02 | The *pandas* Library

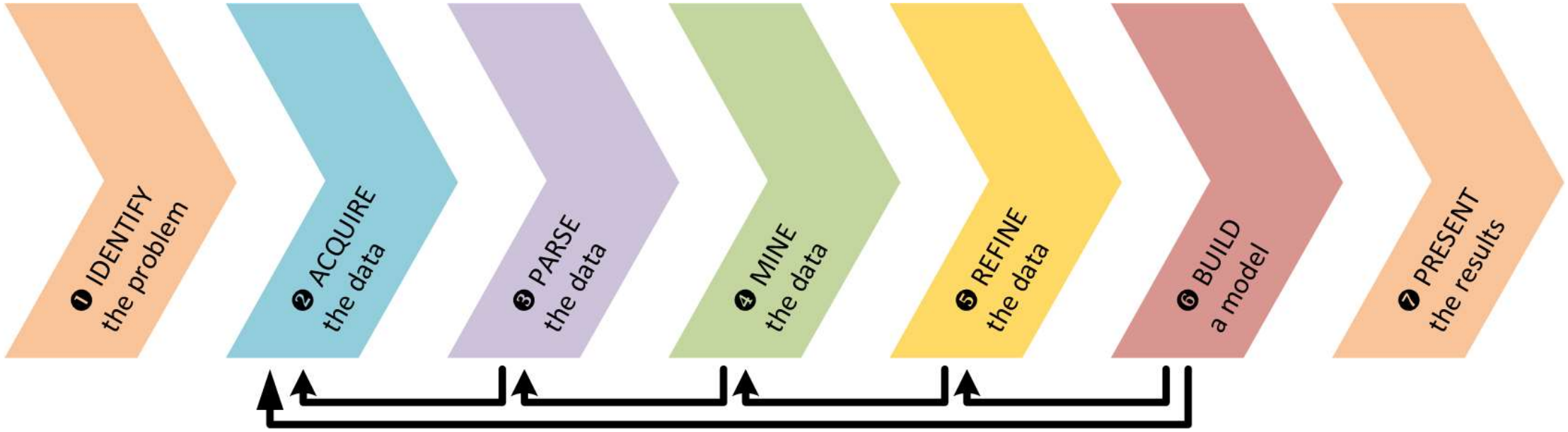
Ivan Corneillet

Data Scientist

GENERAL ASSEMBLY

Models, Feature Matrix X, Response Vector y, and Tidy Data

The Data Science Workflow



When reaching the ❻ **BUILD a model** step, our data needs to be **tidy** and in the form of a **feature matrix $\boldsymbol{X}$** (i.e., the stimuli, e.g., *"ring bell"*) and a **response vector $\boldsymbol{y}$** (i.e., the response, e.g., *"dog salivates"*)

