

Answers to Application Activities in Chapter 8

8.1.1 Application Activity: Choosing a T-Test

- 1 **Reading attitudes.** Paired (the same students are tested at two different times).
- 2 **Reading ability.** Independent (one group received a treatment independent of the other group). Notice that since you have two scores though (score at the beginning and score at the end), if you compare scores of one group against the other you'll probably want to take a gain score (the final score minus the pretest score) to compare.
- 3 **Listening comprehension.** Independent (the beginner and advanced groups are considered independent of each other; had the question been whether the beginners performed equivalently on both parts of the test, a paired t -test would have been appropriate).
- 4 **Pitch.** Paired (the same people are being measured on two different but related tests, so their scores will not be independent).
- 5 **Learning strategy instruction.** Paired (the same students' scores are compared from the beginning to the end of the instruction period).
- 6 **Vocabulary learning I.** Paired (the same students were compared with themselves at the beginning and end of the semester).
- 7 **Vocabulary learning II.** Independent (the two groups were randomly selected and independent of each other). As with exercise #2, you'd want to look at the gain scores of the two groups to do an independent t -test. In other words, you shouldn't ignore

information you have gathered, and since you have the information from the first interview, you should use it in calculating a gain score.

- 8 **Morphology**. Independent (you want to compare the learners in one group with the learners in a different group).

8.2.7 Application Activities with Boxplots (Answers are Given for Both SPSS and R)

1 Leow and Morgan-Short (2004)

Use Leow and Morgan-Short's (2004) data (LeowMorganShort.sav). Import into R as **leowMs**.

We want to graph a series of two separate variables

SPSS Instructions:

Make gainscore: Create gainscores by going to TRANSFORM > COMPUTE VARIABLE. When the "Compute Variable" box opens up, move RECPOSTSCORE to the box that says "Numeric Expression." Add a minus sign, then move RECPRESCORE to the "Numeric Expression" box so you have the equation "RECPOSTSCORE – RECPRESCORE." In the box labeled "Target Variable," give this new variable the name RECFINAL. Press OK. A new variable is added to the end of the LeowMorganShort.sav file. Repeat with the equation "PROPOSTSCORE – PROPRESCORE" and give this new variable the name PROFINAL.

Make boxplot: See Section 8.2.4. Choose GRAPHS > LEGACY DIALOGS > BOXPLOT, then SIMPLE and "Summaries of separate variables" radio button. Press "Define." In the next dialogue box, put RECFINAL and PROFINAL in the "Variable" box. Press OK.

R Instructions:

Make gainscore: In R Commander, choose DATA > MANAGE VARIABLES IN ACTIVE DATASET > COMPUTE NEW VARIABLE. Choose `recpostscore` and double-click it, and it will show up in the “Expression to compute” box. Add a minus sign after it, then double-click on `recprescore` to finish. Now you have the equation “`recpostscore – recprescore`.” Give the new variable the name `recfinal` in the box labeled “New variable name.” Press OK. A new variable is added to the end of the `leowMs` file. The code for this procedure is:

```
leowMs$recfinal <- with(leowMs, recpostscore- recprescore)
```

Repeat with the equation “`propostscore – proprescore`.” Give the new variable the name `profinal`.

Make boxplot: See Section 8.2.5. In R Commander we cannot choose more than one variable at a time but we want to plot two separate variables, so use R code for the boxplot:

```
boxplot(leowMs$recfinal, leowMs$profinal, ylab="gainscore", names=c("Receptive test",  
"Productive test"), col="grey", boxwex=.5, medcol="white")
```

Results:

The median score is not that different between the receptive gainscore and the productive gainscore, and it is very close to zero for both. However, the distributions are quite different. The

range of the whiskers for the boxplot for the receptive gainscore is quite large (from about -3 to 17) while the range while the range for the whiskers of the boxplot for the productive gainscore is very small (from about -1 to 2) but there are many outliers (marked by circles in SPSS) and extreme outliers (marked by stars in SPSS). The distributions definitely do not have the same size box, and there are many outliers in the productive gainscore. Neither one of the variables has a normal distribution; the distribution for the receptive gainscore is skewed positively.

2 Leow and Morgan-Short (2004)

Use Leow and Morgan-Short's (2004) data (LeowMorganShort.sav). Import into R as `leowMs`. Prepare the variables as outlined in Exercise #1 above. Here we want to have two separate variables plotted but also have the variables split into groups (see Section 8.2.6).

SPSS Instructions:

Call for this type of side-by-side boxplot by choosing **GRAPHS > LEGACY DIALOGS > BOXPLOT**, then **CLUSTERED** and "Summaries of separate variables." Put `recfinal` and `profinal` in the box that says "Boxes Represent." Put `GROUP` in the "Category Axis" box.

R Instructions:

Call for this type of side-by-side boxplot in R by setting it up to print out two boxplots side by side:

```
par(mfrow=c(1,2)) #1 row, 2 columns of data
```

```
levels(leowMs$group)=c("NTA", "ThinkAloud") #Make labels shorter so they'll print
```

```
boxplot(recfinal~group,data=leowMs,col="gray", medcol="white", main="Receptive  
gainscore",boxwex=.5)
```

```
boxplot(profinal~group,data=leowMs, col="gray",medcol="white", main="Productive  
gainscore",boxwex=.5)
```

Results:

Every box except for the Think Aloud group in the receptive gainscore has outliers. The group that improved the most on the receptive measure will be the one with highest gainscore. The medians for both groups are exactly the same so from the boxplots it looks like the groups are equal (mean scores may be higher for the Think Aloud group as higher scores are not outliers for that group). The same situation holds for the productive measure as the median score is right at zero for both groups. The Think Aloud group has more spread than the NTA group for the receptive gainscore, but the NTA group has slightly more spread than the Think Aloud group for the productive gainscore.

If the boxplot is larger (whether just the box or the whole distribution from the end of the whiskers to the other end) it means the standard deviation of the distribution will be larger.

3 Yates (2003)

Use the Yates2003.sav dataset. If using in R, import as **yates**. In this file there are separate columns for each group and for the pretest and posttest. That means we basically want to graph a series of four separate variables, even though we would analyze the test as a paired-samples *t*-test.

SPSS Instructions:

See Section 8.2.4. For side-by-side boxplots of separate variables choose **GRAPHS > LEGACY DIALOGS > BOXPLOT**, then **CLUSTERED** and “Summaries of separate variables.” Put **LABBEFORE**, **LABAFTER**, **MIMICRYBEFORE** and **MIMICRYAFTER** in the box that says “Boxes Represent.” Press **OK**.

R Instructions:

See Section 8.2.5. In R Commander we cannot choose more than one variable at a time but we want to plot four separate variables, so use R code for the boxplot:

```
attach(yates)
boxplot(labbefore, labafter, mimicrybefore, mimicryafter, names=c("LabBefore", "LabAfter",
"MimicryBefore", "MimicryAfter"), col="grey", boxwex=.5, medcol="white",
main="Suprasemental practice")
detach(yates)
```

Results:

The lab group’s median score actually got worse in the post-test (**LABAFTER**) but the range of the distribution was smaller, which is an improvement. The median score for mimicry went down slightly too, but the range of distribution stayed the same, which seems to indicate a proficiency decrease. There are no outliers. Almost all of the variables have about the same amount of spread except for the **LabAfter** variable.

4 DeKeyser (2000)

Use the DeKeyser (2000) data (DeKeyser2000.sav) and import into R as `dekeyser`. The boxplot will consist of one variable which we want to split into groups.

