13

Data Cleaning

INTRODUCTION

This chapter furthers our goal of reducing the gap between the raw data students usually collect in a research methods course and the pristine data usually analyzed in statistics courses. We began the demystification of the gap between collecting and analyzing data in the last chapter with our discussion of data entry. In this chapter we move on to data cleaning. Some elements of data cleaning, such as cosmetic cleaning, are very much related to comparable elements of data entry. The same types of decisions must be made, and there are the same documentation goals. However, in this chapter we extend data cleaning to include examining diagnostics, interviewer or mode effects, and longitudinal attrition. Data cleaning involves simple and effective steps that will ensure the highest quality data for analysis purposes.

This chapter assumes that, regardless of how the survey was administered, we now have a raw data file in a statistical software format. It also assumes that some sort of prior planning and setting of protocols were used to enter the data, create variable names, and handle response values and missing values. This next step, data cleaning, ranges from simple cosmetic fixes that make the dataset easier to analyze to diagnostics to assess the quality of the variables and the suitability of the dataset for regression analysis.

We advise all data cleaners to do their cleaning work in **syntax** (statistical software language). Using syntax, which is sometimes called computer coding or programming, simply means to write out the cleaning commands in the statistical software language rather than using a mouse to point and click through dialog boxes. Certainly the point-and-click method can be simpler (because the cleaner doesn't have to remember programming

codes), but it usually does not leave a trail of changes made to the dataset. There is no record of what has been done. SPSS Statistics, however, does allow a user to paste syntax from a point-and-click dialog box into a record of the executed procedures. In fact, recent versions of SPSS Statistics automatically paste syntax to the output window whenever a command is executed from a point-and-click dialog box. Syntax can be saved permanently to a file, and therefore there is a record of all data manipulations. This way, errors in data cleaning, when found during analysis, can be easily fixed.

We also advise not directly correcting the data, even though programs like SPSS make it so convenient (and tempting!) to do so. Again, there will be no record of the correction, so documentation will be incorrect. Mistakes are very easy to make (even for experts), and we often have to revert back to the raw data. This means that any fixes made directly to the data will have to be reentered. With syntax, we simply fix the mistake in the syntax and rerun it on the original dataset. In addition to leaving a visible record of all changes made to the dataset, writing in syntax will save much time and energy in the long run and ensure higher quality data.

The first section will discuss the basics of data cleaning for a simple cross-sectional survey. Next, additions for diagnostic cleaning and then longitudinal data will be addressed.

SIMPLE CROSS-SECTIONAL DATA CLEANING

Before cleaning the data, it is good to think through the process first and come up with some consistent practices that make the whole procedure easy to do and easy to understand. Figure 13.1 provides a checklist of all the data-cleaning items needed to properly clean a cross-sectional dataset. We start and end by examining each variable using a frequency procedure. Note which variables are string variables, which are scale items, and which are open-ended. This is important, because we cannot include string values and numeric variables in the same syntax code. Take notes on what each variable needs, and organize the variables by what they need. Starting with the cosmetics, make sure each variable has a variable label that links it to the questionnaire. Make sure that each response value has a value label. Determine whether the formatting of the variables needs to be changed. Note whether the variable has missing values and what those missing values are. Create a list of all the skip patterns, and note which are the gateway variables and which variables they skip. Similarly, create a list of all the variables that have an other-specify option as part of the question. Additional lists of any multiple-response variables and open-ended questions should be created in order to systematically ensure all variables are cleaned.

Start with a frequency of the unique identifier. If there is an identifier with a frequency of more than 1, then it is not unique. Either it was created wrong, it's a data entry mistake, or something went wrong in the syntax. Checking the unique identifier in the beginning and at the end of the syntax or any time datasets are merged is one way to

Figure 13.1 Cleaning Raw Cross-Sectional Data

1. Unique identifier
2. Cosmetics
   a. Value labels
   b. Variable labels
   c. Formats
3. Missing values
4. Skip patterns
5. Multiple-response questions
6. Other-specify responses
7. Open-ended questions
8. Notes
9. Multiple records

determine if any data manipulations have had a negative impact on the dataset being cleaned or analyzed. Mini Case 13.1 provides a frequency of a unique identifier called "Subject" with the variable label "respondent's ID number." If we scroll down the column labeled "Frequency," we want to see all 1s. This indicates uniqueness. Unfortunately, subject 1122 has a frequency of 2. This means two subjects were given the same identifier. Now we cannot distinguish between them. If this frequency were run prior to data cleaning, then we know we have to look to data entry to find this error. We may find that two interviewers used this same number. In fact, we may find that one interviewer transposed the unique identifier number, entering 1122 when he meant to enter 2211.

MINI CASE 13.1

Partial Frequency of a Unique Identifier

We start with the syntax to first access the data and then call for a frequency table to be run. Below the syntax we show the SPSS output the syntax generates. In SPSS, the syntax would look like this:

```
Get file "c:\my documents\survey\data.sav".
Frequency variables=subject.
```

The output is as follows:

Respondent's ID Number				
SUBJECT	**Frequency**	**Percent**	**Cumulative Frequency**	**Cumulative Percent**
1029	1	0.26	1	0.26
1040	1	0.26	2	0.52
1101	1	0.26	3	0.79
1104	1	0.26	4	1.05
1106	1	0.26	5	1.31
1107	1	0.26	6	1.57
1112	1	0.26	7	1.83
1113	1	0.26	8	2.09
1115	1	0.26	9	2.36
1117	1	0.26	10	2.62
1118	1	0.26	11	2.88
1121	1	0.26	12	3.14
1122	2	0.52	14	3.66
……..	…	…	…	…
9612	1	0.26	382	100.00

Alternatively, if this frequency were run at the end of data cleaning and we knew the unique identifier was fine prior to running the data cleaning syntax, we would then know that there is a mistake in the syntax that needs to be found and rectified. The first step is to search for all syntax that involves the unique identifier 1122 and examine it for mistakes. If we find a mistake, we can fix it, rerun the syntax on the original data, and then rerun the frequency of the identifier. If the frequency shows one case with the identifier 1122 then we are done. If, however, there are still two cases with the identifier 1122, then the next step is to look for any places in the syntax in which data are merged. Again, check the syntax for mistakes, correct any found, and, after rerunning the syntax, check a new frequency of the unique identifier. If the problem still has not been resolved, run

the syntax piecemeal, rerunning the frequency of the identifier after each small piece of syntax that has been run. Eventually the problematic piece of syntax will be isolated, and it can be corrected; then the identifier will once again become unique.

