1 Importing and Exporting Data

1.1 Importing Data using pandas

All of the data reading functions in pandas load data into a pandas DataFrame, and so these examples all make use of the values property to extract a NumPy array.

In practice, the DataFrame is much more useful since it includes useful information such as column names read from the data source. In addition to the three main formats, pandas can also read json, SQL, html tables or from the clipboard, which is particularly useful for interactive work since virtually any source that can be copied to the clipboard can be imported.

1.2 CSV and other formatted text files

Comma-separated value (CSV) files can be read using `read_csv`. When the CSV file contains *mixed data*, the default behavior will read the file into an array with an object data type, and so further processing is usually required to extract the individual series.

```
>>> from pandas import read_csv
>>> csv_data = read_csv('FTSE_1984_2012.csv')
>>> csv_data = csv_data.values
>>> csv_data[:4]
array([['2012-02-15', 5899.9, 5923.8, 5880.6,
 5892.2, 801550000L, 5892.2],
['2012-02-14', 5905.7, 5920.6, 5877.2, 5899.9, 832567200L, 5899.9],
['2012-02-13', 5852.4, 5920.1, 5852.4, 5905.7, 643543000L, 5905.7],
['2012-02-10', 5895.5, 5895.5, 5839.9, 5852.4, 948790200L, 5852.4]],
dtype=object)
>>> open = csv_data[:,1]
```

When the entire file is numeric, the data will be stored as a homogeneous array using one of the numeric data types, typically float64. In this example, the first column contains Excel dates as numbers, which are the number of days past January 1, 1900.

```
>>> csv_data = read_csv('FTSE_1984_2012_numeric.csv')
>>> csv_data = csv_data.values
>>> csv_data[:4,:2]
array([[ 40954. , 5899.9],
[ 40953. , 5905.7],
[ 40952. , 5852.4],
[ 40949. , 5895.5]])
```

1.2.1 Excel files

Excel files, both 97/2003 (xls) and 2007/10/13 (xlsx), can be imported using `read_excel`. Two inputs are required to use `read_excel`, the filename and the sheet name containing the data. In this example, pandas makes use of the information in the Excel workbook that the first column contains dates and converts these to datetimes. Like the mixed CSV data, the array returned has object data type.

```
>>> from pandas import read_excel
>>> excel_data = read_excel('FTSE_1984_2012.xls','FTSE_1984_2012')
>>> excel_data = excel_data.values
>>> excel_data[:4,:2]
array([[datetime.datetime(2012, 2, 15, 0, 0), 5899.9],
[datetime.datetime(2012, 2, 14, 0, 0), 5905.7],
[datetime.datetime(2012, 2, 13, 0, 0), 5852.4],
[datetime.datetime(2012, 2, 10, 0, 0), 5895.5]], dtype=object)
>>> open = excel_data[:,1]
```