1.05 Pandas I & Pandas II

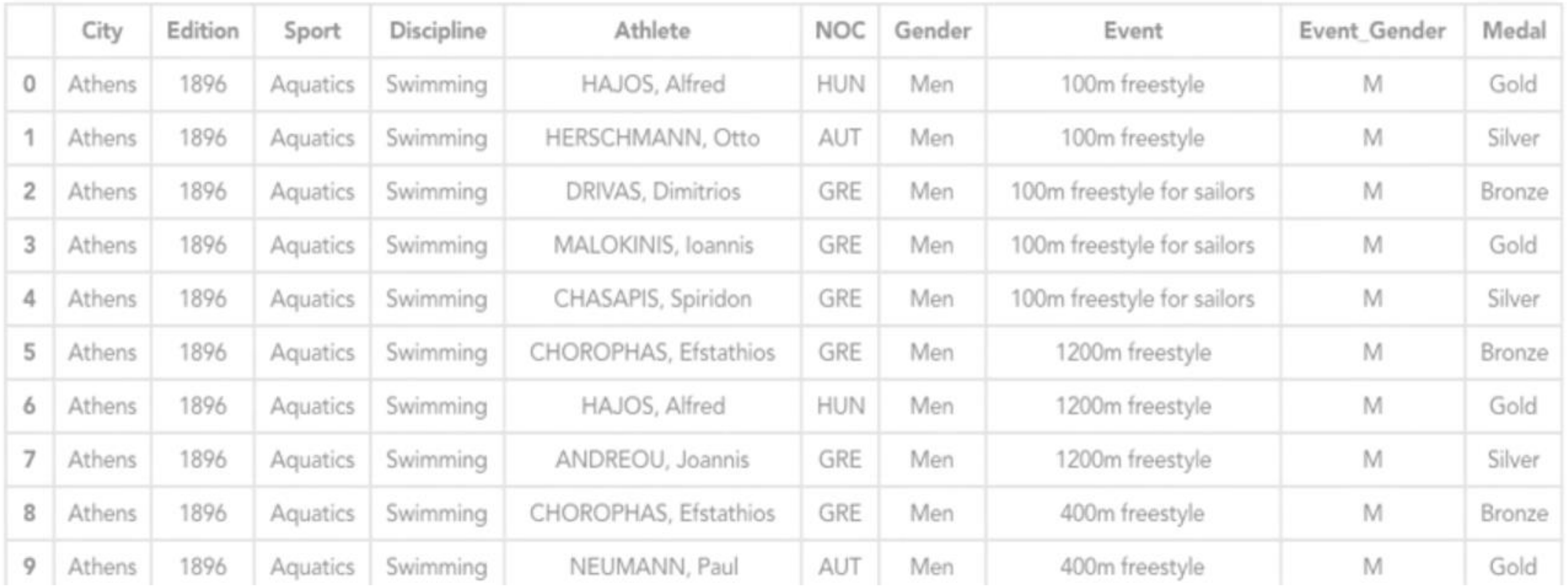
Importing Pandas

- Import pandas
  - pandas.<method name>

- Import pandas as pd
  - pd.<method name>

Data Frames

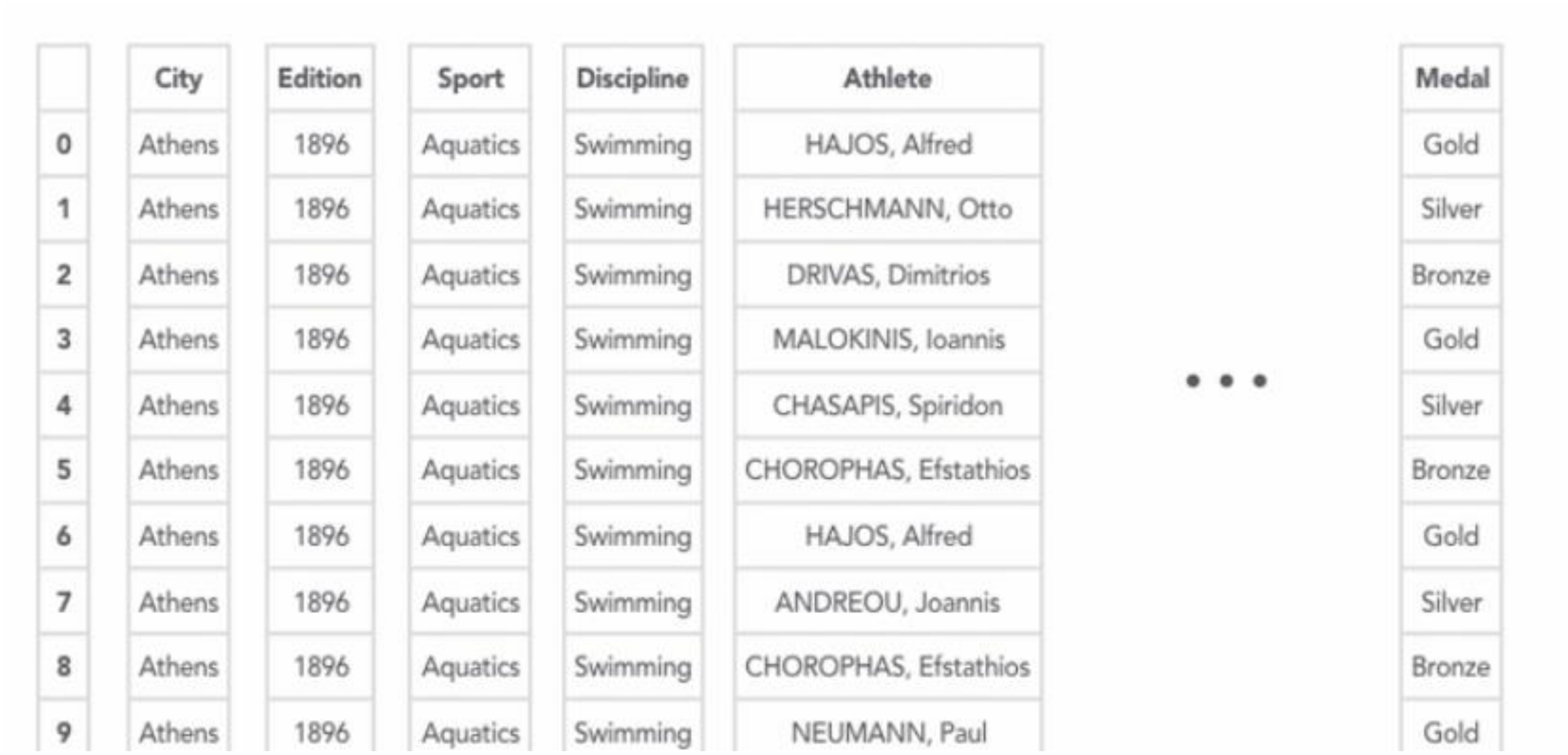
- A Data Frame is a 2-dimensional array
- It is a sequence of series that share the same index



	City	Edition	Sport	Discipline	Athlete	NOC	Gender	Event	Event_Gender	Medal
0	Athens	1896	Aquatics	Swimming	HAJOS, Alfred	HUN	Men	100m freestyle	M	Gold
1	Athens	1896	Aquatics	Swimming	HERSCHMANN, Otto	AUT	Men	100m freestyle	M	Silver
2	Athens	1896	Aquatics	Swimming	DRIVAS, Dimitrios	GRE	Men	100m freestyle for sailors	M	Bronze
3	Athens	1896	Aquatics	Swimming	MALOKINIS, Ioannis	GRE	Men	100m freestyle for sailors	M	Gold
4	Athens	1896	Aquatics	Swimming	CHASAPIS, Spiridon	GRE	Men	100m freestyle for sailors	M	Silver
5	Athens	1896	Aquatics	Swimming	CHOROPHAS, Efstathios	GRE	Men	1200m freestyle	M	Bronze
6	Athens	1896	Aquatics	Swimming	HAJOS, Alfred	HUN	Men	1200m freestyle	M	Gold
7	Athens	1896	Aquatics	Swimming	ANDREOU, Joannis	GRE	Men	1200m freestyle	M	Silver
8	Athens	1896	Aquatics	Swimming	CHOROPHAS, Efstathios	GRE	Men	400m freestyle	M	Bronze
9	Athens	1896	Aquatics	Swimming	NEUMANN, Paul	AUT	Men	400m freestyle	M	Gold

