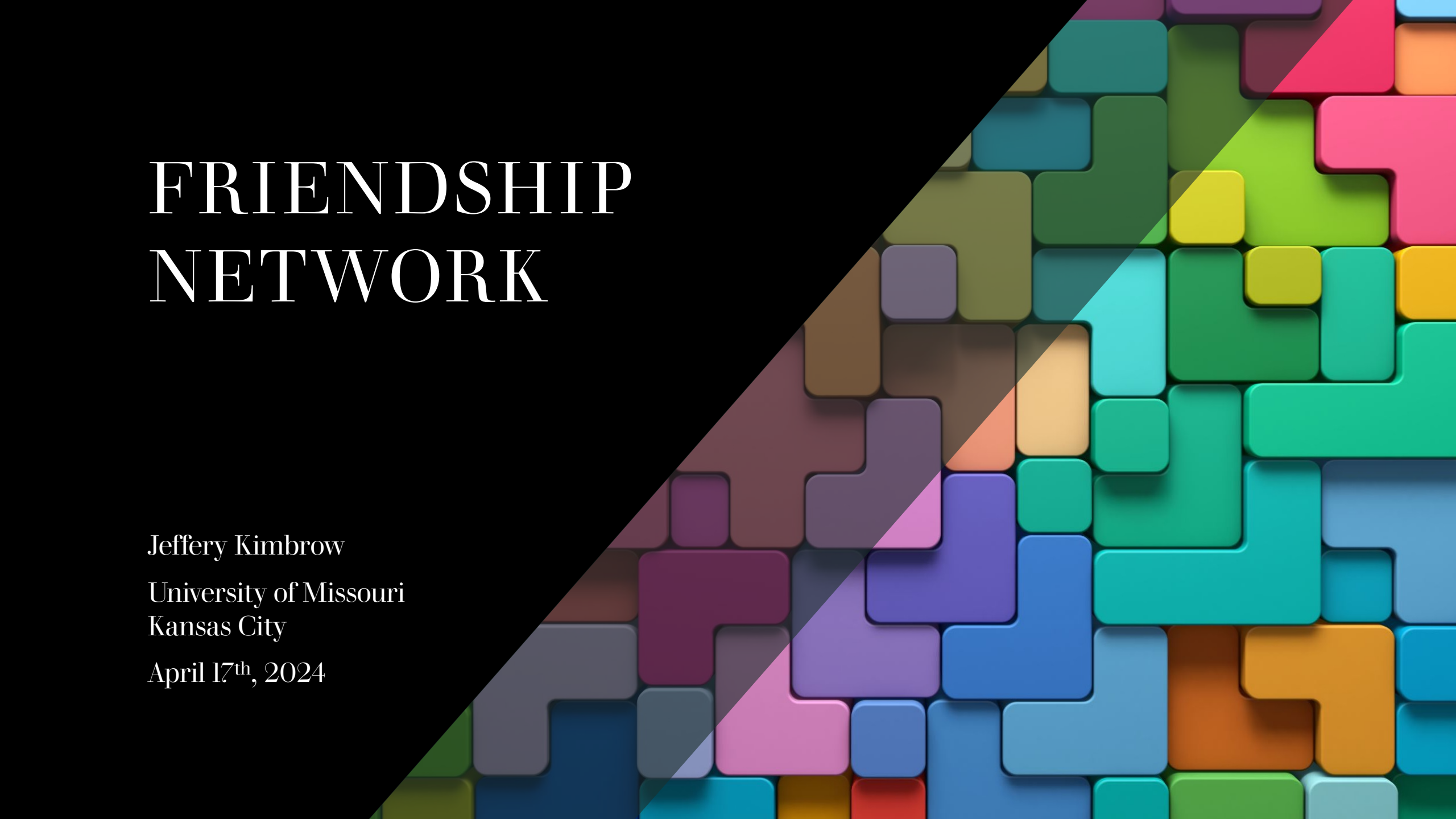
FRIENDSHIP NETWORK

Jeffery Kimbrow

University of Missouri
Kansas City

April 17th, 2024



Agenda

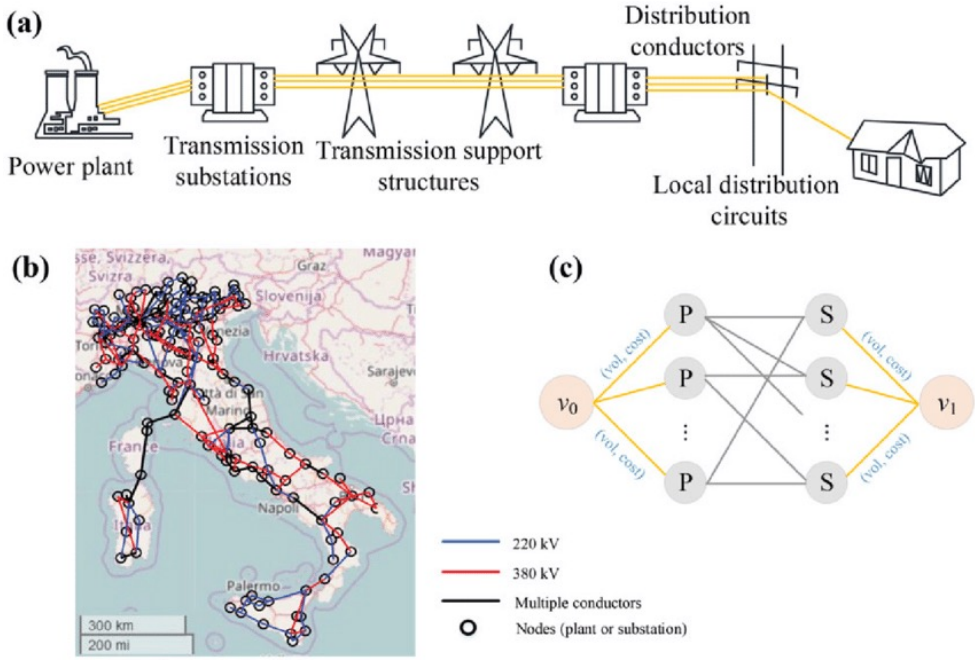
- INTRODUCTION
- IMPORTANCE
- METHODOLOGY
- IMPLEMENTATION
- RESULTS
- DISCUSSION
- CONCLUSION
- REFERNCES

Introduction

- The purpose of this project was to represent friendship networks through matrix representation.
- Objectives:
