Chapter 12

Getting Familiar with Your Data

In This Chapter

- Organizing data properly
- Importing data
- Examining your data
- Knowing data-mining terminology

Before a French chef whips up a dazzling dish, she sets out all the ingredients and tools. She checks that the ingredients are fresh and good, and that the tools work properly. She does not begin to cook until she puts everything in place.

A data miner is no different. Before you whip up a dazzling predictive model, you get acquainted with the data that you will use. You put it where you need it. You make sure that you understand what data you have, how it's arranged and stored, and whether it is complete and correct.

This chapter shows you how to analyze and evaluate your data.

Organizing Data for Mining

Data mining has very strict requirements for data organization. They are not exotic, complex, or difficult requirements to meet, but they are strict.

Let me use an example to show how data must be organized for data mining. Figure 12-1 shows a sample of data viewed as a table in data-mining software. (See Chapter 2 for more about this data and an example data-mining application.) Each row represents one parcel of real estate. Information about the parcels of real estate is organized in columns. The first column contains the tax identification number (TAXKEY), the second column contains the assessed value of the land from a prior assessment (P_A_LAND), and so on. Every entry in any one row pertains to one specific parcel of land. Every

entry in any one column is the same type of information. No rows or columns are left blank for reasons relating to style and readability. This data is properly organized for investigating differences among the parcels of real estate.

ExampleSet (162403 examples, 0 special attributes, 58 regular attributes)03 / 162,403 examples): all

Row No.	TAXKEY	P_A_LAND	NR_UNITS	C_A_LAND	LAND_USE	C_A_CLASS	C_A_TOTAL	CHK_DIGIT
1	10001000	48200	1	48200	8810	1	229600	3
2	10011000	146200	0	150700	5093	3	602800	8
3	10021000	115000	0	115000	1794	2	384000	2
4	10022000	0	0	0	8880	9	0	8
5	18100000	100	0	100	6	4	37000	7
6	18101000	100	0	100	6	4	37000	2
7	19989000	0	0	0	4010	9	0	X
8	19990000	0	0	0	4010	9	0	5
9	19991000	0	0	0	4010	9	0	0
10	19992100	40600	0	40600	4010	2	40600	2
11	19996100	53400	6	53400	8830	7	179600	4
12	19996210	0	0	0	8885	9	0	8
13	19998200	47800	1	47800	8810	1	153700	1
14	19999100	139700	0	139700	4225	2	268800	0
15	20032000	495700	0	495700	5171	4	15729000	X
16	20051000	204100	0	204100	5171	4	475000	3
17	20052000	114700	0	114700	4225	4	120000	9
18	20071100	0	0	1734900	5172	4	12638000	9

Figure 12-1: Data organized properly for data mining.

If, instead of real estate, you investigate people, each person would be represented by one row in the data, and all the details about the people would be organized into columns. If you investigate chest x-rays, each chest x-ray would be represented by one row in the data, and all the details about the chest x-rays would be organized into columns. In data analysis terminology, the things you're studying — the things in the rows — are called *cases* or *records.* And the details about them, which are in the columns, are called *variables*. You will also hear the columns called *fields,* especially in the context of databases.

So, data mining requires data organized with a single row for each case and a single column for each variable. Many sources of data are already organized in this way. Statisticians organize data this way by habit. Database professionals may not use this approach for much of their work, but they'll usually understand what you want if you call it a *flat table.*

You'll find subtle variations in data structure. Some types of software use descriptive information in a header before the data, such as certain specialty formats associated with the Orange and Weka data-mining applications. Some complex analytic procedures have additional or slightly varied requirements (these are quite unusual). But the core of the data still has the cases in rows and variables in columns.

Getting Data from There to Here

Your first hands-on step with data is getting it from wherever it is to the place where you need it to be.

The steps you will take to import data for use in data mining can vary a lot from one situation to another. Your own skills, your work style, company policies and procedures, and the specifics of any particular project may affect the way that you go about accessing data. The most important influences include

- **Data format:** The format the data is in. Examples include relational database, NoSQL database, text file, spreadsheet, XML, or others.
- **Data organization:** The structure of your data. The data structure may be convenient for data mining (and your particular project) or not.
- **Software:** Each product has its own procedures for importing data, and variations exist, even within a single product.

Text files

Text formats are common, and you're likely to encounter them often. You'll find several varieties, but some of the most common are *comma-separated value (`.csv`), tab-delimited,* and *fixed-column* text. Most public data sources, including government sources and nonprofit agencies, offer data as text files. Many researchers love text files because they are not tied to specific products or platforms, and they are compact (that is, they use minimal space for the data they contain). Here's the news about text files:

- **The good news:** Every data-mining application can import data from text files.
- **The bad news:** Every data-mining application has its own way of importing data from text files, and some of them are pretty challenging to use.
- **Even worse news:** Some data-mining applications can import some kinds of text files but not others.

Consider an example. Figure 12-2 shows data in a text file. The data is in `.csv` comma-separated value format. The first row contains variable names, separated by commas. All the other rows contain the data, one row for each brand of cigarettes. The data includes the brand name, the region where it is sold, tar content, and other variables. These values are separated by commas. This data is well organized for data mining. What's the process for opening this data?

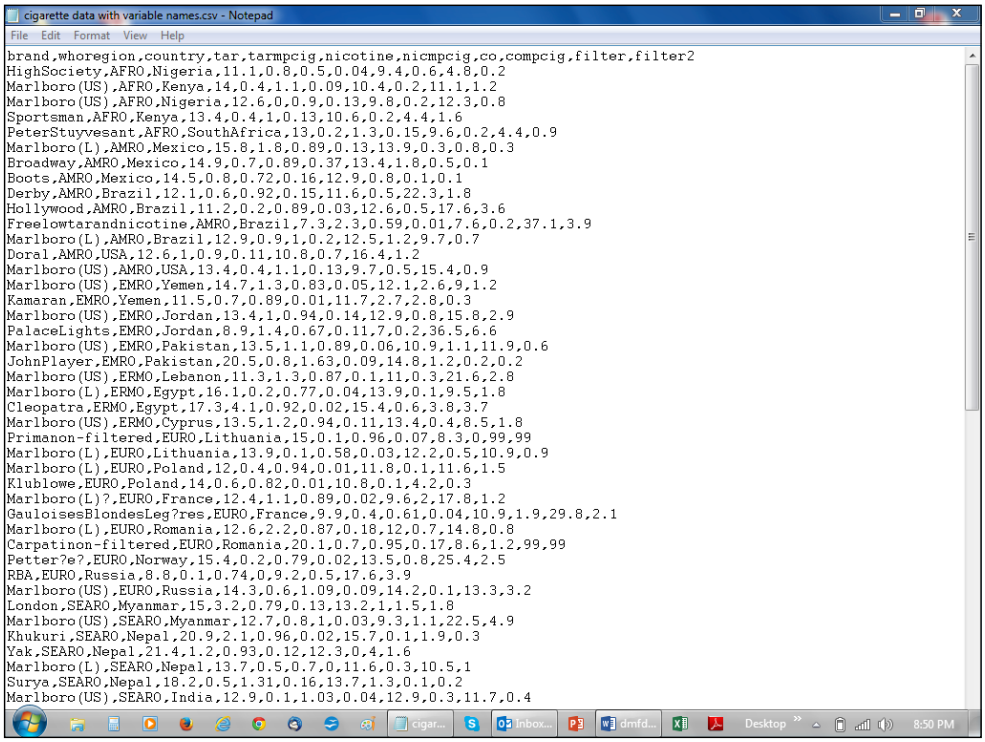

Figure 12-2: Sample data in a text file.

