Chapter 10

One-Way ANOVA

10.1 Numerical and Visual Inspection of the Data, Including Boxplots Overlaid with Dotcharts

To get a sense first of what is happening with the Ellis and Yuan (2004) data, let's take a look at a numerical summary of the dependent variable of syntactic variety. In R:

```
numSummary(ellisyuan$SyntaxVariety , groups=ellisyuan$Group,
statistics=c("mean", "sd"))
```

Table 10.1 in the SPSS book (*A Guide to Doing Statistics in Second Language Research Using SPSS*, p. 272) shows the means and standard deviations for each of Ellis and Yuan's three groups.

Turning now to visual examination of the data, boxplots would be good for examining whether the distribution of the data was normal and whether variances look similar. The chapter on t-tests has already explained how to call for boxplots, but the next section provides a specialized boxplot that is harder to create in R, but may be a visual that you think is worth your time creating.

10.1.1 Boxplots with Overlaid Dotcharts

A boxplot overlaid with a dotchart will plot individual data points over the group variables. I like this type of graph because all of the data are visible, even in a situation where you want the grouping of the data to be apparent as well. This type of chart can be useful for seeing individual patterns of responses that are hidden within the grouping of the boxplot. This plot uses data from Ellis and Yuan (2004).

© Taylor & Francis

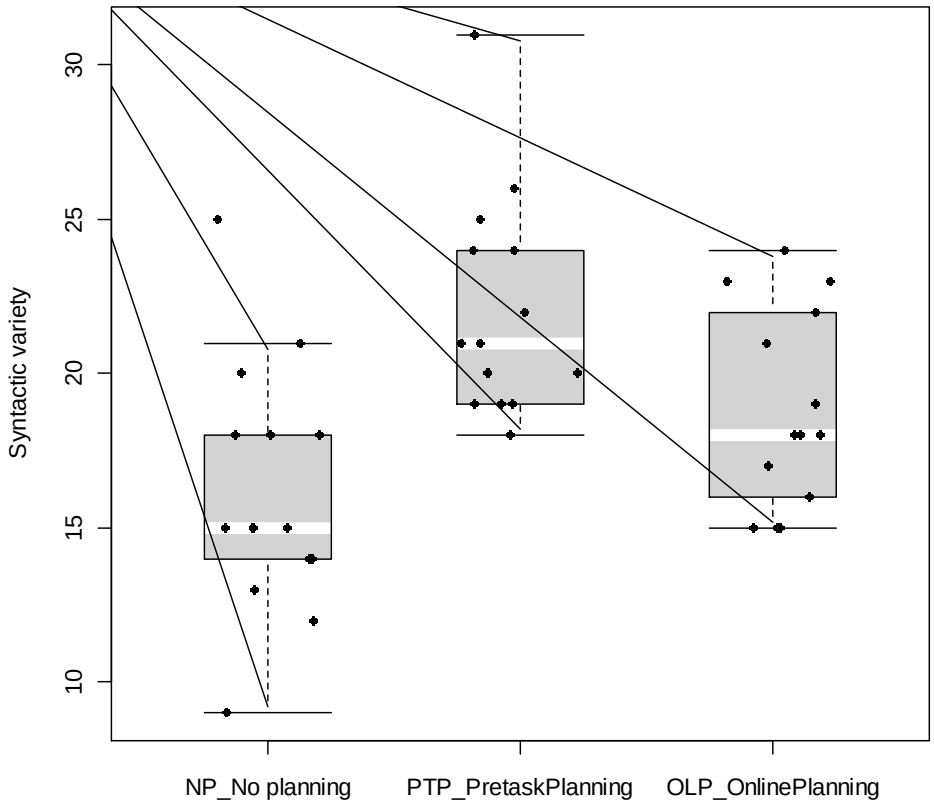


Figure 10.1 Boxplots with overlaid dotcharts of the Ellis and Yuan (2004) data.

The boxplots in Figure 10.1 show that the pre-task planning group (PTP) was able to produce the most syntactic variety, as it has the highest mean. The boxplot also labels the highest point in the NP group as an outlier, while the highest point in the PTP group is not considered an outlier. No attention should be give to the horizontal spread of the points on the boxplot, as this is arbitrary.

The command for creating this graphic is somewhat complicated. However, with the explanations given in the boxes below, it should be possible to adapt the command to different numbers of groups, and groups whose sizes are unequal (the group sizes in Ellis and Yuan are equal).

Step 1: bx.p<-boxplot(SyntaxVariety~Group, data=ellisyuan)

bx.p<-boxplot ()	bx.p is a name I created to name my object; I'm forcing the results of the boxplot into a

© Taylor & Francis

	dummy variable which I can then further modify.
boxplot(SyntaxVariety~Group, data=ellisyuan)	This is the normal syntax for a boxplot.

