A logo for a university

Description automatically generated

Higher diploma in science in computing

Lecturer: Patrizio Simeoni

Subject: Scripting

Lab Eight – Web scraping

Course: CW\_KRSIT\_H

Date: 18/1/2025

Student: John Paul Larkin

Student number: C00001754

## Title: Part one – FBI most wanted

A person in a uniform and a wanted poster

AI-generated content may be incorrect.

The API I chose was the FBI most wanted list. The URL is ‘https://api.fbi.gov/wanted/v1/list’

I chose the FBI Most Wanted API because I found it to be an interesting subject. It offers insights into high-profile cases, which makes it engaging to explore and provides an opportunity to learn web scraping with real-world, publicly available data.

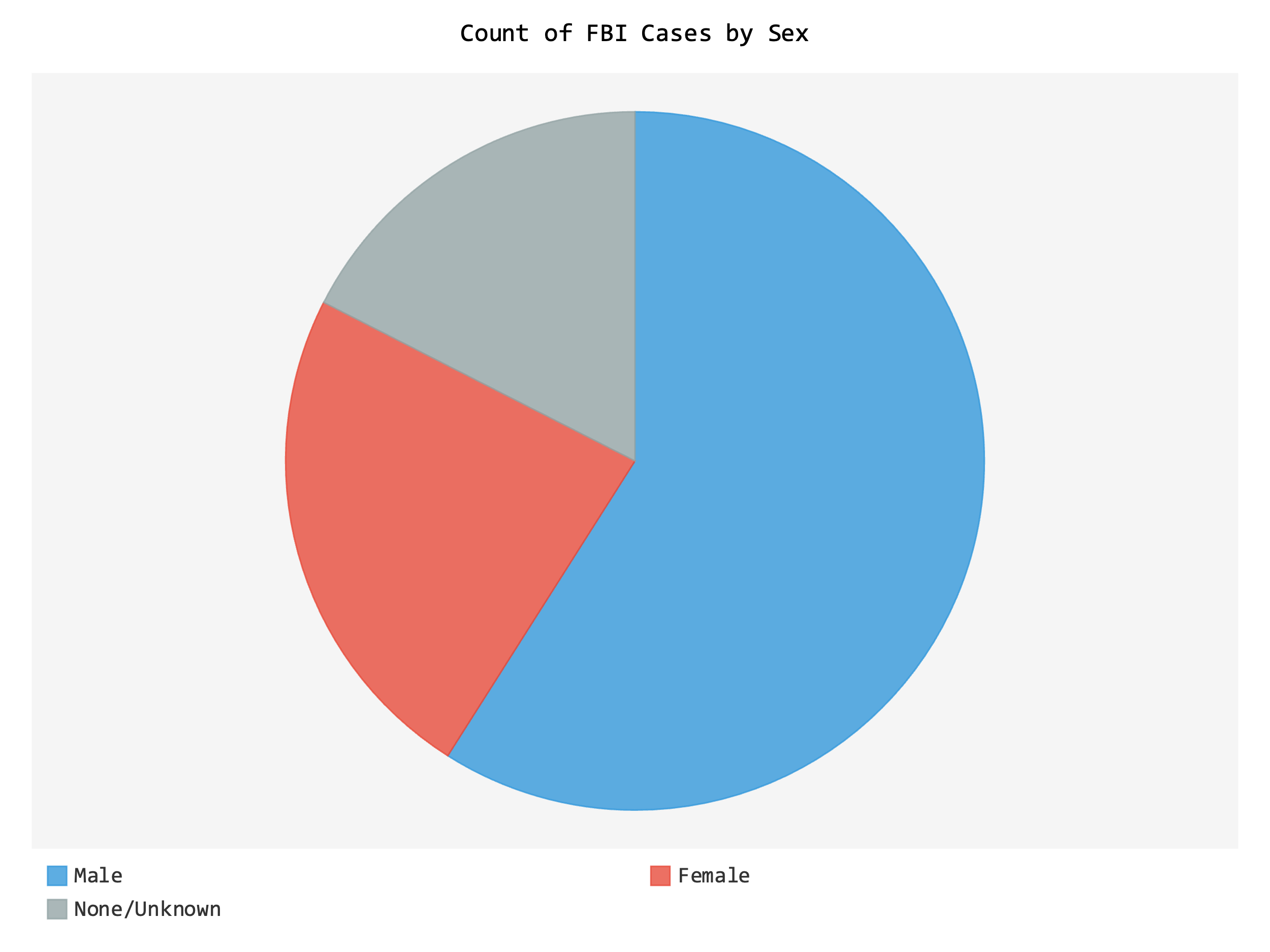
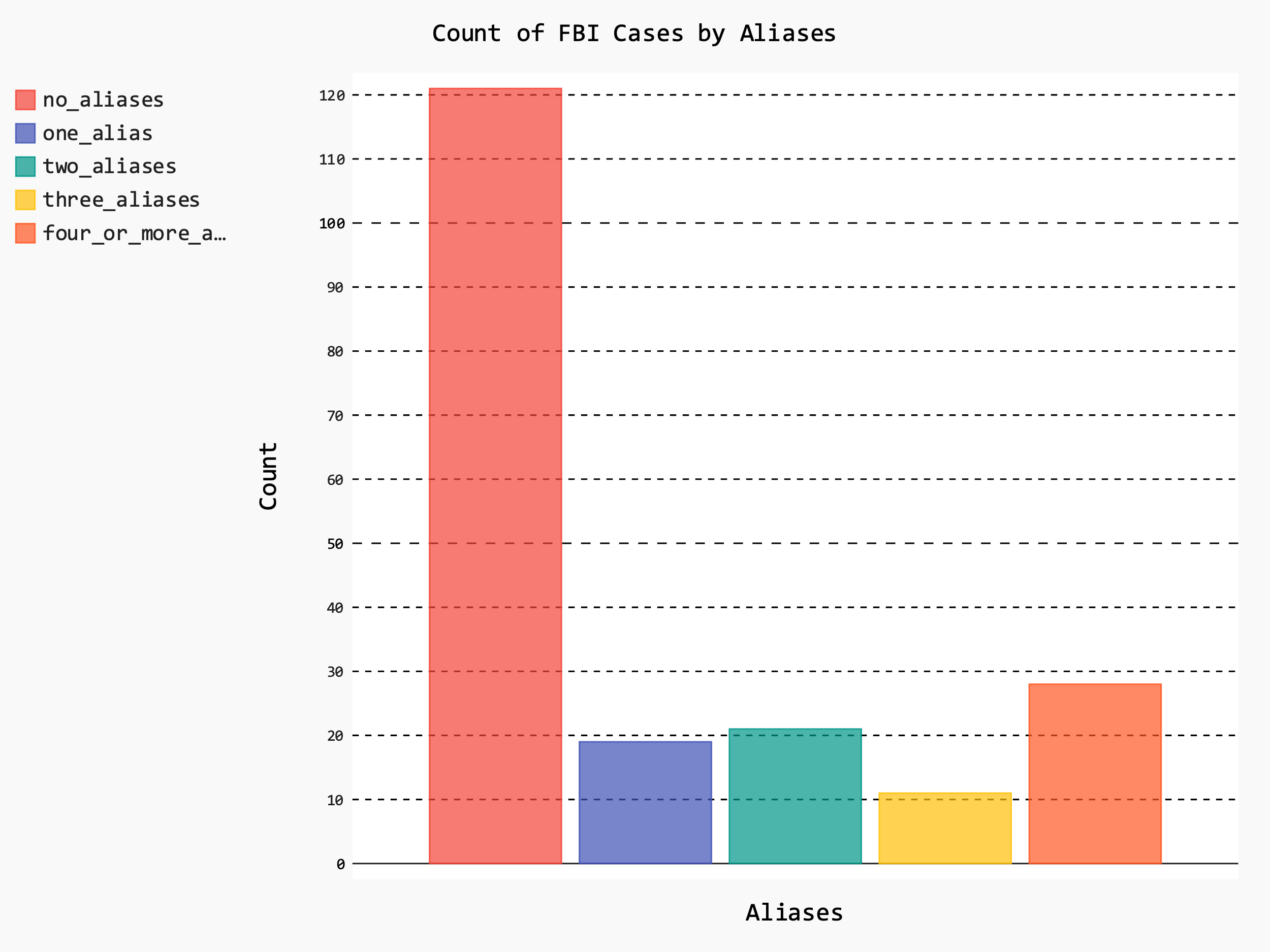
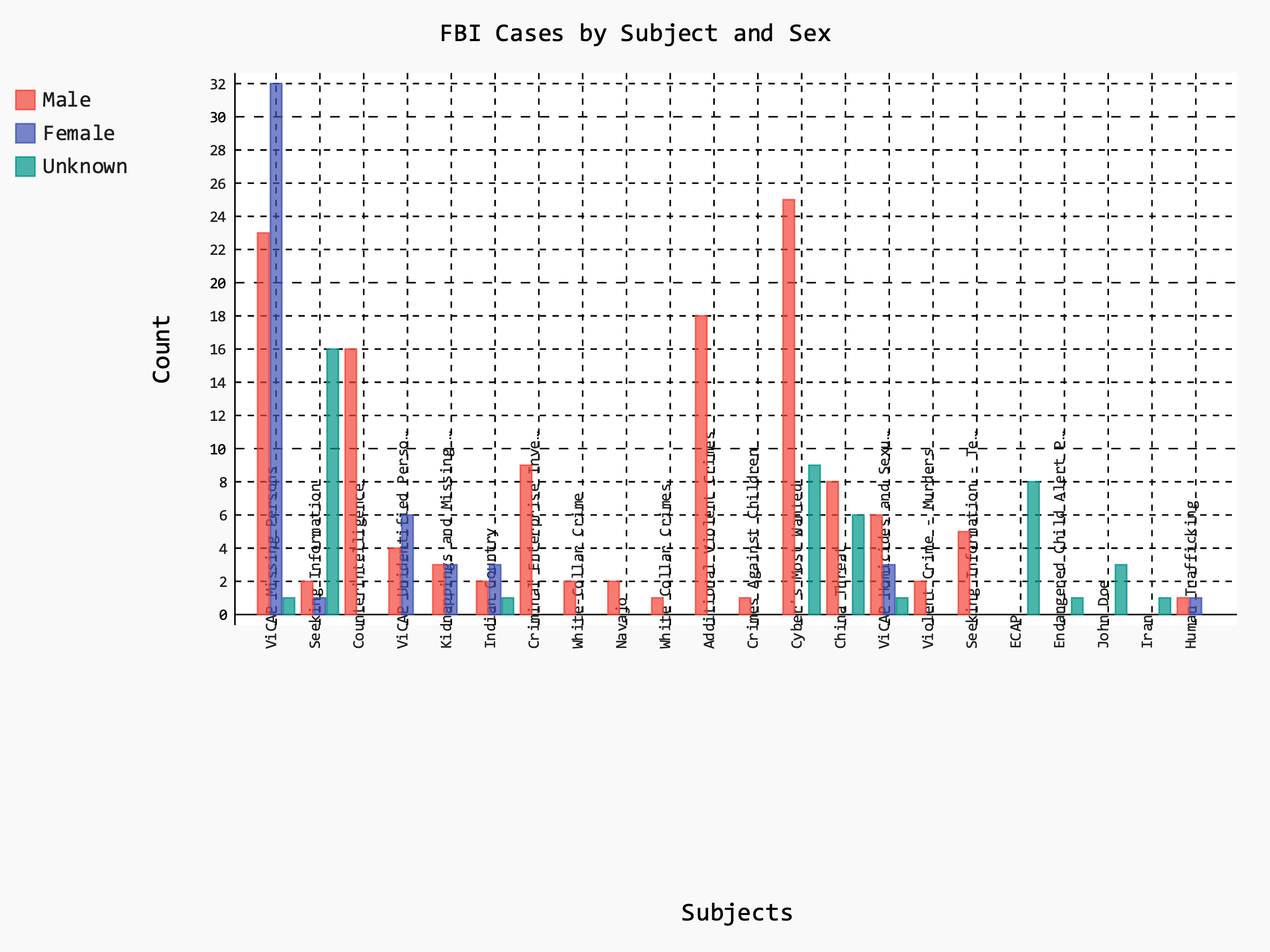
