CS506 Team Weekly Scrum Report 5

1. **What we accomplished this week:**
    - Reached out to the TA and Spark PM to discuss the challenges and blockers we are currently encountering. Specifically, we recognized certain fields in the dataset, such as Overtime type (X, R, S, C) and Officer Rank, for which clear definitions are lacking. Additionally, we observed that some fundamental questions are directed towards information we don't currently have access to, such as officers' demographic characteristics and BDP fundings, …

    - Using the Spark PM’s suggestion, we delved into expanding our datasets using online resources to gather additional information related to the fundamental questions.

    - In a group meeting, we observed redundant efforts among team members addressing the same core questions. Subsequently, we devised a strategy to synchronize tasks to enhance efficiency and minimize duplication of offorts within the team

    - We were able to link real-life social events to our data analysis results. Notably, our analysis in 2014 revealed a significant surge in injured officers and injury payments. We attributed this increase to the events surrounding the George Floyd murder and the Black Lives Matter protests.

Members and targeting base questions:
(not more than 2 members on 1 question:

- How do overtime hours paid compare to overtime hours worked? What does the discrepancy financially amount to, year after year? (Riva)
  What is the distribution of ratios of overtime worked vs. overtime paid? Are there any outliers? (WRKDHRS vs. OTHOURS in the court OT database). (Riva + Truc)
- How has overtime for court appearances changed year-over-year? (Truc)

- How much overlap is there between frequency overtime users and officers who:
    - have the highest salaries on the force? (Can)
    - are listed on the Suffolk County police watch list? (Nurassyl, only found 2020 data for suffolk police watch lsit)
    - have previously been disciplined for overtime abuse or other misconduct?
    - have internal affairs complaint records? (Al)
- for internal affairs complaint: I found this data source
  https://www.wokewindows.org/help/internal_affairs
  I added the csv dataset to our google drive

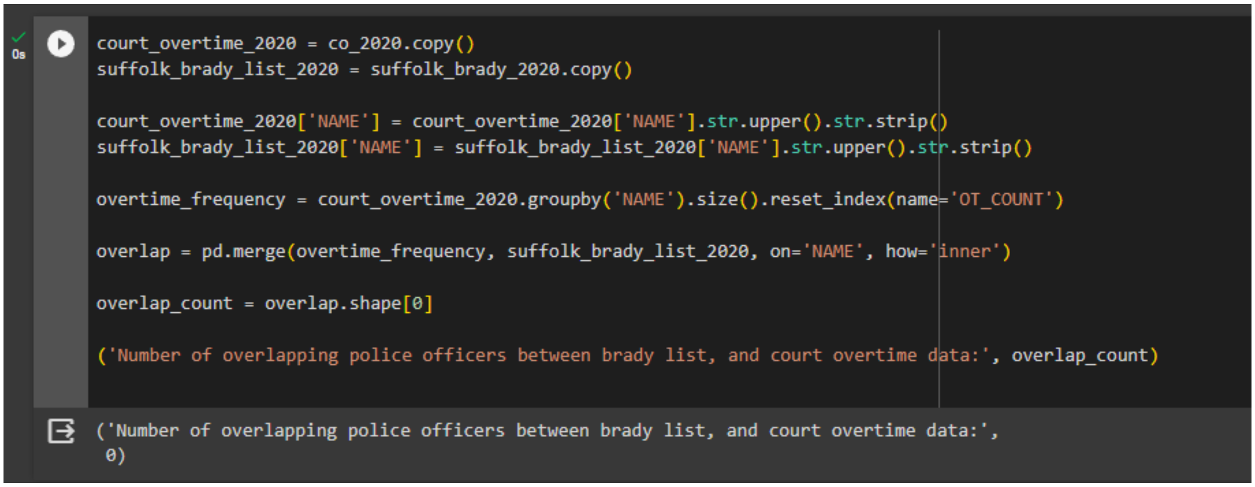
(a) Deliverable Links:

