Yodit_Ayalew_ Homework2_Report

1. Data Source

This project is based on a dataset that contains information on traffic violations in Montgomery County, Maryland. The dataset is available in different file formats and JSON file was downloaded for this homework task which is found at https://catalog.data.gov/dataset/traffic-violations. The data contains information about where the violation happened, the type of car, demographics on the person receiving the violation, and some other interesting information.

2. Data Preparation

The first thing was to look at the first few lines of the downloaded json file, and got the below output.

```
{
  "meta" : {
    "view" : {
      "id" : "4mse-ku6q",
      "name" : "Traffic Violations",
      "assetType" : "dataset",
      "averageRating" : 0,
      "category" : "Public Safety",
      "createdAt" : 1403103517,
      "description" : "This dataset contains traffic violation information
from all electronic traffic violations issued in the County.  Any informat
ion that can be used to uniquely identify the vehicle, the vehicle owner o
r the officer issuing the violation will not be published.\r\n\r\nUpdate F
requency:  Daily",
```

The above output has provided general information about the data set and we can also tell that the JSON data looks like a dictionary.

Extracting information on the columns

Steps:

- Opened the *marylandtraffic.json* file, then used the items method in ijson to extract a list from the file.
- Specified the path to the list using the *meta.view.columns* notation. Meta is a top-level key, which contains *view* inside, which contains *columns* inside it.
- Then, specified *meta.view.columns.item* to indicate that we should extract each individual item in the *meta.view.columns* list. The *items function* returned a generator, so used the list method to turn the generator into a Python list.

Code

```
#exploring columns and printing the first
with open(filename, 'r') as f:
  objects = ijson.items(f, 'meta.view.columns.item')
  columns = list(objects)
print(columns[0])
```

Output:

```
{'id': -1, 'name': 'sid', 'dataTypeName': 'meta_data', 'fieldName': ':sid'
, 'position': 0, 'renderTypeName': 'meta_data', 'format': {}, 'flags': ['h
idden']}
```

From the above output, it looks like each item in columns is a dictionary that contains information about each column. To get column names, we just extracted the *fieldName* key from each item in columns. And more than 50 column names are displayed. However, for this project, only some of the columns are selected and considered.

Based on defined columns we care about, and again used *ijson* to iteratively process the JSON file.

```
trafficdata = []

with open(filename, 'r') as f:

  objects = ijson.items(f, 'data.item')

  for row in objects:

    selected_row = []

    for item in selected_columns:

      selected_row.append(row[column_names.index(item)])

      trafficdata.append(selected_row)
```

Lastly, the JSON file data are transferred into Pandas Dataframe for running different analysis.

```
trafficdata = pd.DataFrame(trafficdata, columns=selected_columns)
```

Furthermore, time of day and the date of the stop are stored in two separate columns, *time_of_stop*, and *date_of_stop*. So, both columns are parsed, and turned them into a single *datetime* column. This will later help doing time-based analysis.

3. Analysis

Based on the dataset, the below few questions could get answer like

- What types of cars are most likely to be pulled over for speeding?
- what kind of police unit created the citation?
- What times of day are police most active?
- Which gender group is mostly pulled over?
- What are the most frequent types of violation?

A. Car Color

The below output shows how many stops are made by car color. **BLACK** cars are taking the lead followed by **SILVER**, and **CHROME** color is bottom of the list.

Output:

```
BLACK          594500
SILVER         501580
WHITE          454860
GRAY           326000
RED            225220
BLUE           222380
GREEN          108900
GOLD            86700
BLUE, DARK      64400
TAN             57700
MAROON          52360
GREEN, DK       36040
BLUE, LIGHT     35920
BEIGE           33020
N/A             24100
GREEN, LGT      17320
BROWN           14520
YELLOW          12180
ORANGE          11380
PURPLE           5680
BRONZE           5580
MULTICOLOR       2140
CREAM            1900
PINK              600
COPPER            580
CAMOUFLAGE        160
CHROME             40
Name: color, dtype: int64
```

B. Police Unit/Arrest Type

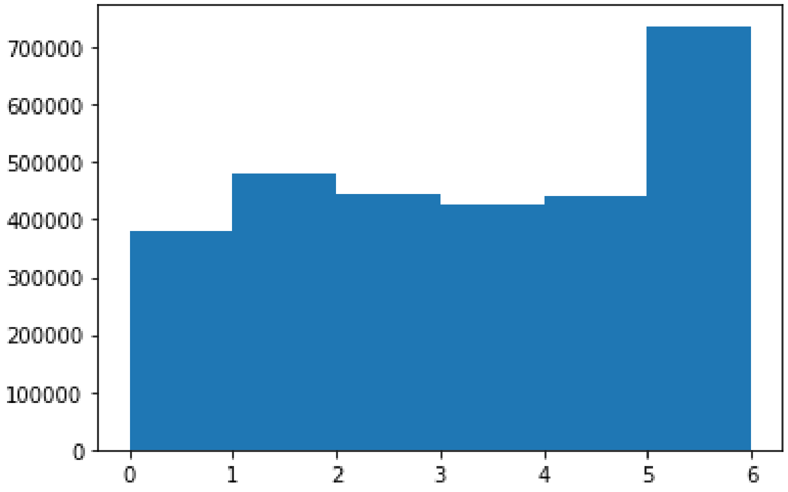
There are different kinds of policy units that are in charge for regulating traffic violations. And based on the analysis, **Marked Patrol Cars** take the lion share in creating the citation.

