nypd

May 13, 2021

1 NYPD Civilian Complaints

This project contains data on 12,000 civilian complaints filed against New York City police officers. Interesting questions to consider include: - Does the length that the complaint is open depend on ethnicity/age/gender? - Are white-officer vs non-white complaintant cases more likely to go against the complainant? - Are allegations more severe for cases in which the officer and complaintant are not the same ethnicity? - Are the complaints of women more succesful than men (for the same allegations?)

There are a lot of questions that can be asked from this data, so be creative! You are not limited to the sample questions above.

1.0.1 Getting the Data

The data and its corresponding data dictionary is downloadable here.

Note: you don't need to provide any information to obtain the data. Just agree to the terms of use and click "submit."

1.0.2 Cleaning and EDA

- Clean the data.
  - Certain fields have "missing" data that isn't labeled as missing. For example, there are fields with the value "Unknown." Do some exploration to find those values and convert them to null values.
  - You may also want to combine the date columns to create a `datetime` column for time-series exploration.
- Understand the data in ways relevant to your question using univariate and bivariate analysis of the data as well as aggregations.

1.0.3 Assessment of Missingness

- Assess the missingness per the requirements in `project03.ipynb`

1.0.4 Hypothesis Test / Permutation Test

Find a hypothesis test or permutation test to perform. You can use the questions at the top of the notebook for inspiration.

2 Code

[196]:
```
import matplotlib.pyplot as plt
import numpy as np
import os
import pandas as pd
import seaborn as sns
%matplotlib inline
%config InlineBackend.figure_format = 'retina'  # Higher resolution figures
```

3 Summary Of Findings

3.1 Intro

The NYPD dataset contains multiple factors of information regarding civilian complaints towards NYPD officers. These factors include (but are not limited to) the ages and ethnicities of both parties as well as the result of the filed case. This dataset is open to the public, and it is useful in uncovering underlying behaviors of NYPD officers and the Civilian Complaint Review Board.

When looking at the NYPD dataset, we can use analysis to answer different types of question regarding both parties involved for each allegation. One important question that we might be able to answer is one regarding the fidelity of the Civilian Complaint Review Board. We can use Hypothesis and Permutation tests to analyze our dataset to gain insight on the legitimacy of the board. The question we have for this report is: **Is the Civilian Complaint Review Board biased towards cases involving White officers and Black complainants?**

3.2 Data Cleaning Steps

In the data-cleaning process, we chose **not to remove any null values**, as those null values could provide information on a given column's influence on another column. For example, some columns like ethnicity, age, or gender, could be null for certain reasons, and those null values could help us draw parallels to the values of other columns.

We **replaced 'Unknown'** values with **null values** because, after further analysis, we saw that columns containing 'Unknown' values did not contain null values. Thus, replacing them with null would not falsely label those values as a different category.

We also **removed** any columns that were **abbreviations** of other columns, as they simply repeated other columns in the dataset and were not necessary for data analysis.

We also **combined** the **first and last names** of each officer into one column, because the scope of our analysis does not concern differences in first names or surnames. Therefore, we did not need to leave them separate and combined them for easier comprehension.

We **combined** the **month and year** columns for both received and closed, and converted them into datetime objects, for easier comprehension and potential data analysis using those columns.

3.3 Missingness

We believe that the data, specifically the column **'complainant_ethnicity'** (which has missing values in the data) is MAR in correlation with the **'mos_ethnicity'** column. The hypothesis

to explain the missingness correlation here is that complainants might not want to disclose their ethnicity depending on the officer's race.

We can also test **'complaintant_ethnicity'** with **'complaint_id'**. However, we believe that the missingness in 'complainant_age_incident' will not be dependent, because it seems unlikely for there to be a reason to omit ethnicity based on this category.