Now, this may seem like a lot of work. Why not just delete the extra 1122 case? Often the problem is not in the identifier—the identifier just lets us know there may be a deeper problem in the data. If we just delete the extra case, we will not know that there is another data error that has not been resolved.

COSMETIC CLEANING

Cosmetic cleaning takes place on each and every variable. As we did with the unique identifier, we run a frequency of each variable before and after any cosmetic cleaning is done. This allows us to see exactly how our syntax affected the variable and if the syntax did what we wanted it to do to the variable. Note that we cannot include string values and numeric variables in the same syntax code.

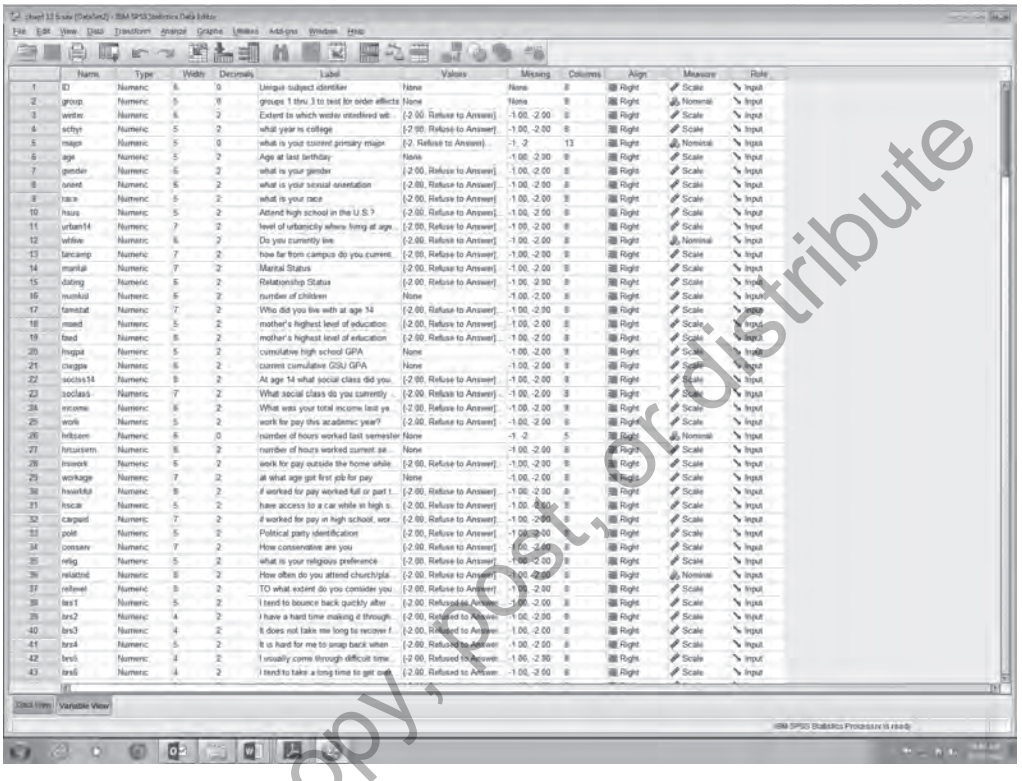
Variable Labels

In order to have a clear understanding of each variable, give each a variable name and variable label that make sense and that link it to the question on the questionnaire from which it originated. (See Chapter 12 for variable naming conventions.) The variable label is a title that is associated with a given variable name. It can describe the content of the variable. Some survey software will adapt the survey question into a variable label. If the software does not create a label, or if a pen-and-paper survey is used, researchers will have to create a label themselves. Once created, the variable label will appear in statistical output, making the output easy to interpret. The variable label will also appear in the variable view screen on SPSS. Figure 12.2 (Chapter 12) shows the variable view screen, and the variable label is visible.

Response Category Values

In Figure 13.2, in the column just to the right of the column showing labels, is a column headed "Values." *Values* here means value labels, which are the definitions given to each possible response category available for a given question. Value labels assign the wording found in the questionnaire for each response to the value or number found in the data. Each question on the survey will have response options. For example the question, "Have you given help with child care to one or more of your neighbors?" could have several possible response options. We could ask respondents to circle a 1 if the answer

Figure 13.2 SPSS Variable View of the Data

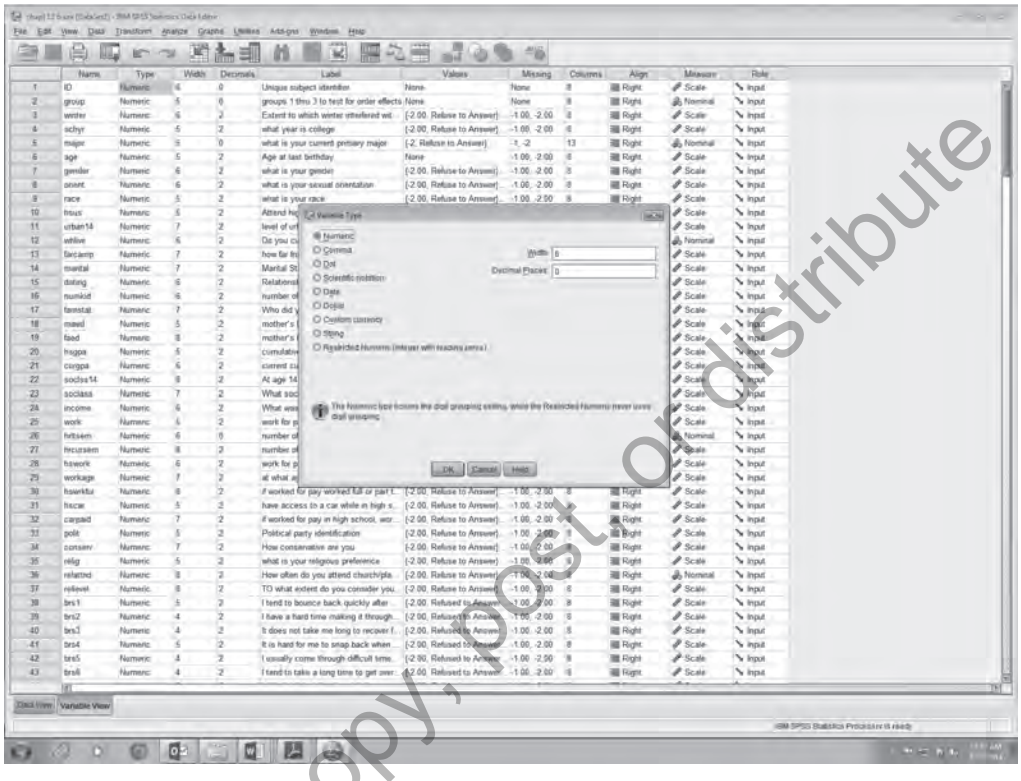


is yes, or a 2 if it is no. In the data, we would see only 1s and 2s. In order to make sense of the 1s and 2s, we would assign a value label of yes to the 1s and no to the 2. Once these have been assigned, make sure that all yes/no questions have the same values and value labels—document it thus, and even include a rule for this in a data cleaning manual. Consistency in setting up response category values such as no and yes is one way to make the dataset more user friendly.