SPSS Instructions:

See Section 8.2.2. Choose **GRAPHS > LEGACY DIALOGS > BOXPLOT**, then **SIMPLE** and “Summaries for groups of cases” radio button. Press “Define.” In the next dialogue box, put **GJTScore** in the “Variable” box and **STATUS** in the “Category Axis” box. Press **OK**.

R Instructions:

See Section 8.2.3. Use the `ggplot2` package for a boxplot with overlaid dots. If you do not already have this library installed, install it with this code:

```
install.packages("ggplot2")
```

Open the library and then use the `ggplot()` code to get a boxplot:

```
library(ggplot2)
```

```
ggplot(data=dekeyser,aes(x=status,y=gjtscore))+geom_boxplot(outlier.shape=NA)+  
geom_jitter(position=position_jitter(width=.3, height=0))+labs(x="Group",  
y="GJT score")
```

Results:

The groups performed quite differently, with the younger group receiving higher scores. These distributions do not have equal sizes of boxes (IQR), meaning the variances are different; the Under 15 group has a very small box while the Over 15 group has a bigger box. The distribution of the data within each of these boxplots is not totally symmetrical, but not extremely skewed either. There is one outlier in the Under 15 data (in the R output this is not marked by any special marker but you can see that the point lies outside the bottom whisker line).

5 Inagaki & Long (1999)

Use the Inagaki and Long (1999) *t*-test data (InagakiLong1999.Ttest.sav), and import it into R as `inagaki`. The boxplot will consist of one variable which we want to split into groups.

SPSS Instructions:

See Section 8.2.2. Choose **GRAPHS > LEGACY DIALOGS > BOXPLOT**, then **SIMPLE** and “Summaries for groups of cases” radio button. Press “Define.” In the next dialogue box, put **GAINSCORE** in the “Variable” box and **GROUP** in the “Category Axis” box. Press **OK**.

R Instructions:

See Section 8.2.3. Use the `ggplot2` package for a boxplot with overlaid dots. If you do not already have this library installed, install it with this code:

```
install.packages("ggplot2")
```

Open the library and then use the `ggplot()` code to get a boxplot:


```
library(ggplot2)

ggplot(data=inagaki,aes(x=group,y=gainscore))+geom_boxplot(outlier.shape=NA)+
geom_jitter(position=position_jitter(width=.3, height=.1))+labs(x="Group",
y="Gainscore")
```

Note that I changed the height of the jitter a bit because I could tell I wasn't seeing all the points on the zero median line, which is why the points around zero are not all on the line but instead above and below the line.

Results:

These are funny-looking boxplots because the median is close to zero so they are not symmetric. For the group that heard models of the structures (Model group) the median gainscore is zero, while for the group who heard recasts (Recast group) the median score is slightly higher (around .25). So can we say that the groups performed differently? A little but not too much.

One thing I notice in the `ggplot()` boxplot is that there are just not very many data points! There are just 8 participants for each group, so this is a very small N.

The box for the Recast group is slightly bigger than for the Model group, but they are not too different either, and variances are fairly equal. In the SPSS boxplot there are no outliers but the groups are not normally distributed because the boxes are not symmetric around the medians and the whiskers do not extend in both directions. The distributions are positively skewed (there are

more lower scores than would be expected in a normal distribution). In the `ggplot()` boxplot there is one outlier in the Model group (the person at 3; notice that the person at 1.5 looks like an outlier because it is outside of the box's whisker but this is only because the height is jittered in this graph). This means this group does not have a normal distribution, and both groups have positively skewed distributions as discussed for SPSS.

6 Larson-Hall and Connell (2005)

Use the Larson-Hall and Connell (2005) data (LarsonHall.Forgotten.sav) imported into R as `forget`. Here we want to have two separate variables plotted but also have the variables split into groups (see Section 8.2.6).

SPSS Instructions:

Call for this type of side-by-side boxplot by choosing **GRAPHS > LEGACY DIALOGS > BOXPLOT**, then **CLUSTERED** and “Summaries of separate variables.” Put **ACCENT ON R WITH 4 JUDGED** (**ACCENTR**) and **ACCENTL** in the “Boxes represent” box. Put **STATUS** in the “Category Axis” box. Put the ID variable in the “Label cases by” box. Press **OK**.

R Instructions:

Call for this type of side-by-side boxplot in R by setting it up to print out two boxplots side by side:

```
par(mfrow=c(1,2)) #1 row, 2 columns, one column for each variable
```

```
names(forget) #get names of variables to put in code
```

```
boxplot(accentr~status,data=forget,col="gray", medcol="white", main="Accent of  
R",boxwex=.5)
```

```
boxplot(accentl~status,data=forget, col="gray",medcol="white", main="Accent of  
L",boxwex=.5)
```

Results:

The participants all received higher accent scores on words beginning with /l/. The variances are not equal across groups; in general, the Non group had the widest variance. The medians show that the mean accent score increases from the Non group being the worst, the Late group being a little better, and the Early group being better than them. The boxplots for the words with /l/ look fairly symmetrical, except the Late group's box is skewed to the left. Also, there are outliers in the Late group. Almost all of the boxplots for the /r/ words are skewed, but in different directions. Only the boxplot for the Late group looks relatively symmetric.

7 French and O'Brien (2008)

Use the SPSS file French & O'Brien grammar.sav, and import it into R as **French**. The boxplot will consist of four variables plotted side by side, even though we would analyze the test as a paired-samples *t*-test.

SPSS Instructions:

See Section 8.2.4. For side-by-side boxplots of separate variables choose GRAPHS > LEGACY DIALOGS > BOXPLOT, then CLUSTERED and "Summaries of separate variables." Put ANWR_1, ANWR_2, ENWR_1 and ENWR_2 in the box that says "Boxes Represent." Press OK.

R Instructions:

See Section 8.2.5. In R Commander we cannot choose more than one variable at a time but we want to plot four separate variables, so use R code for the boxplot:

```
attach(French)
```

```
boxplot(anwr_1,anwr_2,enwr_1,enwr_2,names=c("Arabic Time 1", "Arabic Time 2",  
"English Time 1", "English Time 2"), col="grey", boxwex=.5, medcol="white",  
main="Phonological memory")
```

```
detach(French)
```

Results:

Looking at the median lines, the score for the English test went up slightly from Time 1 to Time 2, while the score for the Arabic test went down slightly from Time 1 to Time 2. The variances for the Arabic test are pretty much the same (slightly smaller for Time 2) and my suspicion is that there is really no difference in scores from Time 1 to Time 2 for the Arabic test. For the English test, the difference in median score is not much but the variance is smaller, so there may be some statistical improvement from Time 1 to Time 2 for this test (which is, of course, what the authors did *not* want!). The English Time 2 score is the most skewed, having more lower scores than one would expect from a normal distribution. The other distributions do not have the median line exactly in the middle of the IQR box but are not highly skewed.

8.4.7 Application Activities for the Independent-Samples T-Test

1 Larson-Hall (2008)

Use the Larsonhall2008.sav file, imported to R as `lh2008`.

