philadelphia_EDA

April 30, 2025

1 Philadelphia Open Policing Project (OPP)

The Stanford Open Policing Project dataset comprises standardized records of stops (vehicular or pedestrian) collected from various U.S. law enforcement agencies. In this case we focus on Philadelphia data. Each entry represents a single stop and includes fields such as date, time, location, driver demographics, reason for the stop, and outcome.

[3]:
```python
# Import Libraries

import zipfile
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.tsa.seasonal import seasonal_decompose
import re
import folium
from folium.plugins import HeatMap, MarkerCluster
from folium import Element
import geojson
from branca.colormap import linear, LinearColormap


# Set pandas
pd.set_option('display.max_columns', None)
pd.set_option('display.float_format', lambda x: '%.2f' % x)

# Set visualization
plt.rcParams['figure.figsize'] = (20, 6)
plt.style.use('ggplot')
```

[4]:
```python
# Load the data

zip_path = "philadelphia_data.zip" # path for zip file

with zipfile.ZipFile(zip_path) as z: # CSV in zip file
    print(z.namelist())
```

```
    with z.open(z.namelist()[0]) as f: # read CSV file
        df = pd.read_csv(f)

df.head()
```

```
['pa_philadelphia_2020_04_01.csv']

C:\Users\acast\AppData\Local\Temp\ipykernel_30128\495672545.py:9: DtypeWarning:
Columns (7) have mixed types. Specify dtype option on import or set
low_memory=False.
  df = pd.read_csv(f)
```

[4]:

```
   raw_row_number        date      time                 location    lat     lng  \
0          411981  2014-01-01  01:14:00                      NaN    NaN     NaN
1          407442  2014-01-01  01:57:00                      NaN    NaN     NaN
2          217556  2014-01-01  03:30:00    3400 BLOCK SPRUCE ST 39.95 -75.19
3          217557  2014-01-01  03:40:00    3400 BLOCK SPRUCE ST 39.95 -75.19
4          230988  2014-01-01  08:30:00  N 56TH ST / UPLAND WAY 39.98 -75.23

   district service_area  subject_age subject_race subject_sex        type  \
0     19.00          191        31.00        black        male  pedestrian
1     12.00          121        21.00        black        male  pedestrian
2     18.00          183        24.00        black        male  pedestrian
3     18.00          183        20.00        black        male  pedestrian
4     19.00          193        31.00        black        male   vehicular

   arrest_made outcome contraband_found  frisk_performed  search_conducted  \
0         True  arrest             True            False              True
1         True  arrest            False             True              True
2        False     NaN              NaN            False             False
3        False     NaN              NaN            False             False
4        False     NaN              NaN            False             False

   search_person  search_vehicle            raw_race  \
0           True           False  Black - Non-Latino
1           True           False  Black - Non-Latino
2          False           False  Black - Non-Latino
3          False           False  Black - Non-Latino
4          False           False  Black - Non-Latino

   raw_individual_contraband  raw_vehicle_contraband
0                       True                   False
1                      False                   False
2                      False                   False
3                      False                   False
4                      False                   False
```

Column name	Column meaning	Example value
raw_row_number	An number used to join clean data back to the raw data	38299
date	The date of the stop, in YYYY-MM-DD format. Some states do not provide the exact stop date: for example, they only provide the year or quarter in which the stop occurred. For these states, stop_date is set to the date at the beginning of the period: for example, January 1 if only year is provided.	2017-02-02
time	The 24-hour time of the stop, in HH:MM format.	20:15
location	The freeform text of the location. Occasionally, this represents the concatenation of several raw fields, i.e. street_number, street_name	248 Stockton Rd.
lat	The latitude of the stop. If not provided by the department, we attempt to geocode any provided address or location using Google Maps. Google Maps returns a "best effort" response, which may not be completely accurate if the provided location was malformed or underspecified. To protect against suprious responses, geocodes more than 4 standard deviations from the median stop lat/lng are set to NA.	72.23545
lng	The longitude of the stop. If not provided by the department, we attempt to geocode any provided address or location using Google Maps. Google Maps returns a "best effort" response, which may not be completely accurate if the provided location was malformed or underspecified. To protect against suprious responses, geocodes more than 4 standard deviations from the median stop lat/lng are set to NA.	115.2808
district	Police district. If not provided, but we have retrieved police department shapefiles and the location of the stop, we geocode the stop and find the district using the shapefiles.	8
service_area	Police service area. If not provided, but we have retrieved police department shapefiles and the location of the stop, we geocode the stop and find the service area using the shapefiles.	8
subject_age	The age of the stopped subject. When date of birth is given, we calculate the age based on the stop date. Values outside the range of 10-110 are coerced to NA.	54.23
subject_race	The race of the stopped subject. Values are standardized to white, black, hispanic, asian/pacific islander, and other/unknown	hispanic
subject_sex	The recorded sex of the stopped subject.	female
type	Type of stop: vehicular or pedestrian.	vehicular
arrest_made	Indicates whether an arrest made.	FALSE
outcome	The strictest action taken among arrest, citation, warning, and summons.	citation
contraband_found	Indicates whether contraband was found. When search_conducted is NA, this is coerced to NA under the assumption that contraband_found shouldn't be discovered when no search occurred and likely represents a data error.	FALSE
frisk_performed	Indicates whether a frisk was performed. This is technically different from a search, but departments will sometimes include frisks as a search type.	TRUE
search_conducted	Indicates whether any type of search was conducted, i.e. driver, passenger, vehicle. Frisks are excluded where the department has provided resolution on both.	TRUE
search_person	Indicates whether a search of a person has occurred. This is only defined when search_conducted is TRUE.	TRUE

Column name	Column meaning	Example value
search_vehicle	Indicates whether a search of a vehicle has occurred. This is only defined when search_conducted is TRUE.	TRUE
raw_race	Raw racial data as received before standardization.	h
raw_individual_contraband	Raw field related to contraband on the individual.	drug
raw_vehicle_contraband	Raw field related to contraband in the vehicle.	

[6]:
```
df.shape
```

[6]:
```
(1865096, 22)
```

We had more than 1.8M records.

[8]:
```
df.info(show_counts = True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1865096 entries, 0 to 1865095
Data columns (total 22 columns):
 #   Column                     Non-Null Count    Dtype
---  ------                     --------------    -----
 0   raw_row_number             1865096 non-null  object
 1   date                       1865096 non-null  object
 2   time                       1865096 non-null  object
 3   location                   1827596 non-null  object
 4   lat                        1760399 non-null  float64
 5   lng                        1760399 non-null  float64
 6   district                   1865095 non-null  float64
 7   service_area               1865092 non-null  object
 8   subject_age                1860537 non-null  float64
 9   subject_race               1865096 non-null  object
 10  subject_sex                1864446 non-null  object
 11  type                       1865096 non-null  object
 12  arrest_made                1865096 non-null  bool
 13  outcome                    95476 non-null    object
 14  contraband_found           116455 non-null   object
 15  frisk_performed            1865096 non-null  bool
 16  search_conducted           1865096 non-null  bool
 17  search_person              1865096 non-null  bool
 18  search_vehicle             1865096 non-null  bool
 19  raw_race                   1865096 non-null  object
 20  raw_individual_contraband  1865096 non-null  bool
 21  raw_vehicle_contraband     1865096 non-null  bool
dtypes: bool(7), float64(4), object(11)
memory usage: 225.9+ MB
```

[9]:
```
df.isna().sum()
```

[9]:
```
raw_row_number                     0
date                               0
time                               0
location                       37500
lat                           104697
lng                           104697
district                           1
service_area                       4
subject_age                     4559
subject_race                       0
subject_sex                      650
type                               0
arrest_made                        0
outcome                      1769620
contraband_found             1748641
frisk_performed                    0
search_conducted                   0
search_person                      0
search_vehicle                     0
raw_race                           0
raw_individual_contraband          0
raw_vehicle_contraband             0
dtype: int64
```

- Most of the columns are completed
- There are around 100,000 missing values for *lat*, *lng*
- *outcome* and *contraband_found* have more than 1.7M missing values. But that could be because police didn't found contraband or took actions after the stops.

[11]:
```
df.describe()
```

[11]:
```
             lat         lng    district  subject_age
count 1760399.00 1760399.00 1865095.00   1860537.00
mean       39.99      -75.16       18.97        34.83
std         0.04        0.05       10.55        13.34
min        39.88      -75.28        1.00        10.00
25%        39.96      -75.20       12.00        24.00
50%        39.99      -75.16       18.00        31.00
75%        40.02      -75.13       25.00        44.00
max        40.14      -74.96       77.00       110.00
```

[12]:
```
df.describe(include = "O")
```

[12]:
```
       raw_row_number        date      time                  location  \
count         1865096     1865096   1865096                   1827596
unique        1865096        1565      1440                     59246
top            411981  2015-10-27  20:00:00  3200 BLOCK KENSINGTON AV
```

```
freq                 1        2139      17957                      3610

       service_area subject_race subject_sex       type outcome  \
count       1865092      1865096     1864446    1865096   95476
unique          270            6           2          2       1
top             242        black        male  vehicular  arrest
freq          86375      1244249     1397206    1167683   95476

       contraband_found            raw_race
count            116455             1865096
unique                2                   7
top               False  Black - Non-Latino
freq              83225             1244249
```

Some patterns are shown but these would be analyzed for each column

1.1 Columns Analysis

1.1.1 1. raw_row_number

The column shows a numeric ID but in some rows there are more than one number

[17]:
```python
df["raw_row_number"] = df["raw_row_number"].str.replace("|", "-")
```

[18]:
```python
df[df["raw_row_number"].str.contains("-")]["raw_row_number"]
```

[18]:
```
86                  231739-231740
133                 358835-358836
243                 249320-249321
437                 156597-156598
447          250868-250870-400834
                        …
1864369           1788091-1791591
1864375           1788931-1789797
1864807           1790300-1790309
1864966           1794964-1794969
1865012           1790578-1790847
Name: raw_row_number, Length: 24796, dtype: object
```

Because this is an number used to join clean data back to the raw data, this column is related to the database structure but not the recorded information. Therefore, this column would be deleted

[20]:
```python
df.drop(columns = ["raw_row_number"], inplace = True)
df.head()
```

[20]:
```
         date      time                location   lat    lng  district  \
0  2014-01-01  01:14:00                     NaN   NaN    NaN     19.00
1  2014-01-01  01:57:00                     NaN   NaN    NaN     12.00
2  2014-01-01  03:30:00    3400 BLOCK SPRUCE ST 39.95 -75.19     18.00
```

```
3  2014-01-01  03:40:00    3400 BLOCK SPRUCE ST 39.95 -75.19     18.00
4  2014-01-01  08:30:00  N 56TH ST / UPLAND WAY 39.98 -75.23     19.00

  service_area  subject_age subject_race subject_sex        type  arrest_made  \
0          191        31.00        black        male  pedestrian         True
1          121        21.00        black        male  pedestrian         True
2          183        24.00        black        male  pedestrian        False
3          183        20.00        black        male  pedestrian        False
4          193        31.00        black        male   vehicular        False

  outcome contraband_found  frisk_performed  search_conducted  search_person  \
0  arrest             True            False              True           True
1  arrest            False             True              True           True
2     NaN              NaN            False             False          False
3     NaN              NaN            False             False          False
4     NaN              NaN            False             False          False

   search_vehicle            raw_race  raw_individual_contraband  \
0           False  Black - Non-Latino                       True
1           False  Black - Non-Latino                      False
2           False  Black - Non-Latino                      False
3           False  Black - Non-Latino                      False
4           False  Black - Non-Latino                      False

   raw_vehicle_contraband
0                   False
1                   False
2                   False
3                   False
4                   False
```

1.1.2 2. date

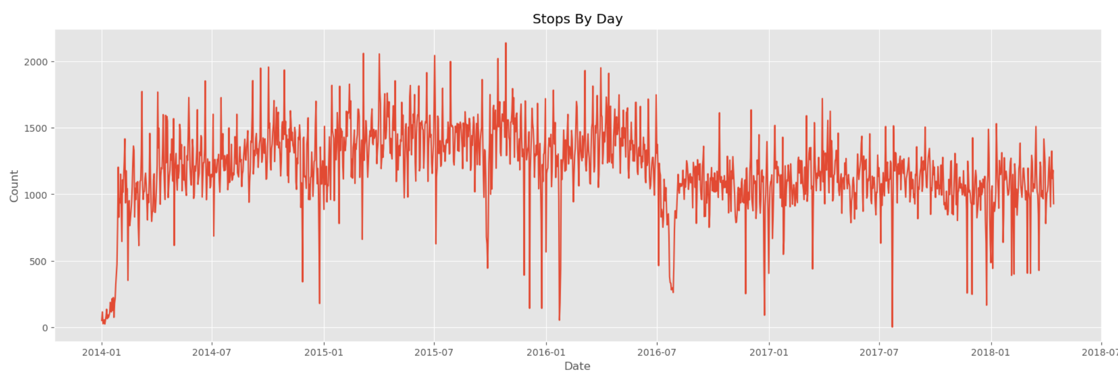
[22]:
```
df["date"]
```

[22]:
```
0          2014-01-01
1          2014-01-01
2          2014-01-01
3          2014-01-01
4          2014-01-01
              …
1865091    2018-04-14
1865092    2018-04-14
1865093    2018-04-14
1865094    2018-04-14
1865095    2018-04-14
Name: date, Length: 1865096, dtype: object
```

[23]:
```
df["date"] = pd.to_datetime(df["date"]) # Convert objecto to date time
df["date"]
```

[23]:
```
0          2014-01-01
1          2014-01-01
2          2014-01-01
3          2014-01-01
4          2014-01-01
              …
1865091    2018-04-14
1865092    2018-04-14
1865093    2018-04-14
1865094    2018-04-14
1865095    2018-04-14
Name: date, Length: 1865096, dtype: datetime64[ns]
```

[24]:
```
date_data = df.groupby("date").size() # Group by day
sns.lineplot(data = date_data)
plt.xlabel("Date")
plt.ylabel("Count")
plt.title("Stops By Day")
plt.show()
```



[25]:
```
print(f"Day with the most stops was {date_data.idxmax()} with {date_data.max()}␣
↪events")
```

```
Day with the most stops was 2015-10-27 00:00:00 with 2139 events
```

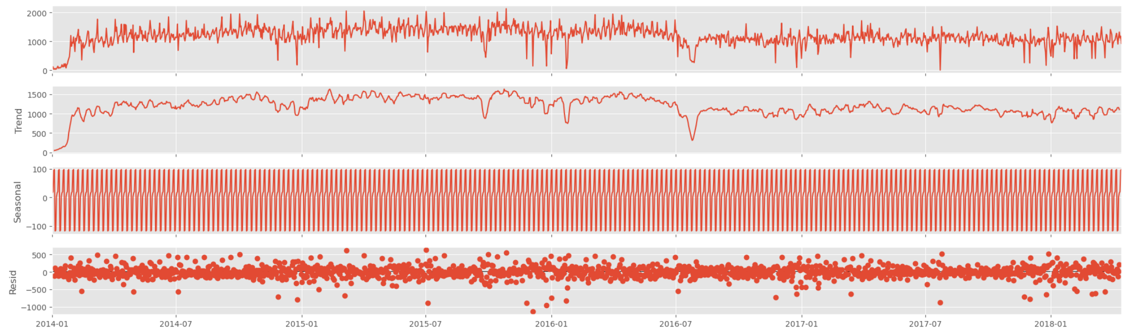
[26]:
```
print(f"Day with fewest sotps was {date_data.idxmin()} with {date_data.min()}␣
↪events")
```

```
Day with fewest sotps was 2017-07-23 00:00:00 with 1 events
```

[27]:
```
date_data.index.inferred_freq # Data freqency
```

[27]: 'D'

[28]:
```
# Seasonal decompose
decompose = seasonal_decompose(date_data, model='additive', period = 7)
decompose.plot()
plt.tight_layout()
plt.show()
```



The observed data shows a relatively high and stable daily count until early 2016, after which a noticeable drop occurs and stabilizes at a lower level. The trend component confirms this shift, highlighting a gradual increase in activity through 2014–2015 followed by a sharp decline around early 2016 and a flatter pattern afterward. The seasonal component displays a strong, consistent weekly cycle, suggesting that the data exhibits predictable fluctuations tied to days of the week. This pattern remains stable in shape and amplitude throughout the entire period. Lastly, the residual component shows moderate dispersion around zero, with occasional outliers, indicating that while the decomposition explains much of the variability, there are still some irregular, potentially exceptional events not captured by the model. Overall, this decomposition suggests strong weekly seasonality, a meaningful long-term trend shift, and relatively well-behaved residuals, making it a valuable basis for further forecasting or anomaly detection.

The decrease in stops in Philadelphia in 2016 was due to a combination of factors, including the implementation of police reforms (stop and frisk), increased oversight of stop practices, and a focus on racial equity. These measures reflect a concerted effort by the city to promote fairer and more effective policing practices.