  - *Apply a matrix representation for the analysis of friendship networks.*
  - *Calculate and interpret the metrics for the network.*

Objectives:

- Showcase the ability of the Python Libraries, NumPy NetworkX, and Matplotlib, to create, analyze, and visualize a friendship network.



Examples:

- Traffic Routing
- Disease Tracking
- Power Grid Management

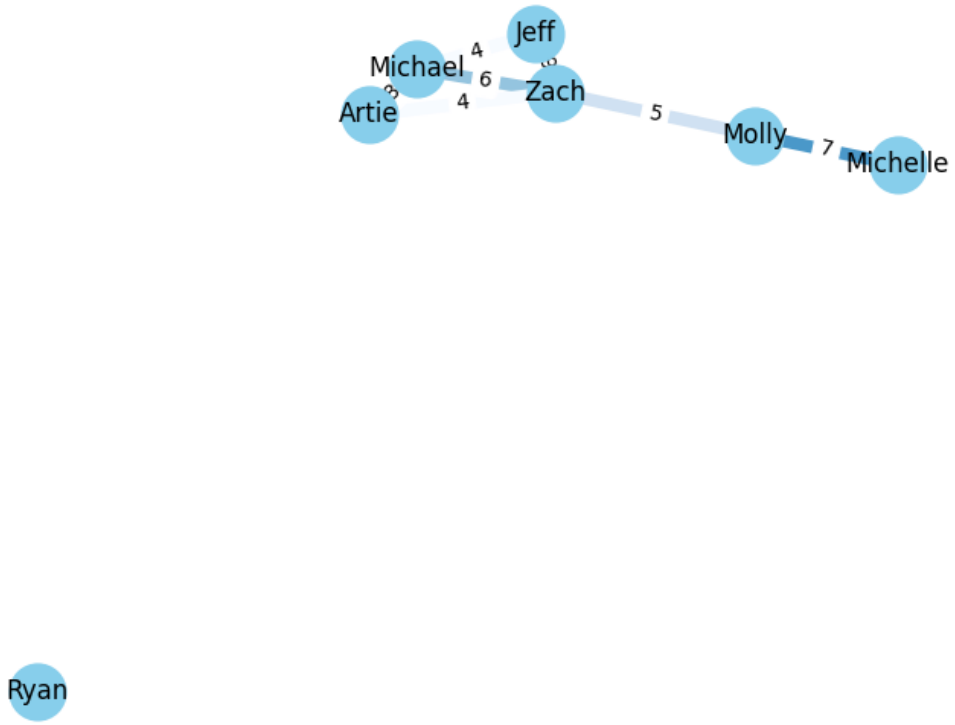
How can graph theory be used to analyze complex networks?

