Project Write-Up: Graph Analysis and Visualization

Table of Contents

1. Overview
2. Dataset Description
3. Methodology
    - Data Loading and Processing
    - Market Cap Distribution
    - Global Representation
    - Outlier Detection
    - Graph Construction
    - Degree Centrality
    - Visualization: Centrality Without Labels
    - Visualization: Centrality with US Labels
    - Tests
4. Outcomes and Insights
    - Market Cap Distribution
    - Global Representation
    - Outliers
    - Graph Centrality
5. Conclusion

Overview

This project leverages graph theory and visualization techniques to analyze a dataset of companies. The goal is to examine centrality measures, market cap distributions, global representation, and outliers. Two distinct visualizations—centrality with and without labels—highlight insights, particularly focusing on United States companies.

The code is implemented in Rust, utilizing the plotters library for visualizations and a modular design for clarity and scalability. Below, we delve into the methods used, their implementation, and the outcomes observed.

Dataset Description

The dataset contains:

- Market Capitalization: Represents the size of each company.
- Stock Prices: Numerical values indicating the trading price of each company's stock.
- Countries: Locations where the companies are based.

The data is processed from a CSV file, and relationships between companies are built based on shared attributes, such as countries.

Methodology

Data Loading and Processing

The project begins by reading the dataset using csv::Reader. Three vectors are initialized:

- market cap: Stores the market capitalization of each company.
- prices: Stores stock prices.
- countries: Stores the country of each company.

```
let mut marketcap = Vec::new();
let mut prices = Vec::new();
let mut countries = Vec::new();

for record in reader.records() {
    let record = record?;
    if let (Ok(mc), Ok(price), Ok(country)) = (
        record[3].parse::<f64>(),
```

```
        record[4].parse::<f64>(),
        record[5].parse::<String>(),
    ) {
        marketcap.push(mc);
        prices.push(price);
        countries.push(country);
    }
}
```

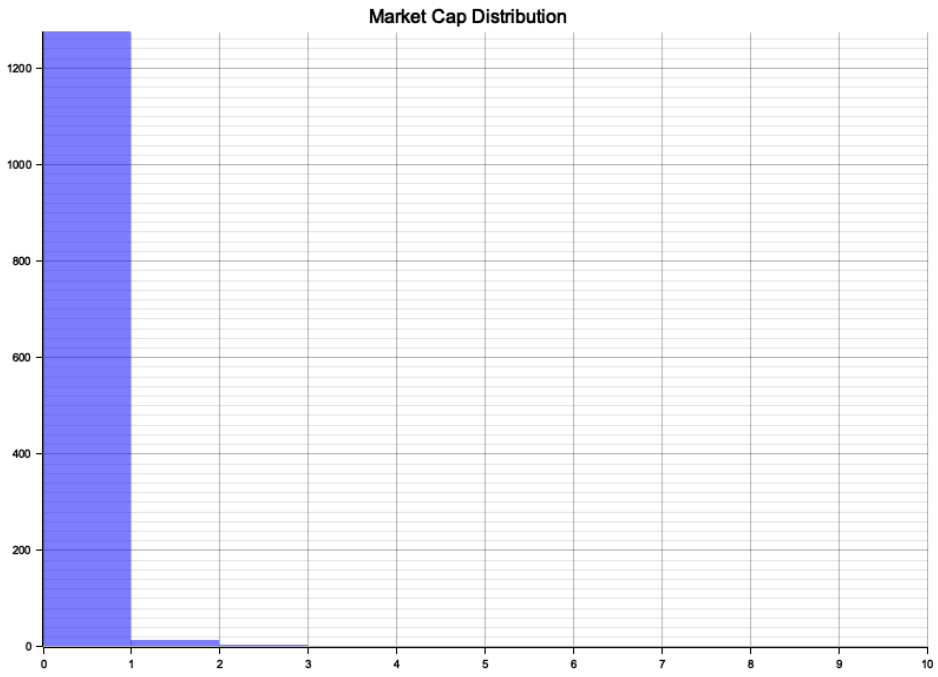
Market Cap Distribution

The dataset's market cap is divided into bins to observe its distribution.

- A histogram is generated using plotters, showing the frequency of market cap values.
- The distribution reveals a strong clustering of companies with smaller market caps.
- Most companies (1274) fall into **Bin 0**, representing the smallest market cap values.
- A sharp drop is seen in larger bins, with only 14 companies in **Bin 1** and fewer than 10 companies across all other bins.

This shows that the majority of companies in the dataset are small to medium-sized firms, while only a few large corporations dominate the higher market cap bins.

