

# Using Pandas to Get Familiar With Your Data

The first step in any machine learning project is familiarize yourself with the data. You'll use the Pandas library for this. Pandas is the primary tool data scientists use for exploring and manipulating data. Most people abbreviate pandas in their code as `pd`. We do this with the command

```
In [1]: import pandas as pd
```

The most important part of the Pandas library is the DataFrame. A DataFrame holds the type of data you might think of as a table. This is similar to a sheet in Excel, or a table in a SQL database.

Pandas has powerful methods for most things you'll want to do with this type of data.

As an example, we'll look at [data about home prices](#) in Melbourne, Australia. In the hands-on exercises, you will apply the same processes to a new dataset, which has home prices in Iowa.

The example (Melbourne) data is at the file path `../input/melbourne-housing-snapshot/melb_data.csv`.

We load and explore the data with the following commands:

In [2]:

```
# save filepath to variable for easier access
melbourne_file_path = '../input/melbourne-housing-snapshot/melb_data.csv'
# read the data and store data in DataFrame titled melbourne_data
melbourne_data = pd.read_csv(melbourne_file_path)
# print a summary of the data in Melbourne data
melbourne_data.describe()
```

Out[2]:

	Rooms	Price	Distance	Postcode	Bedroom2	Bathroom	Count
count	13580.000000	1.358000e+04	13580.000000	13580.000000	13580.000000	13580.000000	13580
mean	2.937997	1.075684e+06	10.137776	3105.301915	2.914728	1.534242	2.938
std	0.955748	6.393107e+05	5.868725	90.676964	0.965921	0.691712	0.956
min	1.000000	8.500000e+04	0.000000	3000.000000	0.000000	0.000000	1.000
25%	2.000000	6.500000e+05	6.100000	3044.000000	2.000000	1.000000	2.000
50%	3.000000	9.030000e+05	9.200000	3084.000000	3.000000	1.000000	3.000
75%	3.000000	1.330000e+06	13.000000	3148.000000	3.000000	2.000000	3.000
max	10.000000	9.000000e+06	48.100000	3977.000000	20.000000	8.000000	10.000

## Interpreting Data Description

The results show 8 numbers for each column in your original dataset. The first number, the **count**, shows how many rows have non-missing values.

Missing values arise for many reasons. For example, the size of the 2nd bedroom wouldn't be collected when surveying a 1 bedroom house. We'll come back to the topic of missing data.

The second value is the **mean**, which is the average. Under that, **std** is the standard deviation, which measures how numerically spread out the values are.

To interpret the **min**, **25%**, **50%**, **75%** and **max** values, imagine sorting each column from lowest to highest value. The first (smallest) value is the min. If you go a quarter way through the list, you'll find a number that is bigger than 25% of the values and smaller than 75% of the values. That is the **25%** value (pronounced "25th percentile"). The 50th and 75th percentiles are defined analogously, and the **max** is the largest number.

## Your Turn

Get started with your [first coding exercise](#)