- https://colab.research.google.com/drive/1gVfObsV1cbzK5XpTjjA9xc1m9cBou1LY#scrollTo=zV_hbsOC55Yq
- https://drive.google.com/drive/folders/1s5SHPEVq_ScGbUmSTzI8JXkjWZIom0p_
-

2. Individual team member updates:

[Nurassyl Medeu]

- Found Suffolk County brady list 2020 [link]
    - Brady List - list of officers flagged by prosecutors as either having engaged in, or been accused of, misconduct that the DA's office might legally need to disclose to the defense.
- How much overlap is there between frequency overtime users and officers who are listed on the Suffolk County police watch list?
    - So far only found brady list data for year 2020, (going to ask the client if that data can be considered the same as 'watch list')
    - For year 2020 no overlaps have been found

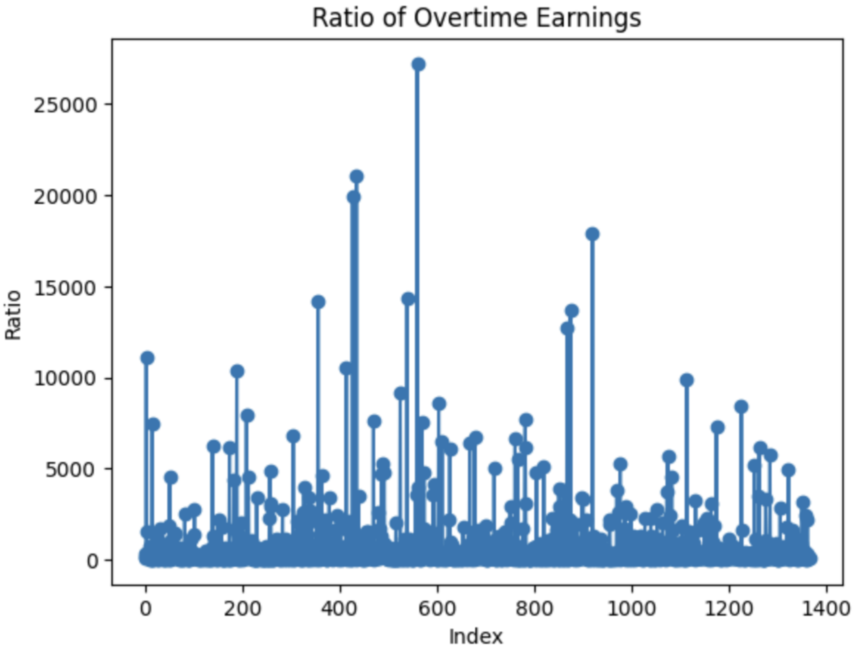


```
court_overtime_2020 = co_2020.copy()
suffolk_brady_list_2020 = suffolk_brady_2020.copy()

court_overtime_2020['NAME'] = court_overtime_2020['NAME'].str.upper().str.strip()
suffolk_brady_list_2020['NAME'] = suffolk_brady_list_2020['NAME'].str.upper().str.strip()

overtime_frequency = court_overtime_2020.groupby('NAME').size().reset_index(name='OT_COUNT')

overlap = pd.merge(overtime_frequency, suffolk_brady_list_2020, on='NAME', how='inner')

overlap_count = overlap.shape[0]

('Number of overlapping police officers between brady list, and court overtime data:', overlap_count)
```

```
('Number of overlapping police officers between brady list, and court overtime data:',
 0)
```