Visualization: Market Cap Distribution



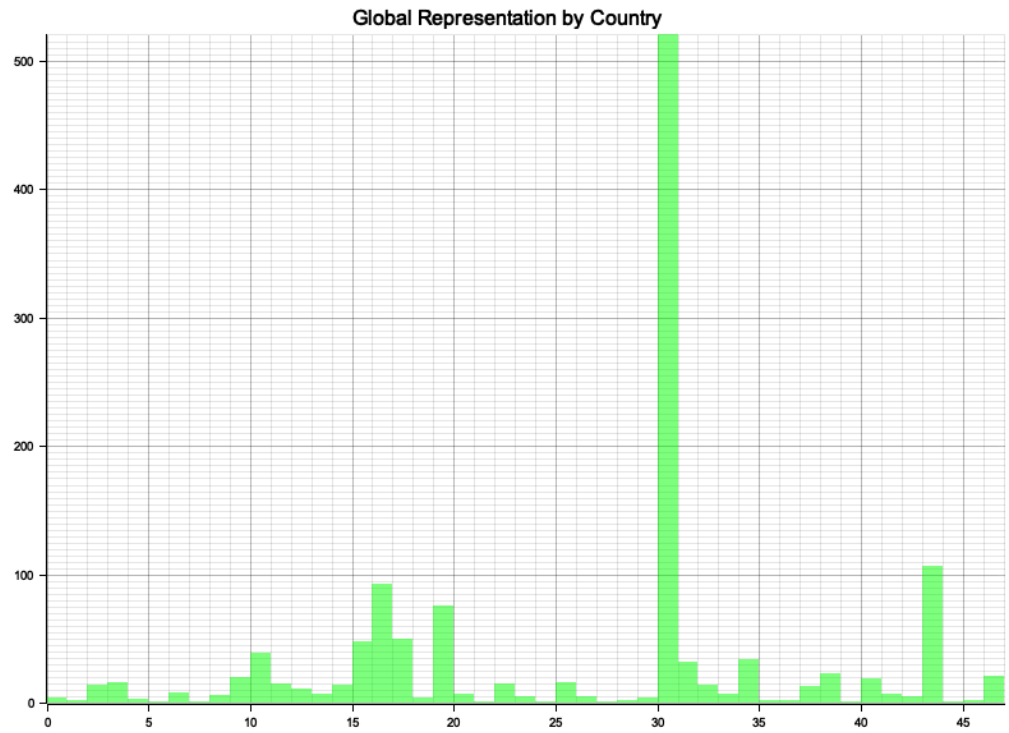
Global Representation

Using a HashMap, the number of companies per country is computed. A bar chart is generated to visualize the global representation, highlighting the dominance of certain countries (e.g., the United States).

```
let mut country_map = HashMap::new();
for country in &countries {
    country_map.entry(country.clone()).or_default() += 1;
}
```

A bar chart reveals the global representation of companies by country. The United States dominates the dataset with 520 companies, followed by China (107), Japan (93), and India (76). These figures underscore the economic influence of these nations in the global market.

Visualization: Global Representation by Country



Smaller countries such as **Argentina** and **Czech Republic** have only one company represented in the dataset, highlighting disparities in market representation.

Outlier Detection

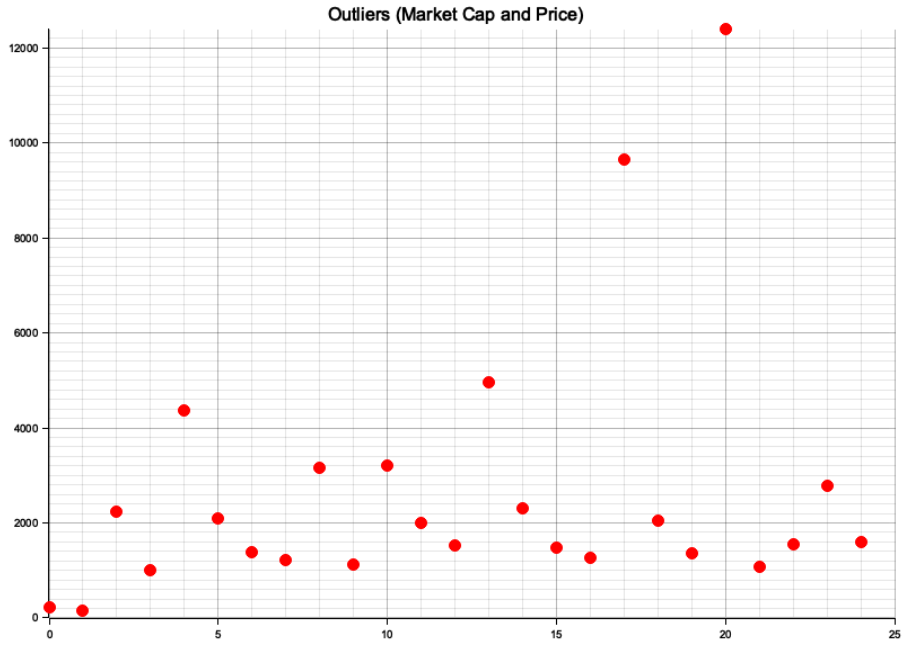
Outliers are identified based on high market caps or stock prices. The detection criteria:

- Market cap exceeding the 90th percentile.
- Stock price exceeding a threshold (e.g., $1000).