Here's how it's done in four example data-mining applications. Review these procedures, and you'll start to understand how these applications look and how they are used.

To open the sample data in KNIME (see Appendix C for information about KNIME and where to get it):

1. **Start KNIME. (See Figure 12-3.)**
2. **Find the CSV Reader in the Node Repository (a menu). It's grouped with other tools for importing data. (See Figure 12-4.)**
3. **Drag the CSV Reader to the work area. (See Figure 12-5.)**
4. **Right-click and select Configure. Browse to find the cigarette data. (See Figure 12-6.)**
5. **Adjust settings. Make sure to select the proper delimiters (commas) and indicate that column headers (variable names) are in the first line of data (See Figure 12-7.)**
6. **Click the Execute button (shown in Figure 12-3) to import the data.**

   The CSV Reader will show a green indicator (see Figure 12-8, bottom right) when the data has been imported.

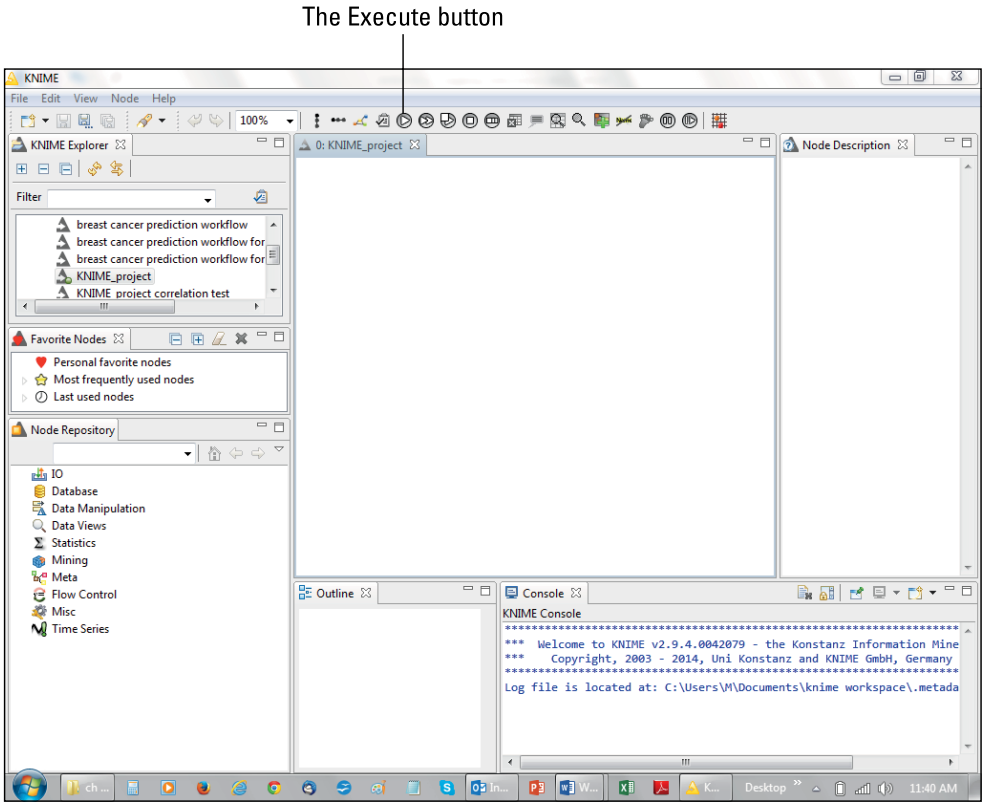

Figure 12-3: KNIME.

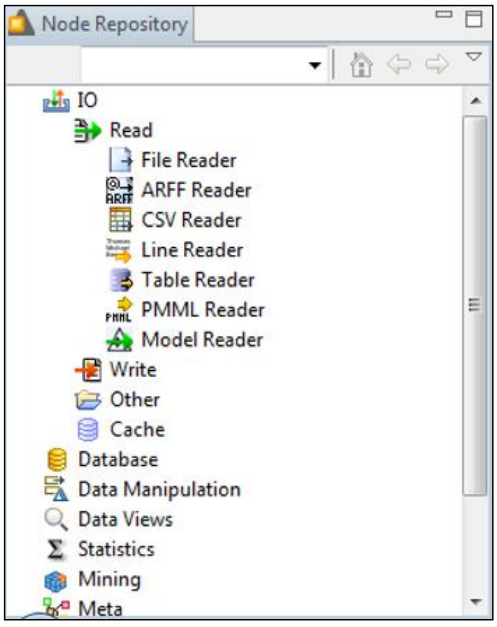

Figure 12-4: Finding the CSV Reader in KNIME.

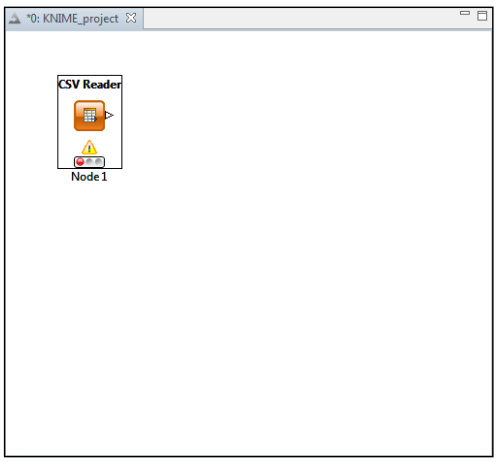

Figure 12-5: CSV Reader in the KNIME work area.

