Nashville Software School
Data Analytics Jumpstart

Scenario: Metro Council has engaged your analytics consulting company to help the council understand if citizens are less satisfied in areas where adverse events are more likely to occur.

You have been asked to explore both quantitative and qualitative datasets:

1. **police_calls_2018.csv:** a Nashville police department calls for service, dataset from 2018 that details where calls were made from (emergency and non-emergency calls)
2. **hubNashville_2018.csv:** hubNashville 311 service requests, data from 2018 that shows all requests for service made to hubNashville
3. **metro_survey.db:** a database containing the results of a 2018 resident satisfaction survey

Metro Council has asked your company to **prepare a brief (5-7 minutes) presentation of your findings to deliver to the city council and representatives from the mayor's office**. This guide will walk you through the initial data exploration and the steps needed to answer these specific questions:

- What kinds of police calls occurred most often in 2018?
- Where did these calls originate (which zipcodes)?
- In which months do calls occur most frequently?
- What kinds of requests for service are made to hubNashville?
- Are they being handled promptly?
- Are there particular kinds of requests in particular areas (zipcodes) that are especially problematic?
- How do the results of the 2018 survey align with what you observe in the police calls and hubNashville data? Are there any actionable insights from comparing the data? Are there any surprises?
- Are there any other findings you would like to share?

Week One Tasks

- Unzip the shared file to create a directory for your Analytics Jumpstart work. It should be called '**Analytics Jumpstart'**.
- Inside that directory, be sure there is a folder called '**data**'.

- 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 ‘**jumpstart_analysis’**.

1. Import the packages by running the code below in a Jupyter notebook cell. If you decide to import additional packages over the next few weeks, be sure to add them to this cell. Also add the command that starts with **%** to the bottom of this cell. This is called a *magic* command and displays your plots in the notebook without having to also call a separate command.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import sqlite3 as sql

%matplotlib inline
```

2. Read in the 2018 police calls data (police_calls_2018.csv to a dataframe called **police_calls**.
   a. Look at the first 5 rows.
   b. Look at the last 3 rows.
   c. How many rows and columns does **police_calls** contain?
3. Keep just these columns:
   a. 'Call Received'
   b. 'Shift'
   c. 'Tencode'
   d. 'Tencode Description'
   e. 'Disposition Code'
   f. 'Disposition Description'
   g. 'Unit Dispatched'
   h. 'Sector'
   i. 'Zone'
   j. 'Latitude'
   k. 'Longitude'
   l. 'zipcode'
   m. 'PO'
4. Rename the columns above:
   a. ‘call_time'
   b. 'shift'
   c. 'tencode'

d. 'tencode_desc'
e. 'disposition'
f. 'disposition_desc'
g. 'unit_dispatched'
h. 'sector'
i. 'zone'
j. 'lat'
k. 'lng'
l. 'zipcode'
m. 'po'

5. What are the unique disposition descriptions?
6. Remove all rows from police_calls where the disposition description is missing (*nan*) or one of these two: DISREGARD / SIGNAL 9, NO RESPONSE. Check to see that you have 624841 rows remaining.
7. Create a dataframe from the **tencode_desc** value counts called **tencode_counts**. It should have two columns called **tencode** and **tencode_count**.
8. Create a seaborn horizontal barplot to show the 2018 calls for police service by tencode. Adjust the figsize so that you can see all of the data.
9. Find the counts of calls by **zipcode** and plot that. Zip codes look like numeric data, but should usually be treated as categorical. Convert the **zipcode** column to a string before plotting to avoid having big gaps where there are numbers but no zip codes. Give the plot a meaningful title.
10. The Metro Council is interested in the effect of community policing activities. For which zip codes do calls for “Community Policing Activity” most frequently occur? How do these zip codes compare to what you see when looking at the overall counts by zip code?
11. Convert the **call_time** column in **police_calls** to a pandas datetime. You’ll likely want to specify the format argument in order to speed up execution. Create a new column in **police_calls** to show the month that a call for service occurred. In which month(s) did most calls occur? What do you notice about the months for which data is provided?
12. Which days of the week tend to get the most calls? Which tend to get the least?

Week Two Tasks

13. Take a look at the 2018 hubNashville data by reading it into a DataFrame called **hub**.

14. Clean the **hub** column names to make everything lowercase and eliminate spaces so you can use dot-notation. Make the new names **'request_id'**, **'status'**, **'request_type'**, **'subrequest_type'**, **'add_subreqest_type'**, **'opened'**, **'closed'**, **'origin'**, **'zipcode'**, **'lat'**, **'lng'.**

15. Drop the rows from **hub** where **closed** is missing. You should end up with 80,866 rows Then create a new column, **resolution_time** that calculates how long the request was open. You'll need to convert **opened** and **closed** to pandas datetimes before calculating the time delta.

16. Were any requests open for longer than a year? How many? What request type was most commonly open for more than a year? Save the requests that were open for longer than a year to a DataFrame named **slow_to_resolve**.

17. Create a new **resolution_time_hours** column by dividing the **resolution_time** column by pd.Timedelta(hours = 1). The code to do this is

