test used in SPSS tests the hypothesis that the difference score is zero (Wilcox, 2003, p. 363), and this is true for R as well (note that the paired t-test output in R explicitly states the alternative hypothesis: "true difference in means is not equal to 0").

If you are following along with me, I will use the French and O'Brien (2008) data (import it and name it french). To test both kinds of hypotheses robustly, use the following Wilcox command in the R Console (specify which kind of hypothesis you want in the dif=T argument) (for more detailed information about these commands, see Wilcox, 2003).

rmmcppb(french$ANWR_1, french$ANWR_2, alpha=.05, est=mean, tr=.2, dif=T, nboot=2000, BA=T, hoch=F)

rmmcppb (x, y=NA, . . .)	Computes a percentile bootstrap to compare paired data.
french$ANWR_1 french$ANWR_2	These are vectors of data from the french data set.
alpha=.05	The alpha level.
est=mean	By default, est=mest (M-estimator), but it won't work with this data; can use median as well.
tr=.2	If you use the mean, also use 20% means trimming.
plotit=T	This will return a plot either of the bootstrapped difference scores (if dif=T) or pairs of bootstrap values, with a polygon around the 95% confidence region for the parameters.
dif=T	Here is where we can decide which hypothesis to test; dif=T sets the command to test the hypothesis of no difference (H_0: $\theta_D = 0$); if F, tests hypothesis of equal measures of location (H_0: $\theta_1 = \theta_2$).
nboot=2000	Number of bootstrap samples to use.
BA=F	When using dif=F, BA=T uses a correction term that is recommended when using MOM.
hoch=F	If dif=F, it is recommended to set this to TRUE.
Additional arguments:	
seed=F	If true, you can set the seed of the random number generator so results of bootstrap can be duplicated.
seed(2)	If you use seed, have to specify the seed with the number in parentheses.

This test will return a confidence interval and a *p*-value. The rest of the data can be ignored for now.

```
[1] "dif=T, so analysis is done on difference scores"
$output
     con.num  psihat p.value p.crit ci.lower  ci.upper
[1,]       1 -0.4375   0.005   0.05 -0.71875 -0.171875

$con
     [,1]
[1,]    1
[2,]   -1

$num.sig
[1] 1
```

The read-out reminds us which hypothesis we tested, in this case that the mean difference between the scores was zero (H_0: $\mu_D = 0$), the same type of hypothesis that the parametric test is using. The estimate of the difference between the two groups is .44 (from the psihat value), with a 95% confidence interval for the 20% trimmed mean difference of [−.7, −.17]. This confidence interval does not contain zero, so we can reject the null hypothesis and conclude that there was a difference in outcomes. Now the parametric t-test told us there was no difference between groups, and I think the authors were happy with that outcome, because they showed that the students improved on the test of phonological memory that involved the language they were learning (English) and stayed the same on the test of memory that involved a language they didn't know (Arabic). You might think then that they wouldn't want to use a robust test and report a result that showed a difference. Perhaps not, but I maintain that what is important is the effect size, and that hasn't changed. The effect size is still very small, so we see that there is a statistical difference between groups, but the largest amount that we predict that the scores will differ from Time 1 to Time 2 is only 0.7 points out of 40. There is a slight increase over time, but the increase is still very small! What I think this exercise should show you is that *p*-values are not important. They will change with the size of the sample, but the effect size is what's important, and also confidence intervals give you much more information than *p*-values do!

When you use the Wilcox's rmmcppb() command, a plot of the bootstrapped mean differences is also returned. I'm not sure what you'd use it for but it's fun to look at to remember what the bootstrapping process is doing!

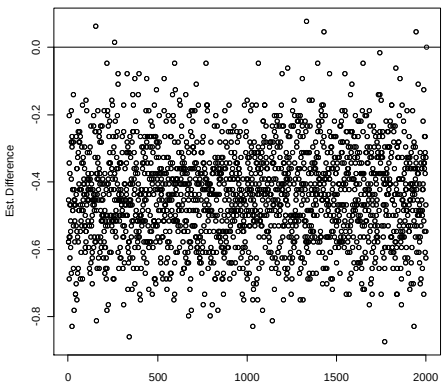

Figure 9.7 Estimated differences in mean scores between paired samples after a 20% trimmed means bootstrap.

Figure 9.7 shows this plot of the 2,000 estimated mean differences, and you can see that most of them are not even close to zero (95% of them will lie in our 95% CI zone—that's how it's calculated!).

Performing a Robust Paired-Samples T-Test

First, the Wilcox library must be loaded into R and opened:

```
library(WRS)
```

If you want to test the hypothesis that the difference in means is zero (H_0: $\mu_D = 0$), the basic code is:

```
rmmcppb(french$ANWR_1, french$ANWR_2, est=mean, tr=.2, dif=T, plotit=T)
```

If you want to test the hypothesis that the group means are equivalent (H_0: $\mu_1 = \mu_2$), the basic code is the same but the argument dif=T should be changed to dif=F (and you should add the following arguments: BA=T, hoch=T).

