Day 10: *Group By in Pandas*

After spending time tasting wine and viewing Renaissance art in Italy, Dot's journey in Europe came to a close. They weren't heading back home just yet, though – there was so much more of the world they wanted to explore! Dot hopped on a plane from Florence to voyage over to a luxurious, warm city sitting against the Persian Gulf: Dubai. They had organized the trip to arrive in the emirate just as night fell. They had some time to grab a falafel sandwich from a streetside food cart before making their way to a nightclub. Lining up before the entrance, Dot saw a diverse crowd of people from different cultures and countries, speaking to each other in various languages. The bouncer smiled at Dot and ushered them inside. Dot was momentarily struck frozen by the interior: a giant space with multiple levels, lit by flashing, differently coloured lights, filled with people dancing happily to the pounding music.

When Dot woke up the following day, they gazed out their hotel window at the hyper-modern cityscape. The Burj Khalifa, the world's tallest building, stood grandly among a crowd of different skyscrapers. Dot smiled to themselves and thought about how incredible Dubai was: the nightlife, the food, the luxury and cutting-edge architecture. "How I would love to live in this place!" they said aloud. They knew, though, that living in Dubai would be expensive. Just how much would it be to purchase property in the city, and what neighbourhoods would be too expensive for Dot to afford? Can you help Dot work through a dataset that lists property prices in Dubai?

Tutorial

Within Pandas, one of the most essential and useful functions for data analysis is the *group by* function.

Group by does one thing: it groups the dataset according to a categorical column or columns. However, the grouping function can't stand on its own. The user needs to apply a specific aggregate function to the dataset after using *group by*. Check the example below.

```
import pandas as pd

df = pd.read_csv('dubai_properties_data.csv', index_col = 0)

df.groupby(['quality']).mean()
```

In the above code, we grouped the dataset by the **quality** column, then used the ***mean()*** aggregate function to see the average of **ALL** numerical columns for each year. However, we don't have to use the **sum()** function with group by. We can easily use other aggregate functions, such as ***sum(), min(), max()...***.

Below is a list of aggregate functions we can use on our group bys.

- count() – Number of non-null observations
- sum() – Sum of values
- mean() – Mean of values
- median() – Arithmetic median of values

- min() – Minimum
- max() – Maximum
- mode() – Mode
- std() – Standard deviation
- var() – Variance
- size() - Number of rows

We can specify the columns we want to group by:

```
df.groupby(['quality'])[['price','size_in_sqft','no_of_bedrooms']].mean()
```

Try this line of code below and see what it does:

```
df.groupby(['view_of_landmark','view_of_water'])
[['price','no_of_bedrooms']].mean()
```

Before you continue to the challenge, play around with the *group by* function a bit. You can read more on the function in this article.

To learn more about the various pandas functions, check out the user guide in the pandas documentation.

In [1]:
```
import pandas as pd
df = pd.read_csv('dubai_properties_data.csv')
```

Challenge

Which neighborhood has the highest average property price and the highest size_in_sqft?

In [2]:
```
# SOLUTION

grouped = df.groupby('neighborhood')[['price','size_in_sqft']].mean()
grouped.sort_values('price',ascending=False).head(1)
```

Out[2]:

neighborhood	price	size_in_sqft
Palm Jumeirah	4.379435e+06	2084.134831

In [3]:
```
grouped.sort_values('size_in_sqft',ascending=False).head(1)
```

Out[3]:

neighborhood	price	size_in_sqft
Dubai Festival City	2445000.0	2778.4