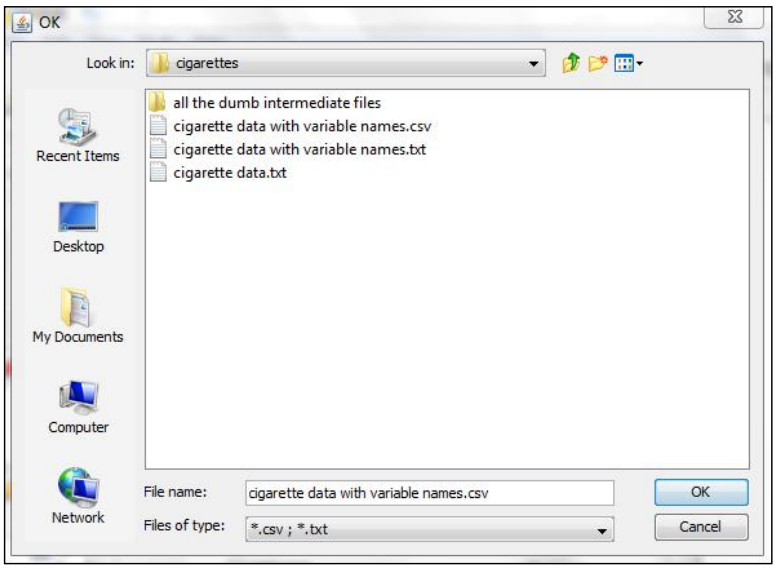

Figure 12-6: Browsing to find the cigarette data in KNIME.

To open the sample data in Orange, follow these steps (see Appendix C for information about Orange and where to get it):

1. **Start Orange Canvas. (See Figure 12-9.)**
2. **Find the File widget. It's in the Data group, the only data import tool. (See Figure 12-10.)**
3. **Click the File widget once to place it on the work area. (See Figure 12-11.)**

4. **Right-click and select Open. Browse to find the cigarette data. (See Figure 12-12.)**

   Oops! The drop-down list of file types doesn't offer an option for the `.csv` format. You'll have to convert the data to another format before you can open it in this data-mining application.

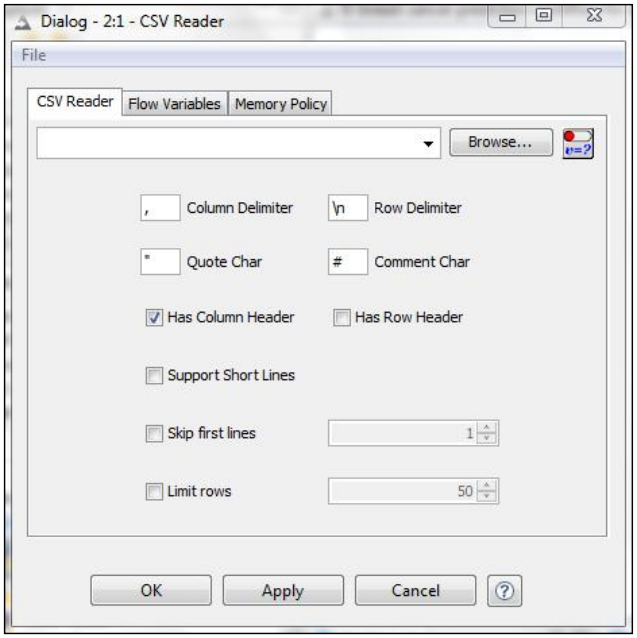

Figure 12-7: Adjusting import settings in KNIME.

Figure 12-8: KNIME CSV Reader after importing data.

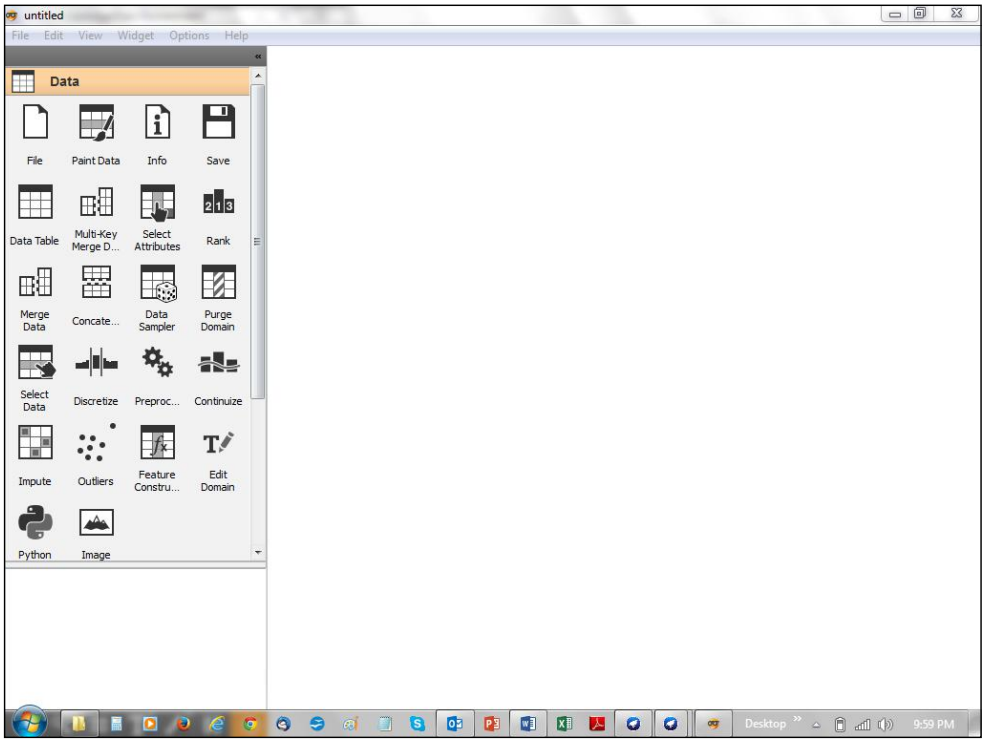

Figure 12-9: Orange Canvas.

To open the sample data in RapidMiner, follow these steps (see Appendix C for information about RapidMiner and where to get it):

1. **Start RapidMiner Studio. (See Figure 12-13.)**
2. **Find the Read CSV operator. It's grouped with other tools for importing data. (See Figure 12-14.)**
3. **Drag the Read CSV operator to the work area. (See Figure 12-15.)**
4. **Click the Read CSV operator. Settings for the Read CSV operator will be displayed in the Parameters area (See Figure 12-16.)**
5. **In the Parameters area, click the Import Configuration Wizard button and use the wizard to browse for the cigarette data. (See Figure 12-17.)**
6. **Adjust settings. The wizard gives you cues to help get the settings right. (See Figure 12-18.) Click the Finish button to return to the work area.**

7. **Click the Execute button (shown in Figure 12-13) to import the data.**

   The Read CSV operator will show a round green indicator (see Figure 12-19, lower left) when the data has been imported.