A. Explore Assumptions (Normal Distribution, Equal Variances):

To explore whether the data are normally distributed and have equal variances, look at boxplots of the data. Look at multiple variables divided by group, but with so many variables a multiple boxplot may be too cluttered to be a good diagnostic, so I suggest looking at each variable separately, split into groups, as explained below:

SPSS Instructions:

Go TO GRAPHS > BOXPLOT, pick SIMPLE and “Summaries for groups of cases.” Put one of the variables such as [APTScore] into the “Variables” box. Put “1 is no [ERLYEXP]” into the “Category Axis” box. Repeat for the other variables: “Total score of 4 situations for use of English in life now [USEENG],” GJTSCORE, score on the R/L/W test [RLWScore]).

R Instructions:

```
library(ggplot2)

ggplot(data=lh2008,aes(x=erlyexp,y=aptscore))+geom_boxplot(outlier.shape=NA)+
geom_jitter(position=position_jitter(width=.3, height=0))+labs(x="Group",
y="Aptitude score")
```

This is for the aptitude score; repeat for the three other variables: `useeng`, `gjtscore` and `rlwscore`.

Results:

Only the GJT score appears to satisfy parametric assumptions. For APTSCORE, variances look equal but distribution is negatively skewed and there are outliers; for USEENG, again, variances look equal but distribution is positively skewed and there are many outliers; for GJTSCORE, these boxplots look normally distributed and variances are very similar; for RLWSCORE boxplots look symmetric and variances are similar, but there are some outliers. For almost all of these variables it would be good to use robust methods as well as parametric methods.

B. Run the Independent Samples T-Test**SPSS Instructions:**

Choose ANALYZE > COMPARE MEANS > INDEPENDENT SAMPLES T-TEST. Move the four variables (USEENG, GJT, RLWSCORE and APTSCORE in this case) into the “Test Variable(s)” box. Move the categorical grouping variable (ERLYEXP) into the “Grouping Variable” box. Click on the “Define groups” button (if you can’t, click again on the variable that is now in the “Grouping Variable” box but says “ErlyExp(? ?)” and highlight it, and then the “Define groups” button should work). Put in the number “1” for Group 1 and the number “2” for Group 2 (there are only two groups in this data). Press CONTINUE. Press OK to run the analysis.

R Instructions:

Make sure **lh2008** is the active dataset in R Commander. Click on STATISTICS > MEANS > INDEPENDENT SAMPLES T-TEST. Choose the variable **aptscore** (you can only pick one at a time).

The **erlyexp** variable should already be selected under the “Groups (pick one)” heading. Unless you have a good reason to change them, leave the “Options” tab alone. After you finish this, you can go back and do the same thing for the other three variables: **useeng**, **gjtscore** and **rlwscore**.

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

```
t.test(aptscore~erlyexp, alternative='two.sided', conf.level=.95,  
var.equal=FALSE, data=lh2008)
```

This test will give you the mean scores of both groups but not the standard deviations nor the number in each group. There are a number of ways to get this information but an easy way in R Commander is to choose STATISTICS > SUMMARIES > NUMERICAL SUMMARIES. You can pick all 4 of the variables at once, and then choose the “Summarize by groups” button and the only group variable in the data, **erlyexp**, will already be chosen. Choose OK for this command, then OK again to run.

Results:

For the parametric *t*-test, use the “Equal variances not assumed row” in SPSS as a default (if you have not changed the options it is already the default for R Commander). To get effect sizes, you can use an online calculator that pools the means and standard deviations of both groups or calculate it by hand. I used the calculator at www.uccs.edu/~lbecker/.

Here are the results in tabular form (Results are from R, these differ a little bit in the CI from SPSS)

Parametric *t*-tests

Variable	95% CI of the difference btw. means	Mean 1 (SD1) not early	Mean 2 (SD2) early	N1/N2	Effect size
Aptitude Score	-1.58,.99	31.2 (4.4)	31.5 (4.2)	139/61	.07
Use of English	-.95,.76	7.0 (2.8)	7.1 (2.7)	131/56	.04
GJT Score	-5.63,.76	112.3 (10.1)	114.7 (10.7)	139/61	.23
R/L/W Score	-9.74, -1.06	49.3 (13.2)	54.7 (14.7)	139/61	.39

Starting with the mean scores, we can see that although in all cases the mean score for the early learners was higher, it was not higher by a lot. Only the RLWSCORE's CI does not pass through zero for the regular 95% CI. However, we can see that the difference between the groups for the R/L/W score may be as small as about 1 point (1.06 to be exact). Note additionally that the effect size for all comparisons are quite small. So I would say that these results show that the earlier learners could do slightly better in the pronunciation measure, but the actual effect of the earlier study was very slight. The fact that there was no difference between the groups in their aptitude score shows that there is no evidence that the people who began studying English at an earlier age did so because they were just naturally better at learning languages. In addition, starting to study English at an earlier age in this minimal input environment did not lead to increased use of English.

2 Inagaki & Long (1999)

Use the Inagaki and Long (1999) *t*-test data (InagakiLong1999.Ttest.sav), imported into R as `inagaki`. The boxplot will consist of one variable which we want to split into groups.

A. Explore Assumptions (Normal Distribution, Equal Variances):

Look back at Exercise #5 in Section 8.2.7. There we saw that the data is definitely not normally distributed, with a very skewed distribution and one outlier, but variances seemed fairly equal.

B. Run the Independent Samples T-Test

SPSS Instructions:

Choose ANALYZE > COMPARE MEANS > INDEPENDENT SAMPLES T-TEST. Move GAINSCORE into the “Test Variable(s)” box. Move the categorical grouping variable (GROUP) into the “Grouping Variable” box. Click on the “Define groups” button (if you can’t, click again on the variable that is now in the “Grouping Variable” box but says “Group(? ?)” and highlight it, and then the “Define groups” button should work). Put in the number “1” for Group 1 and the number “2” for Group 2 (there are only two groups in this data). Press CONTINUE. Open the “Bootstrap” button and check the “Perform bootstrapping” box. Change the number of samples to 10,000 instead of 1000. Change the confidence intervals to “BCa”. Press CONTINUE. Press OK to run the analysis.

R Instructions:

Make sure **inagaki** is the active dataset in R Commander. Click on STATISTICS > MEANS > INDEPENDENT SAMPLES T-TEST. When you open the menu choice, the **group** variable will already be selected under the “Groups (pick one)” heading, as will the **gainscore** variable under the “Response Variable (pick one)” heading. Unless you have a good reason to change them, leave the “Options” tab alone.

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

```
t.test(gainscore~group, alternative='two.sided', conf.level=.95,  
var.equal=FALSE, data=inagaki)
```

This test will give you the mean scores of both groups but not the standard deviations nor the number in each group. There are a number of ways to get this information but an easy way in R Commander is to choose STATISTICS > SUMMARIES > NUMERICAL SUMMARIES. You can pick all 4 of the variables at once, and then choose the “Summarize by groups” button and the only group variable in the data, `group`, will already be chosen. Choose OK for this command, then OK again to run.

Run a bootstrapped independent t -test as well:

```
MeanDifference <- function(data, i) {  
  temp <- data[i, ]  
  aov.temp <- aov(gainscore ~ group, data = temp)  
  Tuk <- TukeyHSD(aov.temp)  
  return(Tuk$erlyexp[ , 1])  
}
```

```
library(boot) #use this if needed
```

```
MeanDifferenceBoot <- boot(inagaki, MeanDifference, 10000)
```

```
MeanDifferenceBoot.ci <- boot.ci(MeanDifferenceBoot, conf=0.95, type="bca")
```

```
print(MeanDifferenceBoot.ci)
```

Results:

For the parametric t -test, use the “Equal variances not assumed row” in SPSS as a default (if you have not changed the options it is already the default for R Commander). To get effect sizes, you can use an online calculator that pools the means and standard deviations of both groups or calculate it by hand. Here I used the calculation found in Section 8.4.5 with the standardizer being the pooled standard deviation of the two groups.