Formatting Variables

Formatting variables is an important step in cosmetic cleaning. The *type* of variable should be designated ahead of time. The most basic format is the type of variable—string,

Figure 13.3 Types of Variables



numeric, or date. If it is designated as numeric, then no letters can be included in the values during data entry, only numbers. If it is a string variable, then all values—even numbers—will be treated like text and cannot be analyzed statistically without additional manipulation (being turned into a numeric variable). There are other types of variables, as can be seen in Figure 13.3. Date variables like string variables will need additional manipulation prior to being usable by a statistical program. The other types simply make understanding the numeric data easier. Choosing a particular type restricts the data being entered to that type. By restricting ahead of time, errors may be less likely to happen or may be more easily discovered.

Along with choosing the type of variable, the researcher must choose the width (known as length in some programs), and if necessary, allow for decimal places. In the past, when data storage was expensive, it was important to limit the width of variables to as small as possible. A question with a yes-or-no response really needs a width of only

1 character space. Reducing the width of variables reduces the file size of the overall dataset. If storage space is not a problem, allowing the software program to use the default setting is fine. If space is a problem, reduce the width of variables where possible.

Mini Case 13.2 provides a demonstration of cosmetic cleaning using three questions from a survey. In the data, we find that the corresponding variables are named Q1, Q2, and Q3. None of the variables has a variable label or value label, and all have a width of 8 characters with 2 decimal places. Given that these are not continuous variables, the decimal places do not make much sense. Given that Questions 1 and 2 are related (both are about smoking), our first decision is to better link these two questions through their variables names. We choose to rename Q2 to be Q1a, and note in the variable label that Q1a is Q2 on the questionnaire. The SPSS syntax for making these changes is shown below Mini Case 13.2. Note that comparable syntax for SAS and STATA are included in Table 13.5 at the end of the chapter.

MINI CASE 13.2

Examples of Survey Questions

This example assumes questions are being asked of 14- to 16-year-old high school students.

1. Have you ever smoked a cigarette, even just a puff, in your whole life?
   1. No (go to Question 3)
   2. Yes
2. How many packs of cigarettes have you smoked in the last week?
   3. None
   4. Less than 1 pack
   5. 1–2 packs
   6. 3–4 packs
   7. 5–6 packs
   8. 7 or more packs
   9. Don't know
   10. Not applicable
   11. Refuse to answer

(Continued)

(Continued)

3. Do you plan to go to college?

12. No
13. Yes
14. Undecided
15. Don't know
16. Not applicable
17. Refuse to answer

We start by taking a frequency of each variable. Table 13.1 shows a frequency of each variable. All we see is the question number and values. Thus, we have little information in the table. If we do not cosmetically clean the variables, we will have to go back and forth between output and survey in order to interpret our findings.

The first step is to rename variable Q2 as Q1a:

```
RECODE Q2 (MISSING=SYSMIS) (ELSE=Copy) INTO Q1a.
EXECUTE.
```

This code creates a new variable called Q1a and copies all the values from Q2 to Q1a exactly as they were. Next, we create variable labels for each of the three variables making sure to note that variable Q1a refers to Q2 on the questionnaire:

```
VARIABLE LABELS Q1 "Have you ever smoked a cigarette"
Q1a "(Old Q2) How many packs of cigarettes have you smoked in the last week"
Q3 "Do you plan to go to college?"
EXECUTE.
```

The next step is to assign value labels to the response values found in each question.

```
VALUE LABELS Q1
1 "No"
2 "Yes"
```

Table 13.1 Precleaning Frequencies of Variables

Q1

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	55	24.2	24.2	24.2
	2	172	75.8	75.8	100.0
	Total	227	100.0	100.0	
Total		227	100.0		

Q2

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	4	1.8	1.8	1.8
	2	145	63.9	63.9	65.7
	3	19	8.4	8.4	74.1
	4	3	1.3	1.3	75.4
	5	0	0	0	75.4
	6	1	.4	.4	75.8
	777	55	24.2	24.2	100.0
Total		227	100.0	100.0	100.0

Q3

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	38	16.7	16.7	16.7
	2	132	58.1	58.1	74.8
	3	14	6.2	6.2	81.0
	888	17	7.5	7.5	88.5
	999	26	11.5	11.5	100.0
	Total	227			
Total		227	100.0	100.0	100.0

```
VALUE LABELS Q1a
1 "None"
2 "Less than 1 pack"
3 "1–2 packs"
4 "3–4 packs"
5 "5–6 packs"
6 “More than 7 packs"
777 "Don’t know"
VALUE LABELS Q3
1 "No"
2 "Yes"
3 “Undecided”
888 "Don’t know"
999 "Refuse to answer"
```

Last, we format the type and width of the variables:

```
FORMATS Q1 Q1a Q3 ( f2.0).
```

This syntax is assigning a numeric format with 2 character spaces for each variable and zero character spaces after the decimal, meaning no decimal places. Table 13.2 shows the postcleaning frequencies. Now, there are descriptive variable labels and value labels that help us to interpret the data without having to go back to the survey.

For the most part, the numbers in the tables match up. However, there is one difference, and that is this: In the postcleaning frequencies, missing values have been assigned to the nonresponses. The next section will discuss missing values.

