Introduction

You'll learn all about **pandas**, the most popular Python library for data analysis.

In this tutorial, you will learn how to create your own data, along with how to work with data that already exists.

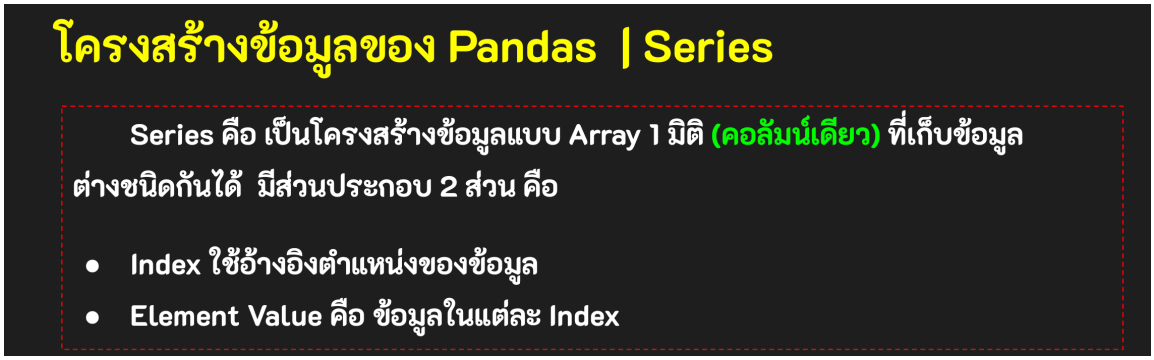
Getting started

To use pandas, you'll typically start with the following line of code.

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

Creating data

There are two core objects in pandas: the **DataFrame** and the **Series**.



DataFrame

A DataFrame is a table. It contains an array of individual *entries*, each of which has a certain *value*. Each entry corresponds to a row (or *record*) and a *column*.

For example, consider the following simple DataFrame:

In [2]:
```
pd.DataFrame({'Yes': [50, 21], 'No': [131, 2]})
```

Out[2]:

	Yes	No
0	50	131
1	21	2

In this example, the "0, No" entry has the value of 131. The "0, Yes" entry has a value of 50, and so on.

DataFrame entries are not limited to integers. For instance, here's a DataFrame whose values are strings:

In [3]:
```
pd.DataFrame({'Bob': ['I liked it.', 'It was awful.'], 'Sue': ['Pretty good.
```

Out[3]:

	Bob	Sue
0	I liked it.	Pretty good.
1	It was awful.	Bland.

We are using the `pd.DataFrame()` constructor to generate these DataFrame objects. The syntax for declaring a new one is a dictionary whose keys are the column names (`Bob` and `Sue` in this example), and whose values are a list of entries. This is the standard way of constructing a new DataFrame, and the one you are most likely to encounter.

The dictionary-list constructor assigns values to the *column labels*, but just uses an ascending count from 0 (0, 1, 2, 3, ...) for the *row labels*. Sometimes this is OK, but oftentimes we will want to assign these labels ourselves.

The list of row labels used in a DataFrame is known as an **Index**. We can assign values to it by using an `index` parameter in our constructor:

In [4]:
```
pd.DataFrame({'Bob': ['I liked it.', 'It was awful.'],
              'Sue': ['Pretty good.', 'Bland.']},
             index=['Product A', 'Product B'])
```

Out[4]:

	Bob	Sue
Product A	I liked it.	Pretty good.
Product B	It was awful.	Bland.

Series

A Series, by contrast, is a sequence of data values. If a DataFrame is a table, a Series is a list. And in fact you can create one with nothing more than a list:

In [5]:
```
pd.Series([1, 2, 3, 4, 5])
```

Out[5]:
```
0    1
1    2
2    3
3    4
4    5
dtype: int64
```

A Series is, in essence, a single column of a DataFrame. So you can assign row labels to the Series the same way as before, using an `index` parameter. However, a Series does not have a column name, it only has one overall `name`:

In [6]:
```
pd.Series([30, 35, 40], index=['2015 Sales', '2016 Sales', '2017 Sales'], na
```

Out[6]:
```
2015 Sales    30
2016 Sales    35
2017 Sales    40
Name: Product A, dtype: int64
```

In [7]:
```
pd.DataFrame(pd.Series([30, 35, 40], index=['2015 Sales', '2016 Sales', '201
```

Out[7]:

	Product A
2015 Sales	30
2016 Sales	35
2017 Sales	40

The Series and the DataFrame are intimately related. It's helpful to think of a DataFrame as actually being just a bunch of Series "glued together". We'll see more of this in the next section of this tutorial.

Reading data files

Being able to create a DataFrame or Series by hand is handy. But, most of the time, we won't actually be creating our own data by hand. Instead, we'll be working with data that already exists.

Data can be stored in any of a number of different forms and formats. By far the most basic of these is the humble CSV file. When you open a CSV file you get something that looks like this:

```
Product A,Product B,Product C,
30,21,9,
35,34,1,
41,11,11
```

So a CSV file is a table of values separated by commas. Hence the name: "Comma-Separated Values", or CSV.