Methodology

- Used a matrix representation to denote friendship strengths and connections, transformed into a graph for analysis and visualization.

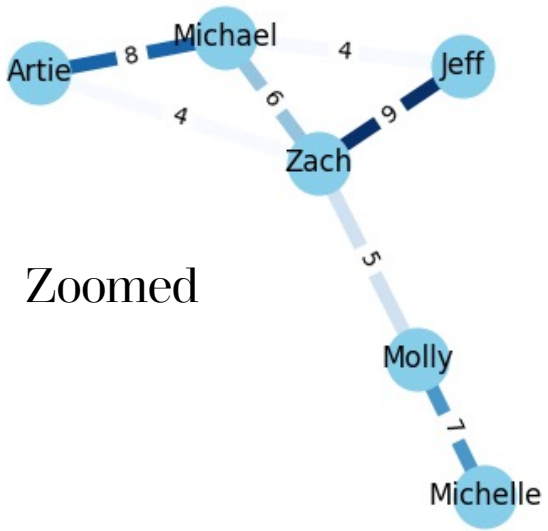
Tools used:

- Python 3.12
- NumPy
- NetworkX
- Matplotlib



- Weights represent the strength of the friendships.
- Configure the matrix to represent the connections.
- Use NetworkX to convert the matrix to graph format.
- Matplotlib was used to create the visual representation of the network.

Implementation



Zoomed

- Graph that visually represents the friendship network.
- Most Popular Person: Zach
- Isolated individuals: Ryan
- Total Unique Friendships: 7
- Average Friendship Strength: ≈ 6.14

```
PS C:\Users\kimbr.KIMBROWSLICE\OneDrive\Desktop\School\DiscreteStructures291> py FriendNetwork.py
Total Friendships: 7
Friendship Pairs (and strengths): [('Jeff', 'Zach', 9), ('Jeff', 'Michael', 4), ('Zach', 'Molly', 5), ('Zach', 'Michael', 6), ('Zach', 'Artie', 4), ('Molly', 'Michelle', 7), ('Michael', 'Artie', 8)]
Most Popular Person: Zach
Isolated Individuals: ['Ryan']
Average Strength of Friendships: 6.142857142857143
PS C:\Users\kimbr.KIMBROWSLICE\OneDrive\Desktop\School\DiscreteStructures291>
```

Results

Discussion

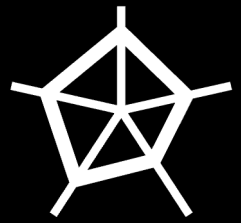
Conclusion

- Project Achievements:

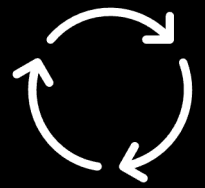
  - Successfully created a visual representation of a friendship network using Python, NumPy, NetworkX, and Matplotlib to represent a network using Graph Theory.

