

tasks

May 30, 2023

1 Tasks

```
[1]: from ProcessGameState import Coordinate, ProcessGameState
import numpy as np
import seaborn as sns

[2]: path = "../data/game_state_frame_data.parquet"
boundaries = [Coordinate(-1735, 250), Coordinate(-2024, 398), Coordinate(-2806, 742),
               ↪ Coordinate(-2472, 1233),
               Coordinate(-1565, 580)]
```

1.1 Extract and process data

```
[3]: game = ProcessGameState(path)
game.processWeaponClasses()
game.processWithinBounds(boundaries)
```

1.2 2a

```
[4]: t2TInBounds = game.getSideTeamWithinBounds('T', "Team2")
t2Total = game.getSideTeam('T', "Team2")
print(f"{t2TInBounds.shape[0]/t2Total.shape[0] * 100}%")
```

0.04647749510763209%

With a 0.046% chance, it seems unlikely for Team2 on terrorist side to enter via the light blue boundary.

However, this analysis is incomplete as this is just the percentage of time that Team2 spends within the light blue boundary on the terrorists side. More analysis, including the coordinates and boundaries of areas outside the blue box itself is required to determine whether they entered through the light blue boundary, as opposed to from the bottom of the map (such as through the edges 5-6 or 8-9).

1.3 2b

```
[5]: import pandas as pd

t2TInArea = game.getSideTeamInArea('T', "Team2", "BombsiteB")
ticks = t2TInArea["tick"]
time = []
for tick in ticks:
    rows = game.getTick(t2TInArea, tick)
    weapons = rows["weaponClasses"].apply(pd.Series).stack()
    if weapons.value_counts().get("Rifle", 0) + weapons.value_counts().
↳get("SMG", 0) >= 2:
        time.append(rows.iloc[0]["seconds"])

print(f"{np.mean(time)}s")
```

32.875s

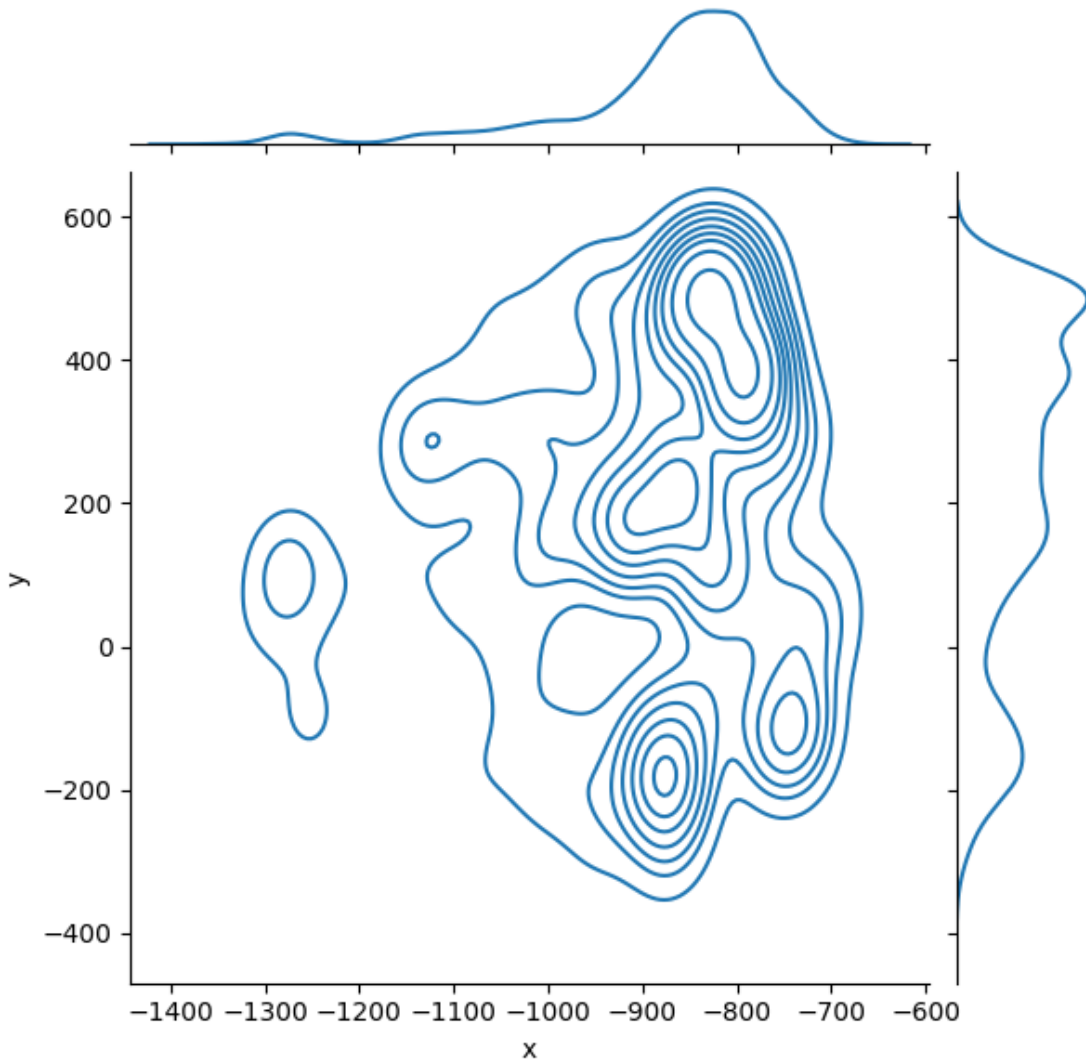
The average timer that Team2 on terrorist side enters Bombsite B with at least 2 rifles or SMGs is 32.875 seconds.

However, this also simply determines the average times that Team2 on terrorist side spends in Bombsite B – it is not able to determine whether they just entered the area depending on the definition of “enter”. There would be different answers if it were to mean the first time they enter the area during the game, or the first time they just step in or out of the area during the game.

1.4 2c

```
[6]: t2CTInArea = game.getSideTeamInArea("CT", "Team2", "BombsiteB")
sns.jointplot(x=t2CTInArea['x'], y=t2CTInArea['y'], kind="kde")
```

```
[6]: <seaborn.axisgrid.JointGrid at 0x146c63070>
```



From the heatmap, we can see that the players on Team2 are most likely waiting inside the top right area of Bombsite B as counter-terrorists.

1.5 3

I would propose creating a web application to view these – for instance, using React. I believe this would be the best way to display the data, especially where maps and visual data are involved. For instance, I would also be able to overlay the heatmap generated in 2c over the correct areas of the map, to better illustrate visual data. Coaching staff should also be able to select values for queries interactively, rather than having to type actual data queries in code or having to ask engineers. In turn, the website should be able to generate the correct answers from the data and provide a definitive response to the coaches and players. I think that I would definitely be able to create a fullstack application with a frontend where coaches can create and submit queries to get responses from the data within the span of a week.

However, a drawback is that some results are definitely open-ended in terms of querying the data. Manual oversight would still be required to draw interpretations that convert trends and data into actual coaching advice and actionables for players.