Step 2: with (ellisyuan,
{bxp (bx.p, staplewex=1, boxfill="light grey", medlwd=8, medcol="white", boxwex=.5, ylab="Syntactic variety", outl=F)
points (jitter (rep (1:3, each=14), 1.2),
unlist (split (SyntaxVariety, Group)),
cex=.8, pch=16)
}
)

with(data, expression, . .)	Evaluates an R expression in an environment constructed from data.
{ . .. }	The curly braces are used if we will execute more than one command inside the with () command.
bxp()	Draws boxplots from a list of summaries (the shaded commands are arguments to the bxp command).
bx.p	Our boxplot object.
staplewex=1	Adjusts the length of the line that marks the extreme ends of the whiskers; this command makes those lines a little longer than normal so the dots look proportional.
boxfill="lightgrey"	Specifies the color inside the box.
medlwd=8	Specifies the width of the median line.
medcol="white"	Specifies the color of the median line.
boxwex=.5	Adjusts the width of the box (makes the results look more professional, in my opinion).
ylab=" "	Gives a name to the y-axis.
outl=F	Tells the bxp command not to print outliers (we don't want them printed twice).
points (x, y, . . .)	Draws points at specified coordinate vectors x and y.
jitter (x, 1.2)	Adds noise to numbers so they will not cluster on top of one another; the number "1.2" gives the amount the dots should be spread out from each other.
rep(1:3, each=14)	Replicates the specified sequence (here, from 1 to 3) the specified number of times (here, 14 times each).[1]

[1] Murrell (2006) says that this works because "the *i*th box is located at position *i* on the x-axis." In other words, the points command plots the dots at position 1, then 2, and then 3, covering in turn boxplot 1, boxplot 2, and boxplot 3. To understand how to alter this command, consider this.

The command rep (1:3, each=4) would create the following output:

1 1 1 1 2 2 2 2 3 3 3 3

Deleted: .

© Taylor & Francis

unlist (split (SyntaxVariety, Group)	Takes a list and simplifies it into a vector; the split command divides the SyntaxVariety data into groups defined by the Group factor; this command is needed for getting the dots into the correct place.
cex=.8	Specifies the size of the overlaid dots.
pch=16	Specifies the plotting character (here, dots).

If you need to alter some part of the graphic, you would need to choose the correct place to alter it. For example, if you wanted to limit the y-axis of the boxplot to the range 1–5, the command ylim=c(1,5) would be inserted somewhere in the bxp command, after the data call (the bx.p in this case). If you wanted to change the size or type of plotting character of the dots, this needs to be done within the points command.

Note: If you are copying code directly from my documents and get an error message about an unexpected ')' in ") " then the problem might be the quotation marks! My MS Word document replaces straight quotation marks with "smart" quotation marks, and R doesn't like these!

10.2 Application Activities for Boxplots with Overlaid Dotcharts

1. Ellis and Yuan (2004). Import the EllisYuan.sav data set as ellisyuan and run the boxplot with overlaid dotcharts command by inserting the commands from Step 1 and Step 2 of the "One way ANOVA.Visual summary with boxplots overlaid with dotcharts" online document into the R Console. Once you have verified you can get the same figure as in this document, change the plotting character for the points to pch=15. What character do you get?

2. Make a boxplot with an overlaid dotchart for the disfluencies measure (Disfluencies) in the Ellis and Yuan data, make the box color white and the median line black, and make the boxplots notched (notch=T; this is an argument of the bxp command).

3. Practice changing the numbers of participants. Make a boxplot with an overlaid dotchart for the Leow and Morgan-Short data (leow), using the receptive task post-test data (RecPostScore). There are 38 participants in the think-aloud group and 39 in the non-think-aloud group. Make the box blue and the median line red. There are 20 points possible on this test, so make sure the y-axis goes from 0 to 20.

10.3 One-Way ANOVA Test

I will illustrate how to perform a one-way ANOVA using Ellis and Yuan's (2004) variable of SyntaxVariety, which was explored graphically at the beginning of Chapter 10 of the SPSS book (*A Guide to Doing Statistics in Second Language Research Using SPSS*). The research question we are interested in is whether the groups with differing amounts of planning time

---

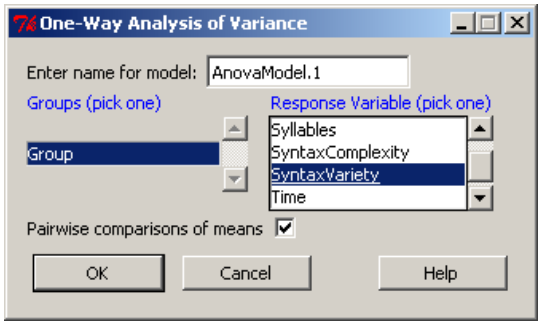
If you have two groups with unequal group sizes, this command should be altered. Say you had 54 total participants, 22 in group A, 15 in group B, and 17 in group C. The following command will distribute points correctly:

```
rep(c(1:3), c(22, 15,17))
```

This will repeat the 111 . . .222 . . . 333 sequence the specified number of times (first 22 times, then 15 times, and then 17 times).

© Taylor & Francis

are statistically different in the amount of syntactic variety they produce. The independent variable is experimental group. To answer this question using R Commander, pull down the menu sequence STATISTICS > MEANS > ONE-WAY ANOVA as shown in Figure 10.2.



Pick your IV from the "Groups" box (this must be a character variable in R) and your DV from the "Response Variable" box. Tick the box for "Pairwise comparison of means."

Figure 10.2 How to get a one-way ANOVA in R.

