Using Knowledge Graph to Identify Relationships Between People

Rebecca Noy
University of California, Los Angeles
rebecca.noy@ucla.edu

May 31, 2024

Faculty Advisor: Dr. Shuyang Sheng

Abstract

In complex communities, identifying meaningful relationships and key organizations can be challenging. By using technologies like web-scraping, network analysis, and regression analysis, we can better uncover and understand these connections. Consider, for example, *Putin's List*, a website profiling individuals linked to the Putin regime. The website has only short biographies of people, but the information likely hides the wealth of implicit connections between them. Automatically extracting relationships from this unstructured information can help any analytics team gain insight into connections that might not be obvious initially. They can find connections between people and organizations and test hypotheses about influence, power, and reach. In this work, we automatically scrape *Putin's List* to obtain structured data suitable for analysis. We apply a set of machine-learning tools to extract from their biographies the organizations individuals are associated with and the crimes that they are accused of. With that information, we create network graphs, knowledge graphs, and bipartite graphs to highlight hidden relationships in the data set. Subsequently, we perform regression analysis to measure the impact certain factors and organizations have on driving an individual to be accused of crimes of specific types and severity. With the combination of visual and quantitative results, we identify the most influential people and organizations and understand the most influential factors in determining crime severity. While outside of the scope of this work, supplementing this analysis with criminal expertise would make the analysis more targeted and insightful.

Keywords: *Knowledge graph, Bipartite graph, Networks, Web-Scraping, Named Entity Recognition, N-gram tokenization, Binary Variable Formation, Ordinary least squares regression, Logit Regression, Crime, Suspect list*

I. Introduction

In many complex communities, it may be hard to identify relationships between individuals and to understand which organizations and connections are the most meaningful to a specific outcome. Using technologies like web scraping to structure information for analysis, network analysis to visualize connections, and regression analysis to quantify importance, we can more easily uncover these connections. As an example community, we use *Putin's List,* a website featuring profiles of individuals identified as having connections to the Putin regime.[1] The website was created by the Free Russia Forum in order to "collect and systemize data on crimes and those who use their connections to the government to evade legal responsibility." [1] The website aims to be a resource and a place for "public defamation of people responsible for the destruction of freedom in Russia". [1] *Putin's List* website is run by volunteers and compiles information from open sources.

There are 1,676 profiles on the site. Each profile contains a person's name, date of birth, citizenship, professional biography, information of what they are accused of, and their "criminal category." The criminal categories include power-holders, executors, law-destroyers, aggressors, beneficiaries, oligarchs and corrupt officials, propagandists, accomplices, and any combination of these categories.[2]

This project systematically scrapes the Putin's List website to create a data frame for data analysis. Machine learning tools extract information from biographical descriptions to create insightful variables, such as organizational affiliations and crimes that individuals are accused of. Network analysis then helps us visualize hidden relationships in the data. Finally, regression analysis not only tests hypotheses regarding the impacts of influence, reach, and power, but also quantifies these effects.[3] The Putin's List website resembles a criminal suspect list; therefore, we can envision that our analysis would be a helpful tool for someone with criminal-justice expertise. Our techniques can create a powerful investigative tool for subject matter experts.

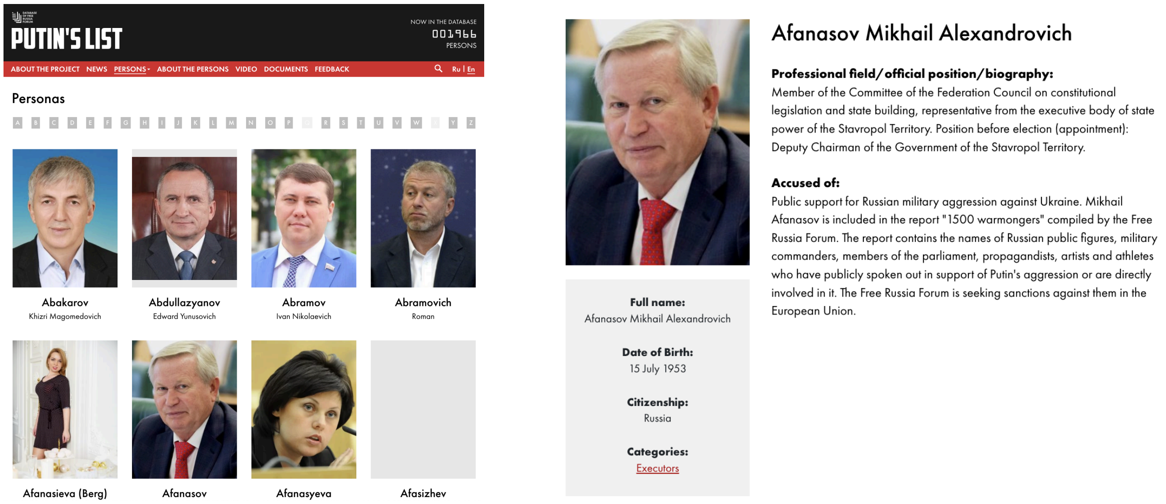
[1] Putin's list website: https://www.spisok-putina.org/en/personas/
[2] Information on what these categories mean can be found here: https://www.spisok-putina.org/en/about-the-project/
[3] Link to the full workbook with all of the results and analysis: https://shorturl.at/LMGd2

II. Data

A. Web Scraping

Biographical information on the *Putin's List* website is captured in unstructured paragraphs. See *Figure 1* for an example of a website page and a biographical page. Our first task is to scrape this data into a structured table, such that each row represents a person and each column represents a piece of biographical information: Date of Birth, Citizenship, Professional Biography, Accusations, etc.

Figure 1: Screenshot of Putin's List Website and a profile of an individual



Afanasov Mikhail Alexandrovich

Professional field/official position/biography:
Member of the Committee of the Federation Council on constitutional legislation and state building, representative from the executive body of state power of the Stavropol Territory. Position before election (appointment): Deputy Chairman of the Government of the Stavropol Territory.

Accused of:
Public support for Russian military aggression against Ukraine. Mikhail Afanasov is included in the report "1500 warmongers" compiled by the Free Russia Forum. The report contains the names of Russian public figures, military commanders, members of the parliament, propagandists, artists and athletes who have publicly spoken out in support of Putin's aggression or are directly involved in it. The Free Russia Forum is seeking sanctions against them in the European Union.

Full name:
Afanasov Mikhail Alexandrovich

Date of Birth:
15 July 1953

Citizenship:
Russia

Categories:
Executors

We use a Python script that utilizes a combination of basic packages, regular expressions, and BeautifulSoup, a package designed for parsing HTML and XML documents.[4] The Python script collects all the URLs for each person's profile, then goes through each URL and extracts the necessary personal information. In the end, it returns a completed data frame with each row representing a person.

There are 142 pages of people, with 12 people per page. In order to retrieve the URL for each individual's profile we parse the HTML of each page of people. Individual URLs follow the same structure *(https://www.spisok-putina.org/en/personas/[person name]/)*. Therefore we use regular expressions to extract each individual URL from the HTML of the pages of profiles.[5] The

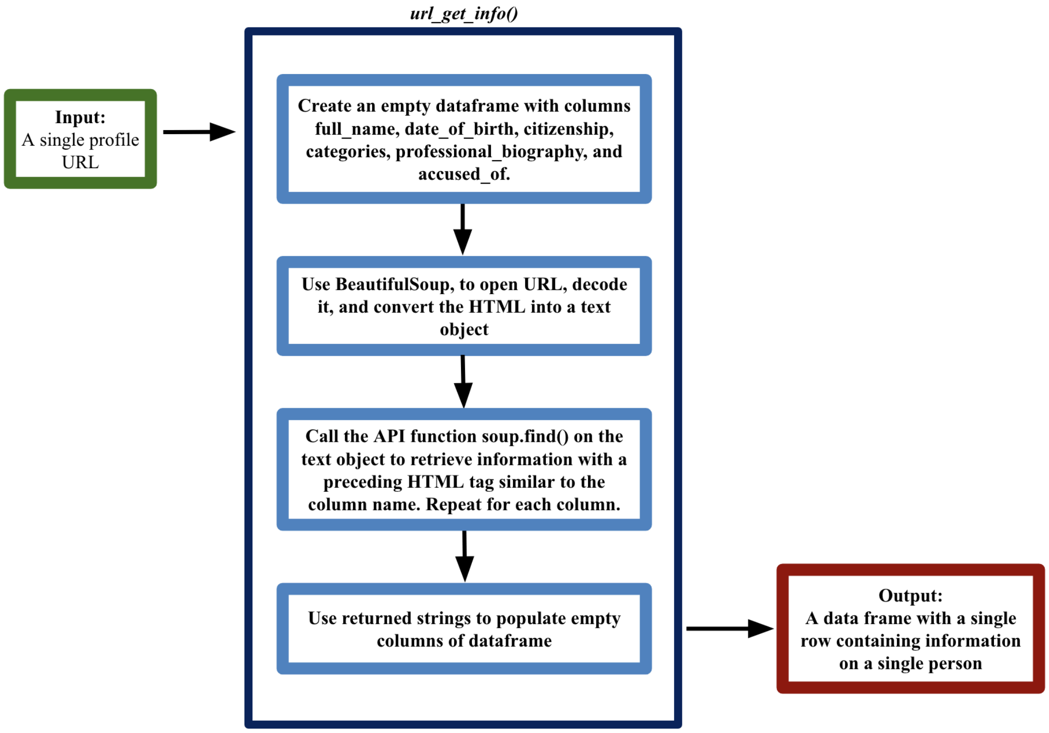
[4] Documentation for BeautifulSoup: https://pypi.org/project/beautifulsoup4/

[5] The regular expression used: <a class="preview-material__link" href="(https:\/\/www\.spisok-putina\.org\/en\/personas\/[a-zA-Z1-9\-]+\/)" rel="bookmark">

regular expression looks for the HTML tag denoting a URL. A capture group is specified purely around the URL, so that we match the entire pattern but only retrieve the necessary text.