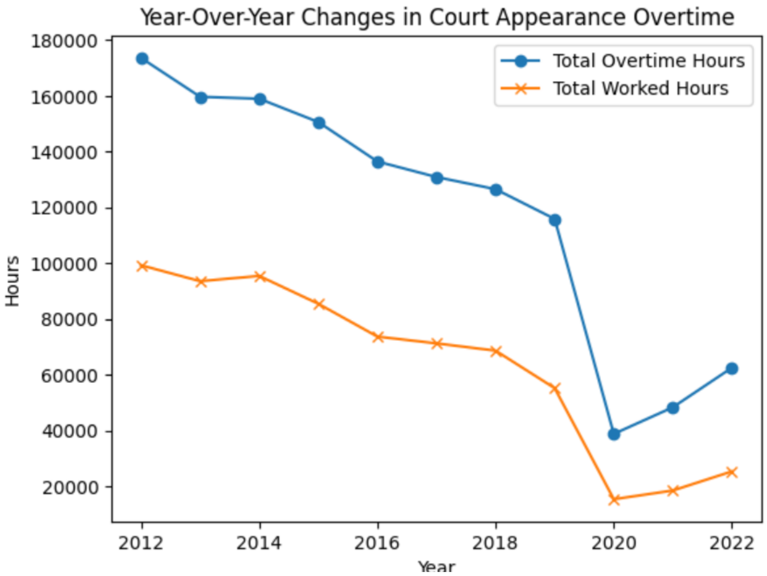
[Riva Sun]

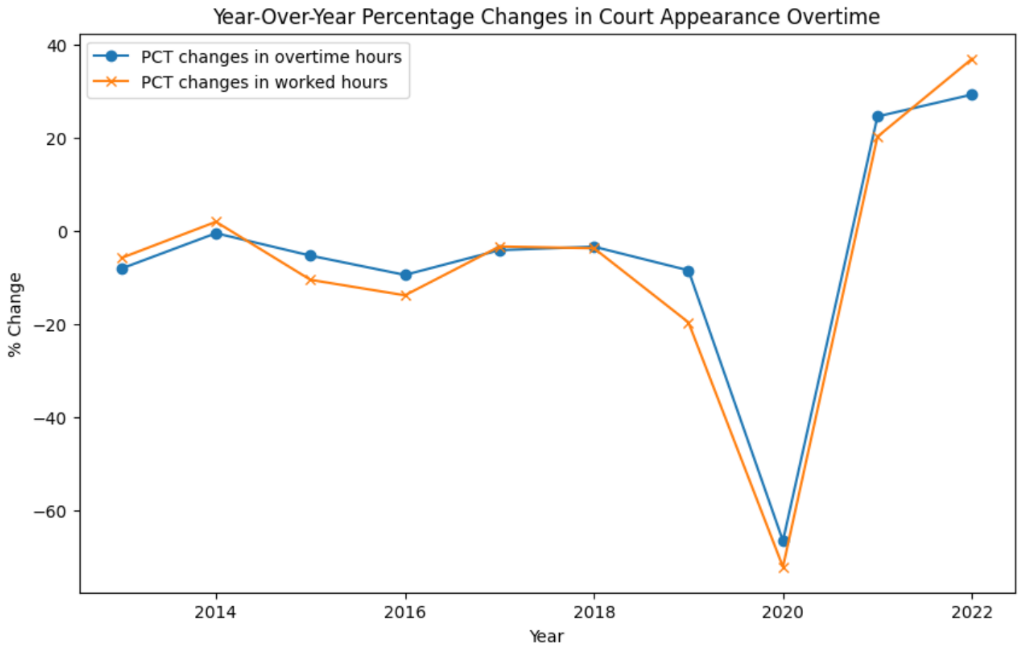
- In 2020, the amount of money bpd getting paid for each hour of overtime work is not fixed.
- By assumption, the earnings for overtime/hour should be maintained the same among each person. However, in 2020 (only this dataset contains EMP_ID identification), we can see from the graph that overtime earnings per hour is changing.
- This can be a suspicious aspect of some expenditure recorded under overtime earning.



[Truc Duong]

- Performed data analysis on the 'Court overtime' data set, targeting the question "How has overtime for court appearances changed year-over-year?"
- Hypothesis and assumption: I used the reported WRKHRS and OTHRS as a measurement for "appearances" in court