Although the dialogue box is simple, the output produced by this command is quite extensive! The first piece of information returned is the omnibus F test.

```
            Df Sum Sq Mean Sq F value    Pr(>F)
Group        2  240.9 120.452  9.0513 0.0005901 ***
Residuals   39  519.0  13.308
```

This output shows that there is a statistical omnibus effect, so there is some difference between the groups, with $F_{2,39}$=9.05, p=.0006. The hypothesis df is 2 (the number of groups minus 1), and the error df is 39, and both of these dfs must be reported. Even though the p-value is small, remember that this doesn't mean that there is a 6 in 10,000 chance that the result happened by chance. What it really means is that, *if* the null hypothesis were true (that all the groups have equal means), the probability that you would get this particular result is very small. So we reject the null hypothesis that the groups are all the same.

Technically, if your omnibus ANOVA is not statistical then you should stop your analysis and not continue with post-hocs. Since you have called for post-hocs before seeing the omnibus test, however, they will be provided, but just ignore them if the omnibus is not statistical.

The next pieces of output give vital descriptive statistics for the groups—means, standard deviations, and the counts.

```
                          mean       sd  n
NP_No planning        16.21429 4.098378 14
PTP_PretaskPlanning   22.07143 3.583387 14
OLP_OnlinePlanning    18.85714 3.207135 14
```

We see from the means that the group who got the most planning time (PTP) produced the most syntactic variety (22.07).

The next piece of output performs the post-hoc tests to find out exactly which of the groups are different from the others. Howell (2002) recommends the LSD test as the most powerful post-hoc test to find differences if you have only three means (in R, use the choice "none" for

© Taylor & Francis

this). If you have more than three, both Howell (2002) and Maxwell and Delaney (2004) recommend Bonferroni or Tukey's post-hocs. Bonferroni has more power to find differences if fewer tests are performed, while Tukey's has more power if more tests are performed. Both Tukey's and Bonferroni are conservative tests and strictly control the familywise error rate. If you would like more power to find differences, Maxwell and Delaney (2004) recommend Benjamini and Hochberg's FDR (available in R). If variances are not equal, Howell (2002) recommends Games–Howell, and Maxwell and Delaney (2004) like it too (but this is not available in the R packages I am using). Note that the choice of post-hocs can have a clear effect on results, so I recommend that, if you choose to control the familywise error rate, you should choose the tests with the highest power (the FDR).

```
                                              Estimate lwr     upr
PTP_PretaskPlanning - NP_No planning == 0      5.8571   2.4971  9.2172
OLP_OnlinePlanning - NP_No planning == 0       2.6429  -0.7172  6.0029
OLP_OnlinePlanning - PTP_PretaskPlanning == 0 -3.2143  -6.5743  0.1457
```

This piece of output first shows the mean difference between the two groups (under "Estimate"), and then the lower and upper cut-off points of a 95% confidence interval. Just as with all of the other confidence intervals we have seen, the test will be statistical if it *does not* pass through zero. This is because, if zero is in the interval, that means there is a possibility that the real difference between groups is zero. So, to summarize, we find a statistical difference between the PTP and NP group, but not between OLP and NP or OLP and PTP. The last part of the output from R Commander produces a graphic that plots these three confidence intervals for the mean difference between groups, which shows graphically what we find from the numerical output above (I do not show this because I will show you a better graphic later in the chapter).

The R code for performing a one-way ANOVA is actually the same type of syntax that is used for performing regression (except that we used the lm() command for regression in the previous chapter while we use the aov() command in this chapter). ANOVA is simply a special form of the linear regression model, and this is reflected in the syntax which R uses to perform the one-way ANOVA (Miles and Shevlin, 2001 is a fairly accessible book which explains how regression and ANOVA are related). This is made explicit in the code that R Commander uses for the ANOVA:

AnovaModel.1 <- aov(SyntaxVariety ~ Group, data=ellisyuan)	
aov()	The aov command creates an analysis of variance model. It actually uses the lm() command seen in Chapter 7, but differs in that the output from the summary() and print() commands are more suitable for an ANOVA.
AnovaModel.1=aov()	Put the aov model into an object which R Commander labeled AnovaModel.1 automatically (we can change this name, of course).
aov(SyntaxVariety ~ Group)	The model says that the scores on the Syntactic variety variable (SyntaxVariety) are modeled as a function of Group.
data=ellisyuan	Specifies what data set should be used.

So the first command is to create a regression model (an ANOVA model) and put it into an object. But, in order to see the ANOVA table and find the omnibus F test, one will need to

© Taylor & Francis

call for a summary of the regression model. Thus, this is the next step of the syntax that R Commander uses:

```
summary(AnovaModel.1)
```

You have seen the results of this call in the ANOVA table above. If you would like to change the type of sums of squares used by the test, you can change the call for the summary to the Anova() command:

```
Anova(AnovaModel.1, type=c("II"))
```

We don't see any changes in results with any of the types of sums of squares, however, because the Ellis and Yuan (2004) data are equal for each group (a balanced design).

Another possible command for the ANOVA omnibus test is a oneway.test():

```
oneway.test(SyntaxVariety~Group,data=ellisyuan, var.equal=T)
```

```
        One-way analysis of means

data:  SyntaxVariety and Group
F = 9.0513, num df = 2, denom df = 39, p-value = 0.0005901
```

Notice that the syntax is equivalent to aov(), with the addition of the argument about whether variances should be considered equal or not (this can be set to TRUE or FALSE). With this command the entire ANOVA table is not returned, just the F-value, degrees of freedom, and *p*-value.

