Chapter 8

Chi-Square

8.1 Summarizing and Visualizing Data

With categorical data, tables are a useful tool for seeing a summary of the data and getting a quick view of the patterns that may be present. In the case of goodness-of-fit data, because there is only one variable, a simple tabular summary is the best way to examine the data. For group comparison data, variables can be crosstabulated to produce a contingency table of the data.

8.1.1 Summary Tables for Goodness-of-Fit Data

In producing a table to summarize patterns with a goodness-of-fit situation, data from Geeslin and Guijarro-Fuentes (2006) will be used. The authors wanted to know whether Spanish speakers (both L1 and L2), in a given context, preferred the Spanish verb *ser*, *estar*, or the use of both. The dependent variable was the choice of verb, which is categorical and has no inherent ranking or value. Note that the dependent variable is a count of how many times each of the three possibilities of verbs was chosen. The independent variable was also categorical and consisted of membership in one of the three populations (Spanish L1 speaker, Portuguese L1 speaker, or Portuguese L1 learner of Spanish L2). The data set you will use is one I calculated by using the report for percentage use of the verbs in Appendix B of Geeslin and Guijarro-Fuentes's paper and knowing how many participants were in each group. Note that a summary of the data over all of the items would result in a situation like that of Scenario Four (mentioned in the book on p. 214), so only the responses of native speakers of Spanish from item 3 will be examined (you can follow along with me by importing the SPSS file GeeslinGF3_5.sav and saving it as geeslin3).

To make a frequency table when you have raw count data, in R commander choose STATISTICS > SUMMARIES > FREQUENCY DISTRIBUTIONS. Choose the variables you want to see summarized. The additional box you can check on the dialogue box (shown in Figure 8.1) will conduct a chi-square goodness-of-fit test, which we will examine later in the chapter, so for now don't check it.

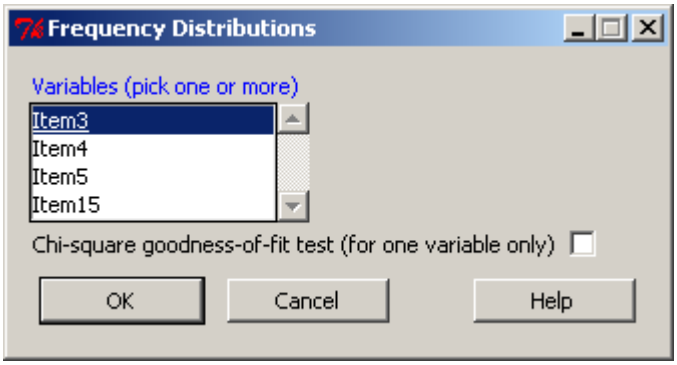

Figure 8.1 R Commander frequency distribution dialogue box.

The output is a simple table that counts the number of responses in each category and a table of the percentage of the counts:

```
    Estar   Ser  Both
       13     4     2

   Estar      Ser     Both
68.42105 21.05263 10.52632
```

The output shows that, although the majority of native speakers chose to use *estar* in the situation for item 3 (68.4%), there was some variation with the other two choices as well.

The R code for this action in R Commander has several steps because it wants to produce both a raw count table and a percentage table, as I've shown above:

```
.Table <- table(geeslin3$Item3)    #puts the table into an object that can be called again
.Table                             # counts for Item3
100*.Table/sum(.Table)             # percentages for Item3
remove(.Table)                     #takes the table out of the desktop
```

Creating a Frequency Table with One Categorical Variable

1. In R Commander, choose STATISTICS > SUMMARIES > FREQUENCY DISTRIBUTIONS. Choose the variable(s) you want to summarize.

2. R code:

```
table(geeslin3$Item3) #gives the raw counts
100* table(geeslin3$Item3)/sum(table(geeslin3$Item3)) #gives percentages
```

8.1.2 Summaries of Group Comparison Data (Crosstabs)

When you have more than one variable, you will want to look at the crosstabulation of the variables to understand the interactions that are occurring. I will use data from the Dewaele and Pavlenko (2001–2003) Bilingualism and Emotion Questionnaire (BEQ) (to follow along with me, import the SPSS file BEQ.Dominance and name it beqDom). I will be asking the question of whether the number of languages someone speaks has any relationship to whether they choose their first language as their dominant language, their non-dominant language, or a co-dominant language. In this data set there are two categorical variables, that of number of

languages, and language dominance (CatDominance). For the crosstabs in R Commander, pull down STATISTICS > CONTINGENCY TABLES > TWO-WAY TABLE. In this instance, with only two variables, as seen in Figure 8.2, you will choose the two-way table option but from the Contingency tables menu you should have seen that there is a multi-way table available as well. Note that all the variables need to be classified as "character" variables in R for the two-way table to be a choice in the dialogue box (see Appendix A if you need help with changing a numeric vector into a categorical vector). This information pertains to the situation where you have raw counts of data, as in the BEQ.Dominance.sav file.

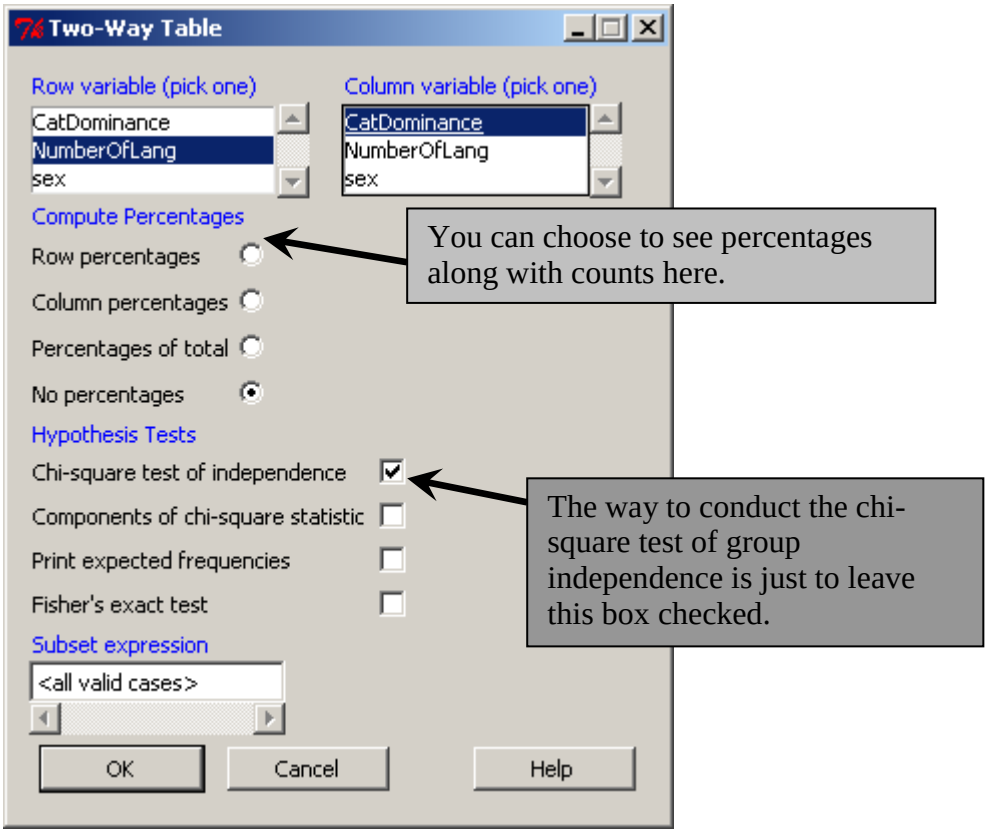

Figure 8.2 Obtaining a crosstab with two categorical variables in R.

The output of the two-way table (here done without looking at any percentages) is shown in Table 8.1 (with the format improved).

Table 8.1 Dewaele and Pavlenko Two-Way Table

No. of Languages	*L1 Dominant*	*Other Dominant*	*L1 + Other(s) Dominant*
Two	94	26	17
Three	159	26	83
Four	148	23	110
Five	157	30	163

Very quickly we can see that, although generally the majority of people reply that their L1 is dominant, the number of L1+ answers gets larger as the number of languages goes up, until for those who know five languages this is the answer with the largest count. On the other

hand, the number of those who are dominant in a language that is not their L1 is smaller, but this doesn't seem to increase as the number of languages known increases.

Here is the analysis of the R code that R Commander uses for the crosstab (without any percentages called for) in R:

```
.Table <- xtabs(~NumberOfLang+CatDominance, data=beqDom)
```

.Table <- . . .	The data is put into an object named ".Table."
.xtabs(~NumberOfLang+CatDominance, data=beqDom)	The xtabs() command will build a contingency table out of the variables following the tilde. Order affects how contingency tables look—put the row variable first and the column variable second.

After you finish this command you will need to ask R to *show* the data for the table like this:

```
.Table
```

Another situation that might arise is when you have the information from a summary table already (something like what is shown in Table 8.1). You can then create your own matrix from the data and perform the commands listed below (such as rowPercents()) on this data as well. The following commands create a 2×2 data set that crosses the use of relative clauses with teaching method:

```
TM<-matrix(c(12,0,18,16),nrow=2,ncol=2,byrow=T, dimnames=list(c("Relative Clauses", "No +RCs"), c("Method A", "Method B")))
```

The dimnames() command lists the names to give the rows first ("Relative clauses," "No RCs"), and then the names to give the columns.

If you have more than two categorical variables and raw count data, you should choose the MULTI-WAY TABLE option under CONTINGENCY TABLES. The format is the same as for the two-way table except that the Hypothesis Tests are not included. For three-way or higher tables, you may need to experiment to see which order brings out the format of your data that is best suited to your purposes. As a table can have only two dimensions, the control variable is the one which will split your data into separate tables. Just to show you an example of what this would look like, using the same beqDom data I will investigate the question of whether sex interacts with the other variables, and since I have chosen it as the control variable I will get one table with only the data for females and one table with the data for males. Figure 8.3 shows the multi-way table.