Series

- Series is a one-dimensional array of indexed data



	City	Edition	Sport	Discipline	Athlete		Medal
0	Athens	1896	Aquatics	Swimming	HAJOS, Alfred		Gold
1	Athens	1896	Aquatics	Swimming	HERSCHMANN, Otto		Silver
2	Athens	1896	Aquatics	Swimming	DRIVAS, Dimitrios		Bronze
3	Athens	1896	Aquatics	Swimming	MALOKINIS, Ioannis		Gold
4	Athens	1896	Aquatics	Swimming	CHASAPIS, Spiridon	• • •	Silver
5	Athens	1896	Aquatics	Swimming	CHOROPHAS, Efstathios		Bronze
6	Athens	1896	Aquatics	Swimming	HAJOS, Alfred		Gold
7	Athens	1896	Aquatics	Swimming	ANDREOU, Joannis		Silver
8	Athens	1896	Aquatics	Swimming	CHOROPHAS, Efstathios		Bronze
9	Athens	1896	Aquatics	Swimming	NEUMANN, Paul		Gold

Series

- Accessing a single Series via
  - DataFrame[‘SeriesName’]
  - DataFrame[“SeriesName”]
  - DataFrame.SeriesName
- Accessing multiple Series
  - DataFrame[[‘SeriesName1’,’SeriesName2’]]

Data Input

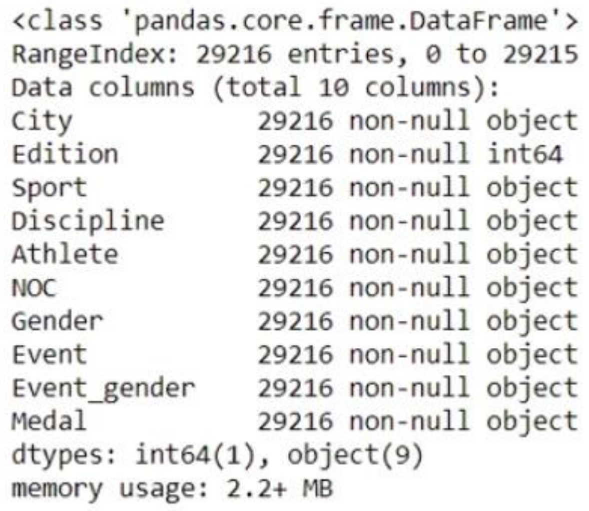
- Input
  - read_excel(…)
  - read_json(…)
  - Read_sql_table(…)

- Read a CSV file into a DataFrame
  - pandas.read_csv(filepath)

Data Frame – Useful Methods

- DataFrame.shape ➔ Returns number of rows and columns
- DataFrame.head(n) ➔ Returns first n rows
- DataFrame.tail(n) ➔ Returns last n rows