In reporting the results of a robust test, you will want to mention what robust method was used, the N, the mean for each group, and a confidence interval for the mean difference between the groups. For an example see the final paragraph of the online document "T-tests: A robust independent-samples t-test."

9.8 Application Activities for the Paired-Samples T-Test

1. You saw an example analysis of the French and O'Brien (2008) data in the online document "T-tests: The paired samples t-test." French and O'Brien (2008) also performed paired-samples t-tests to see whether the participants in the study improved on receptive vocabulary (RVOCAB), productive vocabulary (PVOCAB), and grammar measures (GRAM) (note that all of these will have a "1" or a "2" appended to them to show whether they were a pre-test or a post-test). Maximum points were 60 on both vocabulary measures and 45 on the grammar measure. Perform three t-tests to investigate whether the schoolchildren made progress on these measures over the course of their English immersion. First comment on the distribution of the data by using boxplots; then report on the t-tests no matter whether distributions are normal or not. Use the French and O'Brien grammar.sav file, imported as french. Be sure to report on effect sizes as well.

2. Yates (2003). You examined boxplots from this data if you performed the application activities in the online document "T-tests: Application activity_Creating boxplots" (import the Yates.sav file as yates now if you have not already done so). Compare the accent scores of the lab group before and after training, and also the mimicry group before and after training. Be sure to look at effect sizes. How might you explain the results?

3. The Larson-Hall and Connell (2005) data (import LarsonHall.Forgotten.sav as forget) can be analyzed with a paired-samples t-test. A paired-samples t-test will answer the question of whether the participants performed the same way with their accent on words beginning with /r/ (AccentR) and /l/ (AccentL). Be sure to look at effect sizes. What are the results?

Deleted:
Formatted: Font: Not Bold, Italic
Formatted: Font: Not Bold, Italic
Formatted: Font: Not Bold
Deleted: H_o
Formatted: Font: Italic
Deleted:
Deleted: H_o
Formatted: Font: Italic
Deleted:
Formatted: Font: Italic
Deleted:
Formatted: Font: (Default) Arial

4. In the online document "T-tests: The paired samples t-test" I performed a robust t-test on French and O'Brien's (2008) variable ANWR, using the hypothesis that the mean difference between the scores was zero (H_0: = 0). Run this robust t-test again and vary some of the parameters. For example, try the same hypothesis but cut means trimming down to 10%. Try running the robust test with the hypothesis that the group means are equal (H_0: θ1 = θ2), and try that with different levels of means trimming. Do you find the groups to be statistically different or not with these other tests? How can a researcher deal with the fact that different tests produce different outcomes?

5. Activity 2 looked at results from Yates. Yates was hoping to find that mimicry improved accent ratings more than traditional lab work (scores are from five-point Likert scales for comprehensibility and accent, apparently added together from different judges instead of averaged). Perform robust paired-samples t-tests with mimicry (which had a higher effect size than lab) and see if there are any test configurations which allow you to find a statistical difference between groups (reference ideas for test configurations in activity 4).

9.9 Performing a One-Sample T-Test

We will examine the question of whether ESL learners preferred NESTs or non-NESTs in the areas of culture and speaking in this example. I use the Torres.sav file, imported as torres. For the one-sample t-test, in R Commander choose STATISTICS > MEANS > SINGLE-SAMPLE T-TEST (see Figure 9.8).

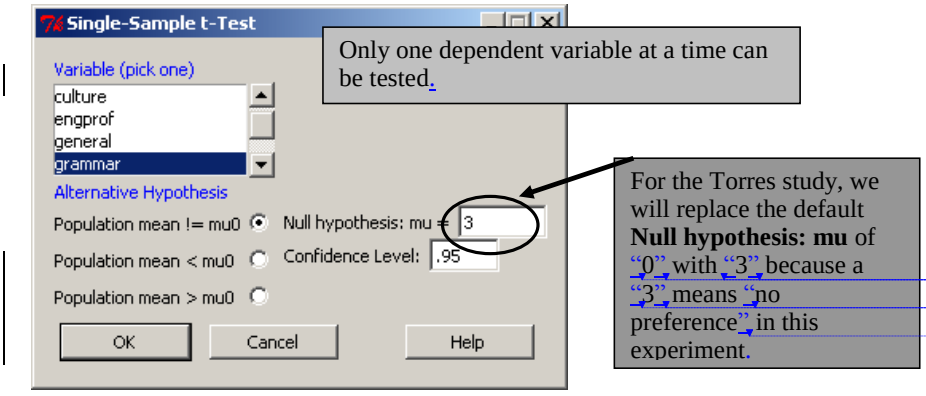

Figure 9.8 Performing a one-sample t-test in R.

The output of the one-sample t-test looks like this:

```
        One Sample t-test

data:  torres$grammar
t = 1.7724, df = 101, p-value = 0.07934
alternative hypothesis: true mean is not equal to 3
95 percent confidence interval:
 2.976625 3.415532
sample estimates:
mean of x
 3.196078
```

On the first line you find the variable that you tested for, which in this case was Grammar. It is important to look at the mean score for this, which is the last line of the output. The mean of the grammar variable is 3.20, which means there is a slight preference above the neutral value for NESTs. It is important to look first at your data to make sure you have a feel for it. Look also at the df on the second line, which tells you how many people were in your test, plus one (because there is one degree of freedom subtracted from the N for the one-sample t-test). Make sure to get a feel for your data before looking at the results of the statistical test.