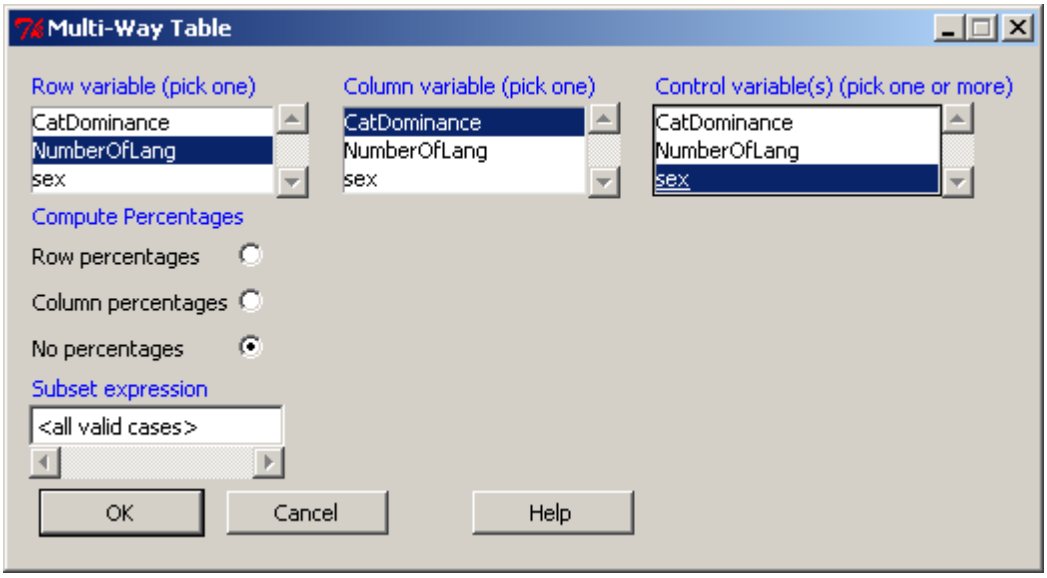

Figure 8.3 How to make a crosstab with three or more categorical variables in R.

The result is two separate tables, split by sex:

```
, , sex = F

             CatDominance
NumberOfLang YES  NO YESPLUS
       Two    66  19      14
       Three 113  18      61
       Four   95  18      92
       Five   99  23     112

, , sex = M

             CatDominance
NumberOfLang YES  NO YESPLUS
       Two    28   7       3
       Three  46   8      22
       Four   53   5      18
       Five   58   7      51
```

The R command is very simple—just add the third variable to the end of the equation seen for the two-way table:

```
.Table <- xtabs(~NumberOfLang+CatDominance+sex, data=beqDom)
.Table
```

In R you cannot build one table that shows raw counts and percentages in the same count table, but you can easily calculate the percentages and create your own formatted table like this:

```
rowPercents(.Table) # Row Percentages
colPercents(.Table) # Column Percentages
totPercents(.Table) # Percentage of Total; this only works with two-way tables
```

Creating a Crosstabulated Table in R with Two or More Categorical Variables

1. On R Commander's drop-down menu, choose STATISTICS > CONTINGENCY TABLES > TWO-WAY TABLE (or MULTI-WAY TABLE)

2. Choose the variables for row and column. The "Control" variable in the multi-way table will split your data. In order to get the data in a format that is easy to understand you might need to experiment with changing the order of the variables in these areas.

3. The syntax for obtaining crosstabs in R is:

```
.Table <- xtabs(~NumberOfLang+CatDominance, data=begDom)
```

where any number of variables can be entered after the tilde but the first one will be the rows of the table, the second the columns, and the third and further variables will split the data into separate tables.

8.1.3 Visualizing Categorical Data

Categorical data consist of counts of frequencies, so a traditional way of visualizing such data has been with barplots. In this section I will examine how to make barplots with R, but I do not recommend using them. In this section I will show you three new and exciting plots for categorical data that are more helpful than barplots in visualizing your data. Friendly (2000) notes that, while methods for visualizing quantitative data have a long history and are widely used, methods for visualizing categorical data are quite new, and most people do not yet know about them. Take a look at what association plots, mosaic plots, and doubledecker plots can do to showcase your categorical data.

8.1.4 Barplots in R

Perhaps because barplots are not a highly favored graphic by more statistically minded people, they are not very sophisticated in R Commander. You can use R Commander to create a barplot for one categorical variable that will provide a frequency count of the different categories of the variable. To make more sophisticated barplots, R code will need to be used. I will demonstrate the use of a barplot for one categorical variable in R by using the Geeslin and Guijarro-Fuentes data (geeslin3). In this data set, native speakers of Spanish chose their preferred verb. A barplot will simply provide a visual view of the frequency with which each choice was picked.

In R Commander, choose GRAPHS > BAR GRAPH. Pick the variable you want to examine. You will get output that looks like Figure 8.4.

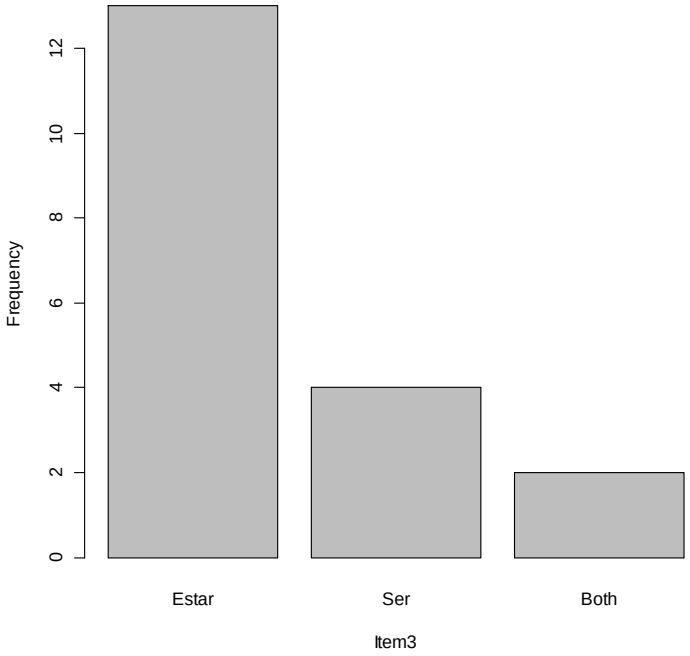

Figure 8.4 Barplot of one categorical variable in R from Geeslin and Guijarro-Fuentes (2006) data.

The R syntax which generates this figure can be analyzed as follows:

```
barplot(table(geeslin3$Item3), xlab="Item3", ylab="Frequency")
```

barplot()	The command to make the barplot.
table (geeslin3$Item3)	Turns geeslin3$Item3 into a contingency table (we saw this command earlier in the chapter to make a summary count table of one variable).
xlab, ylab	These arguments label the x and y axes respectively.

To look at barplots with two categorical variables, let's examine the Dewaele and Pavlenko BEQ data (beqDom file). We want to look at the relationship between the number of languages a person knows and which language they say is their dominant language. Look at a barplot of this data in Figure 8.5.

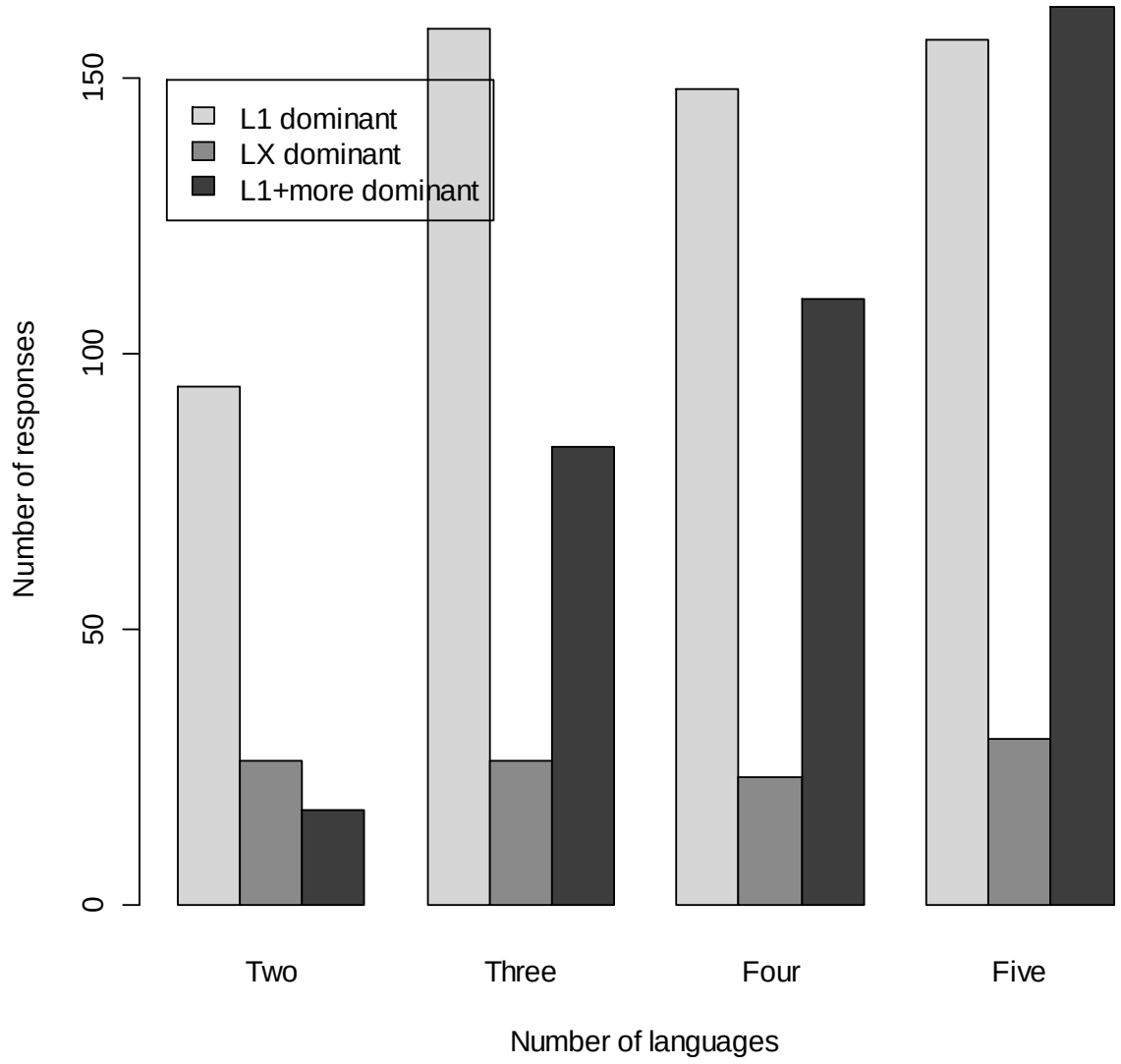

Figure 8.5 Barplot of two categorical variables in R from Dewaele and Pavlenko (2001–2003) data.