Results are from SPSS, these differ a little bit in the CI from R)

The 95% CI for the difference in mean scores between the groups is [-1.27, 1.02], which contains zero, so there is no statistical difference between groups. Mean (model) = .69, s.d. = 1.10, N = 8; Mean(recast) = .81, s.d. = 1.03, N = 8. Effect size is $0.12/1.07 = .11$, which is a very small effect size.

3 Larson-Hall and Connell (2005)

Use the Larson-Hall and Connell (2005) data (LarsonHall.Forgotten.sav) imported into R as `forget`. We won't worry about the assumptions for this dataset (we already looked at them in Exercise #3 of Section 8.2.7 and saw that they are not exactly normally distributed).

A. Run the Independent Samples T-Test

SPSS Instructions:

Choose ANALYZE > COMPARE MEANS > INDEPENDENT SAMPLES T-TEST. Move SENTENCEACCENT into the “Test Variable(s)” box. Move the categorical grouping variable (STATUS) into the “Grouping Variable” box. Click on the “Define groups” button (if you can’t, click again on the variable that is now in the “Grouping Variable” box but says “Group(? ?)” and highlight it, and then the “Define groups” button should work). Put in the number “1” for Group 1 and the number “2” for Group 2 (there are three groups in this data, so this run compares only the “Non” and “Early” groups). Press CONTINUE. Press OK to run the analysis. Repeat this again by following the same menu choice (ANALYZE > COMPARE MEANS > INDEPENDENT SAMPLES T-TEST) and clicking on the “Define groups” button (click the variable that is now in the “Grouping Variable” box but says “Status(1,2)” and the “Define Groups” button will become visible). Now change the group numbers to “Group 1: 2” and “Group 2: 3.” This will compare the “Late” and “Early” groups. Press OK and then OK again to run the second comparison. There is no need to step with the “Bootstrap” button, as the previous settings will stay the same. Repeat this again to compare group 1 to group 3.

R Instructions:

Make sure `forget` is the active dataset in R Commander. Click on STATISTICS > MEANS > INDEPENDENT SAMPLES T-TEST. When you open the menu choice, you will see that there is only one variable available under the “Groups (pick one)” heading, and it is `sex`, not the `status` variable you should choose! This means the `status` variable is not in the correct format. Press “Cancel” and leave the dialogue box for the *t*-test. Go to R console and type:

`str(forget)`

This will show you the structure of this dataset. You can see that `status` is a factor with three levels, while `sex` is a factor with two levels. This means that `status` is already in the correct form (because it's a factor) and there is no need to change it, but it seems that R Commander won't include a factor that has more than two levels as a choice in its dialogue. However, we might suspect there is a way to specify which groups to compare in the levels of a factor in the *t*-test code, so look at the help page for the `t.test()` command by typing:

`?t.test`

A help page will appear and show usage. It looks like if we had one set of variables in *x* and one set in *y* they could be compared, so it looks like we'll need to subset the `sentenceaccent` variable into its parts. First, I look at the grouping variable and see which rows are contained in which level of status (Non, Late or Early), then I create a new variable with the `sentenceaccent` data for those rows:

```
> forget$status
[1] Non   Non   Non   Non   Non   Non   Non   Non   Non   Non   Non   Non
[13] Non   Non   Non   Late  Late  Late  Late  Late  Late  Late  Late  Late
[25] Late  Late  Late  Late  Late  Late  Early Early Early Early Early Early
[37] Early Early Early Early Early Early Early Early Early
Levels: Non Late Early
> Non<-forget$sentenceaccent[1:15]
> Late<-forget$sentenceaccent[16:30]
> Early<-forget$sentenceaccent[31:44]
```

Now we can run the `t.test()` command with Non as the *x*-variable and Late as the *y*-variable, and all of the other permutations (Non vs. Early, and Late vs. Early), like this:

```
t.test(Non, Late, alternative='two.sided', conf.level=.95, var.equal=FALSE)
```

Means, standard deviations and Ns (needed for effect sizes, and should be reported as well) can be found by choosing STATISTICS > SUMMARIES > NUMERICAL SUMMARIES for **sentenceaccent** split by the **status** variable in R Commander.

Results:

First, here are descriptive statistics:

	Mean	SD	N	20% Trimmed mean
Non	2.51	1.01	15	2.46
Late	3.10	0.93	15	3.03
Early	4.59	1.12	14	4.55

Here are results from SPSS and the effect sizes:

	95% CI of mean difference	Effect size (d)
Non vs. Late	-1.32, 0.14	.61
Late vs. Early	-2.28, -0.70	1.45
Early vs. Non	-2.90, -1.27	1.59

The descriptive statistics show that the Early learners of English performed better on the scale than the Late or Non users of English, but with relatively similar standard deviations (variances).

The CIs show that the Early learners are definitely different from the Late and Non users of English. Because sentence accent was judged on a scale of 0–7, a difference in even 1 point can be considered to be a large difference. The effect sizes show (I used an online calculator and pooled SDs for the calculation) that the effect for the difference in the Early learners was a very large difference, about 1½ times the difference in standard deviations.

4 Larson-Hall (2008)

SPSS Instructions:

Choose ANALYZE > COMPARE MEANS > INDEPENDENT SAMPLES T-TEST. Move the variable (GJTscore) into the “Test Variable(s)” box. Move the categorical grouping variable (ERLYEXP) into the “Grouping Variable” box. Click on the “Define groups” button (if you can’t, click again on the variable that is now in the “Grouping Variable” box but says “ErlyExp(? ?)” and highlight it, and then the “Define groups” button should work). Put in the number “1” for Group 1 and the number “2” for Group 2 (there are only two groups in this data). Press CONTINUE. Open the “Bootstrap” button and check the “Perform bootstrapping” box. Change the number of samples to 10,000 instead of 1000. Change the confidence intervals to “BCa.” Press CONTINUE. Press OK to run the analysis.

R Instructions:

Run a bootstrapped independent t -test:

```
MeanDifference <- function(data, i) {  
  temp <- data[i, ]  
  aov.temp <- aov(gjtscore ~ erlyexp, data = temp)  
  Tuk <- TukeyHSD(aov.temp)  
  return(Tuk$erlyexp[ , 1])  
}  
  
library(boot)
```

```
MeanDifferenceBoot <- boot(lh2008, MeanDifference, 10000)
```

```
MeanDifferenceBoot.ci <- boot.ci(MeanDifferenceBoot, conf=0.95, type="bca")
```

```
print(MeanDifferenceBoot.ci)
```

Now let's try the bootstrapping explained in Section 8.4.4 of the book. First, load in all of Wilcox's commands so you can open the WRS library, as explained in Section 8.4.4, then open the library:

```
library(WRS)
```

To use the `trimpb2()` command we'll need to cut the data up into data for the `gjtscor` for each group. This is a little trickier than was seen in the previous exercise (#3) because the "Not Early" and "Early" groups are interleaved in this dataset (we can't just say that we want rows 1–100 for one group, and 101–200 for the other group!). Fortunately, doing so is really not hard; it just involves specifying what the setting is for the group variable (`erlyexp`). Here's now I did it:

```
NotEarly<-lh2008$gjtscor[lh2008$erlyexp=="Not Early"]
```

```
Early<-lh2008$gjtscor[lh2008$erlyexp=="Early"]
```

Now use the `trimpb2()` command with the new variables and find the trimmed means:

```
trimpb2(Early,NotEarly)
```

```
mean(Early, tr=.2)
```



```
mean(NotEarly, tr=.2)
```

This command will result in a percentile-bootstrapped and 20% means-trimmed 95% CI.

