



Figure 1-1. Simple line plot of the sine function using *matplotlib*

## pandas

*pandas* is a Python library for data wrangling and analysis. It is built around a data structure called the *DataFrame* that is modeled after the R *DataFrame*. Simply put, a *pandas DataFrame* is a table, similar to an Excel spreadsheet. *pandas* provides a great range of methods to modify and operate on this table; in particular, it allows SQL-like queries and joins of tables. In contrast to NumPy, which requires that all entries in an array be of the same type, *pandas* allows each column to have a separate type (for example, integers, dates, floating-point numbers, and strings). Another valuable tool provided by *pandas* is its ability to ingest from a great variety of file formats and databases, like SQL, Excel files, and comma-separated values (CSV) files. Going into detail about the functionality of *pandas* is out of the scope of this book. However, *Python for Data Analysis* by Wes McKinney (O'Reilly, 2012) provides a great guide. Here is a small example of creating a *DataFrame* using a dictionary:

In[7]:

```
import pandas as pd

# create a simple dataset of people
data = {'Name': ["John", "Anna", "Peter", "Linda"],
        'Location': ["New York", "Paris", "Berlin", "London"],
        'Age': [24, 13, 53, 33]}

data_pandas = pd.DataFrame(data)
# IPython.display allows "pretty printing" of dataframes
# in the Jupyter notebook
display(data_pandas)
```

This produces the following output:

	Age	Location	Name
0	24	New York	John
1	13	Paris	Anna
2	53	Berlin	Peter
3	33	London	Linda

There are several possible ways to query this table. For example:

**In[8]:**

```
# Select all rows that have an age column greater than 30
display(data_pandas[data_pandas.Age > 30])
```

This produces the following result:

	Age	Location	Name
2	53	Berlin	Peter
3	33	London	Linda

## mglearn

This book comes with accompanying code, which you can find on [GitHub](#). The accompanying code includes not only all the examples shown in this book, but also the `mglearn` library. This is a library of utility functions we wrote for this book, so that we don't clutter up our code listings with details of plotting and data loading. If you're interested, you can look up all the functions in the repository, but the details of the `mglearn` module are not really important to the material in this book. If you see a call to `mglearn` in the code, it is usually a way to make a pretty picture quickly, or to get our hands on some interesting data.



Throughout the book we make ample use of NumPy, `matplotlib` and `pandas`. All the code will assume the following imports:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import mglearn
```

We also assume that you will run the code in a Jupyter Notebook with the `%matplotlib notebook` or `%matplotlib inline` magic enabled to show plots. If you are not using the notebook or these magic commands, you will have to call `plt.show` to actually show any of the figures.