[30]:
```
# Create additional date values

df["Year"] = df["date"].dt.year
df["Month"] = df["date"].dt.month_name()
df["Day"] = df["date"].dt.day
df["Day_Week"] = df["date"].dt.day_name()
df[["Year", "Month", "Day", "Day_Week"]].head()
```

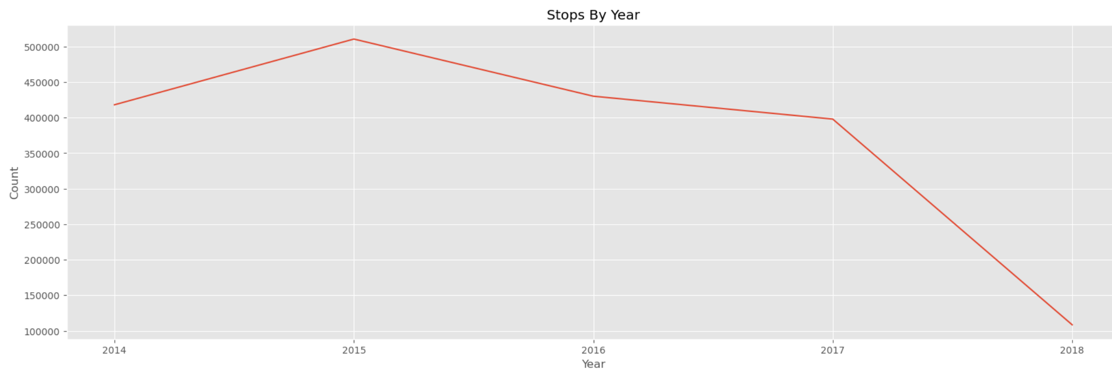
[30]:

	Year	Month	Day	Day_Week
0	2014	January	1	Wednesday
1	2014	January	1	Wednesday

```
2  2014  January    1  Wednesday
3  2014  January    1  Wednesday
4  2014  January    1  Wednesday
```

2.1 Year

[32]:
```
year_data = df.groupby(df["Year"]).size().reset_index(name = "Count")
sns.lineplot(data = year_data, x = "Year", y = "Count")
plt.xticks(year_data["Year"])
plt.title("Stops By Year")
plt.show()
```



[33]:
```
year_data
```

[33]:

	Year	Count
0	2014	418031
1	2015	510534
2	2016	430114
3	2017	397908
4	2018	108509

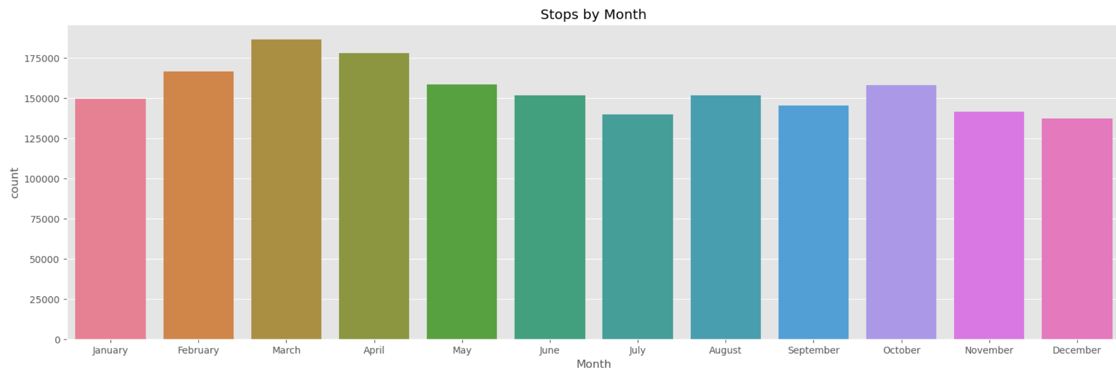
In 2014, 2016 and 2017 around 40,000 stops were made. The highest number of stops was in 2015, with more than 50,000. The lower figure compared to 2018 is due to the fact that data is available up to April 14, 2018.

The decrease after 2015, (2016-2017) is related to the modification of *stop and frisk* practices among police forces in order to guarantee racial equity.

2.2 Month

[36]:
```
months_order = ['January', 'February', 'March', 'April', 'May', 'June',
                'July', 'August', 'September', 'October', 'November', 'December']

df["Month"] = pd.Categorical(df["Month"], categories = months_order, ordered =␣
 ↪True)
```

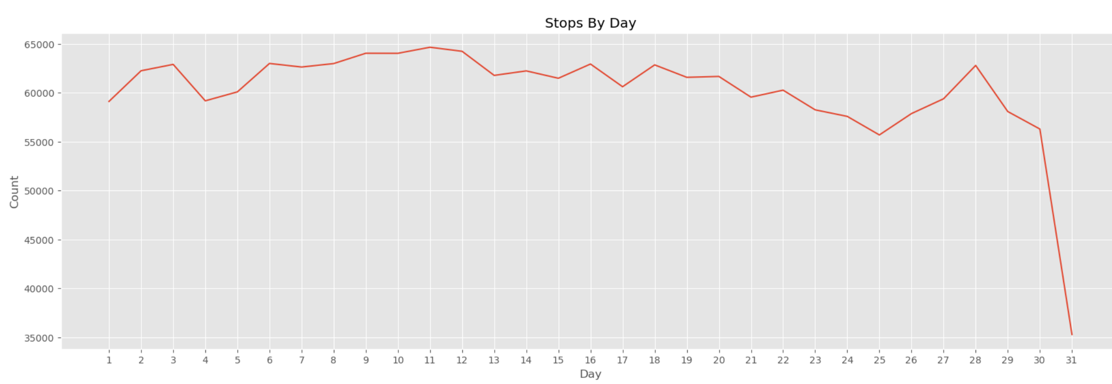
```
sns.countplot(data = df, x = "Month", hue = "Month")
plt.title("Stops by Month")
plt.show()
```



The bar chart shows that March has the highest number of stops, followed closely by April and February. This peak in March may be explained by several factors. First, it marks the transition from winter to spring, which often leads to increased traffic as weather conditions improve. Additionally, law enforcement agencies may launch seasonal traffic enforcement campaigns during this time, focusing on issues like speeding or impaired driving. March also coincides with the end of the first fiscal quarter, which may prompt intensified operations for reporting or budgetary reasons. Furthermore, the return to school or university after winter breaks may increase daily flow. This idea is further supported by the noticeable decline in stops starting in November, when winter begins and road activity typically decreases due to colder weather and holiday-related slowdowns.

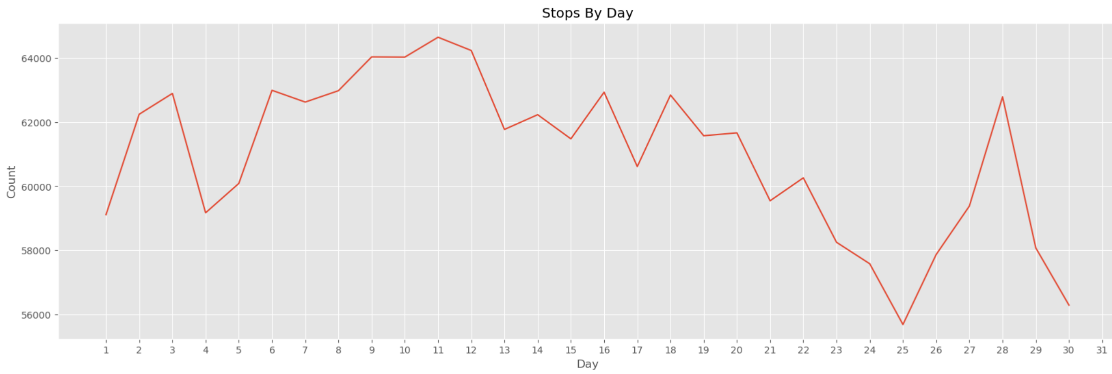
2.3 Day

[39]:
```
day_data = df.groupby("Day").size().reset_index(name = "Count")
sns.lineplot(data = day_data, x = "Day", y = "Count")
plt.xticks(day_data["Day"])
plt.title("Stops By Day")
plt.show()
```



There are few data on day 31 because not all months have 31 days.

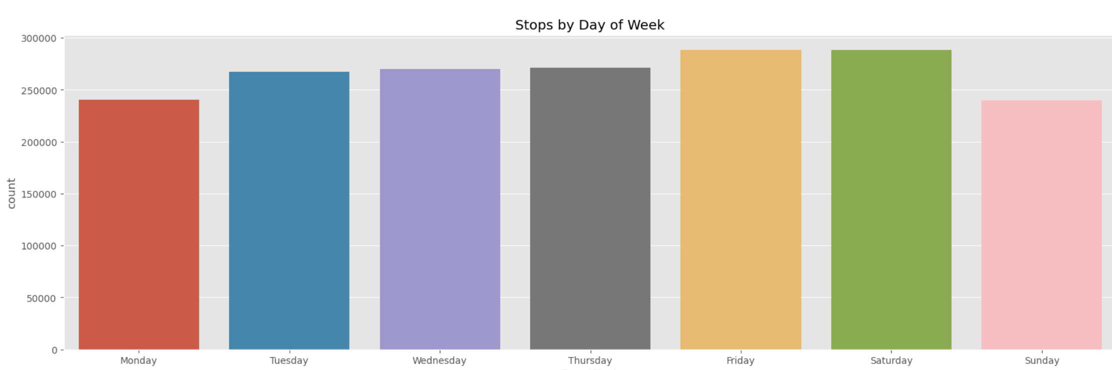
[41]:
```
# Do not take into account day 31 info
sns.lineplot(data = day_data[day_data["Day"] < 31], x = "Day", y = "Count")
plt.xticks(day_data["Day"])
plt.title("Stops By Day")
plt.show()
```



The first half of the month, particularly the first 12 days, shows consistently high numbers of stops, peaking around the 11th. After that, there's a gradual decline in stops, reaching a significant low around the 25th. Interestingly, a brief surge occurs between the 27th and 29th before dropping again at the end of the month. This pattern may suggest increased enforcement activity at the beginning of the month, possibly linked to administrative cycles, resource availability, or policy targets, followed by a slowdown, and then a final push toward the month's end.

2.4 Day of the Week

[44]:
```
days_order = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday",␣
↪"Saturday", "Sunday"]
df["Day_Week"] = pd.Categorical(df["Day_Week"], categories = days_order,␣
↪ordered = True)
sns.countplot(data = df, x = "Day_Week", hue = "Day_Week")
plt.title("Stops by Day of Week")
plt.show()
```



Stops gradually increase from Monday through Saturday, with Friday and Saturday showing the highest counts. This suggests intensified traffic monitoring and enforcement toward the end of the workweek and into the weekend, possibly due to higher traffic volumes or a greater focus on weekend-related infractions. In contrast, Monday and Sunday show the lowest number of stops, which may reflect lighter traffic, fewer enforcement operations, or reduced mobility during those days. Overall, the chart highlights a weekly cycle in stops that aligns with expected fluctuations in daily traffic behavior.

1.1.3 3. time

[47]:
```python
df["time"]
```

[47]:
```
0          01:14:00
1          01:57:00
2          03:30:00
3          03:40:00
4          08:30:00
              …
1865091    21:36:00
1865092    22:01:00
1865093    22:48:00
1865094    22:48:00
1865095    23:10:00
Name: time, Length: 1865096, dtype: object
```

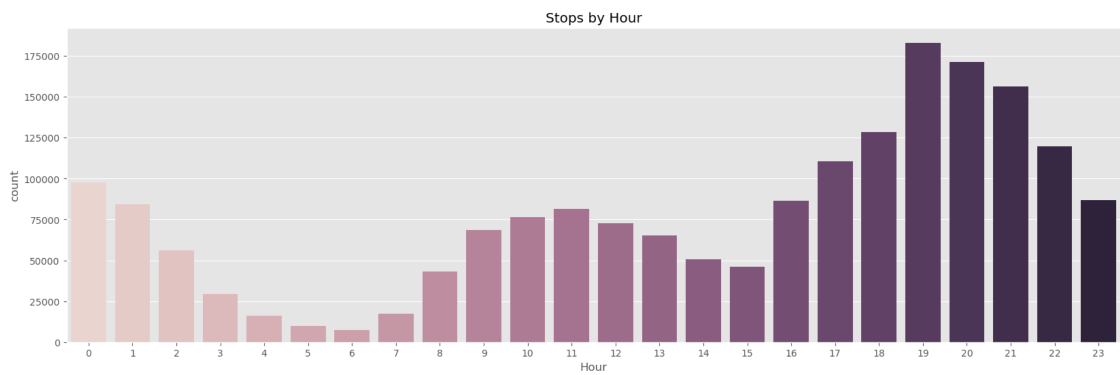
[48]:
```python
df["time"] = pd.to_datetime(df["time"], format = '%H:%M:%S').dt.time
df["time"]
```

[48]:
```
0          01:14:00
1          01:57:00
2          03:30:00
3          03:40:00
4          08:30:00
              …
1865091    21:36:00
1865092    22:01:00
1865093    22:48:00
1865094    22:48:00
1865095    23:10:00
Name: time, Length: 1865096, dtype: object
```

[49]:
```python
df["Hour"] = df["time"].apply(lambda x: x.hour)
df["Hour"]
```

[49]:
```
0           1
1           1
2           3
3           3
4           8
           ..
1865091    21
1865092    22
1865093    22
1865094    22
1865095    23
Name: Hour, Length: 1865096, dtype: int64
```

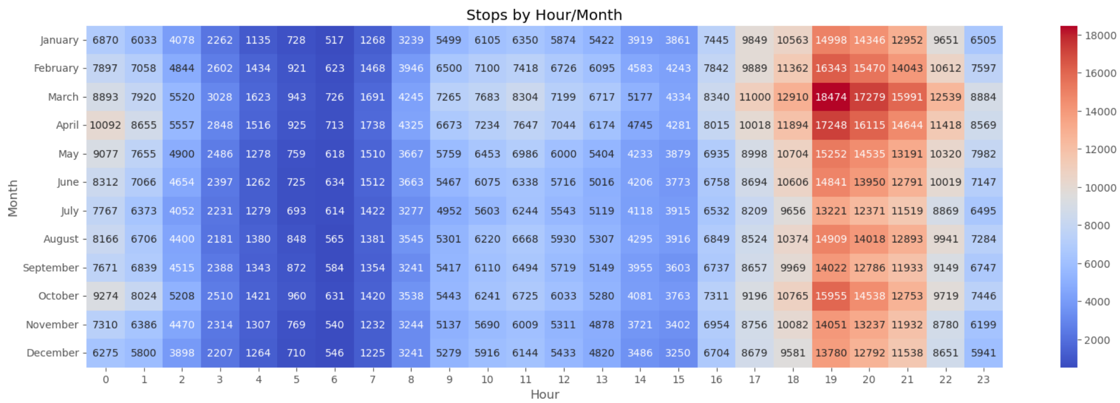
[50]:
```
sns.countplot(data = df, x = "Hour", hue = "Hour")
plt.legend().remove()
plt.title("Stops by Hour")
plt.show()
```



The early morning hours, particularly between 0:00 and 2:00, show relatively high stop counts, possibly linked to nighttime patrols or late-night traffic enforcement. Activity then drops sharply between 3:00 and 7:00, likely reflecting reduced traffic and a potential change of shift for police forces around 6:00, which may temporarily lower enforcement presence. From 8:00 onward, the number of stops begins to rise, with moderate activity observed during late morning and a slight dip around 12:00–14:00, which could correspond to lunch hours, both for drivers and officers. The most significant surge begins at 16:00 and peaks around 19:00, aligning with evening rush hour and increased road activity. After 20:00, the counts gradually decline but remain relatively high through 23:00. Overall, the chart suggests that enforcement patterns are strongly influenced by daily traffic rhythms, operational schedules, and practical considerations like meal breaks and shift transitions.

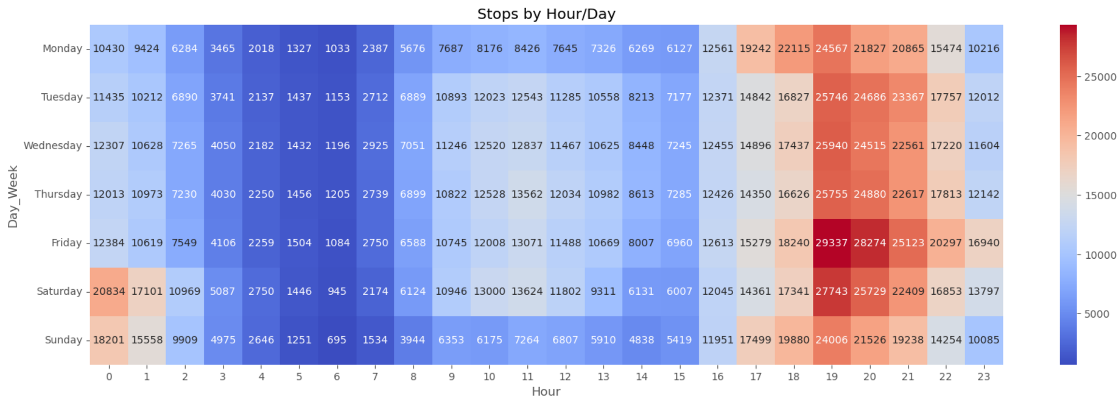
[52]:
```
pivot_month_hour = df.pivot_table(index = "Month",
                                  columns = "Hour",
                                  aggfunc = "size",
                                  observed = False)
```

```
sns.heatmap(pivot_month_hour, cmap = "coolwarm", annot = True, fmt='g')
plt.title("Stops by Hour/Month")
plt.show()
```



Consistent with earlier analyses, the highest concentration of stops occurs between 17:00 and 21:00, reflecting peak traffic periods, especially during evening commutes. March, April, and May stand out with the most intense activity during these hours, supporting the idea that stops increase in spring due to improved weather, higher traffic flow, and possibly seasonal enforcement efforts. Conversely, the lowest levels of activity are observed between 3:00 and 6:00 across all months, likely due to reduced mobility and early morning police shift transitions. December, January, and February show relatively lower totals overall, which may be attributed to winter conditions that limit driving and reduce the frequency of stops.