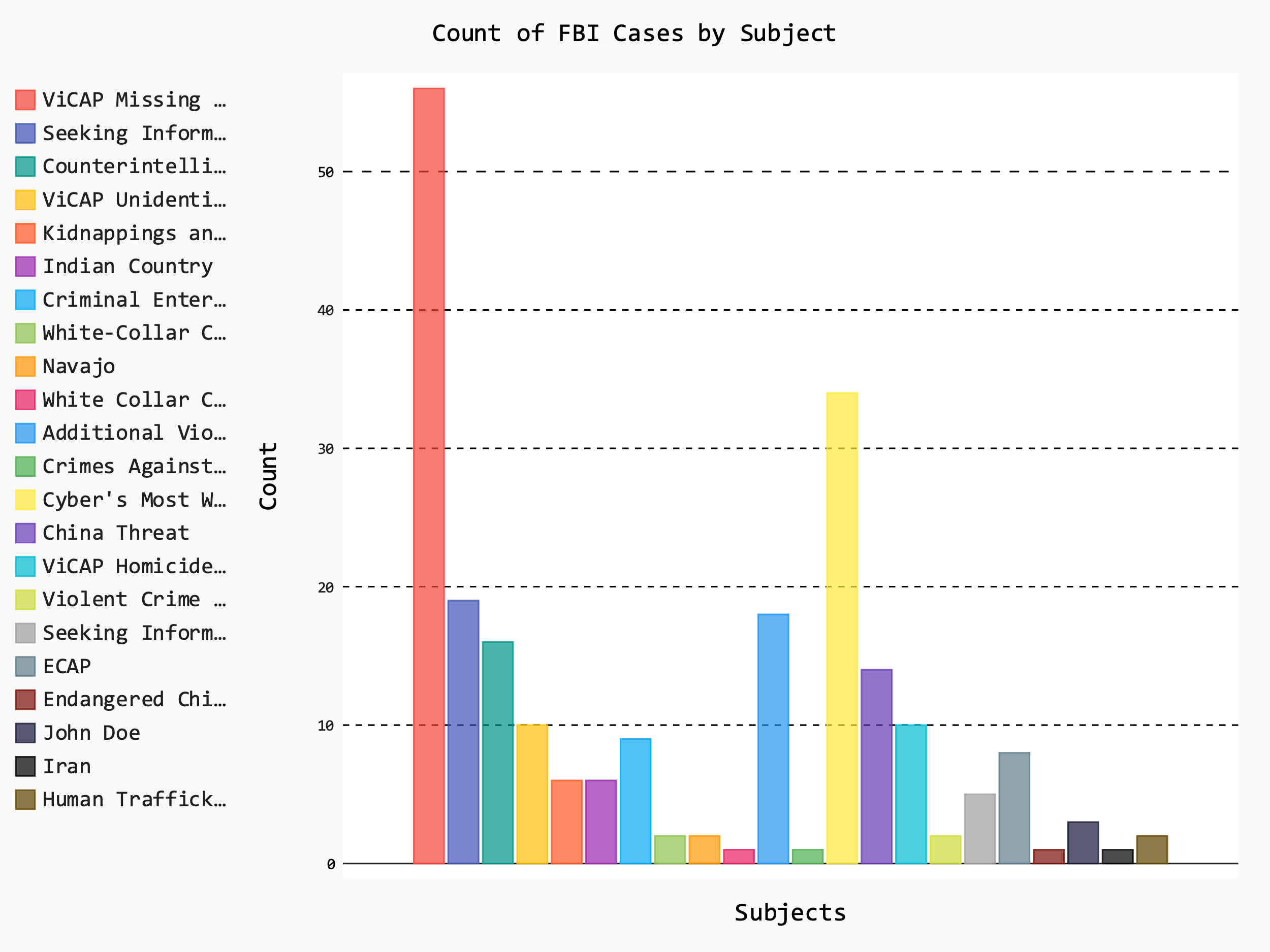
This structured format includes fields such as age, sex, occupation, subjects, and field\_offices, among others. Each entry provides detailed information about individuals on the FBI's wanted list, facilitating efficient data retrieval and processing.