Our alpha levels for both test will be **0.05**

Our p-value for 'complainant_ethnicity' vs 'mos_ethnicity' was **0.0** (Dependent), while our p-value for 'complainant_ethnicity' vs 'complaint_id' was **1.0**(Not Dependent). See *"Assessment of Missingness" for more information.*

3.4 Hypothesis Testing

For our hypothesis test, we decided to use the **Total Variation Distance Permutation Test** with a significance level of **0.05**.

Our **Null Hypothesis** was that the Board handles allegations involving White officers against Black complainants the same way as all other allegations. Our **Alternate Hypothesis** was that the Board has bias towards allegations involving White officers against Black complainants differently.

Our **test statistic** was the **Total Variation Distance of Board Dispositions**.

Our calculated p-value is **0.001**, meaning that we **reject the null hypothesis**. Thus, there **is** enough statistically significant evidence to conclude that, for this dataset, the board handled allegations involving White officers against Black complainants differently in comparison to any other allegation.

4 Cleaning and EDA

4.1 Cleaning

The following code cleans our dataset as described in our summary findings, making it easier for comprehension and data analysis.

[197]:
```
nypd_fp = os.path.join('data', 'allegations_202007271729.csv')
nypd = pd.read_csv(nypd_fp)
    # Removes unneccessary 'Abbreviated' columns and fills 'Unknown with null'
nypd = nypd[[x for x in nypd.columns if "abbrev" not in x]].replace('Unknown',␣
↪np.nan)
    # Combines the month and year columns for received and closed,␣
↪respectively, and converts to datetime object
nypd['date_received'] = pd.to_datetime(nypd['month_received'].apply(str) + '/'␣
↪+ nypd['year_received'].apply(str)).dt.to_period('M')
nypd['date_closed'] = pd.to_datetime(nypd['month_closed'].apply(str) + '/' +␣
↪nypd['year_closed'].apply(str)).dt.to_period('M')
nypd = nypd.drop(columns=['month_received', 'year_received', 'month_closed',␣
↪'year_closed'])

    # Combines first and last name to "full name" column
```

```
nypd['full_name'] = nypd['first_name'] + ' ' + nypd['last_name']
nypd = nypd.drop(columns=['first_name', 'last_name'])
nypd.head()
```

[197]:

```
   unique_mos_id command_now  shield_no  complaint_id command_at_incident  \
0          10004     078 PCT       8409         42835              078 PCT
1          10007     078 PCT       5952         24601                 PBBS
2          10007     078 PCT       5952         24601                 PBBS
3          10007     078 PCT       5952         26146                 PBBS
4          10009     078 PCT      24058         40253              078 PCT

         rank_now   rank_incident mos_ethnicity mos_gender  mos_age_incident  \
0  Police Officer  Police Officer      Hispanic          M                32
1  Police Officer  Police Officer         White          M                24
2  Police Officer  Police Officer         White          M                24
3  Police Officer  Police Officer         White          M                25
4  Police Officer  Police Officer      Hispanic          F                39

   … complainant_age_incident           fado_type  \
0  …                      38.0  Abuse of Authority
1  …                      26.0         Discourtesy
2  …                      26.0  Offensive Language
3  …                      45.0  Abuse of Authority
4  …                      16.0               Force

                     allegation precinct  \
0  Failure to provide RTKA card     78.0
1                        Action     67.0
2                          Race     67.0
3                      Question     67.0
4               Physical force     67.0

                                contact_reason  \
0                      Report-domestic dispute
1                             Moving violation
2                             Moving violation
3  PD suspected C/V of violation/crime - street
4                               Report-dispute

                outcome_description                          board_disposition  \
0  No arrest made or summons issued  Substantiated (Command Lvl Instructions)
1   Moving violation summons issued                    Substantiated (Charges)
2   Moving violation summons issued                    Substantiated (Charges)
3  No arrest made or summons issued                    Substantiated (Charges)
4    Arrest - other violation/crime       Substantiated (Command Discipline A)

  date_received date_closed      full_name
```

```
0       2019-07     2020-05  Jonathan Ruiz
1       2011-11     2012-08     John Sears
2       2011-11     2012-08     John Sears
3       2012-07     2013-09     John Sears
4       2018-08     2019-02   Noemi Sierra

[5 rows x 22 columns]
```

