18.2.2

Pandas Refresher

When it comes to preprocessing data, you have good news for Martha. The Pandas Python library is really good at this! When Martha asks for a quick refresher on how to use Pandas for data munging, you know just the dataset to use—the iris dataset from the University of California, Irvine (UCI) Machine Learning Repository.

Pandas is a Python library that is excellent for data munging. We'll be using the **iris dataset from the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/iris)**, a common dataset used throughout machine learning:

1. Store the raw **iris.csv (https://2u-data-curriculum-team.s3.amazonaws.com/dataviz-online/module_18/iris.csv)**
2. Open a new Jupyter Notebook.
3. Import your libraries:

```
import pandas as pd
```

4. To load the dataset in a Pandas DataFrame, enter the code below. Be sure to use the path to the stored CSV file (stored in an easy-to-access location):

```
file_path = "<folder path to stored data sets>/iris.csv"
iris_df = pd.read_csv(file_path)
iris_df.head()
```

5. Select the fields of data you want:

```
file_path = "Resources/iris.csv"
iris_df = pd.read_csv(file_path)
iris_df.head()
```

	sepal_length	sepal_width	petal_length	petal_width	class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

Which field should we drop from the DataFrame?

- ○ sepal_length
- ○ sepal_width
- ○ petal_length
- ○ petal_width
- ○ class

Check Answer

Finish ▶

NOTE

Unsupervised learning will be used to determine the class of the iris plants later on in the module.

6. Drop the class field using the code below:

```
new_iris_df = iris_df.drop(['class'], axis=1)
new_iris_df.head()
```

	sepal_length	sepal_width	petal_length	petal_width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

SKILL DRILL

Try reordering the columns so the sepal and petal lengths are the first two columns and the widths are the last two columns.

Cleaning this dataset appears complete with all the data in numerical form and the same type, so no data processing is needed. However, you'll encounter data transformations on datasets that contain categorical data or non-numeric features (e.g., transforming male and female categorical values to 0 and 1, respectively).

Finally, the preprocessed DataFrame is saved on a new CSV file for future use. This is done by storing the file path in a variable, then using the Pandas `to_csv()` method to export the DataFrame to a CSV by supplying the file path and file name as arguments, as shown below:

```
output_file_path = "<path to folder>/new_iris_data.csv"
new_iris_df.to_csv(output_file_path, index=False)
```