### Issues

While working with the FBI Most Wanted API, I encountered issues with the data as many fields were null, making it challenging to extract meaningful insights. Additionally, some fields didn’t contain the expected data. For example, the "dates\_of\_birth\_used" field lacked a consistent, structured format, complicating parsing and analysis. Similarly, the "weight" field often presented ranges as strings, rather than numerical values, requiring additional handling to standardise the data. These inconsistencies highlighted the need for flexible processing logic to address missing and irregular data while ensuring accurate interpretation.

Another issue I faced was rate limits on the API. I had originally planned to parse the entire data set of 1053 wanted individuals, however I started receiving 429 status codes, so I limited the API call to 5 pages, for a total of 200 people. I saved this to a json file and parsed the data from the file, rather than making API calls.

The list of fields for each item(person) on the wanted list includes:

files, age\_range, uid, weight, occupations, field\_offices, locations, reward\_text, sex, hair, ncic, dates\_of\_birth\_used, caution, nationality, age\_min, age\_max, scars\_and\_marks, subjects, aliases, race\_raw, suspects, publication, title, coordinates, hair\_raw, languages, complexion, build, details, status, legat\_names, eyes, person\_classification, description, images, possible\_countries, weight\_min, additional\_information, remarks, path, eyes\_raw, weight\_max, reward\_min, url, possible\_states, modified, reward\_max, race, height\_max, place\_of\_birth, height\_min, poster\_classification, warning\_message, pathId.  

## Title - Part two – Premier league rank

I initially looked on [www.fandom.com](http://www.fandom.com) in search of html table to scrape. I could not find one, so I changed to <https://fbref.com> instead. This site had many tables of statistics, so I chose the premier league ranking table. I used the element inspector to find the id of the table element.

The bottom\_three function takes the table, and strips out the rank and team name from each row. I then sort them and return the lowest three.

Rank 18: Ipswich Town

Rank 19: Leicester City

Rank 20: Southampton

The top\_five\_highest\_attendance strips out the attendance numbers and returns the top five.

Attendance 73,713: Manchester Utd

Attendance 62,468: West Ham

Attendance 61,339: Tottenham

Attendance 60,276: Liverpool

Attendance 60,269: Arsenal

### Conclusion

Both the FBI Most Wanted API and the Premier League ranking table presented unique challenges and learning opportunities in data scraping and processing.

Working with the FBI Most Wanted API highlighted the importance of handling inconsistent and incomplete data. Null fields, irregular formats, and rate limits emphasised the need for robust data cleaning. Saving data locally for analysis proved to be a useful workaround for overcoming rate limits, enabling further exploration of this dataset.

The Premier League ranking data scraping project demonstrated the utility of web scraping techniques to extract meaningful insights from structured HTML tables. Identifying key elements using tools like the browser inspector and parsing the data into usable forms provided a practical application of web scraping. The ability to derive insights, such as the bottom-ranked teams and top attendance figures, showcased the power of automation in extracting relevant data for analysis.