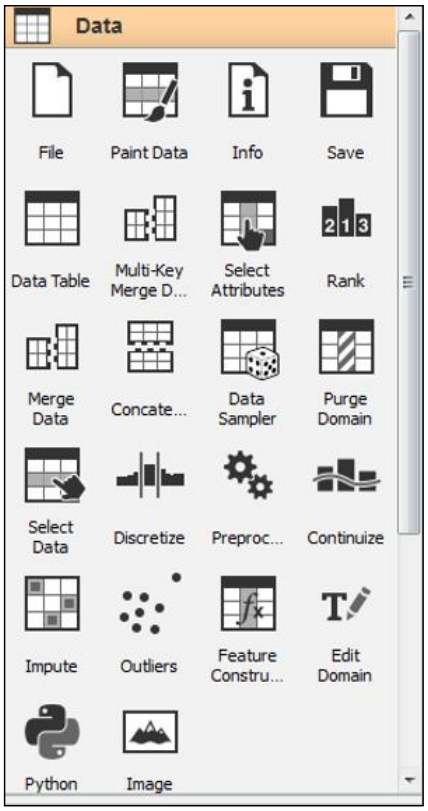

Figure 12-10: Finding the File widget in Orange Canvas.

Figure 12-11: Placing the File widget on the work area.

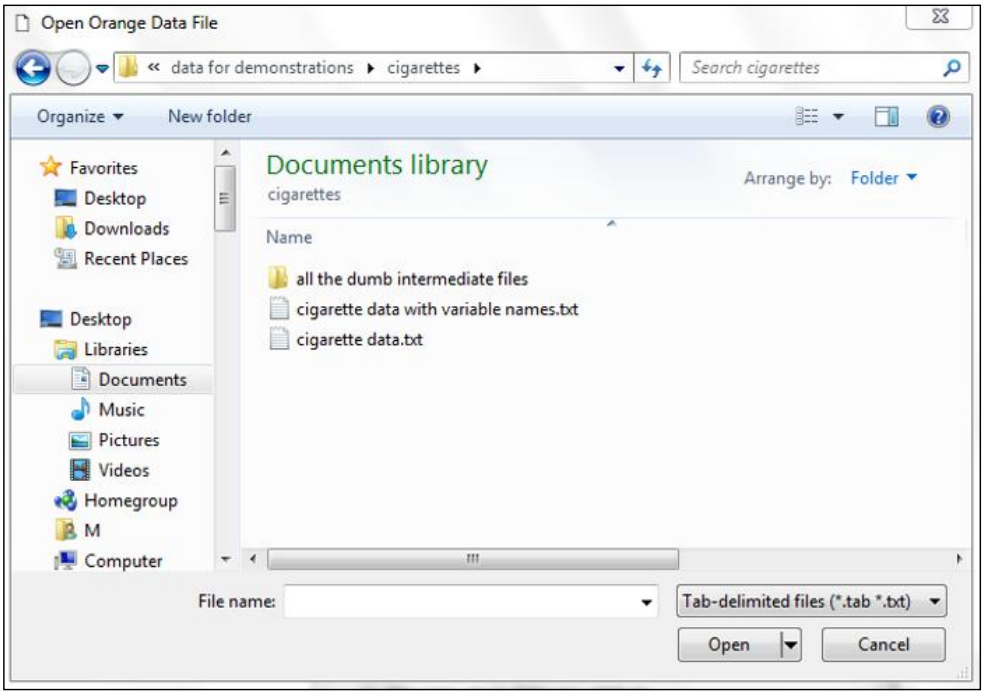

Figure 12-12: The list of file types in Orange Canvas doesn't include `.csv`.

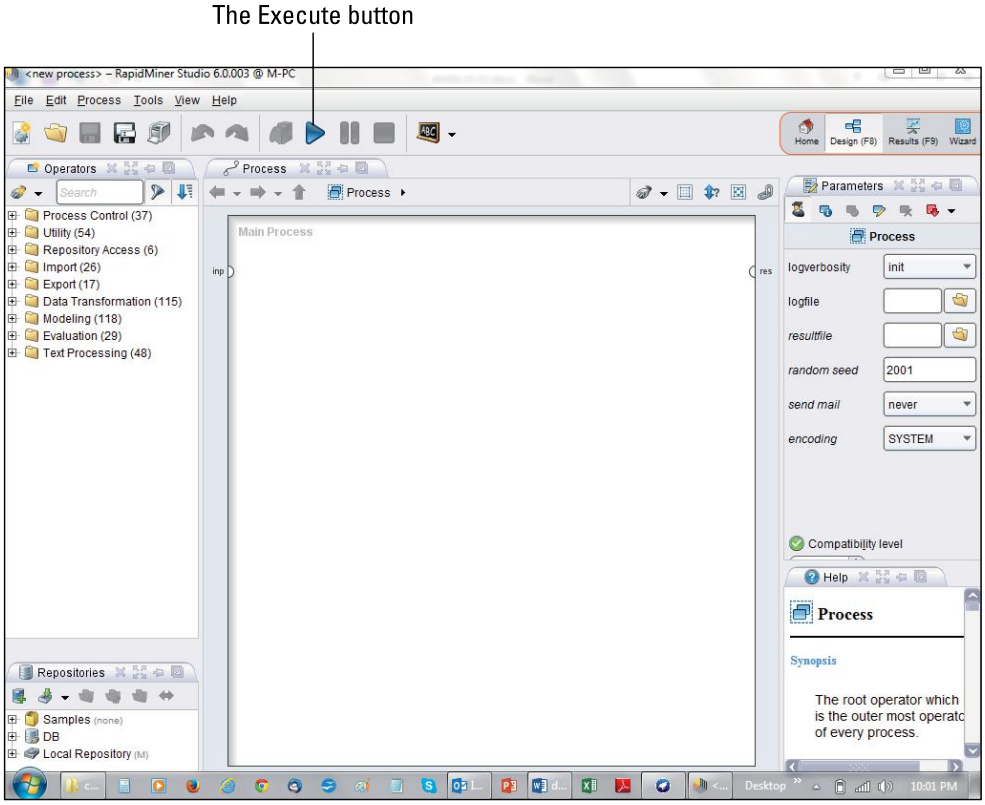

Figure 12-13: RapidMiner Studio.

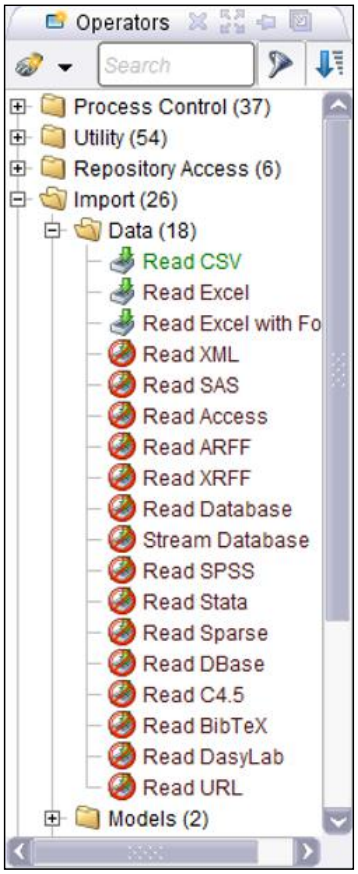

Figure 12-14: Finding the Read CSV operator in RapidMiner Studio.