Missing Values

Being able to distinguish between a refusal, a “don’t know,” and a “not applicable” response may provide interesting information both about the respondents and about your questionnaire. For data analysis purposes, each type of missing response may be handled differently. For example, a “don’t know” response (coded as 888 in Table 13.1) could end up being a valid and usable response. A “not applicable” (coded as 777) is missing for a valid reason and also may be useable. But a refusal (coded as 999) is not ever usable, and a

Table 13.2 Postcleaning Frequencies of Variables

Ever Smoked a Cigarette

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No	55	24.2	24.2	24.2
	Yes	172	25.8	75.8	100.0
	Total	227	100.0	100.0	
Total		227	100.0		

(old Q2) Packs of Cigarettes Smoked in Past Week

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	None	4	1.8	2.3	1.8
	Less than 1 pack	145	63.9	84.3	65.7
	1–2 packs	19	8.4	11.1	74.1
	3–4 packs	3	1.3	1.7	75.4
	5–6 packs	0	0	0	75.4
	More than 7 packs	1	.4	. 6	75.8
	Total	172	75.8	100.0	
Missing	Not applicable	55	24.2		100.0
Total		227	100.0	100.0	100.0

Plan to Attend College

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No	38	16.7	20.7	16.7
	Yes	132	58.1	71.7	74.8
	Undecided	14	6.2	7.6	
	Total	184	81.0	100.0	81.0
	Don't know	17	7.5		88.5
Missing	Refused to answer	26	11.5		100.0
	Total				
Total		227	100.0	100.0	100.0

participant may have refused for any number of reasons. Therefore it is important to keep the various types of missing responses distinguished with a separate code for each. Variables with a single character (e.g. response options 1–5) can use a single-digit code for a missing value. For example, you might use 7 for a Likert scale item with five valid responses. But you might need 77 or even 777 for a variable like age (77 could be a valid age). The researcher must make sure for each question that the missing value assigned is not also a valid response. What codes are assigned to missing values may depend upon the survey software used, and missing value codes will have to be adjusted to fit with the statistical analysis software program used when cleaning the data.

How we handle **missing data** codes also depends on which statistical program is being used. Most statistical programs have a default "system missing" category for string variables, which is a blank space " ", and a dot " . " for numerical variables. SPSS Statistics, in addition, allows the researcher to specify up to three additional codes as missing *per variable,* meaning that different values can be assigned as missing codes for each variable. SAS and STATA allow an additional 26 missing value codes in addition to " . " Theirs include ".a" through ".z". SAS reads the missing codes as low values (less than zero), whereas STATA treats the missing codes as very large numbers—larger than your largest code. This information can be used to your advantage. In SAS, to avoid missing values on a variable, simply use the phrase "greater than or equal to zero" in your syntax. In STATA, the comparable code would specify "less than or equal to" the highest valid response value. If there are multiple users of the data who work with different software packages, handling the missing codes might be tedious.

If multiple people will be using the data with different statistical packages, you may want to create simple-to-use missing data codes that can be adjusted once they are translated into whichever statistical software package will be used. Negative numbers may be an excellent solution, because they are never used as real values in surveys, and they solve the SPSS Statistics problem of needing different values to mean a refusal for variables with large and small numbers of response categories. If you use negative numbers, -1 (-2, -3) can signify refusal ("don't know," "not applicable") for a Likert scale item, a yes/no item and an income item with up to 7 characters. Since the values will be consistent, the syntax to convert into "system missing" is very simple regardless of which software package is used.

Recall the three questions from Mini Case 13.2. For Question 1, those who have never smoked a cigarette will answer no. If they answered no to the first question, there is little point in asking them Question 1a (old Q2). Thus, nonsmokers will automatically be entered in the "not applicable" category for Question 1a. The syntax below assigns missing values in SPSS. Comparable SAS and STATA syntax can be found in Table 13.5 at the end of the chapter.

Question 3 from Mini Case 13.2 asks if the students plan to go to college. There is a valid "undecided" category, and there is a missing code "don't know" available to respondents. How is "don't know" different from "undecided"? Perhaps it is used by students who

might like to go to college but are not sure they will have the funds. In other words, they have decided yes, but do not know if it is possible. Perhaps 14-year-olds haven't even thought about college yet, so "undecided" does not quite fit for them. However, we do not know all the reasons older students may be "undecided" either. How much speculation do we want to do? Depending on the research interest, an investigator may decide that "don't know" and "undecided" are essentially the same category and lump the two categories together as valid. Or, the investigator may decide that "don't know" is a valid response and keep it as a valid category that is separate from the "undecided" category. We have no flexibility, however, if we simply lump all missing categories into the same code. Plan to spend some time thinking through what response values may mean to respondents and how to handle them. In general, if separate categories were used in the question, at cleaning it makes sense to keep them separate.

Since there were no missing values in Q1, below we write the syntax to assign missing values and then ask SPSS Statistics to treat them as missing codes. The first piece of syntax simply recodes the original 777, 888, and 999 codes used to capture nonresponse to the more user friendly -1, -2, and -3 codes respectively. The second line of syntax tells SPSS that for the variables Q1a and Q3, values of -1, -2, and -3 should be treated as missing response values. What is nice about this syntax is that it does not matter if each variable has all three types of missing values. It will not hurt the variable to assign a code that doesn't exist in the variable. Thus, in cleaning we can be consistent.

Recode *Q1a Q3* (777 = -1) (888 = -2) (999 = -3).

Missing values *Q1a* Q3 (-1,-2,-3).

SKIP PATTERNS

In the example provided in Mini Case 13.2, there was a **skip pattern**. In a skip pattern, one question acts as a gateway to answering future questions. So, Question 1 acted as a gateway to Question 2 (Q1a) and allowed only those who answered yes to Question 1 to see Question 2. In a web-based survey or interviewer-administered computer assisted survey, the participant will not see the skip. The programmed survey will automatically skip Question 2 if the participant says no to Question 1 (and the survey was programmed correctly). If the survey is in a self-administered pen-and-paper format, participants will see the skip pattern and may or may not follow it. Therefore, a researcher should check to see if skip patterns were handled correctly. This can be checked by running crosstabs on the gateway variable and any skipped variables. For a computer assisted survey or online survey, the skip pattern programming should have been verified to work correctly, and thus the data should reflect a skip program that has been followed correctly.