[54]:
```
pivot_day_hour = df.pivot_table(index = "Day_Week",
                                  columns = "Hour",
                                  aggfunc = "size",
                                  observed = False)

sns.heatmap(pivot_day_hour, cmap = "coolwarm", annot = True, fmt='g')
plt.title("Stops by Hour/Day")
plt.show()
```



As seen in previous analyses, the highest volume of stops occurs between 17:00 and 21:00 across all days, with Friday and Saturday showing the most intense activity—peaking notably around 19:00 and 20:00. This likely reflects increased traffic volume and police presence during weekend nights, possibly targeting leisure-related mobility and impaired driving. In contrast, the early morning hours between 3:00 and 6:00 consistently register the lowest stop counts, aligning with expected reductions in traffic and potential police shift changes. Weekdays exhibit a smoother progression of stops from morning through evening, while weekends show a broader spread of higher activity throughout the day, especially starting from midday. Sunday maintains elevated stop levels until late evening, though slightly lower than Saturday.

1.1.4 4. Location

[57]:
```python
# All stops locations
location_data = pd.DataFrame(df["location"].dropna())
location_data
```

[57]:
```
                                      location
2                         3400 BLOCK SPRUCE ST
3                         3400 BLOCK SPRUCE ST
4                       N 56TH ST / UPLAND WAY
5            CHESTNUT ST   / S SCHUYLKILL AV W
6                         N 52ND ST / GAINOR RD
…                                           …
1865091                 S 59TH ST / ELMWOOD AV
1865092                   2600 BLOCK JUDSON ST
1865093                   500 BLOCK E OLNEY AV
1865094                   500 BLOCK E OLNEY AV
1865095                  200 BLOCK W LEHIGH AV

[1827596 rows x 1 columns]
```

[58]:
```python
# Locations with higest stops
location_data.value_counts().head(10)
```

[58]:

```
location
3200 BLOCK KENSINGTON AV      3610
3100 BLOCK KENSINGTON AV      3576
800 BLOCK E ALLEGHENY AV      3471
3000 BLOCK KENSINGTON AV      2925
5900 BLOCK MARKET ST          2847
100 BLOCK W LEHIGH AV         2696
100 BLOCK E TUSCULUM ST       2619
4600 BLOCK E ROOSEVELT BLVD   2588
100 BLOCK W CAMBRIA ST        2356
600 BLOCK E INDIANA AV        2352
Name: count, dtype: int64
```

[59]:

```
# Locations with fewest stops
location_data.value_counts().tail(10)
```

[59]:

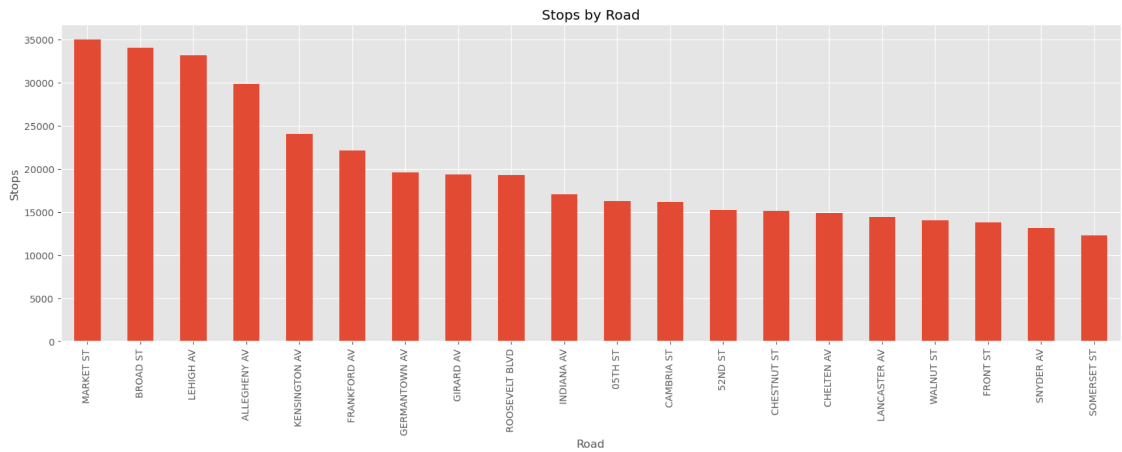
```
location
N 1500 BUTLER ST            1
55TH & RACE ST              1
N 14TH ST / WINDRIM AV      1
N 14TH ST / W CAYUGA ST     1
N 13TH ST/POPLAR            1
N 13TH ST/ W RUSCOMB ST     1
55TH & ARCH ST              1
55TH & CATHERINE ST         1
55TH & GIRARD AVE           1
s BROAD ST / PATTISON AV    1
Name: count, dtype: int64
```

Let's extract the name of the streets, avenues, or roads where a stop ocurred

[61]:

```
def name_road(location):

    location = location.upper() # all the lettes in upper case
    location = location.split('/')[-1] # Some stops have several roads, let's␣
 ↪focus on the last one
    location = re.sub(r'\b\d+\b', '', location) # remove just numbers
    location = re.sub(r'\b(BLOCK|N|S|E|W)\b', '', location) # remove BLOCK and␣
 ↪cardinal points
    location = re.sub(r'\s+', ' ', location) # remove consecutive spaces      ␣
 ↪
    return location.strip() # remove spaces before and after the word

# Aplicar la función
location_data['clean_street'] = location_data['location'].apply(name_road)
```

```
[62]: # Top 20 roads with hightest stops

location_data["clean_street"].value_counts().head(20).plot(kind = "bar")
plt.xlabel("Road")
plt.ylabel("Stops")
plt.title("Stops by Road")
plt.show()
```



"MARKET ST" has the highest number of stops, followed closely by "BROAD ST" and "LEHIGH AV". These roads are likely major thoroughfares with high traffic volumes, explaining their elevated stop counts.

We can visualize the "dangerous" roads using `folium`

```
[64]: top_stops_streets = pd.DataFrame(location_data["clean_street"].value_counts().
      ↪head(10).reset_index(name = "count"))
      top_stops_streets
```

```
[64]:      clean_street  count
      0       MARKET ST  34972
      1        BROAD ST  34094
      2       LEHIGH AV  33196
      3    ALLEGHENY AV  29815
      4   KENSINGTON AV  24004
      5    FRANKFORD AV  22158
      6   GERMANTOWN AV  19618
      7       GIRARD AV  19344
      8  ROOSEVELT BLVD  19296
      9      INDIANA AV  17077
```

```
[65]: top_stops_streets[["name", "type"]] = top_stops_streets["clean_street"].str.
      ↪split(" ", expand = True)
```

```python
top_stops_streets["type"] = top_stops_streets["type"].str.replace("AV", "AVE")
top_stops_streets
```

[65]:

	clean_street	count	name	type
0	MARKET ST	34972	MARKET	ST
1	BROAD ST	34094	BROAD	ST
2	LEHIGH AV	33196	LEHIGH	AVE
3	ALLEGHENY AV	29815	ALLEGHENY	AVE
4	KENSINGTON AV	24004	KENSINGTON	AVE
5	FRANKFORD AV	22158	FRANKFORD	AVE
6	GERMANTOWN AV	19618	GERMANTOWN	AVE
7	GIRARD AV	19344	GIRARD	AVE
8	ROOSEVELT BLVD	19296	ROOSEVELT	BLVD
9	INDIANA AV	17077	INDIANA	AVE

[66]:

```python
with open(r"GeoJson_Files\streets.geojson") as f:
    data_streets = geojson.load(f) # Philadelphia Strees GeoJson from https://
↪www.pasda.psu.edu/uci/DataSummary.aspx?dataset=7102

map = folium.Map(location=[39.96, -75.15], zoom_start=12,␣
↪tiles='cartodbpositron') # Base map

# Color setup
max_val = max(top_stops_streets["count"])
min_val = min(top_stops_streets["count"])
colormap = linear.YlOrRd_05.to_step(10)
colormap = colormap.scale(min_val, max_val)
colormap.caption = 'Roads with more stops'

# Find the GeoJson Data for the Top Streets
features_filtered = []
for feature in data_streets['features']:

    name = feature['properties'].get('ST_NAME', '').upper()
    type = feature['properties'].get('ST_TYPE', '').upper()

    for i in list(zip(top_stops_streets["name"], top_stops_streets["type"])):
        if (name == i[0]) and (type == i[1]):
            feature['properties']['value'] = int(top_stops_streets.
↪loc[top_stops_streets["name"] == name, "count"].values[0])
            features_filtered.append(feature)

geojson_filtered = {
    "type": "FeatureCollection",
    "features": features_filtered}

# Style
```

```
def style(feature):
    value = feature['properties']['value']
    return {
        'color': colormap(value),
        'weight': 4,
        'opacity': 0.8
    }

# Add the lines to the map
folium.GeoJson(
    geojson_filtered,
    style_function = style,
    tooltip = folium.GeoJsonTooltip(fields=["ST_NAME", "value"], aliases=["Road:
 ↪", "Value:"])).add_to(map)

colormap.add_to(map)
map
```

[66]: <folium.folium.Map at 0x1534795d340>

[67]:
```
map.save("HTML_Maps/top_roads.html") # Save map as HTML
```

The concentration of stops is clearly aligned with the city's major arterial roads, particularly those running through central and north Philadelphia. The color gradient on the map effectively illustrates the density of stops, with deeper red tones highlighting the roads with the heaviest enforcement. This distribution suggests that these corridors are key areas for traffic enforcement and possibly reflect regions with higher traffic volume or law enforcement focus.

1.1.5 5. Lat & Lng

[70]:
```
coordinates = df[["lat", "lng"]].dropna()
coordinates.head()
```

[70]:

	lat	lng
2	39.95	-75.19
3	39.95	-75.19
4	39.98	-75.23
5	39.95	-75.18
6	39.99	-75.23

[71]:
```
len(coordinates)
```

[71]: 1760399

[72]:
```
coordinates_sample = coordinates.sample(frac = 0.01, random_state = 42)
```

[73]:
```
#Heat Map
map2 = folium.Map(location=[39.96, -75.15], zoom_start=12,␣
↪tiles='cartodbpositron') # Base map
folium.plugins.HeatMap(coordinates_sample.values.tolist(), radius = 8, blur =␣
↪9, max_zoom = 13).add_to(map2)
map2
```

[73]:
```
<folium.folium.Map at 0x1534b03e690>
```

[74]:
```
map2.save("HTML_Maps/stop_heatmap.html") # Save map as HTML
```

[75]:
```
# Points

map3 = folium.Map(location=[39.96, -75.15], zoom_start = 12,␣
↪tiles='cartodbpositron') # Base map

for i, row in coordinates_sample.iterrows():
    folium.Circle(
        location=[row['lat'], row['lng']],
        radius = 20,
        color = 'red',
        fill = True,
        fill_opacity = 0.3
    ).add_to(map3)

map3
```

[75]:
```
<folium.folium.Map at 0x1535d3be900>
```

[76]:
```
map3.save("HTML_Maps/stop_points.html") # Save map as HTML
```

[77]:
```
# Cluster

map4 = folium.Map(location=[39.96, -75.15], zoom_start = 12,␣
↪tiles='cartodbpositron') # Base map

cluster = MarkerCluster().add_to(map4)
for i, row in coordinates_sample.iterrows():
    folium.Marker(
        location=[row['lat'], row['lng']],
        popup=f"Lat: {row['lat']}, Lng: {row['lng']}"
    ).add_to(cluster)

map4
```

[77]:
```
<folium.folium.Map at 0x1535d3be120>
```

[78]:
```python
map4.save("HTML_Maps/stop_clusters.html") # Save map as HTML
```

These maps reveal a high concentration of stops in central and northeastern parts of the city. Particularly dense clusters are visible around key arterial roads and intersections, suggesting areas with significant traffic or heightened police presence. The central area, encompassing Center City and nearby neighborhoods, stands out due to its consistent density of stops. Moreover, corridors extending north and northwest from the city center also show elevated stop activity, potentially reflecting traffic enforcement patterns along major routes.

1.1.6 6. District

[81]:
```python
df["district"]
```

[81]:
```
0          19.00
1          12.00
2          18.00
3          18.00
4          19.00
            …
1865091    12.00
1865092    39.00
1865093    35.00
1865094    35.00
1865095    25.00
Name: district, Length: 1865096, dtype: float64
```

[82]:
```python
df["district"] = df["district"].astype("Int64")
```

[83]:
```python
district_data = df.groupby("district").size().reset_index(name = "count")
district_data.sort_values(by = "count", ascending = False)
```

[83]:

	district	count
16	24	161845
14	19	147454
9	14	139746
19	35	137265
20	39	134397
17	25	128258
13	18	123172
15	22	119692
8	12	117845
10	15	88697
12	17	76598
2	3	75871
1	2	69581
18	26	62838
11	16	59871

```
0          1   46452
7          9   41961
4          6   41665
6          8   36386
5          7   29690
3          5   21131
21        77    4680
```

[84]:

```
with open(r"GeoJson_Files\Boundaries_District.geojson") as f:
    district_geojson = geojson.load(f) # Philadelphia District GeoJson from␣
 ↪https://opendataphilly.org/datasets/police-districts/

map5 = folium.Map(location=[39.96, -75.15], zoom_start = 12, tiles =␣
 ↪"cartodbpositron") # Base map

district_data['district'] = district_data['district'].astype(str)

for feature in district_geojson['features']:
    district_id = str(feature['properties']['DIST_NUMC'])
    count_row = district_data[district_data['district'] == district_id]
    if not count_row.empty:
        feature['properties']['count'] = int(count_row['count'].values[0])
    else:
        feature['properties']['count'] = 0

folium.Choropleth(
    geo_data = district_geojson,
    data = district_data,
    columns = ['district', 'count'],
    key_on = 'feature.properties.DIST_NUMC',
    fill_opacity = 0.7,
    line_opacity = 0.2,
    legend_name = 'Stops by Police District',
    highlight = True,
    fill_color = "OrRd"
).add_to(map5)

tooltip = folium.GeoJson(
    district_geojson,
    style_function=lambda x: {'fillColor': 'transparent', 'color':␣
 ↪'transparent', 'weight': 0},
    tooltip=folium.GeoJsonTooltip(
        fields=['DIST_NUMC', 'count'],
        aliases=['District', 'Stops'],
        localize=True,
        sticky=True,
        labels=True,
```

```
        style=("background-color: white; color: #333333; font-family: Arial;␣
↪font-size: 12px; padding: 5px;")
    )
).add_to(map5)

map5
```

[84]: <folium.folium.Map at 0x1534b002f00>

[85]:
```
map5.save("HTML_Maps/stop_districts.html") # Save map as HTML
```

The spatial distribution of police stops in Philadelphia exhibits clear geographic patterns, with certain districts consistently showing higher levels of enforcement activity. Areas with elevated stop rates are not randomly distributed but appear concentrated in specific parts of the city, particularly in districts that encompass densely populated neighborhoods or those historically affected by social and economic challenges. These patterns suggest that police presence and activity may be influenced by localized conditions such as crime prevalence, drug-related issues, or community-policing priorities. In contrast, districts with relatively low stop rates are often found on the periphery or in less urbanized zones, possibly reflecting different demographic profiles or lower perceived need for intervention.