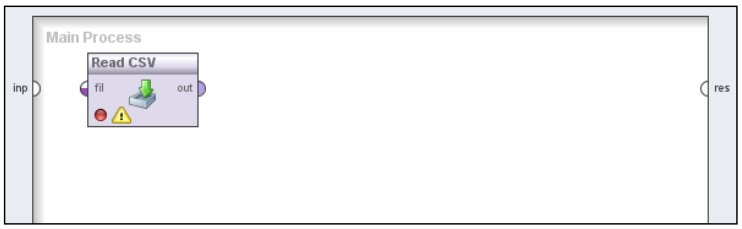

Figure 12-15: Dragging the Read CSV operator to the RapidMiner Studio work area.

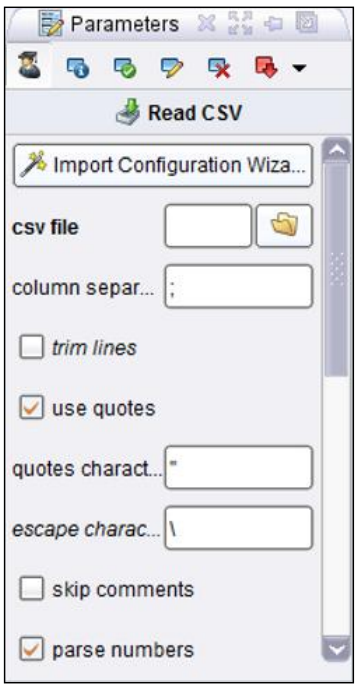

Figure 12-16: Settings for the Read CSV operator.

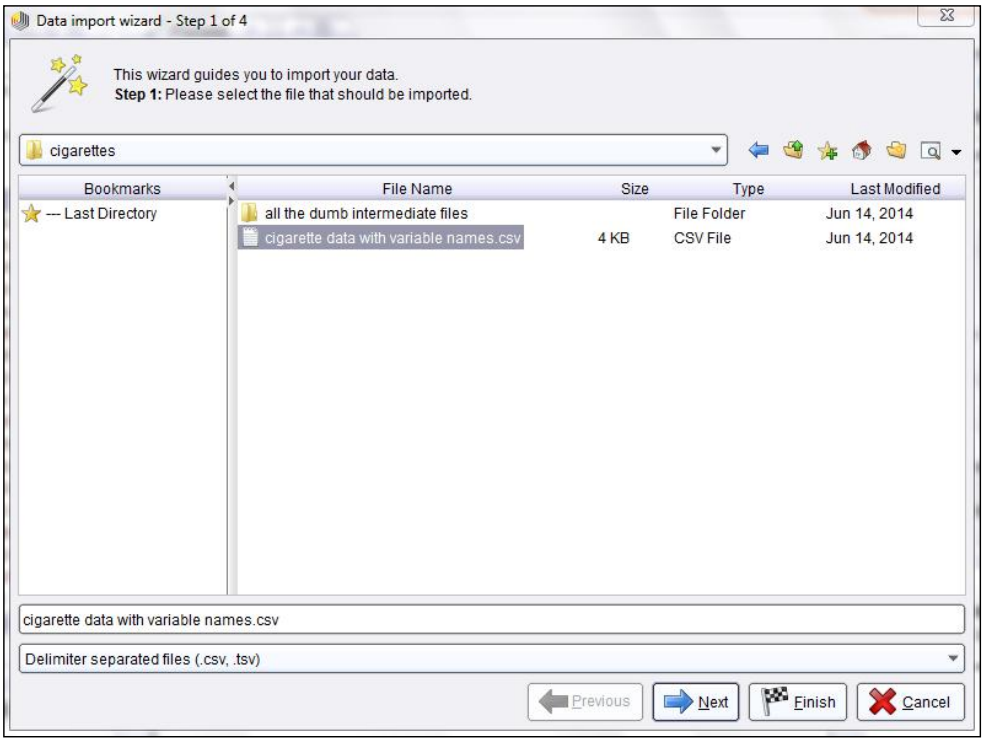

Figure 12-17: Using the wizard to browse for the cigarette data.

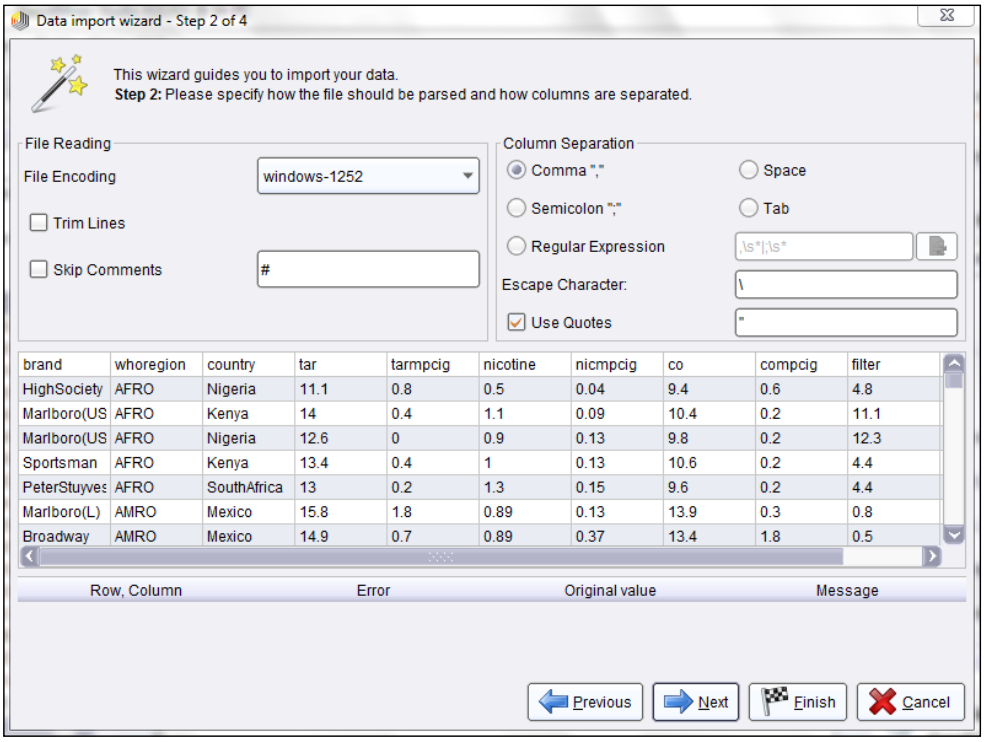

Figure 12-18: Adjusting data import settings with a wizard in RapidMiner Studio.