Table 13.3 Cross-Tabulation of Ever Smoked a Cigarette by Number of Packs Smoked in Last Week

Ever Smoked a Cigarette * Packs of Cigarettes Smoked in Past Week Cross-Tabulation

Count

		Packs of Cigarettes Smoked in Past Week							Total
		None	Less Than One	One to Two	Three to Four	Five to Six	More Than Seven	Not Applicable	
Ever smoked a cigarette	No	0	0	0	0	0	0	55	55
	Yes	4	145	19	3	0	1	0	171
Total		4	145	19	3	0	1	55	227

If skip patterns were not followed correctly, the researcher can manually skip all nonsmokers who did not follow the skip pattern by assigning the "not applicable" code to all the smoking items. First, run a crosstab on Q1 and Q1a. Syntax and output for SPSS Statistics are shown in this example, while SAS and STATA syntax can be found in Table 13.5 at the end of the chapter. The results are displayed in Table 13.3.

```
CROSSTABS
/TABLES=Q1 BY Q1a
/FORMAT=AVALUE TABLES
/CELLS=COUNT
/COUNT ROUND CELL.
```

As can be seen, the skip pattern worked perfectly. All 55 respondents who claimed never to have smoked a cigarette skipped Question 1a and are located in a missing value labeled "not applicable." If we are interested in explaining the smoking patterns of smokers, then we would be done. If, however, we are interested in explaining the smoking patterns of teenagers, then we would have to recode the 55 "not applicable" respondents into "none" respondents. We would use the following SPSS syntax:

```
IF (Q1 = 1) Q1a = 1.
```

MULTIPLE-RESPONSE, OTHER-SPECIFY, AND OPEN-ENDED QUESTIONS

Multiple-response, other-specify, and open-ended questions were discussed in depth in the data discussion in the previous chapter. If the data were entered into a statistical program through a web-based or CAPI program, then these variables need to be addressed in cleaning. Multiple-response variables will need no further work. Simply clean them cosmetically and check for missing values.

Other-specify options will be included in the data as separate variables. The example given in the data entry section was:

Q5 Would you say you are… (race)

(1) White

(2) Black

(3) Asian

(4) Hispanic

(5) Other (please specify _____)

All the respondents who chose 5 will have a valid text response to the variable "raceothr." But, what we want is a single variable that includes everyone from Q5 and everyone from raceothr. Therefore, the responses on raceothr will need to be recoded into Q5. First we read through the responses to determine if they belong in an existing Q5 category, if a new category needs to be added to the values and value labels of Q5, or if they should stay in "other." Respondents who write in "human" cannot be recoded into any category, and they can be left in the "other" response category. But responses of "Caucasian" or "multiracial," for example, can be coded. This is not an easy task—despite the easy example. People can use many ways to say the exact same thing. Therefore, you must *read the responses carefully several times* and pay careful attention to the *wording* of the original question.

The syntax below recodes cases of participants who chose Response 5 into an existing category and a new category. Remember that the value labels will have to be updated if new categories are added.

The SPSS syntax to recode is quite simple:

If (*raceothr* = "Caucasian") *Q5* = 1.

If (*raceothr* = "Multiracial") *Q5* = 6.

VALUE LABELS Q5

1. "White"
2. "Black"
3. "Asian"
4. "Hispanic"
5. "Other"
6. "Multiracial"

Finally, run a crosstab of the recoded original variable with the other-specify variable, and make sure there are no mistakes in the recoding syntax. Once you are sure the syntax worked correctly, then the other-specify variable can be deleted from the dataset (not from the raw data, however, just in case).

Open-ended questions will exist in the dataset in text form as well. Follow the instructions in Chapter 12 on how to code an open-ended item. Once it is recoded, the open-ended question can be recoded from the text to the new numeric codes. Again, run a crosstab of the recoded variable with the open-ended variable, and make sure there are no mistakes in the recoding syntax. Once you are confident the new recoded variable is correct, the original open-ended variable can be deleted from the dataset.

Notes: In some cases, respondents provide additional information. They will write comments in the margins of a pen-and-paper questionnaire, or an interviewer will take notes on additional spoken comments. About 20% of the time, the note or comment will provide information that requires us to change the original response. Consider this skip pattern example from the smoking question: A participant responds no and skips all the smoking questions. But in a side note the participant writes, "I haven't smoked in 20 years." This comment lets us know that the participant is a former smoker. We may want to change the response to the gateway question from no to yes. If our original question were worded, "Have you ever quit smoking?" we would change the response to yes. Or if it were worded "Do you currently smoke?" we would leave the answer as no. In the majority of cases, however, the notes or comments simply provide interesting context, and participants' answers do not need to be changed.

A problem with changing information or creating new variables based on the contextual information provided is that it is not standardized. Not everyone provided a comment, and therefore the information provided cannot apply to all respondents. While researchers may want to use the comments like they use open-ended question items, they simply cannot do justice to a new variable created from notes—the results will be biased.

What do we do when our syntax is complete? After all the issues are checked, and our syntax file is created, we run the syntax. We would like to think we are done at this point, but now we need to check and see what additional problems our syntax has

created. To check, simply run a frequency on each variable, and compare it to the frequency run on the raw, precleaned data. Frequencies of responses shouldn't change much, unless a variable was recoded. If we ran frequencies on the three variables from Mini Case 13.2, we would find a problem. Since we recoded our missing values from 777, 888, and 999 to -1, -2, and -3 in SPSS, we need to recreate our value labels to reflect that. The value labels do not include these new values. Once we are sure all the variables look exactly how we want them, it is time to turn to examining the variables diagnostically.

CLEANING FOR DIAGNOSTICS

Figure 13.4 provides a list of the issues you may want to examine diagnostically. This is called **diagnostic cleaning**. Here we are assessing the quality of the data, rather than making the data easy to use and understand.

We start with **implausible values**, which are values that simply cannot possibly exist in the data. This is a problem with noncomputerized surveys more than computerized surveys, because programmed surveys often will not allow responses that do not fit within a specified range. For example, consider a self-administered mailed survey in which respondents are asked a question with 5-point Likert scale response categories. In order to show a response that is "off the charts," a respondent puts a 7 into the scale and circles it. The data entry personnel do not notice this and simply type in a 7. When the frequency is run, and a sole 7 shows up without a value label for a question with 5 possible responses, the researcher has an implausible value. This example assumes the participant wanted to place emphasis on the response. But in reality, we cannot make such assumptions. Perhaps no missing values were allowed on this question, but earlier questions may have had a 7 for a "don't know" response. In general, implausible values must be turned into missing values, because we simply cannot know what the participant intended. We also want to handle all issues as consistently as possible.

Figure 13.4 Diagnostic Cleaning

1. Implausible values
2. Variation
3. Finding bad data through scale items
4. Mode of administration or interviewer effects

Variation

In order to statistically analyze our variables, there must be variation within each variable. Variables that produce minimal variation may not be validly measured and may need to be dropped. At a minimum, the lack of variation should be remarked upon in the codebook (see last section of this chapter).