1.1.7 7. Service Area

[88]:
```
df["service_area"]
```

[88]:
```
0          191
1          121
2          183
3          183
4          193
         …
1865091    123
1865092    393
1865093    352
1865094    352
1865095    253
Name: service_area, Length: 1865096, dtype: object
```

[89]:
```
df["service_area"] = pd.to_numeric(df['service_area'], errors='coerce').
↪astype("Int64")
df["service_area"]
```

[89]:
```
0          191
1          121
2          183
3          183
4          193
```

```
              …
1865091    123
1865092    393
1865093    352
1865094    352
1865095    253
Name: service_area, Length: 1865096, dtype: Int64
```

[90]:
```
sa_data = df.groupby("service_area").size().reset_index(name = "count")
sa_data.sort_values(by = "count", ascending = False)
```

[90]:
```
    service_area   count
50           242  100444
43           192   89972
60           352   64189
27           141   62666
28           142   55469
..             …       …
9             52    7960
30           144    7165
1             12    6128
65          7700    4680
10            53    3469

[66 rows x 2 columns]
```

[91]:
```
with open(r"GeoJson_Files\Boundaries_PSA.geojson") as f:
    sa_geojson = geojson.load(f) # Philadelphia Service Areas GeoJson from␣
 ↪https://opendataphilly.org/datasets/police-service-areas/

map6 = folium.Map(location=[39.96, -75.15], zoom_start = 12,␣
 ↪tiles='cartodbpositron') # Base map

sa_data['service_area'] = sa_data['service_area'].astype(str)
sa_data["service_area"] = sa_data["service_area"].str.replace("7700", "77A")
sa_data["service_area"] = sa_data["service_area"].astype(str).str.zfill(3)

for feature in sa_geojson['features']:
    sa_id = str(feature['properties']['PSA_NUM'])
    count_row = sa_data[sa_data['service_area'] == sa_id]
    if not count_row.empty:
        feature['properties']['count'] = int(count_row['count'].values[0])
    else:
        feature['properties']['count'] = 0

folium.Choropleth(
    geo_data = sa_geojson,
```

```
    data = sa_data,
    columns = ['service_area', 'count'],
    key_on = 'feature.properties.PSA_NUM',
    fill_opacity = 0.7,
    line_opacity = 0.2,
    legend_name = 'Stops by Police Service Area',
    highlight = True,
    fill_color = "OrRd"
).add_to(map6)

tooltip = folium.GeoJson(
    sa_geojson,
    style_function=lambda x: {'fillColor': 'transparent', 'color':␣
 ↪'transparent', 'weight': 0},
    tooltip=folium.GeoJsonTooltip(
        fields=['PSA_NUM', 'count'],
        aliases=['Service Area', 'Stops'],
        localize=True,
        sticky=True,
        labels=True,
        style=("background-color: white; color: #333333; font-family: Arial;␣
 ↪font-size: 12px; padding: 5px;")
    )
).add_to(map6)

map6
```

[91]: <folium.folium.Map at 0x1534b04dbe0>

[92]:
```
map6.save("HTML_Maps/stop_service_areas.html") # Save map as HTML
```

Several service areas located in the central and eastern sections of the city appear to have the highest concentrations of stops, particularly where major transportation corridors, densely populated neighborhoods, and commercial activity converge. These areas often overlap with historically marginalized communities, suggesting a possible link between urban demographics, socioeconomic conditions, and patterns of police activity. Meanwhile, service areas in the far northwestern and southwestern edges of the city exhibit notably lower stop frequencies, possibly reflecting their more residential or suburban character, lower population density, or fewer patrolling routes

1.1.8 8. subject_age

[95]:
```
df["subject_age"]
```

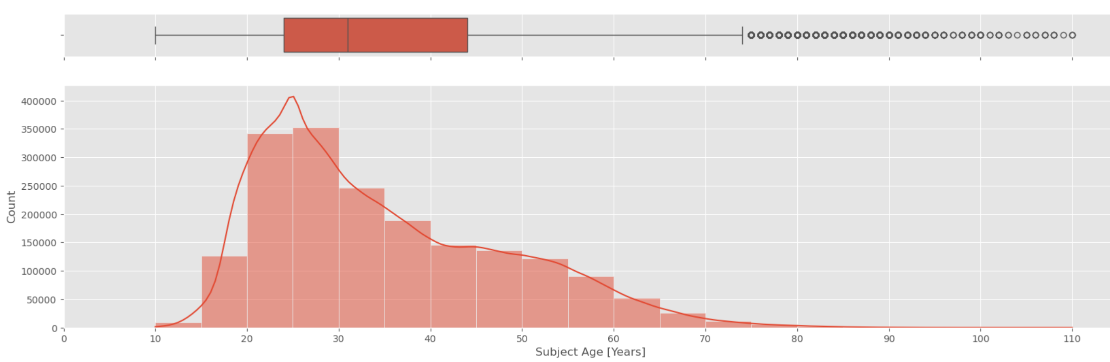
[95]:
```
0          31.00
1          21.00
2          24.00
3          20.00
```

```
4          31.00
             …
1865091    60.00
1865092    33.00
1865093    21.00
1865094    22.00
1865095    69.00
Name: subject_age, Length: 1865096, dtype: float64
```

[96]:
```
df["subject_age"] = df["subject_age"].astype("Int64")
df["subject_age"]
```

[96]:
```
0          31
1          21
2          24
3          20
4          31
           ..
1865091    60
1865092    33
1865093    21
1865094    22
1865095    69
Name: subject_age, Length: 1865096, dtype: Int64
```

[97]:
```
f, (ax_box, ax_hist) = plt.subplots(2, sharex=True,␣
 ↪gridspec_kw={"height_ratios": (.15, .85)})
sns.boxplot(df["subject_age"], orient = "h", ax = ax_box)
sns.histplot(data = df, x="subject_age", bins = 20, ax = ax_hist, kde = True)
plt.xticks(np.arange(0, df["subject_age"].max() + 10, 10))
plt.xlabel("Subject Age [Years]")
plt.show()
```



[98]:
```
df["subject_age"].describe()
```

[98]:
```
count    1860537.00
mean          34.83
std           13.34
min           10.00
25%           24.00
50%           31.00
75%           44.00
max          110.00
Name: subject_age, dtype: Float64
```

The age distribution of individuals subjected to stops reveals a strongly right-skewed pattern, with a clear concentration in early adulthood. The histogram and density curve indicate that most stops occur among younger individuals, with the frequency gradually decreasing as age increases. This trend suggests that police stops disproportionately affect people in their late teens through their forties, tapering off significantly for older adults. The boxplot confirms this skewness and also reveals the presence of outliers at the upper end of the age range. These patterns may reflect law enforcement priorities focused on age groups statistically more likely to be involved in public activity or criminalized behaviors, though it also raises questions about potential age-related profiling and the need to examine how justifiable these stop patterns are in relation to actual risk or threat.

1.1.9 9. subject_race

[101]:
```
df["subject_race"]
```

[101]:
```
0                            black
1                            black
2                            black
3                            black
4                            black
                    …
1865091                      black
1865092    asian/pacific islander
1865093                      black
1865094                      black
1865095                      black
Name: subject_race, Length: 1865096, dtype: object
```

[102]:
```
df["subject_race"].unique()
```

[102]:
```
array(['black', 'white', 'hispanic', 'unknown', 'asian/pacific islander',
       'other'], dtype=object)
```

[103]:
```
df["subject_race"].value_counts()
```

[103]:
```
subject_race
black                     1244249
white                      375862
```

```
hispanic                   184184
asian/pacific islander      40245
unknown                     14958
other                        5598
Name: count, dtype: int64
```

[104]:
```
df["subject_race"] = df["subject_race"].replace({"asian/pacific islander" :␣
↪"asian", "unknown" : "other"})
df["subject_race"].value_counts()
```

[104]:
```
subject_race
black       1244249
white        375862
hispanic     184184
asian         40245
other         20556
Name: count, dtype: int64
```

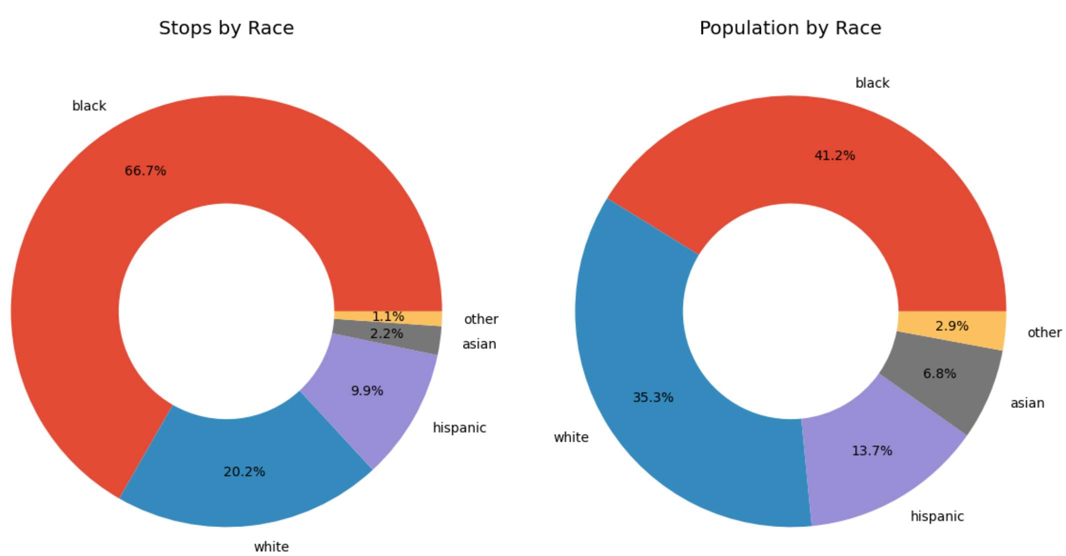
[105]:
```
df["subject_race"].value_counts(normalize = True).to_frame()
```

[105]:
```
              proportion
subject_race
black               0.67
white               0.20
hispanic            0.10
asian               0.02
other               0.01
```

[106]:
```
# Set Order
race_order = df["subject_race"].value_counts().index
df["subject_race"] = pd.Categorical(df["subject_race"], categories =␣
↪race_order, ordered = True)
```

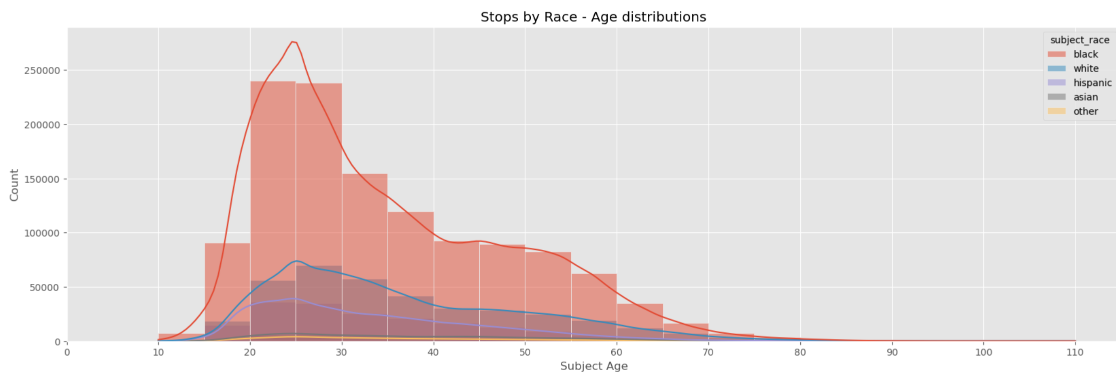
[107]:
```
# Race Population in Philadelphia

population = {
    'black': 41.22,
    'white': 35.34,
    'hispanic': 13.68,
    'asian': 6.84,
    'other': 2.92
}

# Sources
#https://en.wikipedia.org/wiki/Demographics_of_Philadelphia
# https://www.census.gov/quickfacts/fact/table/philadelphiacountypennsylvania/
↪AGE775223
```

[108]:
```
fig, axes = plt.subplots(1, 2, figsize=(12, 6))

wedges, texts, autotexts = axes[0].pie(
    df["subject_race"].value_counts().values,
    labels = df["subject_race"].value_counts().index,
    autopct = '%1.1f%%',    # Mostrar porcentaje
    wedgeprops = {'width': 0.5},
    labeldistance = 1.1,
    pctdistance = 0.75
)

axes[0].set_title("Stops by Race")

wedges2, texts2, autotexts2 = axes[1].pie(
    population.values(),
    labels=population.keys(),
    autopct='%1.1f%%',
    wedgeprops={'width': 0.5},
    labeldistance=1.1,
    pctdistance=0.75
)
axes[1].set_title("Population by Race")

plt.tight_layout()
plt.show()
```



The comparison between the racial distribution of police stops and the general population in Philadelphia reveals a pronounced disproportionality. One racial group (blacks), in particular, is stopped at a significantly higher rate relative to its share of the city's population, while others—especially white, Asian, and Hispanic individuals—are underrepresented in stop statistics compared to their population proportions. This disparity suggests that policing practices may not align with the demographic makeup of the city and raises important questions about potential racial bias or profiling.

[110]:
```
sns.histplot(data = df, x = "subject_age", hue = "subject_race", bins = 20,␣
  ↪alpha = 0.5, kde = True)
plt.xticks(np.arange(0, df["subject_age"].max() + 10, 10))
plt.xlabel("Subject Age")
plt.title("Stops by Race - Age distributions")
plt.show()
```



[111]:
```
df.groupby("subject_race")["subject_age"].describe()
```

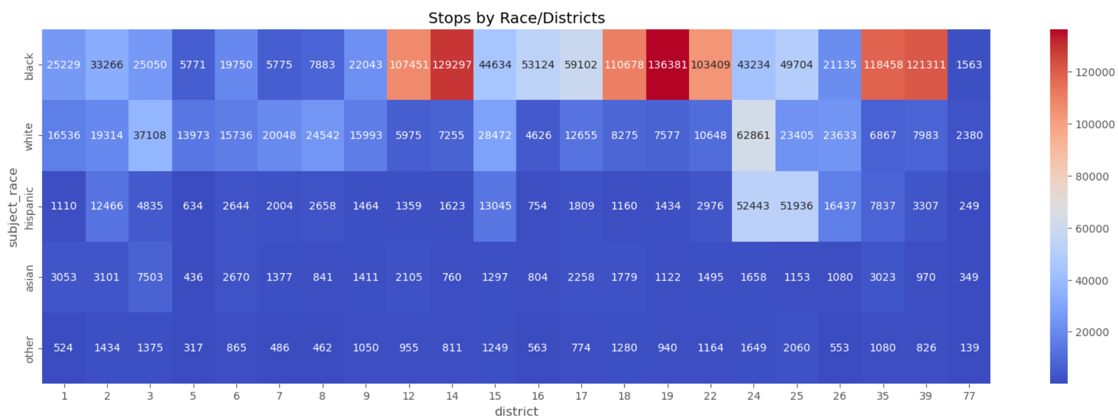
```
C:\Users\acast\AppData\Local\Temp\ipykernel_30128\1458744204.py:1:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
  df.groupby("subject_race")["subject_age"].describe()
```

[111]:

subject_race	count	mean	std	min	25%	50%	75%	max
black	1241512.00	34.54	13.40	10.00	24.00	31.00	44.00	110.00
white	374836.00	36.33	13.57	10.00	26.00	33.00	45.00	107.00
hispanic	183625.00	33.14	11.99	10.00	24.00	30.00	41.00	109.00
asian	40123.00	37.25	13.71	10.00	26.00	35.00	47.00	110.00
other	20441.00	35.33	12.87	10.00	25.00	33.00	44.00	105.00

The distribution of stops by age and race shows a consistent concentration of police interventions among younger individuals across all racial groups, with the majority of stops occurring before middle age. However, the magnitude and shape of the distribution vary notably by race. Black individuals not only account for the highest number of stops overall but also display a sharper concentration in early adulthood, suggesting more intense policing in younger age brackets within this group. While white and Hispanic individuals also experience the highest stop rates in similar age ranges, their distributions are broader and less skewed. Asian and other racial groups show significantly lower volumes of stops, though the age pattern remains similar. These trends reflect not only age-based targeting by law enforcement but also racial disparities in how these age-based strategies are applied, reinforcing concerns about systemic bias and highlighting the intersection between race and age in stop-and-frisk dynamics.

[113]:
```
pivot_month_hour = df.pivot_table(index = "subject_race",
                                  columns = "district",
                                  aggfunc = "size",
                                  observed = False)

sns.heatmap(pivot_month_hour, cmap = "coolwarm", annot = True, fmt='g')
plt.title("Stops by Race/Districts")
plt.show()
```



Districts with a higher concentration of Black residents—particularly in central and southwestern sections of the city—correspond to those with the most stops, suggesting that racial demographics may be a major factor influencing enforcement patterns. While some high-stop districts also have dense, diverse populations, the disproportionate number of stops among Black individuals across nearly all districts stands out. Hispanic populations appear to experience elevated stop levels in specific areas but not as pervasively as Black individuals. Meanwhile, white and Asian individuals are more likely to be stopped in districts where their demographic presence is strongest, though the overall stop levels for these groups remain comparatively lower. These patterns reflect a racialized geography of policing, where the distribution of law enforcement interventions aligns not only with population density but also with longstanding racial divides across the city, pointing to systemic differences in how communities experience public safety efforts.