For Levene's test for homogeneity of variances (with the usual caveat that this test may not have enough power to be useful), use the command:

```
levene.test(ellisyuan$SyntaxVariety, ellisyuan$Group)
```

```
Levene's Test for Homogeneity of Variance
      Df F value Pr(>F)
group  2  0.1639 0.8494
      39
```

For the Ellis and Yuan (2004) data, Levene's test shows that there is no reason to reject the assumption that the variances of the groups are homogeneous (reject this assumption only if $p<.05$). Previous inspection of graphs and numerical summaries has also shown us that there does not seem to be a great difference in variances between the groups.

To obtain means, standard deviations, and counts for each group, R Commander uses the numSummary() command that we have seen in previous chapters:

```
numSummary(ellisyuan$SyntaxVariety , groups=ellisyuan$Group,
statistics=c("mean", "sd"))
```

To call post-hoc tests, R Commander uses the glht() command from the multcomp package. This command sets up multiple comparisons, but it provides more detail if the result is placed into an object and then a summary and confidence intervals can be called.

© Taylor & Francis

```
library(multcomp)
.Pairs <- glht(AnovaModel.1, linfct = mcp(Group = "Tukey"))
confint(.Pairs) # confidence intervals
plot(confint(.Pairs))
```

glht(AnovaModel.1)	The general linear hypothesis test (glht) needs a model, and we specified this previously.
linfct=	Specifies what linear hypothesis should be tested.
mcp	Multiple comparisons are returned by using this argument.
(Group="Tukey")	Tells mcp to use Tukey post-hoc comparisons using the Group variable divisions.

The data that is returned by simply performing the glht() command is not nearly as much as can be returned by using the confint() command and summary() command. The confint command will return confidence intervals, as shown in the R Commander output above for the pairwise comparisons. In addition, the summary() command, which you can add if you like, will return *t*-values and *p*-values for the pairwise comparisons:

```
         Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Tukey Contrasts


Fit: aov(formula = SyntaxVariety ~ Group, data = ellisyuan)

Linear Hypotheses:
                                            Estimate Std. Error t value Pr(>|t|)
PTP_PretaskPlanning - NP_No planning == 0      5.857      1.379   4.248 0.000361 ***
OLP_OnlinePlanning - NP_No planning == 0       2.643      1.379   1.917 0.147408
OLP_OnlinePlanning - PTP_PretaskPlanning == 0 -3.214      1.379  -2.331 0.063124 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Adjusted p values reported -- single-step method)
```

What if you don't want to use the Tukey adjustment, and instead would like to use a different type of adjustment for multiple *p*-values? Just add an additional argument to the summary() command:

```
summary(.Pairs,adjusted(type=c(p.adjust.methods=c("fdr"))))
```

Here I have used the FDR method; the choices of adjustment types available in p.adjust.methods arguments are: "bonferroni", "holm", "hochberg", "hommel", "BH", "BY", "fdr", and "none".

The last piece of data that R Commander calls for is a plot of the confidence intervals for the pairwise comparisons. R Commander uses this code:

```
plot(confint(.Pairs))
```

However, an MMC (mean–mean multiple comparisons plot) that I like even better can be found in the HH library. Create a new object using the glht.mmc() command, and then plot it.

```
library(HH)
```

© Taylor & Francis

```
ellisyuan.mmc=glht.mmc(AnovaModel.1,linfct=mcp(Group="Tukey"))
plot(ellisyuan.mmc)
```

The plot not only contains the confidence intervals of the comparisons which correspond to the actual width of the interval if one consults the contrast value along the x-axis, but also shows means of each group along the left-hand y-axis.

