**MATH 7241: Probability and Statistics**
**Tamanna Urmi | Nov 22, '22**
**Instructor: Prof. Christopher King**

The city of Baltimore collects and publicly shares the victim-based data for crime in the city updated monthly (source: link). For this project, I will be using this data from 2011-2016 to examine if the crime types that take place over time follow a Markov chain.

## Data preparation

The raw data had 285,807 rows and 14 rows.
There were 9 types of crimes with different number of subtypes for each category. The subtypes were merged and pooled to the broader crime type converting this to a 9 state (crime types) timeseries.

| Crime Type | Crime code | Aggregated new Crime Code |
|---|---|---|
| AGG. ASSAULT | 4A, 4B, 4D, 4C | AA |
| COMMON ASSAULT | 4E | CA |
| HOMICIDE | 1K, 1O | HC |
| ROBBERY - STREET | 3AF, 3AK, 3NF, 3AO, 3NK, 3NO | RS |
| SHOOTING | 9S | ST |
| RAPE | 2A | RP |
| ROBBERY - COMMERCIAL | 3CK, 3CF, 3GF, 3EK, 3EF, 3GK, 3CO, 3LF, 3LO, 3GO, 3EO, 3LK | RC |
| ROBBERY - RESIDENCE | 3JF, 3JO, 3JK | RR |
| ROBBERY - CARJACKING | 3AJF, 3AJO | RCa |

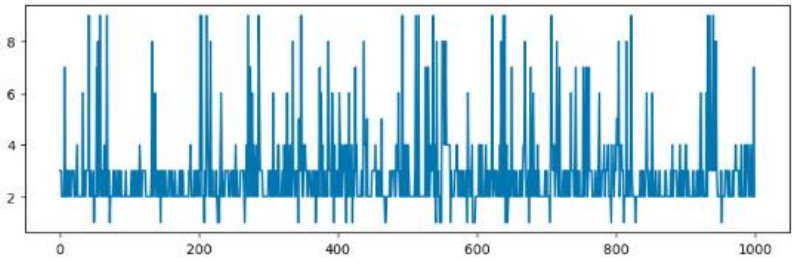
However, the dataset had a lot of null values for different columns which were dropped. Approximately 188k rows were dropped producing a final dataset with 96, 941 rows. The timestamps are mapped to step numbers and the new crime code and neighborhood of crime are converted to integer representation of categorical values.

| steps | CrimeDate | CrimeTime | CrimeCode | CrimeCode_new | Neighborhood | CrimeCode_cat | Neighborhood_cat |
|---|---|---|---|---|---|---|---|
| 0 | 01/01/2011 | 00:05:00 | 4B | AA | Frankford | 3 | 7 |
| 1 | 01/01/2011 | 00:15:00 | 4D | AA | Inner Harbor | 3 | 40 |
| 2 | 01/01/2011 | 00:20:00 | 4C | AA | Windsor Hills | 3 | 49 |
| 3 | 01/01/2011 | 00:30:00 | 4E | CA | Inner Harbor | 2 | 40 |
| 4 | 01/01/2011 | 00:39:00 | 4E | CA | Remington | 2 | 21 |

## Time series of crime type

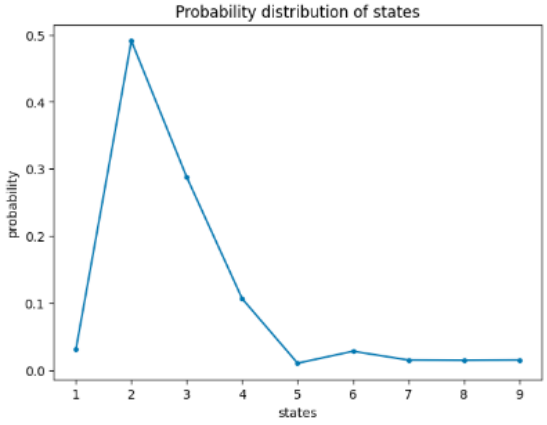
Entire duration:

1000 timesteps:

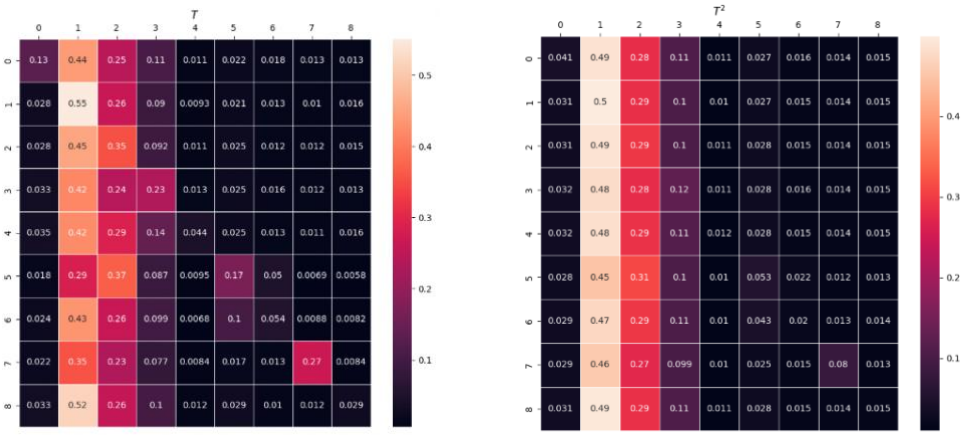



**Analysis**

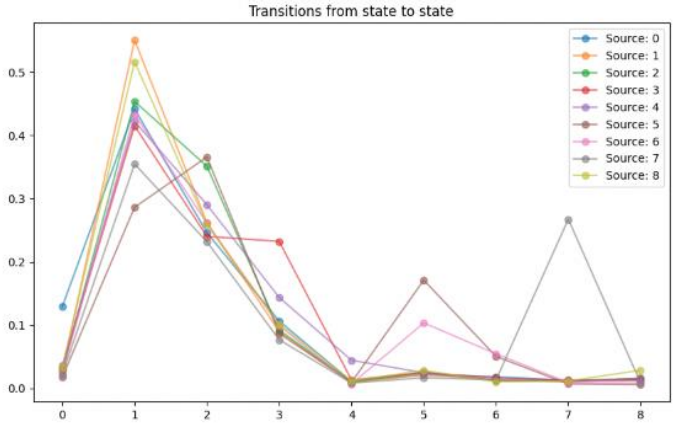
Crime type state frequency:




The jumps from state to state is then converted to edges on a graph/transition to calculate the frequency of each jump. The transition count is then divided by the total number of outward transitions from that state to find the transition probability:



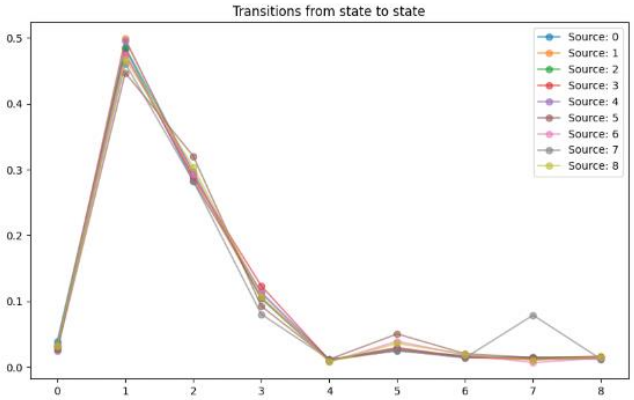
The probability distribution of going to each of the destination from a particular source is as follows:



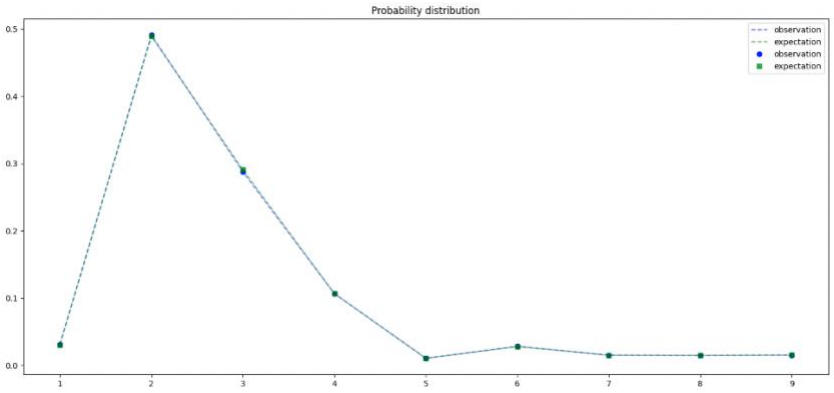

The stationary distribution of the transition matrix obtained by comparing $T^{10}, T^{100}$ and $T^{1000}$ is:

```
[0.031 0.491 0.288 0.107 0.01 0.028 0.015 0.015 0.015]
```

Simulation:




From visual inspection the simulated and the observed time series looks like they have almost the same probability distribution function.