The barplot shows graphically that, although the response for LX (any other language besides L1) stays the same no matter how many languages a person knows, the response for having L1 *plus* at least one other language be dominant increases with the number of languages a person knows.

The code for this barplot is:

```
library(epitools)
colors.plot(TRUE) #use this to pick out three colors
 x y color.names
1 14 12 grey83
2 14 11 grey53
3 14 10 grey23
```

```
attach(beqDom) #by doing this, we can just specify variable names
barplot(tapply(CatDominance, list(CatDominance, NumberOfLang), length),
col=c("grey84", "grey54", "grey24"),beside=T,ylab="Number of responses",
xlab="Number of languages")
locator() #allows you to pick a point on graph to locate the legend
$x $y
[1] 0.8272034 158.2098
legend(.827,149.7,legend=c("L1 dominant", "LX dominant", "L1+more dominant"),
fill=c("grey84", "grey54", "grey24"))
detach(beqDom)
```

The analysis of the barplot command is:

```
barplot(tapply(CatDominance, list(CatDominance, NumberOfLang), length),
col=c("grey84", "grey54", "grey24"),beside=T,ylab="Number of responses",
xlab="Number of languages")
```

barplot()	The command to make the barplot.
tapply (x, index, function)	Applies a function to the array in x using a list of factors in the index argument.
x= CatDominance	This array for the tapply command is the CatDominance factor. Choose the dependent variable. I want my barplot to show how many people are in each of the categories of dominance (there are three: L1, LX, and L1PLUS).
index=list(CatDominance, NumberOfLang)	This index for the tapply command gives a list of two factors. If I put in only NumberOfLang, I would end up with only one bar for each factor in NumberOfLang. In order to split these up into three bars for each factor (one for each of the dominance categories), I need to put both factors that I want in this index list.
function=length	This function for tapply is just the number of responses, so I use the function length. Other possible functions you might want to use include: mean, median, sum, range, quantile.
col=c(. . .)	Specifies the colors to fill in the boxes with; I used the epitools library, color.plot=TRUE command to bring up a graphic containing all of R's colors (see code above). I then clicked on the three colors I wanted, and the print-out returned the names of the colors.
beside=T	Causes the bars of the barplot to stand upright.
ylab="", xlab=""	Gives labels to the x- and y-axes.

The analysis of the legend command is:

```
legend(.827,158,legend=c("L1 dominant", "LX dominant", "L1+more dominant"),
fill=c("grey83", "grey53", "grey23"))
```

legend (x, y, legend, fill)	The command to plot a legend at point x, y, and to add a legend and fill colors for the legend.
x=.827, y=158	I obtained this value for x, y by previously using the locator command.
legend=c("L1dominant", "LX dominant", "L1+more dominant")	The part inside the c() specifies what characters to use to label the legend.

fill=c("grey83", . . .)	If you want to have boxes that show colors matching your barplot boxes, add this argument.

Although barplots can be useful to help visualize categorical data, I will note once again that I do not recommend barplots for graphing interval data. However, if you do insist on using a barplot for interval data you really ought to put error bars on it at least (the frequency counts shown here cannot have error bars because categorical data is summarized using counts and percentages, not measures of dispersion). Instructions on how to write code that will produce error bars in R is given in Crawley (2007) starting on page 56. You should also choose "mean" for the function instead of "length" as was shown here for count data. You can also see the barplot and code in Chapter 11 for the Writing.txt file.

Creating a Barplot in R

1. If you have one variable, you can use R Commander's drop-down menu. Choose GRAPHS > BAR GRAPH. Pick your one variable.

2. The R syntax for one variable is:

```
barplot(table(geeslin3$Item3))
```

3. If you have two variables, use the R syntax:

```
barplot(tapply(CatDominance, list(CatDominance, NumberOfLang), length),
col=c("grey84", "grey54", "grey24"),beside=T,ylab="Number of responses",
xlab="Number of languages")
legend(.827,158,legend=c("L1 dominant", "LX dominant", "L1+more dominant"),
fill=c("grey83", "grey53", "grey23"))
```

If you have summary data instead of raw count data, you can also easily create a barplot with that data. Using the data set mentioned previously in this section (TM), I would simply use the following code to get a nice-looking barplot:

```
barplot(TM,beside=T, main="Teaching Method and Production of Relative Clauses")
legend(3.24,14.1,c("Rel.clauses",  "No RCs"),fill=c("grey22", "grey83"))
```

8.1.5 New Techniques for Visualizing Categorical Data

The next part of this document will contain ways of visualizing categorical data as flat contingency tables. Meyer, Zeileis, and Hornik (2007) explain that, in these kinds of plots, "the variables are nested into rows and columns using recursive conditional splits. . . . The result is a "flat" representation that can be visualized in ways similar to a two-dimensional table" (p. 5). The three plots that I like best are association plots, mosaic plots, and doubledecker plots. Although commands for association plots and mosaic plots exist in the base R system (assocplot, mosaicplot), I have found that it can be difficult to get data into the correct format to work with these commands. A better choice is the vcd library, which provides an easy way to format data through the structable command, and provides more ways to manipulate graphic parameters (see Meyer, Zeileis, & Hornik, 2007 for examples). The vcd library also provides other plots for visualizing categorical data, such as the sieve plot, cotab plot, and pairs plot, although these plots will not be explored here.

8.1.6 Association Plots

The Cohen–Friendly association plot in the vcd library (assoc) shows visually where the counts in the data exceed the expected frequency and where they show less than the expected frequency. More technically, it shows the "residuals of an independence model for a contingency table" (Meyer, Zeileis, & Hornik, 2007). In this way, the association plot can help the reader quickly ascertain which interactions are different from what is expected by the assumption that the two variables are independent.

Let's examine the Dewaele and Pavlenko (beqDom) data for the relationship between number of languages and language dominance to illustrate this graphic.

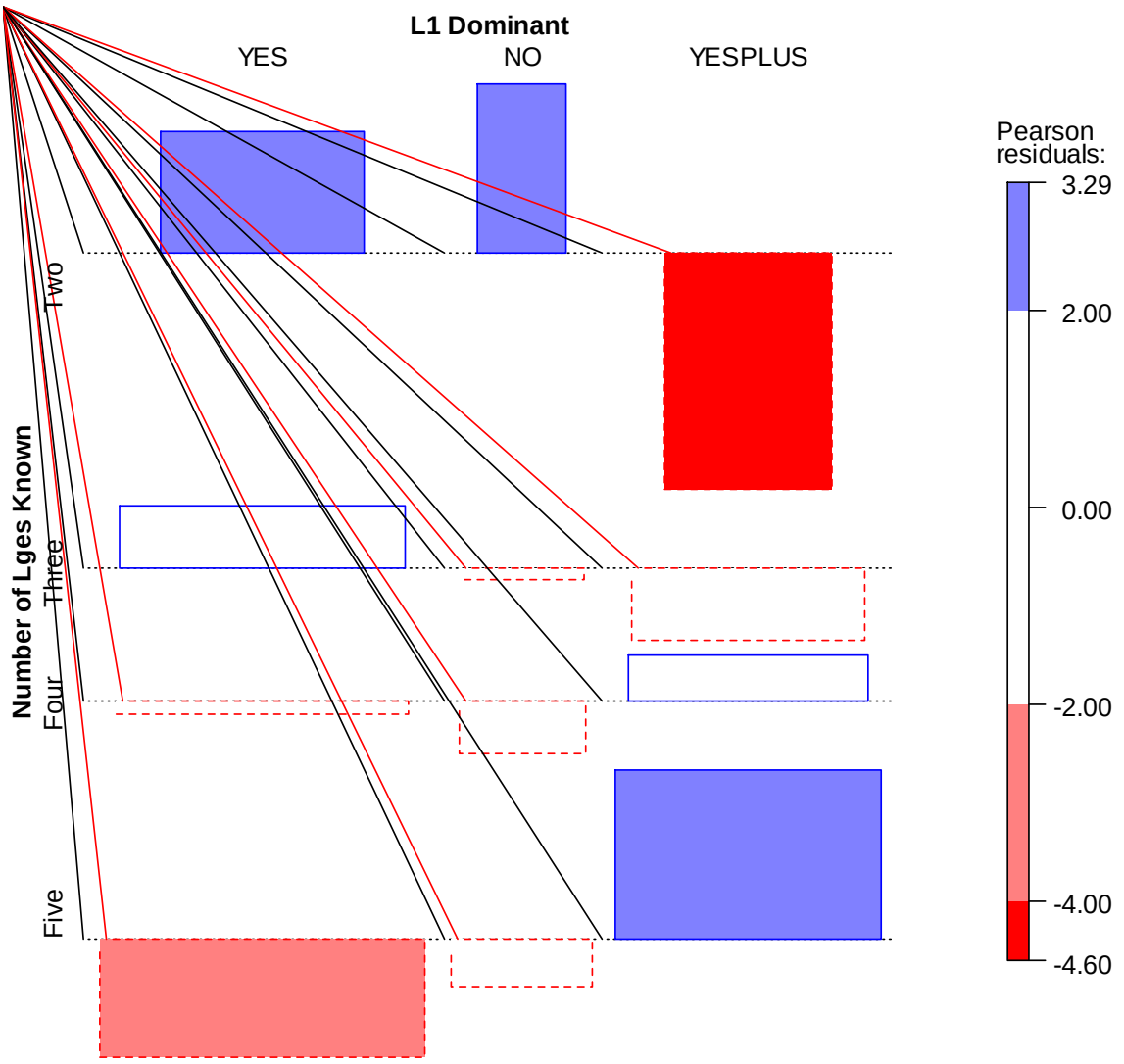

Figure 8.6 Association plot (Dewaele and Pavlenko data).