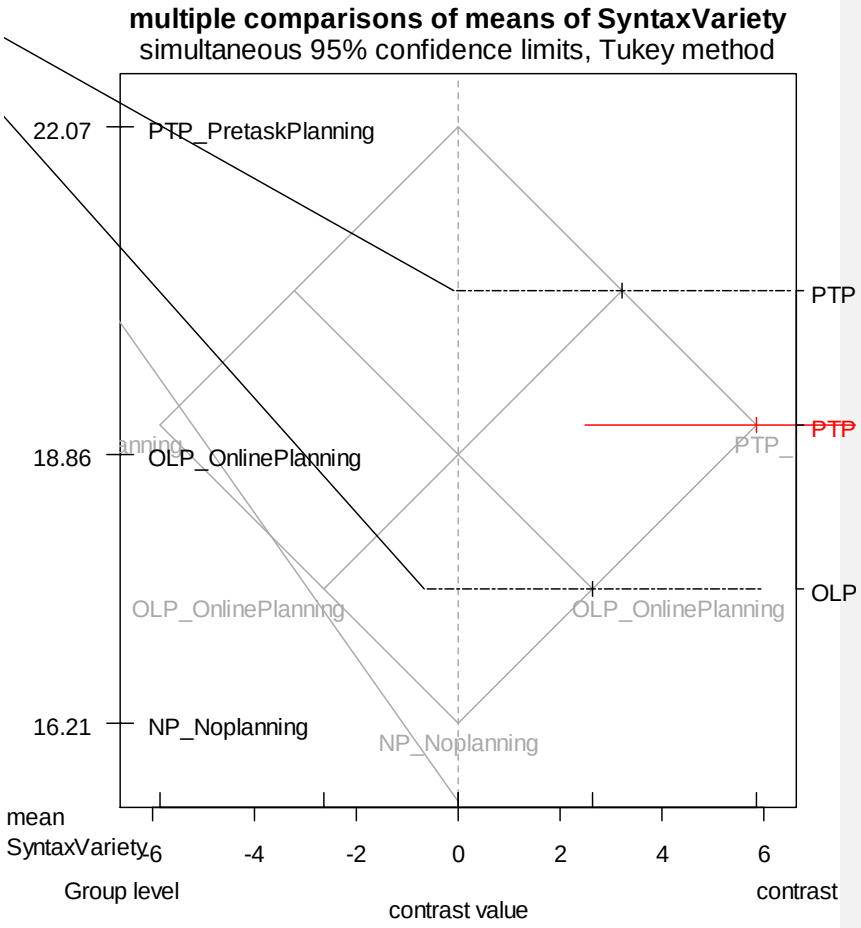


Figure 10.3 MMC (multiple comparison of means) plot with Ellis and Yuan (2004) data.

The line type and color of statistical comparisons differ from non-statistical comparisons as well on this plot. The solid line in the middle is red and indicates a statistical comparison, while the dotted black lines go through zero and indicate non-statistical comparisons. The height of the comparison is also plotted along the average mean of the two groups. You can see the mean scores on the left-hand y-axis, so, for example, the average mean of the NP vs. OLP group is between 16.21 and 18.86. The grey triangles that are constructed in the

© Taylor & Francis

background also reflect the relative distances of the means, although because they are fairly equidistant in this case it is not readily apparent (Heiberger & Holland, 2004).

Conducting a One-Way ANOVA Using R

In R Commander, choose STATISTICS > MEANS > ONE-WAY ANOVA. Choose a Group variable (=independent variable) and Response variable (=dependent variable). Tick the box for pairwise comparison of means. Change the name of the regression model if you like.

I have suggested the following R code to obtain the same results as R Commander, but sometimes with an improvement:

```
AnovaModel.1= aov(SyntaxVariety ~ Group, data=ellisyuan) #create model
Anova(AnovaModel.1, type=c("II")) #omnibus test
levene.test(ellisyuan$SyntaxVariety, ellisyuan$Group) #test homogeneity of variances
numSummary(ellisyuan [,"SyntaxVariety "], groups = ellisyuan$Group)
library(multcomp)
.Pairs=glht(AnovaModel.1,linfct=mcp(Group="Tukey")) #model for multiple comparisons
confint(.Pairs) #returns confidence intervals for comparisons
summary(.Pairs, adjusted(type=c(p.adjust.methods=c("fdr")))) #returns p-values for comparisons, gives choice of p-value adjustment methods
library(HH)
ellisyuan.mmc=glht.mmc(AnovaModel.1,linfct=mcp(Group="Tukey")) #model for MMC
plot(ellisyuan.mmc) #MMC plot
```

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

10.3.1 Conducting a One-Way ANOVA Using Planned Comparisons

If you would like to perform the post-hoc tests without the omnibus F-test (as recommended by Wilcox, 2003 as a way of increasing the statistical power of the test), this section will explain how to do it. You could also, of course, just use a series of t-tests comparing the pairs of groups that you are interested in. However, if you did this you might be concerned about inflating the error rate by having multiple tests (this is not something that bothers me too much, however, as I stand on the side of higher power). Howell (2002) opines that if you have only one or two comparisons to make then the t-test approach is not unreasonable, but if you have a really large number of comparisons then using planned comparisons is better.

To illustrate this procedure we will look again at the comparison among the three groups of the Ellis and Yuan (2004) data for the SyntacticVariety variable without doing the omnibus F-test. Suppose that we wanted to test all of the possible comparisons among the groups. In this case, there are three groups, so there will be three possible comparisons. In each comparison, we want to ignore one of the groups and compare the other two. Study Table 10.1 to examine the pattern of coefficients that need to be used in order to do an ordinary comparison of the three groups.

Table 10.1 Planned Comparison Coefficients for Two-Way Comparisons

Groups	*Compare Group 1 to Group 2*	*Compare Group 1 to Group 3*	*Compare Group 2 to Group 3*

© Taylor & Francis

NP=Group 1	1	1	0
PTP=Group 2	−1	0	1
OLP=Group 3	0	−1	−1

From Table 10.1 you can see that you put a number in the row of the groups you want to compare. The number could be 1, 2, or 1,000, but there needs to be an equally large number to be subtracted so that, when the columns are added up, they add up to zero. Any row of the column which contains a non-zero number will be included in the comparison. For the planned comparison shown in Table 10.1, only two groups at a time are being compared. All possible pairings are made (that is, NP versus PTP, NP versus OLP, and PTP versus OLP).

