4.5.1 Find Missing Values

Maria is aware that at least one of the datasets needs to be cleaned before any analysis can be performed. She would like you to use Pandas to do a more thorough inspection of the datasets than you did when you opened them with Excel. Cleaning the data is essential because any missing, malformed, or incorrect data in the rows can jeopardize the analysis.

As a first step in the data-cleaning process, we'll determine if there are missing values in the rows of the CSV files. Let's look at each CSV file separately.

First, open `schools_complete.csv`. We can see that each row contains a School ID, school name, type of school, student size, and budget. Therefore, there are no missing values in any of the rows, which are also called rows with **null values**. See the following image:

	A	B	C	D	E
1	School ID	school_name	type	size	budget
2	0	Huang High School	District	2917	1910635
3	1	Figueroa High School	District	2949	1884411
4	2	Shelton High School	Charter	1761	1056600

5	3	Hernandez High School	District	4635	3022020
6	4	Griffin High School	Charter	1468	917500
7	5	Wilson High School	Charter	2283	1319574
8	6	Cabrera High School	Charter	1858	1081356
9	7	Bailey High School	District	4976	3124928
10	8	Holden High School	Charter	427	248087
11	9	Pena High School	Charter	962	585858
12	10	Wright High School	Charter	1800	1049400
13	11	Rodriguez High School	District	3999	2547363
14	12	Johnson High School	District	4761	3094650
15	13	Ford High School	District	2739	1763916
16	14	Thomas High School	Charter	1635	1043130

While `schools_complete.csv` has only 15 rows of data and one row for the headers, the `student_complete.csv` has 39,170 rows. It would be very time-consuming to find missing values in a file so large. Luckily, Pandas has a few methods that can help us determine whether there are missing values in large datasets: the `count()` method, `isnull()` method, and `notnull()` method.

The count() Method

With the `count()` method, we can get a count of the rows for each column containing data. By default, "null" values are not counted, so you can often quickly identify which columns have missing data.

Let's use the `count()` method on the `school_data_df` DataFrame. Add the following code to a new cell and run the cell:

```
# Determine if there are any missing values in the school data.
school_data_df.count()
```

The output returns the name of the columns and the number of rows that are not null. For the `school_data_df` DataFrame, there are no missing values, because there are 15 rows that contain data in `schools_complete.csv`. In the output, the number 15 is next to each column

header, as shown in the following image:

```
School ID       15
school_name     15
type            15
size            15
budget          15
dtype: int64
```

These results confirm what we observed when we looked at the `schools_complete.csv` file.

Now let's use the same method on the `student_data_df` DataFrame. Add the following code to a new cell and run the cell:

```
# Determine if there are any missing values in the student data.
student_data_df.count()
```

Like in the `schools_complete.csv` file, there are no missing values in any of the columns because the output shows 39,170 rows for the `student_complete.csv` file:

```
Student ID       39170
student_name     39170
gender           39170
```

```
grade             39170
school_name       39170
reading_score     39170
math_score        39170
dtype: int64
```

NOTE

> For more information, see the **Pandas documentation on the count method (https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.count.html)**.

The isnull() Method

The Pandas library also has the `isnull()` method for determining empty rows. When you apply the `isnull()` method to a column, Series, or a DataFrame, a Boolean value will be returned, either "True" for the row or rows that are empty, i.e., null, or "False" for the rows that are not empty.

Let's use the `isnull()` method on the `school_data_df` DataFrame.

```
# Determine if there are any missing values in the school data.
school_data_df.isnull()
```

When we execute this code, we see that every row that is not empty is given the Boolean value "False," which tells us that there are no missing values. See the following screenshot:

```
# Determine if there are any missing values in the school data.
school_data_df.isnull()
```

	School ID	school_name	type	size	budget
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
5	False	False	False	False	False
6	False	False	False	False	False
7	False	False	False	False	False
8	False	False	False	False	False
9	False	False	False	False	False
10	False	False	False	False	False
11	False	False	False	False	False
12	False	False	False	False	False
13	False	False	False	False	False
14	False	False	False	False	False

Now we'll use the same method on the `student_data_df` DataFrame. Add the following code to a new cell and run the cell:

```
# Determine if there are any missing values in the student data.
student_data_df.isnull()
```

The output shows that there are no rows that contain missing values, as they are all labeled "False." See the following image:

```
student_data_df.isnull()
```

	Student ID	student_name	gender	grade	school_name	reading_score	math_score
0	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False
5	False	False	False	False	False	False	False

To get the total number of empty rows, or rows that are "True," we can use

the Pandas `sum()` method after the `isnull()` method, like this:

```
# Determine if there are any missing values in the student data.
student_data_df.isnull().sum()
```

REWIND

The process of joining two or more methods or functions together that are separated by a period is called **chaining**.