Feature Matrix X

	col0	col1	col2	col3
row0				
row1				
row2				
row3				

Response Vector y

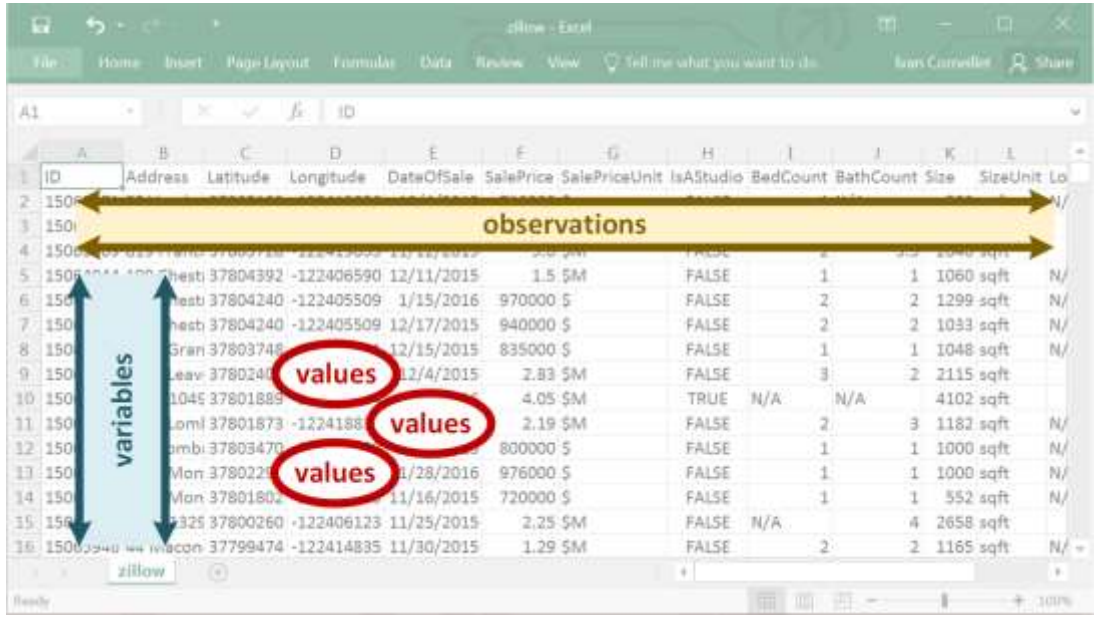
	col
row0	
row1	
row2	
row3	

The San Francisco Housing Dataset: a dataset we will use throughout this course

- Recently Sold Homes (Source: Zillow)
  - 1,000 homes sold in San Francisco between 11/10/2015 and 2/12/2016

What is Tidy Data?

- Your data is tidy if you follow these three rules:
  - Each **observation** (or **sample**) in the dataset is placed in its own **row**
  - Each **variable** (or **feature**) is placed in its own **column**
  - Each **value** is placed in its own **cell**

