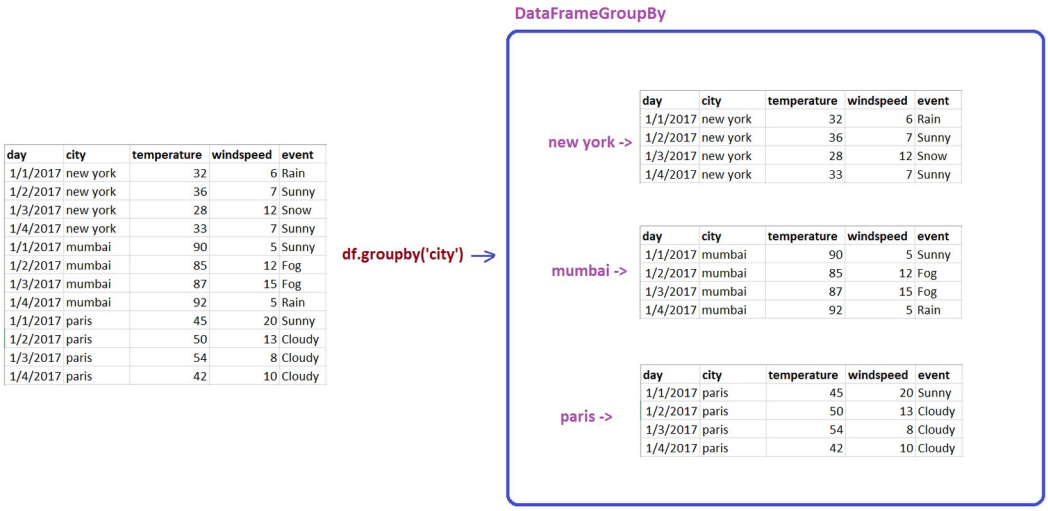
```
import pandas as pd
df = pd.read_csv("/weather_by_cities.csv")
df
```

	day	city	temperature	windspeed	event
0	1/1/2017	new york	32	6	Rain
1	1/2/2017	new york	36	7	Sunny
2	1/3/2017	new york	28	12	Snow
3	1/4/2017	new york	33	7	Sunny
4	1/1/2017	mumbai	90	5	Sunny
5	1/2/2017	mumbai	85	12	Fog
6	1/3/2017	mumbai	87	15	Fog
7	1/4/2017	mumbai	92	5	Rain
8	1/1/2017	paris	45	20	Sunny
9	1/2/2017	paris	50	13	Cloudy
10	1/3/2017	paris	54	8	Cloudy
11	1/4/2017	paris	42	10	Cloudy

```
g = df.groupby("city")
g
```

```
<pandas.core.groupby.generic.DataFrameGroupBy object at
0x7f5ca70cb550>
```




```
for city, data in g:
    print("city:",city)
    print("\n")
    print("data:",data)
```

```
city: mumbai


data:         day    city  temperature  windspeed  event
4  1/1/2017  mumbai           90          5  Sunny
```

```
5  1/2/2017  mumbai           85         12    Fog
6  1/3/2017  mumbai           87         15    Fog
7  1/4/2017  mumbai           92          5   Rain
city: new york


data:         day      city  temperature  windspeed  event
0  1/1/2017  new york           32          6   Rain
1  1/2/2017  new york           36          7  Sunny
2  1/3/2017  new york           28         12   Snow
3  1/4/2017  new york           33          7  Sunny
city: paris


data:          day   city  temperature  windspeed   event
8   1/1/2017  paris           45         20   Sunny
9   1/2/2017  paris           50         13  Cloudy
10  1/3/2017  paris           54          8  Cloudy
11  1/4/2017  paris           42         10  Cloudy
```

This is similar to SQL,

SELECT * from weather_data GROUP BY city

```
g.get_group('mumbai')
```

```
        day    city  temperature  windspeed  event
4  1/1/2017  mumbai           90          5  Sunny
5  1/2/2017  mumbai           85         12    Fog
6  1/3/2017  mumbai           87         15    Fog
7  1/4/2017  mumbai           92          5   Rain
```

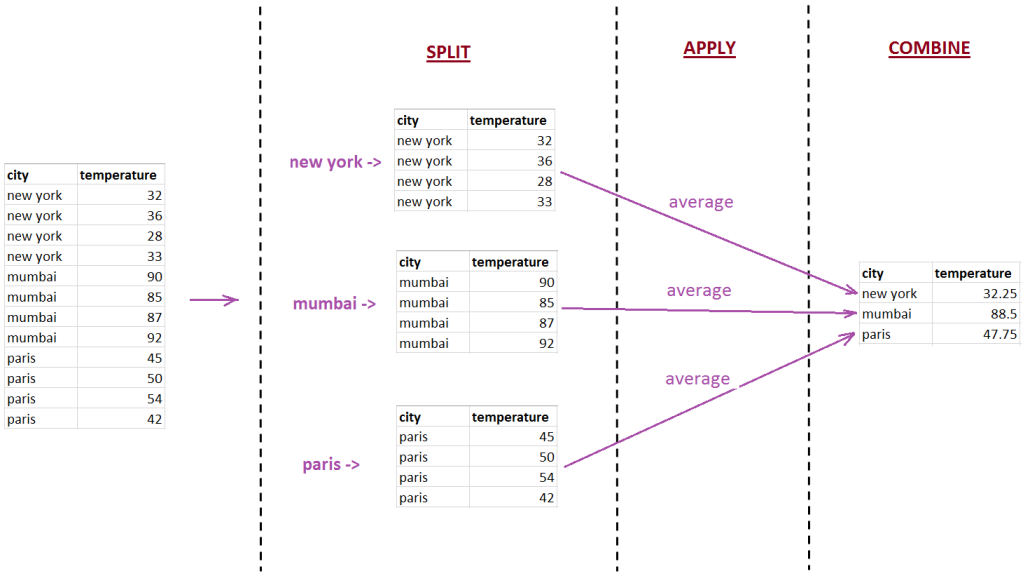
```
g.max()
```

```
               day  temperature  windspeed  event
city
mumbai    1/4/2017           92         15  Sunny
new york  1/4/2017           36         12  Sunny
paris     1/4/2017           54         20  Sunny
```

```
g.mean()
```

```
          temperature  windspeed
city
mumbai          88.50       9.25
new york        32.25       8.00
paris           47.75      12.75
```

This method of splitting your dataset in smaller groups and then applying an operation (such as min or max) to get aggregate result is called Split-Apply-Combine. It is illustrated in a diagram below



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	32.25
mumbai	88.5
paris	47.75

```
g.min()
```

```
              day  temperature  windspeed   event
city
mumbai    1/1/2017           85          5     Fog
new york  1/1/2017           28          6    Rain
paris     1/1/2017           42          8  Cloudy
```

```
g.describe()
```

```
        temperature                                ... windspeed

              count   mean       std   min   ...       25%   50%
75%   max
city                                           ...

mumbai          4.0  88.50  3.109126  85.0  ...      5.00   8.5
12.75  15.0
new york        4.0  32.25  3.304038  28.0  ...      6.75   7.0
8.25  12.0
paris           4.0  47.75  5.315073  42.0  ...      9.50  11.5
14.75  20.0

[3 rows x 16 columns]
```

```
g.size()
```

```
city
mumbai      4
new york    4
paris       4
dtype: int64
```

```
g.count()
```

city	day	temperature	windspeed	event
mumbai	4	4	4	4
new york	4	4	4	4
paris	4	4	4	4