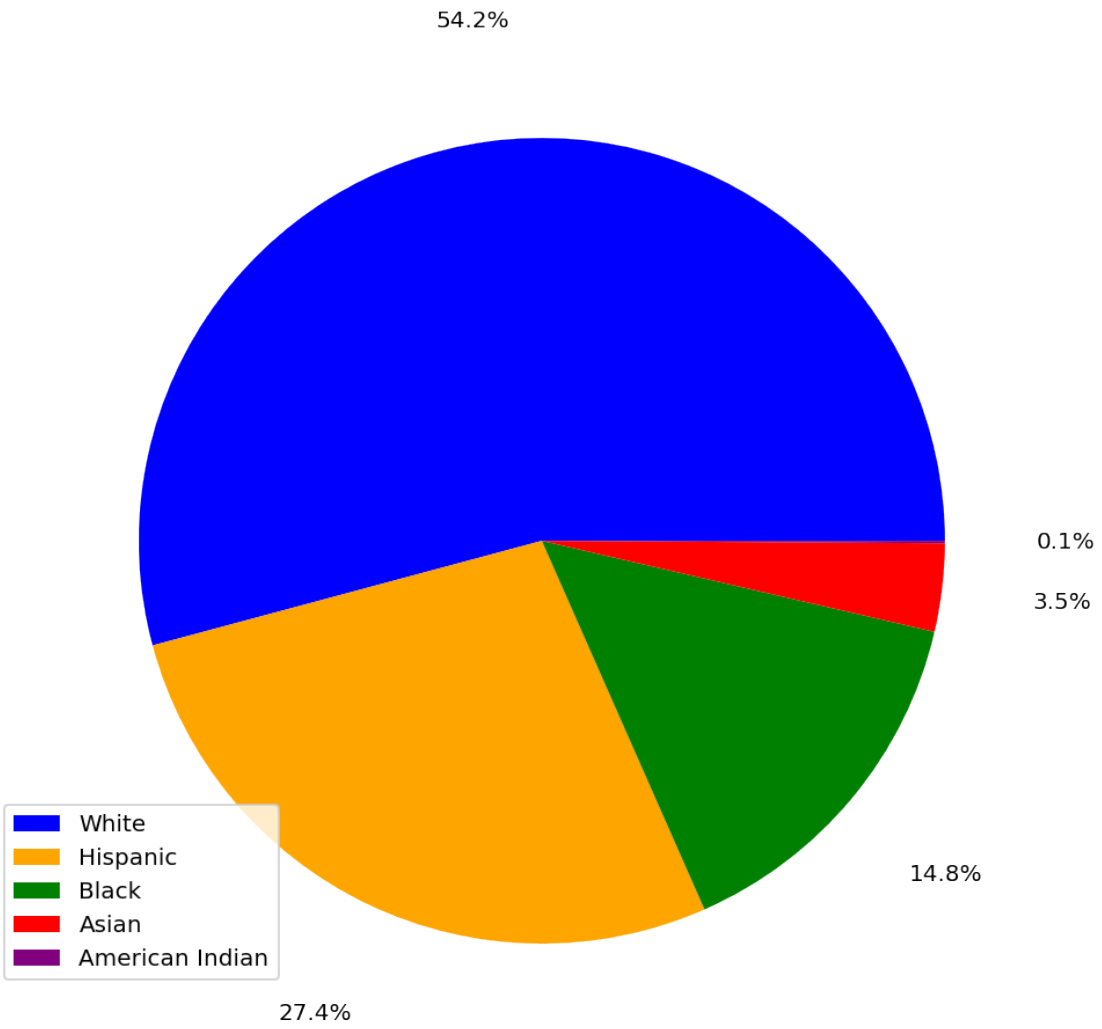
4.2 EDA

4.2.1 Distributions of Ethnicity (Officers and Complainants)

Based on the pie charts below, we can see that **White Officers** and **Black Complainants** make up a majority of the people involved in these allegation cases. However, it is not possible to make any claims or conclusions without further analysis.