- DataFrame.info() ➔ Returns number of values, null status of columns



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 29216 entries, 0 to 29215
Data columns (total 10 columns):
City            29216 non-null object
Edition         29216 non-null int64
Sport           29216 non-null object
Discipline      29216 non-null object
Athlete         29216 non-null object
NOC             29216 non-null object
Gender          29216 non-null object
Event           29216 non-null object
Event_gender    29216 non-null object
Medal           29216 non-null object
dtypes: int64(1), object(9)
memory usage: 2.2+ MB
```

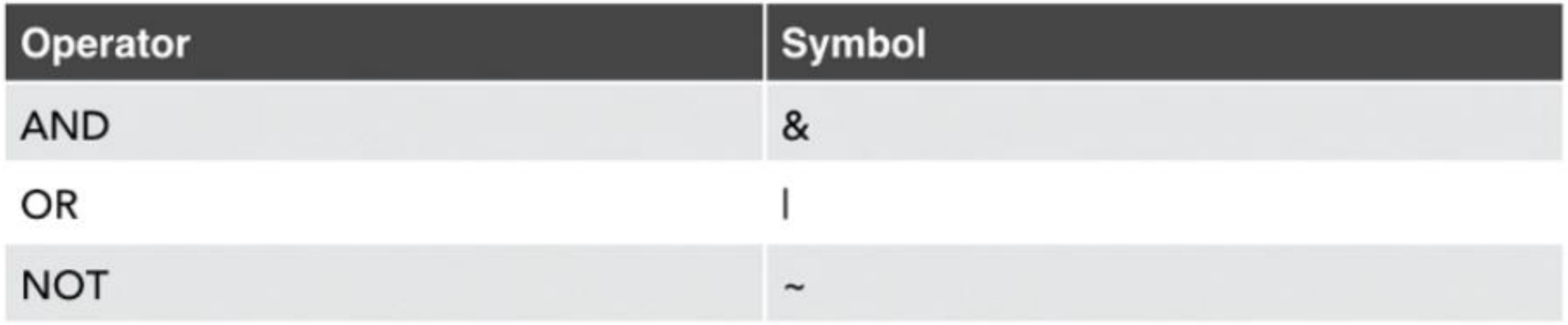
Data Frame – Useful Methods

Series.value_counts()

- Returns counts of unique values for that series

Boolean Indexing

- Boolean vectors (symbols) can be used to filter data