After obtaining a list of URLs for each person's profile, each URL is separately passed into the *url_get_info()* function via a for loop. *Figure 2* explains the *url_get_info()* function and the process to get from a single profile url to a data frame with a single row containing biographical information for an individual.

Figure 2: Explains how the url_get_info function works [6]


url_get_info()
Input:
A single profile URL
Create an empty dataframe with columns full_name, date_of_birth, citizenship, categories, professional_biography, and accused_of.
Use BeautifulSoup, to open URL, decode it, and convert the HTML into a text object
Call the API function soup.find() on the text object to retrieve information with a preceding HTML tag similar to the column name. Repeat for each column.
Use returned strings to populate empty columns of dataframe
Output:
A data frame with a single row containing information on a single person

After we create the data frame, we use a Python script to perform cleaning tasks such as standardizing the citizenship column (e.g., 'Russia' and 'Rusiija' should both have the citizenship value 'Russia'). We then export the data frame as a CSV file and import it into Google Sheets.[7] After extra commas and spaces are removed, it is imported back into Python under the name *putins_list*.

[6] For full code defining the function see Snippet A.1 and A.2

[7] Link to full data frame: https://shorturl.at/izLO4

B. Named Entity Recognition

Now that we have a data frame representing the structure of the individual descriptions, we can do further analysis to extract information from the biographical paragraphs. We need to extract the organizations each person is associated with because the end goal is to understand the interactions between individuals based on their professional affiliations.

To identify the organizations present in the professional biographies, the Python script utilizes a package called SpaCy, which performs Named Entity Recognition (NER).[8] NER locates named entities in structured text and then classifies them into predefined categories such as organizations, people, countries, monetary values, etc.[9]

The script converts the entire professional biography column into a string of text (Snippet A.3, line 2). It then applies the *nlp()* function to the string to create a doc object, enabling the extraction of named entities (Snippet A.3, line 3). The resulting object is saved to a variable. The script focuses on the following categories: organizations, significant political parties, sanctions lists, and regions. Therefore, the script extracts only entities tagged as 'ORG', 'PRODUCT', 'EVENT', 'FAC', 'LAW', and 'NORP' (Snippet A.3, line 6).[10] After all of the entities are extracted they are saved to a list variable (Snippet A.3, lines 4 and 7).

The package effectively identifies all entities, however, many entities are extracted multiple times under slightly different names (*Table 1*). To address this issue, we export a list of all named entities to Google Sheets. We manually review, standardize, and categorize each entity. Although the script identified 900 entities, we reconcile them to only 321 unique organizations with multiple people associated with them.[11] See *Table 1* for examples.

[8] Python documentation for SpaCy: https://spacy.io/

[9] More information on Natural Language Processing found here: https://realpython.com/natural-language-processing-spacy-python/

[10] ORG: Names of companies, agencies, institutions, etc. PRODUCT: Names of products or items. EVENT: Names of events, such as sports events, conferences, etc. FAC: Names of facilities like buildings, airports, highways, bridges, etc. LAW: Names of laws, statutes, regulations, etc. NORP: Nationalities or religious or political groups

[11] Organization types: Award, Bank, Company, Company (Oil), District Court, Federation Council, Gang, Government Agency, Lobby/Union, Media/TV, Military Academy, Military Agency, Military District, Outside Gov, Outside Org, Political Party, Region, Religious, Sanctions, Security Agency, Sports, Theatre, University, War On Ukraine, and Other

Table 1: Example of manual standardization in Named Entity Recognition

SpaCy's Extraction *(variable: appears_as)*	**Standardized Name** *(variable: binary_var_match)*	**Type of Organization**
Armed Forces of the Russian Federation	Armed Forces of the Russian Federation	Military Agency
Armed Forces	Armed Forces of the Russian Federation	Military Agency
Russian Armed Forces	Armed Forces of the Russian Federation	Military Agency
RF Armed Forces	Armed Forces of the Russian Federation	Military Agency
Main Military-Political Directorate	Armed Forces of the Russian Federation	Military Agency
General Staff of the Armed Forces	Armed Forces of the Russian Federation	Military Agency

After creating a data frame, in google sheets, of the 321 unique organizations, with columns for the extracted entity, the standardized name, and the organization type, the data frame is imported into Python.[12] The variable containing SpaCy's extraction is called "*appears_as*" and the variable indicating the standardized name, which will become the binary variable, is called "*binary_var_match*". In the main data frame, *putins_list*, Python generates a column for each standardized entity and initially fills it with 0 to prepare for binary variable formation.

The script iterates through each row of *putins_list*. In each row, within the *professional_biography* column, the script searches for every "*appears_as*" phrase using regular expressions. If the phrase appears in the professional biography paragraph, we update the corresponding binary variable column to a 1. We update a binary variable cell to 1 if its value had previously been 0, ensuring previously filled information remains unchanged as the script goes through the whole cell (Snippet A.4). See *Table 2* for examples.

[12] Link to list of extracted entities and their corresponding standardized name: https://shorturl.at/kmMQ0

Table 2: Example of two rows in the table corresponding to two individuals after extracting organization affiliations. The actual table has a column for each of the 321 organizations.

Full Name	Professional Biography	Communist Party	Federation Council
Afonin Yuri Vyacheslavovich	Russian politician. **Deputy of the State Duma of the Federal Assembly of the Russian Federation** of the 5th, VI, VII, and VIII convocations. Member of the **Communist Party faction.**	1	1
Agalarov Araz lskender oglu	After receiving his diploma, he worked for six years at a research institute in Baku, and then at the Baku city committee of the **Communist Party**.	1	0

Each row corresponds to a person; it includes their information from the website and whether they are associated with any of the 321 organizations.

Verifying extraction quality: To ensure the script is working properly, we randomly choose a particular organization. Using Google Sheets, we search the professional biography column for all possible names of that organization. We count the unique rows where one of these names appears and confirm that this total matches the sum of the organization's binary column. By performing this process for a subset of organizations, we gain reasonable confidence that our Python script is correctly populating each row.

C. N-grams

Since the outcome of interest is criminal behavior, it is important to extract what each individual is accused of. The column describing the reasons why individuals are included on the list and the crimes they are accused of is called *accused_of*. To extract specific crimes, we need to process each paragraph in the *accused_of* column. However, we cannot use named entity recognition for this task because the types of crimes are not named entities, unlike organizations and places. Rather, we need to identify actions. We use N-gram tokenization, a method to identify the most common phrases in a string of text, to identify the most common crimes.

First, we convert the *accused_of* column into a string of text. Then we remove all stop words.[13] We pass this string to the *ngram()* function. This function requires a specification for the size of common phrases it will extract; 3-letter phrases, 4-letter phrases, etc. To identify important crimes, the string runs through many sizes of n-grams. A low value of n could identify phrases discussing the same crime multiple times, while a high value of n may be too broad to capture all important crimes. We combine a combination of results from analysis of each size to obtain the most common phrases and their frequencies. It becomes clearer which phrases likely belong together and which ones constitute their own crime category. We identify 11 types of crimes and convert them into binary variables, similar to the organizations. [14] We fill these binary variables with either a 1 or a 0; a 1 indicating the person is accused of the crime and 0 indicating they are not. We apply a lambda function to the *accused_of* string in each row. For each row, if the n-grams indicate the crime appears in the *accused_of* column, we fill the crime column with a 1, otherwise, we fill it with a 0 (Snippet A.5). See *Table 3* for examples.

Table 3: Example of a few individuals's output row after extracting criminal accusations. The actual table has a column for each of the 11 crimes.

