Analysis of GitHub Repositories and User Profiles

About the dataset

The data comprises two tables: one detailing GitHub repositories and the other focusing on user profiles. The repositories table includes 52,488 entries with columns such as login, full_name, created_at, stargazers_count, watchers_count, language, has_projects, has_wiki, and license_name. The user profiles table contains 518 entries with columns like login, name, company, location, email, hireable, bio, public_repos, followers, following, and created_at.

From the sample data of the repositories table, we observe that the login field is consistent across the first five entries, indicating a single user, 'munificent', with various projects. These projects have varying levels of popularity, as seen in the stargazers_count and watchers_count, with the project 'munificent/bantam' being the most popular. The projects are primarily coded in languages such as Dart, C#, Java, and HTML, and all have active project and wiki features.

In the users table, the sample data highlights prominent users like 'bradfitz' and 'tenderlove', who have significant followings and numerous public repositories. The data also reveals that these users are associated with well-known companies like TAILSCALE and SHOPIFY, and are primarily located in Seattle. The bio field provides insights into their professional backgrounds and contributions to the tech community. Overall, the data offers a comprehensive view of GitHub's ecosystem, showcasing the diversity of projects and the influence of key contributors.

Relevant Inquiries

What is the trend in the number of repositories created over time in the repositories.csv?

Data Analysis

- **Time Frame**: The data spans from 2008 to 2024.
- **Monthly Repository Count**: The average number of repositories created per month is approximately 261, with a standard deviation of 146.79. The minimum and maximum monthly counts are 2 and 587, respectively.

