DATA SCIENCE 102:
PANDAS PART 2

AGENDA

- Data Transformation
- Data Aggregation

DATA TRANSFORMATION

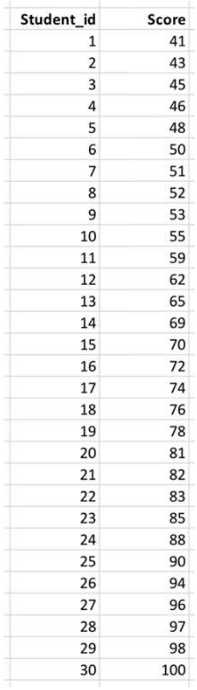
- .apply()
- def vs. lambda

DATA TRANSFORMATION

- In computing, data transformation is the process of converting data from one format or structure into another format or structure
- Scaling
- Word stemming
- Many other possibilities

DATA TRANSFORMATION

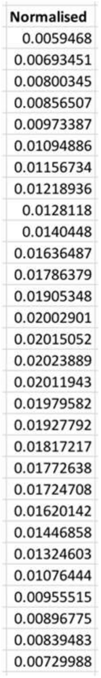
You have the results of a recent test your class took. You wish to analyse the results and fit them into a bell curve (gaussian distribution)



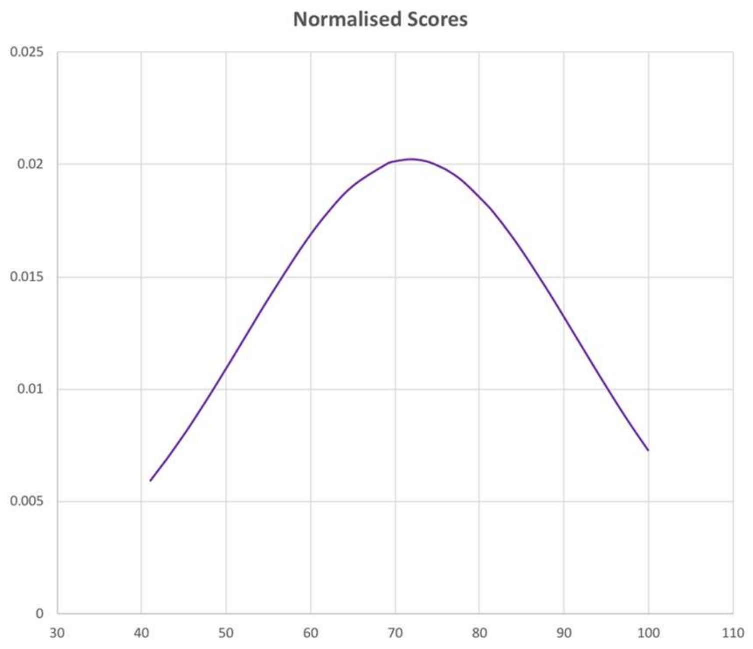
Student_id	Score
1	41
2	43
3	45
4	46
5	48
6	50
7	51
8	52
9	53
10	55
11	59
12	62
13	65
14	69
15	70
16	72
17	74
18	76
19	78
20	81
21	82
22	83
23	85
24	88
25	90
26	94
27	96
28	97
29	98
30	100

The Normal Probability Density Function

$$f_x(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]}$$



Normalised
0.0059468
0.00693451
0.00800345
0.00856507
0.00973387
0.01094886
0.01156734
0.01218936
0.0128118
0.0140448
0.01636487
0.01786379
0.01905348
0.02002901
0.02015052
0.02023889
0.02011943
0.01979582
0.01927792
0.01817217
0.01772638
0.01724708
0.01620142
0.01446858
0.01324603
0.01076444
0.00955515
0.00896775
0.00839483
0.00729988