Creating your own planned comparisons in R is straightforward using the glht() command seen in section 10.3 and used to conduct post-hoc multiple comparisons. To set up planned comparisons like those in Table 10.1, simply specify your contrasts by referring to the levels of your categorical variable. For example, for the Groups variable in Ellis and Yuan (2004), the names of the levels can be obtained with the levels command:

```
levels(ellisyuan$Group)
[1] "NP_No planning" "PTP_PretaskPlanning" "OLP_OnlinePlanning"
```

These names are a little complicated! I will simplify them:

```
levels(ellisyuan$Group)=c("NP", "PTP", "OLP")
```

Now I can specify the coefficients for the comparisons I want to make in an object I'll call contr (look back to the columns of Table 10.1 to understand the numbers I use). The numeric coefficients refer to the inclusion of the levels in the order they are found in R in the levels() command you just looked at.

```
contr=rbind("NP-PTP"=c(1,-1,0),
            "NP-OLP"=c(1,0,-1),
            "PTP-OLP"=c(0,1,-1))
```

Now I run the glht() command again, specifying that the testing for Group will be the object I just created. I can then run summary() and confint() on this object.

```
Ellisyuan.Pairs=glht(AnovaModel.1,linfct=mcp(Group=contr))
```

The results will be exactly the same as in section 10.3, since I am specifying the same contrasts (they are reversed in sign, but this does not make any difference ultimately). To make planned comparisons with more than three groups, or only among certain groups, the idea is the same, but the number used in the comparison needs to be adjusted so that each column adds up to zero.

So let's say what we really wanted to know with the Ellis and Yuan (2004) data was whether having planning time made a difference or not, and whether having unlimited time made a difference or not. For the question of planning time, we could compare the group with planning time (PTP) to the two others without planning time (NP and OLP). For the question of unlimited writing time, we could compare the group with unlimited time (OLP) to the groups who had only 17 minutes of writing time (NP and PTP). Study the coefficients in Table 10.2 to see how we could call for these comparisons.

© Taylor & Francis

Table 10.2 Planned Comparison Coefficients for Multiple Comparisons with Elis and Yuan (2004)

Groups	*Compare PTP to NP and OLP*	*Compare OLP to NP and PTP*
NP=Group 1	−1	−1
PTP=Group 2	2	−1
OLP=Group 3	−1	2

Table 10.3 shows coefficients that could be used in a study with four groups and various research questions.

Table 10.3 Planned Comparison Coefficients for Multiple Comparisons with L1 Groups

Groups	*Compare Latinate Groups to NS*	*Compare Japanese Group to NS*	*Compare all NNS groups to NS*
Native speakers	2	1	3
Spanish L1	−1	0	−1
Portuguese L1	−1	0	−1
Japanese L1	0	−1	−1

Creating Planned Comparisons in R

1. Specify the coefficients for the comparisons you want to make. You'll need to use the names of the levels of the independent variable, so find out the names and change them first if they are rather complicated:

```
levels(ellisyuan$Group) #find out names
levels(ellisyuan$Group)=c("NP", "PTP", "OLP") #set new names
```

2. rbind() the coefficients together into a matrix. It would probably help to first write the coefficients down in a table like the ones I have shown in the text. List the coefficients in the order they are found in the levels() command. Make sure the column adds up to zero. Create a new object with the rbind() command.

```
contr=rbind("NP-PTP"=c(1,-1,0),
"NP-OLP"=c(1,0,-1),
"PTP-OLP"=c(0,1,-1))
```

3. Run the glht() command again, using the new object as the model for the independent variable:

```
Ellisyuan.Pairs=glht(AnovaModel.1,linfct=mcp(Group=contr))
```

4. You may now obtain the statistics from this model with summary() and confint().

10.4 Performing a Robust One-Way ANOVA Test

© Taylor & Francis

Using the same Ellis and Yuan (2004) data used in this chapter for a one-way ANOVA, let's look at another variable called Words. This is a variable that gives a count of the number of words the writers in the various experimental conditions produced. A boxplot of this variable in Figure 10.4 shows one extreme outlier in the OLP condition, but otherwise fairly symmetric distributions with equal variances.

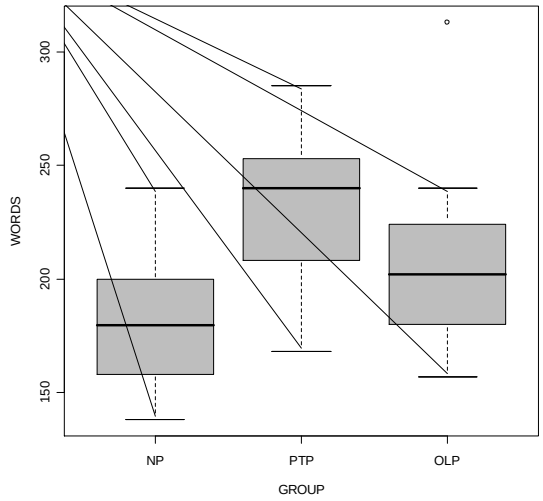


Figure 10.4 Boxplots of Ellis and Yuan (2004) data for Words.

Pairwise comparisons using Tukey's adjustment find only the comparison between PTP and NP statistical:

```
ellisyuan.anova <- aov(Words ~ Group, data=ellisyuan)
ellisyuan.Pairs <- glht(ellisyuan.anova, linfct = mcp(Group = "Tukey"))
summary(ellisyuan.Pairs)
```

```
Fit: aov(formula = Words ~ Group, data = ellisyuan)

Linear Hypotheses:
               Estimate Std. Error t value Pr(>|t|)
PTP - NP == 0     49.79      12.73   3.912  0.00101 **
OLP - NP == 0     25.57      12.73   2.009  0.12327
OLP - PTP == 0   -24.21      12.73  -1.903  0.15138
```

Now I'll reanalyze this variable using Wilcox's trimmed means bootstrap-t function t1waybt().