Lastly, Section 8.4.4 specifies how to do a bootstrap-t which works best without any trimming:

```
yuenbt(Early, NotEarly)
```

Results:

Bootstrapped differences in means (your results will differ from these because of the random sampling in the bootstrapping)

	Early vs. Not Early
95% CI of the difference in means (SPSS)	-5.43, 1.11
95% Bootstrapped BCa CI (SPSS)	-5.37, 0.97
95% Bootstrapped BCa CI (R)	-0.72, 5.62
95% percentile bootstrap 20% trimmed means CI	-0.71, 6.27
95% bootstrap-t	-0.96, 6.07

In no case does the CI from any of the robust tests *not* run through zero, although some CIs are slightly narrower (and more precise) than others.

5 Cole (1927)

Use the SPSS file Cole1927.sav, imported into R as **cole**.

SPSS Instructions:

Make gainscore: Create gainscores by going to TRANSFORM > COMPUTE VARIABLE. When the “Compute Variable” box opens up, move, for example, TIME2_GRAMMAR_PERCENT to the box that says “Numeric Expression.” Add a minus sign, then move TIME1_GRAMMAR_PERCENT to the “Numeric Expression” box so you have the equation “TIME2_GRAMMAR_PERCENT – TIME1_GRAMMAR_PERCENT.” In the box labeled “Target Variable,” give this new variable the name GRAMMAR. Press OK. A new variable is added to the end of the Cole1927.sav file. Repeat with the equation “TIME2_VOCAB_PERCENT – TIME1_VOCAB_PERCENT” and name it VOCAB, and also “TIME2_READING_PERCENT – TIME1_READING_PERCENT” and name it READING.

Run the Independent Samples T-Test: Choose ANALYZE > COMPARE MEANS > INDEPENDENT SAMPLES T-TEST. Move GRAMMAR, VOCAB & READING into the “Test Variable(s)” box. Move the categorical grouping variable (GROUP) into the “Grouping Variable” box. Click on the “Define groups” button (if you can’t, click again on the variable that is now in the “Grouping Variable” box but says “Group(? ?)” and highlight it, and then the “Define groups” button should work). Put in the number “1” for Group 1 and the number “2” for Group 2 (there are only two groups in this data). Press CONTINUE. Open the “Bootstrap” button and check the “Perform bootstrapping” box. Change the number of samples to 10,000 instead of 1000. Change the confidence intervals to “BCa”. Press CONTINUE. Press OK to run the analysis.

R Instructions:

Make gainscore: In R Commander, make sure **cole** is the active dataset. Choose DATA >

MANAGE VARIABLES IN ACTIVE DATASET > COMPUTE NEW VARIABLE. Choose

time2_grammar_percent and double-click it, and it will show up in the “Expression to compute” box. Add a minus sign after it, then double-click on **time2_grammar_percent** to finish. Now you have the equation “**time2_grammar_percent – time1_grammar_percent.**”

Give the new variable the name **grammar** in the box labeled “New variable name.” Press OK. A new variable is added to the end of the **cole** file. The code for this procedure is:

```
cole$grammar <- with(cole, time2_grammar_percent- time1_grammar_percent)
```

Repeat with the equation “**time2_vocab_percent – time1_vocab_percent**” and give the new variable the name **vocab**, then “**time2_reading_percent – time1_reading_percent**” and give the new variable the name **reading**.

Run the Independent Samples T-Test: Make sure **cole** is the active dataset in R Commander.

Click on STATISTICS > MEANS > INDEPENDENT SAMPLES T-TEST. When you open the menu choice, the **group** variable will already be selected under the “Groups (pick one)” heading. Choose first the **grammar** variable under the “Response Variable (pick one)” heading. Unless you have a good reason to change them, leave the “Options” tab alone. Press OK.

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

```
t.test(grammar~group, alternative='two.sided', conf.level=.95, var.equal=FALSE,  
data=cole)
```

Repeat with the other variables of `vocab` and `reading`.

This test will give you the mean scores of both groups but not the standard deviations nor the number in each group. There are a number of ways to get this information but an easy way in R Commander is to choose STATISTICS > SUMMARIES > NUMERICAL SUMMARIES. You can pick all 3 of the variables at once, and then choose the “Summarize by groups” button and the only group variable in the data, `group`, will already be chosen. Choose OK for this command, then OK again to run.

Robust test instructions:

Run a bootstrapped independent *t*-test as well:

```
MeanDifference <- function(data, i) {  
  temp <- data[i, ]  
  aov.temp <- aov(vocab ~ group, data = temp)  
  Tuk <- TukeyHSD(aov.temp)  
  return(Tuk$group[ , 1])  
}  
  
library(boot) #use this if needed
```

```
MeanDifferenceBoot <- boot(cole, MeanDifference, 10000)
```

```
MeanDifferenceBoot.ci <- boot.ci(MeanDifferenceBoot, conf=0.95, type="bca")
```

```
print(MeanDifferenceBoot.ci)
```

To get the other results I just substituted in each of the other two variables (`grammar`, `reading`) into the slot where `vocab` is in the code above and pasted in that code to R to define the

`MeanDifference()` function (ending with the curly bracket), then ran the last three lines of code (starting with `MeanDifferenceBoot <- boot`) to get the bootstrapped results.

Try the robust *t*-tests in Section 8.4.4 as well.

```
library(WRS)
```

To use the `trimpb2()` command we'll need to cut the data up into data for each group. Just as in a previous exercise (#3) we can look at the lines that define the groups this way and cut up each variable (`reading`, `grammar`, `vocab`) into its parts to run the command:

```
cole$group
```

```
trimpb2(cole$grammar[1:15], cole$grammar[16:30])
```

```
trimpb2(cole$vocab[1:15], cole$vocab[16:30])
```

```
trimpb2(cole$reading[1:15], cole$reading[16:30])
```

Last, we can run the `yuenbt()` command:

```
yuenbt(cole$grammar[1:15], cole$grammar[16:30])
```

```
yuenbt(cole$vocab[1:15], cole$vocab[16:30])
```

```
yuenbt(cole$reading[1:15], cole$reading[16:30])
```

Results:

Variable	95% CI of the difference btw. means	Mean 1 (SD1) Composition	Mean 2 (SD2) Grammar Translation	N1/N2	Effect size
Grammar	-6.51, 6.25	5.6 (10.29)	5.73 (6.04)	15/15	.02
Vocab	-4.97, 4.09	13.07 (5.19)	13.51 (6.78)	15/15	.07
Reading	-7.59, 4.73	3.81 (9.68)	5.24(6.31)	15/15	.18

	Grammar	Vocab	Reading
95% Bootstrapped BCa CI (SPSS)	-6.19, 6.30	-5.04, 3.92	-7.43, 4.28
95% Bootstrapped BCa CI (R)	-6.13, 5.64	-3.56, 4.89	-3.87, 7.75
95% percentile bootstrap 20% trimmed means CI	-7.11, 5.33	-4.74, 5.04	-5.16, 4.37
95% bootstrap-t	-6.81, 4.37	-3.73, 5.31	-5.38, 4.21

The results show that there is no way to get any statistical results for any of the tests, no matter whether they are robust tests or not! The effect sizes basically show that the effects for differences between groups are very small, so no amount of statistical manipulation will cause any differences to become salient. You've heard about how experiments designed to test whether different teaching approaches didn't find any differences between groups? This data shows how that occurs!