Full Name	**Accused of**	**Money Laundering**	**Corruption**
Barsukov (Kumarin) Vladimir	Creation and leadership of the criminal community, **national and cross-border corruption**, **money laundering**, contract killings, and drug smuggling	1	1
Afanasieva (Berg) Yulia	Involvement in **corruption schemes**, disinformation, and malicious operations in Africa and Europe in collaboration with the “troll factory” of Yevgeny Prigozhin.	0	1

A continuous y variable indicating the severity of the crimes accused of can help build regressions and identify factors influencing crime accusations. To create a continuous y variable, we rank the 11 crimes based on severity (Table C.3). In Python, we then create a column for the

[13] Stop words are commonly used words in a language that a search engine is taught to ignore, such as “the”, “is”, “in”, “for”, etc. To identify Python’s list of stopwords run: nltk.download('stopwords') and print(stopwords.words('english'))

[14] The 11 crimes are: Annexation of Crimea, Crimes Related to Ukraine, Election Fraud, Illegitimate and anti-democratic constitutional coup, Political Repression, Propaganda, Money laundering, Illegal enrichment at the expense of Russian taxpayers, Corruption, Organized Crime, Fake news

continuous y variable and iterate through each row. If there is a 1 in the column for a specific crime binary variable, we add its corresponding severity value to the column (Snippet A.6). See *Table 4* for an example. Among those who were accused of one of the 11 identified crimes, the summary statistics for the continuous y variable, describing severity of combined crimes, can be found in Table C.1.

Table 4: An example of crime severity rating calculation for an individual

Full Name	Annexation of Crimea	Corruption	Organized Crime	Crime Severity
Aksyonov Sergey	1	1	1	16
Crime severity rating:	11	3	2	Total: 16

We identified only 11 crimes for this project, as a proof of concept. These are the 11 crimes that were the most common in our data set. However this process can be repeated to identify all of the crimes present in the data, and to create a more comprehensive list of crimes of interest.

D. Variables

We discuss seven categories of variables throughout this paper; we summarize them here. The variables are: demographic and categorical variables, variables we use to create edges and nodes in the network graphs, and variables we use to build regressions with predictive power.[15]

Demographic variables: We include several biographical variables: name, date of birth, age, and citizenship. We also create a standardized citizenship column to handle variations in how citizenships are written, ensuring that occurrences such as “Rusjia” and “Russia” both appear as “Russia.”

Categories: The website creators classify the people on the list into eight categories: power-holders, executors, law-destroyers, aggressors, beneficiaries, oligarchs and corrupt officials, propagandists, and accomplices, including combinations of these categories.[16] For

[15] This is a workbook detailing all of the variables, breaking down their tags, categories, and summary statistics: https://shorturl.at/j7u5S

[16] Information on what these categories mean can be found here: https://www.spisok-putina.org/en/about-the-project/

regression analysis, we convert each category combination into a binary variable, with 1 indicating an individual belongs to that category and 0 indicating they do not.

Paragraphs for processing: We process two types of paragraphs: a professional biography and a description of why the individual is on the list and what they are accused of.

Variables indicating organization affiliation: To indicate organization affiliation, we process the professional biographies to extract the organizations people are associated with. We then turn each identified organization into a binary variable, with 1 indicating affiliation and 0 indicating no affiliation. There are 321 organizations individuals can be associated with, among which there are 25 types. Each type is also converted into a binary variable, with 1 indicating association with at least one organization of that type and 0 indicating no association. We gather summary statistics on the number of people belonging to each organization to identify variables with predictive power during regression analysis. Frequencies for types of organizations are in Table C.2 and summary statistics for variables used in paper regressions are in Table C.1.

Variables indicating crimes people are accused of: To indicate crimes people are accused of, we process the accused paragraph to extract the crimes. We convert each identified crime into a binary variable, with 1 indicating accusation and 0 indicating no accusation. There are 11 crimes of interest and 27 different combinations of accused categories (executor, accomplice, oligarchs and corrupt officials, etc.). Frequencies for types of crimes and associated severity are in Table C.3 and frequencies for accused of category combinations are in Table C.4.

Variable for crime severity: For predicting factors that influence people to be accused of more severe crimes, with a regression analysis, we create a continuous variable for crime severity. We rank each of the 11 identified crimes on a scale of 1-11 to obtain a severity rating. A person's crime severity rating is the sum of the ratings of the crimes they have been accused of. A breakdown of severity ratings is in Table C.3 and summary statistics of crime severity are found in Table C.1.

Degree: We create a network graph among the people in the data set and obtain a measure of influence called degree. Degree represents how many connections a node has: for a person node, it indicates how many organizations that individual is associated with, and for an organization node, it shows how many people are affiliated with that specific organization.

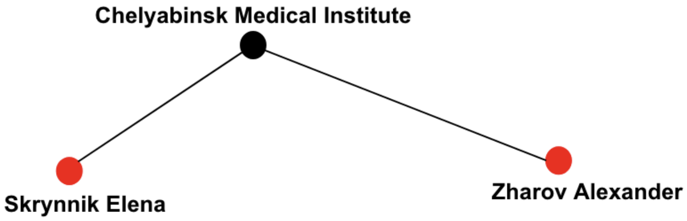
III. Empirical Strategy

A. Building, Visualizing, and Analyzing Networks

Now that we have a data frame representing the structure of the individual descriptions, and binary variables for affiliations and crimes, we can build a network graph among these people. A network graph is a visual representation of interconnected nodes, where each node represents an entity and the edges between them depict their relationships.

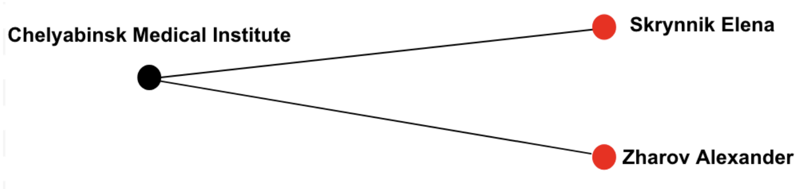
A knowledge graph serves as a valuable tool for visualizing clusters of people associated with each other and uncovering relationships that may not have been initially obvious from the data frame. We utilize the network package, NetworkX, in Python to construct network graphs. To build a knowledge graph: one initializes a graph, adds nodes, draws the edges, and then plots the graph with a chosen layout (Snippet A.7). In this knowledge graph, nodes represent individuals and organizations. Edges are established between a person node and an organization node if the individual has a value of 1 in the binary variable column corresponding to that organization. Node colors are assigned as black for organizations and different colors, determined by citizenship, for individuals. See *Figure 3* for an example.

Figure 3: If only two, people Elena Skrynnik and Alexander Zharov are associated with the Chelyabinsk Medical Institute, the nodes and edges are drawn as follows:



While a knowledge graph visualizes clusters of people and organizations, a bipartite graph more clearly highlights prominent organizations and individuals. It arranges the two types of nodes on opposite sides and connects them with edges, resembling a matching game. To build a bipartite graph in Python one uses the same process as the one to build a knowledge graph, except, the layout is specified as *nx.bipartite_layout().* Since edges can only connect from a person node to an organization node, a bipartite graph is optimal for identifying important organizations rather than analyzing interactions among the whole group. One side of the graph comprises person nodes, while the other side consists of organization nodes. The most popular entities are those organization nodes with the most connections. See *Figure 4* for an example.

Figure 4: Using the same example from above, the bipartite graph would display the same information as follows:



B. Regression Analysis

Regression analysis is used to quantify the effect that organizations, or types of organizations, can have on tendencies to be accused of crimes.

Snippet A.8 shows how we built an ordinary least squares regression in Python. After the model is constructed the coefficients of each variable, and the model statistics are recorded in a google sheet. Many different regressions can be built out to try and identify new, quantifiable insights about effect.

As the independent and dependent variables change, the regression has a different equation. For example, a regression which uses age, degree, and organization type to predict an individual's overall crime severity rating would have the following equation:

$severity_i = B_0 + \acute{B}_1\ degree_i + B_3\ organization\ type_i + B_4\ age_i\ + \varepsilon_i$

As a robustness check, we also consider a logit regression version when we replace the severity rating by a binary variable indicating whether an individual conducts a more severe crime.

