Introduction / Business Problem

My partner is a jewellery designer/maker, with a successful shop in one of the trendier parts of London. She has in the past expressed an interest in opening a second shop, and her usual methodology for this might be to go do some market research. Rather than simply ask the coolest people we know, I'd like to use data science to identify an appropriate location.

The basic question then is this: can we work out which are the hippest areas in London, and the areas that are on the verge of becoming hip? And given that where would be a good place to open a cool jewelry shop?

This approach could more broadly apply to anyone interested in opening an independent store or brand in London, and if the approach is successful might be more broadly applied to any city for which we have the appropriate data.

Data

We'll break this down into 2 components: quantifying trendiness into clusters, and finding competitive advantage.

To quantify trendiness we'll explore different options, among them identifying whether there are a large number of chain (non-independent) stores in the area, or stores with particular or unusual categories. We'll look at the frequency and density of specific category types such as record stores and coffee shops.

To finding competitive advantage, we'll look at details of the specific jewelry shops, including rating and price. Consideration should be given to the number of total shops and restaurants in the area (which would generate more footfall and be a positive influence), as well as other jewellery shops (which would mean more competition and be a negative influence).

The data we'll need includes:

- **Geocoding** - to identify the specific locations
- **Foursquare API** - to identify the qualifying places (jewelry stores, restaurants etc.) and essential details around them

Geocoding

Geonames.org (https://www.geonames.org/) is a fantastic free resource that publishes geocoded names and their latitudes/longitudes many countries around the world. The GeoNames geographical database is available for download free of charge under a creative commons attribution license. The resource includes an explicit accuracy for the latitude/longitude, which will be useful for filtering out bad/inaccurate/duplicate data.

Below is a sample of the resulting data. Note the availabel Latitude/Longitude, and the Accuracy record.

Out[2]:

	Countrycode	Postalcode	Placename	Adminname1	Admincode1	Adminname2	Admincod
0	GB	DN14	Goole	England	ENG	East Riding of Yorkshire	116090
1	GB	DN14	Pollington	England	ENG	East Riding of Yorkshire	116090
2	GB	DN14	Faxfleet	England	ENG	East Riding of Yorkshire	116090
3	GB	DN14	Laxton	England	ENG	East Riding of Yorkshire	116090
4	GB	DN14	Old Goole	England	ENG	East Riding of Yorkshire	116090

According to the documentation, 'Accuracy' is defined as below:

```
Accuracy is an integer, the higher the better :
1 : estimated as average from numerically neigbouring postal codes
3 : same postal code, other name
4 : place name from geonames db
6 : postal code area centroid
```

Consequently we'll want to limit ourselves to codes with an Accuracy of greater or equal to 3 (averages of neighboring postal codes isn't useful). And because we're going to be using the Latitude/Longitude, we'll remove duplicate records of that and use only the first name. Postalcode will be tricky to use as it bridges Placenames, but we'll want to keep it around for reference and collapse it into a single column joined by a comma.

Finally, we'll be limiting ourselves to Greater London.

Below is a resulting sample of the data we'll use:

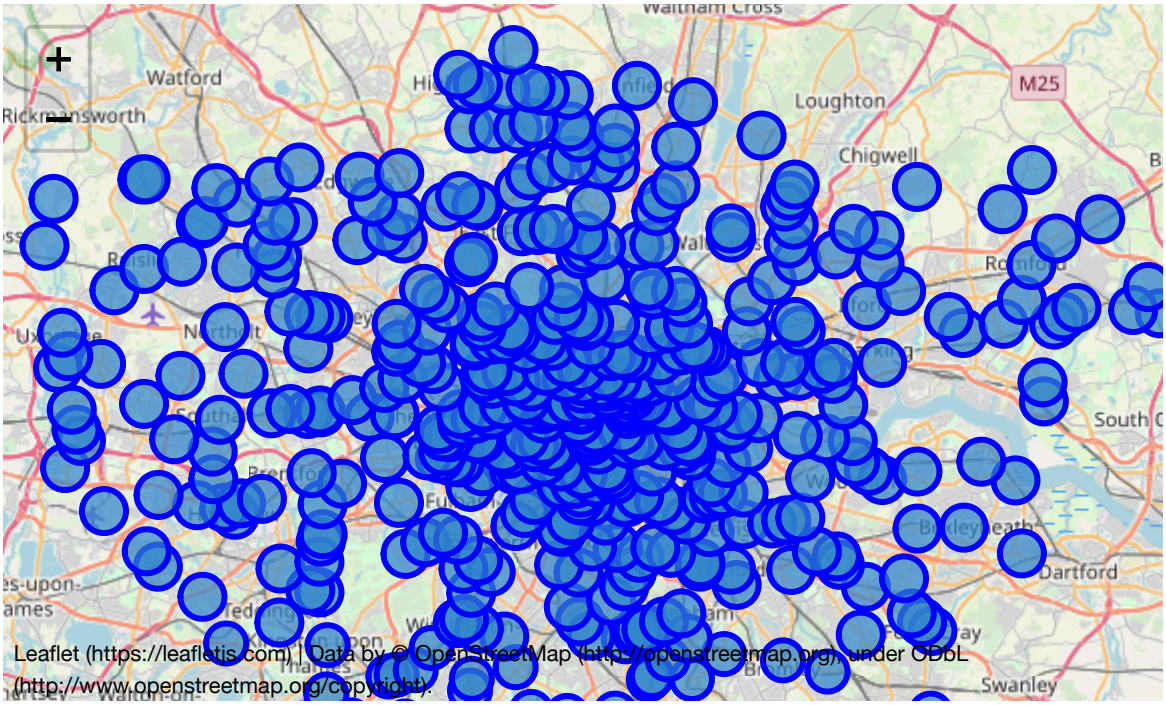
Out[3]:

	Latitude	Longitude	Placename	Postalcode
0	51.3133	0.0343	Biggin Hill	TN16
1	51.3148	-0.1570	Hooley	CR5
2	51.3192	0.0712	Cudham	TN14
3	51.3200	-0.1409	Coulsdon	CR5
4	51.3264	-0.1011	Kenley	CR8

```
There are 468 Places we will evaluate in London
```

A review of a map of London with the locations marked at a radius of 1000m exposes the amount of overlapping areas and the high coverage for central London. This should be sufficient for our evaluation.

Out[5]:



Foursquare API

We will retrieve both the places and necessary details from the Foursquare API. We'll get the precise location and category of the shop, as well as details such as price and rating where necessary. Some of these calls will be necessarily premium calls, so to keep ourselves within the boundaries of free usage we'll limit them.

When using the search and explore endpoints the Foursquare API will naturally constrain the responses, presumably to keep people from scraping/farming locations. We'll need to keep this in mind, as it means that our results will not be "complete" for a given radius.

We'll begin by retrieving the full list of categories from Foursquare and identifying which of them corresponds to jewelry (note in the UK this is spelled "jewellery").

Out[7]:

	id	name
742	4bf58dd8d48988d111951735	Jewelry Store

We want to focus on a specific types of venues, and the Foursquare API allows us to break things down into category "sections": shops, food, drinks, coffee, arts. We'll make separate calls for each of these venue sections and collect the relavant details. Although this may not get us everything, narrowing the search on sections will get us significantly more options.

Then we'll grab all the Jewelry Shop category venues in London exclusively. Below is a sample of these stores, and a heatmap of all the stores across London.

Out[16]:

	Placename	Latitude	Longitude	Venue ID	Venue	Venue Latitude	Ve Longit
669	Warren Street	51.5136	-0.1498	4ac518eff964a52063ad20e3	Grays Antiques	51.513622	-0.148
573	Mayfair	51.5095	-0.1490	4ac518eff964a52066ad20e3	Montblanc Boutique	51.508787	-0.140
821	Soho	51.5144	-0.1354	4ac518eff964a5208aad20e3	Storm	51.513183	-0.138
516	Mayfair	51.5095	-0.1490	4ac518f0f964a520aaad20e3	Tiffany & Co.	51.509579	-0.141
973	Oxford Circus	51.5154	-0.1414	4ac518f0f964a520c9ad20e3	H. Samuel	51.515108	-0.143

Out[17]:

Now we have some data, we can do some exploration.