In Figure 8.6 you can see that all of the boxes above the line are blue and have solid lines and those that are below the line are red and have dotted lines. What we can see here is that there are more persons (solid lines) with two languages who say their L1 is their dominant language (YES) or is not their dominant language (NO) than expected, while there are fewer persons with two languages (dotted lines) who say they are dominant in more than one

language (YESPLUS) than we would expect if there were no relationship between number of languages and language dominance (the null hypothesis). The departure from expected is not so striking for those with three and four languages, but we again see some striking differences for persons with five languages. There are fewer persons (dotted lines) with five languages who say their L1 is their dominant language (YES), and there are more persons (solid lines) with five languages who say they are dominant in more than one language (YESPLUS) than would be expected if the variables were independent.

The Pearson residuals plot to the right uses both color and saturation to indicate inferences that can be made from the data. The most saturated hue (the one below −4.00) for the red (dotted line) allows you to identify areas where the null hypothesis that the variables are independent can be rejected (Meyer, Zeileis, & Hornik, 2007). Residuals above 4 or below −4 indicate a difference that means the null hypothesis can be rejected. Residuals between 2 and 4 (and −2 and −4) are medium-sized and do not indicate a statistical rejection of the null hypothesis (Meyer, Zeileis, & Hornik, 2007). Therefore, in Figure 8.6, the only cell which we can see is individually statistical is the one between persons with two languages and L1 dominance in more than one language (YESPLUS).

In order to create an association plot, the data need to be in the form of a contingency table. The beqDom file is a data frame, so a new object that is a contingency table must be created. The variables that will be used are CatDominance (the answer for which language is dominant) and NumberOfLang (the number of languages a person knows). The commands I used to generate this plot are below.

```
library(vcd)
(DOM=structable(CatDominance~NumberOfLang,data=beqDom))
assoc(DOM, gp=shading_Friendly, labeling_args=list(set_varnames=
c(CatDominance ="L1 Dominant", NumberOfLang ="Number of Lges Known")))
```

Here is the analysis for getting the data into the correct format:

(DOM=structable(CatDominance ~ NumberOfLang,data=beqDom))

(. . .)	Putting the parentheses around the entire command will cause the table to appear without having to call the object; I have named the object DOM but it could be named anything—x, blah, etc.
structable()	This vcd command creates a structured contingency table which automatically removes NA values and allows you to specify exactly which columns you want to use out of a certain data frame.
CATDOMIN~ NUMBEROF	The formula inside the structable() command is exactly like the regression equation (see Chapter 7). Here it means that the outcome, dominance in a language, is modeled by the predictor variable, number of languages spoken.
data=beqDom	Specifies the data frame to be used.

This command is the one that produces the actual graphic:

```
assoc(DOM, gp=shading_Friendly, labeling_args=list(set_varnames=
c(CatDominance ="L1 Dominant", NumberOfLang ="Number of Lges Known")))
```

assoc ()	The command for an association plot in the vcd library.
gp=shading_Friendly	Specifies the type of shading on the plot; if left off, the association plot will consist of equally colored grey boxes. I like the Friendly shading because it distinguishes positive and

	negative with solid and dotted lines, making it better for black and white production. One other nice shading option is gp=shading_hcl.
labeling_args=list(set_varnames= c(. . .))	This argument allows you to specify exactly what names you want to give to your variables. Another useful labeling argument for renaming factor levels could be added with a comma after the argument to the left (still in the labeling_args argument though): set_labels=list(CatDominance =c("L1", "LX", "L1+LX")

Note that the mosaic plot could be used as well with three variables. The object created by structable would just add one more variable. The variable of sex is contained in the beqDom data frame, so it is a simple matter to create a new object with three variables:

```
DOM3=structable(CatDominance~NumberOfLang+sex,data=beqDom)
assoc(DOM3)
```

If you do not have raw count data but instead summary data, it even easier to use the association plot command. First, create a data set with the data. The code shown here is for an imaginary experiment that looked at anxiety levels in a language classroom and the interaction with whether students had studied abroad (SA) or not.

```
anxietylevel<-matrix(c(14,44,38,58,4,21),nrow=3,ncol=2,byrow=T,
+ dimnames=list(c("low", "mid", "high"), #gives names for rows
+ c("SA", "NoSA"))) #gives names for columns
assoc(anxietylevel,gp=shading_Friendly,main="Study Abroad")
```

However, I think the mosaic plot or doubledecker plot is a better visual for this more complex situation, and so I will recommend you use those plots when you have three categorical variables.

Creating Association Plots in R

1. Open the vcd library:

```
library(vcd)
```

2. Put your data in the correct form:

```
(DOM=structable(CatDominance ~ NumberOfLang,data=beqDom))
```

3. Call the plot:

```
assoc(DOM, gp=shading_Friendly, labeling_args=list(set_varnames=
c(CatDominance ="L1 Dominant", NumberOfLang ="Number of Lges Known")))
```

Another plot which is quite similar to the association plot is the mosaic plot. In order to compare plots, we will use the same Dewaele and Pavlenko data in this section (beqDom).

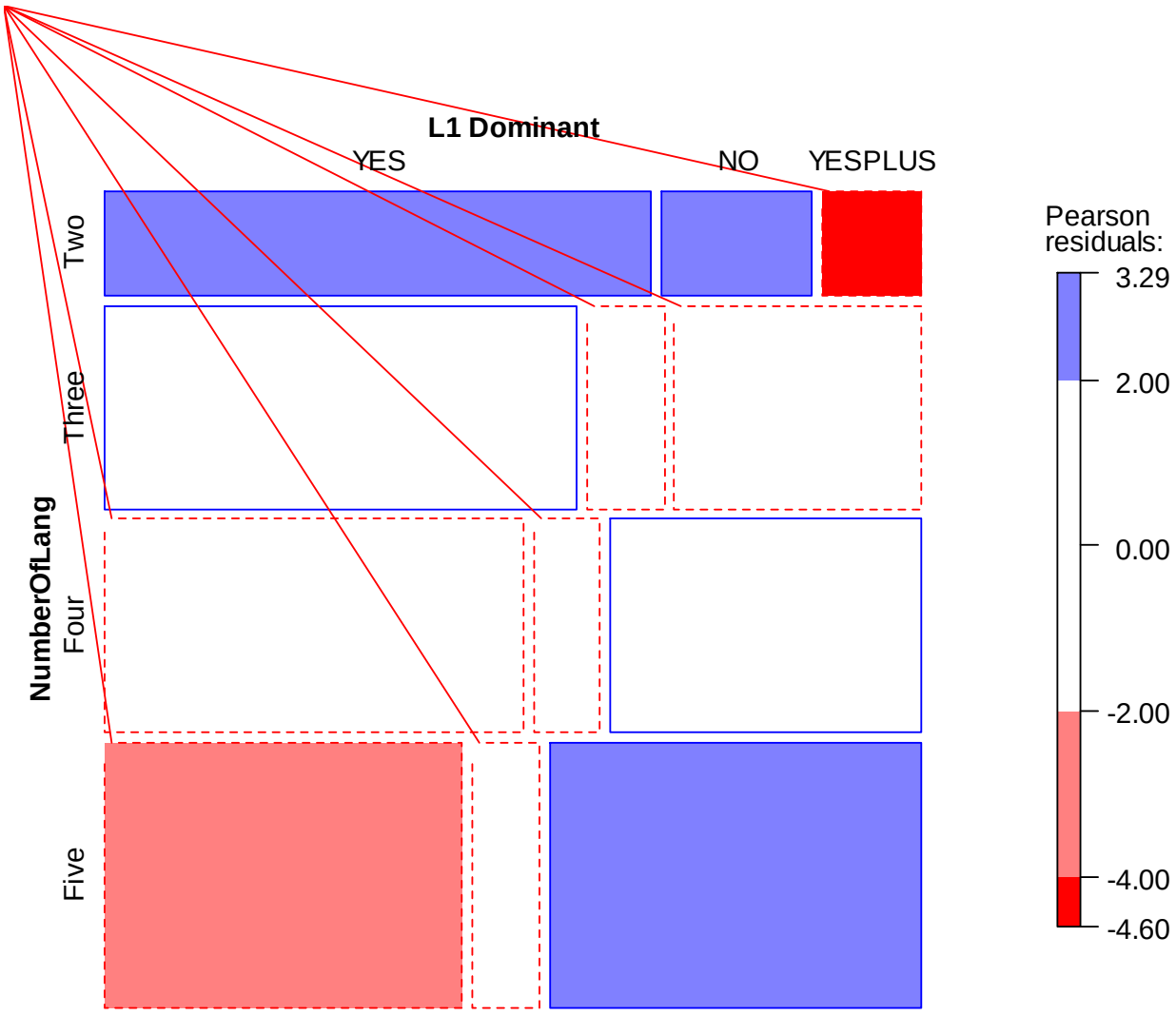

Figure 8.7 Mosaic plot with two variables (Dewaele and Pavlenko data).

It should be clear that the mosaic plot uses the same kind of reasoning as the association plot, with red-shaded areas (which are dotted) indicating values that are less than expected, and blue-shaded areas (which are solid) indicating values that are more than expected. The area of the boxes also gives an indication of its proportion to the whole, which the association plot did relative to the row but not to the whole data set. Using the same values of shading as in the association plot (from Friendly), individual cells that violate the assumption of independence are more deeply colored.