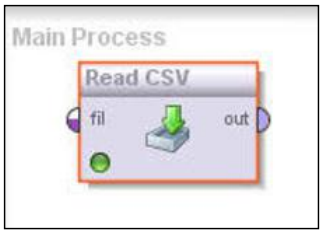

Figure 12-19: Read CSV operator after data import.

To import the sample data in Weka, follow these steps (see Appendix C for information about Weka and where to get it):

1. **Start Weka KnowledgeFlow. (See Figure 12-20.)**
2. **Find the CSVLoader in the Design toolbar. It's grouped with other tools for importing data. (See Figure 12-21.)**
3. **Click the CSVLoader, and then click in the work area to place the CSVLoader in the work area. (See Figure 12-22.)**

4. **Right-click and select Configure. Browse to find the cigarette data. (See Figure 12-23.)**
5. **Adjust settings. (See Figure 12-24.)**
6. **Click the Run Process button (shown in Figure 12-20) to import the data. The Status area updates (see Figure 12-25) when the data has been imported.**

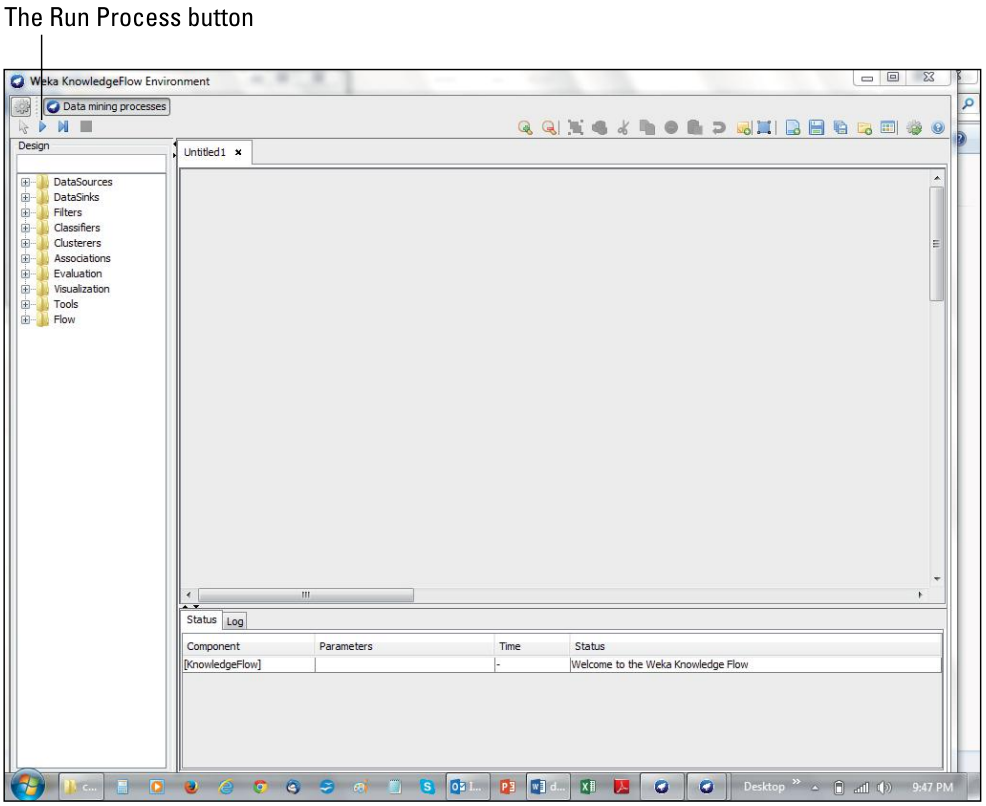

Figure 12-20: Weka KnowledgeFlow.

The look of the applications, the organization of the tools, and the details of setup vary, but the main steps are all quite similar. As long as your application can read the format that you have, the results will be the same.

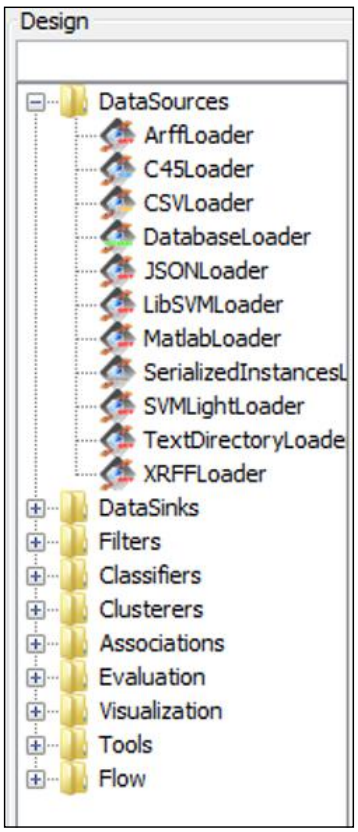

Figure 12-21: CSVLoader in the Design toolbar.

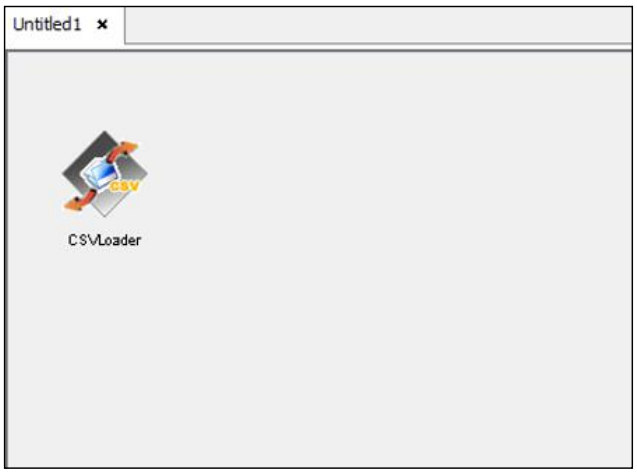

Figure 12-22: CSVLoader in the work area.

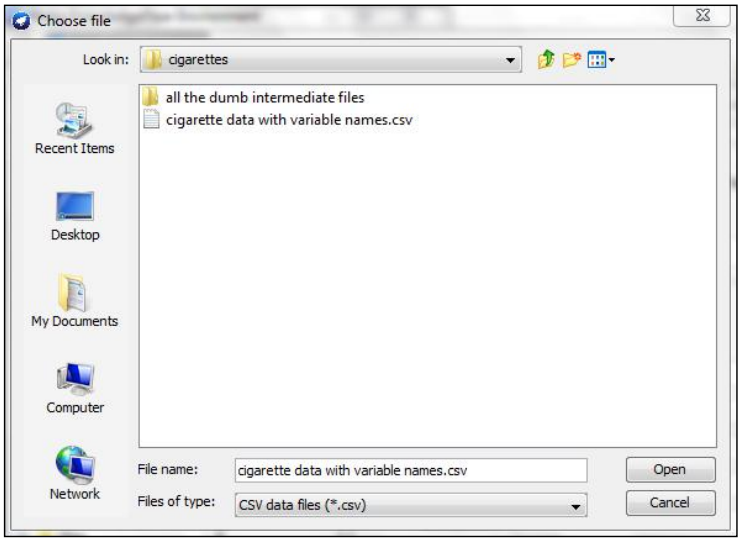

Figure 12-23: Browsing to find the cigarette data in Weka KnowledgeFlow.