- Multiple conditions must be grouped using brackets



Operator	Symbol
AND	&
OR	|
NOT	~

- Example: df[(df.Medal == ‘Gold’) & (df.Gender == ‘Women’)]

String Handling

- Available to every Series using the str attribute

- Series.str – access values of series as strings and apply several methods to it

- Examples
  - Series.str.contains()
  - Series.str.startswith()
  - Series.str.isnumeric()

loc[]

- DataFrame.loc[]
- A label-based indexer for selection by label
- loc[] will raise a KeyError when the items are not found

iloc[]

- DataFrame.iloc[]
- iloc[] is primarily integer position based (from 0 to length-1 of the axis)
- Facilitates slicing of data

groupby

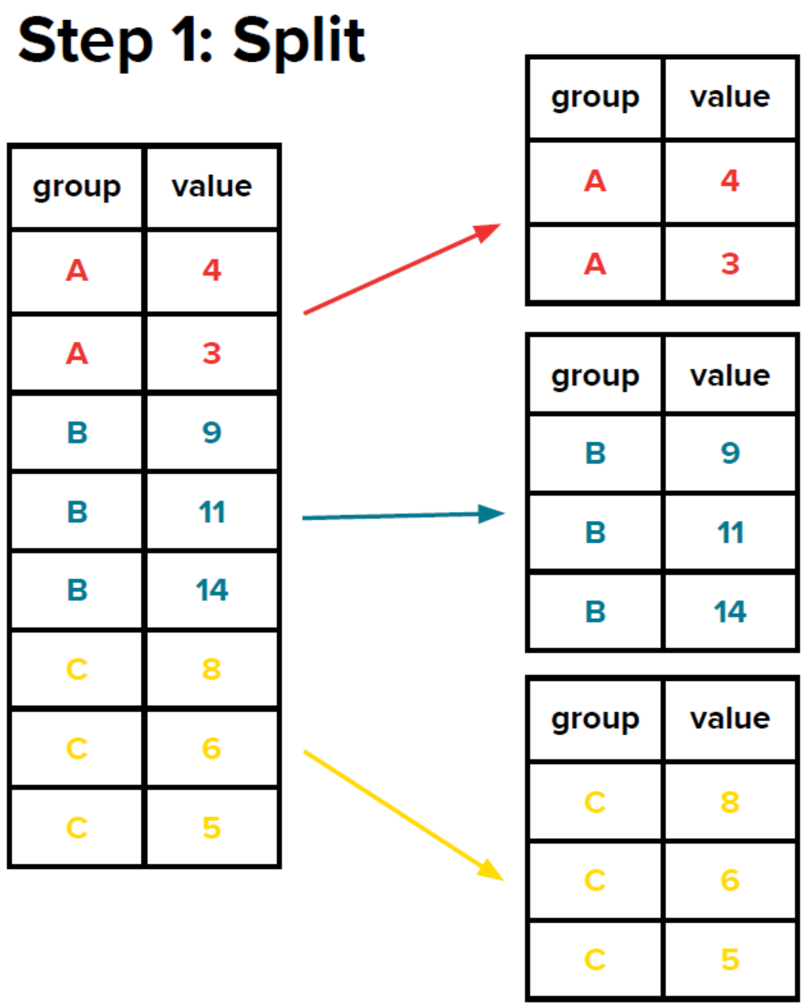
- pandas.DataFrame.groupby(‘column_name’)
- How GroupBy works