IV. Results

A. Networks

While providing only a general picture, a knowledge graph visualizes the data and uncovers hidden connections within the data frame. The complete knowledge graph for this dataset is displayed in *Figure 5.* It shows a large cluster of people in the center with a few smaller clusters around the edges. Clusters on the outskirts contain an organization and a few individuals, indicating isolated pools of interaction. Additionally, some people are scattered around the graph without any organizational ties. Colors indicate citizenship:

Russian: Red

Kingdom of Spain: Purple

Ukraine: Yellow

Syria: Green

USA: Blue

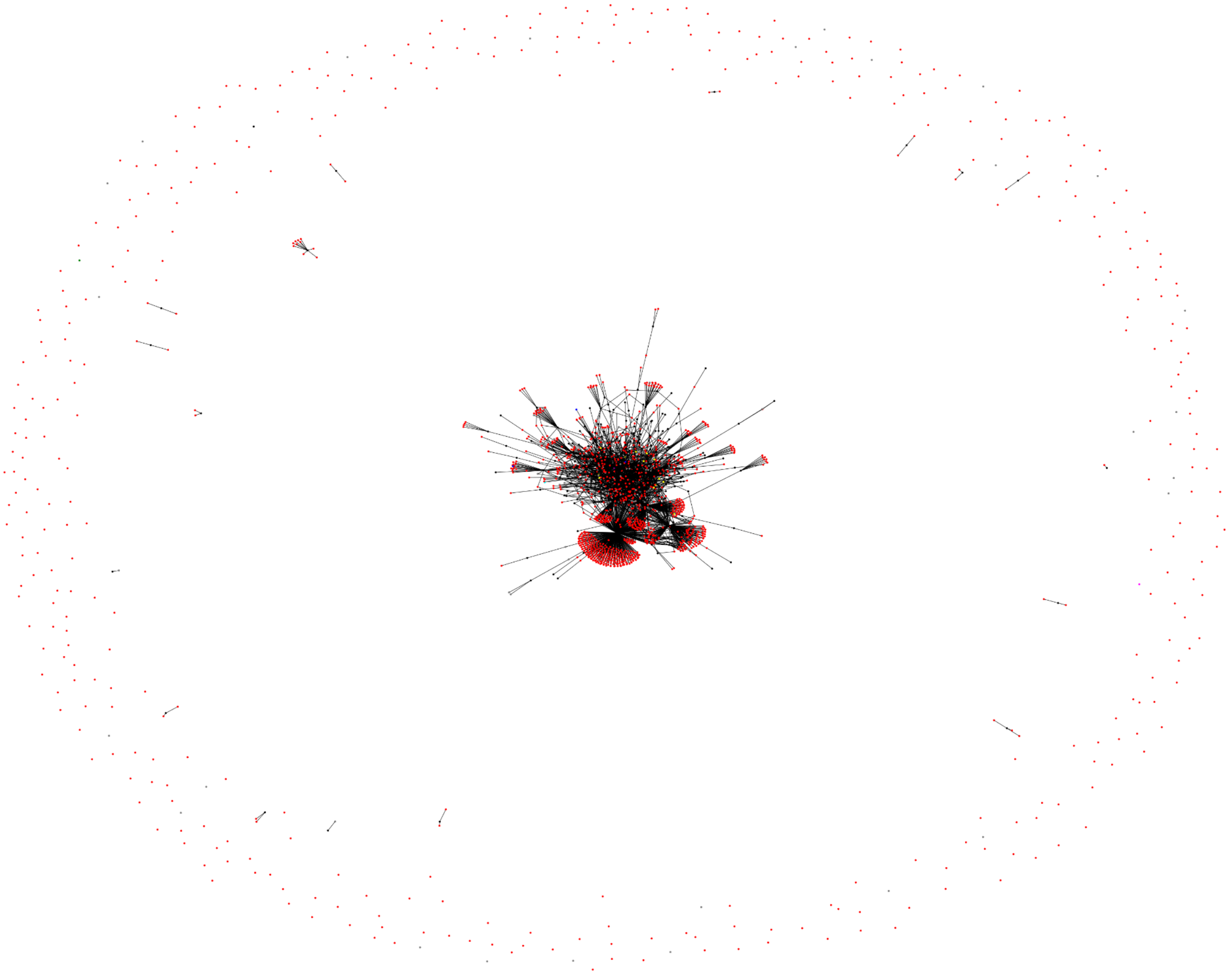
Germany: Olive

France: Orange

United Kingdom: Cyan

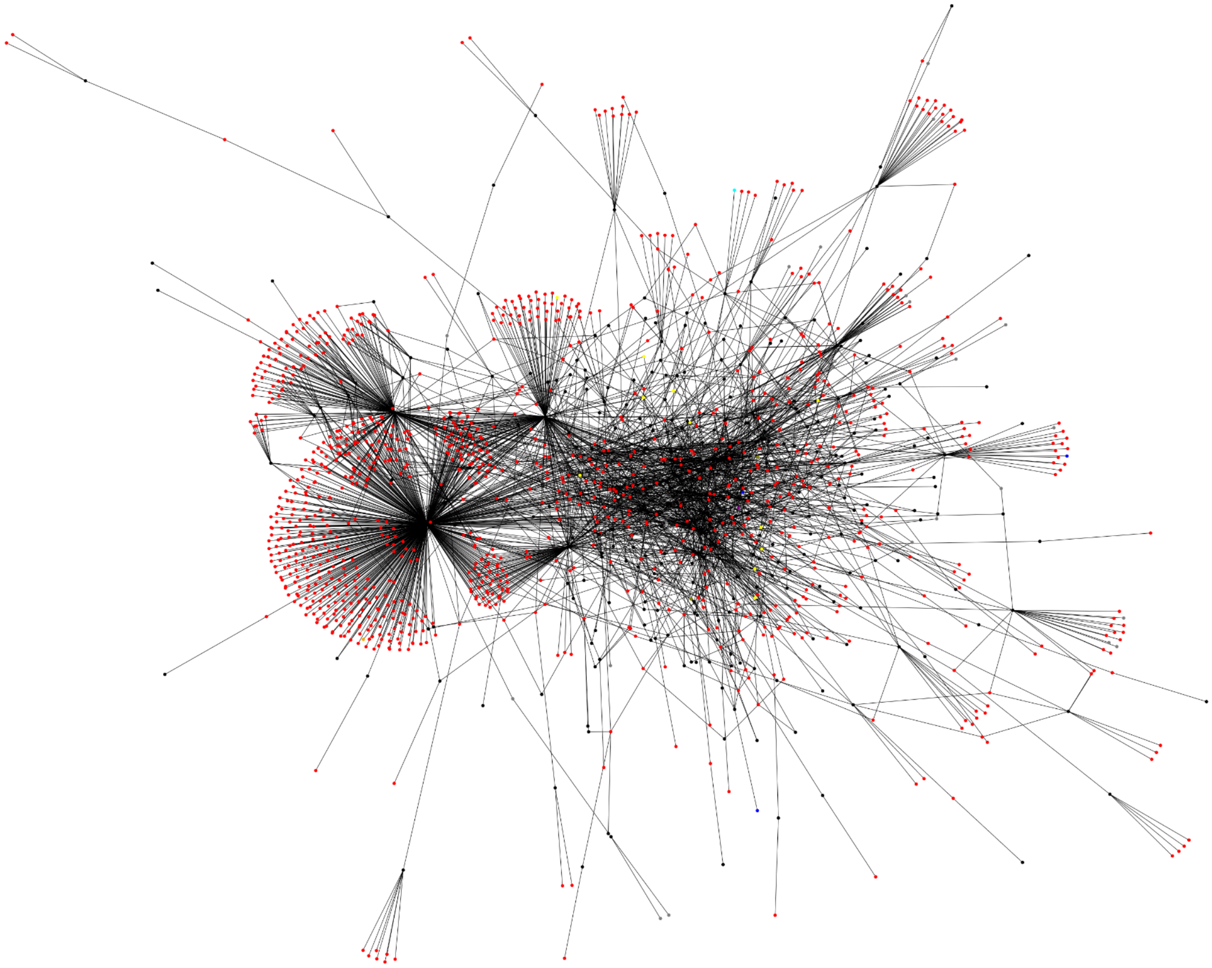
Republic of Latvia: Magenta

Figure 5: Knowledge Graph Depicting Indivudal's Associations with Organizations



Most connections happen in the largest, central cluster, so *Figure 6* shows a plot of that cluster on its own. On the left side of the figure, a few organization nodes stand out, with the most edges emanating from them, indicating popular organizations. We can clearly see four such organization nodes in the visualization. This observation mirrors what we can see in the bipartite graph in *Figure 9.*

Figure 6: Largest Component of the Knowledge Graph Depicting Indivudal's Associations with Organizations



It is also interesting to look at the clusters on the outskirts, since they represent isolated groups of people and could be sources for further research. For example, in *Figure 7,* There are ten people associated with the Moscow Cossack Choir, and these ten people are not connected to any other organization. Therefore, they are an isolated cluster. *Figure 8*, also shows an isolated cluster, but this cluster is of size three.