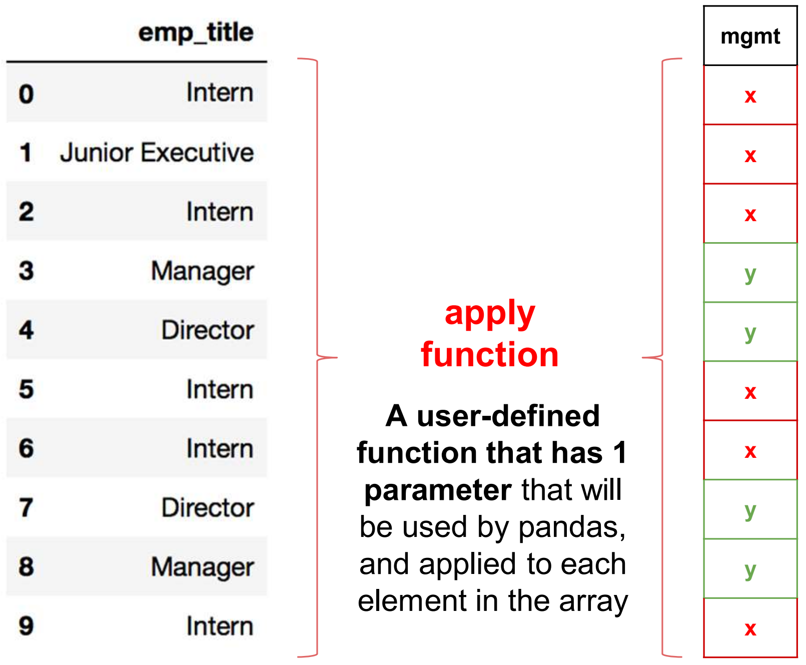
DATA TRANSFORMATION - .apply()

Example

We have a column representing the job titles of employees.

Say we wish to transform the column to have values reflecting if a given employee is either management or a rank and file employee:

- **‘Management’** if employee is manager / director,
- **‘Rank and file’** otherwise



	emp_title
0	Intern
1	Junior Executive
2	Intern
3	Manager
4	Director
5	Intern
6	Intern
7	Director
8	Manager
9	Intern

apply function

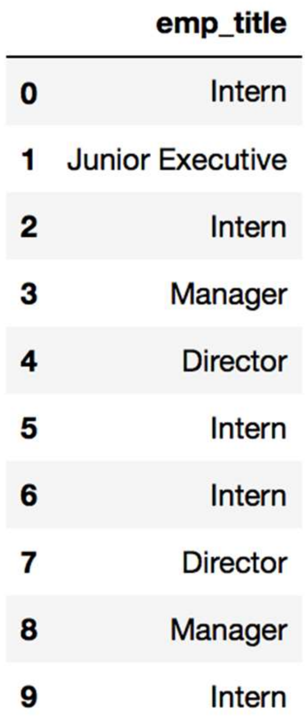
A user-defined function that has 1 parameter that will be used by pandas, and applied to each element in the array

mgmt
x
x
x
y
y
x
x
y
y
x

DATA TRANSFORMATION - .apply()

Example

The .apply method of a Series accepts a user-defined function, and will be applied to each cell in a Series.



	emp_title
0	Intern
1	Junior Executive
2	Intern
3	Manager
4	Director
5	Intern
6	Intern
7	Director
8	Manager
9	Intern

.apply(**function**)

DATA TRANSFORMATION - .apply()

Syntax

```
df['y_col'].apply(function)
```

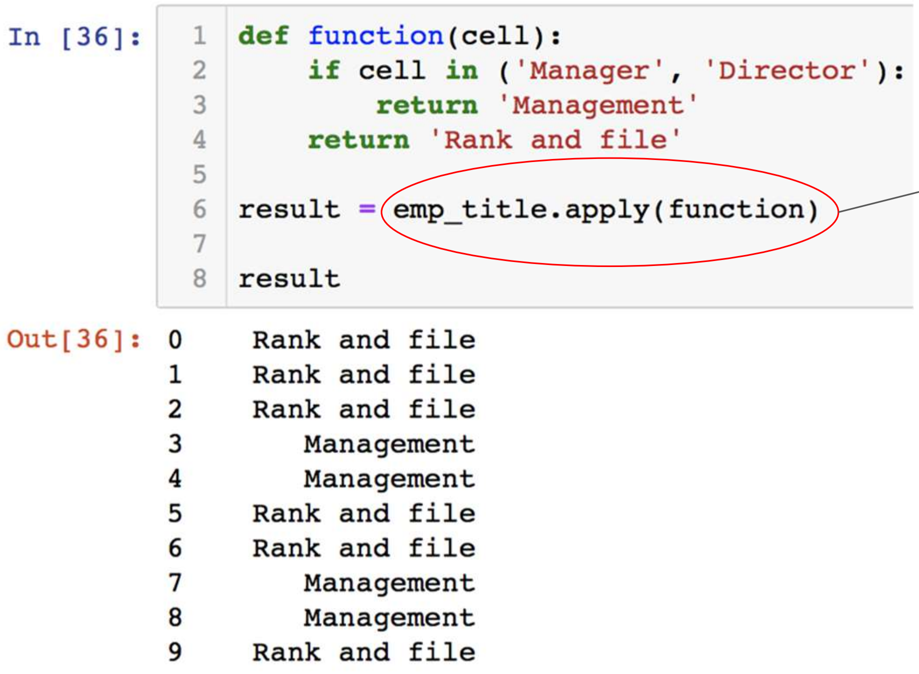
$$\underbrace{\texttt{df['y\_col']}}_{\texttt{Series}}.\underbrace{\texttt{apply}}_{\texttt{method}}(\underbrace{\textit{function}}_{})$$