Outliers are visualized as red points, representing anomalies in the dataset.

```
for (i, &cap) in marketcap.iter().enumerate() {
    if cap > cap_threshold || prices[i] > 1000.0 {
        outliers.push((cap, prices[i], countries[i].clone()));
    }
}
```

Visualization: Outliers (Market Cap and Price)



Outliers, based on exceptionally high market caps or stock prices, highlight industry giants and significant anomalies in the dataset. Notable examples include:

1. **Market Cap of $3.59 trillion (US)** with a price of $236.48.
2. **Market Cap of $234 billion (France)** with a price of $2235.20.
3. **Market Cap of $146 billion (US)** with a price of $4363.72.

These outliers likely represent companies with significant global influence, such as major tech firms or energy giants.

Graph Construction

A graph is constructed using the Graph struct. Nodes represent companies, and edges are created between nodes if companies share the same country.

```
let mut graph = Graph::new();
for (i, country) in countries.iter().enumerate() {
    for (j, other_country) in countries.iter().enumerate() {
        if i != j && country == other_country {
            graph.add_edge(i, j);
        }
    }
}
```

Degree Centrality

The centrality of each node is calculated by counting its neighbors. This provides insights into the influence or connectedness of nodes within the graph.

```
pub fn compute_degrees(&self) -> HashMap<usize, usize> {
    let mut degrees = HashMap::new();
    for (node, neighbors) in &self.adjacency_list {
        degrees.insert(node, neighbors.len());
```

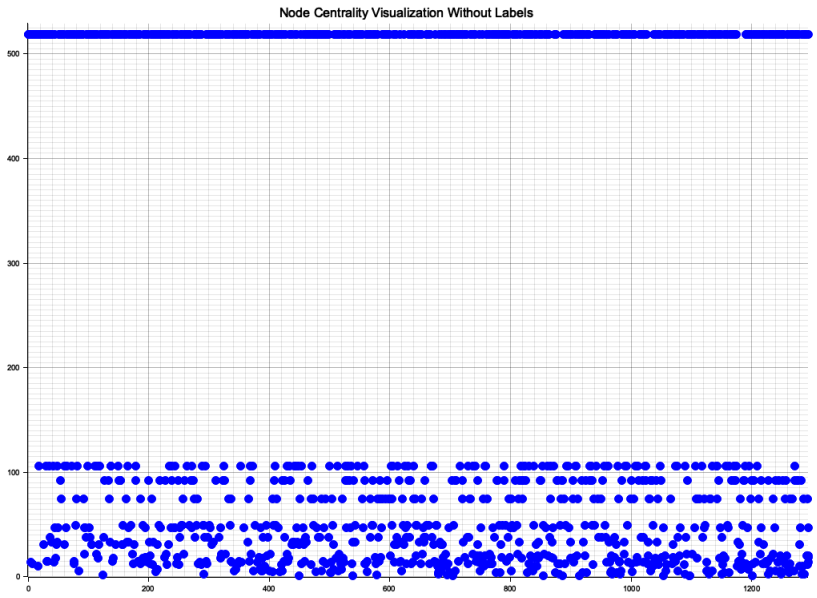
```
    }
    degrees
}
```

The degree centrality analysis reveals that **U.S. companies dominate the network structure**, with numerous nodes having a centrality score of 519. This highlights the interconnectedness and dominance of U.S.-based firms in the dataset.

U.S.-Focused Centrality Highlights:

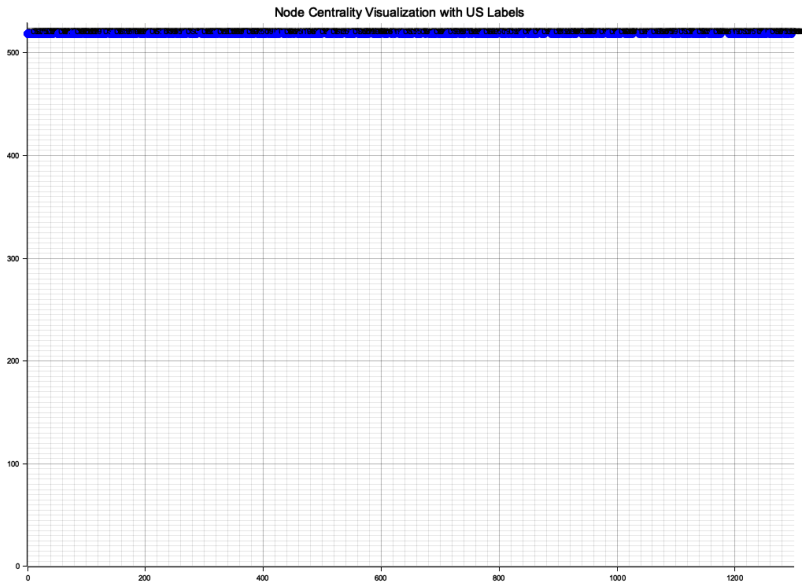
- Nodes such as **1095, 546, 915, and 24** are among those with the highest centrality scores of 519.
- This centrality indicates a strong network influence and high interconnections based on shared attributes.