[116]:
```
# stratified sample

df_lat_lng = df.dropna(subset = ["lat", "lng"])
n_total = 0.01 * len(df_lat_lng)

proportion = df_lat_lng["subject_race"].value_counts(normalize = True)
```

```
n_by_race = (proportion * n_total).round().astype(int)

df_sample = df_lat_lng.groupby('subject_race', group_keys = False).apply(lambda␣
↪x: x.sample(n = n_by_race[x.name], random_state = 42))
```

```
C:\Users\acast\AppData\Local\Temp\ipykernel_30128\1405218886.py:13:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
  df_sample = df_lat_lng.groupby('subject_race', group_keys =
False).apply(lambda x: x.sample(n = n_by_race[x.name], random_state = 42))
C:\Users\acast\AppData\Local\Temp\ipykernel_30128\1405218886.py:13:
DeprecationWarning: DataFrameGroupBy.apply operated on the grouping columns.
This behavior is deprecated, and in a future version of pandas the grouping
columns will be excluded from the operation. Either pass `include_groups=False`
to exclude the groupings or explicitly select the grouping columns after groupby
to silence this warning.
  df_sample = df_lat_lng.groupby('subject_race', group_keys =
False).apply(lambda x: x.sample(n = n_by_race[x.name], random_state = 42))
```

[117]:
```
df_sample["subject_race"].value_counts(normalize = True).to_frame()
```

[117]:
```
              proportion
subject_race
black               0.66
white               0.20
hispanic            0.10
asian               0.02
other               0.01
```

[118]:
```
df_sample.shape
```

[118]: (17604, 26)

[119]:
```
race_colors = {
    'white': '#e41a1c',
    'black': '#377eb8',
    'hispanic': '#ff7f00',
    'asian': '#4daf4a',
    'other': '#ffff33'
}

map7 = folium.Map(location=[39.95, -75.16], zoom_start = 11, tiles =␣
↪"cartodbpositron")

for _, row in df_sample.iterrows():
```

```python
    race = row['subject_race']
    color = race_colors.get(race, 'gray')

    folium.CircleMarker(
        location=[row['lat'], row['lng']],
        radius=2,
        color=color,
        fill=True,
        fill_color=color,
        fill_opacity=0.7,
        weight=0
    ).add_to(map7)


legend_html = """
<div style="
    position: fixed;
    bottom: 30px;
    left: 30px;
    width: 250px;
    height: 160px;
    background-color: white;
    border:2px solid grey;
    z-index:9999;
    font-size:14px;
    padding:10px;
    box-shadow: 2px 2px 6px rgba(0,0,0,0.3);
">
<b>Legend: Subject Race Stops</b><br>
<i style="background: #e41a1c; width: 10px; height: 10px; float: left; margin-right: 6px; border-radius: 50%; display: inline-block"></i>White<br>
<i style="background: #377eb8; width: 10px; height: 10px; float: left; margin-right: 6px; border-radius: 50%; display: inline-block"></i>Black<br>
<i style="background: #ff7f00; width: 10px; height: 10px; float: left; margin-right: 6px; border-radius: 50%; display: inline-block"></i>Hispanic<br>
<i style="background: #4daf4a; width: 10px; height: 10px; float: left; margin-right: 6px; border-radius: 50%; display: inline-block"></i>Asian<br>
<i style="background: #ffff33; width: 10px; height: 10px; float: left; margin-right: 6px; border-radius: 50%; display: inline-block"></i>Other
</div>
"""

map7.get_root().html.add_child(Element(legend_html))

map7
```

[119]: <folium.folium.Map at 0x1536deea810>

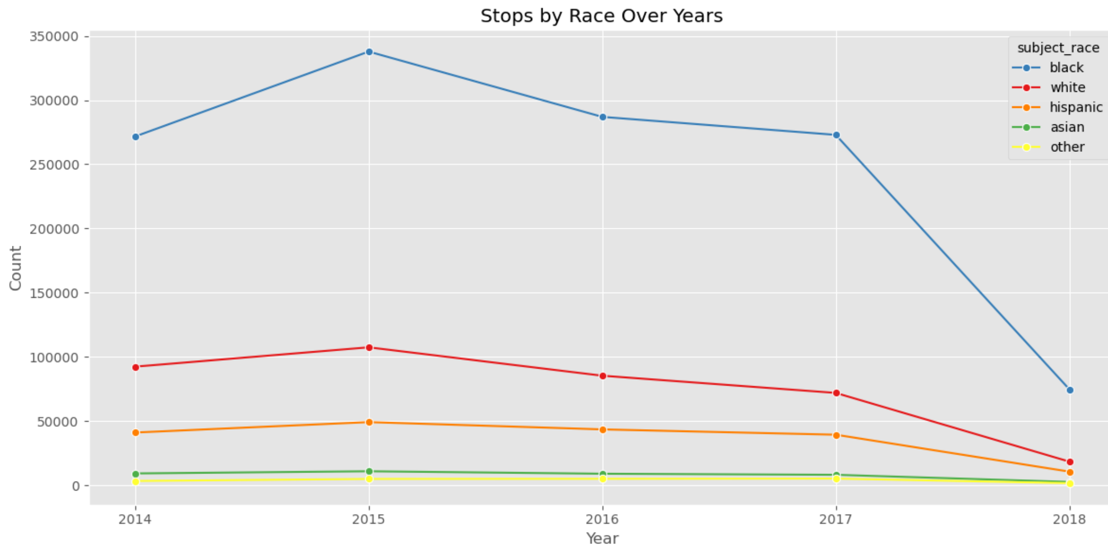
[121]:
```
map7.save("HTML_Maps/stop_race.html") # Save map as HTML
```

When comparing the map of police stops by race with the demographic distribution of Philadelphia's population, clear patterns emerge that highlight racial and spatial disparities in enforcement. The stop map shows a dense clustering of stops involving Black individuals in the western, southwestern, and central parts of the city—areas that align closely with neighborhoods where Black residents are most densely concentrated, as seen in the population map. Similarly, Hispanic stops are heavily concentrated in central and lower northeastern sections, also reflecting population distributions. However, the concentration of stops in these areas appears more intense than the proportional presence of these racial groups in the population, particularly for Black individuals, suggesting that population size alone does not fully explain the enforcement intensity.

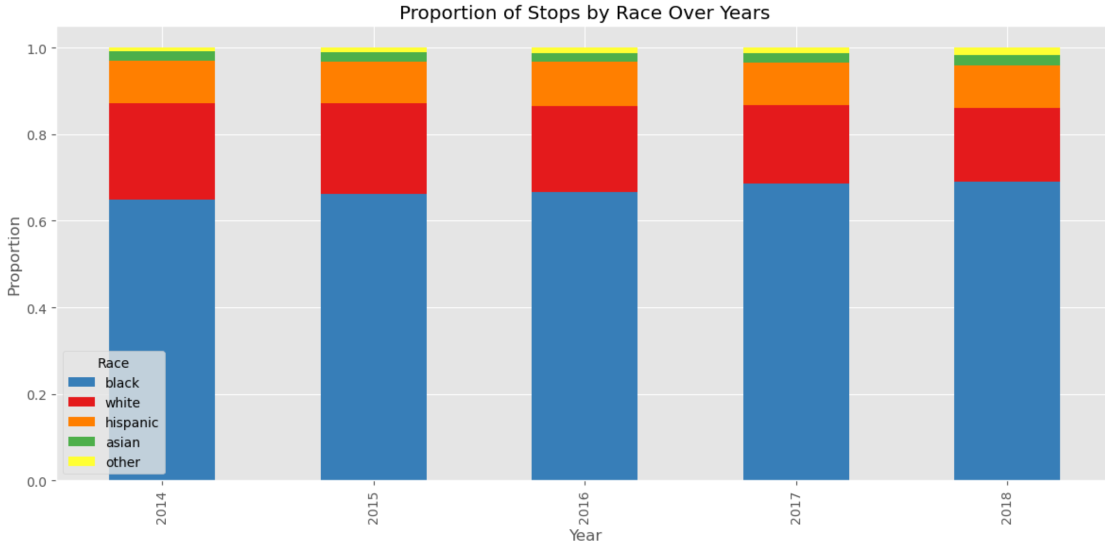
Conversely, majority-white areas in the far northeast and northwest show a much lighter footprint of stops, even though they are home to large white populations. This contrast indicates that stop practices may not align uniformly with demographic presence across the city. The spatial overlap between race and enforcement patterns reveals a tendency for policing strategies to be more aggressive in racially marginalized neighborhoods, which raises concerns about potential systemic bias and reinforces long-standing divisions in how different communities experience public safety and law enforcement.

[123]:
```
df_race_year = df.groupby("Year")["subject_race"].value_counts().
 ↪to_frame(name='count').reset_index()

race_colors = {
    'white': '#e41a1c',
    'black': '#377eb8',
    'hispanic': '#ff7f00',
    'asian': '#4daf4a',
    'other': '#ffff33'
}

# Crear el gráfico
plt.figure(figsize=(12, 6))
sns.lineplot(
    data=df_race_year,
    x="Year",
    y="count",
    hue="subject_race",
    palette=race_colors,
    marker="o"
)

plt.title("Stops by Race Over Years")
plt.ylabel("Count")
plt.xticks(year_data["Year"])
plt.grid(True)
```

```
plt.tight_layout()
plt.show()
```



[124]:
```
df_prop = (df.groupby("Year")["subject_race"].value_counts(normalize=True).
↪rename("proportion").reset_index())


df_wide = df_prop.pivot(index="Year", columns="subject_race",␣
↪values="proportion").fillna(0)

# Orden de razas (opcional)
ordered_races = ['black', 'white', 'hispanic', 'asian', 'other']

# Crear gráfico
ax = df_wide[ordered_races].plot(
    kind="bar",
    stacked=True,
    figsize=(12, 6),
    color=[race_colors[r] for r in race_order]
)

plt.title("Proportion of Stops by Race Over Years")
plt.ylabel("Proportion")
plt.xlabel("Year")
plt.legend(title="Race")
plt.tight_layout()
plt.show()
```



The overall trend in police stops over the years shows a decline across all racial groups, though the apparent drop in the final year is largely attributable to the dataset covering only a portion of 2018. Despite fluctuations in the absolute number of stops, the proportional distribution by race remains relatively stable, with some groups consistently overrepresented compared to others. This persistence in the racial makeup of stops suggests that underlying patterns of enforcement remained largely unchanged, even as overall activity varied year to year. The data highlight the resilience of systemic disparities in stop practices, pointing to the need for deeper structural evaluations beyond simple reductions in volume.

1.1.10 10. subject_sex

[127]:
```
df["subject_sex"].value_counts().to_frame()
```

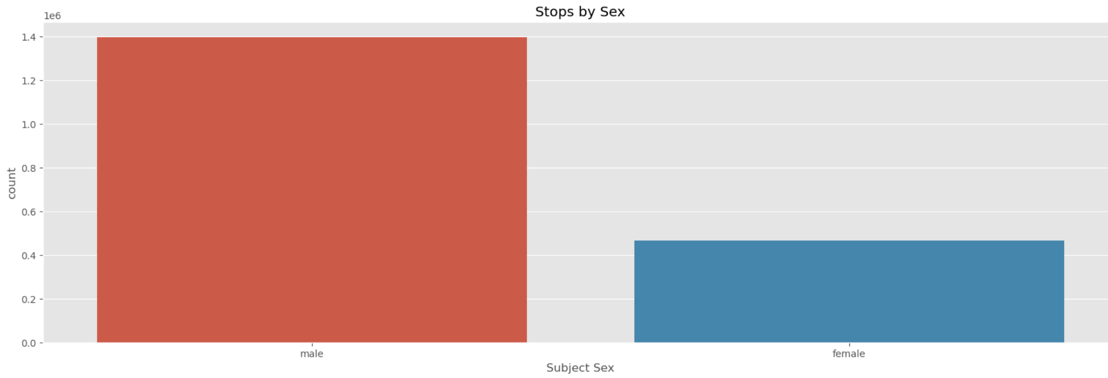
[127]:
```
               count
subject_sex
male         1397206
female        467240
```

[128]:
```
df["subject_sex"].value_counts(normalize = True).to_frame()
```

[128]:
```
             proportion
subject_sex
male               0.75
female             0.25
```

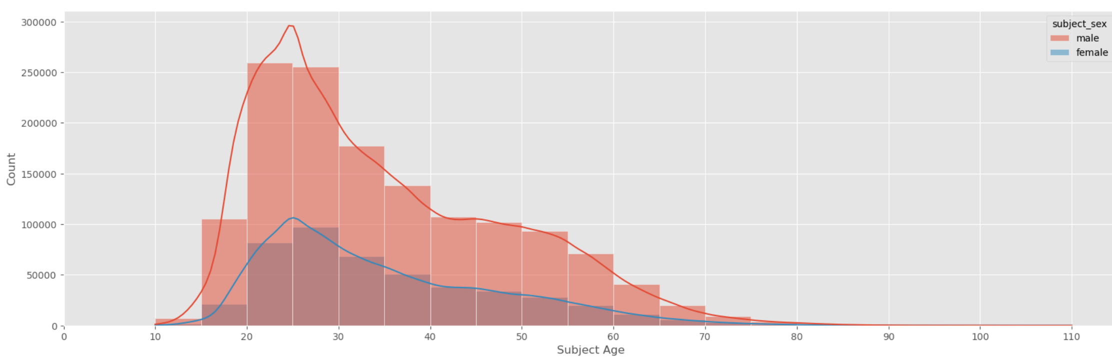
[129]:
```
sns.countplot(data = df, x = "subject_sex", hue = "subject_sex")
plt.legend().remove()
plt.xlabel("Subject Sex")
plt.title("Stops by Sex")
```

```
plt.show()
```



The distribution of police stops by sex reveals a marked disparity, with male individuals being stopped far more frequently than female individuals. This imbalance likely reflects broader trends in law enforcement where men, particularly young men, are more often perceived as subjects of interest or risk in public safety operations.

[131]:
```
sns.histplot(data = df, x = "subject_age", hue = "subject_sex", bins = 20,␣
↪alpha = 0.5, kde = True)
plt.xticks(np.arange(0, df["subject_age"].max() + 10, 10))
plt.xlabel("Subject Age")
plt.show()
```



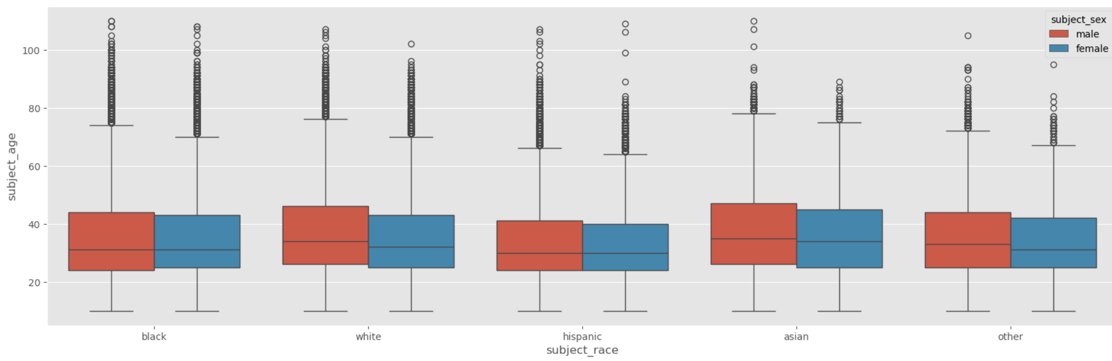
The age distribution of police stops by sex reveals similar overall patterns for both males and females, with the highest concentration occurring in early adulthood and a gradual decline as age increases. However, the volume of stops among males is substantially higher at every age, especially in the younger brackets, where the gap between sexes is most pronounced. This suggests that policing practices disproportionately affect young men, reinforcing the idea that gender and age intersect as key factors in enforcement dynamics. While the trends follow a similar shape for both sexes, the magnitude difference underscores a gendered experience in public-police interactions, particularly during the most active years of early life.

[133]:
```
df.groupby("subject_sex")["subject_age"].describe()
```

[133]:

subject_sex	count	mean	std	min	25%	50%	75%	max
female	466194.00	34.86	12.76	10.00	25.00	32.00	43.00	109.00
male	1393709.00	34.82	13.52	10.00	24.00	31.00	44.00	110.00

[134]:
```
sns.boxplot(data = df, x = "subject_race", y = "subject_age", hue =␣
↪"subject_sex")
plt.show()
```



Across all racial categories, the interquartile range for both sexes is relatively narrow, clustering around early to mid-adulthood, though male distributions often show lower medians and slightly more compressed lower bounds. The presence of numerous outliers at higher ages, particularly among females, suggests that while stops are heavily concentrated among younger individuals, older adults are occasionally subject to intervention as well. These patterns highlight a gendered and racial consistency in age-based stop patterns, with some variations that may reflect differences in perceived risk, behavior, or visibility across intersections of identity.