The command needed to create this plot is mosaic (), and uses exactly the same arguments as were needed for the association plot, including the object I've named DOM which has the data structured already:

```
mosaic(DOM, gp=shading_Friendly, labeling_args=list(set_varnames=
c(CatDominance="L1 Dominant", NumberOfLangs="Number of Lges Known")))
```

Mosaic plots are also good at representing the intersection of three categorical variables. Let's look at the Mackey and Silver (2005) data (mackey), this time not only looking at the relationship between experimental group and absence or presence of development in question formation on the delayed post-test, but also considering how pre-test developmental level interacted with the other factors.

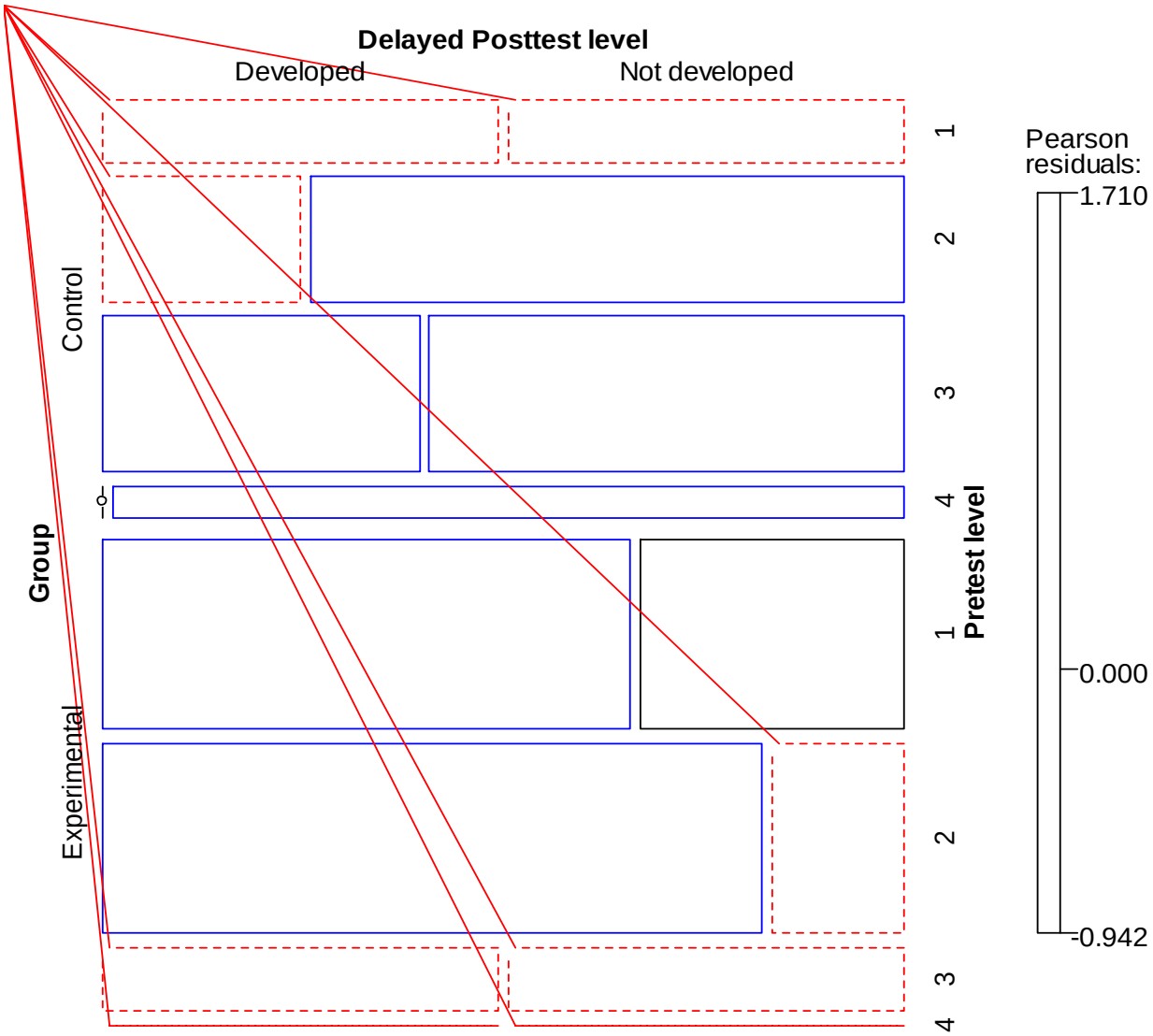

Figure 8.8 Mosaic plot with three variables (Mackey and Silver data).

Here is the R code that I used to make this three-variable association plot:

```
(DEV=structable(DevelopDelPost~Group+PreTest,data=mackey))
mosaic(DEV, gp=shading_Friendly, labeling_args=list(set_varnames=
c(DevelopDelPost="Delayed Posttest level", Group="Group", PreTest="Pretest
level")))
```

Just looking at the pattern of results with dotted lines being places where results are less than expected and solid lines those where results are more than expected, those in the control group who were in the lowest level of the pre-test had fewer results than expected in both the "developed" and "not developed" categories, whereas those in the highest levels (3 and 4) of the experimental group had fewer results in both the "developed" and "not developed" categories than expected. Notice that the boxes for level 4 are both quite skinny and the one under the control condition has a dot over it. This means there were not enough participants in this category to evaluate it. The Friendly shading shows us that none of the trends were large enough to merit much attention, and we would expect no statistical differences here from the null hypothesis that all of the variables are independent.

Creating Mosaic Plots in R

1. Open the vcd library:

```
library(vcd)
```

2. Put your data in the correct form:

```
(DOM=structable(CatDominance ~ NumberOfLang,data=beqDom))
```

3. Call the plot:

```
mosaic(DOM, gp=shading_Friendly, labeling_args=list(set_varnames=
c(CatDominance ="L1 Dominant", NumberOfLang ="Number of Lges Known")))
```

The doubledecker plot is a very nice visual when you have three or more categorical variables and you want to visualize conditional independent structures. For this section I will illustrate with the *Titanic* data (the data set is called Titanic and is in R's base data set, so you should be able to perform all of the commands in this section without importing any special data set). This data of course has nothing to do with linguistics, but understanding this table in the context of something you may already understand somewhat may help with figuring out how this table works. The table looks at what factors influenced whether passengers on the *Titanic* survived. The Titanic data set is a table, and if you type Titanic into R you will see that there are three predictor factors that are included in the Titanic survival tables: sex (male or female), age (child or adult) and class (first, second, third, or crew). With three predictors and one response variable (survival), we *can* include all of the variables in a mosaic or association plot,[1] but to my way of thinking the doubledecker plot is a more elegant solution.

The doubledecker plot in Figure 8.9 gives labels in layers underneath the table. The darker shadings are for those who survived (the skinny key that shows this is on the far right of the graph).

[1] If you want to see a mosaic plot utilizing all of the Titanic variables, run the following code (from Meyer, Zeileis, & Hornik, 2007, p. 31):

```
>mosaic(Titanic, labeling_args=list(rep=c(Survived=F,Age=F)))
```

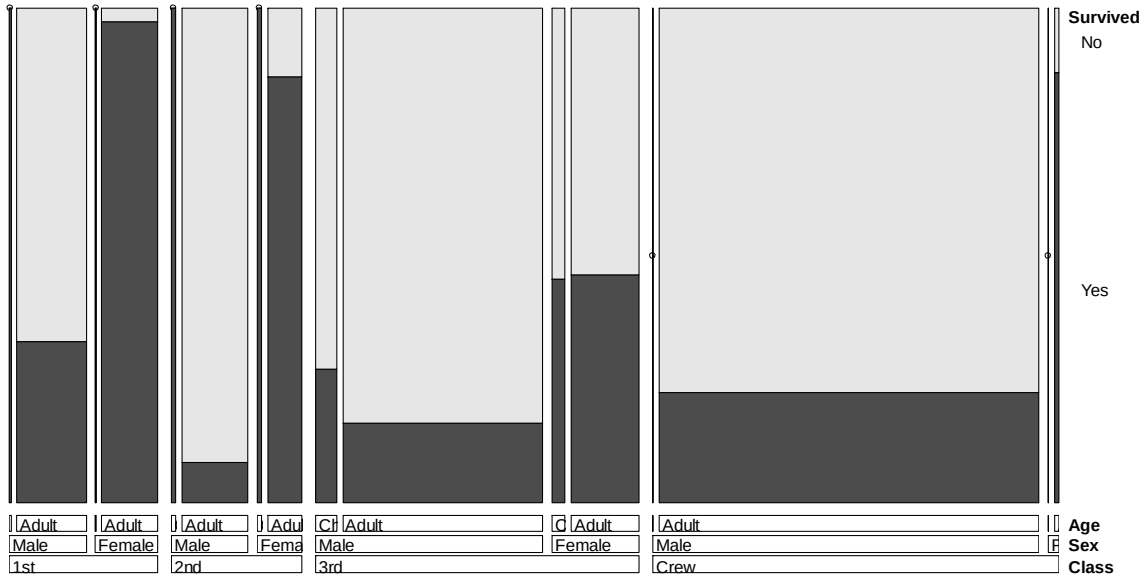

Figure 8.9 Doubledecker plot using four categorical variables (*Titanic* data).

As your eye moves from left to right over Figure 8.9, first you can see that proportionally more people in first class survived than in third class or crew (width of boxes represents proportions of numbers). Among first-class passengers (almost all of whom were adults; the children are a slim black line to the left of the adults in first class), proportionally more females than males survived.

The command for this table is (after the vcd library is opened):

```
doubledecker(Survived~Class+Sex+Age, data=Titanic)
```

The order of the arguments will change the order in the table, so if you create your own doubledecker plot you might want to experiment with the order.

Creating Doubledecker Plots in R

1. Open the vcd library:

```
library(vcd)
```

2. Call the plot:

```
doubledecker(Survived~Class+Sex+Age, data=Titanic)
```

8.2 Application Activities for Summarizing and Visualizing Data

8.2.1 Application Activities with Crosstabs

1. Use the Dewaele and Pavlenko BEQ.Dominance.sav data. Recreate the three-way intersection between number of languages known, dominance, and sex shown after Figure 8.3. Does the pattern noted overall between number of languages and dominance seem to hold equally well for both males and females?