    - Split a DataFrame into groups based on some criteria
    - Apply a function to each group independently
    - Combine the results into a DataFrame

groupby via Split-Apply-Combine

Suppose we want to find the mean “value” per “group”

group	value
A	4
A	3
B	9
B	11
B	14
C	8
C	6
C	5

groupby via Split-Apply-Combine



group	value
A	4
A	3
B	9
B	11
B	14
C	8
C	6
C	5

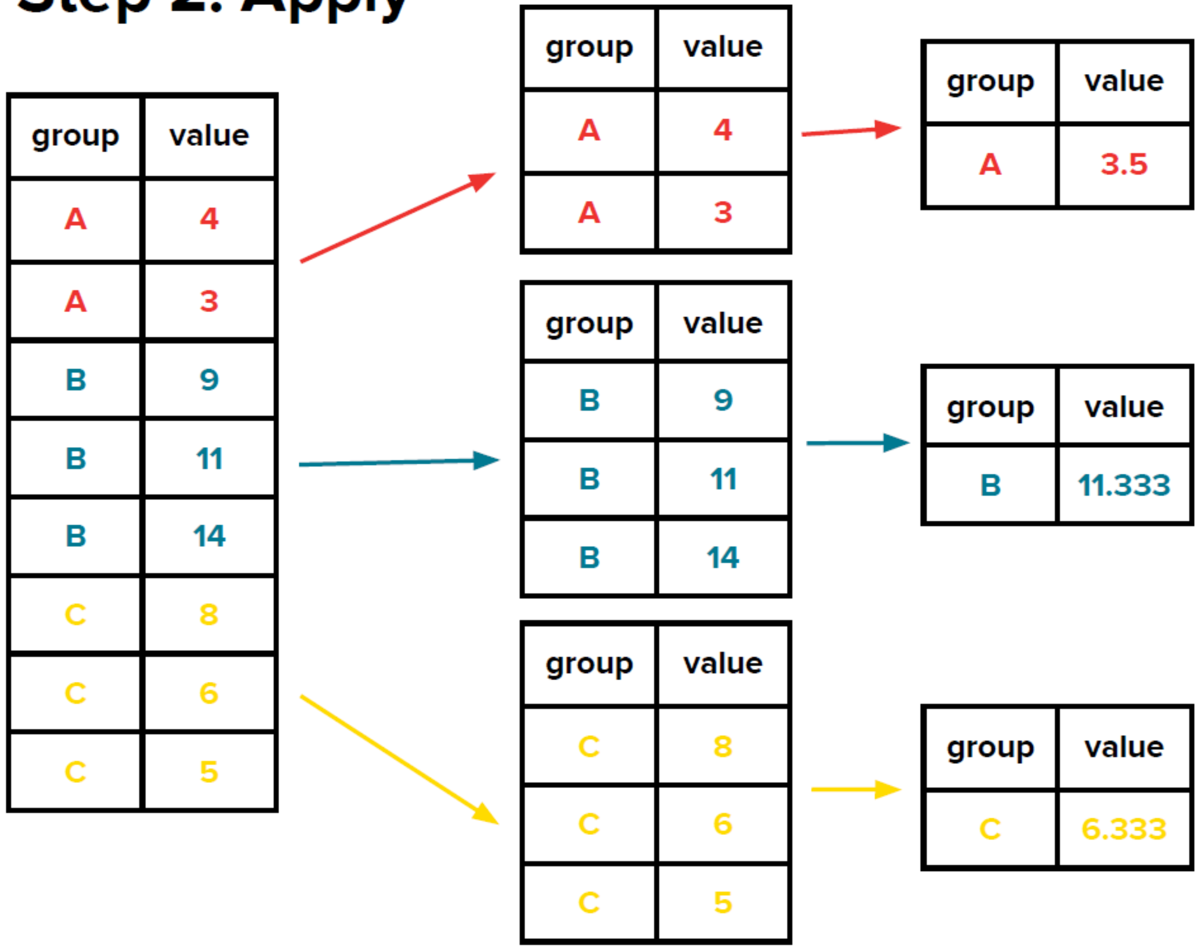
group	value
A	4
A	3

group	value
B	9
B	11
B	14

group	value
C	8
C	6
C	5

groupby via Split-Apply-Combine

Step 2: Apply



group	value
A	4
A	3
B	9
B	11
B	14
C	8
C	6
C	5

group	value
A	4
A	3

group	value
A	3.5

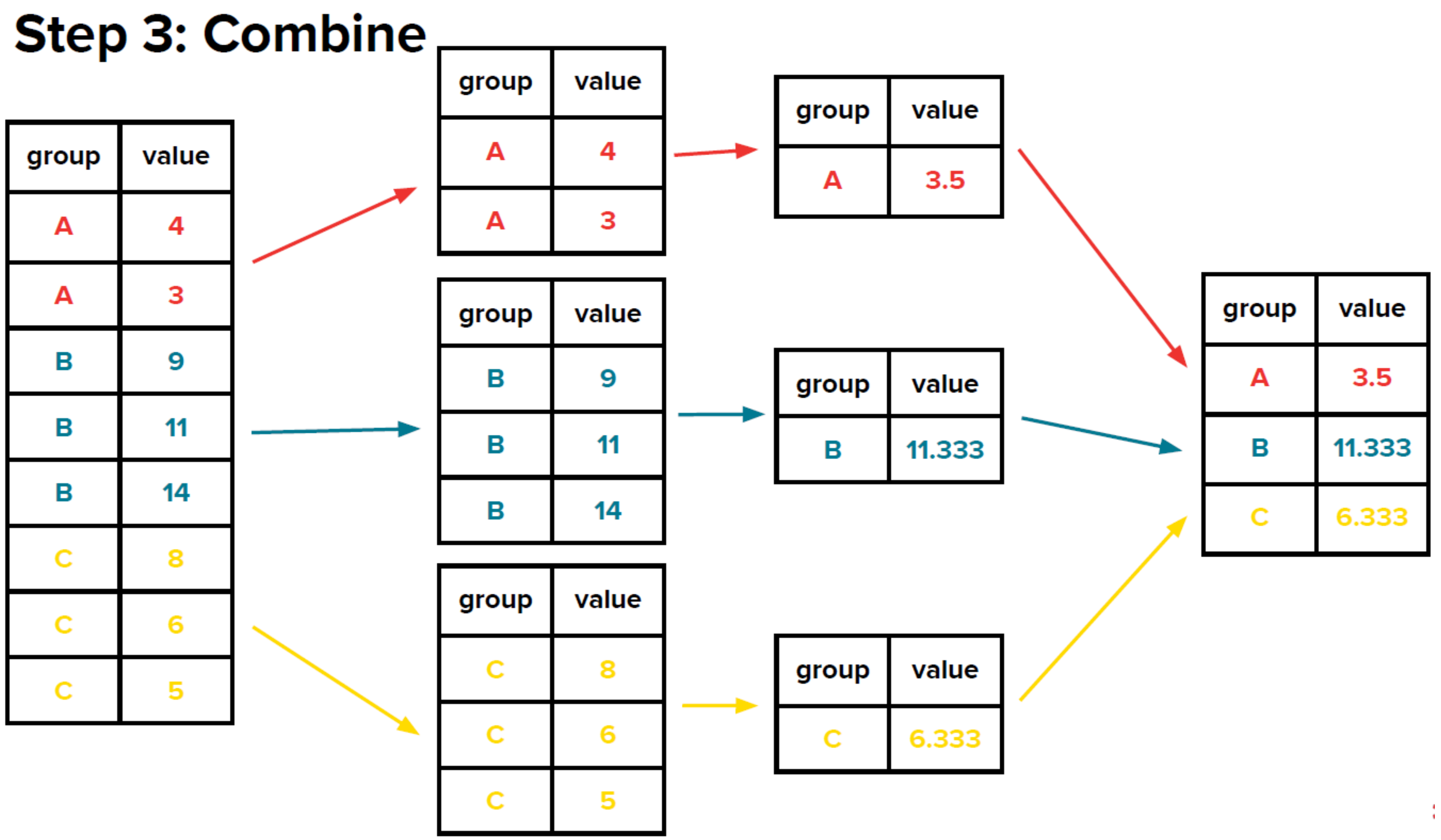
group	value
B	9
B	11
B	14

group	value
B	11.333

group	value
C	8
C	6
C	5

group	value
C	6.333

groupby via Split-Apply-Combine



Step 3: Combine

group	value
A	4
A	3
B	9
B	11