[136]:
```
df.groupby("subject_race")["subject_sex"].value_counts(normalize = True).
↪to_frame()
```

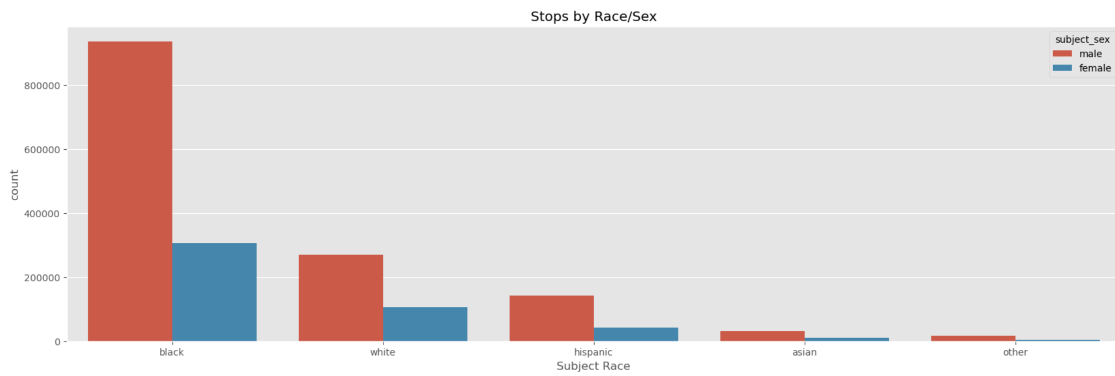
```
C:\Users\acast\AppData\Local\Temp\ipykernel_30128\3136352317.py:1:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
  df.groupby("subject_race")["subject_sex"].value_counts(normalize =
True).to_frame()
```

[136]:

subject_race	subject_sex	proportion
black	male	0.75
	female	0.25
white	male	0.72

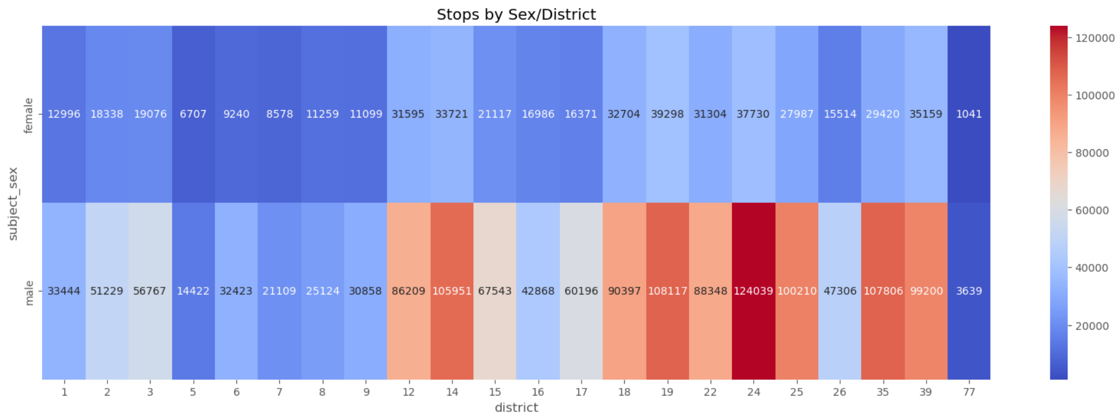
```
              female            0.28
hispanic      male              0.77
              female            0.23
asian         male              0.76
              female            0.24
other         male              0.82
              female            0.18
```

[137]:
```
sns.countplot(data = df, x = "subject_race", hue = "subject_sex")
plt.xlabel("Subject Race")
plt.title("Stops by Race/Sex")
plt.show()
```



The analysis of stops by both race and sex reveals a consistent pattern of gender disparity across all racial groups, with males representing the vast majority of individuals stopped. The consistent proportions suggest that gender plays a dominant role in enforcement decisions regardless of racial background. While racial disparities in stop counts are clear, the addition of sex-based proportions highlights a layered dynamic in which men of all races—especially men of color—face a disproportionately high level of police scrutiny.

[139]:
```
pivot_month_hour = df.pivot_table(index = "subject_sex",
                                  columns = "district",
                                  aggfunc = "size",
                                  observed = False)

sns.heatmap(pivot_month_hour, cmap = "coolwarm", annot = True, fmt='g')
plt.title("Stops by Sex/District")
plt.show()
```



Males are stopped significantly more often than females in every district. While the absolute number of stops varies by district—particularly concentrated in some high-activity areas—the gender disparity remains constant, indicating that the gap is not simply due to district-specific dynamics but is instead a widespread feature of policing practices. Districts with the highest overall stop counts, such as those in the central and eastern parts of the city, amplify this disparity further, reinforcing the notion that sex-based differences in stops are both structurally embedded and spatially widespread throughout Philadelphia.

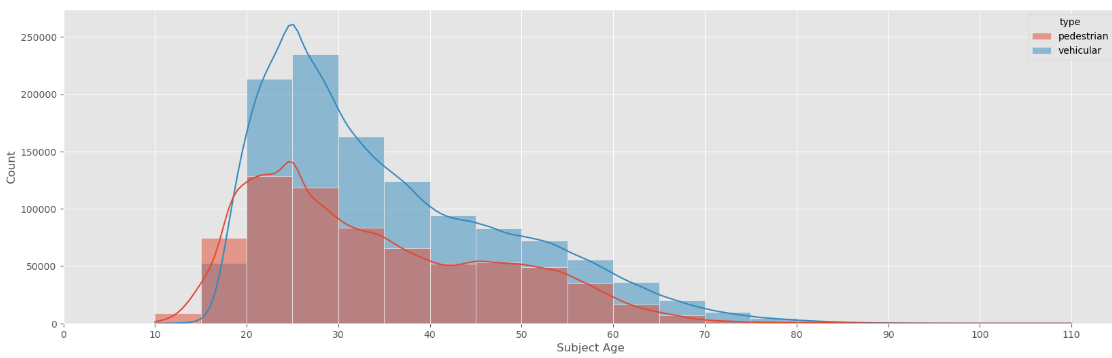
1.1.11 11. type

[142]:
```
# Function for complete analysis
def age_sex_race(column):

    # Propotion
    display(df[column].value_counts(normalize = True).to_frame())

    # Age distribution
    sns.histplot(data = df, x = "subject_age", hue = column, bins = 20, alpha =␣
 ↪0.5, kde = True)
    plt.xticks(np.arange(0, df["subject_age"].max() + 10, 10))
    plt.xlabel("Subject Age")
    plt.show()
    display(df.groupby(column)["subject_age"].describe())

    # Sex distribution
    sns.countplot(data = df, x = "subject_sex", hue = column)
    plt.show()
    display(df.groupby("subject_sex")[column].value_counts(normalize = True).
 ↪to_frame())

    # Race Distribution
    sns.countplot(data = df, x = "subject_race", hue = column)
```

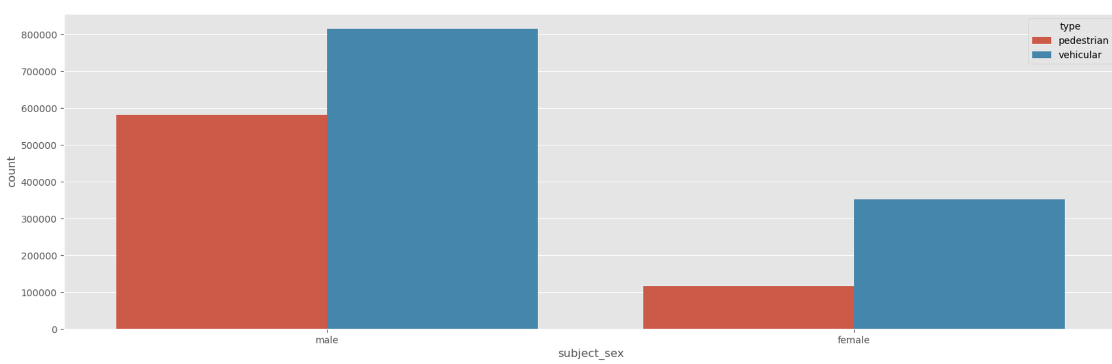
```
    plt.show()
    display(df.groupby("subject_race")[column].value_counts(normalize = True).
↪to_frame())

    # District Distribution
    pivot_district = df.pivot_table(index = column,
                                    columns = "district",
                                    aggfunc = "size",
                                    observed = False)

    sns.heatmap(pivot_district, cmap = "coolwarm", annot = True, fmt='g')
    plt.show()
```

[143]:
```
age_sex_race("type")
```

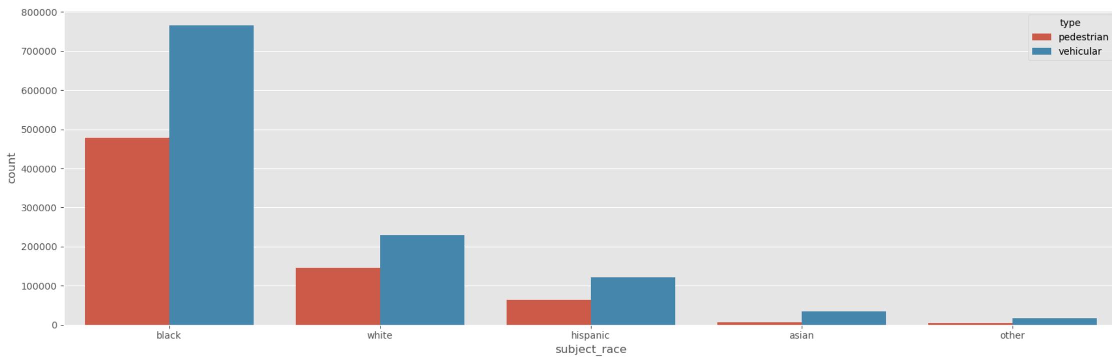
```
            proportion
type
vehicular         0.63
pedestrian        0.37
```



	count	mean	std	min	25%	50%	75%	max
type								
pedestrian	695233.00	33.74	13.29	10.00	23.00	30.00	44.00	110.00
vehicular	1165304.00	35.48	13.32	10.00	25.00	32.00	44.00	110.00

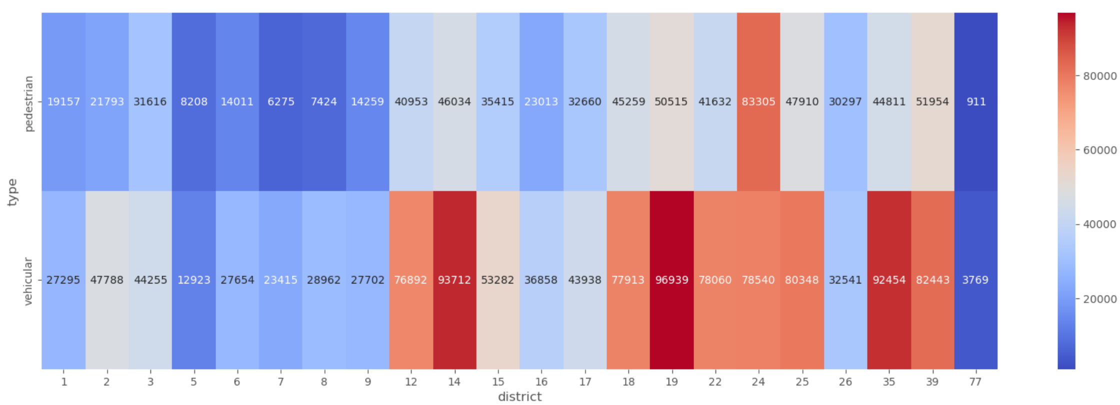


```
                        proportion
subject_sex type
female      vehicular         0.75
            pedestrian        0.25
male        vehicular         0.58
            pedestrian        0.42
```




```
C:\Users\acast\AppData\Local\Temp\ipykernel_30128\2111065542.py:22:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
  display(df.groupby("subject_race")[column].value_counts(normalize =
True).to_frame())
```

```
                         proportion
subject_race type
black        vehicular         0.62
             pedestrian        0.38
white        vehicular         0.61
             pedestrian        0.39
hispanic     vehicular         0.66
             pedestrian        0.34
asian        vehicular         0.83
             pedestrian        0.17
other        vehicular         0.81
             pedestrian        0.19
```



Vehicular stops are significantly more common than pedestrian stops overall, with the proportion of vehicular stops being even higher among women and among individuals identified as Asian or from other racial backgrounds. In contrast, pedestrian stops represent a larger share of the enforcement activity directed at males and at Black and Hispanic individuals, suggesting a racialized and gendered dimension to the mode of enforcement. Younger individuals are more frequently involved in pedestrian stops, while vehicular stops tend to span a slightly older age range. Spatially, the most active districts show high counts of both types, but vehicular stops dominate across almost all districts. These patterns indicate that the method of stop is not uniformly applied but rather intersects with demographic characteristics in ways that may reflect underlying biases or targeted enforcement strategies.

1.1.12 12. arrest_made

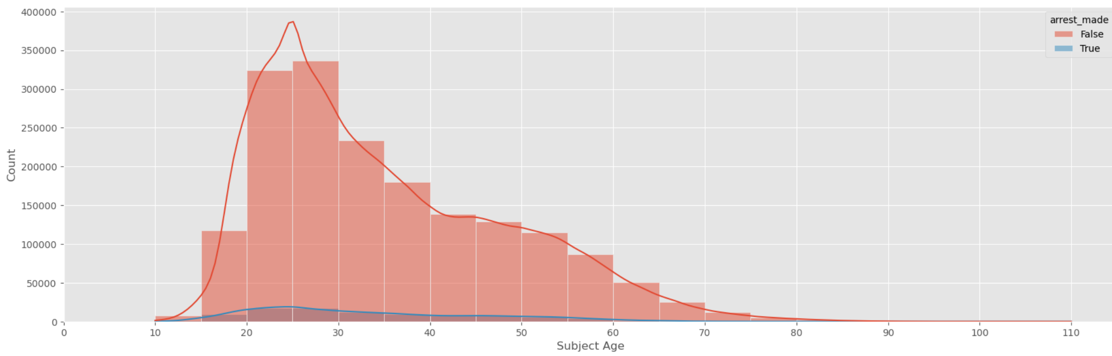
[146]:
```
df["arrest_made"].value_counts().to_frame()
```

[146]:

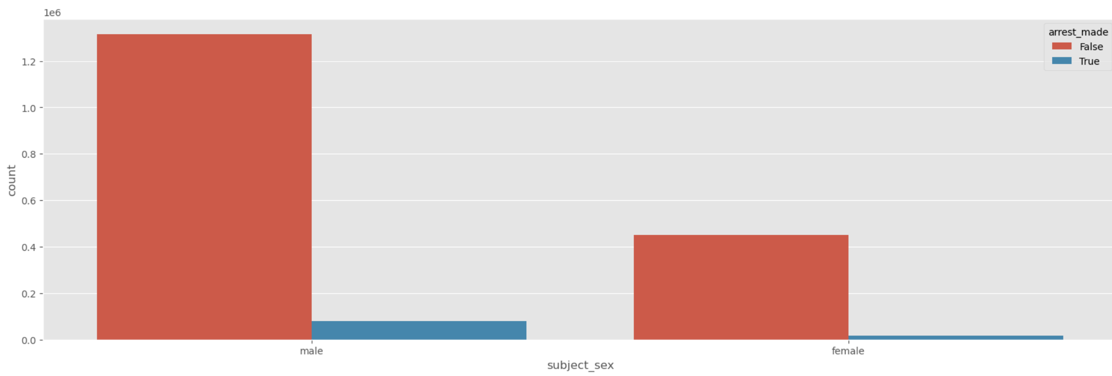
	count
arrest_made	
False	1769620
True	95476

[147]:
```
age_sex_race("arrest_made")
```

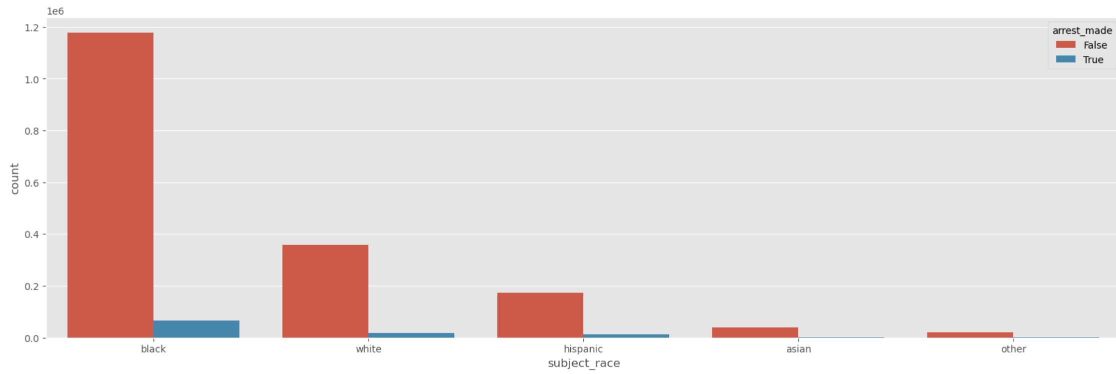
	proportion
arrest_made	
False	0.95
True	0.05



	count	mean	std	min	25%	50%	75%	max
arrest_made								
False	1765366.00	34.91	13.37	10.00	24.00	31.00	44.00	110.00
True	95171.00	33.37	12.57	10.00	23.00	30.00	42.00	105.00