Figure 7: Component of Size 11 from Knowledge Graph Depicting Individual's Associations with Organizations

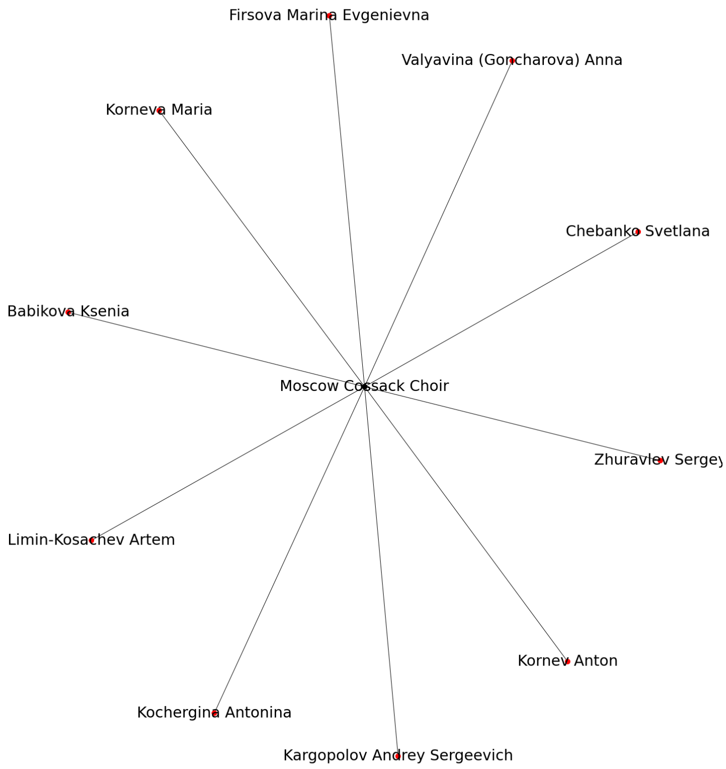
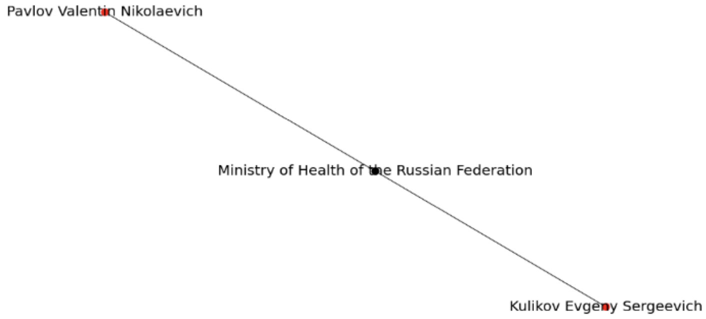
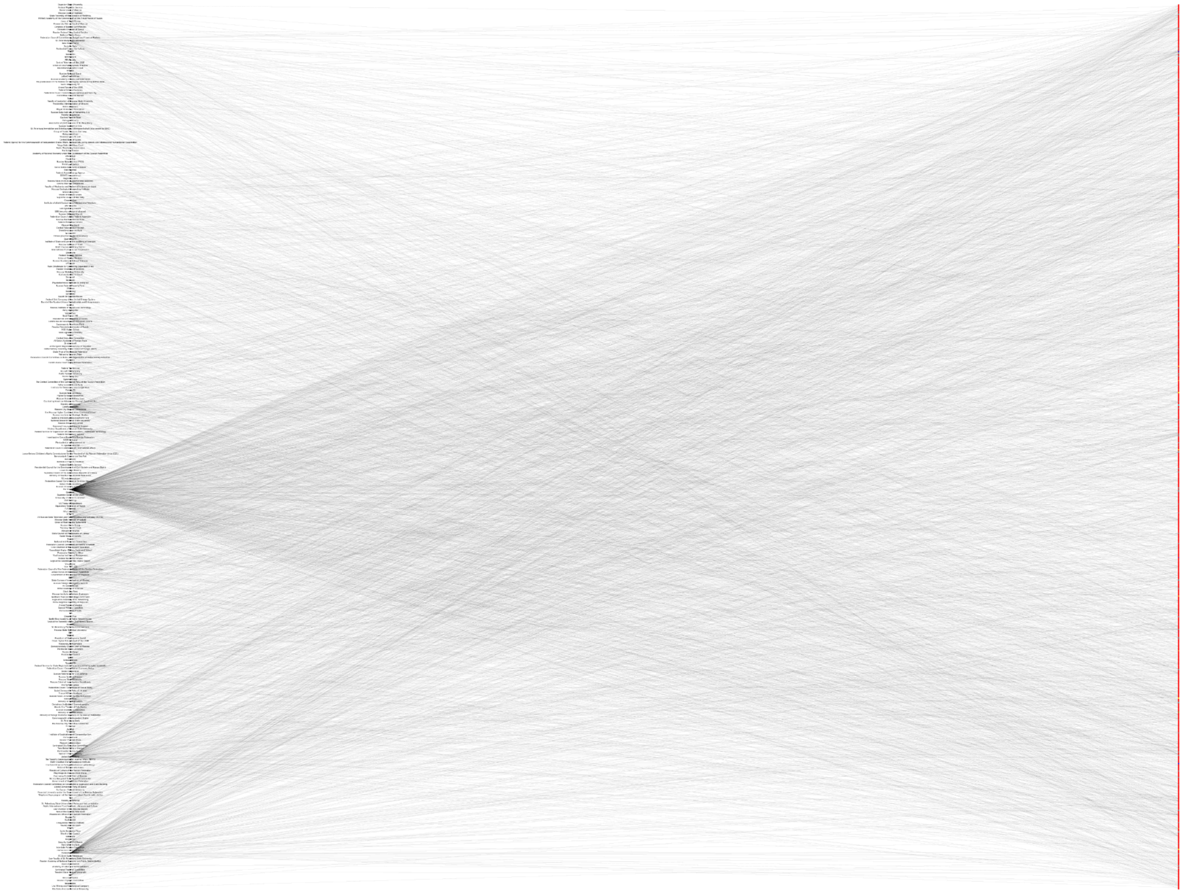


Figure 8: Component of Size 3 from Knowledge Graph Depicting Individual's Associations with Organizations


Pavlov Valentin Nikolaevich
Ministry of Health of the Russian Federation
Kulikov Evgeny Sergeevich

Overall, the knowledge graph offers a useful way to visualize general groups of people, but it becomes too crowded for effective labeling. A bipartite graph (*Figure 9*), however, proves helpful in highlighting the most prominent organizations and individuals, as it separates the nodes accordingly. While organization nodes are labeled (321 in total), people nodes are not (1,676 in total). The overwhelming red color on the right side indicates Russian citizenship for a majority of individuals. Nodes with significant shading nearby signify importance, as many lines stem from them. For instance, it's evident that four organizations are associated with the most people. Similarly, a few people nodes at the top are linked to numerous organizations. Given the graph's abundance of nodes, it's beneficial to consult a table breaking down each node's degree.[17] The degree of each person node indicates how many organizations they are a part of, while the degree of each organization node reflects how many people are associated with it. Combining the table with visualization clarifies nodes' influence and the relative magnitude thereof.

Figure 9: Bipartite Graph Depicting Individual's Associations with Organizations



[17] This is a spreadsheet showing the degree of each node in the knowledge graph: https://shorturl.at/0WeaL

After creating a network graph, each node acquires a degree value indicating its connections to other nodes. In this data set, a person node's degree signifies the number of organizations they are affiliated with, while an organization node's degree represents the number of individuals affiliated with it. After computing these degrees, we can identify the nodes with the most associations. Furthermore, various measures of connectedness can be calculated. For instance, closeness centrality measures a node's proximity to all other nodes in the network. A node’s closeness centrality is calculated as the reciprocal of the sum of the length of the shortest paths between the node and all other nodes in the graph. For example, a node with a degree of 50, isolated within a cluster, will exhibit lower closeness centrality compared to a similarly degreed node with connections extending to other network nodes. A snapshot depicting some nodes, their degrees, and their closeness centrality can be found in Table C.5.

Analyzing degree distributions also provides valuable insights into the connectivity patterns among network nodes. From the chart depicting the degree distribution of organizations (Figure B.1), it appears that the majority of organizations have few people associated with them, indicated by low degrees. Most organizations have a degree below 100, with an even larger majority having a degree of less than 10. Notably, only four organizations have a degree exceeding 100. These organizations warrant further research to understand why they produce a high number of individuals ending up on a "victim list".

Similarly, examining the degree distribution for people nodes reveals that slightly over a third of individuals have a degree of 1, meaning they are associated with just one organization. Around another 500 people have a degree between 2 and 5. A smaller proportion of individuals have a degree higher than 10, and these individuals merit closer examination. They represent the most connected individuals in the network, as they are associated with the highest number of people (Figure B.2).

Furthermore, plotting a log-log chart of the degree distribution allows for the identification of power law relationships and facilitates handling a wide range of values (Figure B.3).

The graphs visually depict clusters of people and organizations. They quickly highlight influential nodes and provide analyses, such as degree analysis, to determine the prominence of each node. The built out network graphs represent the most specific visualization possible, including every node. However, a more generalized and less crowded picture is achievable. For instance, we can reduce organization nodes from 321 to 25 by categorizing them based on organization types rather than specific organizations. Similarly, grouping people into categories such as age, citizenship, or tagged crime category can simplify the visualization. This approach allows for the creation of graphs illustrating the most common types of organizations or categories, targeting a subset of interest.