User-defined function

Pandas will use the function and **call** it on every *cell* in the Series.

Note that the function needs to take in **at least one** parameter.

DATA TRANSFORMATION - .apply()



```
In [36]: def function(cell):
             if cell in ('Manager', 'Director'):
                 return 'Management'
             return 'Rank and file'

         result = emp_title.apply(function)

         result

Out[36]: 0    Rank and file
         1    Rank and file
         2    Rank and file
         3       Management
         4       Management
         5    Rank and file
         6    Rank and file
         7       Management
         8       Management
         9    Rank and file
```

The .apply method takes accepts a function, then calls that function on each cell in the Series.

Finally, it will return a Series with the transformed values, like so.

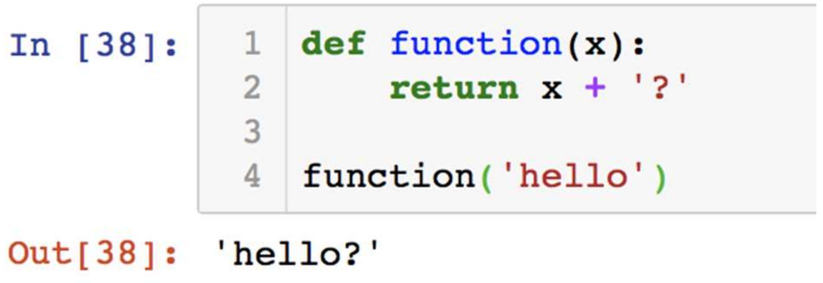
It is a very common way of transforming data.

DATA TRANSFORMATION - def vs. lambda

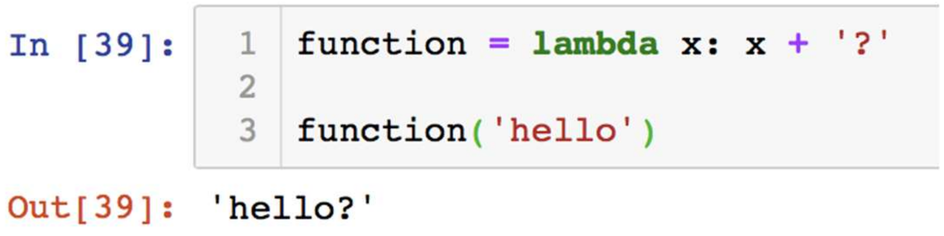
def and lambda do **essentially the same thing**. Lambda is simply a fancy word for **"make function"**.

Lambdas are used when you wish to write a **simple**, **one-line function**, to pass it into the apply() method.

These two lines are equivalent in python.



```
In [38]:  def function(x):
              return x + '?'

          function('hello')

Out[38]:  'hello?'
```