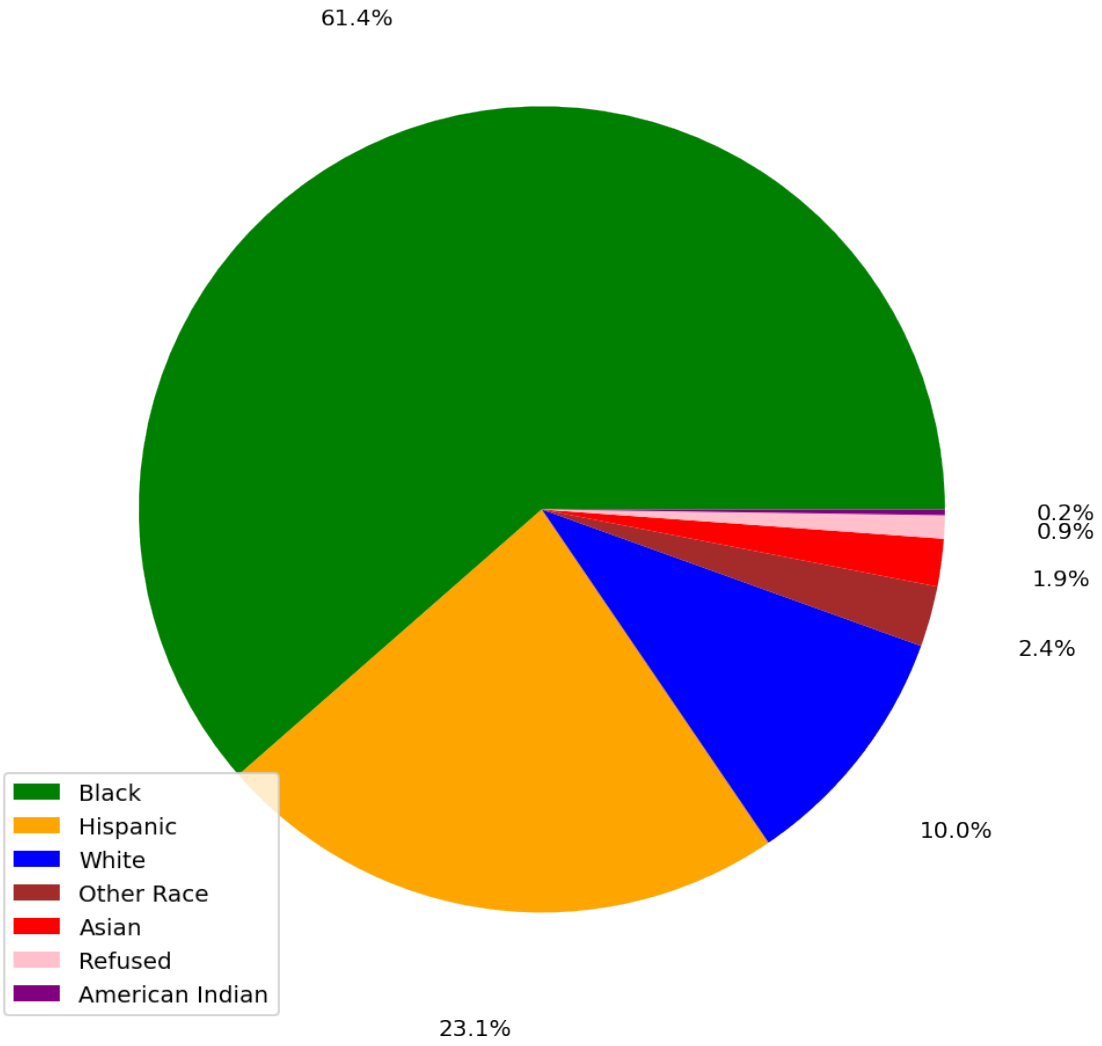
[198]:
```
# pie chart for proportion of officer ethnicities
officer_ethnicity = nypd['mos_ethnicity'].value_counts(normalize = True)
plt.pie(officer_ethnicity,normalize=False, autopct='%1.1f%%',radius=2,␣
 ↪pctdistance=1.3,colors=['blue','orange','green','red','purple'])
plt.legend(officer_ethnicity.index,bbox_to_anchor=(0,0))
```