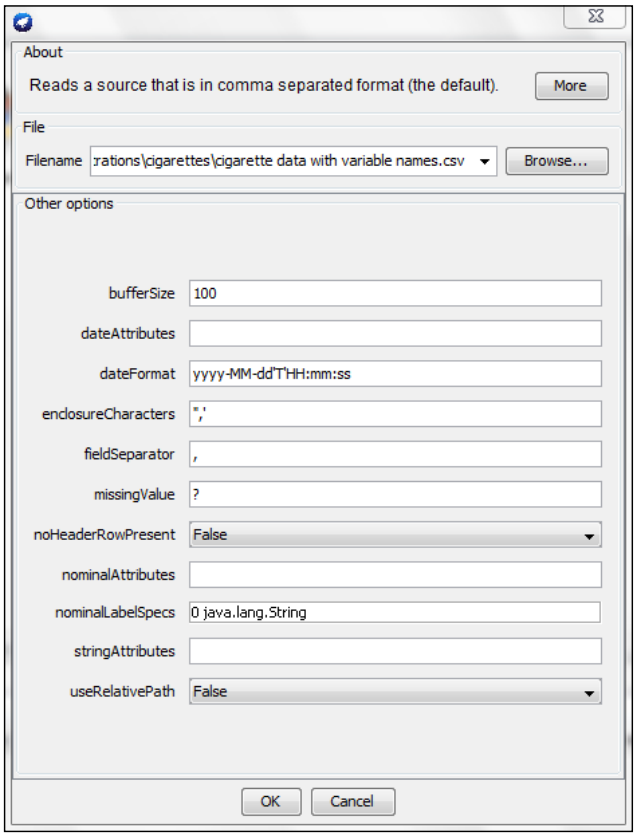

Figure 12-24: Adjusting data import settings in Weka KnowledgeFlow.

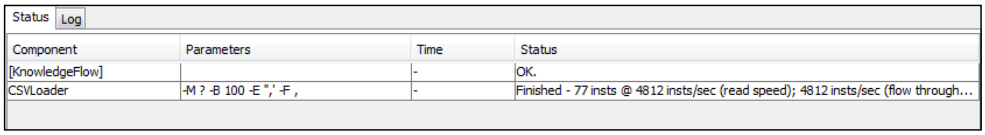

Figure 12-25: Status update after data import.

Databases

Data collected by large organizations in the course of everyday business is usually stored in databases. But database administrators may not be willing to allow data miners direct access to these data sources, and direct access may not be the best option from your point of view either. Direct access to operational (used for routine business operations) databases can be a bad idea because

- **Data miners use a lot of data.** You could unintentionally tie up resources and interfere with ordinary business operations.
- **Legal and other business obligations matter.** You could unintentionally violate a data privacy law or other data management requirement if your data access is not properly controlled.
- **Operational databases are not organized for data mining.** You could spend a lot of time struggling to get the data you need, and still not be sure of getting it right.

When you need data from an operational database (and you have the appropriate approval to use the data), you should discuss your needs with the administrator responsible for that data. You'll need to explain exactly what data you need, the format you need for data mining, and whether you need the data just once or on an ongoing basis. The best approach for one-time requests is often for the administrator to extract the data for you and deliver it in a text file or other acceptable format.

Ongoing data access is another matter. The administrator may not want to provide data extracts over and over, and giving you direct access to business systems is risky. A common solution is to create an *analytic database.* This is an ordinary relational database that is separate from conventional business systems. Data is routinely (and automatically) transferred from business systems to the analytic database, and data miners can access it at any time.

If you use an analytic database, make sure that it is organized properly to support data mining. Help your database administrator by sketching a diagram like Figure 12-1 to show how the data must be organized. If the database administrator insists that the data can't be stored this way, ask whether it's possible to create a *view* (a stored query that can be queried as if it were a conventional data table) with the organization that you need.

Many data-mining products are able to read data from databases. The steps required vary based on the

- Design of the data-mining application
- Structure of the source database
- Middleware, usually called a *driver* (*ODBC driver, JDBC driver*), special software that mediates between the database and applications software

Documentation for your data-mining application should tell you whether it can read data from a database, and if so, what tool or function to use, and how. The administrator who sets up the analytics database can provide details about accessing the database.

If you're already comfortable working with databases and other applications, you'll find nothing surprising about doing the same things with a data-mining application. If databases are new to you, get a knowledgeable person from your organization to walk you through the process with your own database and data-mining application.

Spreadsheets, XML, and specialty data formats

You may need to use data that's in a spreadsheet, XML (extensible markup language), or any of dozens of less common formats. The key question will always be: Does your data-mining application import data in that format?

As long as your data-mining application has a tool to read the data format you need, the process will be straightforward — just a small variation on the examples you can read in the "Text files" section, earlier in this chapter. You may need to select a different data import tool or change a few settings, but the process will be very similar.

When your data-mining application can't import data in a particular format, try these alternative approaches:

- **Check your data source for other formats.** Many sources offer choices.
- **Convert the data format yourself.** Some conversions are easy, and some are difficult.
- **Use a different data-mining application.** Data import capability is an important factor in your choice of data-mining software, but if you're already committed to a particular product, it may not be practical to change.

Becoming fluent in data mining

The data-mining profession has its own vocabulary.

Traditional data analysts call something that you'd like to predict a *dependent variable,* but a data miner may call that a *target* or an *output.* The traditional data analyst's name for something that might influence the dependent variable is *independent variable,* but a data miner may prefer *predictor, input,* or *attribute.*

The kinds of variables that you are using affect your options for data manipulation and modeling. These terms are used by both traditional data analysts and data miners:

Categorical variable types include

- **Nominal:** Names or categories with no order (such as Male and Female).
- **Ordinal:** Ranked or ordered categories such as letter grades or stars in a product review. (Ordinal measures are not meant to be used in mathematical operations, even if they are represented by numbers. However, people violate this rule all the time. Sometimes the results are useful. Often, they're not.)

Continuous variable types include

- **Interval:** Measures such as time and Fahrenheit temperature, which are appropriate to use in some mathematical operations, but not all, because the measurements scales have no clear zero value. (0 degrees Fahrenheit is not the absence of all warmth, for example, but it is pretty unpleasant.)
- **Ratio:** Measures such as weights, lengths, and Kelvin temperature, which can be used in mathematical operations and which have a clear zero value.