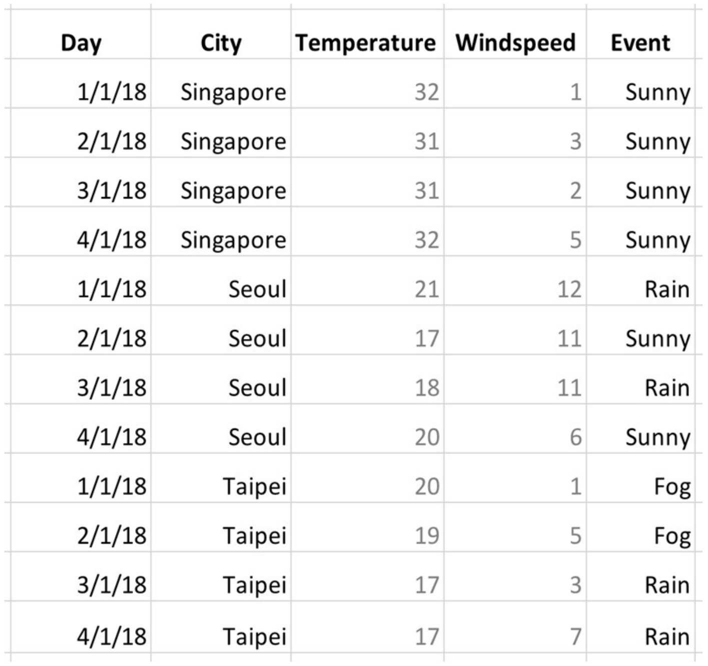
```
In [39]:  function = lambda x: x + '?'

          function('hello')

Out[39]:  'hello?'
```

DATA AGGREGATION

- GroupBy
- Multiple Aggregation

DATA AGGREGATION - GROUPBY

Data aggregation is a type of data and information mining process where data is searched, gathered and is in a summarized format for analysis.



Day	City	Temperature	Windspeed	Event
1/1/18	Singapore	32	1	Sunny
2/1/18	Singapore	31	3	Sunny
3/1/18	Singapore	31	2	Sunny
4/1/18	Singapore	32	5	Sunny
1/1/18	Seoul	21	12	Rain
2/1/18	Seoul	17	11	Sunny
3/1/18	Seoul	18	11	Rain
4/1/18	Seoul	20	6	Sunny
1/1/18	Taipei	20	1	Fog
2/1/18	Taipei	19	5	Fog
3/1/18	Taipei	17	3	Rain
4/1/18	Taipei	17	7	Rain

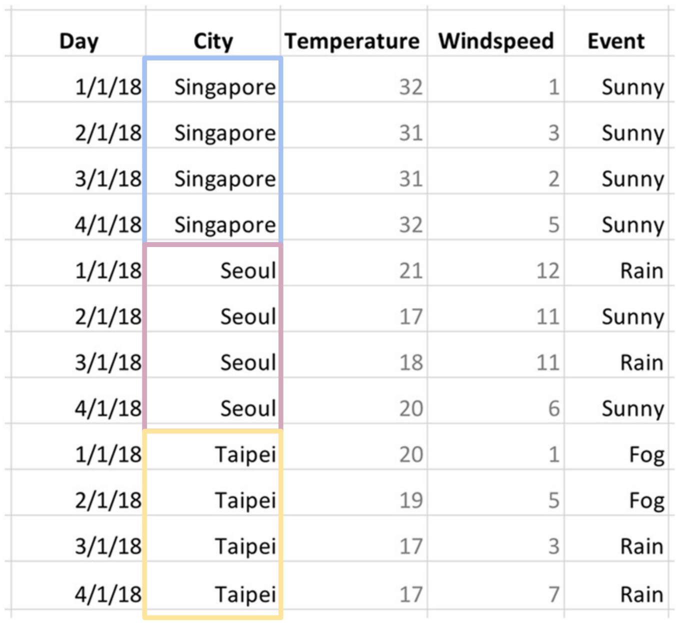
Example

We wish to find each city's mean temperature and windspeed.

We can use the powerful .groupby() method to help us do this.

DATA AGGREGATION - GROUPBY

Find each city's mean temperature and wind speed:



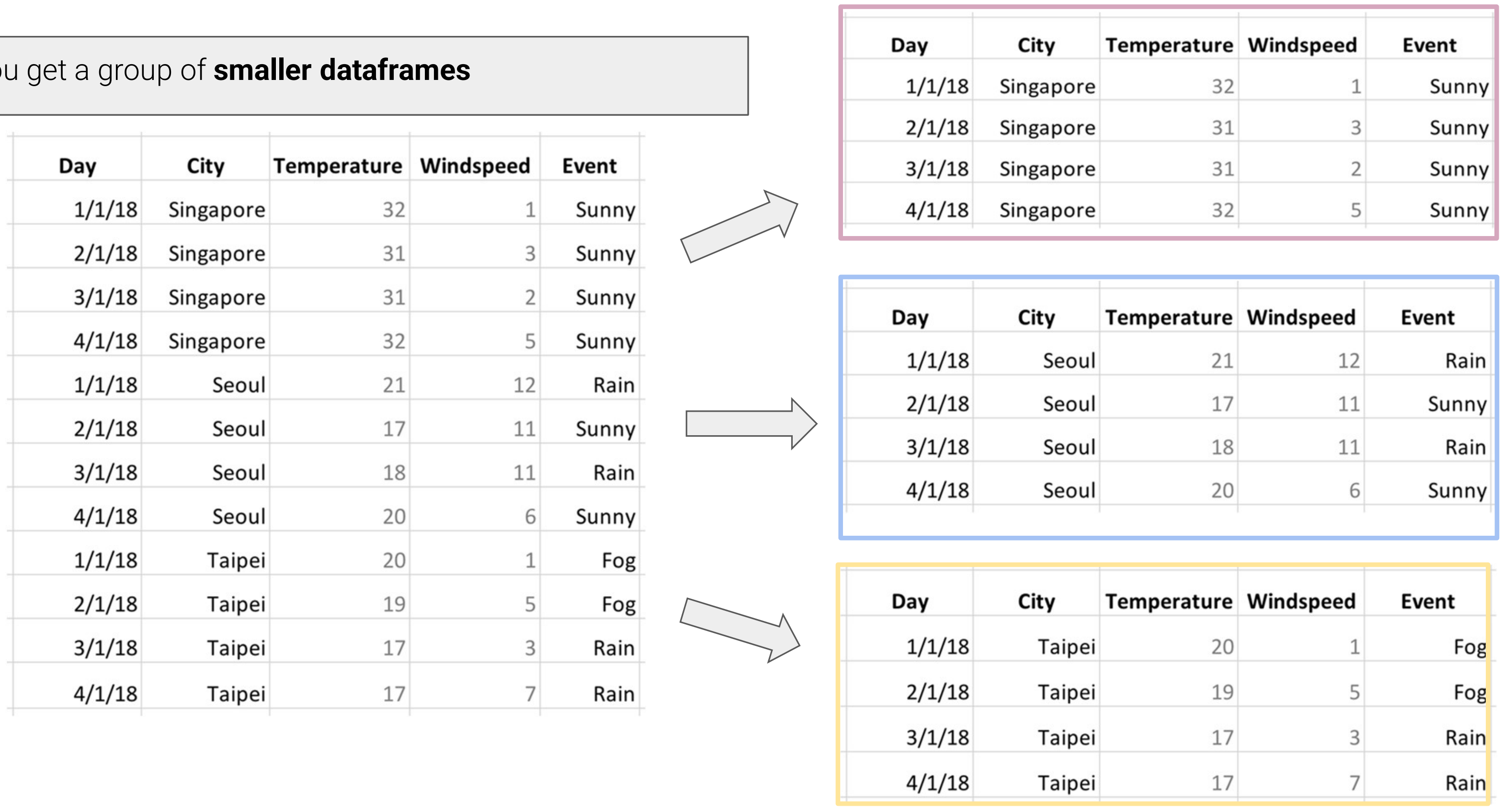
Day	City	Temperature	Windspeed	Event
1/1/18	Singapore	32	1	Sunny
2/1/18	Singapore	31	3	Sunny
3/1/18	Singapore	31	2	Sunny
4/1/18	Singapore	32	5	Sunny
1/1/18	Seoul	21	12	Rain
2/1/18	Seoul	17	11	Sunny
3/1/18	Seoul	18	11	Rain
4/1/18	Seoul	20	6	Sunny
1/1/18	Taipei	20	1	Fog
2/1/18	Taipei	19	5	Fog
3/1/18	Taipei	17	3	Rain
4/1/18	Taipei	17	7	Rain

Syntax

```
cities = df.groupby('City')
```

DATA AGGREGATION - GROUPBY

You get a group of **smaller dataframes**



Day	City	Temperature	Windspeed	Event
1/1/18	Singapore	32	1	Sunny
2/1/18	Singapore	31	3	Sunny
3/1/18	Singapore	31	2	Sunny
4/1/18	Singapore	32	5	Sunny
1/1/18	Seoul	21	12	Rain
2/1/18	Seoul	17	11	Sunny
3/1/18	Seoul	18	11	Rain
4/1/18	Seoul	20	6	Sunny
1/1/18	Taipei	20	1	Fog
2/1/18	Taipei	19	5	Fog
3/1/18	Taipei	17	3	Rain
4/1/18	Taipei	17	7	Rain

