***Scenario****: You are an analyst for the Citizen’s Policing and Crime Brigade (CPCB), a community organization that follows crime trends in Nashville. You have been asked to explore the Nashville police calls data, look at differences from 2013 to 2018,* ***and prepare a brief (5-7 minutes) presentation on your findings to deliver to the city council****. This guide will walk you through the initial data exploration and the steps needed to answer these specific questions:*

* *Which patrol units were most active in these years?*
* *Which crimes were most prevalent?*
* *How have burglaries, robberies, and other thefts change from 2013 to 2018?*
* *What does the pattern of calls and call disposition look like for these?*
* *Does the theory that robberies occur near major roads hold true for these years?*
* *Does the theory that more burglaries occur on rainy days hold true for these years?*

**Week One**

* *Unzip the shared file to create a directory for your Analytics Jumpstart work. Call it ‘****Analytics Jumpstart’****.*
* *Inside that directory, be sure there is a directory called ‘****data****’.*
* *Verify the* ***weather.db*** *file is in the data folder.*
* *Over the next 3 weeks, you will be exploring the calls for police service in Metro Nashville Davidson County. Download the CSV files for 2013 and 2018 police calls from* [*https://data.nashville.gov*](https://data.nashville.gov) *and place them in the data folder.*
* *Launch Jupyter notebook from Anaconda Navigator. In your Jupyter server browser window, navigate to your Analytics Jumpstart directory and create a new Python 3 notebook. Call your notebook ‘****police\_calls’****.*

1. Import the packages below. If you decide to import additional packages over the next few weeks, be sure to add them to this cell. Also add the magic command to show your plots without having to call **plt.show()** to the bottom of this cell.

**import pandas as pd**

**import matplotlib.pyplot as plt**

**import seaborn as sns**

**import numpy as np**

**import sqlite3 as sql**

**%matplotlib inline**

1. Read in police call data for 2013 to a data frame called **police\_2013** and look at the first 5 rows. Look at the last five rows of **police\_2013.**
2. How many rows and columns does **police\_2013 contain?**
3. Keep just these 13 columns:
   1. ‘Event Number’
   2. ‘Call Received’
   3. ‘Complaint Number’
   4. ‘Shift’
   5. ‘Tencode’
   6. ‘Tencode Description’
   7. ‘Disposition Code’
   8. ‘Disposition Description’
   9. ‘Unit Dispatched’
   10. ‘Sector’
   11. ‘Zone’
   12. ‘Latitude’
   13. ‘Longitude’
4. Rename the columns above:
   1. 'id'
   2. 'call\_time'
   3. 'complaint\_id'
   4. 'shift'
   5. 'tencode'
   6. 'tencode\_desc'
   7. ‘disposition’
   8. ‘disposition\_desc’
   9. ‘unit’
   10. ‘sector’
   11. ‘zone’
   12. ‘lat’
   13. ‘lng’
5. Which unit answered the most calls in 2013? How many calls did they answer?
6. How many units answered ***more than*** ***1000*** calls in 2013?
7. What tencode description is associated with the greatest number of calls in 2013?
8. Create a data frame from the tencode description value counts. It should have two columns called ‘crime’ and ‘count’.
9. Make a horizontal barplot of the crime counts.

**Week Two**

1. Repeat steps 1 through 9 for the 2018 data.
2. Since the 2018 data is incomplete, find **the latest date** for this dataset. So that comparisons between 2013 and 2018 are commensurate, remove rows from the 2013 data that are later than the latest month and day in the 2018 data.
3. Create subsets of the 2013 and 2018 police call data thatincludes these five tencodes:

* *Holdup / Robbery*
* *Bank Robbery*
* *Burglary – Residence*
* *Burglary - Non-Residence*
* *Theft*

1. Add a column named ‘year’ to each of these new subset data frames. Set year equal to 2013 in the 2013 subset and set year equal to 2018 in the 2018 subset.
2. Combine the 2013 and 2018 subset data into a single data frame called **crimes**. Since they have the same columns, you can concatenate the data frames. How many rows and columns are there in the combined data frame?
3. Check the value counts for the tencode description in the crimes data. Using the pandas map function, change the descriptions so there are just three categories: **Burglary** (includes ‘Burglary - Residence' and 'Burglary - Non-Residence'), **Robbery** (includes 'Bank Robbery' and 'Holdup / Robbery'), and **Theft** (includes ‘Theft’). Check the value counts again to confirm your changes.
4. Create a data frame called **aggregated\_df** with the year, crime category, and count of calls for the year and category.
5. Plot a Seaborn facet grid to compare the 2013 and 2018 count of crimes for each category. Has any category changed significantly from 2013 to 2018?
6. Which sector had the least number of calls for 2013? How many calls? What about 2018?
7. What percentage of calls for 2013 had a disposition of “disregard”? Did this increase or decrease for 2018?

**Week Three**

1. Install the folium package. You can do this from a terminal or Anaconda prompt or by typing in your notebook (notice the exclamation point): 
2. Add ‘**import folium**’ to the top cell where your package installs are.
3. Create a folium map of Nashville using [36.1612, -86.7775] as the location to center the map on. Experiment with different values for the **zoom\_start** argument.
4. You have been asked whether more robberies occur near highways. One theory is that robbers rob near highways so that they can make a speedy getaway. As an initial test of this assumption, map the locations of robberies in your crimes data that have a disposition description of ‘GONE ON ARRIVAL’. First create a subset of the data that contains only the rows of interest. Then create a folium map of locations by building a for-loop and using the iterrows() function to create point locations, and map each point as you loop through your dataframe.
5. Another theory says that more burglaries occurs on rainy days. Weather data is available in a SQLite database file called **weather.db**.
   1. First execute the following code to see what tables are in the database:

**db = <path to weather.db>**

**con = sql.connect(db)**

**mycursor = con.cursor()**

**mycursor.execute("SELECT name FROM sqlite\_master WHERE type='table' ORDER BY name;")**

**tables=(mycursor.fetchall())**

**print(tables)**

* 1. Load the weather data for 2013 where name is like ‘NASHVILLE’ to a pandas data frame.
  2. How many rows and columns are in this data frame?
  3. From how many different stations was the data collected?

1. Plot the unique weather stations on a map to see where they are located around Nashville.
2. Create a new dataframe with the average rainfall for each day in September 2013.

1. Plot the precipitation for September 2013 with the burglaries for September 2013 to visually assess the relationship between precipitation and burglaries. Share your impressions.