Along with frequencies, the means and standard deviations of all the continuous, ordinal, and dummy variables should be assessed to determine the level of variation. The mean is an average or measure of central tendency. The standard deviation is a measure of the amount of dispersion about the mean. If we wanted to know if there was variation by age in a sample, we could run a mean on the age variable and ask for the standard deviation. These two statistics tell us everything we want to know about how much variation is in a variable. If the data about age collected in the 2003 wave of the Wisconsin Longitudinal Study (1957–2005) were analyzed in this way, they would give a mean of age 65 with a standard deviation of less than 1 year. These statistics show there is little variation by age in this study. In the Atlanta Public Housing Study (Oakley, Ruel, and Wilson 2008; Ruel, Oakley, Wilson, and Maddox 2010), the sample ranges in age from 18 to 96. The mean or average age for the sample is 51 years, and the standard deviation is 17.3 years. There is a great deal of variation in age in this sample. Thus, age could be used in analyses of the Atlanta study but not the Wisconsin study.

Bar charts are useful for visualizing the distribution of ordinal and nominal variables. They help the researcher to determine how to handle these variables later in analyses. Ordinal variables may show little variation across the responses. Rather than using all 5 options in a Likert scale, the majority of users may have only used two categories—the "agree" and "disagree" options, for example. If it's not part of a multi-item scale, it may make sense to transform the variable into a dichotomous variable by collapsing categories. Or, if only one or two respondents chose the "strongly disagree" option, it may be an outlier response. Then it may make sense to turn the variable into a four-category variable by top coding the ordinal variable to a "disagree" maximum allowed response. Figure 13.5 provides a bar chart of a Likert scale item that asks respondents to agree or disagree with the statement, "This neighborhood is a good place to raise kids." While most respondents either agree or disagree, there are quite a few who strongly disagree, suggesting that there is good variation across this variable.

Finding Bad Data Through Scale Items

These are a series of questions that together represent a single unobservable construct. Respondents might complete the survey but will not necessarily answer truthfully. If it's a long survey, they may get bored and just mark the same response for all variables. For example, they may choose option 2 as a response option for every variable after the 20th

Figure 13.5 Bar Chart of an Ordinal Variable

question. Scale items are useful for locating these respondents, particularly if some of the questions in the set of scale items are reverse coded. **Reverse coding** refers to inserting a positively worded item in among negatively worded items, or vice versa. For example, if most of the items in a questionnaire mention positive things about a neighborhood, such as it's a good place to raise kids, reverse-coded items would focus on negative neighborhood aspects. An example of such an item would be, "People in this neighborhood do not share the same values." Thus, if 2 usually means "agree" on the reverse-coded item, a consistent response would be 4, for "disagree." The reverse-coded item will capture inconsistent cases. You will want to create a crosstab of all items in the scale to search for these types of inconsistencies. SPSS will not allow you to do this, so if you are working in SPSS, copy the scale items into Excel, sort by each item, and examine which cases or respondents give the same responses throughout the scale. Once you locate them, you can assess their entire answer set to determine if they gave bad data. If they did, it makes sense to delete them from the dataset.

INTERVIEWER EFFECTS AND MODE EFFECTS

A major benefit of surveys is that they are standardized; all participants receive the same question in the same way. Thus, the only thing that distinguishes participants from each other is their own personal characteristics. If we use a single mode of survey administration, such as interviewing, but we use multiple interviewers, we may find that differences among the interviewers lead to significant differences in answers to questions. This is called an **interviewer effect**. Another way of introducing differences in responses is to administer a multimodal survey—reaching some respondents by phone and others through the Internet, for example. If there is a statistically significant difference in responses to questions administered by different methods, then the responses are no longer standardized. There is now another difference between the groups—the mode of administration is added to the potential interviewer effect of the phone portion of the survey. This is called the **mode effect**.

It is simple to assess whether or not there are interviewer effects. First, in the same way we create a unique identifier for each respondent, we can create an identifier variable for each interviewer. Then, test each variable in the dataset by either running an **analysis of variance (ANOVA)** with continuous variables, or a **chi-square test** of independence with ordinal and nominal variables. If either test is found to be statistically significant (see Chapter 14), then there are interviewer effects on those variables. If there are interviewer effects on all the variables, it may be that there is an outlier interviewer; in this case, controlling for that interviewer (using a dummy variable) in all regression analyses will solve for that problem.

If there are a large number of interviewers, this is not a practical solution. Instead, think about how characteristics of the interviewers might interact with characteristics of the sample. For example, if the sample consists of older adults, younger interviewers might not build as good rapport with the sample as older interviewers do. For this, we can create a dummy variable that is 1 for older interviewers and 0 for younger interviewers. Then we can conduct *t*-tests for continuous variables and chi-square tests for ordinal and nominal variables in the dataset to determine if the age of the interviewer makes a difference in responses. Again, if the results are statistically significant, there are interviewer age effects in the data.

For mode effects, we can create a dummy variable (1 = mode of administration 1) and 0 = mode of administration 2) and run ANOVA or chi-square tests of independence on all of the variables to see if there are statistically significant differences in responses to questions based on the mode of administration. Again, if we find differences on some variables, we need to control for the mode effects by including the mode of administration dummy variable in our analyses.

In our example we run an ANOVA. The ANOVA tests the hypothesis that the mean effect of each group (of interviewers) on the substantively interesting variables is all the

Table 13.4 Analysis of Variance (ANOVA) to Test for Interviewer Effects

ANOVA

		Sum of Squares	*df*	Mean Square	*F*	Sig.
This neighborhood is a good place to raise kids	Between groups	85.203	46	1.852	1.020	.442
	Within groups	591.880	326	1.816		
	Total	677.083	372			
People around here are willing to help neighbors	Between groups	68.351	46	1.486	1.025	.434
	Within groups	479.860	331	1.450		
	Total	548.212	377			
People in this neighborhood generally don't get along with each other	Between groups	51.263	45	1.139	.789	.833
	Within groups	467.969	324	1.444		
	Total	519.232	369			
People in this neighborhood can be trusted	Between groups	51.420	45	1.143	.797	.822
	Within groups	470.454	328	1.434		
	Total	521.874	373			
People in this neighborhood do not share the same values	Between groups	56.446	45	1.254	.984	.506
	Within groups	411.722	323	1.275		
	Total	468.168	368			

same, that is, that there is no interviewer effect on how respondents answered the questions. Results are presented in Table 13.4 The columns in the figure provide the sum of squares, which tells us how much of the variance in each variable is broken out among the grouping variables and within each grouping variable, in this case the interviewers. The column labeled df refers to degrees of freedom; this tells us how many independent pieces of information exist, that is, how many pieces of information the *F* test is based on. Generally, there are $n - 1$ degrees of freedom, where n = sample size. The *F* test is the omnibus test of association or whether or not the mean across the interviewers is the same. The column labeled "Sig" provides the level of significance of the test. Usually we set the significance level at .05, meaning that if significance is .05 or higher, then there are no significant differences among the groups, or in this case, there are no interviewer

effects. If the significance level is less than .05, then there are significant differences, we reject the hypothesis, and we have interviewer effects.