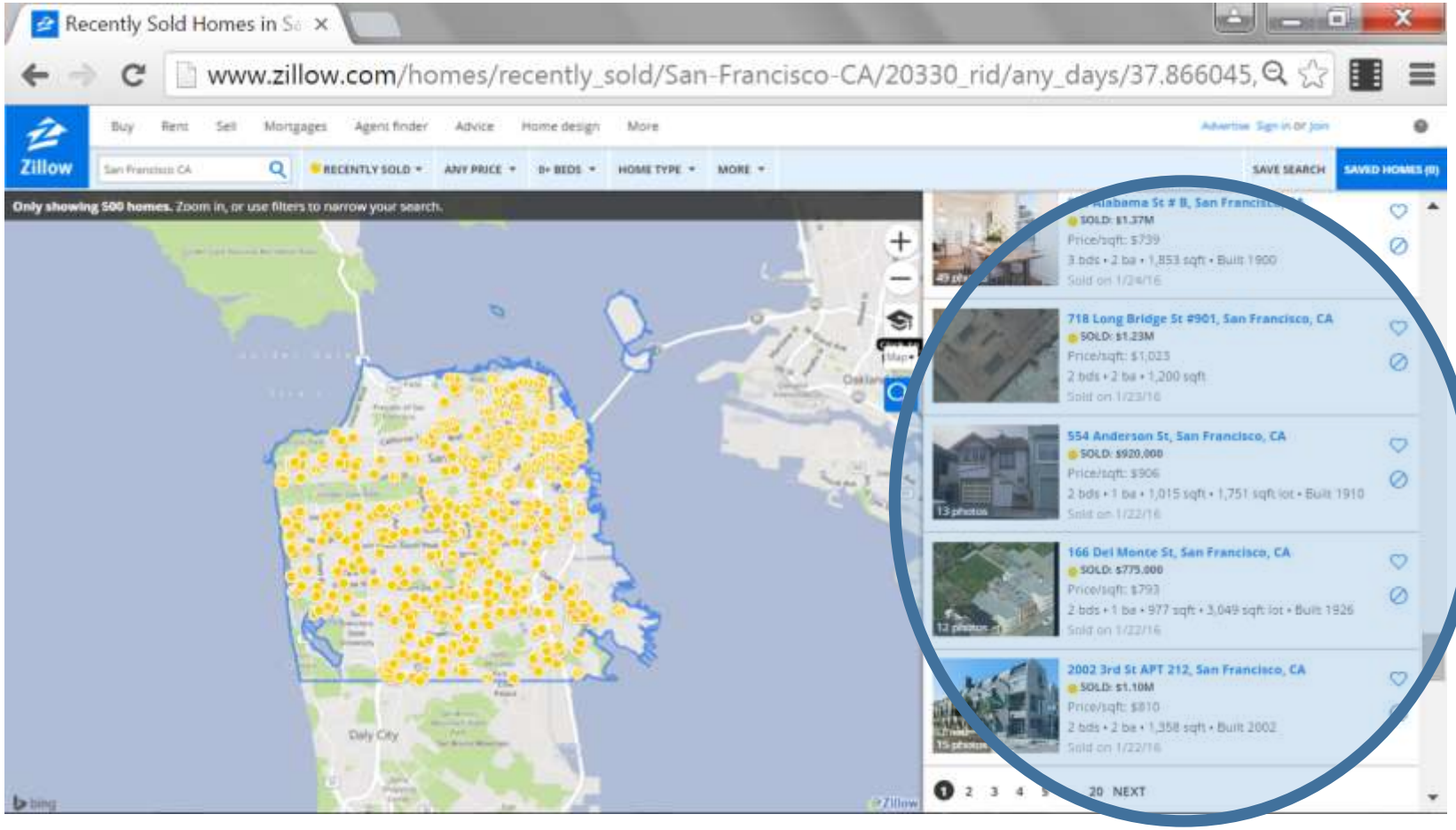


Unfortunately, the output of step ❷ **ACQUIRE** the **data** will usually be **raw**, i.e., **unstructured**…


```
<div class="property-info"
id="yui_3_18_1_1_1456167242885_71870"><strong
id="yui_3_18_1_1_1456167242885_71869"><dt class="property-address"
id="yui_3_18_1_1_1456167242885_71868"><a href="/homedetails/149-
Shipley-St-San-Francisco-CA-94107/15147894_zpid/" class="hdp-link
routable" title="149 Shipley St, San Francisco, CA Real Estate"
id="yui_3_18_1_1_1456167242885_71873">149 Shipley St, San
Francisco, CA</a></dt></strong><dt class="listing-type zsg-
content_collapsed" id="yui_3_18_1_1_1456167242885_71875"><span
class="zsg-icon-recently-sold type-icon"></span>Sold:
$1.18M</dt><dt class="zsg-fineprint"
id="yui_3_18_1_1_1456167242885_71877">Price/sqft: $1,116</dt><dt
class="property-data" id="yui_3_18_1_1_1456167242885_71880"><span
class="beds-baths-sqft">3 bds • 2 ba • 1,057 sqft</span><span
class="built-year" id="yui_3_18_1_1_1456167242885_71879"> • Built
1992</span></dt><dt class="sold-date zsg-fineprint"
id="yui_3_18_1_1_1456167242885_71975">Sold on 2/22/16</dt></div>
```

(E.g., unstructured data scrapped from websites)

CASE STUDY



… and/or **messy**…

- Trouble tickets inspect and maintain manholes in New Year City
- "Service box," a common piece of infrastructure, had at least 38 variants, including `SB`, `S`, `S/B`, `S.B`, `S?B`, `S.B.`, `SBX`, `S/BX`, `SB/X`, `S/XB`, `/SBX`, `S.BX`, `S &BX`, `S?BX`, `S BX`, `S/B/X`, `S BOX`, `SVBX`, `SERV BX`, `SERV-BOX`, `SERV/BOX`, and `SERVICE BOX`