[198]:
```
<matplotlib.legend.Legend at 0x7fc92752fc40>
```



[199]:
```
# pie chart for proportions of complainant ethnicities
complaintant_ethnicity = nypd['complainant_ethnicity'].value_counts(normalize =␣
 ↪True)
plt.pie(complaintant_ethnicity,normalize=False, autopct='%1.1f%%',radius=2,␣
 ↪pctdistance=1.
 ↪3,colors=['green','orange','blue','brown','red','pink','purple'])
plt.legend(complaintant_ethnicity.index,bbox_to_anchor=(0,0))
```

[199]:
```
<matplotlib.legend.Legend at 0x7fc927522e20>
```



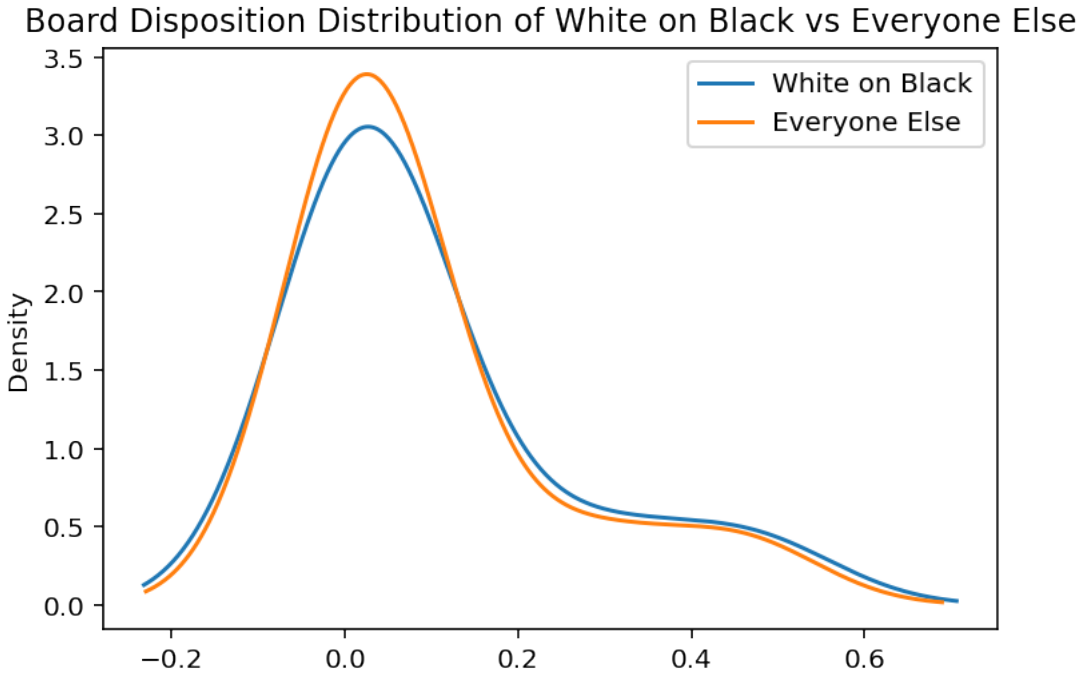
4.2.2 Distributions of Board Dispositions (White Officers vs Black Complainants, Everyone Else)

Although the distributions of Board disposition seem the same for both White officer vs Black complainant cases and everyone else, further analysis can be conducted to better understand the true distributions.