To use this you will need to gain access to Rand Wilcox's R library. At the moment it cannot be retrieved from the drop-down list method in R Console. Instead, type this line directly into R:

© Taylor & Francis

```
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)
```

Before using Wilcox's function the data will need to be massaged into the correct form, which will involve a list where each part of the list is a separate vector for each group's data on the dependent variable. Note that for the section on planned comparisons (found in the online document "One way ANOVA.One-way ANOVA test") I changed the names of the levels of this group, and I use those here (if you import it afresh from the SPSS file the names of the levels of the groups will be different!).

```
ellis=list() #creates a list called ellis
ellis[[1]]=subset(ellisyuan,subset=Group=="NP",select=c(Words))
ellis[[2]] <- subset(ellisyuan, subset= Group =="PTP", select=c(Words))
ellis[[3]] <- subset(ellisyuan, subset= Group =="OLP", select=c(Words))
```

The last three lines enter one subset each of the Words variable into a group of data in the list. For example, ellis[[1]] references the first group of data in the list. The subset function has three arguments: 1) the data; 2) the specification of the subset, using the name of your group variable (mine is Group); 3) the variable you want to subset. Actually, I used R Commander to subset the first group in order to remember the syntax for subsetting. To use R Commander, choose DATA > ACTIVE DATA SET > SUBSET ACTIVE DATA SET. The data is now ready to use in the t1waybt() function.

```
t1waybt(ellis, tr=.2, nboot=599)
```

t1waybt()	Performs a bootstrap-t (parametric bootstrap) comparison of trimmed means.
ellis	The first argument is the data, which needs to be in list or matrix form.
tr=.2	Default is that 20% of the means should be trimmed.
nboot=599	Specifies the number of bootstraps to perform, with 599 as the default.

```
$test
[1] 10.35645

$p.value
[1] 0.005008347
```

The output returned by this test is the F-value (in $test) and the *p*-value of that F-test (*p*=.005). Remember, this test is just giving the omnibus F-test, which was also statistical in the non-robust case. The output does not return degrees of freedom, since these are not associated with bootstrapping procedures. In some cases you may not care even to check the omnibus test, but instead just proceed directly to multiple comparisons.

In order to perform multiple comparisons to determine which groups are different from each other, you can use the mcppb20() command. It performs multiple comparisons using a

© Taylor & Francis

percentile bootstrap method for comparing 20% trimmed means. Two other commands you might be interested in for multiple comparisons are lincon(), which uses trimmed means only, or linconb(), which uses trimmed means plus the bootstrap-t. Wilcox says that the bootstrap-t is relatively effective with small sample sizes, but, when samples sizes are all 15 or below, power can drop. The basic syntax of these commands is the same as mcppb20, so I will not repeat it.

```
mcppb20(ellis)
$psihat
     con.num psihat       se ci.lower ci.upper p-value
[1,]       1  -53.7 11.60203    -80.8    -24.3  0.0000
[2,]       2  -22.6 12.03546    -52.9      7.5  0.0775
[3,]       3   31.1 12.37243     -0.1     59.9  0.0180

$crit.p.value
[1] 0.017

$con
     [,1] [,2] [,3]
[1,]    1    1    0
[2,]   -1    0    1
[3,]    0   -1   -1
```

First, remember that group1=NP, group2=PTP, and group3=OLP (that's the way the list is set up). The information under $con specifies the contrasts, and this is read in columns, so that the first contrast is comparing groups 1 and 2, the second groups 1 and 3, and the third groups 2 and 3 (the places where the 1s are). Under $psihat, we see that the difference between groups 1 and 2 is 53.7 words, with a standard error of 11.6. The confidence interval for this difference could be as large as 80.8 words or as small as 24.3 words, but this interval does not pass through zero, so we can conclude, with 95% confidence, that there is a real difference between these groups. Although the other comparisons are not statistical (the third comparison between PTP and OLP has a *p*-value under .05, but the critical value for being statistical is α=.017 in the $crit.p.value), notice that their confidence intervals are shorter with the robust analysis than they were in the non-robust analysis ([−.1,59.9] for PTP_OLP in the robust measure, [−55.2, 6.79] in the non-robust measure).

Although I used just the bare minimum for the mcppb20() command, below I elaborate on the structure of this command:

mcppb20(ellis, crit=NA, con=0, tr=.2, alpha=.05, nboot=2000, grp=NA)	
mcppb20(ellis)	The data for this command need to be in list or matrix form.
crit=NA	Specifies the critical value (the cut-off point) for p; for α=.05, critical value=$2p_{crit}$. For $\alpha \neq .05$, and if crit=NA, α/C (where C=# of hypotheses to be tested) is used (basically, the Bonferroni method).
con=0	Space for specifying planned comparisons. Do this by creating vectors, such as one=c(1,−2,1) and then using cbind() to pull together all your comparisons: con=cbind(one, two, three)
tr=.2	Default is that 20% of the means should be trimmed.
alpha=.05	Sets alpha level; default is .05.
nboot=2000	Specifies number of bootstraps; default is 2,000.