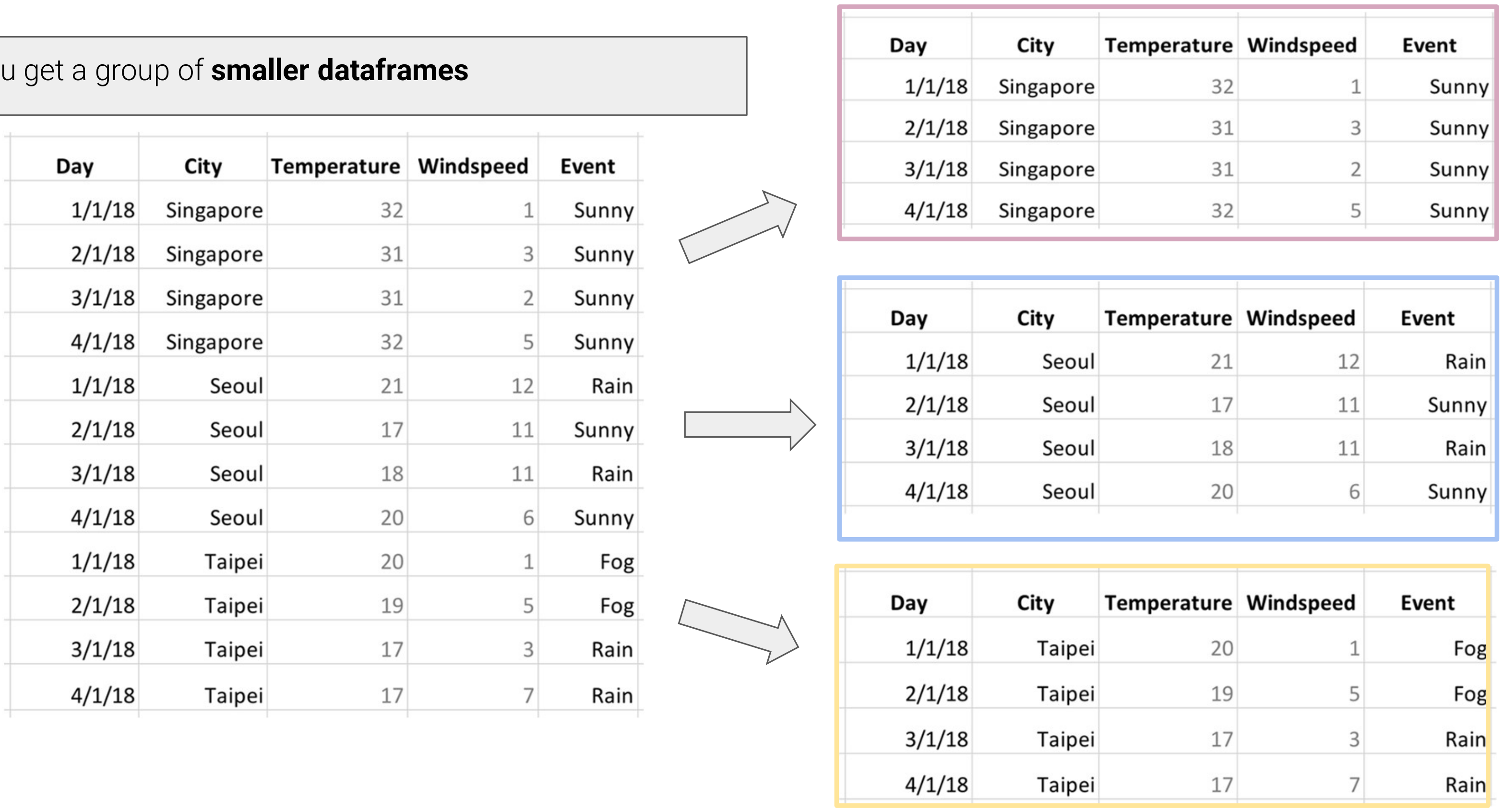
Day	City	Temperature	Windspeed	Event
1/1/18	Singapore	32	1	Sunny
2/1/18	Singapore	31	3	Sunny
3/1/18	Singapore	31	2	Sunny
4/1/18	Singapore	32	5	Sunny