B	14
C	8
C	6
C	5

group	value
A	4
A	3

group	value
B	9
B	11
B	14

group	value
C	8
C	6
C	5

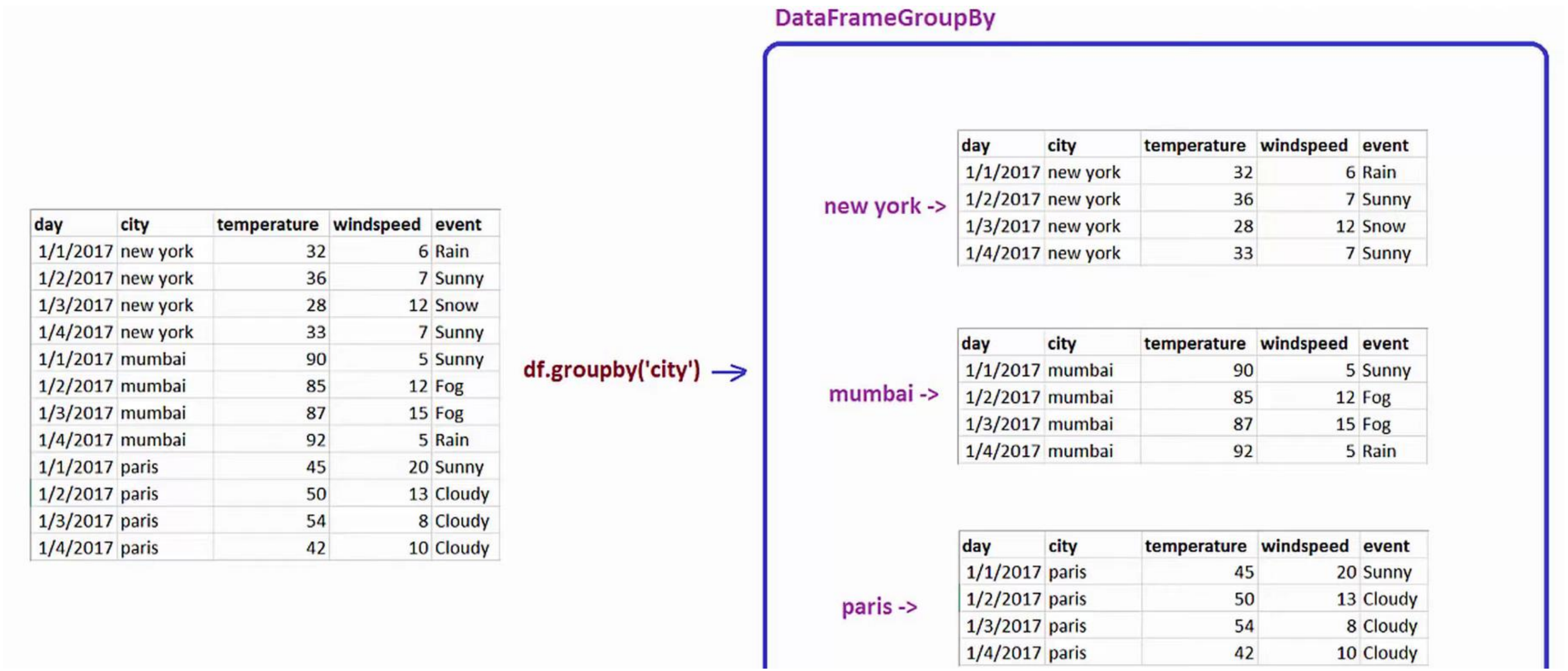
group	value
A	3.5

group	value
B	11.333

group	value
C	6.333

group	value
A	3.5
B	11.333
C	6.333

Group By (Split Apply Combine) - Other Examples



day	city	temperature	windspeed	event
1/1/2017	new york	32	6	Rain
1/2/2017	new york	36	7	Sunny
1/3/2017	new york	28	12	Snow
1/4/2017	new york	33	7	Sunny
1/1/2017	mumbai	90	5	Sunny
1/2/2017	mumbai	85	12	Fog
1/3/2017	mumbai	87	15	Fog
1/4/2017	mumbai	92	5	Rain
1/1/2017	paris	45	20	Sunny
1/2/2017	paris	50	13	Cloudy
1/3/2017	paris	54	8	Cloudy
1/4/2017	paris	42	10	Cloudy

df.groupby('city') →

DataFrameGroupBy

new york ->

day	city	temperature	windspeed	event
1/1/2017	new york	32	6	Rain
1/2/2017	new york	36	7	Sunny
1/3/2017	new york	28	12	Snow