```
hub['resolution_time_hours'] = hub['resolution_time'] / pd.Timedelta(hours = 1)
```

18. Look at the distribution of resolution times. What do you notice?

19. Calculate the median resolution time (in hours) by zipcode for requests of type "Streets, Roads & Sidewalks". We are using median time since the distribution of resolution times is highly skewed. Save the results as a dataframe called **streets_median** with column names **zipcode** and **median_resolution_time**.

20. Create a connection to the survey data **(metro_survey.db**) and then create a cursor in order to find all the available tables in the database. They should match the tables shown on the **metro_survey_ERD** diagram.

21. In this question, we're going to look at the relationship between community policing calls and citizen's satisfaction with the police overall.
    a. The **safety** table has survey results that pertain to fire and police service, and the **info** table has zip code and other information for survey respondents. Write a SQL SELECT statement to join the two tables on **Id** and load them to a single pandas DataFrame (**safety_exp**).
    b. Slice **safety_exp** to get the **ZIP Code**, and '**Police - Overall**', columns. It's fine to save it back to the **safety_exp** variable.

c. Create a new DataFrame, **safety_total** which contains the total number of responses per zip code. Name the columns of this DataFrame **zipcode** and **total_responses.**
d. Filter **safety_exp** down to rows where the "**Police - Overall**" column is equal to "Dissatisfied" or "Very Dissatisfied". Then count the number of rows per zipcode. Save this to a DataFrame name **safety_dissatisfied** with columns named **zipcode** and **total_dissatisfied**.
e. Create a new dataframe, **safety_by_zip** by merging **safety_dissatisfied** and **safety_total**. Be sure to keep all zip codes in the result. You may want to fill in missing values with 0.
f. Finally, create a **pct_dissatisfied** column by dividing the **total_dissatisfied** by **total_responses** and multiplying the result by 100.
g. How do the zip codes where people are dissatisfied with the police compare to those with a large amount of community policing activities?

22. Follow similar steps as the previous question but using the "**Streets and Sidewalks - Overall**" column from the **general_services** table. Does there seem to be any relationship between median response time by zip code and percentage of survey respondents who are either Dissatisfied or Very Dissatisfied? You will want to used the **streets_median** DataFrame that you created in question 19.

Week Three Tasks

23. Install the folium package (if you haven't already done so). Type **!pip install folium** in a cell by itself (notice the exclamation point).

24. Delete the cell where you installed folium and add '**import folium**' to the top cell where your packages are imported.

25. Construct 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.

26. Assign the folium map you created in step 25 to a variable **nash_map**. Write a for loop that makes use of the iterrows() method to:
    a. Create a location for every hubNashville request in the **slow_to_resolve** DataFrame.
    b. Create a popup that gives information about the request_type and resolution_time for each request.
    c. Create a marker using the folium Marker() constructor.

d. Add the marker to **nash_map.**

After you exit the for loop, you can simply call nash_map to display your map with the markers you created.

You may want to construct an icon using one of the Font Awesome icons (https://fontawesome.com/v4.7.0/icons/). You can pass that icon to the **icon** argument in the **folium.Marker()** function.

The syntax for creating an icon is:

icon = folium.Icon(color = <pick a color>, icon = <pick and icon>, prefix = ‘fa’)

Team Exploration Section

You will be assigned to a team. Prepare for week 3 by brainstorming the things your team wants to explore, analyze, and present to the Metro Council. Choose a name for your consulting firm and create a Slack channel to collaborate over.

After finishing the analysis guide, spend the rest of the week working with your team to decide where to focus your efforts. Then complete your analysis, and create your presentation. There are lots of areas you could dive into in the hubNashville data and the survey!