Day	City	Temperature	Windspeed	Event
1/1/18	Seoul	21	12	Rain
2/1/18	Seoul	17	11	Sunny
3/1/18	Seoul	18	11	Rain
4/1/18	Seoul	20	6	Sunny

Day	City	Temperature	Windspeed	Event
1/1/18	Taipei	20	1	Fog
2/1/18	Taipei	19	5	Fog
3/1/18	Taipei	17	3	Rain
4/1/18	Taipei	17	7	Rain

DATA AGGREGATION - GROUPBY

You get a group of **smaller dataframes**



Day	City	Temperature	Windspeed	Event
1/1/18	Singapore	32	1	Sunny
2/1/18	Singapore	31	3	Sunny
3/1/18	Singapore	31	2	Sunny
4/1/18	Singapore	32	5	Sunny
1/1/18	Seoul	21	12	Rain
2/1/18	Seoul	17	11	Sunny
3/1/18	Seoul	18	11	Rain
4/1/18	Seoul	20	6	Sunny
1/1/18	Taipei	20	1	Fog
2/1/18	Taipei	19	5	Fog
3/1/18	Taipei	17	3	Rain
4/1/18	Taipei	17	7	Rain

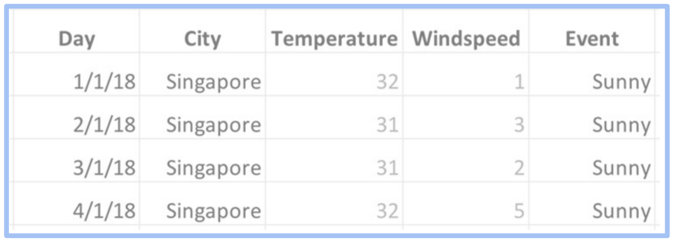
Day	City	Temperature	Windspeed	Event
1/1/18	Singapore	32	1	Sunny
2/1/18	Singapore	31	3	Sunny
3/1/18	Singapore	31	2	Sunny
4/1/18	Singapore	32	5	Sunny

Day	City	Temperature	Windspeed	Event
1/1/18	Seoul	21	12	Rain
2/1/18	Seoul	17	11	Sunny
3/1/18	Seoul	18	11	Rain
4/1/18	Seoul	20	6	Sunny

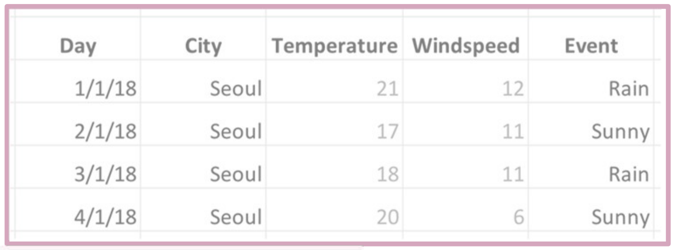
Day	City	Temperature	Windspeed	Event
1/1/18	Taipei	20	1	Fog
2/1/18	Taipei	19	5	Fog
3/1/18	Taipei	17	3	Rain
4/1/18	Taipei	17	7	Rain

DATA AGGREGATION - GROUPBY

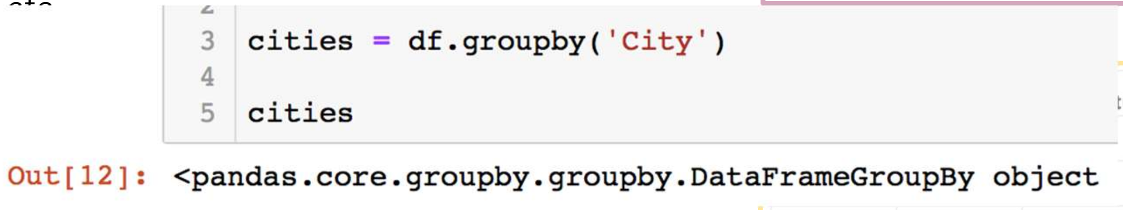
Pandas will group up the rows based on the City column's unique values, and return a GroupBy object. With this GroupBy object, you can perform several different aggregation operations, such as mean, stdev,



Day	City	Temperature	Windspeed	Event
1/1/18	Singapore	32	1	Sunny
2/1/18	Singapore	31	3	Sunny
3/1/18	Singapore	31	2	Sunny
4/1/18	Singapore	32	5	Sunny

