

4.5.2 Handle Missing Data

Maria is thrilled that there is no missing data in the CSV files. However, as a precautionary measure, she would like you to understand the options available for handling missing data in case the need arises. She's given you a CSV file that has some missing data so that you can practice.

There are a few options for handling missing data:

- Do nothing.
- Drop the row that has the missing value.
- Fill in the row that has the missing value.

It's not always obvious which of these options you should use, so let's review each one.

Click the following link to download the `missing_grades.csv` file into your Resources folder. (This is the practice dataset from Maria.)

[Download missing_grades.csv](https://2u-data-curriculum-team.s3.amazonaws.com/dataviz-online/module_4/missing_grades.csv) (https://2u-data-curriculum-team.s3.amazonaws.com/dataviz-online/module_4/missing_grades.csv)

Open the `missing_grades.csv` file. You will notice there are two cells, E4 and E7, that are empty. The empty cells are shown in the following image:

A	B	C	D	E	F
Student ID	student_name	gender	grade	reading_score	math_score
0	Paul Bradley	M	9th	66	79
1	Victor Smith	M	12th	94	61
2	Kevin Rodriguez	M	12th		60
3	Dr. Richard Scott	M	12th	67	58
4	Bonnie Ray	F	9th	97	84
5	Bryan Miranda	M	9th	94	
6	Sheena Carter	F	11th	82	80
7	Nicole Baker	F	12th	96	69

To handle this, first create a new Jupyter Notebook file in the School_District_Analysis folder and rename it `cleaning_data.ipynb`.

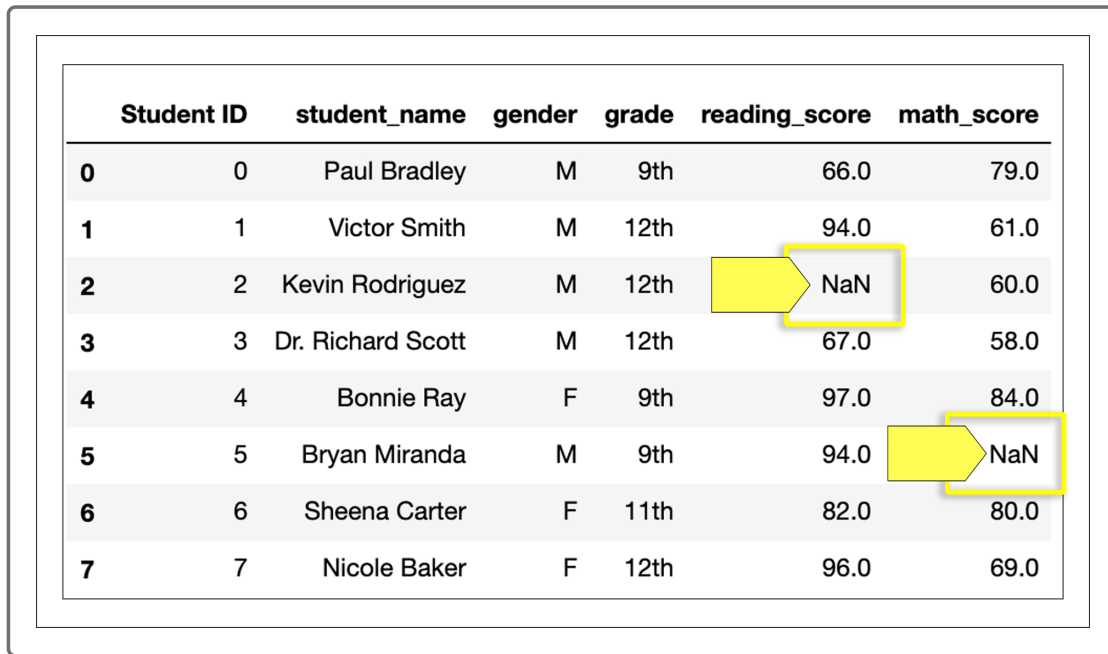
In the first cell, import the Pandas library as the dependency and run the cell.

```
# Add the Pandas dependency.  
import pandas as pd
```

In the next cell, declare a variable and assign it to the `missing_grades.csv` file that is located in the Resources folder.

```
# Files to load  
file_to_load = "Resources/missing_grades.csv"  
  
# Read the CSV into a DataFrame  
missing_grade_df = pd.read_csv(file_to_load)  
missing_grade_df
```

When we run this cell, the empty rows will be represented by "NaN," as shown in the following image:



	Student ID	student_name	gender	grade	reading_score	math_score
0	0	Paul Bradley	M	9th	66.0	79.0
1	1	Victor Smith	M	12th	94.0	61.0
2	2	Kevin Rodriguez	M	12th	NaN	60.0
3	3	Dr. Richard Scott	M	12th	67.0	58.0
4	4	Bonnie Ray	F	9th	97.0	84.0
5	5	Bryan Miranda	M	9th	94.0	NaN
6	6	Sheena Carter	F	11th	82.0	80.0
7	7	Nicole Baker	F	12th	96.0	69.0

IMPORTANT

NaN means "not a number" and cannot be equal to zero.

Now let's review our options for handling the missing data.

Option 1: Do Nothing

If we do nothing, when we sum or take the averages of the reading and math scores, those NaNs will not be considered in the sum or the averages (just as they are not considered in the sum or the averages in an Excel file). In this situation, the missing values have no impact.

However, if we multiply or divide with a row that has a NaN, the answer will be NaN. This can cause problems if we need the answer for the rest of our code.