8.5.6 Application Activities with Paired-Sample T-Tests

1 French and O'Brien (2008)

Use the SPSS file French & O'Brien grammar.sav, imported into R as **French**.

A. Explore Assumptions (Normal Distribution, Equal Variances):

Assumptions (Boxplots): Look back at Exercise #7 in Section 8.2.7. The boxplots showed that variances were pretty similar and that the distributions were not highly skewed, so we might surmise these are fairly normally distributed variables.

B. Run the Paired Samples T-Test

SPSS Instructions:

Go to ANALYZE > COMPARE MEANS > PAIRED-SAMPLES T TEST. Put the pairs of variables (ANWR_1:ANWR_2, ENWR_1:ENWR_2) in the “Paired Variables” box. Unless you have some unusual circumstances you would not need to use the “Options” button. Press OK.

R Instructions:

Make sure **French** is the active dataset in R Commander. Choose STATISTICS > MEANS > PAIRED T-TEST. You can only do this one pair at a time, so the first time choose **anwr_1** for the first variable under the heading “First variable (pick one)” and then **anwr_2** for the second variable under the heading “Second variable (pick one).” There’s no need to use the Options tab unless you want a one-sided hypothesis or to change the confidence level from 95%. Press OK.

The R code for this test is:

```
with(French, (t.test(anwr_1, anwr_2, alternative='two.sided',  
conf.level=.95, paired=TRUE)))
```

Repeat this with the other pair of variables (`enwr_1` and `enwr_2`).

This test will give you the confidence interval of the difference between mean scores of the groups but not the mean scores of the groups, the standard deviations nor the number in each group. There are a number of ways to get this information but an easy way in R Commander is to choose STATISTICS > SUMMARIES > NUMERICAL SUMMARIES. You can pick all 4 of the variables at once (`anwr_1:anwr_2`, `enwr_1:enwr_2`) and then press OK for this command (here we are not dividing results into any groups). The R code for this is:

```
numSummary(French[,c("anwr_1", "anwr_2", "enwr_1", "enwr_2")],  
statistics=c("mean", "sd", "IQR", "quantiles"), quantiles=c(0,.25,.5,.75,1))
```

To get correlation values between the pairs of variables, choose STATISTICS > SUMMARIES > CORRELATION MATRIX. Choose all of the variables and hit OK.

Results:

I used the cognitivelyflexibility.org/effectsize calculator and entered in the correlation between the paired variables to get the Cohen's d effect size given here. This effect size uses the average SD of the two variables.

Pairing	95% CI of the mean difference	Mean time 1 (SD1)	Mean time 2 (SD2)	N1/N2	Correlation (r)	Cohen's d effect size
ANWR_1 & ANWR_2	-0.52, .02	16.36 (5.22)	16.61 (4.81)	104/104	.97	0.2
ENWR_1 & ENWR_2	-3.54, -2.68	16.34 (4.13)	19.44 (3.74)	104/104	.85	1.44

The confidence intervals show that there was a difference between Time 1 and Time 2 for the English non-words. The authors wanted a measure that wouldn't change between Time 1 and Time 2, and the confidence interval shows that the Arabic non-words test contains zero so we will assume there was no difference from Time 1 to Time 2. Of course, this experiment doesn't *prove* that there is no difference between times (absence of evidence is not evidence of absence) but the small effect size should lead to the conclusion that the Arabic non-words test is one where the students are much less likely to improve over the course of time.

2 French and O'Brien (2008)

Use the SPSS file French & O'Brien grammar.sav, imported into R as **French**.

A. Explore Assumptions (Normal Distribution, Equal Variances):

Make boxplots (we haven't previously seen boxplots of these variables)

SPSS Instructions:

These instructions are for seeing all three variables at their paired times in one graph. Choose GRAPHS > LEGACY DIALOGS > BOXPLOT, then SIMPLE and "Summaries of separate variables" radio button. Press "Define". In the next dialogue box, put RVOCAB_1, RVOCAB_2, PVOCAB_1, PVOCAB_2, GRAM_1 and GRAM_2 in the "Variable" box Press OK.

R Instructions:

These instructions are seeing boxplots of paired variables, and then combining the graphs so there are three graphs together. In R Commander we cannot choose more than one variable at a time, so use R code for the boxplot:

```
par(mfrow=c(1,3)) #1 row, 2 columns of data  
boxplot(French$rvocab_1, French$rvocab_2, names=c("Receptive vocab Time 1", "Receptive vocab Time 2"), las=1, notch=F, col="grey", boxwex=.5, medcol="white")  
boxplot(French$pvocab_1, French$pvocab_2, names=c("Productive vocab Time 1", "Productive vocab Time 2"), las=1, notch=F, col="grey", boxwex=.5, medcol="white")  
boxplot(French$gram_1, French$gram_2, names=c("Grammar Time 1", "Grammar Time 2"), las=1, notch=F, col="grey", boxwex=.5, medcol="white")  
par(mfrow=c(1,1)) #return to normal
```

Results:

Boxplots: Grammar Time 1 has a number of outliers, but except for this variable, the other variables are fairly symmetrically and normally distributed, and variances are similar among all the variables too.

B. Run the Paired Samples T-Test

SPSS Instructions:

Go to ANALYZE > COMPARE MEANS > PAIRED-SAMPLES T TEST. Put the pairs of variables (RVOCAB_1: RVOCAB_2, PVOCAB_1:PVOCAB_2, GRAM_1: GRAM_2) in the “Paired Variables” box.

Unless you have some unusual circumstances you would not need to use the “Options” button.
Press OK.

R Instructions:

Make sure **French** is the active dataset in R Commander. Choose STATISTICS > MEANS > PAIRED T-TEST. You can only do this one pair at a time, so the first time choose **gram_1** for the first variable under the heading “First variable (pick one)” and then **gram_2** for the second variable under the heading “Second variable (pick one).” There’s no need to use the Options tab unless you want a one-sided hypothesis or to change the confidence level from 95%. Press OK.

The R code for this test is:

```
with(French, (t.test(gram_1, gram_2, alternative='two.sided',  
conf.level=.95, paired=TRUE)))
```

Repeat this with the other pairs of variables.

This test will give you the confidence interval of the difference between mean scores of the groups but not the mean scores of the groups, the standard deviations nor the number in each group. There are a number of ways to get this information but an easy way in R Commander is to choose STATISTICS > SUMMARIES > NUMERICAL SUMMARIES. You can pick all 6 of the variables at once (**rvocab_1**, **rvocab_2**, **pvocab_1**, **pvocab_2**, **gram_1** and **gram_2**) and then press OK for this command (here we are not dividing results into any groups). The R code for this is:

```
numSummary(French[,c("gram_1", "gram_2", "pvocab_1", "pvocab_2", "rvocab_1",
"rvocab_2")], statistics=c("mean", "sd", "IQR", "quantiles"),
quantiles=c(0,.25,.5,.75,1))
```

To get correlation values between the pairs of variables, choose STATISTICS >SUMMARIES > CORRELATION MATRIX. Choose all of the variables and hit OK.