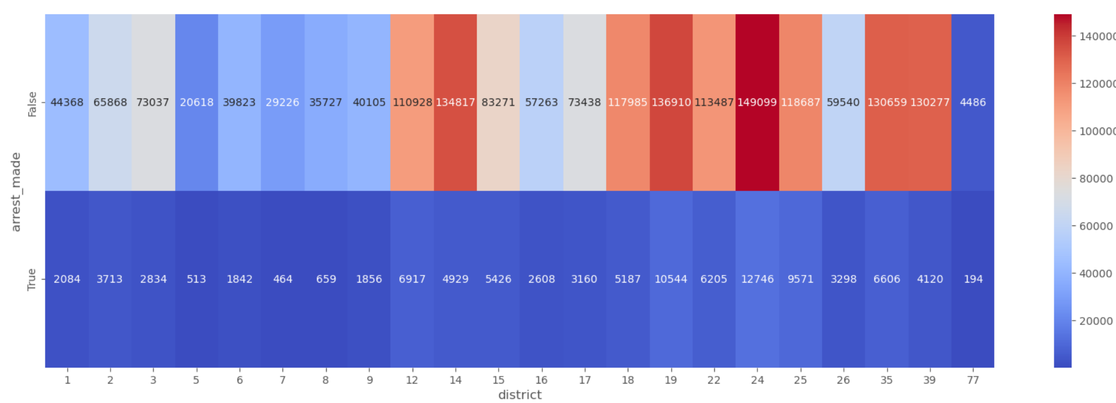


		proportion
subject_sex	arrest_made	
female	False	0.97
	True	0.03
male	False	0.94
	True	0.06




```
C:\Users\acast\AppData\Local\Temp\ipykernel_30128\2111065542.py:22:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
  display(df.groupby("subject_race")[column].value_counts(normalize =
True).to_frame())
```

subject_race	arrest_made	proportion
black	False	0.95
	True	0.05
white	False	0.95
	True	0.05
hispanic	False	0.94
	True	0.06
asian	False	0.98
	True	0.02
other	False	0.98
	True	0.02



Arrests are most concentrated among young adults, reflecting a common pattern where individuals in early adulthood are more frequently subjected to enforcement action with legal consequences. Men are more likely to be arrested than women, and this disparity is consistent across all racial groups. While the overall proportions of arrests are similar by race, the absolute volume is significantly higher among Black individuals, mirroring broader trends in stop data. Spatially, the districts with the highest overall stop activity also show the highest number of arrests, though the arrest rate itself remains low. These patterns suggest that while most stops do not result in arrests, the enforcement burden—particularly in terms of exposure to arrest—is not evenly distributed across the population.

1.1.13 13. outcome

[150]:
```
df["outcome"].value_counts()
```

[150]:
```
outcome
arrest    95476
Name: count, dtype: int64
```

Is the same information as *arrest_made* column, therefore it will be deleted

[152]:
```
df.drop(columns = ["outcome"], inplace = True)
```

1.1.14 14. contraband_found

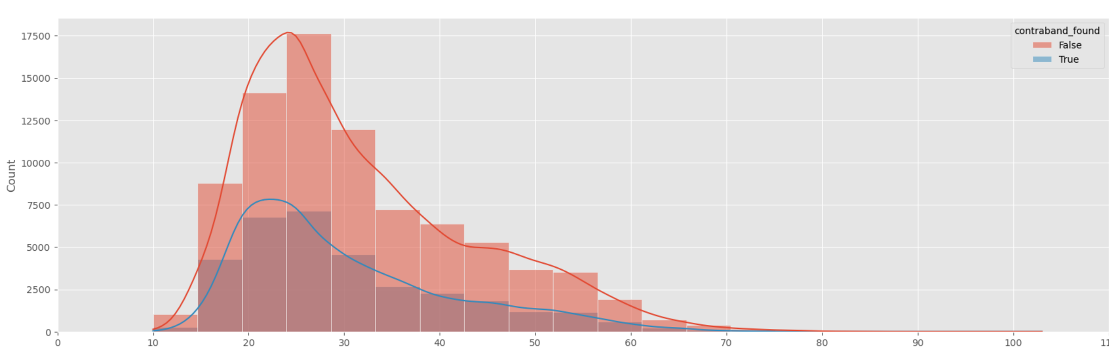
[154]:
```
df["contraband_found"].value_counts().to_frame()
```

[154]:

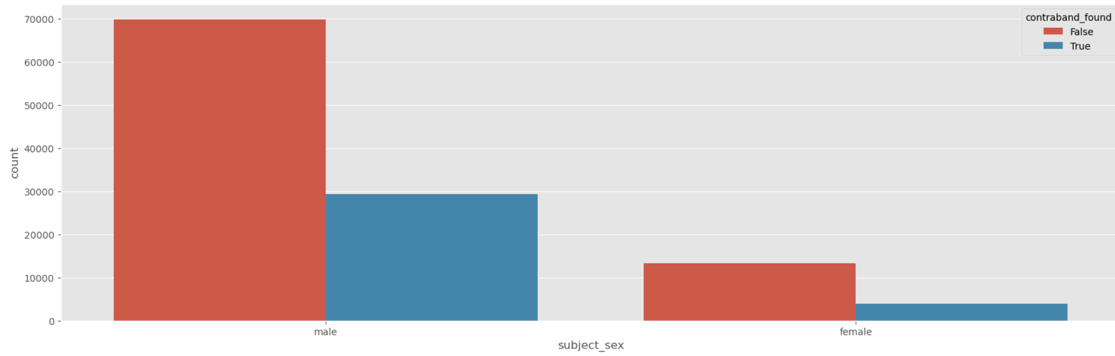
contraband_found	count
False	83225
True	33230

[155]:
```
age_sex_race("contraband_found")
```

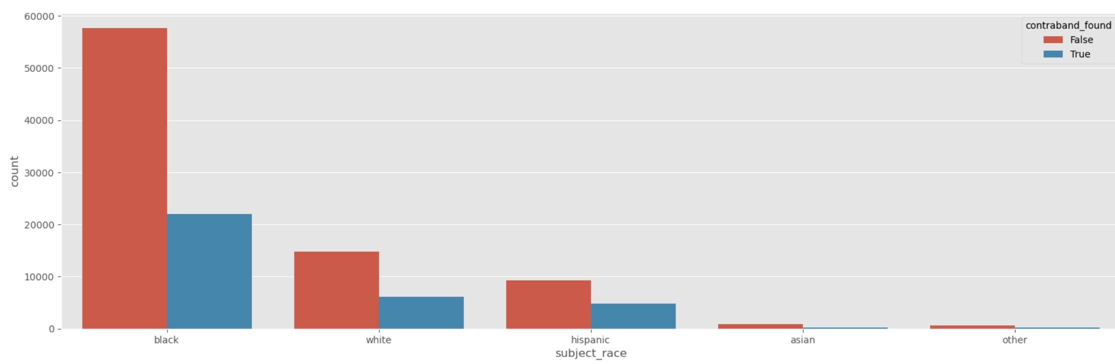
contraband_found	proportion
False	0.71
True	0.29



```
                  count  mean   std   min   25%   50%   75%    max
contraband_found
False          82968.00 31.60 11.87 10.00 23.00 28.00 38.00 103.00
True           33115.00 30.13 11.22 10.00 22.00 27.00 36.00 103.00
```

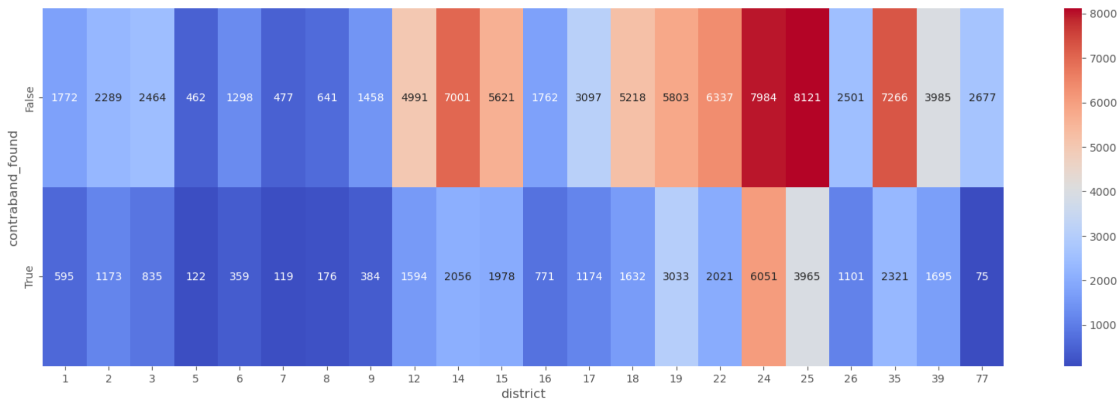



```
                              proportion
subject_sex contraband_found
female      False                   0.78
            True                    0.22
male        False                   0.70
            True                    0.30
```




```
C:\Users\acast\AppData\Local\Temp\ipykernel_30128\2111065542.py:22:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
  display(df.groupby("subject_race")[column].value_counts(normalize =
True).to_frame())
                               proportion
```

```
subject_race contraband_found
black        False               0.72
             True                0.28
white        False               0.71
             True                0.29
hispanic     False               0.66
             True                0.34
asian        False               0.78
             True                0.22
other        False               0.79
             True                0.21
```



Contraband is more frequently found among younger adults, particularly those in their twenties, and the rate of discovery declines steadily with age. Males are more likely to be involved in stops where contraband is found compared to females, and this difference is consistent across racial groups. Among racial categories, Hispanic individuals exhibit the highest relative rate of contraband discovery, though absolute counts are highest among Black individuals due to their higher stop volume. At the district level, areas with the most stop activity also show the highest counts of contraband discoveries, suggesting a relationship between policing intensity and these outcomes.

1.1.15 15. frisk_performed

[158]:
```
df["frisk_performed"].value_counts()
```

[158]:
```
frisk_performed
False    1698212
True      166884
Name: count, dtype: int64
```

same infomation

1.1.16 16. search_conducted

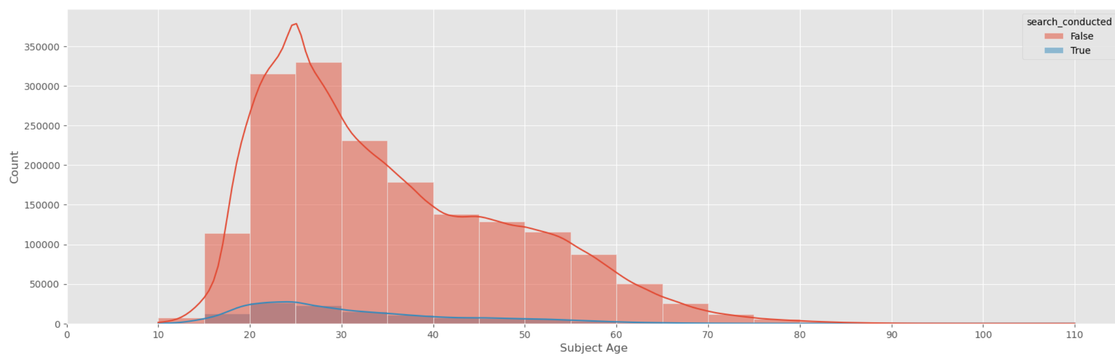
[161]:
```
df["search_conducted"].value_counts().to_frame()
```

[161]:

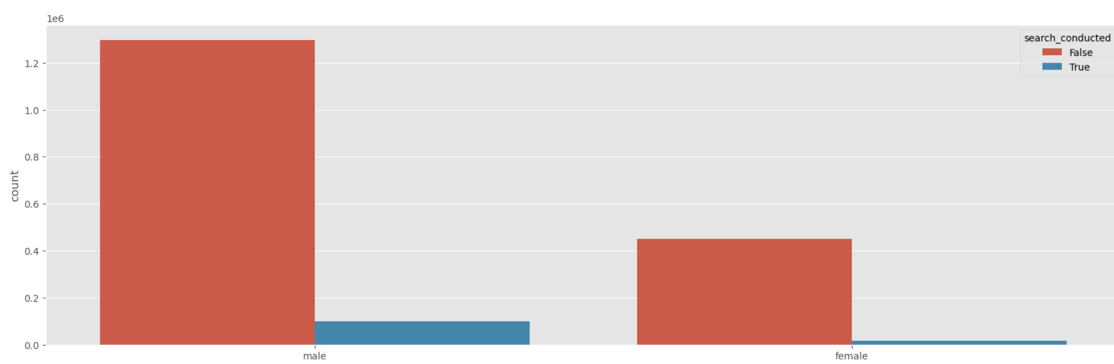
	count
search_conducted	
False	1748641
True	116455

[162]:
```
age_sex_race("search_conducted")
```

	proportion
search_conducted	
False	0.94
True	0.06

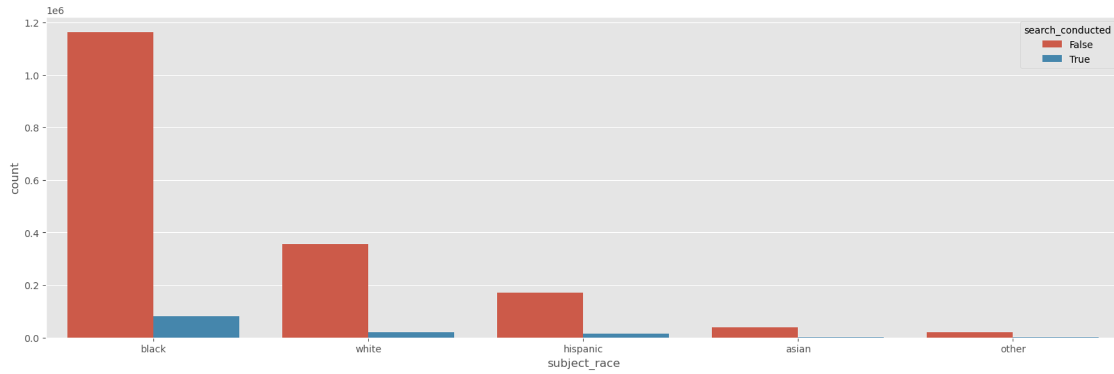


	count	mean	std	min	25%	50%	75%	max
search_conducted								
False	1744454.00	35.07	13.40	10.00	24.00	32.00	44.00	110.00
True	116083.00	31.18	11.71	10.00	22.00	28.00	38.00	103.00



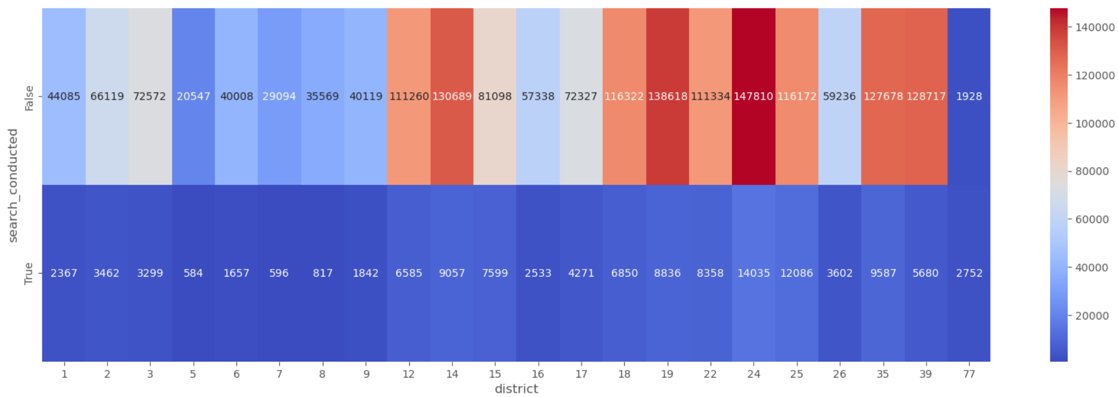
proportion

```
subject_sex search_conducted
female      False                   0.96
            True                    0.04
male        False                   0.93
            True                    0.07
```




```
C:\Users\acast\AppData\Local\Temp\ipykernel_30128\2111065542.py:22:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
  display(df.groupby("subject_race")[column].value_counts(normalize =
True).to_frame())
```

```
                              proportion
subject_race search_conducted
black        False                  0.94
             True                   0.06
white        False                  0.94
             True                   0.06
hispanic     False                  0.92
             True                   0.08
asian        False                  0.97
             True                   0.03
other        False                  0.96
             True                   0.04
```



Searches conducted during stops are relatively rare overall. Younger individuals are slightly more likely to be searched, with the age distribution peaking in early adulthood and tapering off in older age groups. Males are subjected to searches at a notably higher rate than females, both in absolute terms and proportional to stops. Racially, the likelihood of being searched is fairly consistent across major groups, though Hispanic individuals exhibit a slightly higher proportion of searches relative to stops. Spatially, District 24 leads in the number of searches conducted—reflecting its broader trend of intense policing—likely influenced by its high population density, elevated levels of public drug activity, and its location at the core of the city's opioid crisis. Despite differences in volume, the proportion of searches remains low throughout all districts, raising questions about the frequency, justification, and effectiveness of search practices as a policing tool.