2. Use the Mackey and Silver (2005) data set (use the MackeySilver2005.sav file and import it as mackey) and examine the frequency of the differing developmental levels of question formation in the pre-test. What is the most frequent developmental level? What is the least frequent?

3. Use the Mackey and Silver (2005) data set to examine whether there is any difference between experimental groups on the pre-test data. From just eyeballing the data, does it appear that students in both experimental groups had roughly the same distribution of developmental levels for question formation?

8.2.2 Application Activities with Barplots

1. Create a one-variable barplot for the Mackey and Silver (2005) data (use the MackeySilver2005.sav file and import it as mackey) for the delayed post-test data. Did more students from the experimental group develop in their ability to form questions or not?

2. Use the Mackey and Silver (2005) data set and create a barplot that examines the development of questions on the immediate post-test (DevelopPost), categorized by experimental group (Group). Can you see any patterns for the groups?

3. Using the fabricated data set of LanguageChoice.sav (import as languageChoice), create a barplot that shows the distribution of which language students choose to study, based on which university (Population) they come from. Comment on differences in language preference between the two universities.

4. Using the fabricated data set Motivation.sav (import as motivation), create one barplot that shows the distribution of YES and NO responses to the question "Do you like this class?" for the five teachers at the beginning of the semester (first). Then create another barplot that shows the distribution of responses at the end of the semester (last). What do you notice from the visual data?

8.2.3 Application Activities with Association Plots, Mosaic Plots, and Doubledecker Plots

1. Using the Dewaele and Pavlenko BEQ.Swear.sav file (import as beqSwear; notice this is a different file from the beqDom file we have been working with in the chi-square documents), create a mosaic plot examining the relationship between language dominance (L1dominance) as an outcome variable, and educational level (degree) and number of languages (numberoflanguages) as predictor variables. Educational level (degree) has four levels: Ph.D., MA, BA and A level. In order not to make the mosaic plot too messy, don't worry about changing the default labels on the graph, and instead add this argument to your mosaic plot: labeling_args=list(rep=c(degree=F) (it cuts out repetition of the educational-level labels). Do you see any patterns to remark upon?

2. Using the same beqSwear file, create a doubledecker plot with educational level and number of languages as the predictor variables and language dominance as the response variable. You might want to play around with the order of the two predictor variables to see which seems better. Do any patterns stand out? Compare how this same data looks in the mosaic plot versus the doubledecker plot. Which do you prefer?

3. Use the Mackey and Silver (2005) data set (mackey) and create an association plot that examines the development of questions on the delayed post-test (DevelopDelPost) as a function of development on the immediate post-test (DevelopPost) and experimental group (Group). What stands out? Compare the information you glean in this plot with the

information you got from the barplots in activities 1 and 2 in "Application Activities with Barplots" (above). Which do you feel is the more informative plot? Additionally, create a mosaic plot of the same data. Which do you prefer for this data set—the association plot or the mosaic plot?

4. Use the fabricated data set Motivation.sav (import as motivation). This data set shows the distribution of YES and NO responses as to whether students are motivated to learn a second language (the class they are enrolled in) at the beginning (first) and end (last) of the semester. Data is further divided into responses given in each of five teachers' classes. Study this data and try to come up with the best visual plot you can for the data. What would be most appropriate—an association plot, a mosaic plot, or a doubledecker plot? Try all of them and see what looks best to you. Furthermore, you will have to decide which of the variables you call the outcome and which of the others the predictor variables.

8.3 One-Way Goodness-of-Fit Test

Using the data available from Geeslin and Guijarro-Fuentes (2006), I will examine the question of whether native speakers had a preference for any of the three answers possible (the verb *ser*, the verb *estar*, or both verbs equally) on three items of their test (by the way, this is not the question the authors themselves asked). I imported the GeeslinGF3_5.sav file into R and called it geeslin3. The data can be in character or numerical form for this test.

To call for the goodness-of-fit chi-square test in R Commander, you follow exactly the same sequence as was seen in section 8.1, "Summarizing and Visualizing Data," to get a numerical summary: STATISTICS > SUMMARIES > FREQUENCY DISTRIBUTIONS. Pick the variable you want to test and tick the box that says "Chi-square goodness-of-fit test." After you press OK, an additional box will pop up, as shown in Figure 8.10.

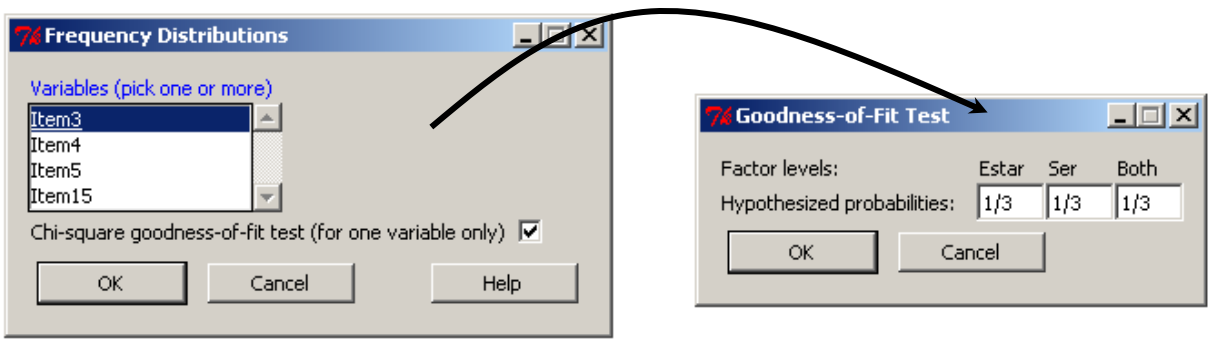

Figure 8.10 Dialogue boxes for goodness-of-fit chi-square test.

The null hypothesis that will be tested is that every category is equally likely. Since there are three categories, that means each category has a 1/3 chance of being selected. The "Goodness-of-fit" test dialogue box automatically assumes that each level has an equal chance unless you change this ratio manually.

The output for this test is shown here:

```
        Chi-squared test for given probabilities

data:  .Table
X-squared = 10.8421, df = 2, p-value = 0.004422
```

The chi-square test shows that the probability of getting this distribution of answers is highly unlikely (p=.004) if the null hypothesis is true. We therefore conclude that the L1 speakers of Spanish had a preference for some of the answers over others, although the chi-square test cannot tell us which answer was preferred. The chi-square does not tell us whether two of the choices (of the verb *ser*, *estar*, or both) are the same and one is different, or if all three are different. For that, however, look to the counts, which are reproduced again below. They clearly show that the native speakers favored the verb *estar* in this particular situation.

```
Estar   Ser  Both
   13     4     2
```

The simple R code for the chi-square test is:

```
chisq.test(table(geeslin3$Item3), correct=FALSE)
```

chisq.test()	Performs a chi-square test for group independence as well as goodness of fit.
table()	Creates a contingency table even if there is only one variable (xtabs is normally used if there is more than one variable).
(geeslin3$Item3)	The column of raw data.
correct=FALSE	Specifies that the continuity correction should not be applied. Default is TRUE.

By default, the probability of each choice is set to be equal, which is what I wanted in this case. However, if you would like to specify different probabilities using R code, you will need to create a new object where the probabilities sum to 1, and then include that object in the arguments to the chi-square command. For example, to test the hypothesis that there was a 40% chance of picking *estar* or *ser* and only a 20% chance of picking both, you would specify your *p*-value and then include it in the command like this:

```
prob=c(.4,.4,.2)
chisq.test(table(ggf35$ITEM3),correct=FALSE,p=probs)
```

To conduct the test on items 4 and 5, the steps above can be repeated. For item 4, however, a table summary shows that the only answer chosen was the first one, *estar*. Running the chi-square statistic will result in an error message that 'x' (your vector of numbers) must have at least two elements. However, it is hardly necessary to run a formal chi-square for this item, since it is clear there is little probability that all participants would choose *estar* if each of the three choices were equally likely. For item 5, the result is that χ^2=11.79, df=2 (because there are three choices), and p=.0028. Here the native speakers of Spanish chose *estar* 15 times, so it appears that, in all three items (3 through 5), the native speakers statistically preferred the choice of *estar* over the other choices.

Performing a One-Way Goodness-of-Fit Chi-Square

1. If desired you can specify the probabilities of each choice, making sure the numbers sum to 1: probs=c(.4, .4, .2)

2. Perform the chi-square:

chisq.test (table(ggf35$item5), p=probs)

(but leave out the last argument if your null hypothesis is that all choices are equally likely)

8.4 Two-Way Group-Independence Test

The chi-square test for independence uses two variables and tests the null hypothesis that there is no relationship between the two variables. To illustrate the use of this test I will use the Dewaele and Pavlenko data from their Bilingual Emotion Questionnaire (BEQ.Dominance, imported into R as beqDom). Dewaele and Pavlenko received answers from 1,578 multilinguals to the question as to what language they considered their dominant language. They categorized their answers into yes (L1 dominant), no (L1 not dominant), and yesplus (L1 plus more dominant). I will examine the question of whether the number of languages that a person speaks affects their language dominance.

To perform the two-way chi-square in R Commander, choose STATISTICS from the drop-down menu, and then CONTINGENCY TABLES, as shown in Figure 8.11. At this point, you will need to decide which choice in this menu is appropriate for your data.

