# Data for Australian city grouping

To help build a model to compare Australian cities with each other, we propose to follow the process below and gather the data required for the project.

1. Finalize the list of cities to be considered for analysis. This is needed because it is not practically possible to cover every inch of Australia and analyze it for similarity. Plus, there isn't data available for every small city/town or settlement. For the purpose of this project, we intend to get the list of cities available on Wikipedia https://en.wikipedia.org/wiki/List_of_cities_in_Australia_by_population This data also gives us the recent population of each city.

Here is quick look at the data

```
In [2]: import pandas as pd
        from geopy.geocoders import Nominatim
        import requests

        #Get major cities in Austraila from web
        url = 'https://en.wikipedia.org/wiki/List_of_cities_in_Australia_by_population'

In [3]: au_cities = pd.read_html(url)

In [14]: df_city_raw = pd.DataFrame(au_cities[1])
         df_city_raw.reset_index(drop =True, inplace = True)
         df_city_raw.columns=['Rank','Significant Urban Areas','State','Population2019','Population2011','Growth','percent_of
         df_city_raw.drop(['Rank','Growth','percent_of_national_population','Population2011'],axis=1, inplace=True)
         df_city_raw.head()
```

Out[14]:

| | Significant Urban Areas | State | Population2019 |
|---|---|---|---|
| 0 | Sydney | New South Wales | 4914343 |
| 1 | Melbourne | Victoria | 4893870 |
| 2 | Brisbane | Queensland | 2430180 |
| 3 | Perth | Western Australia | 2045479 |
| 4 | Adelaide | South Australia | 1340794 |

2. Get the latitude and longitude for each city chosen for analysis in step 1 using **geopy** API

Take a quick look at latitude and longitude for each City

```
In [5]: df = pd.DataFrame(columns=['City','Latitude','Longitude'])
        list_city = []
        list_lat  = []
        list_lng  = []

        for city in df_city_raw["Significant Urban Areas"]:
            address = '{}, {}'.format(city, 'AU')
            geolocator = Nominatim(user_agent='Australia_explorer')
            location = geolocator.geocode(address)
            if(location is not None):
                lat = location.latitude
                lng = location.longitude
                list_city.append(city)
                list_lat.append(lat)
                list_lng.append(lng)


        df['City'] = list_city
        df['Latitude'] = list_lat
        df['Longitude'] = list_lng
        df
```

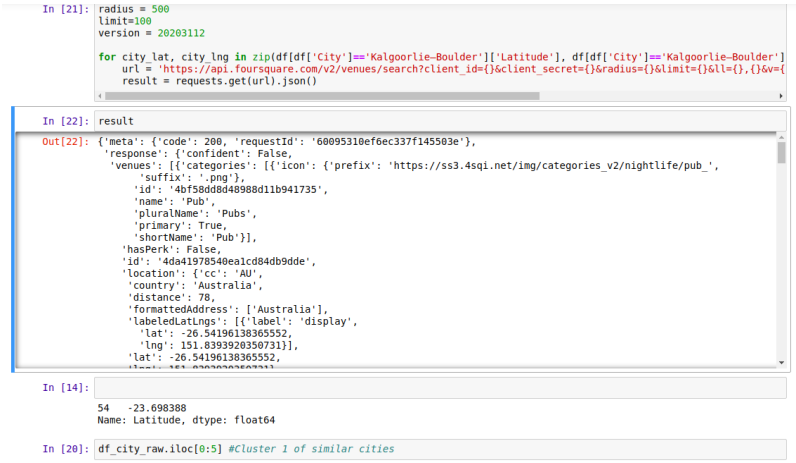
Out[5]:

| | City | Latitude | Longitude |
|---|---|---|---|
| 0 | Sydney | -33.854816 | 151.216454 |
| 1 | Melbourne | -37.814218 | 144.963161 |
| 2 | Brisbane | -27.468968 | 153.023499 |
| 3 | Perth | -31.952712 | 115.860480 |
| 4 | Adelaide | -34.928181 | 138.599931 |
| ... | ... | ... | ... |
| 94 | Esperance | -33.440355 | 122.405438 |
| 95 | Parkes | -32.822541 | 147.976136 |
| 96 | Swan Hill | -35.339328 | 143.559203 |

3. Use Foursquare API to get the venues for each of the cities chosen in step1 using its latitude and longitude found in step 2

(Point 3 continued...)

A quick look at it



```
In [21]:
radius = 500
limit=100
version = 20203112

for city_lat, city_lng in zip(df[df['City']=='Kalgoorlie-Boulder']['Latitude'], df[df['City']=='Kalgoorlie-Boulder']
    url = 'https://api.foursquare.com/v2/venues/search?client_id={}&client_secret={}&radius={}&limit={}&ll={},{}&v={
    result = requests.get(url).json()

In [22]:
result

Out[22]:
{'meta': {'code': 200, 'requestId': '60095310ef6ec337f145503e'},
 'response': {'confident': False,
  'venues': [{'categories': [{'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/nightlife/pub_',
      'suffix': '.png'},
     'id': '4bf58dd8d48988d11b941735',
     'name': 'Pub',
     'pluralName': 'Pubs',
     'primary': True,
     'shortName': 'Pub'}],
    'hasPerk': False,
    'id': '4da41978540ea1cd84db9dde',
    'location': {'cc': 'AU',
     'country': 'Australia',
     'distance': 78,
     'formattedAddress': ['Australia'],
     'labeledLatLngs': [{'label': 'display',
       'lat': -26.54196138365552,
       'lng': 151.8393920350731}],
     'lat': -26.54196138365552,

In [14]:

54   -23.698388
Name: Latitude, dtype: float64

In [20]:
df_city_raw.iloc[0:5] #Cluster 1 of similar cities
```

4. Foursquare API venues give us very good data about what kind of places a city has. It is called as venues.

   Here is sneak peak in to one venue detail

   *{'categories': [{'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/nightlife/pub_',*

   *'suffix': '.png'},*

   *'id': '4bf58dd8d48988d11b941735',*

   *'name': 'Pub',*

   *'pluralName': 'Pubs',*

   *'primary': True,*

   *'shortName': 'Pub'}],*

   *'hasPerk': False,*

   *'id': '4da41978540ea1cd84db9dde',*

   *'location': {'cc': 'AU',*

   *'country': 'Australia',*

   *'distance': 78,*

   *'formattedAddress': ['Australia'],*

   *'labeledLatLngs': [{'label': 'display',*

   *'lat': -26.54196138365552,*

   *'lng': 151.8393920350731}],*

(Point 4 Continued...)

*'lat': -26.54196138365552,*

*'lng': 151.8393920350731},*

*'name': 'Kingaroy RSL',*

*'referralId': 'v-1611223824'}*

The venue categories and population are used as a measure of similarity and dissimilarity. The rationale behind this is a traveler to the city is most interested in the places to visit. These places are given by Foursquare API and are called **venues.** The population of the city plays a major role in deciding many important aspects of a city, including public transport, amenities, crime, poverty, etc. Thus, city population is one important feature to decide similarity and dissimilarity among cities.