Results:

I used the cognitivelyflexibility.org/effectsize calculator and entered in the correlation between the paired variables to get the Cohen's d effect size given here. This effect size uses the average SD of the two variables.

Pairing	95% CI of the mean difference	Mean Time 1 (SD1)	Mean Time 2 (SD2)	N1/N2	Correlation (r)	Cohen's d effect size
RVOCAB_1 & RVOCAB_2	-5.98, -4.67	16.58 (4.48)	27.24 (4.60)	104/104	.85	4.3
PVOCAB_1 & PVOCAB_2	-12.85, -11.93	30.61 (5.87)	43.0 (6.52)	104/104	.55	2.1
GRAM_1 & GRAM_2	-11.50, -9.83	32.81 (5.81)	38.13 (6.38)	104/104	.93	2.3

Participants improved on all three measures over the course of their study. The improvement in grammar was the largest, with an average of 10.7 points out of 45 total and a huge effect size of $d=4.3$ (over 4 standard deviations of change). Participants on average improved least on the receptive vocabulary measure, with an average change of 5.3 points out of 60. The effect sizes for all measures were very large and showed that the participants did very well at Time 2. The CIs are all quite a way away from zero and fairly narrow, showing precision in the estimate of the mean difference, and showing that the effect was really strong.

3 Yates (2003)

Use the Yates2003.sav dataset. If using in R, import as `yates`.

A. Explore Assumptions (Normal Distribution, Equal Variances):

Assumptions (Boxplots): Look back at Exercise #3 in Section 8.2.7. The boxplots showed that variances were pretty similar except for the LabAfter variable. Distributions were not highly skewed, so we might surmise these are fairly normally distributed variables.

B. Run the Paired Samples T-Test

SPSS Instructions:

Go to ANALYZE > COMPARE MEANS > PAIRED-SAMPLES T-TEST. Put the pairs of variables (LABBEFORE: LABAFTER and MIMICRYBEFORE: MIMICRYAFTER) in the “Paired Variables” box. Unless you have some unusual circumstances you would not need to use the “Options” button. Press OK.

R Instructions:

Make sure `yates` is the active dataset in R Commander. Choose STATISTICS > MEANS > PAIRED T-TEST. You can only do this one pair at a time, so the first time choose `labbefore` for the first variable under the heading “First variable (pick one)” and then `labafter` for the second variable under the heading “Second variable (pick one).” There’s no need to use the Options tab unless you want a one-sided hypothesis or to change the confidence level from 95%. Press OK.

The R code for this test is:

```
with(yates, (t.test(labbefore, labafter, alternative='two.sided',
```

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

Repeat this with the other pair of variables (`mimicrybefore`, `mimicryafter`).

This test will give you the confidence interval of the difference between mean scores of the groups but not the mean scores of the groups, the standard deviations nor the number in each group. There are a number of ways to get this information but an easy way in R Commander is to choose STATISTICS > SUMMARIES > NUMERICAL SUMMARIES. You can pick all 4 of the variables at once and then press OK for this command (here we are not dividing results into any groups).

The R code for this is:

```
numSummary(yates[,c("labafter", "labbefore", "mimicryafter",  
"mimicrybefore")], statistics=c("mean", "sd", "IQR", "quantiles"),  
quantiles=c(0,.25,.5,.75,1))
```

To get correlation values between the pairs of variables, choose STATISTICS > SUMMARIES > CORRELATION MATRIX. Choose all of the variables and hit OK.

Results:

I used the cognitivelyflexibility.org/effectsize calculator and entered in the correlation between the paired variables to get the Cohen's d effect size given here. This effect size uses the average SD of the two variables.

Variable	95% CI	Mean Time 1 (SD1)	Mean Time 2 (SD2)	N1/N2	Correlation r)	Effect size
Lab	-9.75, 8.15	109.4 (13.9)	110.2 (10.0)	10/10	.49	.07
Mimicry	-2.92, 13.53	112.0 (16.0)	106.7 (13.2)	10/10	.70	.47

Neither one of the tests is statistical although the effect size for Mimicry is much larger than that of Lab. The confidence intervals show that while the CI for Lab is centered near to zero, that of Mimicry is more lopsided and with further participants (just 10 is rather small) if the precision were increased the CI might lie outside of zero and the participants might have statistically smaller accent ratings (so a reduction in rating over time is desirable).

4 Larson-Hall and Connell (2005)

Use the Larson-Hall and Connell (2005) data (LarsonHall.Forgotten.sav) imported into R as **forget**.

SPSS Instructions:

Go to ANALYZE > COMPARE MEANS > PAIRED-SAMPLES T-TEST. Put the pairs of variables (ACCENTR, ACCENTL) in the “Paired Variables” box. Unless you have some unusual circumstances you would not need to use the “Options” button. Press OK.

R Instructions:

Make sure **forget** is the active dataset in R Commander. Choose STATISTICS > MEANS > PAIRED T-TEST. Choose **accentR** for the first variable under the heading “First variable (pick one)” and

then **accentL** for the second variable under the heading “Second variable (pick one).” There’s no need to use the Options tab unless you want a one-sided hypothesis or to change the confidence level from 95%. Press OK.

The R code for this test is:

```
with(forget, (t.test(accentr, accentl, alternative='two.sided',  
conf.level=.95, paired=TRUE)))
```

This test will give you the confidence interval of the difference between mean scores of the groups but not the mean scores of the groups, the standard deviations nor the number in each group. There are a number of ways to get this information but an easy way in R Commander is to choose STATISTICS > SUMMARIES > NUMERICAL SUMMARIES. Pick 2 of the variables at once (**accentr**, **accentl**) and then press OK for this command (here we are not dividing results into any groups). The R code for this is:

```
numSummary(forget[,c("accentl", "accentr")], statistics=c("mean", "sd",  
"IQR", "quantiles"), quantiles=c(0,.25,.5,.75,1))
```

To get correlation values between the pairs of variables, choose STATISTICS > SUMMARIES > CORRELATION MATRIX. Choose all of the variables and hit OK.

Results:

I used the cognitivelyflexibility.org/effectsize calculator and entered in the correlation between the paired variables to get the Cohen's d effect size given here. This effect size uses the average SD of the two variables.

Variable	95% CI	Mean (SD1)	N	Correlation (r)	Effect size
AccentR	-3.65, -2.70	7.20 (2.04)	44	.65	2.08
AccentL		10.37 (1.61)	44		

The 95% CI does not go through zero, and the difference in means between the groups lies, with 95% confidence, somewhere between 3.7 and 2.7 points. So our Japanese users of English produce a word beginning with /l/ with a more nativelike accent than a word beginning with /ɹ/ (since the mean score for ACCENTL is larger). The effect size is quite large.

5 Cole (1927)

Use the SPSS file Cole1927.sav, imported into R as **cole**. The results of this test will be different from the results from Exercise #5 in Section 8.2.7 because that question asked us to look at whether the *groups* were different from each other on the gains they made on the tests. This test, instead, looks at whether the same group differed in their scores from Time 1 to Time 2, or in other words, whether the groups made any gains over time. In Section 8.2.7 we calculated a gainscore by subtracting the Time 2 score from the Time 1 score, then looked at the difference in gainscores by comparing the two experimental groups to each other. Here we will keep the Time 1 and Time 2 score, but compare them *within* each group using the paired *t*-tests.

SPSS Instructions:

First, we need to split the file. Go to DATA > SPLIT FILE. Tick the radio button “Organize output by groups” and put the GROUP variable into the “Groups Based on” box. Press OK.