The results of the t-test can be found in the second line of the output. Here you will find the t-test value, degrees of freedom, and the *p*-value of the null hypothesis. The *p*-value is not less than .05, so we cannot reject the null hypothesis in this case. Notice that, in this case, the 95% confidence interval does not go through 0; instead, it goes through 3, which we have set up to be our neutral number. So the fact that the mean difference for this sample goes through 3 means we may find no difference between our group and the neutral score we are testing it against (and this means we cannot reject the null hypothesis that the true mean is 3).

Tip: If you use a one-tailed directional hypothesis, you do not have to adjust the *p*-value. The one that is returned has already been adjusted for a one-way hypothesis. Remember that using a one-tailed hypothesis will give you more power to find differences.

Formatted: Font: Italic

The R code for this test is:

```
t.test(torres$grammar, alternative='two.sided', mu=3, conf.level=.95)
```

t.test (x, . . .)	Gives the command for all t-tests, not just the one-sample test.
torres$grammar	This is the grammar variable in the torres data set.
alternative="two.sided"	This default calls for a two-sided hypothesis test; other alternatives: "less", "greater"
mu=3	Tells R that you want a one-sample test; it compares your measured mean score to an externally measured mean.
conf.level=.95	Sets the confidence level for the mean difference.

Effect sizes can be determined quite simply; just take the mean of x listed in the output, subtract your neutral value from it (for the Torres grammar variable this was 3, so 3−3.19=.19) and divide by the standard deviation of the group you have, just as you would do if the variances were not equal (see the SPSS book, *A Guide to Doing Statistics in Second Language Research Using SPSS*, Section 9.4.2, pp. 258–259 if you do not remember this). The effect size for grammar is thus 0.19/1.12=.17, a very small effect size.

Performing a One-Sample T-Test

Formatted: Font: Not Bold, Italic

Formatted: Font: Not Bold, Italic

Formatted: Font: Not Bold

On the R Commander drop-down menu, choose STATISTICS > MEANS > SINGLE-SAMPLE T-TEST.

Basic R code for this command is:

```
t.test(torres$grammar, mu=3)
```

9.10 Performing a Robust One-Sample T-Test

I assume here that you have already looked at the robust t-tests for independent-samples t-tests and paired-samples t-tests. To perform a 20% trimmed mean percentile bootstrap for a one-sample t-test, use Wilcox's command trimpb() (this is very similar to the robust command for independent-samples t-tests, which was trimpb2()). Basically, the only difference in syntax between the two tests is that we add an argument specifying the neutral value: null.value=3. Thus, this test contains means trimming (default is 20%) and uses a non-parametric percentile bootstrap. Be sure to open the WRS library before running this command.

```
trimpb(torres$grammar, tr=.2, alpha=.05, nboot=2000,WIN=F,null.value=3)
```

Here is the output from this call:

```
[1] "The p-value returned by the this function is based on the"
[1] "null value specified by the argument null.value, which defaults to 0"
[1] "Taking bootstrap samples. Please wait."
$ci
[1] 2.967742 3.483871

$p.value
[1] 0.102
```

The output shows the 95% confidence interval for the population mean of the grammar variable that lies over the neutral point of 3, [2.97, 3.48]. Since the CI contains the neutral value of 3, we cannot reject the null hypothesis that the true population mean is equal to 3. The *p*-value also formally tests the null hypothesis that the actual mean is 3.

We would probably like to have some measure of the central location here, so we can calculate the trimmed mean of the original sample (before it was bootstrapped) like this:

```
mean(torres$PRON,tr=.2)
[1] 4.596774
```

Performing a Robust One-Sample T-Test

First, the Wilcox commands must be loaded or sourced into R (see online document "Using Wilcox's R Library").

The basic R code for this command is:

```
trimpb(torres$PRON, tr=.2, null.value=3)
```

Formatted: Font: Not Bold, Italic

Formatted: Font: Not Bold, Italic

Formatted: Font: Not Bold

9.11 Application Activities for the One-Sample T-Test

1. Torres (2004) data. Use the data set Torres.sav, imported as torres. Calculate one-sample t-tests for the variables of listenin and reading using a one-sample parametric test. Comment on the size of the effect sizes.

2. Using the same data set as in activity 1, look at the variables of culture and pronunciation using both parametric one-sample tests and robust one-sample tests. Do you find any differences? What are the effect sizes?

3. Dewaele and Pavlenko Bilingual Emotions Questionnaire (2001–2003) data. Use the BEQ.sav data set, imported as beq. Test the hypothesis that the people who took the online Bilingualism and Emotions Questionnaire will rate themselves as fully fluent in speaking, comprehension, reading, and writing in their first language (ratings on the variable range from 1, least proficient, to 5, fully fluent). Use the variables l1speak, l1comp, l1read, and l1write. Calculate effect sizes and comment on their size.