Nashville Software School
Data Analytics Jumpstart

Scenario: Metro Council has engaged your analytics consulting company to help the council understand some potential factors influencing citizen satisfaction across Davidson County.

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. **population_2018.csv:** contains population values for each zip code in Davidson County
4. **metro_survey.db:** a database containing the results of a 2018 resident satisfaction survey

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 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.

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

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 in total?
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. The unit_dispatched column gives the callsign of the officer responding to the call. How many different callsigns are there in this data?
6. The shift column indicates the shift in which the call was received, with A being day shift, B being evening shift, and C being night shift. Which shift received the highest volume of calls?
7. What are the unique disposition descriptions?
8. In this question, you’ll remove some unneeded rows.
    a. Remove all rows where the disposition description is missing (nan).
    b. Remove any rows where the disposition description is DISREGARD / SIGNAL 9.
    c. Finally, remove any rows where the disposition is NO RESPONSE.
    d. Check to see that you have 624,841 rows remaining.
9. Create a dataframe from the **tencode_desc** value counts called **tencode_counts**. It should have two columns called **tencode** and **tencode_count**.
10. 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.
11. Find the counts of calls by **zipcode** and save it to a new dataframe called **calls_per_zipcode**. Plot this dataframe. 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.

12. The Metro Council is interested in the effect of community policing activities. Create a DataFrame called **community_policing** which contains the count of community policing calls per zip code. 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?
13. 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?
14. Which days of the week tend to get the most calls? Which tend to get the least?

Week Two Tasks

Continue using the same notebook that you used for the week one tasks for the following.

15. Take a look at the 2018 hubNashville data by reading it into a DataFrame called **hub**.
16. 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'.**
17. Drop the rows from **hub** where **closed** is missing. You should end up with 80,866 rows. Then convert **opened** and **closed** to pandas datetimes. Note that the opened and closed columns have different formats. Finally, create a new column, **resolution_time** that calculates how long the request was open.
18. 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**.
19. 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)
```

20. Look at the distribution of resolution times. What do you notice?
21. 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**.
22. Read the population_2018.csv file into a dataframe named **population**. Merge this with the calls_per_zipcode dataframe you created in question 11 and assign it back to the calls_per_zipcode dataframe. Hint: you may need to adjust the data type in one of the dataframes in order for the merge to work. Make sure to use the **how** argument in order to keep all of the zipcodes from the counts calls_per_zipcode dataframe.

23. Create a new column, **calls_per_capita** in the calls_per_zip dataframe by dividing the number of calls by the population of each zipcode. Which zipcodes have the highest number of calls per capita? Be aware when exploring the population data that some zip codes are only partially in Davidson County. The **ratio_in_davidson_county** column indicates how much of the zip code is in Davidson County.
24. 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.
25. 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**). Slice **safety_exp** to get the **ZIP Code**, and '**Police - Overall**', columns. It's fine to save it back to the **safety_exp** variable.
26. Zip code 37203 has a large residential population, and a large number of police calls per capita. Create a bar chart showing the distribution of each response type in the Police - Overall column for this zipcode. How does it compare to the distribution of these values overall?

Week Three Tasks

27. Follow similar steps as the previous question but using the "**Streets and Sidewalks - Overall**" column from the **general_services** table. How does the overall distribution of values look? How does it compare in zipcodes which have a slow resolution time for Streets, Roads, and Sidewalks requests?
28. Install the folium package (if you haven't already done so). Type **%conda install folium** in a cell by itself (notice the percent sign). Delete the cell where you installed folium and add '**import folium**' to the top cell where your packages are imported.
29. 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.
30. Assign the folium map you created in step 29 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

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**. Here are some questions that you can keep in mind as you are preparing your presentation, but there are lots of areas you could dive into in the provided datasets, so feel free to explore the provided datasets and come up with your own questions or ideas.

- What kinds of police calls occurred most often in 2018 and in which areas did these calls originate?
- When do calls occur most frequently?
- What kinds of requests for service are made to hubNashville, and are they being handled promptly? Are there particular kinds of requests in certain 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 other findings you would like to share?