1.1.17 17. search_person

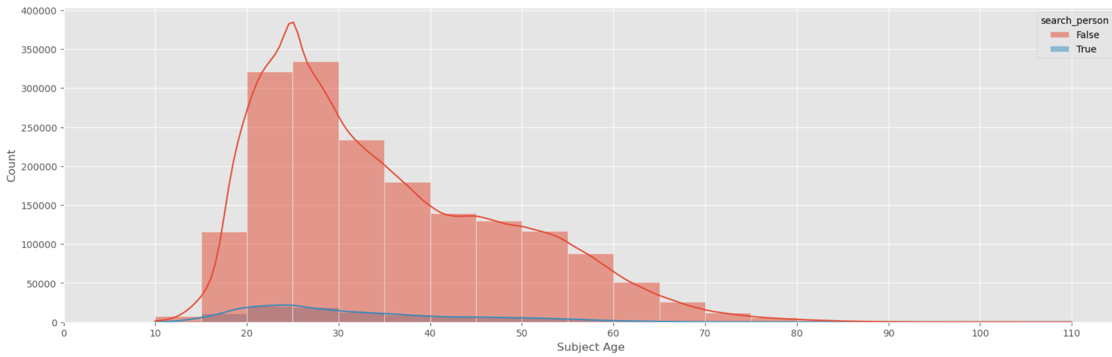
[165]:
```
df["search_person"].value_counts().to_frame()
```

[165]:

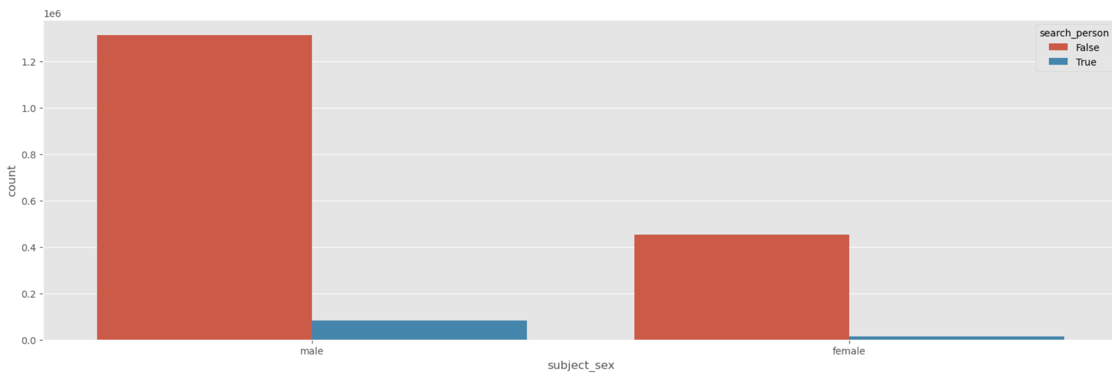
	count
search_person	
False	1768506
True	96590

[166]:
```
age_sex_race("search_person")
```

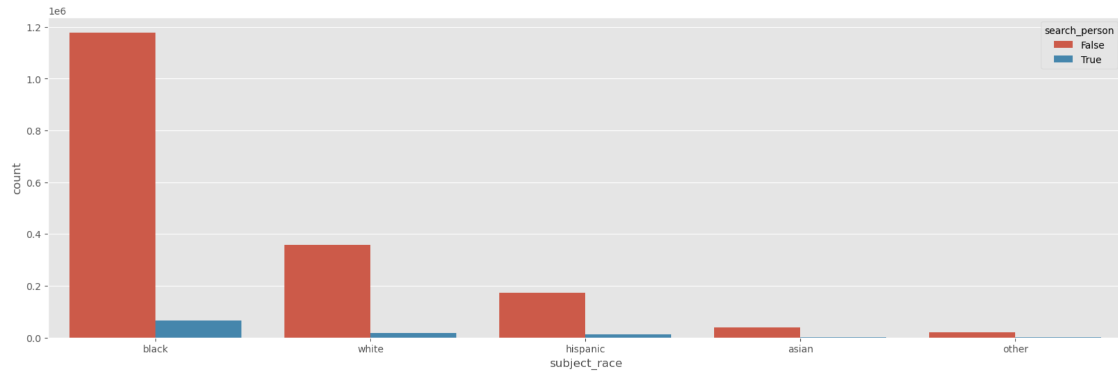
	proportion
search_person	
False	0.95
True	0.05



	count	mean	std	min	25%	50%	75%	max
search_person								
False	1764267.00	35.02	13.39	10.00	24.00	32.00	44.00	110.00
True	96270.00	31.40	11.76	10.00	22.00	28.00	38.00	103.00

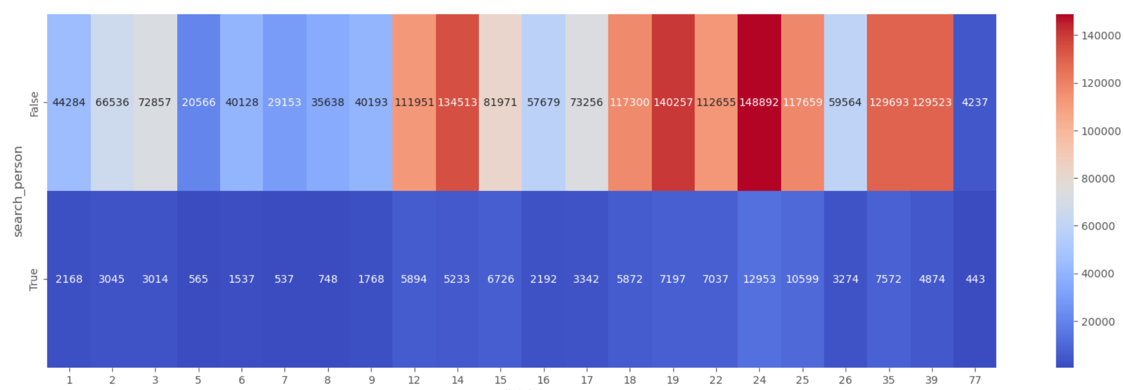


		proportion
subject_sex	search_person	
female	False	0.97
	True	0.03
male	False	0.94
	True	0.06




```
C:\Users\acast\AppData\Local\Temp\ipykernel_30128\2111065542.py:22:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
  display(df.groupby("subject_race")[column].value_counts(normalize =
True).to_frame())
```

```
                            proportion
subject_race search_person
black        False                0.95
             True                 0.05
white        False                0.95
             True                 0.05
hispanic     False                0.93
             True                 0.07
asian        False                0.98
             True                 0.02
other        False                0.97
             True                 0.03
```



Younger individuals are more likely to be subjected to searches, with frequencies peaking in early adulthood and decreasing with age. Males are significantly more likely to be searched than females, and this disparity is consistent across all racial groups. Among racial categories, Hispanic individuals have a slightly higher proportion of searches relative to stops. District 24 again stands out with the highest number of person searches, reflecting its overall prominence in stop activity

1.1.18 18. search_vehicle

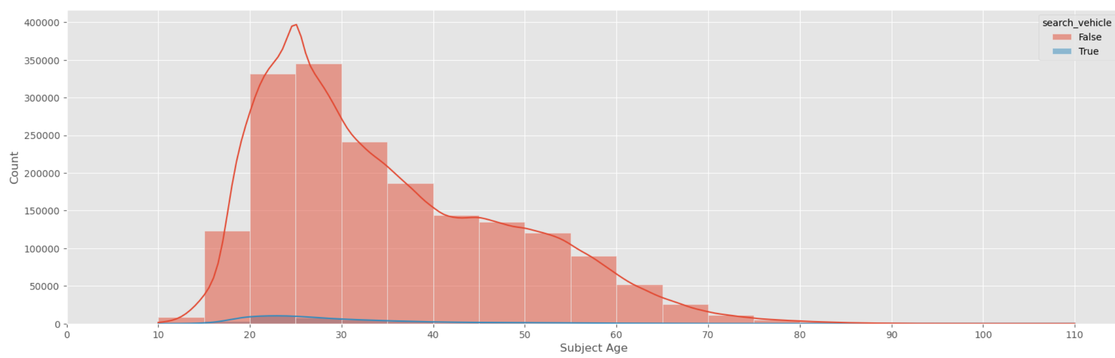
[169]:
```
df["search_vehicle"].value_counts().to_frame()
```

[169]:

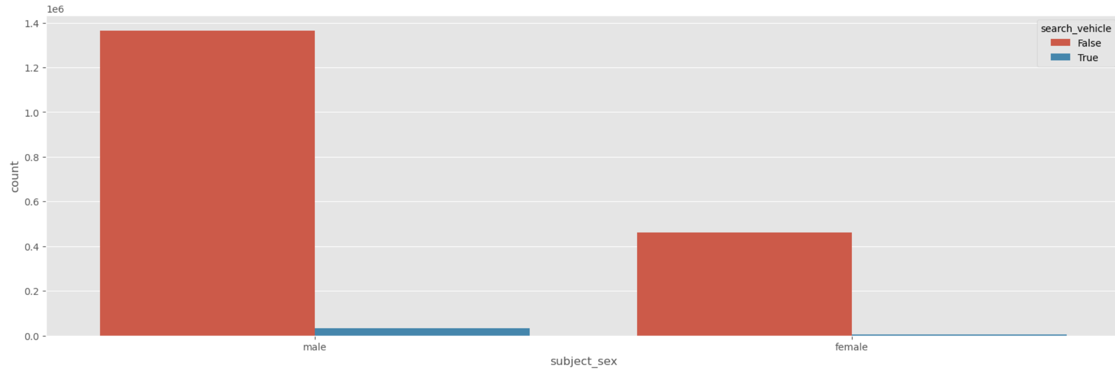
	count
search_vehicle	
False	1828666
True	36430

[170]:
```
age_sex_race("search_vehicle")
```

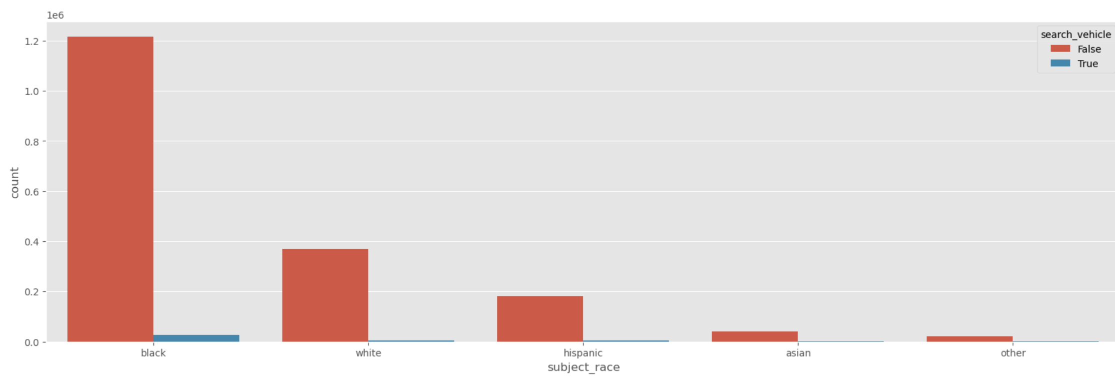
	proportion
search_vehicle	
False	0.98
True	0.02



	count	mean	std	min	25%	50%	75%	max
search_vehicle								
False	1824189.00	34.93	13.36	10.00	24.00	31.00	44.00	110.00
True	36348.00	29.83	10.81	12.00	22.00	27.00	34.00	100.00



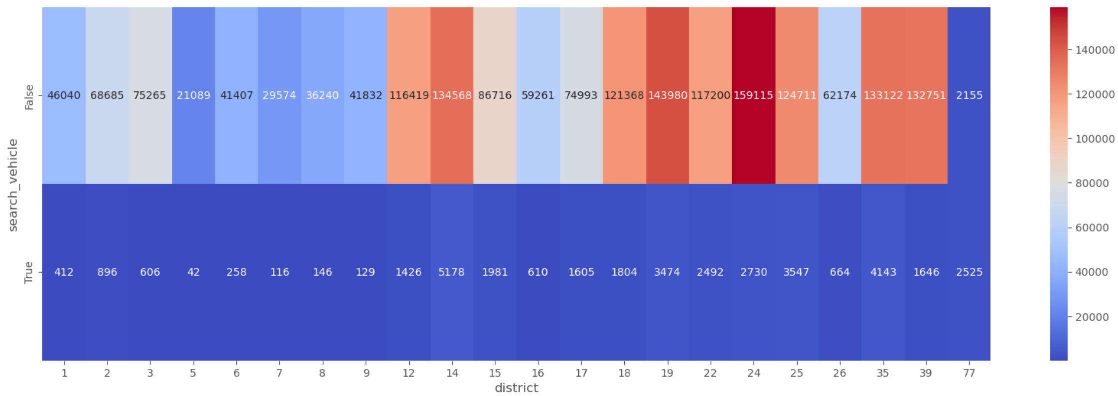

```
                        proportion
subject_sex search_vehicle
female      False             0.99
            True              0.01
male        False             0.98
            True              0.02
```




```
C:\Users\acast\AppData\Local\Temp\ipykernel_30128\2111065542.py:22:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
  display(df.groupby("subject_race")[column].value_counts(normalize =
True).to_frame())
```

```
                         proportion
subject_race search_vehicle
black        False             0.98
             True              0.02
white        False             0.99
             True              0.01
hispanic     False             0.98
```

	True	0.02
asian	False	0.99
	True	0.01
other	False	0.98
	True	0.02



Vehicle searches during stops are exceedingly rare overall, accounting for only a small percentage of vehicular interventions. However, their distribution reflects consistent patterns aligned with broader enforcement disparities. Younger drivers are slightly more likely to have their vehicles searched, with the rate declining progressively with age. Males are disproportionately more likely to experience vehicle searches than females, and while racial disparities in proportions are subtle, Black and Hispanic drivers face more frequent vehicle searches in absolute terms. Spatially, District 24 once again stands out with the highest number of vehicle searches, reinforcing its role as the focal point of policing activity in the city. This high level of scrutiny may be associated with efforts to combat drug trafficking in the area, particularly given the district's connection to the ongoing opioid crisis centered in neighborhoods like Kensington. The data suggest that while searches are rare, they are not randomly distributed and may reflect targeted enforcement based on geography and demographic factors.

1.1.19 19. raw data

[174]:
```
df["raw_race"].value_counts().to_frame()
```

[174]:

raw_race	count
Black - Non-Latino	1244249
White - Non-Latino	375862
White - Latino	162489
Asian	40245
Black - Latino	21695
Unknown	14958
American Indian	5598

[175]:
```python
df["raw_individual_contraband"].value_counts().to_frame()
```

[175]:

	count
raw_individual_contraband	
False	1834072
True	31024

[176]:
```python
df["raw_vehicle_contraband"].value_counts().to_frame()
```

[176]:

	count
raw_vehicle_contraband	
False	1854043
True	11053

This is data related to the raw source, therefore it will not be taken into account

1.2 Conclusions

The exploratory data analysis conducted on the Philadelphia police stop records from 2014 to 2018 reveals significant insights into the demographic, geographic, and temporal dynamics of law enforcement practices in the city. The dataset—comprising over 1.8 million records—uncovered stark disparities in how different population groups are subjected to stops, searches, and arrests.

One of the most prominent findings is the disproportionate targeting of Black individuals, who are consistently overrepresented in stop data relative to their population share. This pattern is evident across nearly every district and persists in all outcomes, including searches, arrests, and instances where contraband is found. Similarly, younger adults—particularly males in their twenties and thirties—experience the highest frequency of stops. Gender disparities are also clear, with men being far more likely to be stopped, searched, and arrested than women, across all racial groups.

Spatial analysis identified District 24, especially the Kensington area, as the most heavily policed zone, aligning with its known challenges related to drug use and public health crises. This suggests that socio-economic factors and localized conditions play a critical role in enforcement intensity. Stops tend to be more frequent in central and high-density areas, with major roads like Market Street and Broad Street showing elevated enforcement activity.

Temporally, stops exhibit clear patterns: they peak during evening rush hours (17:00–21:00) and on weekends, especially Friday and Saturday evenings. Seasonal variation also plays a role, with spring months (March–May) registering the highest number of stops, possibly due to increased mobility and enforcement campaigns.

While vehicular stops are the most common, pedestrian stops are disproportionately directed at young men of color. Despite the high number of stops, the majority do not lead to arrests or the discovery of contraband, raising concerns about the efficiency and justification of these enforcement strategies.

In summary, the data paint a picture of policing practices that are shaped by demographic profiles, neighborhood characteristics, and systemic biases. The findings call for a reevaluation of stop-and-search policies, with emphasis on transparency, accountability, and equitable treatment to ensure public safety efforts do not disproportionately burden specific communities.