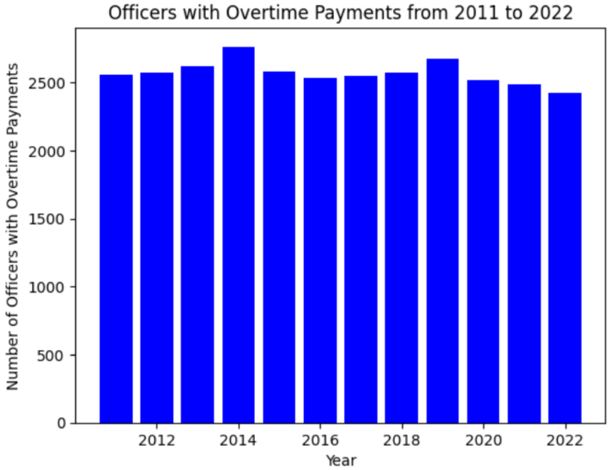


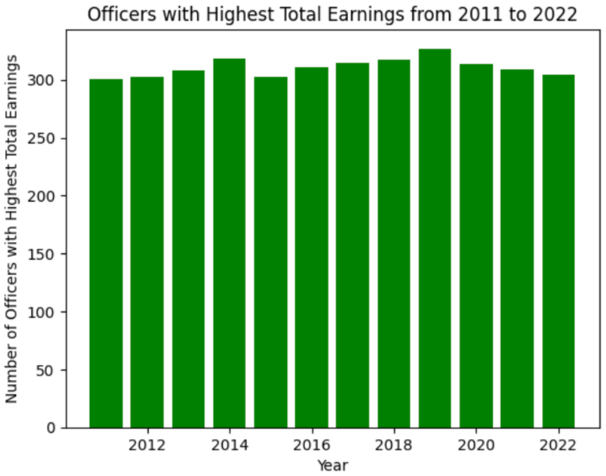
Observation:

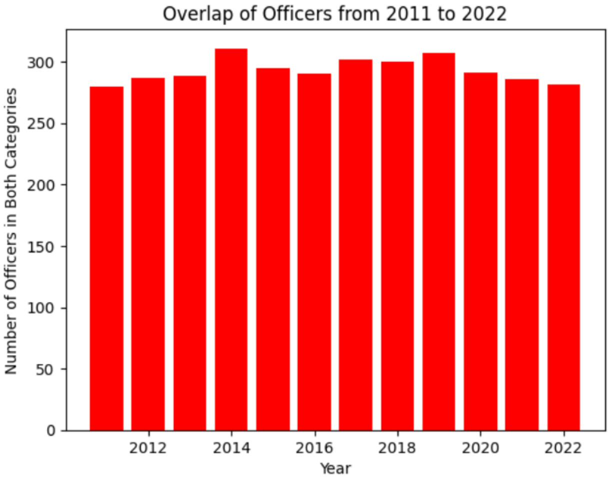
- In general, the total overtime hours consistently appeared to be twice the total work hours.
- The year 2012 recorded the highest reported court overtime and worked hours.
- Conversely, 2020 witnessed the lowest reported court overtime and worked hours, potentially influenced by the COVID-19 pandemic and a surge in remote jobs.

[Can Wang]

- The number of officers overtime are directly related to the officer that got payed the most. Here are the graphs.