- The importance:

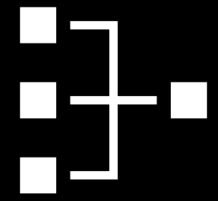
  - The results of this project represent, at a low level, the complexity required to represent a friendship network, among multiple different real-world applications with Graph Theory.



Networkx Graph Tutorial



NumPy. Matrix



Matplotlib.pyplot.plot

References

THANK YOU!

Friendship Network

Jeffery Kimbrow

Discrete Structures II 291

Professor Wajeb Gharibi

April 11th, 2024

Table of Contents

1. Introduction

This project explores the analysis of friendship networks through matrix representation. Utilizing a matrix to denote friendships and their strengths, we convert this representation into a graph to examine various network characteristics. This methodology allows for an insightful exploration into network dynamics, including connectivity and relationship intensities.

2. Objectives

- To apply matrix representation for the analysis of friendship networks.
- To calculate and interpret network metrics such as total friendships, most popular individuals, isolated members, and average friendship strength.
- To demonstrate the utility of ***NumPy***, ***NetworkX, and Matplotlib*** in a social network analysis.

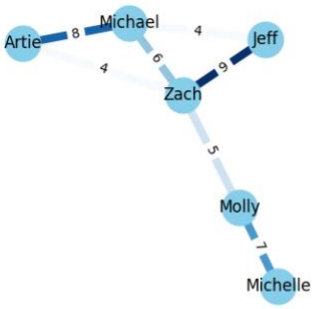
3. Methodology

The methodology involves representing friendship networks as matrices using the **NumPy** library in Python. These matrices are then converted into graphs with the **NetworkX** library for analysis. From that analysis, **Matplotlib** allowed the visual representation of the graph. This approach facilitates the calculation of key network metrics, providing insights into the social structure represented by the network.

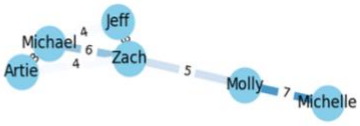
4. Program Description

The program uses a matrix where zeros indicate no friendship and integers (1-10) represent the strength of friendships. This matrix is then transformed into a graph to analyze the network. Key metrics such as total friendships, the most popular person, isolated individuals, and average friendship strength are calculated to evaluate the network's characteristics.

5. Results Analysis



- **Total Friendships:** 7, indicating the presence of seven distinct friendship connections within the network.
- **Most Popular Person:** Zach, who has connections with four other individuals (Jeff, Molly, Michael, Artie), making him the most central node in this network.



- **Isolated Individuals:** Ryan, who does not have any connections with other members, highlighting a completely detached individual within the network.
- **Average Strength of Friendships:** Approximately 6.14, suggesting moderately strong relationships among the network members overall.

These results provide a comprehensive view of the network's structure, emphasizing Zach's pivotal role and the relatively strong bonds formed between members.

Below confirm the results of the program running.

```
PS C:\Users\kimbr.KIMBROWSLICE\OneDrive\Desktop\School\DiscreteStructures291> py FriendNetwork.py
Total Friendships: 7
Friendship Pairs (and strengths): [('Jeff', 'Zach', 9), ('Jeff', 'Michael', 4), ('Zach', 'Molly', 5), ('Zach', 'Michael', 6), ('Zach', 'Artie', 4), ('Molly', 'Michelle', 7), ('Michael', 'Artie', 8)]
Most Popular Person: Zach
Isolated Individuals: ['Ryan']
Average Strength of Friendships: 6.142857142857143
PS C:\Users\kimbr.KIMBROWSLICE\OneDrive\Desktop\School\DiscreteStructures291>
```

6. Conclusion

The findings from this project illustrate a dynamic friendship network characterized by a variety of relationship strengths and one isolated member. The prominent role of Zach as a central figure demonstrates the effectiveness of using matrix representation to uncover key social dynamics within a network. While the results shed valuable light on the network's structure, they also highlight the simplicity of the model and the complexity of real-world social dynamics that might not be fully captured by this analysis.

7. References

- NumPy Documentation
- Networkx Documentation
- Matplotlib Documentation