Write a CSV File

You can save your Pandas DataFrame as a CSV file with .to_csv():

```
df.to_csv('data.csv')
```

That’s it! You’ve created the file data.csv in your current working directory. You can expand the code block below to see how your CSV file should look:

This text file contains the data separated with **commas**. The first column contains the row labels. In some cases, you’ll find them irrelevant. If you don’t want to keep them, then you can pass the argument index=False to .to_csv().

Write an Excel File

Once you have those packages installed, you can save your DataFrame in an Excel file with .to_excel():

```
df.to_excel('data.xlsx')
```

The argument 'data.xlsx' represents the target file and, optionally, its path. The above statement should create the file data.xlsx in your current working directory.

Write Files

Series and DataFrame objects have methods that enable writing data and labels to the clipboard or files. They’re named with the pattern **.to_<file-type>()**, where <file-type> is the type of the target file.

You’ve learned about .to_csv() and .to_excel(), but there are others, including:

- .to_json()
- .to_html()
- .to_sql()
- .to_pickle()

There are still more file types that you can write to, so this list is not exhaustive.