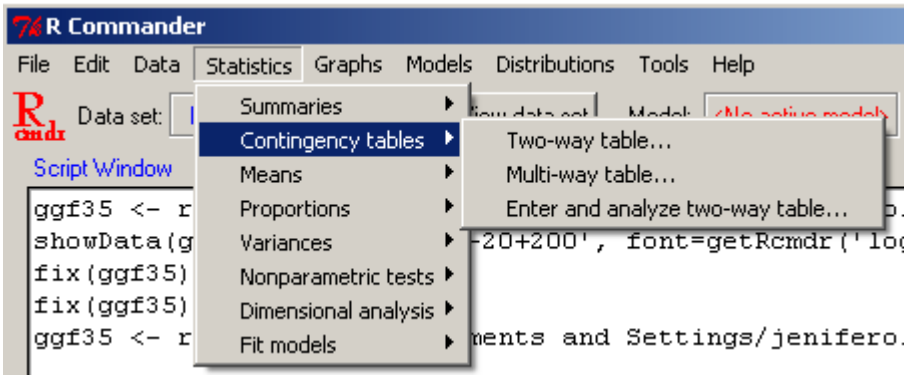

Figure 8.11 Opening up the two-way group-independence chi-square command in R.

The TWO-WAY TABLE choice is used when you have only two variables, no matter how many levels those variables have. Use this choice when your data are in raw input, not summary form (the MULTI-WAY TABLE can be used when you have three variables, but it will not perform a chi-square on the data; it will just produce two tables split by the third variable that you specify). The last choice, ENTER AND ANALYZE TWO-WAY TABLE, should be used when you have summary and not raw data (see further on in the chapter for an example).

The data in beqDom is not in summary form, so I choose TWO-WAY TABLE. We have previously seen the dialogue box that comes up, which lets us choose the row and column variables, whether we want to see percentages of the data, and which hypothesis tests we can pick (this is shown again in Figure 8.12).

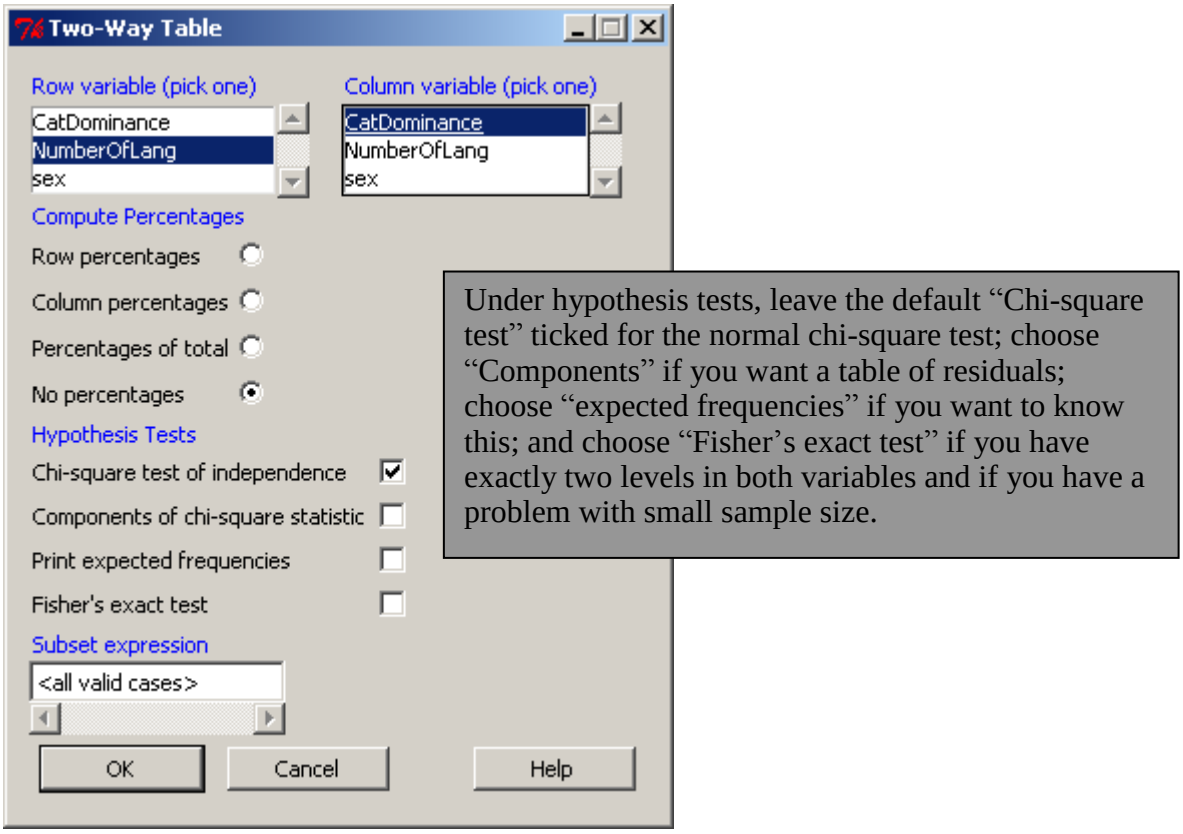

Figure 8.12 How to perform a two-way group-independence chi-square in R.

I did not choose anything besides the chi-square test, so besides the crosstabs summary count (seen previously in section 8.1, "Summarizing and Visualizing Data") my output looks like this:

```
        Pearson's Chi-squared test

data:  .Table
X-squared = 59.5807, df = 6, p-value = 5.476e-11
```

The result of the chi-square is that the χ^2 value is very large (59.6) on 6 degrees of freedom, and so the *p*-value is quite small (p=.000000000055). We reject the null hypothesis and say that there is a relationship between the number of languages someone speaks and which language they say is dominant. Note that the test itself does not tell us anything about the nature of this relationship with regard to specific details about the levels of the variables. For more information about how to do this, see section 8.5.4 of the SPSS book, *A Guide to Doing Statistics in Second Language Research Using SPSS*.

For an explanation of the relationship, plots are much more informative. We saw in the association plot in section 8.1, "Summarizing and Visualizing Data," that there are fewer people than would be expected who say they are dominant in more than one language if they know only two languages, while there are more people than would be expected who say they are dominant in more than one language if they know five languages. The barplot (Figure 8.5) was also informative in showing that the number of people who were dominant in more than one language seemed to increase monotonically as the number of languages known increased. Suffice it here to say that plots should always be used to augment the information given in a chi-square test.

Along with looking at the output, don't forget to check R Commander's message box for a warning if any of the expected frequencies are too low. This was not a problem with the Dewaele and Pavlenko data (this is a huge data set!), but Figure 8.13 is an example where the expected frequencies are small:

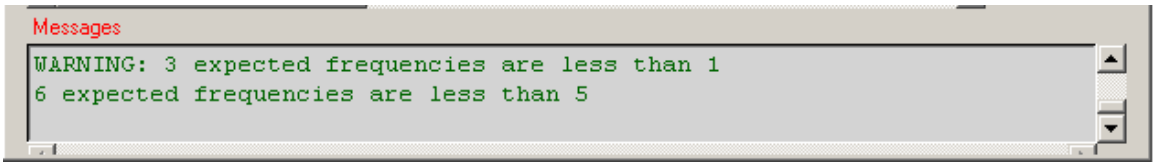

Figure 8.13 R Commander's warning about low expected frequencies.

The R code that was used to generate the chi-square test of group independence was:

```
chisq.test(xtabs(~CatDominance+NumberOfLang, data=beqDom),
correct=FALSE)
```

chisq.test	Performs a chi-square test for contingency tables as well as goodness of fit.
xtabs	Creates a contingency table of the specified variables.
(~CatDominance +NumberOfLang)	The variables that are tested in this two-way chi-square.
data=beqDom	Specifies the data set to be used.
correct=FALSE	Specifies that the continuity correction should not be applied. The correction is generally applied only when there are only two levels for each of the two variables, which is called a 2×2 table (but I have argued that you should not use it even in that case). The default setting is TRUE, so be sure to include this argument.

Going back to the choice of entering data directly into a contingency table (ENTER AND ANALYZE TWO-WAY TABLE in Figure 8.11), let's say that you were looking at a contingency table in a published report but did not have the raw data. It would be easy to perform a chi-square test (but not draw up an association or mosaic plot!) if you only had summary data. An example of this is a study by Shi (2001) in which the author asked whether teachers whose L1 was English emphasized different aspects of essay grading from teachers whose L1 was Chinese. The teachers graded holistically, and then could list three comments, in order of their importance. The chi-square we will conduct will examine whether the teachers differed in the amount of importance they attached to different areas (where the importance of these areas was simply the frequency with which they made comments which fell into these categories). Actually, the author did not give a frequency table, but I am estimating frequencies from the barplot in Figure 2 of the paper, where the author organized comments from the teachers into 12 different categories. My interpolated data is in Table 8.2.

Table 8.2 Grading Importance in Shi's (2001) Study of Writing Teachers

	General	*Content*	*Ideas*	*Argument*	*Organization*	*Paragraph Organization*	*Transitions*	*Language*	*Intelligibility*	*Accuracy*	*Fluency*	*Length*

English L1	11	19	35	57	23	9	5	3	23	22	14	4
Chinese L1	22	4	58	64	42	16	0	0	15	2	4	3

Right off the bat we can guess that this test will be statistical, just because there are so many different categories that it would not be hard to find that some of the groups performed differently from the others! But I am simply using this data to illustrate how to perform a chi-square if all you have is summary data (and, surprisingly, even summary data is hard to find in many published reports using chi-square!).

In R Commander if you go to STATISTICS > CONTINGENCY > ENTER AND ANALYZE TWO-WAY TABLE (as shown in Figure 8.11), then you will see the dialogue box in Figure 8.14.