Source: Big Data: A Revolution That Will Transform How We Live, Work, and Think

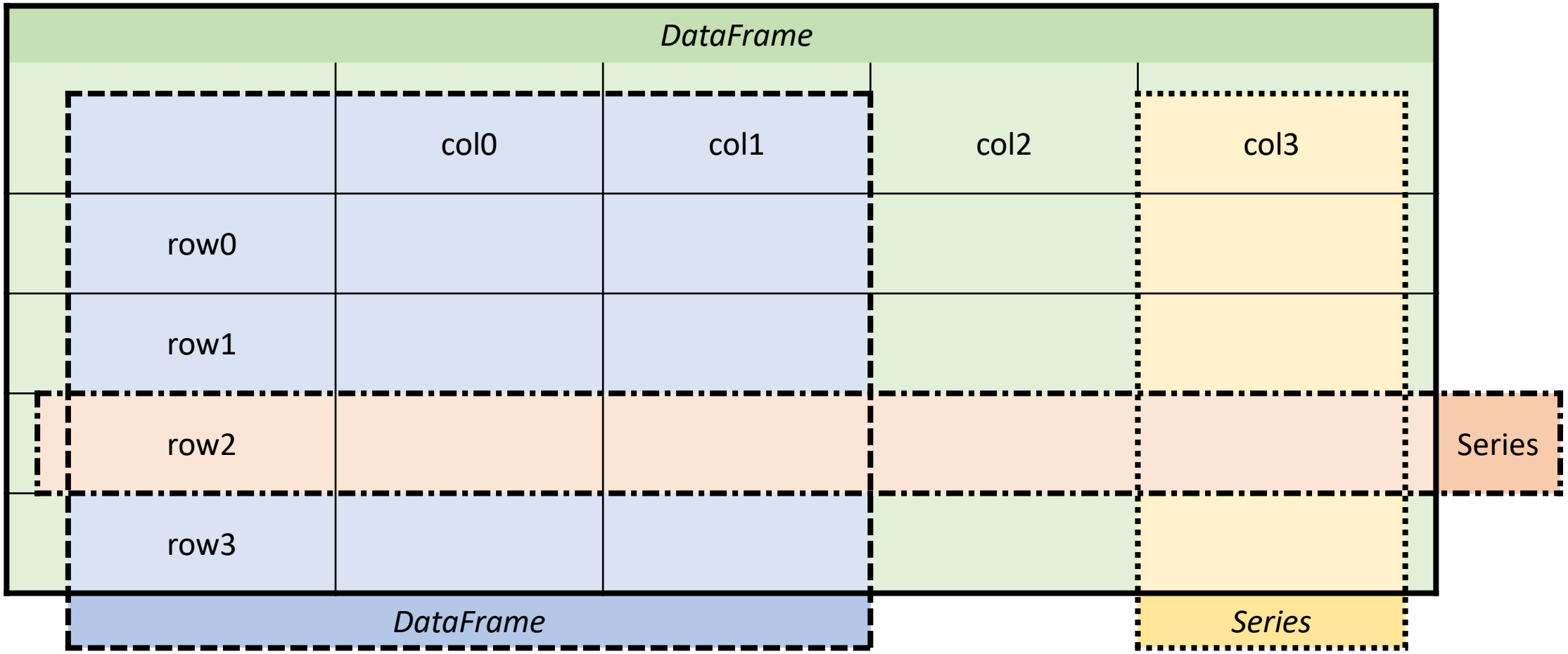
Question: What tool can we use to wrangle **raw data** into **tidy data**?

- Answer: *pandas*
  - *pandas* is a Python library that provides the ability to *index*, *retrieve*, *tidy*, *reshape*, *combine*, *slice*, tabular and other multidimensional datasets
  - *pandas* also provides facilities to perform statistical and mathematical analysis which will come handy for exploratory data analysis

- Wrangling data is the most fruitful skill you can learn as a data scientist. It will save you hours of time and make your data much easier to visualize, manipulate, and model
- Today, we will use *pandas* to explore and manipulate the San Francisco housing dataset

The *pandas* Library

pandas.DataFrame and *pandas.Series* (cont.)



When reaching the ❻ **BUILD a model** step, our **feature matrix $\boldsymbol{X}$** will be modeled as a **DataFrame** and the **response vector $\boldsymbol{y}$** as a **Series**