© Taylor & Francis

grp=NA	Specifies which groups should be included; by default all are.

The results show that, although a robust analysis did not change the final outcome with the Ellis and Yuan (2004) variable of Words using a strict cut-off point of α=.05, confidence intervals were smaller and presumably more accurate for those who would use it to compare results of future testing.

Robust One-Way ANOVAs

1. Perform a robust omnibus one-way ANOVA test with this command for a trimmed-means bootstrap-t function:

```
t1waybt(ellis, tr=.2, nboot=599)
```

2. Perform robust post-hoc multiple comparisons with this command for a trimmed-means with percentile bootstrap:

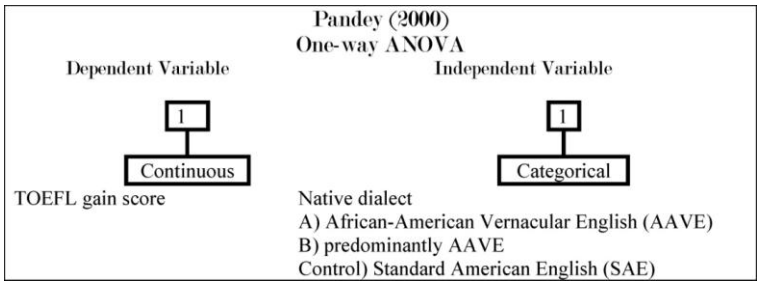
```
mcppb20(ellis, crit=NA, con=0, tr=.2, alpha=.05, nboot=2000, grp=NA)
```

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 probably the confidence intervals for the mean differences between the groups.

10.5 Application Activities for One-Way ANOVAs

1. Using the Ellis and Yuan data (EllisYuan.sav, imported as ellisyuan), look at the variables of error-free clauses, MSTTR (a measure of lexical variety), and speed per minute (SPM). First examine the data visually. What do you find with regard to normality, homogeneity of variances, and outliers? Next, perform a one-way ANOVA on each of the three variables along with post-hocs. Report omnibus F-results; find 95% confidence intervals for the difference in means, and calculate effect sizes for the contrasts. What would you remark upon here?

2. Pandey (2000) conducted research on how well native speakers of different English dialects performed on the TOEFL and reported the raw data scores in her article. Pandey's research design is shown in the following diagram:



Pandey did not perform statistics on her data, but merely observed the trend that both focus groups had quite low TOEFL scores the first time they were tested (mean for Focus Group

© Taylor & Francis

A—443; mean for Focus Group B—429), but, with comparative instruction, both focus groups improved their scores over time. In comparison, the control groups' scores hardly changed at all the second time they took the test. Using the Pandey2000.sav file (import as pandey, investigate statistically the research question of whether Groups A and B are different from each other in their first gain scores (GAIN1), and also whether Group A and B *together* are different from the control group in the first gain scores. You'll need to use planned comparisons to answer this question. Be sure to first visually examine the data (even if it does not fit assumptions, go ahead and analyze it here).

3. Thought question. Imagine a research design that investigates the pragmatic development of Chinese learners of Russian. The researcher asked participants to complete discourse completion tasks at three separate times during the semester. The participants were in three different classes, and each class received each of the three treatments (Control—no pragmatic discussion; explicit pragmatic discussion; and implicit pragmatic discussion), with five weeks between each treatment. Immediately following the treatment, participants took the discourse completion task. The researcher wants to know whether the groups differed depending on their treatment, and plans to conduct a one-way ANOVA with GROUP as the IV and score on the discourse completion task at every time period as the DV, as is shown in the table below. What is wrong with this research design?

	Participant	Group	Time	DCTScore
1	ChiWei	1	1	68
2	ChiWei	1	2	49
3	ChiWei	1	3	87
4	TsiWen	1	1	53
5	TsiWen	1	2	56

4. Inagaki and Long (1999) conducted an experiment with Japanese learners to see whether asking participants to repeat a correct model after they heard it would be as effective as recasting after an incorrect utterance (and whether these would differ from a control condition where participants simply studied *kanji* characters). Two structures, that of correct adjective formation when there are multiple adjectives, and subject placement after a locative phrase, were examined with 24 participants. If possible, conduct two separate one-way ANOVAs to determine whether there was any difference between the three experimental groups on the gain score of the two structures ("GainAdj" and "GainLoc"). The group variable in each case is also different ("AdjModelOrRecast" for the adjective formation, and "LocModelOrRecast" for the locative formation). Be sure to first visually examine the data. Use the InagakiLong.sav file and import it as inagaki1999 (this file has 24 rows and 5 columns).

5. Dewaele and Pavlenko Bilingual Emotions Questionnaire (2001–2003). Use the BEQ.Context.sav file, imported as beqContext. Multilinguals who answered the questionnaire rated their proficiency in L2 speaking, comprehension, reading, and writing on a 1–5 scale (where 5 = fully fluent). Multilinguals also differed in the context in which they had acquired their L2, if it was naturalistic, instructed, or both. Examine the question of whether there are differences in proficiency on L2 speaking (l2speak) and L2 reading (l2read) depending on the context of instruction (L2Context). Be sure to first visually examine the data, and report effect sizes. Perform robust one-way ANOVAs on this data; do the results change?

© Taylor & Francis