Option 2: Drop the Row

Another option is to drop the row where there are NaNs. When we remove the row containing the NaN, we will also remove all the data associated with that row. This can cause problems later if there is data in the other rows that we need.

To drop a row with NaNs, Pandas has the `dropna()` method. Use this method on the `missing_grade_df` DataFrame like this:

```
# Drop the NaNs.  
missing_grade_df.dropna()
```

When we execute this code, the DataFrame will look like the following image. The rows with missing values have been dropped, as indicated by the index numbers, 0, 1, 3, 4, 6, and 7. After these rows are dropped, the index of the DataFrame is not automatically reset to number the rows in consecutive order.

	Student ID	student_name	gender	grade	reading_score	math_score
0	0	Paul Bradley	M	9th	66.0	79.0
1	1	Victor Smith	M	12th	94.0	61.0
3	3	Dr. Richard Scott	M	12th	67.0	58.0
4	4	Bonnie Ray	F	9th	97.0	84.0
6	6	Sheena Carter	F	11th	82.0	80.0
7	7	Nicole Baker	F	12th	96.0	69.0

IMPORTANT

Dropping rows can affect the story you are trying to tell with the data. Before removing rows with NaN, you should ask yourself two key questions:

1. How much data would be removed if NaNs are dropped?
2. How would this impact the analysis?

These questions need to be addressed for every dataset you work with.

NOTE

For more information, see the [Pandas documentation on the dropna\(\) method](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.dropna.html) [\(https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.dropna.html\)](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.dropna.html).

Option 3: Fill in the Row

The third option is to fill in all the NaNs with a value.

IMPORTANT

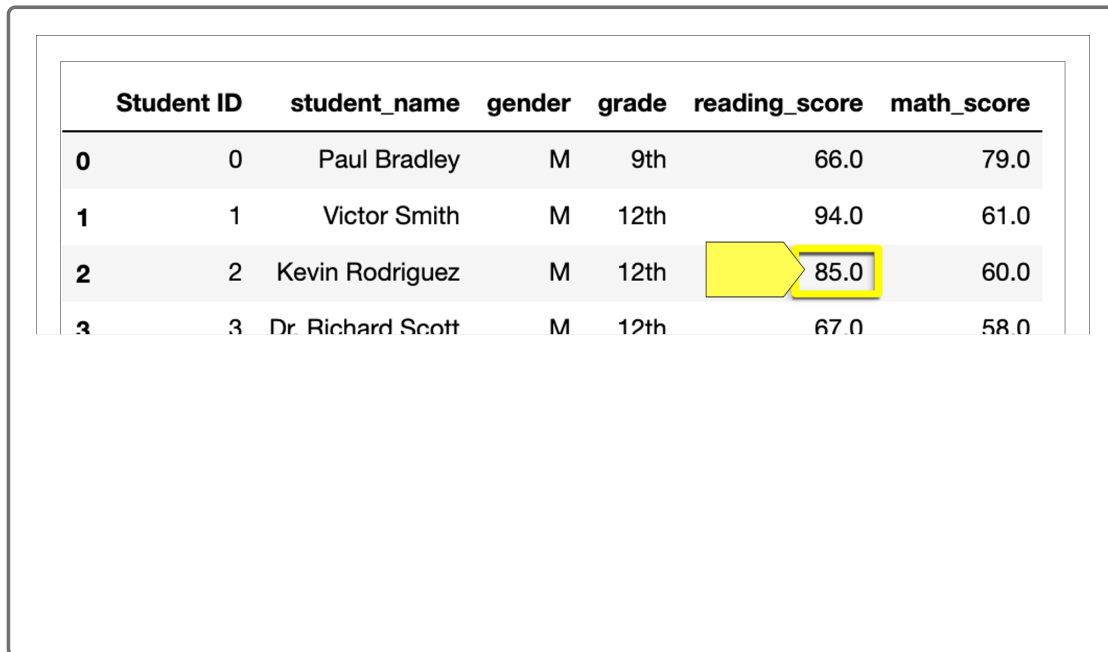
Filling in an empty row must be used with caution. For example, filling in a row with "0" can impact arithmetic calculations. If you decide to fill in empty rows, the values you insert must be carefully considered for every downstream analysis you perform.

In Pandas, we use the `fillna()` method to fill in a row. If we want to fill all empty rows with zero in our DataFrame, we pass the "0" in parentheses like this: `df.fillna(0)`.

Let's convert the `missing_grades.csv` file to a DataFrame again and fill in the NaNs with the number 85, using the following code:

```
# Fill in the empty rows with "85".  
missing_grade_df.fillna(85)
```

When we execute this cell, the cells with NaN will be filled with 85, as shown in the following image:



	Student ID	student_name	gender	grade	reading_score	math_score
0	0	Paul Bradley	M	9th	66.0	79.0
1	1	Victor Smith	M	12th	94.0	61.0
2	2	Kevin Rodriguez	M	12th	85.0	60.0
3	3	Dr. Richard Scott	M	12th	67.0	58.0

NOTE

For more information, see the [Pandas documentation on the fillna\(\) method](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.fillna.html) [_](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.fillna.html)(<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.fillna.html>) and how to [work with missing data](https://pandas.pydata.org/pandas-docs/stable/user_guide/missing_data.html) [_](https://pandas.pydata.org/pandas-docs/stable/user_guide/missing_data.html)(https://pandas.pydata.org/pandas-docs/stable/user_guide/missing_data.html) [_](https://pandas.pydata.org/pandas-docs/stable/user_guide/missing_data.html).