Go to ANALYZE > COMPARE MEANS > PAIRED-SAMPLES T TEST. Put the pairs of variables (TIME1_GRAMMAR_PERCENT: TIME2_GRAMMAR_PERCENT, TIME1_VOCAB_PERCENT: TIME2_VOCAB_PERCENT, TIME1_READING_PERCENT: TIME2_READING_PERCENT) in the “Paired Variables” box. Unless you have some unusual circumstances you would not need to use the “Options” button. Press OK.

R Instructions:

We need to split the file into different groups, but there are different ways to do this. I think the easiest would be to check which rows are for each group. In R Commander, just make sure `cole` is the active dataset in R Commander and open the “View dataset” button. Or use this code:

```
cole$group
```

The Composition group takes lines 1:15, while the Grammar Translation group uses lines 16:30.

Paired-samples t-test: First, let’s get the code for a paired-samples *t*-test, then add the lines.

Choose STATISTICS > MEANS > PAIRED T-TEST. You can only do this one pair at a time, so the first time choose `time1_grammar_percent` for the first variable under the heading “First variable (pick one)” and then `time2_grammar_percent` for the second variable under the

heading “Second variable (pick one).” There’s no need to use the Options tab unless you want a one-sided hypothesis or to change the confidence level from 95%. Press OK.

The code for this procedure is:

```
with(cole, (t.test(time1_grammar_percent, time2_grammar_percent,  
alternative='two.sided', conf.level=.95, paired=TRUE)))
```

Now take this code and add the lines to each variable, like this:

```
with(cole, (t.test(time1_grammar_percent[1:15], time2_grammar_percent[1:15],  
alternative='two.sided', conf.level=.95, paired=TRUE)))
```

Repeat with the other lines:

```
with(cole, (t.test(time1_grammar_percent[16:30], time2_grammar_percent[16:30],  
alternative='two.sided', conf.level=.95, paired=TRUE)))
```

Repeat with `time1_vocab_percent` and `time2_vocab_percent` and then
`time1_reading_percent` and `time2_reading_percent`, again splitting the file as above.

This test will give you the confidence interval of the difference between mean scores of the groups but not the mean scores of the groups, the standard deviations nor the number in each

group. There are a number of ways to get this information but an easy way in R Commander is to choose STATISTICS > SUMMARIES > NUMERICAL SUMMARIES. You can pick all 6 of the variables at once. We want to split by the group, so hit the button for “Summarize by groups” and choose Group. Press OK for this command and then OK to run the analysis. The R code for this is:

```
numSummary(cole[,c("time1_grammar_percent", "time1_reading_percent",  
"time1_vocab_percent", "time2_grammar_percent", "time2_reading_percent",  
"time2_vocab_percent")], groups=cole$group, statistics=c("mean", "sd",  
"IQR", "quantiles"), quantiles=c(0,.25,.5,.75,1))
```

To get correlation values between the pairs of variables, choose STATISTICS > SUMMARIES > CORRELATION MATRIX. Choose two variables and hit OK. Get the code and then we can split the file:

```
with(cole, cor.test(time1_reading_percent, time2_reading_percent,  
alternative="two.sided", method="pearson"))
```

Now run it again with the row numbers for each group, like this:

```
with(cole, cor.test(time1_reading_percent[1:15], time2_reading_percent[1:15],  
alternative="two.sided", method="pearson"))
```

You’ll need to repeat this with the different variables and different row numbers.

Robust paired samples t-test: All of the robust tests are found in Wilcox's WRS library:

```
library(WRS)
```

```
attach(cole)
```

```
trimci(grammar[1:15])
```

```
trimci(grammar[16:30])
```

Repeat the commands with **vocab** and **reading**.

This command gives 20% means trimmed CIs. The first entry should be the difference between the two values, and I already created that gainscore in Exercise #5 of Section 8.4.7. These CI are not bootstrapped. The next type of robust test we have is a bootstrap-t for trimmed means:

```
trimcibt(grammar[1:15])
```

```
trimcibt(grammar[16:30])
```

Repeat the commands with **vocab** and **reading**.

The last type described in my book (there are many more described in Wilcox, 2012, so see that book if you want more!) is the percentile bootstrap for trimmed means.

```
trimpb(grammar[1:15])
```

```
trimpb (grammar[16:30])
```

Repeat the commands with **vocab** and **reading**.

Results:

I used the cognitivelyflexibility.org/efficientsize calculator and entered in the correlation between the paired variables to get the Cohen's d effect size given here. This effect size uses the average SD of the two variables.

Variable	95% CI of the difference btw. means	Time 1 (SD1)	Time 2 (SD2)	N1/N2	Correlation (r)	Effect size
Grammar: Composition	-11.30, 0.10	81.33 (8.97)	86.93 (9.19)	15/ 15	.36	.55
Grammar: Translation	-9.10, -2.39	82.40 (5.82)	88.13 (3.82)	15/ 15	.27	.98
Vocab: Composition	-15.94, -10.19	69.96 (8.81)	83.02 (6.46)	15/ 15	.81	2.78
Vocab: Translation	-17.26, -9.76	64.89 (9.74)	78.40 (6.24)	15/ 15	.72	2.26
Reading: Composition	-9.17, 1.55	77.38 (8.39)	81.19 (6.95)	15/ 15	.21	.40
Reading: Translation	-8.73, -1.74	70.71 (9.94)	75.95 (8.59)	15/ 15	.78	.85

The CIs and effect sizes show that the area where the students gained the most over the semester was in vocabulary. The Translation group had CIs that didn't pass through zero and higher effect sizes for Grammar and Reading, while the Composition group had CIs that passed through zero and had smaller effect sizes than the Translation group, so it seems that the Translation group, who worked more with reading and grammar, somewhat naturally did better and improved more in reading and grammar. The Composition group did have a higher effect size for Vocabulary than the Translation group, and its 95% CI was narrower (more precise) than in the Translation

group, although we saw from Exercise #5 in Section 8.2.7 that there was no statistical difference in gainscores between groups. For both of the cases where the CI passed through zero and we would say that there was no effect for the passage of time (For Grammar and Reading for the Composition group), the CI is not symmetrical around zero and we might suspect that with more subjects the CI would narrow down and not pass through zero. Below, in the results for the robust tests, the bootstrapped 95% CI (results from SPSS) and the percentile bootstrap for trimmed means do show a CI that does not pass through zero for Grammar and Translation for the Composition group, indicating that for the main bulk of the participants (the non-trimmed middle) there was improvement from Time 1 to Time 2, although the CI still remains close to zero meaning the actual effect was rather small.

Variable	Bootstrapped 95% CI of the difference btw. means (SPSS)	20% trimmed means CI	Bootstrap-t CI with trimmed means	Percentile bootstrap for trimmed means
Grammar: Composition	-10.67, -.80	-0.24, 9.57	-0.02, 9.35	0.22, 10.44
Grammar: Translation	-8.67, -2.53	2.94, 9.95	2.38, 10.51	2.67, 9.11
Vocab: Composition	-15.64, -10.31	9.55, 17.12	9.27, 17.39	10.22, 16.44
Vocab: Translation	-17.33, -9.87	9.30, 15.89	9.44, 15.75	9.48, 16.74
Reading: Composition	-8.57, 1.19	0.84, 7.10	0.57, 7.36	1.19, 7.54
Reading: Translation	-8.10, -2.62	-0.37, 8.30	-1.67, 9.61	1.59, 8.33