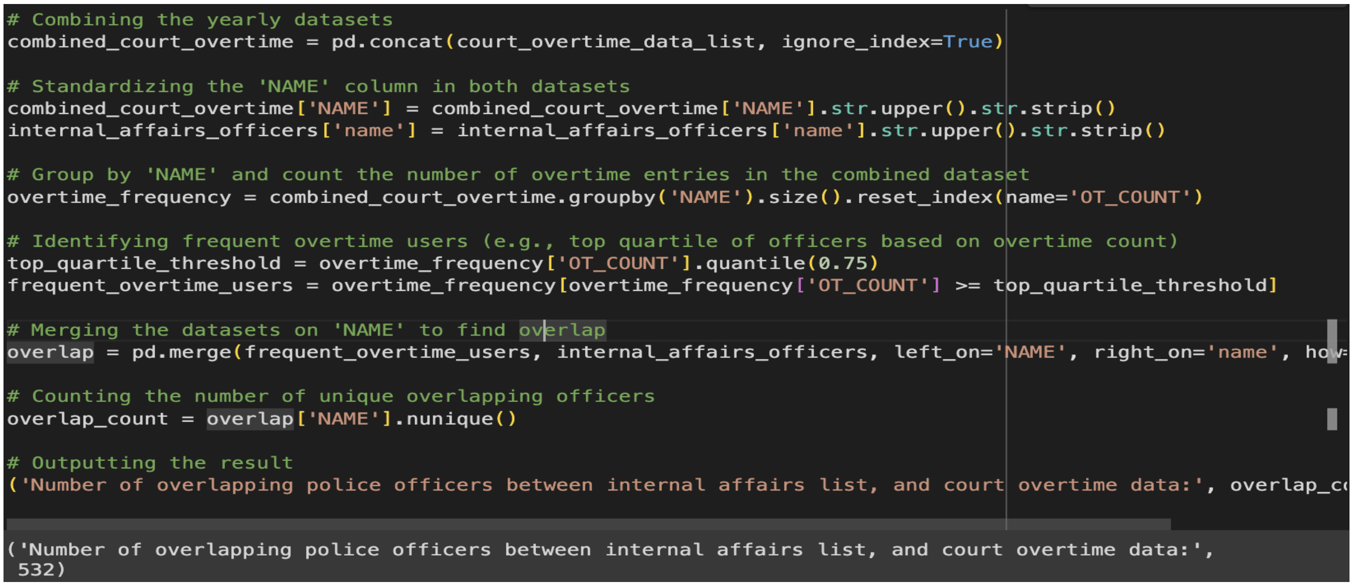






[Al Mbaye]

- Used internal_affairs_officers.csv (provided by Truc)
    - Imported the data
    - Preprocessed the data
- How much overlap is there between frequency overtime users and officers who have internal affairs complaint records?
    - Used the top 25% of overtime users to determine “frequent” users



```
# Combining the yearly datasets
combined_court_overtime = pd.concat(court_overtime_data_list, ignore_index=True)

# Standardizing the 'NAME' column in both datasets
combined_court_overtime['NAME'] = combined_court_overtime['NAME'].str.upper().str.strip()
internal_affairs_officers['name'] = internal_affairs_officers['name'].str.upper().str.strip()

# Group by 'NAME' and count the number of overtime entries in the combined dataset
overtime_frequency = combined_court_overtime.groupby('NAME').size().reset_index(name='OT_COUNT')

# Identifying frequent overtime users (e.g., top quartile of officers based on overtime count)
top_quartile_threshold = overtime_frequency['OT_COUNT'].quantile(0.75)
frequent_overtime_users = overtime_frequency[overtime_frequency['OT_COUNT'] >= top_quartile_threshold]

# Merging the datasets on 'NAME' to find overlap
overlap = pd.merge(frequent_overtime_users, internal_affairs_officers, left_on='NAME', right_on='name', how=

# Counting the number of unique overlapping officers
overlap_count = overlap['NAME'].nunique()

# Outputting the result
('Number of overlapping police officers between internal affairs list, and court overtime data:', overlap_co
```

```
('Number of overlapping police officers between internal affairs list, and court overtime data:',
 532)
```

- From 2012 to 2022, there were 532 overlapping officers

3. Issues or blockers:

- There is inconsistency between datasets. We are attempting to perform a cross-product between two different datasets based on officer names. However, we are facing challenges as officer names are not unique. We are still actively working to resolve it.
- Inconsistent formatting or categorization of data across different datasets makes it difficult in integration and analysis.
- Differentiating between various types of overtime (e.g., court, emergency response) requires a nuanced understanding of police operations and policies. And we don't have a lot of understanding in this field.
- Limited resources restrict the depth and breadth of the project analysis.

4. Plans for next week

- Our next step strategy involves linking the outcomes of individual team members' analyses.Ensure that insights from different team members are compatible and contribute to a comprehensive understanding of the data.
- Tidy up the workspace, and redirect our primary focus towards addressing the fundamental questions.
- Begin the process of integrating individual team members' analyses. Ensure that insights from different team members are compatible and contribute to a comprehensive understanding of the data
- Assess the techniques used in individual analyses and identify opportunities for refinement or improvement. This can involve exploring alternative statistical methods or considering additional variables for a deeper analysis.