The range of terms used within data-mining software is large and varied, perhaps too varied. For example, many data-mining applications use visual programming. That means that functions are represented by little icons that can be moved to a blank space on the screen and connected together to define a data-mining process. In this book, I call those icons *tools,* and some products use that term, too. But others call the same thing a *node, operator,* or some other name. In this book, the blank space is called a work area or *workspace,* but in your data-mining application, it might be called something else, such as a *canvas.* (In the example processes in the "Text files" section, earlier in this chapter, you'll notice the same things called by different names in different products.)

Refer to the glossary in Appendix A for a more extensive list of data-mining terms, and be sure to read the documentation for your data-mining application for product-specific terms.

Surveying Your Data

After you have imported a dataset into your data-mining application, the next step is to review the variables one by one. In your review, you'll examine variables to make sure that you understand what each represents, to find out whether the data is complete, and to assess the quality of the data that you have. The review helps you determine whether your data is adequate to support your data-mining goals.

The data review is part of the data-understanding phase of the CRISP-DM process for data mining. You can find more information about the process in Chapter 4, and you can read about an example data review in Chapter 2.

You will need summaries for each variable of things such as

- Number of missing cases
- Minimum and maximum values
- Averages and standard deviations (measures of variability)
- Values of categorical variables

Some platforms provide data summaries for a slew of variables in one step. Others will require many steps to get this information. One example of a data summary appears in Chapter 2. Here's another, which picks up from the data import in KNIME shown earlier in this chapter. Figure 12-26 shows the process just after importing data.

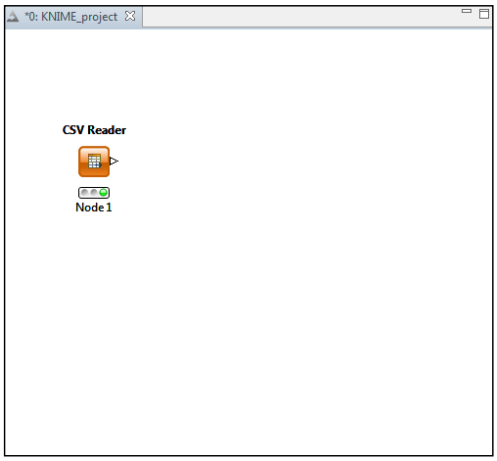

Figure 12-26: A data-mining process in KNIME after importing data.

To create a data summary using KNIME, first complete the KNIME-related steps in the "Text files" section, earlier in this chapter, and then do the following:

1. **Find the Statistics node in the Node Repository (see Figure 12-27) and drag it to the work area. (See Figure 12-28.)**
2. **Click the small arrow on the right side of the CSVReader, and then click the arrow on the left side of the Statistics node.**

3. **Right-click the Statistics node and choose Execute and Open Views from the menu. (See Figure 12-29.)**

   KNIME will display a summary report (see Figure 12-30). It provides key summary statistics for each continuous variable quickly and easily.

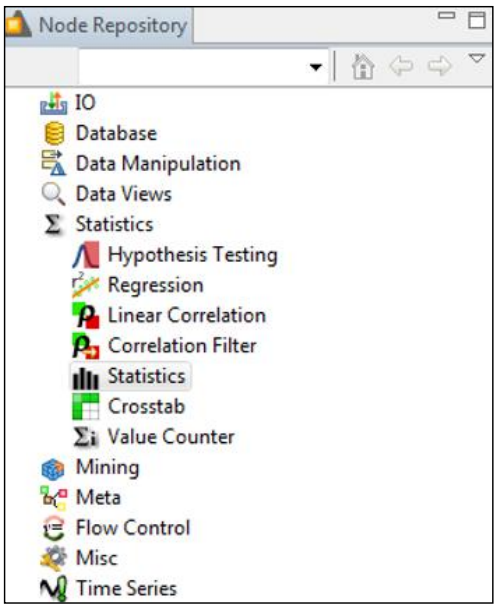

Figure 12-27: Finding the Statistics node in the Node Repository.

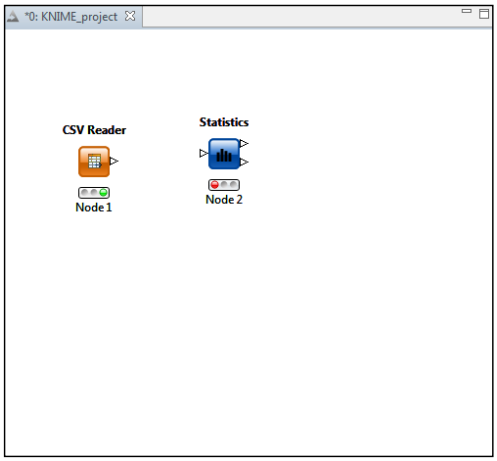

Figure 12-28: Adding the Statistics node to the process.

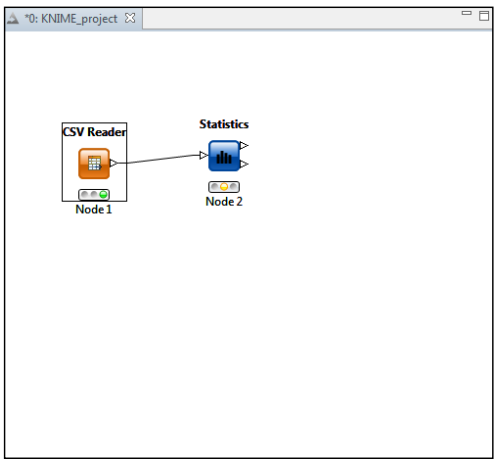

Figure 12-29: Connecting two nodes.

Numeric columns | Nominal columns

Row ID	tar	tarmpcig	nicotine	nicmpcig	co	compcig	filter	filter2
Minimum	6.8	0	0	0	5.9	0	0.1	0.1
Maximum	9,915	4.1	1.63	0.67	17.4	2.7	99	99
Mean	270.73	0.969	0.901	0.106	11.505	0.674	14.062	4.042
Std. deviation	1,585.012	0.853	0.23	0.105	2.396	0.605	16.731	15.659
Variance	2,512,261.803	0.728	0.053	0.011	5.741	0.366	279.923	245.207
Overall sum	20,846.2	74.6	69.34	8.18	885.9	51.9	1,082.8	311.2
No. missings	0	0	0	0	0	0	0	0
Median	?	?	?	?	?	?	?	?
Row count	77	77	77	77	77	77	77	77
No. NaNs	0	0	0	0	0	0	0	0
No. +infinities	0	0	0	0	0	0	0	0
No. -infinities	0	0	0	0	0	0	0	0

Figure 12-30: Data summary displayed in KNIME.