The SPSS Statistics syntax for an ANOVA that checks for interviewer effects is presented below. Here we are assessing interviewer effects on five variables representing neighborhood social cohesion in the Atlanta Public Housing Study (Oakley, Ruel, and Wilson 2008; Ruel, Oakley, Wilson, and Maddox 2010). Given that the values in the significance column are much larger than .05, we can conclude that there are no interviewer effects on these variables.

```
ONEWAY Q1aW1 Q1bW1 Q1cW1 Q1dW1 Q1eW1 BY IntID
/MISSING ANALYSIS.
```

CLEANING LONGITUDINAL DATA

For panel data that come from a longitudinal study in which the same respondents are interviewed multiple times, the cross-sectional cleaning and the diagnostic cleaning need to be conducted after each wave of data collection. In addition to that, there are a couple of additional cleaning checks that are needed. These are presented in Figure 13.6. The first is **consistency in coding**, and the second is **attrition effects**. Attrition was introduced in Chapter 9; it is a special case of missing data.

Consistency in Coding

We are including several issues in this one category. First, we need to check if there is consistency in response categories over time (Van den Broeck et al. 2005), meaning, in the creation of the survey, did we change the response categories for any questions? If so, we need to document that, because clearly responses will be different over time due to changes in the questions themselves. Another change that is not substantive—it is simply a formatting-the-survey change—takes place when we assign new values to the same old responses. For example, if in Wave 1 we set yes to be 1 and no to be 0, then we

Figure 13.6 Cleaning Longitudinal Data Files

1. Consistency in coding
2. Attrition effects

must make sure to use the same response categories in all later waves of data. If in the next wave, yes is assigned to 2 and no is assigned to 1, then comparing the mean across waves will show a large difference. If we were to check the means of variables in Wave 1, Wave 2, and Wave 3, for example, we want any change we find in the average value to be due to real change experienced by respondents and not because we changed the numeric values of the response categories.

One of the goals of longitudinal analysis is to examine change over time. To do this, items on the survey need to be repeated exactly. If the question wording is changed or a new response category is added in later waves, this means items are not repeated over time. If new response categories are added in later waves, make sure they are not treated as implausible values. Make sure these changes are clear in the data and in the codebook (see the section below on the codebook).

Compare means and standard deviations on the same items from each wave to see if things make sense in general. If either the mean or standard deviations differ greatly, there may be a problem with one of the variables that should be examined closely. Make sure you have noted which variables have added new response categories, as this will create a change in the mean and possibly the standard deviation between one variable in a pair and the other.

Check to see how respondents handle some gateway questions (skip patterns). Over time respondents learn how to work the survey. If answering yes to a gateway question means having to answer a long set of questions, fatigued respondents may respond with a no in Wave 3 or 4, where they responded yes in earlier waves. For example, a respondent said yes to Question 1 (Have you ever smoked a cigarette?) in Wave 1, and then had to answer a series of questions on smoking behaviors. In Wave 2, the respondent may change the answer no to Question 1, and then skip the series of questions on smoking behaviors. Other gateway questions may not be so cut and dried. People rarely forget they used to smoke, but attitudes can change, and people may not remember having had different attitudes in the past.

Attrition

Attrition refers to data loss that takes place when respondents decide to no longer participate in a panel study; they drop out over time. Respondents may be missing for other reasons that could affect the quality of the data as well. They may die between waves of the study or become institutionalized in prisons or nursing homes. These forms of missing data may make participants ineligible for one or more waves of data collection. In addition, people may move or change residences between waves of data collection. The researcher may not be able to find them for one wave of data collection but may be able to bring them back into the study at a later time. In most longitudinal studies, attrition is quite large, and the dataset quickly loses its representativeness for the population

from which it was drawn. In other words, findings from research conducted on the data could be biased if attrition is more likely to occur within some subgroups of the population than within others. In fact, Wood, White, and Thompson (2004) have found that in many clinical trials missing data due to attrition on the dependent variable is typically problematic and often mishandled.

In order to assess how representativeness of the sample has changed over time due to attrition, we can compare descriptive statistics (measures of central tendency and dispersion) of demographic characteristics between the Wave 1 dataset and the remaining sample at some future wave. It may be that in the future wave dataset, some age groups, races, or socioeconomic classes will be overrepresented, while others will be underrepresented. We will explore this in depth in Chapter 15 with an example.

THE CODEBOOK

Most computer assisted survey programs will generate a text document, or **codebook**, that matches variable names to the questions on the survey. This is the most important function of the codebook. However, really good codebooks will provide additional information. Long ago, when most secondary datasets were downloaded, they were in a format called ASCII, which is a format that can be easily read into any statistical software package using syntax. ASCII files take up little space on computers; this was an important issue in the not-too-distant past. To ensure that the syntax created to read in the data worked correctly, a researcher would run frequencies on the variables and compare them to frequencies provided in the codebook. Thus, in the past, codebooks often provided material to help researchers make sure the data they were using was correct. Even today, downloaded data can be corrupted without the researcher knowing it, so comparing downloaded data against a codebook is still a good idea.

Other information that was beneficial to secondary users of data was often included in a codebook as well. An example is a list of all the variables skipped by a gateway variable. For each variable skipped, there would be information on what the gateway variable was and how many items were skipped. Current researchers can replicate this resource by providing, for each item on a survey, an eligible list of respondents.

For longitudinal data, a codebook that provides a cross-walk between waves of data is very helpful to all users. In addition, given the advances in web capacities, there are many interactive searchable online codebooks. Searching them is an extensive undertaking and not efficient unless your study is an ongoing, long-term endeavor. Mini Case 13.3 provides an example of a codebook entry from a very well-documented study, the Wisconsin Longitudinal Study (1957–2005). It is time consuming to create codebooks, so reserve some resources for this purpose.