The output after running this code shows the total number of rows that are empty is zero for each column:

```
student_data_df.isnull().sum()

Student ID        0
student_name      0
gender            0
grade             0
school_name       0
reading_score     0
math_score        0
dtype: int64
```

This output allows us to more easily determine at a glance how many rows are empty in the `student_data_df` DataFrame (zero). This is more

straightforward than output that shows "False" in thousands of rows.

NOTE

> For more information, see the **Pandas documentation on the isnull() method (https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.isnull.html)**.

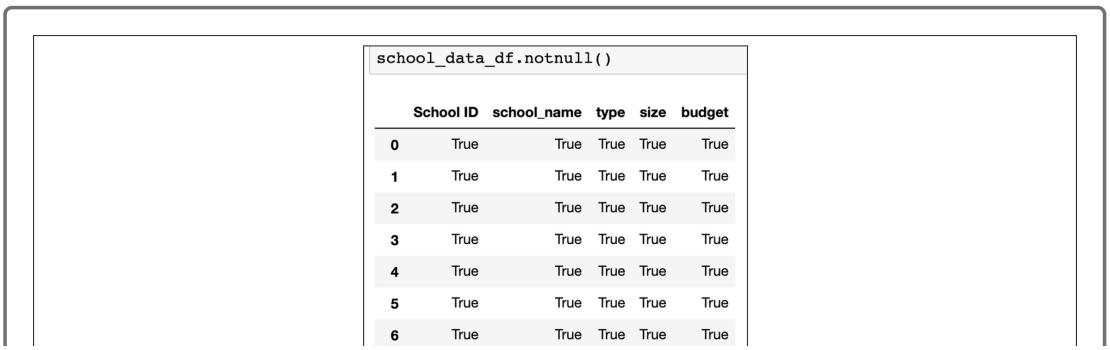
The notnull() Method

Another method that we can use to find missing values is the `notnull()` method. When you apply the `notnull()` method to a column, Series, or a DataFrame, a Boolean will be returned: "True" for the row or rows that are not empty, or "False" for the row or rows that are empty. This method returns the opposite output of the `isnull()` method.

Let's use the `notnull()` method on the `school_data_df` DataFrame. Run the following code:

```
# Determine if there are not any missing values in the school data.
school_data_df.notnull()
```

When we run this code, the output returns a copy of our `school_data_df` DataFrame, where all the rows that do not have any missing values are labeled "True." See the following image.



```
school_data_df.notnull()
```

	School ID	school_name	type	size	budget
0	True	True	True	True	True
1	True	True	True	True	True
2	True	True	True	True	True
3	True	True	True	True	True
4	True	True	True	True	True
5	True	True	True	True	True
6	True	True	True	True	True

7	True	True	True	True	True
8	True	True	True	True	True
9	True	True	True	True	True
10	True	True	True	True	True
11	True	True	True	True	True
12	True	True	True	True	True
13	True	True	True	True	True
14	True	True	True	True	True

Like we did with the `student_data_df` DataFrame, we can chain the `notnull()` method and the `sum()` method to get the sum of all the columns that are "True."

```
# Determine if there are not any missing values in the student data.
student_data_df.notnull().sum()
```

When we execute this code, we get the number of rows that are not null, which is 39,170 for each column.

```
student_data_df.notnull().sum()

Student ID        39170
student_name      39170
gender            39170
grade             39170
school_name       39170
reading_score     39170
math_score        39170
dtype: int64
```

NOTE

> For more information, see the **Pandas documentation on the notnull()**

> **method (http://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.notnull.html).**