Following the analysis, a few main actors were identified, isolated groups were observed, and nodes with no connections to others were noted. The degree analysis revealed the top two most connected organizations, the State Duma and the Russian Federation Council, along with the two most connected individuals, Yevgeny Primakov and Dmitry Rogozin.[18] The graphs visually illustrate the extent of their influence. Additionally, organizations such as the Moscow Cossack Choir, Moscow State Technical University, Russian Academy of Arts, and Russian University of Transport were highlighted for being associated only with other unconnected individuals. With additional criminal and situational knowledge, these could become insightful areas for future investigation. Similarly, individuals like Leonard Blavatnik, Andrey Ivanovich Bocharov, and Alexander Vladimirovich Bochkarev were found to exist in their own bubble, which could also indicate potential areas for further exploration with more industry knowledge.

B. Regression Analysis

We want to quantify the effect that affiliation with an organization, or type of organization, has on being accused of a specific crime, or type of crime. In order to achieve this, we use regression analysis. In this case, the variables that signal connections and prominent individuals are organizations and the outcome of interest is their crime accusation (the reason they are on the list). Many different regressions can be built, including logits, multinomial logits, linear regressions, and multivariate linear regressions. Even in keeping the x variable related to organization and the y variable related to criminal activity, models can be changed to reflect different levels of specificity. For example the most specific multivariate model creates an x variable for each organization and a y variable for each type of criminal tag (Accomplice, Aggressor, Beneficiaries and Aggressors, etc). On the other hand, a simpler linear regression contains x variables for the 25 types of organizations and y variables for the 11 types of crimes. An even simpler, linear regression model, includes an x variable for each type of organization

[18] These make up the assembly of the Russian Government. The Federation council is the upper house and the State Duma is the lower house.

and regresses it on a continuous y variable which indicates the combined severity of the crimes a person is accused of.[19] Aside from using binary variables for the independent variables, variable controls, such as age and gender, can also be implemented into the model. Combining the knowledge graph analysis with the regression analysis, a possible independent variable is a node's degree whose coefficient quantifies the effect that an extra association has on being accused of a type of crime.

Coefficients can be analyzed based on their magnitude and significance, while models can be compared against each other based on clarity, usefulness, and predictive power with statistics such as F-statistics, R^2, adjusted R^2, and AIC and BIC. Large, statistically significant coefficients, signal independent variables with large effects on the types and severity of crimes people are accused of, as well as being areas for further investigation.

Most regressions can be found in a spreadsheet with detailed coefficient and model information.[20] However, three, new, simple, regressions effectively show how a quantitative analysis of this type can provide insights into the magnitude of an effect of an independent variable on a dependent variable. These regressions are included in the paper and their interpretations are as follows:

Table C.6 shows the results of an Ordinary Least Squares Regression using age and degree to predict overall crime severity. From the coefficients we can see that the x-variable degree is significant, and the x variable age is slightly less significant. From the coefficient of the x variable degree, we can conclude that if an individual is associated with one more organization, they will increase the overall severity of their crimes almost by a full point.

Table C.6 also shows the results of an Ordinary Least Squares Regression using the type of organization an individual is associated with to predict overall crime severity. Since many types of organizations have very few people associated with them, only organization types that were affiliated with around 3% of the population or more are shown. From the coefficients we can see

[19] Each type of crime (Annexation of Crimea, Crimes Related to Ukraine, Election Fraud, Illegitimate and anti-democratic constitutional coup, Political Repression, Propaganda, Money laundering, Illegal enrichment at the expense of Russian taxpayers, Corruption, Organized Crime, Fake news) was assigned a severity rating, and all of a person's criminal ratings were added together.

[20] A separate spreadsheet with the discussed regressions can be found here: https://shorturl.at/hlqmL. A spreadsheet with all of the regressions that we built can be found here: https://shorturl.at/iXSCf

that being associated with a media and TV organization has the highest positive influence on crime severity, followed by universities and then security agencies.

Table C.7 shows the results of a logit predicting whether or not an individual will be accused of a more severe crime (denoted by a 1) or a less severe crime (denoted by a 0). Similarly to the Ordinary Least Squares Regression, the left side shows a logit with x variables only for age and degree, while the right side shows a logit with added variables for organizations that are associated with at least 3% of the population. Universities, security agencies, and media and TV organizations seem to carry the most influence in increasing an individual's crime severity rating.

Because this dataset is relatively small, certain variables lack predictive power due to insufficient representation. With a larger dataset, the coefficients would become more significant. By supplementing this regression analysis with criminal expertise, we can better identify which variables are of interest and which regressions are important.

V. Discussion

A. Implications

The three main pieces of this project are converting unstructured data to structured data, visualizing a table of people and their biographical information, and using regression analysis to quantify the effect of important variables on an outcome of interest. The end goal of the work was to see if we could extract associations among people, visualize the affiliations, and then quantify the effect that individual characteristics had on predicting criminal outcomes.

Web scraping is useful when a website contains lots of important information but it is too tedious to go through and manually extract information. Python can read through the website on its own and compile a table of desired information. As information on the website changes, instead of going through and finding the changes manually, a script can be re-run over the website in a shorter period of time.

A list of people and their information does not lend itself easily to insights on its own. However, automatically extracting relationships can help any analytics team gain insight into connections that might not be obvious at the beginning. They can find connections between people and organizations, and test hypotheses about influence, power, and reach. After identifying clusters of people and understanding the layout of the individuals and organizations in the table, we can run a regression analysis to determine correlations between associations, biographical qualities, and the crimes individuals are accused of. One can imagine building a library of such tools for analysis to use.

Criminal justice is a field for which this analysis could be useful; the Putin's list website and the subsequent data frame are similar to a list of suspects and their biographical information. A suspect list can be turned into a knowledge graph, from which highly connected suspects will be illuminated with a cluster while other suspects will be shoved to the side, showing low interaction with other people on the list. Automatically extracting relationships can help a suspect list become more dynamic. Additionally, with proper criminal expertise, crimes of interest can be better identified and regression analysis can be more targeted. Criminal expertise can also narrow down the types of organizations that are important to look at, or that usually have an effect on being accused of a type of crime. Instead of regressing all types of organizations on all types of crimes, a criminal expert could suggest regressing five types of organizations on one type of crime, since it is known that these organizations were previously associated with certain criminals. Large, statistically significant, coefficients in regressions serve as points of further research for investigators.

The analytical tools used are very powerful. However, criminal experts can focus these tools to target specific categories and to obtain useful results faster.

B. Future Work and Limitations

The SpaCy package for Named Entity Recognition extracts only organizations, not the nature of an individual's connection to the organization, such as employment, university attendance, or political party support. In the future, it would be useful to use tools that could use clues around the entity to determine the type of association.

Due to the size of the dataset, many of the visualizations either did not include labels or had labels that were very hard to read. The full picture is important to see in order to grasp the general patterns. However, it is also important to graph sub-sections of the data to specify important areas of interaction.

These tools are useful, however, without field expertise they are just results without much insightful power. Therefore, supplementing this analysis with specific field intelligence is crucial to producing actionable outcomes.

VI. Conclusion

The goal of this work was to demonstrate the use of structured-data techniques for the analysis of unstructured data in order to identify hidden connections between individuals and groups. Using Putin's List, a website of profiles of individuals identified in connection with Putin's regime, as an example, we develop a collection of methods to find insights into associations between people on the list. While we used *Putin's List* as an example, analysts could use such techniques to find potential criminal activity, trace connections to known criminal elements, or to understand how best to disrupt such networks.[21]

[21] The workbook with all of the data frames, results, diagrams, etc, can be found here: https://shorturl.at/SjMnD

References

"Build a Knowledge Graph in NLP." GeeksforGeeks, GeeksforGeeks, 21 Mar. 2025, www.geeksforgeeks.org/build-a-knowlwdge-graph-in-nlp/.

"Drawing Basics." Memgraph's Guide for NetworkX Library, memgraph.github.io/networkx-guide/visualization/basics/. Accessed 19 May 2025.

GeeksforGeeks. (2023, November 17). *Python lambda functions*. GeeksforGeeks. https://www.geeksforgeeks.org/python-lambda-anonymous-functions-filter-map-reduce/

"Home." OARC Stats, stats.oarc.ucla.edu/other/mult-pkg/faq/general/faq-how-do-i-interpret-odds-ratios-in-logistic-regression/. Accessed 19 May 2025.

Hosseini, S. (2023, August 16). *Exploring the power of N-Grams: A comprehensive guide with examples in Python*. Medium. https://python.plainenglish.io/exploring-the-power-of-n-grams-a-comprehensive-guide-with-examples-in-python-dba6ca1fe6db

Jackson, Mathew O. Social and Economic Networks. Princeton University Press, 2008.

"Logistic Regression in Machine Learning." GeeksforGeeks, 11 Apr. 2025, www.geeksforgeeks.org/understanding-logistic-regression/.