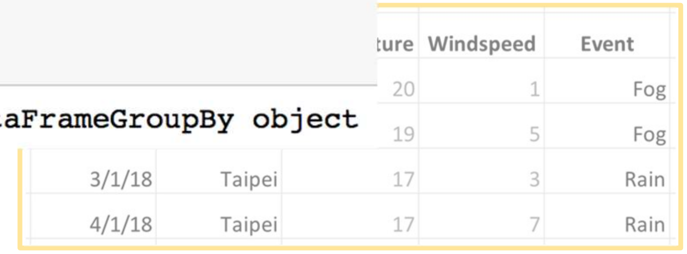


Day	City	Temperature	Windspeed	Event
1/1/18	Seoul	21	12	Rain
2/1/18	Seoul	17	11	Sunny
3/1/18	Seoul	18	11	Rain
4/1/18	Seoul	20	6	Sunny



```
cities = df.groupby('City')

cities
```

```
Out[12]: <pandas.core.groupby.groupby.DataFrameGroupBy object
```

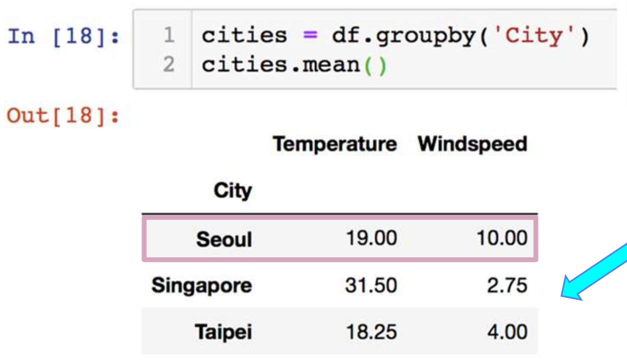


		ture	Windspeed	Event
		20	1	Fog
		19	5	Fog
3/1/18	Taipei	17	3	Rain
4/1/18	Taipei	17	7	Rain

DATA AGGREGATION - GROUPBY

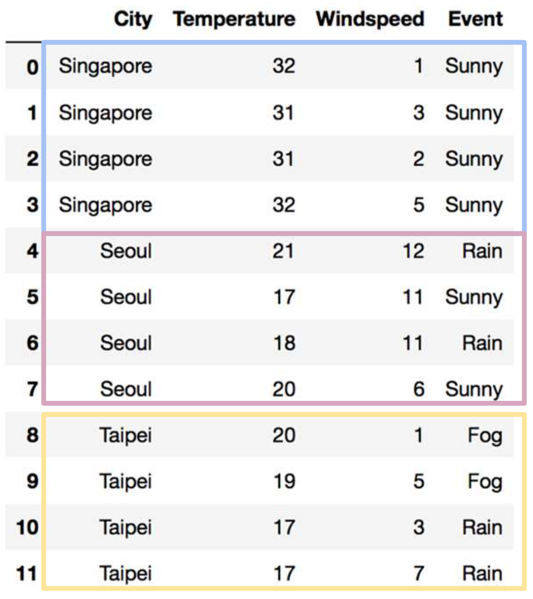
An example aggregation method here is .mean()

For all columns that are numeric, the mean of the columns' values will be computed, according to their groups.



```
In [18]:  cities = df.groupby('City')
          cities.mean()
Out[18]:
```

City	Temperature	Windspeed
Seoul	19.00	10.00
Singapore	31.50	2.75
Taipei	18.25	4.00



	City	Temperature	Windspeed	Event
0	Singapore	32	1	Sunny
1	Singapore	31	3	Sunny
2	Singapore	31	2	Sunny
3	Singapore	32	5	Sunny
4	Seoul	21	12	Rain
5	Seoul	17	11	Sunny
6	Seoul	18	11	Rain
7	Seoul	20	6	Sunny
8	Taipei	20	1	Fog
9	Taipei	19	5	Fog
10	Taipei	17	3	Rain
11	Taipei	17	7	Rain

DATA AGGREGATION - GROUPBY

Getting multiple aggregates with only one function call:

```
df['x_col'].agg({

      'col_name': ['sum', 'mean', ...]

})
```

Column to perform aggregation

Aggregations to perform

THANK YOU!

HACKWAGON

• ACADEMY •