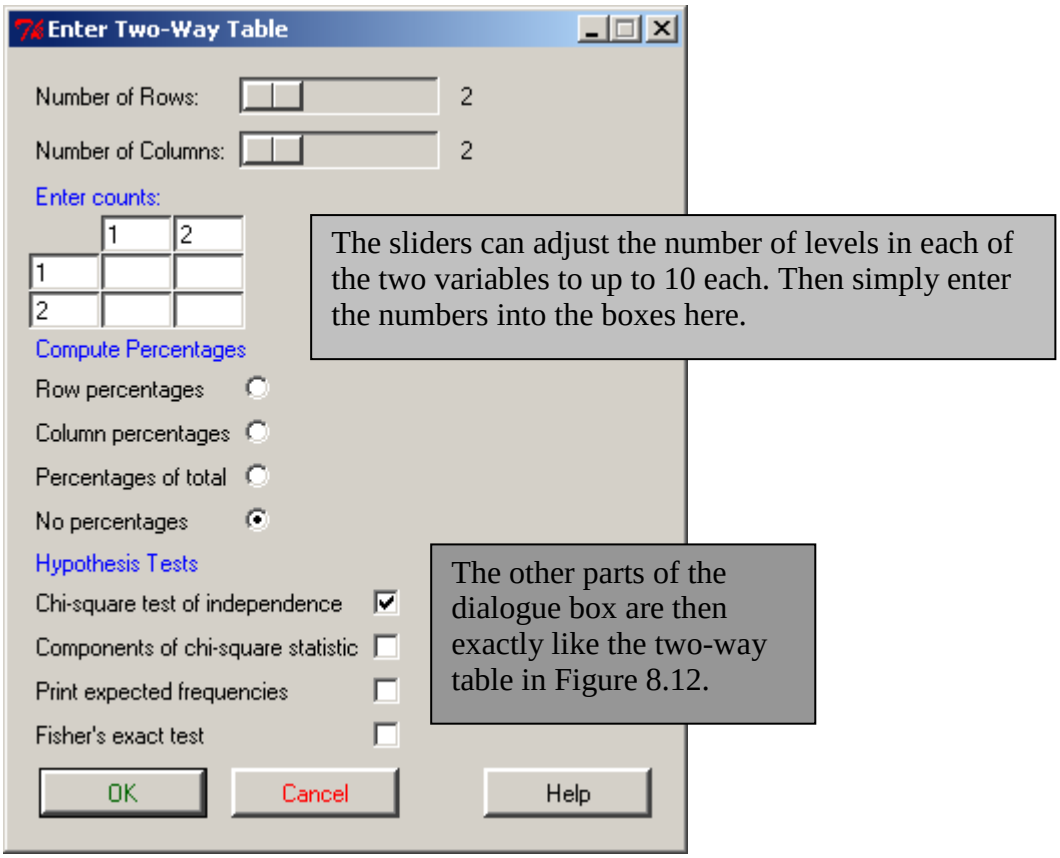

Figure 8.14 Enter a two-way table in R Commander.

Since I couldn't enter all of the variables from Shi's table here (she had 12 variables and R Commander's dialogue box lets me put in only 10), I instead used R syntax to get the entire table analyzed:

```
> T=matrix(c(11,19,35,57,23,9,5,3,23,22,14,4,
+ 22,4,58,64,42,16,0,0,15,2,4,3),2,12,byrow=T)
> T
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
[1,]   11   19   35   57   23    9    5    3   23    22    14     4
[2,]   22    4   58   64   42   16    0    0   15     2     4     3
> chisq.test(T,correct=FALSE)
Warning in chisq.test(T, correct = FALSE) : Chi-squared approximation

        Pearson's Chi-squared test

data:  T
X-squared = 59.0577, df = 11, p-value = 1.387e-08
```

As predicted, this chi-square has a very small *p*-value (the author gives the following chi-square results: χ^2 = 51.14, 19.99, 58.42; df = 11, $p < 0.001$; it is not clear why the author gives three chi-square numbers, but the third one looks very close to my result). Also, note the warning in the output above, after the chi-square command. The full warning says: "Chi-squared approximation may be incorrect." This is the warning that is generated when expected frequencies are smaller than 5. The easiest way to tell how many cells have expected frequencies less than 5 is to use R Commander. However, there is also a way to draw this information out from R. If you put the results of the chi-square test into an object, here named .Test, you can look at the expected values and see how many are less than 5.

```
.Test=chisq.test(T,correct=FALSE) #Remember, the false is for continuity correction
.Test$expected
```

```
         [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]
[1,] 16.31868 11.37363 45.98901 59.83516 32.14286 12.36264 2.472527 1.483516
[2,] 16.68132 11.62637 47.01099 61.16484 32.85714 12.63736 2.527473 1.516484
         [,9]    [,10]    [,11]    [,12]
[1,] 18.79121 11.86813 8.901099 3.461538
[2,] 19.20879 12.13187 9.098901 3.538462
```

Here, expected cell counts for columns 7, 8, and 12 are less than 5, so there are six cells with counts less than 5. Remember that in the section on assumptions for chi-square in the SPSS book (*A Guide to Doing Statistics in Second Language Research Using SPSS*, section 8.4, "Assumptions of Chi-Square," pp. 226–227) I noted that violating this assumption of a "normal" distribution was problematic only insofar as it would result in low power. Since we have found a statistical result we really don't have to worry about the fact that expected values are low.

I would like to note that the output from R's chi-square test gives the chi-square value, the degrees of freedom, and a *p*-value. However, it does not give any measure of effect size. As noted previously in the chapter, Howell (2002) recommends the phi-coefficient (φ) and Cramer's V as appropriate effect sizes for a chi-square test. Phi is used when there are only two levels of each variable. If you want these numbers you can use the command assocstats() for the two-way chi-square (this command comes from the vcd library).

```
summary(assocstats(xtabs(~CatDominance+NumberOfLang, data=beqDom)))
```

Notice the difference now in the amount of output generated by the assocstats() command:

```
Call: xtabs(formula = ~CatDominance + NumberOfLang, data = beqDom)
Number of cases in table: 1036
Number of factors: 2
Test for independence of all factors:
        Chisq = 59.58, df = 6, p-value = 5.476e-11
                     X^2 df   P(> X^2)
Likelihood Ratio 63.742  6 7.7904e-12
Pearson          59.581  6 5.4760e-11

Phi-Coefficient   : 0.24
Contingency Coeff.: 0.233
Cramer's V        : 0.17
```

Now we know that we had 1,036 valid cases tested here. We also have results for not just the Pearson chi-square, but also the likelihood ratio test, which is a popular alternative to the Pearson test that also uses the χ^2 distribution. Howell (2002) says that the likelihood ratio will be equivalent to the Pearson when the sample sizes are large (as we see is the case here). The print-out also shows effect sizes phi (φ=.24) and Cramer's V (V=.17). Since this is a 3×4 table (three levels in CatDominance and four in NumberOfLangs), it is appropriate to use the Cramer's V effect size. Phi and Cramer's V are percentage variance effect sizes, so this means that the number of languages that an individual knows explains 17% of the variance in the variable of L1 dominance. According to Cohen's guidelines in the SPSS book (*A Guide to Doing Statistics in Second Language Research Using SPSS*, Table 4.8, p. 119) for effect size strength, w=.17(sqrt(3-1))=.24, which is a small to medium effect size.

If you are dealing with ordinal data you might want to report the results of the linear-by-linear association test instead of the Pearson's chi-square or the likelihood test. This test can be obtained by using the coin library, with the independence_test command, like this:

```
library(coin)
independence_test(CatDominance~NumberOfLang,data=beqDom,teststat="quad")
```

```
        Asymptotic General Independence Test

data:  CatDominance by
         NumberOfLang (Two, Three, Four, Five)
chi-squared = 59.5232, df = 6, p-value = 5.625e-11
```

This number is only slightly lower than the Pearson chi-square result.

Performing a Two-Way Group-Independence Chi-Square

1. In R Commander, choose STATISTICS > CONTINGENCY > TWO-WAY TABLE (if you have raw data) or ENTER AND ANALYZE TWO-WAY TABLE (if you have summary data).

2. If you have raw data, choose variables (two) and press OK. If you have summary data, adjust the sliders to the correct number of levels for each of the two variables and enter the summary data.

3. The basic syntax in R is:

```
chisq.test(xtabs(~CatDominance+NumberOfLang, data=beqDom), correct=FALSE)
```

4. If you get a warning that the approximation may be incorrect, this means the expected count in at least one of your cells is less than 1. Check the warning message at the bottom of R Commander, or put the chi-square test into an object and pull up the expected counts this way:

```
.Test= chisq.test(xtabs(~CatDominance + NumberOfLang,data= beqDom),
correct=FALSE)
.Test$expected
```

5. In order to get effect sizes and the likelihood ratio test, use the assocstats command in the vcd library:

```
library(vcd)
summary(assocstats(xtabs(~CatDominance+NumberOfLang, data=beqDom)))
```

6. If you have ordinal data and want to get the linear-by-linear association, use the coin library:

```
library(coin)
independence_test(CatDominance~NumberOfLang,data=beqDom,teststat="quad")
```

8.5 Application Activities for Calculating One-Way Goodness-of-Fit and Two-Way Group-Independence Tests

1. Using the Geeslin and Guijarro-Fuentes (2006) data (geeslin3), analyze item15 to see whether the native speakers of Spanish chose each possibility with equal probability. Additionally, generate some kind of visual to help you better understand the data, and describe what you have found.

2. Using the same data as in activity 1 above, test the hypothesis that there is only a 10% probability that speakers will choose answer 3 (“both verbs equally”), while the probability that they will choose the other two choices is equal.

3. Using the Mackey and Silver (2005) data (mackey), investigate the question of whether there was any relationship between question development and experimental group for the immediate post-test (DevelopPost). Do the results for the immediate post-test hold true for the delayed post-test (DevelopDelPost)? Be sure to report on effect size.

4. We saw in previous sections that there seemed to be some statistical associations in the Dewaele and Pavlenko data between the number of languages people knew and their answer as to which language was dominant. Use the beqDom file (variables: CatDominance, NumberOfLang) to investigate this question using the chi-square test.

5. Bryan Smith (2004) tested the question of whether preemptive input or negotiation in a computer-mediated conversation was more effective for helping students to learn vocabulary. Smith considered the question of whether these strategies resulted in successful uptake. He found that 21 participants who heard preemptive input had no uptake, while 2 had successful uptake. He further found that 37 participants who heard negotiated input had no uptake and 6 had successful uptake. It's likely that the data are not independent (as Smith reported n=24 only in his paper), but, assuming that these data are independent, is there any relationship between the way the participants heard the vocabulary and whether they had successful uptake? You'll need to use the method for summary data.