[200]:
```
wob = nypd[(nypd['mos_ethnicity'] == 'White') & (nypd['complainant_ethnicity']␣
↪== 'Black')]['board_disposition'].value_counts(normalize=True)
overall = nypd[(nypd['mos_ethnicity'] != 'White') |␣
↪(nypd['complainant_ethnicity'] != 'Black')]['board_disposition'].
↪value_counts(normalize=True)
dispositions = pd.DataFrame(wob).rename(columns={'board_disposition': 'White on␣
↪Black'})
```

```
dispositions = pd.concat([dispositions,overall],axis=1).
 ↪rename(columns={'board_disposition': 'Everyone Else'})
```

[201]:
```
wob_vs_overall = dispositions.plot.kde().set_title('Board Disposition␣
 ↪Distribution of White on Black vs Everyone Else')
```



5 Assessment of Missingness

From our TVD permutation test comparing complainant and officer ethnicities, we see that our p-value is extremely low, which leads us to conclude that 'complainant_ethnicity' is in fact **dependent** on the 'mos_ethnicity' column, or **MAR**. A likely scenario to explain this would be: if the complainant and the alleged officer in question belong to different ethnicities, the complainant might be more compelled to refrain from revealing their ethnicity to prevent any potential bias from the Civilian Complaint Review Board.

From our test comparing complainant ethnicity and complaint id, we see that our p-value is high, which leads us to conclude that 'complainant_ethnicity' is **not dependent** on 'complaint_id'.

[206]:
```
nypd['complainant_ethnicity_null'] = nypd['complainant_ethnicity'].isnull()
def tvd(data, missing, test):
    distr = data.groupby(missing)[test].value_counts(normalize=True).to_frame().
 ↪rename(columns={test: 'proportion'})
    distr = distr.pivot_table(index=missing,columns=test, values='proportion')
    return distr.T.diff(axis=1).abs().sum().iloc[-1] / 2
```

```
observed_1 = tvd(nypd, 'complainant_ethnicity_null', 'mos_ethnicity')
observed_2 = tvd(nypd, 'complainant_ethnicity_null', 'complaint_id')
tvds_1 = []
tvds_2 = []
for i in range(500):
    out = nypd.copy()
    out['complainant_ethnicity_null'] = out['complainant_ethnicity_null'].
 ↪sample(frac=1, replace=False).reset_index(drop=True)
    tvds_1.append(tvd(out, 'complainant_ethnicity_null', 'mos_ethnicity'))
    tvds_2.append(tvd(out, 'complainant_ethnicity_null', 'complaint_id'))
(np.mean(pd.Series(tvds_1) >= observed_1),
np.mean(pd.Series(tvds_2) >= observed_2))
```

[206]: (0.0, 1.0)

6 Hypothesis Test

Null: The Board handles allegations involving White officers against Black complainants the same way as all other allegations

Alt: The Board has bias towards allegations involving White officers against Black complainants differently

Alpha: 0.05

Test Statistic: Total Variation Distance of Board Dispositions

[205]:
```
nypd['WoB'] = (nypd['mos_ethnicity'] == 'White') &␣
 ↪(nypd['complainant_ethnicity'] == 'Black')
def tvd(data):
    distr = data[['board_disposition','WoB']]
    distr = distr.groupby('WoB')['board_disposition'].
 ↪value_counts(normalize=True).to_frame().rename(columns={'board_disposition':␣
 ↪'proportion'})
    distr = distr.pivot_table(index='WoB',columns='board_disposition',␣
 ↪values='proportion')
    return distr.T.diff(axis=1).abs().sum().iloc[-1] / 2

observed = tvd(nypd)
tvds = []
for i in range(1000):
    out = nypd.copy()
    out['WoB'] = out['WoB'].sample(frac=1, replace=False).reset_index(drop=True)
    tvds.append(tvd(out))
(pd.Series(tvds) >= observed).sum() / 1000
```

[205]: 0.001