6.3 Lesson Summary - Bank Deserts

A significant amount of data analytics examines patterns in location-based data. There are a number of questions that can be addressed by analyzing and visualizing data as it relates to the world around us. What is the average income of different neighborhoods, how does the air pollution of a region change over the course of a year, what precipitation can I expect in my city tomorrow are all examples of questions that can be addressed through location-based analysis. For this task we need tools to properly describe the location information in our data and then convert this data into a meaningful visualizations.

Concept: Google offers a suite of map-based APIs that can assist in location-based analysis and visualizations. It's often easiest to have your location described in longitude and latitude coordinates. **Google Maps Geocoding** API will convert an address into longitude and latitude coordinates. **Google Maps Places** API will give you longitude and latitude coordinates for points of interest including popular destinations, landmarks, businesses and more. To get geocoding data for the city of Raleigh, North Carolina you could use the following code:

*import requests*

*geo\_data = requests.get(https://maps.googleapis.com/maps/api/geocode/json?address=Raleigh&key={your\_api\_key}).json()*

To get location data for every Thai restaurant in a certain region you could use Google Maps places API with the following code:

*target\_coordinates = "43.6187102, -116.2146068"*

*target\_search = "Chinese"*

*target\_radius = 8000*

*target\_type = "restaurant"*

*params = {*

*"location": target\_coordinates,*

*"keyword": target\_search,*

*"radius": target\_radius,*

*"type": target\_type,*

*"key": gkey*

*}*

*base\_url = "https://maps.googleapis.com/maps/api/place/nearbysearch/json"*

*response = requests.get(base\_url, params=params)*

* Activity: 01-Ins\_Google\_Geocode, 02-Ins\_Google\_Places, 03-Stu\_Google\_Drills
* Suppl link: <https://cloud.google.com/maps-platform/>, <https://cloud.google.com/maps-platform/places/>

Concept: Pandas DataFrame object plays an important role in analyzing data. It is often useful to be able to iterate through each row of a DataFrame. You can accomplish this using the following code:

*for index, row in df.iterrows():*

*print (index, row)*

* Activity: 04-Ins\_NearestRestr, 05-Stu\_Google\_Complex

Concept: Jupyter allows you to add functionality through the use of plug-ins. If you wish to display high-quality Google based maps you can install the *gmaps* plug-in. Run the following code to be able to map data directly in your Jupyter session:

*jupyter nbextension enable --py --sys-prefix widgetsnbextension*

*conda install -c conda-forge gmaps*

*jupyter nbextension enable --py --sys-prefix gmaps*

Concept: You can add layers to your map using the **marker\_layer** method. For example:

*fig = gmaps.figure()*

*markers = gmaps.marker\_layer(marker\_locations)*

*fig.add\_layer(markers)*

* Activity: 06-Evr\_Jupyter\_Gmaps

Concept: **Heat Maps** can be used to illustrate the intensity or magnitude of a data field across a geographic region. Adding a heat map layer to a gmap is similar to adding a marker layer. For example:

*fig = gmaps.figure()*

*heat\_layer = gmaps.heatmap\_layer(locations, weights=rating,*

*dissipating=False, max\_intensity=10,*

*point\_radius=1)*

*fig.add\_layer(heat\_layer)*

* Activity: 07-Stu\_Airport\_Map, 10-Stu\_BankDeserts\_Heatmap