Let's now set aside our toy datasets and see what a real dataset looks like when we read it into a DataFrame. We'll use the `pd.read_csv()` function to read the data into a DataFrame. This goes thusly:

In [8]:
```
wine_reviews = pd.read_csv("datasets/winemag-data-130k-v2.csv")
```

We can use the `shape` attribute to check how large the resulting DataFrame is:

In [9]:
```
wine_reviews.shape
```

Out[9]:
```
(65499, 14)
```

So our new DataFrame has 65,499 records split across 14 different columns. That's almost 1 million entries!

We can examine the contents of the resultant DataFrame using the `head()` command, which grabs the first five rows:

In [10]:
```
wine_reviews.head()   #แสดงผล 5 แถวแรก
```

Out[10]:

	Unnamed: 0	country	description	designation	points	price	province	region_1	
0	0	Italy	Aromas include tropical fruit, broom, brimston...	Vulkà Bianco	87	NaN	Sicily & Sardinia	Etna	
1	1	Portugal	This is ripe and fruity, a wine that is smooth...	Avidagos	87	15.0	Douro	NaN	
2	2	US	Tart and snappy, the flavors of lime flesh and...	NaN	87	14.0	Oregon	Willamette Valley	W
3	3	US	Pineapple rind, lemon pith and orange blossom ...	Reserve Late Harvest	87	13.0	Michigan	Lake Michigan Shore	
4	4	US	Much like the regular bottling from 2012, this...	Vintner's Reserve Wild Child Block	87	65.0	Oregon	Willamette Valley	W

In [11]:
```
wine_reviews.head(10)   #แสดงผล 10 แถวแรก
```

Out[11]:

	Unnamed: 0	country	description	designation	points	price	province	region_
0	0	Italy	Aromas include tropical fruit, broom, brimston...	Vulkà Bianco	87	NaN	Sicily & Sardinia	Etn
1	1	Portugal	This is ripe and fruity, a wine that is smooth...	Avidagos	87	15.0	Douro	Nal
2	2	US	Tart and snappy, the flavors of lime flesh and...	NaN	87	14.0	Oregon	Willamett Valle
3	3	US	Pineapple rind, lemon pith and orange blossom ...	Reserve Late Harvest	87	13.0	Michigan	Lak Michiga Shor
4	4	US	Much like the regular bottling from 2012, this...	Vintner's Reserve Wild Child Block	87	65.0	Oregon	Willamett Valle
5	5	Spain	Blackberry and raspberry aromas show a typical...	Ars In Vitro	87	15.0	Northern Spain	Navarr
6	6	Italy	Here's a bright, informal red that opens with ...	Belsito	87	16.0	Sicily & Sardinia	Vittori
7	7	France	This dry and restrained wine offers spice in p...	NaN	87	24.0	Alsace	Alsac
8	8	Germany	Savory dried thyme notes accent sunnier flavor...	Shine	87	12.0	Rheinhessen	Nal
9	9	France	This has great depth	Les Natures	87	27.0	Alsace	Alsac

	Unnamed: 0	country	description	designation	points	price	province	region_
			of flavor with its fresh ...					

In [12]:
```
wine_reviews.tail()   #แสดงผล 5 แถวสุดท้าย
```

Out[12]:

	Unnamed: 0	country	description	designation	points	price	province	region_
65494	65494	France	Made from young vines from the Vaulorent porti...	Fourchaume Premier Cru	90	45.0	Burgundy	Chab
65495	65495	Australia	This is a big, fat, almost sweet-tasting Caber...	NaN	90	22.0	South Australia	McLare Va
65496	65496	US	Much improved over the unripe 2005, Fritz's 20...	Estate	90	20.0	California	D Cre Vall
65497	65497	US	This wine wears its 15.8% alcohol better than ...	Block 24	90	31.0	California	Nap Vall
65498	65498	Spain	A unique take on Manzanilla Sherry, which is o...	Manzanilla	90	10.0	Andalucia	Jer

The `pd.read_csv()` function is well-endowed, with over 30 optional parameters you can specify. For example, you can see in this dataset that the CSV file has a built-in index, which pandas did not pick up on automatically. To make pandas use that column for the index (instead of creating a new one from scratch), we can specify an `index_col`.

In [13]:
```
wine_reviews = pd.read_csv("datasets/winemag-data-130k-v2.csv", index_col=0)
wine_reviews.head()
```

Out[13]:

	country	description	designation	points	price	province	region_1	region_2	t
0	Italy	Aromas include tropical fruit, broom, brimston...	Vulkà Bianco	87	NaN	Sicily & Sardinia	Etna	NaN	
1	Portugal	This is ripe and fruity, a wine that is smooth...	Avidagos	87	15.0	Douro	NaN	NaN	
2	US	Tart and snappy, the flavors of lime flesh and...	NaN	87	14.0	Oregon	Willamette Valley	Willamette Valley	
3	US	Pineapple rind, lemon pith and orange blossom ...	Reserve Late Harvest	87	13.0	Michigan	Lake Michigan Shore	NaN	
4	US	Much like the regular bottling from 2012, this...	Vintner's Reserve Wild Child Block	87	65.0	Oregon	Willamette Valley	Willamette Valley	

Your turn

If you haven't started the exercise, you can **get started here**.