Output:

```
A - Marked Patrol                              2332160
Q - Marked Laser                                352420
B - Unmarked Patrol                              93060
E - Marked Stationary Radar                      24160
G - Marked Moving Radar (Stationary)             20240
S - License Plate Recognition                    16300
R - Unmarked Laser                               16220
M - Marked (Off-Duty)                            12440
O - Foot Patrol                                   9200
L - Motorcycle                                    8280
H - Unmarked Moving Radar (Stationary)            5580
I - Marked Moving Radar (Moving)                  4080
C - Marked VASCAR                                 2880
J - Unmarked Moving Radar (Moving)                2300
F - Unmarked Stationary Radar                     1860
D - Unmarked VASCAR                                940
N - Unmarked (Off-Duty)                            840
P - Mounted Patrol                                 480
K - Aircraft Assist                                 80
Name: arrest_type, dtype: int64
```

C. Times of day

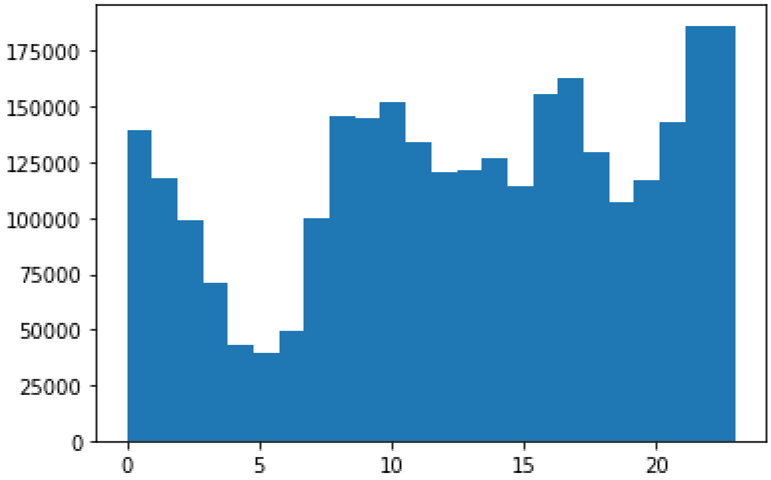
The below plot displayed which days result in the most traffic stops.



Note: in the above plot, Monday is 0, and Sunday is 6. It looks like Sunday has the most stops, and Monday has the least.

.

In addition, we can see from the plot below the most common traffic stop times. And the most stops happened around mid-night, and the fewest early in the morning around 5 am.



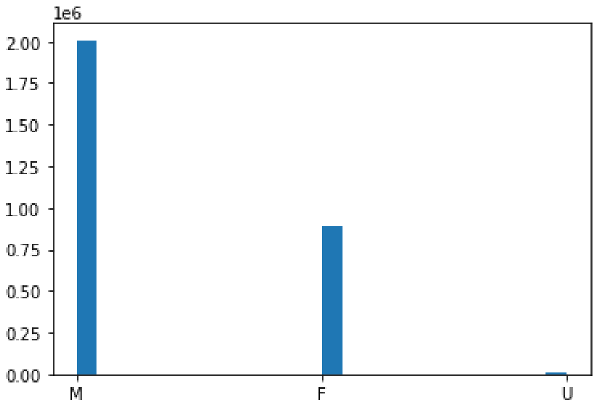
More analysis also made based on date and number of stops, and also number of stops in specific time like morning rush. Details are found in the code.

D. Gender

It is not a surprise that the analysis provided number of **MALE** drivers are twice than **FEMALE** drivers in traffic violation.

Output:

```
M     2009460
F      889220
U        4840
Name: gender, dtype: int64
```

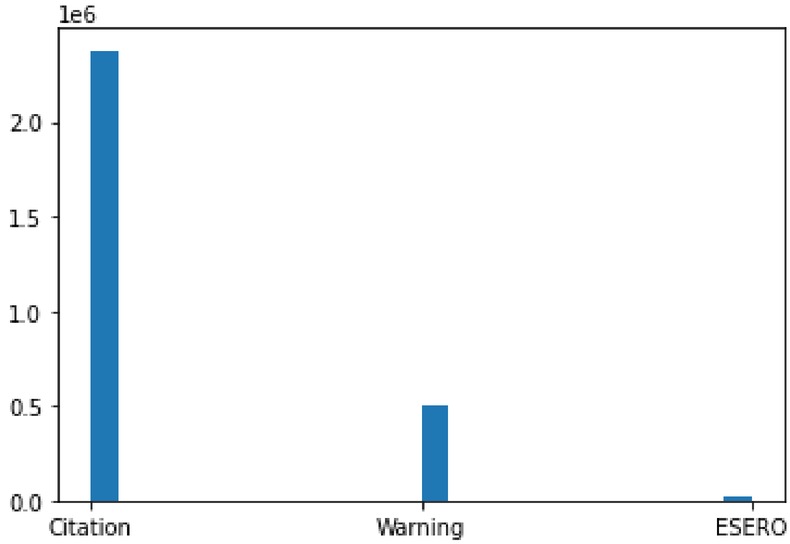


E. Violation Type

Among the three violation types, most drivers got a citation i.e. a written record of what you did wrong while operating your vehicle or while it is parked.

Output:

```
Citation    2378200
Warning      501120
ESERO         24200
Name: violation_type, dtype: int64
```


1e6
Citation
Warning
ESERO