"Multiple Logistic Regression." StatsTest.Com, 18 May 2020, www.statstest.com/multiple-logistic-regression/.

Natasha Noy, Y. G. (2019, August 1). *Industry-scale knowledge graphs: Lessons and challenges*. ACM. https://cacm.acm.org/magazines/2019/8/238342-industry-scale-knowledge-graphs/fulltext

"NetworkX : Python Software Package for Study of Complex Networks." GeeksforGeeks, GeeksforGeeks, 11 July 2022, www.geeksforgeeks.org/networkx-python-software-package-study-complex-networks/.

NewsCatcher. (n.d.). *Named entity recognition(ner) with spacy [with code example]*. https://www.newscatcherapi.com/blog/named-entity-recognition-with-spacy

Otten, N. V. (2023, October 31). *N-grams made simple & how to implement in Python (NLTK)*. Spot Intelligence. https://spotintelligence.com/2023/04/05/n-grams/

Real Python. (2023, January 2). *Natural language processing with spacy in python*. https://realpython.com/natural-language-processing-spacy-python/

Standing Committee of the Free Russia Forum. (n.d.). Personas. Personas – The database "PUTIN'S LIST." https://www.spisok-putina.org/en/personas/

Appendix

A. Snippets

Snippet A.1: Decoding URL and converting to text object

```
for i in tqdm(range(2,142)):
    url = "https://www.spisok-putina.org/en/personas/page/" + str(i) + "/"
    page = urlopen(url)
    html = page.read().decode("utf-8")
    soup = BeautifulSoup(html,"html.parser")
    html_text = soup.prettify()
```

Snippet A.2: Defining custom url_get_info function

```
def url_get_info(test):
    page = urlopen(test)
    html = page.read().decode("utf-8")
    soup = BeautifulSoup(html,"html.parser")

    temp_info = soup.find_all('div', class_='infobox')

    columns = ['Full_Name', 'Date_of_Birth', 'Citizenship', 'Categories',
'Professional_Biography', 'Accused_of']
    df = pd.data frame(columns=columns)

    full_name = soup.find('strong', text ='Full name:').find_next_sibling(text=True).strip()

    dob = soup.find('strong', text='Date of Birth:')
    if dob == None:
        dob_fill = 'NULL'
    else:
        dob_fill = dob.find_next_sibling(text=True).strip()

    citizenship = soup.find('strong', text='Citizenship:')
    if citizenship == None:
        citizenship_fill = 'NULL'
    else:
        citizenship_fill = citizenship.find_next_sibling(text=True).strip()

    category = soup.find('strong', text='Categories:').find_all_next('a')
    categories = [tag.text.strip() for tag in category]
    matched_categories= []
    for cat in categories :
```

```
    if cat in ['Power-holders', 'Executors', 'Law-destroyers', 'Aggressors', 'Beneficiaries',
'Oligarchs and corrupt officials', 'Propagandists', 'Accomplices']:
      matched_categories.append(cat)
      print(matched_categories)

  professionalbio = soup.find('strong', text ='Professional field/official position/biography:')
  if professionalbio == None:
    professionalbio_fill_clean = 'NULL'
  else:
    prof_parent = professionalbio.parent
    professionalbio_fill = prof_parent.find_all('p')
    professionalbio_fill_clean = [tag.get_text(separator=' ', strip=True) for tag in
professionalbio_fill]

  what_accused = soup.find('strong', text ='Accused of:')
  if what_accused == None:
    what_accused_fill_clean = 'NULL'
  else:
    accused_parent = what_accused.parent
    what_accused_fill = accused_parent.find_all('p')
    what_accused_fill_clean = [tag.get_text(separator=' ', strip=True) for tag in
what_accused_fill]

  df = pd.concat([df, pd.data frame({'Full_Name': [full_name], 'Date_of_Birth': [dob_fill],
'Citizenship': [citizenship_fill], 'Categories': [matched_categories],
'Professional_Biography':[professionalbio_fill_clean],
'Accused_of':[what_accused_fill_clean]})], ignore_index=True)

  return(df)
```

Snippet A.3: Creating a list of entities

```
nlp = spacy.load("en_core_web_sm")
pro_bio_text = ''.join(putins_list['Professional_Biography'].astype(str).tolist()) #convert col
into string
pro_bio_text_prosc = nlp(pro_bio_text)   #convert to Doc object for processing
ner_list_specif_desc = [] #create empty list for storing all of the entities
for ent in tqdm(pro_bio_text_prosc.ents):
  if ent.label_ in ['ORG', 'PRODUCT', 'EVENT', 'FAC', 'LAW', 'NORP']:
    ner_list_specif_desc.append(
      f"""
     {ent.text = })
```

```
        spacy.explain('{ent.label_}') = {spacy.explain(ent.label_)}"")
```

Snippet A.4: Creating and filling a binary variable for each organization

```
for index, row in putins_list.iterrows():
  for index2, row2 in match_var.iterrows():
      phrase = row2['appears_as']
      var = row2['binary_var_match']
      if re.search("\W" + re.escape(phrase) + "\W", str(row['Professional_Biography']),
re.IGNORECASE):
         cell_value = putins_list.at[index, var]
         if cell_value == 0:
            putins_list.at[index, var] = 1
```

Snippet A.5: Filling crime type binary variables

```
p_list['constitutional_coup'] = p_list['Accused_of'].str.lower().apply(lambda x: 1 if isinstance(x,
str) and 'illegitimate and anti-democratic constitutional coup' in x else 0)
```

Snippet A.6: Creating a continuous y variable indicating total crime severity

```
for i in crime_severity["word"]:
   # Check if the corresponding binary variable appears with a 1 in p_list
   if i in p_list.columns:
      mask = p_list[i] == 1 #saves column as variable if 1 (mask is a column)
      if mask.any():  # Only proceed if there's at least one occurrence of 1
         # Add the corresponding severity value from crime_severity to continuous y column in
p_list
         p_list['continuous_y'] += mask * crime_severity.loc[crime_severity["word"] == i,
"rating"].iloc[0]
```

Snippet A.7: Creating and plotting a knowledge graph

```
G = nx.Graph() #initialize a graph
for index,row in p_list.iterrows():
   G.add_node(row['Full_Name'], color=citizenship_colors[row['citizenship1']] if
row['citizenship1'] in citizenship_colors else 'gray') #add person nodes and color based on
citizenship
G.add_nodes_from(p_list.columns[15:337].tolist(), color='black') #add black nodes for
organizations
# Add edges based on associations (1-in column means associated, so draw the edge)
for index, row in p_list.iterrows():
   person = row['Full_Name']
```

```
    for entity in p_list.columns[15:]:
        if row[entity] == 1:
            G.add_edge(person, entity)
#plot the knowledge graph with Fruchterman Reingold layout (Different layouts for network graphs described here: https://memgraph.github.io/networkx-guide/visualization/basics/)
pos = nx.fruchterman_reingold_layout(G)
nx.draw(G, pos, with_labels=False, node_color=colors, node_size=17, font_size=8)
```

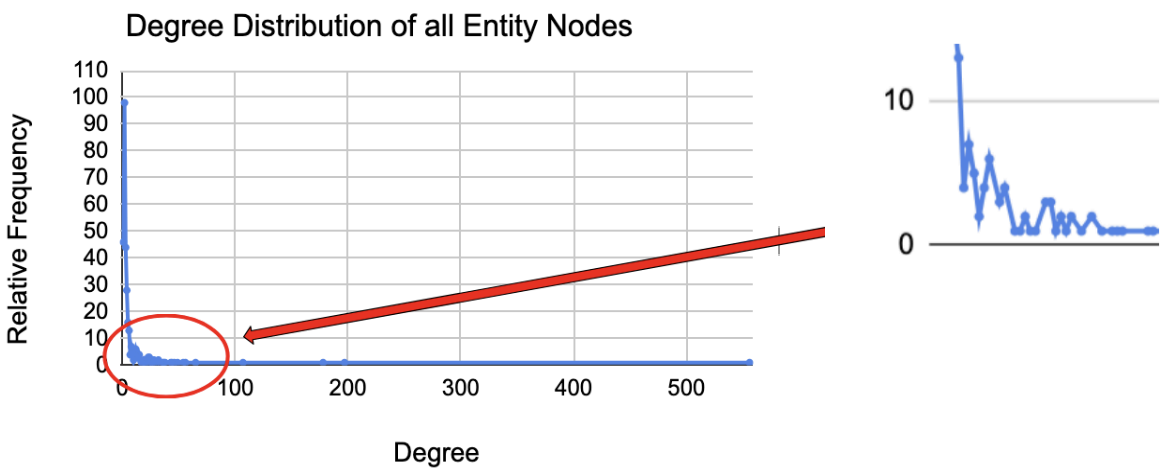
Snippet A.8: Running an Ordinary Least Squares Regression