**Model evaluation**

In order to evaluate the model, the observed jump frequencies from a 2-step transition will be compared to the expected jump frequency from a 2-step transition matrix.

Observed = $N_{i\to*\to j}$

Pseudo-code:

```
Two_step_jump_tuple = [] # empty list
for element in (length of timeseries – 2):
    jump = (timeseries[i], timeseries[i+2]) # find state i and j of jump
    Two_step_jump_tuple.append(jump) # append to the jump list

Count frequency of each tuple
```

Expected = $(T^2)_{ij} \sum N_i$

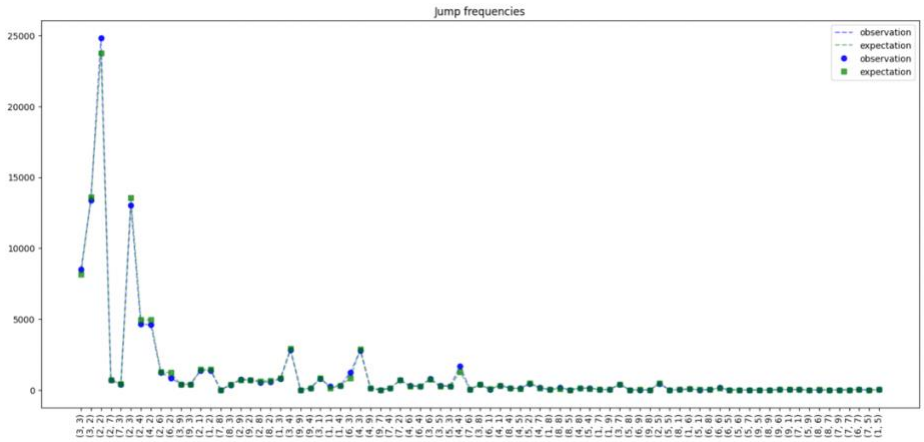
Pseudo-code:

```
edge_frequency_matrix = empty matrix

for row in edge_frequency_dataframe:
    edge_frequency_matrix [source-state][destination-state] = frequency # the
states determine position in matrix

expected_frequency = empty list
for row in row_length_of_transition_matrix:
    deg = sum(row) # number of jumps that exit the state corresponding to row
number
    expected_frequency.append(T^2[row]*deg)
```

Observed and expected frequencies (snapshot):

| | jump | freq_obs | freq_exp |
|---|---|---|---|
| **0** | (3, 3) | 8499 | 8178.090523 |
| **1** | (3, 2) | 13405 | 13623.243634 |
| **2** | (2, 2) | 24818 | 23746.194505 |
| **3** | (2, 7) | 708 | 703.989391 |
| **4** | (7, 3) | 423 | 431.200623 |
| **...** | ... | ... | ... |
| **76** | (7, 9) | 23 | 20.750251 |
| **77** | (7, 7) | 28 | 28.735277 |
| **78** | (6, 7) | 39 | 59.470073 |
| **79** | (7, 5) | 14 | 14.974349 |
| **80** | (1, 5) | 40 | 31.828450 |

The following image is visually representing the two frequencies of comparison.




Goodness of fit = $\frac{\sum(Obs_{ij}-Exp_{ij})^2}{Exp_{ij}}$ = 861.76

Degrees of freedom = # parameters to estimate - # constraints
= 81 – 9
= 72

Decision rule: If goodness of fit > $\chi_{1-0.05,\ 72}$ , $H_0$ will be rejected.

Decision:
From the Chi-squared distribution table, it's obtained that $\chi_{1-0.05,\ 72} = 53.462$. Since the goodness of fit value is larger than the Chi-squared value, the hypothesis is rejected. Markov chain is not a good model for this timeseries of crimes in City of Baltimore because the observed and expected frequency of the two-step transition doesn't match well enough to pass the goodness of fit test.