MINI CASE 13.3

Example of a Codebook Entry

The Wisconsin Longitudinal Study (WLS 1957–2005) has about the best data documentation available. Here is an example of a variable from one of the waves of data collection. It includes the variable name, variable label, the source of data—meaning who responded—the year of data collection, and the mode of survey administration. This is a variable created out of responses to three other variables. This information allows the investigators to examine the original variables if they want or need to. Next, a frequency by sex is provided. Last, a note is provided to explain how the variable was generated from the three source variables. Additionally, a lack of consistency in responses over time has been made clear. See the WLS webpage for more examples: http://www.ssc.wisc.edu/wlsresearch/documentation/.

gb001re: Has graduate attended college?

Data source: Graduate respondent. Collected in: 2004. Mode: phone. Source variables: b3a, rb001re, edexpr

		Frequencies		
Value	*Label*	*Male*	*Female*	*Total*
	System missing - NR	1621	1431	3052
–1		1	7	8
1	Yes	1883	1785	3668
2	No	1486	2103	3589

Note: Respondents who said anything but yes in 1993 and refused (b3a = -3) in 2004 were categorized as refused. Respondents who said anything but a definite yes in 1993 and said "don't know" in 2004 (b3a = -1) were categorized as "don't know." As in 1993, if a respondent said yes in 1993 and changed his/her mind by 2004, the 2004 response was categorized as a yes.

Source: Wisconsin Longitudinal Study.

CONCLUSION

Cleaning data is a time intensive and, occasionally, tedious task. In many ways researchers can be considered social forensic scientists because of how they systematically collect and clean the data. Good cleaning will make the analysis of the data a much easier

process. But it also provides an excellent introduction to your data, meaning that it helps you to see what the data look like prior to beginning data analysis.

We introduced three types of data cleaning. The first we called cosmetic cleaning, as its purpose is to make examining each variable easy on the user. If we add good variable names, variable labels, and value labels, we do not need the codebook or survey instrument to remind us what the substance of a particular variable may be. Part of cosmetic cleaning involves coding text variables such as other-specify responses and responses to open-ended questions. Working on the codebook at the same time we are doing the cosmetic cleaning will ensure that the data and the codebook are consistent.

The next two sets of cleaning are about assessing the quality of the data through checking diagnostics and attrition on single variables. The more diagnostics we check, the better we can argue that the quality of the data is high. We will extend this work to examining diagnostics on pairs of variables in the next section.

Give yourself plenty of time and resources for this part of the investigative process. Data entry, cleaning, and documentation are integral to the research process, and time and staff are necessary to complete these steps effectively.

SYNTAX FOR ANALYSES DESCRIBED IN THIS CHAPTER

All the SPSS syntax commands used in this chapter are included in Table 13.5. In addition, it includes syntax for use with SAS and STATA.

Table 13.5 Syntax for Chapter 13 Analyses

	SPSS	*SAS*	*STATA*
Accessing data	Get file "c:\my documents\survey\ data.sav".	Libname **da** "c:\my documents\survey"; Data.data; (temporary dataset name) Set **da**.data; run;	Use "c:\my documents\survey\ data"
Frequency procedure	Frequency variables=subject.	Proc freq data=da. data; Tables subject; run;	Tab subject

	SPSS	*SAS*	*STATA*
Renaming variables	RECODE Q2 (MISSING=SYSMIS) (ELSE=Copy) INTO Q1a.	Data da.newdata; Set da.olddata; Q1a=Q2; Run;	Compute Q1a=Q2.
Assigning variable labels	VARIABLE LABELS Q1 'Have you ever smoked a cigarette'.	Data da.newdata; Set da.olddata; Attribute Q1 Label= 'Have you ever smoked a cigarette'; Run;	Label Q1 "Have you ever smoked a cigarette"
Assigning value labels	VALUE LABELS Q1 1 "No" 2 "Yes"	Proc format library =fmt; Value Q1f 1 "No" 2 "Yes"; Data da.newdata; Set da.data; Format Q1 Q1f. Run;	Label define Q1f 1 "No" 2 "Yes" Label values Q1 Q1f
Formatting variables	FORMATS Q1 Q1a Q3 (f2.0).	Data da.newdata; Set da.olddata; Attribute Q1 length=2; format=2.0; Q1a length=2; format=2.0; Q3 length=2; format=2.0; Run;	Format Q1 2.0 Format Q1a 2.0 Format Q3 2.0

(Continued)

Table 13.5 (Continued)

	SPSS	*SAS*	*STATA*
Assigning missing values	Recode *Q1a Q3* (777=-1) (888=-2) (999=-3). Missing values *Q1a* Q3 (-1,-2,-3).	If Q1a=777 then Q1a=.n; If Q3=888 then Q3=.d; If Q3=999 then Q3=.r;	Replace *Q1a*=.n if *Q1*==1 Replace *Q3*=.d if *Q3*==888 Replace *Q3*=.d if *Q3*==999
Crosstabs procedure	CROSSTABS/ TABLES=Q1 BY Q1a /FORMAT=AVALUE TABLES /CELLS=COUNT /COUNT ROUND CELL.	Proc freq data=da. newdata; Tables Q1*Q1a/ list; Run;	Tab Q1 Q1a, all
Recoding variables	If (Q1=1) Q1a=1.	If Q1=1 then Q1a=1;	Replace Q1a=1 if Q1==1
ANOVA	ONEWAY Q1aW1 Q1bW1 Q1cW1 Q1dW1 Q1eW1 BY IntID /MISSING ANALYSIS.	Proc anova data=da. newdata; Class intid; Model Q1aw1 q1bw1 q1cw1 q1dw1 q1ew1 = intid; Run;	Oneway Q1aw1 q1bw1 q1cw1 q1dw1 q1ew1 by intid / statistics homogeneity /missing analysis.

KEY TERMS

CRITICAL THINKING QUESTIONS

Below you will find three questions that ask you to think critically about core concepts addressed in this chapter. Be sure you understand each one; if you don't, this is a good time to review the relevant sections of this chapter.

1. Why is it important to clean the data using syntax?
2. How might our data-cleaning needs change if we are doing a multicountry survey?
3. How do we train a staff of four to five people to clean the data? What do we need to think about to ensure the data are cleaned consistently?