Feature Matrix X

DataFrame

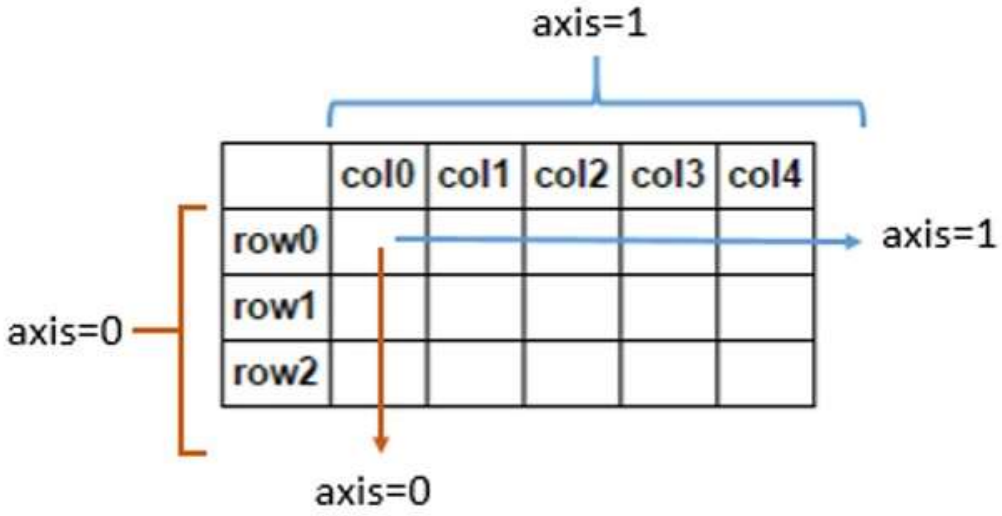
	col0	col1	col2	col3
row0				
row1				
row2				
row3				

Response Vector y

Series

	col
row0	
row1	
row2	
row3	

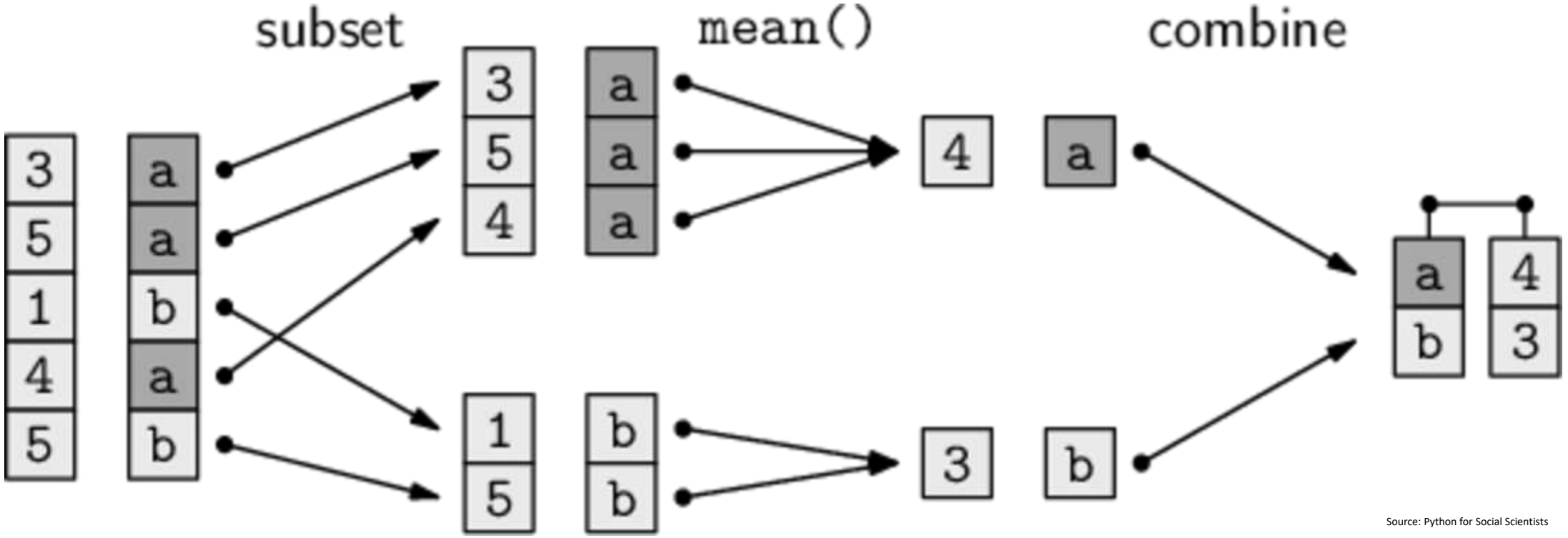
(a note on *pandas* axes)



Source: Stack Overflow

- Axes are defined for arrays with more than one dimension, e.g., a DataFrame
- A DataFrame has two corresponding axes: the first running **vertically downwards across rows (axis 0)**, and the second running **horizontally across columns (axis 1)**

(another note of the split-apply-combine performed by `.groupby()`)



Source: Python for Social Scientists