Example where the x variables are columns 336-361 and the y variable is the variable continuous_y:

```
X = p_list.iloc[:, 337:362]
Y = p_list["continuous_y"]
X = sm.add_constant(X)
model2 = sm.OLS(Y, X).fit()
```

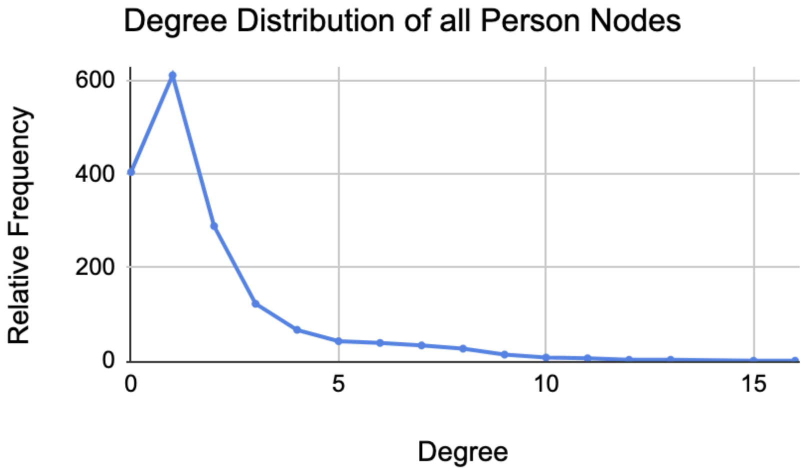
B. Figures

Figure B.1: Degree distribution of organization nodes



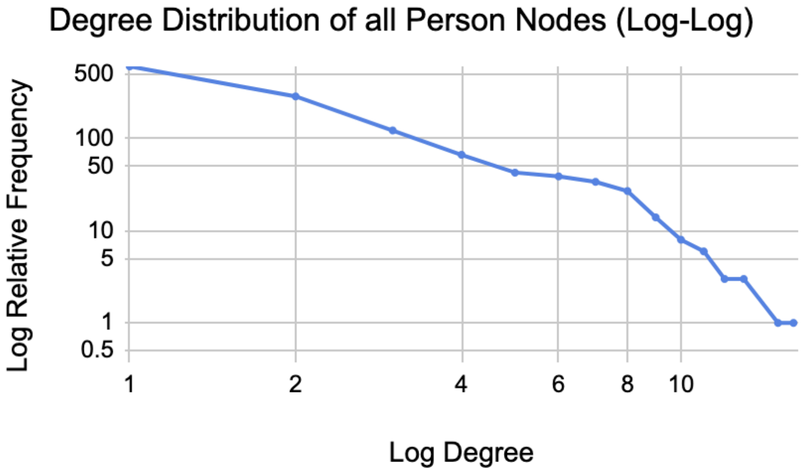
Note: This chart depicts the distribution of degrees among the organizations. Degree measures how many people are associated with an organization. A few organizations have hundreds of people associated with them, however, most have around 70 or less, therefore a zoomed in portion of the graph is shown.

Figure B.2: Degree distribution of person nodes



Note: This chart depicts the distribution of degrees among people. Degree measures how many organizations a person is associated with. Most people have a degree below 5, however, a few have degrees higher than 5, and even fewer have degrees higher than 10.

Figure B.3: Log-log degree distribution of person nodes



Note: The log-log scale helps in visualizing this wide range of values and in identifying the power-law distribution characteristic of such networks. The graph illustrates that as the degree increases, the relative frequency decreases. This pattern is typical of many real-world networks, where a few nodes have a high degree while the majority have a low degree.

C. Tables

Table C.1: Summary Statistics

Variable	Minimum	Maximum	Mean	Standard Deviation
Crime severity	0	27	8.921	5.119
Degree	0	556	3.130	14.666
Age	1	103	57.05	11.916
Government Agency	0	1	0.471	0.499
Media and TV	0	1	0.129	0.336
Political Party	0	1	0.167	0.373
University	0	1	0.125	0.330
Security Agency	0	1	0.045	0.208

Table C.2: Breakdown of the amount of people per organization type

Organization Type	Number of People Affiliated
Award	12
Bank	44
Company	31
Company (Oil)	31
District Court	27
Federation Council	28
Gang	2
Government Agency	223
Lobby/Union	19
Media/TV	101
Military Academy	11
Military Agency	40
Military District	5

Outside Gov	12
Outside Org	0
Political Party	80
Region	18
Religious	11
Sanctions	21
Security Agency	53
Sports	13
Theatre	1
University	135
War On Ukraine	1
Other	3

Table C.3: Breakdown of the amount of people accused of each crime of interest, and the severity rating associated with the crime

Crime	Severity Rating	Number of People Accused
Annexation of Crimea	11	74
Crimes Related to Ukraine	10	9
Election Fraud	9	5
Illegitimate and anti-democratic constitutional coup	8	11
Political Repression	7	85
Propaganda	6	130
Money laundering	5	28
Illegal enrichment at the expense of Russian taxpayers	4	2
Corruption	3	141
Organized Crime	2	32
Fake news	1	13

Table C.4: Breakdown of the amount of people accused of each category

Category Combination	Number of People Tagged with that Category
Executors	744
Accomplices	435
Propagandists	201
Law-destroyers	80
Oligarchs and corrupt officials	63
Beneficiaries	43
Aggressors	26
Executors, Law-destroyers	24
Executors, Oligarchs and corrupt officials	15
Aggressors, Executors	9
Power-holders	8
Executors, Propagandists	7
Beneficiaries, Propagandists	3
Accomplices, Executors	2
Accomplices, Propagandists	2
Aggressors, Power-holders	2
Aggressors, Propagandists	2
Accomplices, Aggressors	1
Accomplices, Beneficiaries	1
Accomplices, Oligarchs and corrupt officials	1
Aggressors, Beneficiaries	1
Aggressors, Oligarchs and corrupt officials	1
Beneficiaries, Executors, Oligarchs and corrupt officials	1
Beneficiaries, Oligarchs and corrupt officials	1
Executors, Power-holders	1
Law-destroyers, Propagandists	1
Oligarchs and corrupt officials, Power-holders	1

Table C.5: Examples of organization and people nodes and their degrees[22]

Panel A: Organization Nodes

Node	Node Type	Degree	Closeness Centrality
The State Duma	Government Agency	556	0.3291
Federation Council	Federation Council	197	0.2519
United Russia Party	Political Party	178	0.2601
Communist Party	Political Party	107	0.2527
Kremlin	Government Agency	65	0.2488
All-Russian State Television and Radio Broadcasting Company (VGTRK)	Media and TV	56	0.2440

Panel B: Person Nodes

Node	Node Type	Degree	Closeness Centrality
Primakov Yevgeny	Person	17	0.2600
Rogozin Dmitry	Person	16	0.2609
Kovalchuk Yuri	Person	15	0.2112
Chemezov Sergey	Person	13	0.2256
Fradkov Mikhail	Person	13	0.2004

Note: Closeness centrality measures how close a node is to all other nodes in the network. It is calculated as the reciprocal of the sum of the length of the shortest paths between the node and all other nodes in the graph. For organization nodes, degree indicates how many people are affiliated with that specific organization. For people nodes, degree indicates how many organizations an individual is associated with.

[22] For full spreadsheet of degrees, click here: https://shorturl.at/0WeaL

Table C.6: Ordinary Least Squares Regression

	Combined Crime Severity Rating	Combined Crime Severity Rating
Degree	0.923*** (0.04)	
Government Agency		1.065*** (0.185)
Media and TV		3.636*** (0.283)
Political Party		0.606** (0.237)
University		3.581*** (0.280)
Security Agency		3.345*** (0.455)
Age	-0.012* (0.007)	0.003 (0.007)
R^2	0.327	0.291
Number of Observations	1,447	1,447

Note: Both OLS regressions aim to quantify the effect that degree (how many organizations a person is associated with), age, and specific organization have on the overall severity of crimes an individual is accused of. The organizations selected for the OLS, are organizations that at least 3% of the dataset is associated with. Significance is indicated as follows: * $p < 0.10$, ** $p < 0.05$, *** $p < 0.01$

Table C.7: Logit Regression

	More Severe	More Severe
Degree	0.441*** (0.037)	0.242*** (0.059)
Government Agency		0.331 (0.294)
Media and TV		0.998*** (0.305)
Political Party		0.134 (0.292)
University		1.343*** (0.270)
Security Agency		0.790** (0.397)
Age	0.001 (0.009)	0.003 (0.010)
R^2	0.222	0.271
Number of Observations	1,447	1,447

Note: This logit models aims to classify whether a person will be accused of a more severe crime or not, based on their age, degree (the number of organizations an individual is associated with), and type of organization they are affiliated with. If a person's overall crime severity is less than 8, they are classified as less severe (0), if their crime severity is greater than or equal to 8, they are classified as more severe (1). Significance is indicated as follows: * p < 0.10, ** p < 0.05, *** p < 0.01