1/4/2017	new york	33	7	Sunny

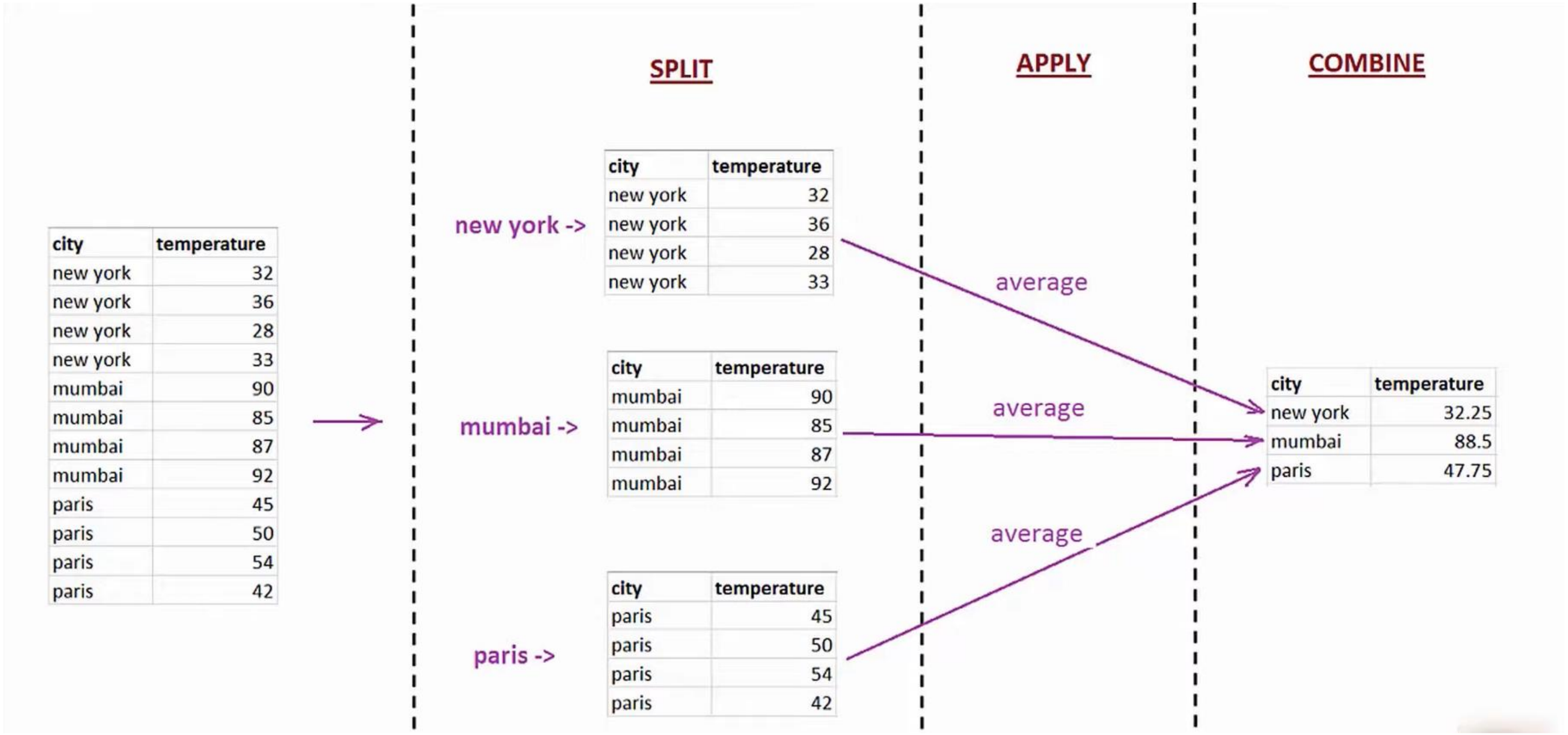
mumbai ->

day	city	temperature	windspeed	event
1/1/2017	mumbai	90	5	Sunny
1/2/2017	mumbai	85	12	Fog
1/3/2017	mumbai	87	15	Fog
1/4/2017	mumbai	92	5	Rain

paris ->

day	city	temperature	windspeed	event
1/1/2017	paris	45	20	Sunny
1/2/2017	paris	50	13	Cloudy
1/3/2017	paris	54	8	Cloudy
1/4/2017	paris	42	10	Cloudy

Group By (Split Apply Combine) - Other Examples



city	temperature
new york	32
new york	36
new york	28
new york	33
mumbai	90
mumbai	85
mumbai	87
mumbai	92
paris	45
paris	50
paris	54
paris	42

city	temperature
new york	32
new york	36
new york	28
new york	33

city	temperature
mumbai	90
mumbai	85
mumbai	87
mumbai	92

city	temperature
paris	45
paris	50
paris	54
paris	42

city	temperature
new york	32.25
mumbai	88.5
paris	47.75