Visualization: Centrality Without Labels



This visualization plots the centrality scores of all nodes without distinguishing labels. It highlights the overall structure of the graph, showing many low-centrality nodes and a few high-centrality outliers.

Visualization: Centrality with US Labels



This visualization focuses on nodes representing U.S. companies. Labels are added for nodes with high centrality (e.g., greater than 30).

```
for (&node, &score) in centrality {
    if let Some(label) = labels.get(&node) {
        if label == "United States" {
            chart.draw_series(std::iter::once(Circle::new((node, score), 5, BLUE.filled())))?;
            chart.draw_series(std::iter::once(
                Text::new(
                    format!("US: {}", score),
                    (node, score + 5),
                    ("sans-serif", 12).into_font(),
                ),
            ))?;
        }
    }
}
```

Tests

Testing is an essential part of the project to ensure the reliability and correctness of the code implementation. For this project, I wrote unit tests to verify the functionality of critical components such as histogram calculation, country representation analysis, and outlier detection. Below is an overview of the tests conducted, their objectives, and the results.

1. Histogram Calculation Test

- Objective: Verify that the histogram function accurately distributes data into bins and accounts for all data points.
- Description: A dataset was divided into 4 bins, and the function was tested to ensure:
    - The total count of values in all bins equals the size of the input dataset.
    - The counts in each bin match the expected distribution based on predefined bin ranges.
- Dataset: data=[0.5,1.5,2.0,2.8,3.3,3.7,4.1,4.9,5.0,5.5]\text{data} = [0.5, 1.5, 2.0, 2.8, 3.3, 3.7, 4.1, 4.9, 5.0, 5.5]data=[0.5,1.5,2.0,2.8,3.3,3.7,4.1,4.9,5.0,5.5]
    - Expected bin counts: [2, 2, 3, 3]
- Result: The test passed successfully, confirming the histogram calculation function works as intended.

2. Country Representation Test

- Objective: Ensure that the country representation function correctly counts the number of companies for each country.
- Description: The function processes a list of countries and outputs the number of companies from each country.
- Dataset: countries=["USA","USA","Canada","India","Canada"]\text{countries} = ["USA", "USA", "Canada", "India", "Canada"]countries=["USA","USA","Canada","India","Canada"]
    - Expected results:
        - {"USA": 2, "Canada": 2, "India": 1}
- Result: The test passed, confirming that the country representation analysis is accurate.

3. Outlier Detection Test

- Objective: Test if the outlier detection function correctly identifies values above a given threshold.

- Description: The function analyzes a dataset and detects values exceeding a specified threshold.
- Dataset: data=[10.0,20.0,30.0,40.0,50.0],threshold=35.0\text{data} = [10.0, 20.0, 30.0, 40.0, 50.0], \text{threshold} = 35.0data=[10.0,20.0,30.0,40.0,50.0],threshold=35.0
    - Expected outliers: [40.0, 50.0]
- Result: The test passed, confirming the function accurately detects outliers in the data.

Outcomes and Insights

Market Cap Distribution

Insight: The majority of companies have smaller market caps, with a steep decline in frequency for larger caps. This reflects the reality that most firms are small to medium-sized, with a few major players dominating.

Global Representation

Insight: The bar chart highlights the dominance of companies based in the United States, which far surpasses representation from other countries. This emphasizes the economic influence of the U.S.

Outliers

Insight: The outlier chart pinpoints companies with extremely high market caps or stock prices. These outliers likely correspond to major corporations with global impact, such as tech giants or industry leaders.

Graph Centrality

Insight: Centrality analysis reveals that most nodes (companies) have low connectivity, representing limited shared attributes. However, U.S. companies stand out with higher centrality, indicating strong interconnectedness and influence within the dataset.
US-Focused Centrality: Nodes labeled as "United States" with significant centrality demonstrate the central role of the U.S.-based companies in the datasets network structure.

Conclusion

This project demonstrates how graph theory and data visualization can uncover patterns and anomalies in a dataset. By combining centrality measures with focused visualizations, the

analysis highlights the structural relationships and dominance of specific entities within the dataset. The modular codebase ensures extensibility, allowing further exploration of other graph metrics or datasets.