**Mohammed Raza Syed: SimPPL Visualization report**

**Connected Network of users quoting other users tweets and posts neo4j**A group of orange dots

Description automatically generated

**Connected Network of user replying to other users posts and tweets neo4j**A group of orange dots

Description automatically generated

**Connected Network of user retweeting to other users posts and tweet**

**Neo4j**A group of orange dots

Description automatically generated

**Users following each other in a network graph using neo4j**

A diagram of a network

Description automatically generated

**A graph of numbers and letters

Description automatically generated with medium confidence**

**A graph with a bar and a red line

Description automatically generated with medium confidence**

**A graph with text overlay

Description automatically generated**

**A graph with numbers and a bar chart

Description automatically generated with medium confidence**

**A graph with numbers and lines

Description automatically generatedA graph of different states

Description automatically generated**

**A computer screen shot of a computer screen

Description automatically generated**

**A screenshot of a computer screen

Description automatically generated**

**A screenshot of a computer screen

Description automatically generated**

**A graph showing a blue and green rectangle

Description automatically generated**

**Neo4j Code**

*# Function to create nodes and relationships*

*def* create\_neo4j\_graph(*tx*, *source\_id*, *target\_id*, *relationship\_type*):

*# Create a node for the source\_id*

    tx.run("MERGE (source:Node {id*:* $id})", *id=*source\_id)

*# Create a node for the target\_id*

    tx.run("MERGE (target:Node {id*:* $id})", *id=*target\_id)

*# Create a relationship only when 'type' is "quoted"*

*if* relationship\_type *==* "quoted":

        tx.run("""

            MATCH (source:Node {id*:* $source\_id}), (target:Node {id*:* $target\_id})

            MERGE (source)-[rel:QUOTED]->(target)

        """, *source\_id=*source\_id, *target\_id=*target\_id)

*if* relationship\_type *==* "replied\_to":

        tx.run("""

            MATCH (source:Node {id*:* $source\_id}), (target:Node {id*:* $target\_id})

            MERGE (source)-[rel:REPLIED\_TO]->(target)

        """, *source\_id=*source\_id, *target\_id=*target\_id)

*if* relationship\_type *==* "retweeted":

        tx.run("""

            MATCH (source:Node {id*:* $source\_id}), (target:Node {id*:* $target\_id})

            MERGE (source)-[rel:RETWEETED]->(target)

        """, *source\_id=*source\_id, *target\_id=*target\_id)

*# Assuming 'new\_df' contains the DataFrame with 'id', 'referenced\_tweet\_id', and 'type'*

*for* \_, row *in* new\_df.iterrows():

*# Establish a connection to Neo4j*

*with* GraphDatabase.driver(uri, *auth=*(username, password)) *as* driver:

*# Create a session and run the graph creation transaction*

*with* driver.session() *as* session:

            session.write\_transaction(create\_neo4j\_graph, row['id'], row['referenced\_tweet\_id'], row['type'])