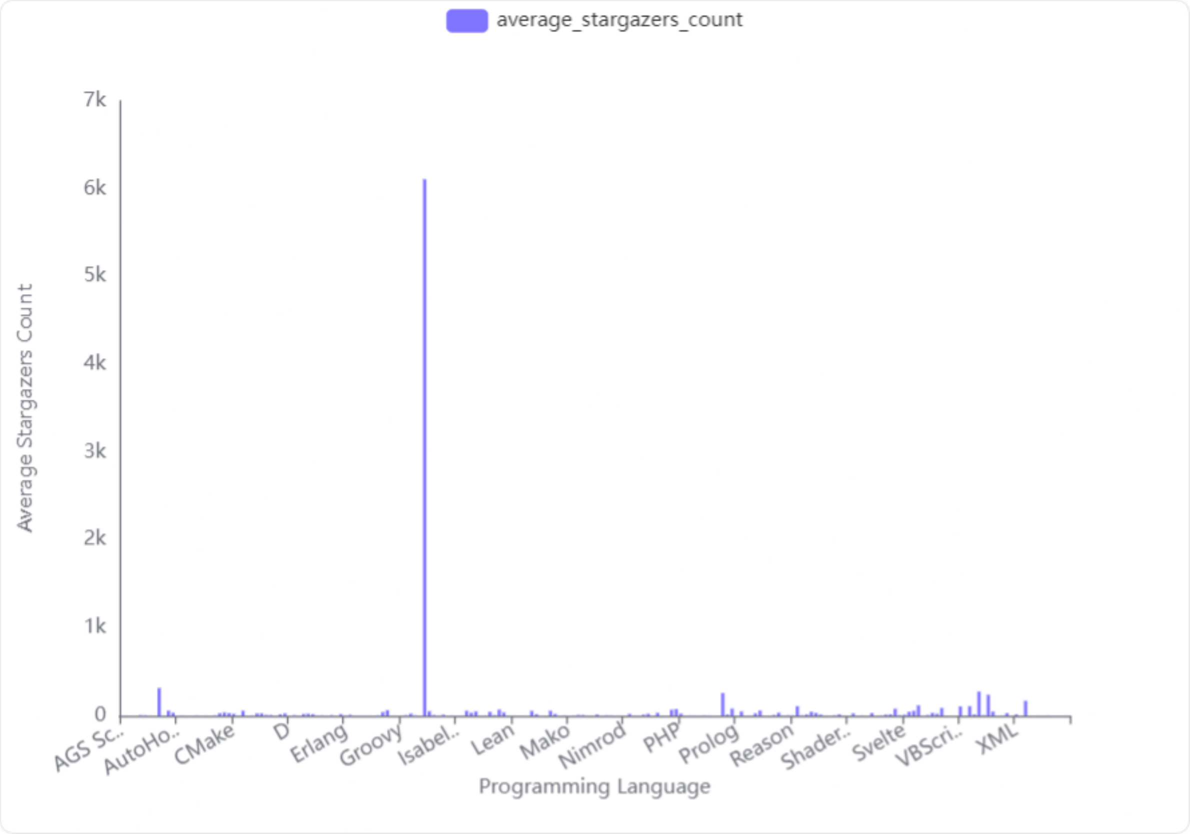
Visualization Insights

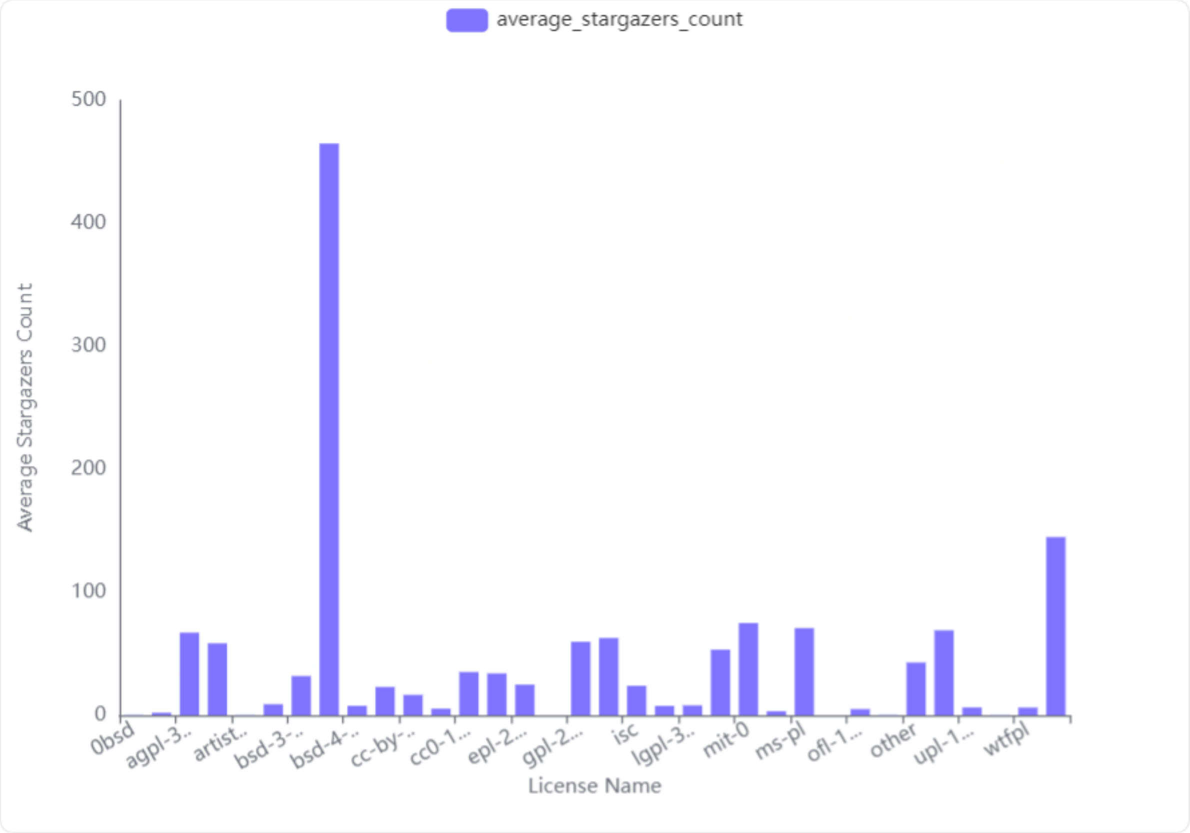
- **Initial Growth**: There is a sharp increase in the number of repositories created from 2008, peaking around 2009.
- **Decline and Stability**: After the peak, the number of repositories decreases and stabilizes over the following years.

Conclusion and Insights

- **Early Surge**: The initial years (2008-2009) saw a rapid increase in repository creation, indicating a period of high activity or adoption.
- **Stabilization**: Post-2009, the trend shows a decline followed by stabilization, suggesting a maturation phase in repository creation activity.

Analyze the correlation between stargazers_count and other variables such as language and license_name in the repositories.csv to identify any patterns or trends.





Correlation with Language

Average Stargazers Count: The mean average stargazers count across languages is 50.46, with a high standard deviation of 428.53, indicating significant variability.

Sample Data: Languages like 'AGS Script' and 'AMPL' have low average stargazers counts (1.5 and 1.0, respectively), while 'ActionScript' has a slightly higher average of 6.97.

Visual Insights: The bar chart shows a wide range of average stargazers counts across different programming languages, with some languages having significantly higher counts, indicating popularity or niche interest.

Correlation with License Name

Average Stargazers Count: The mean average stargazers count across licenses is 41.77, with a standard deviation of 81.51, suggesting moderate variability.

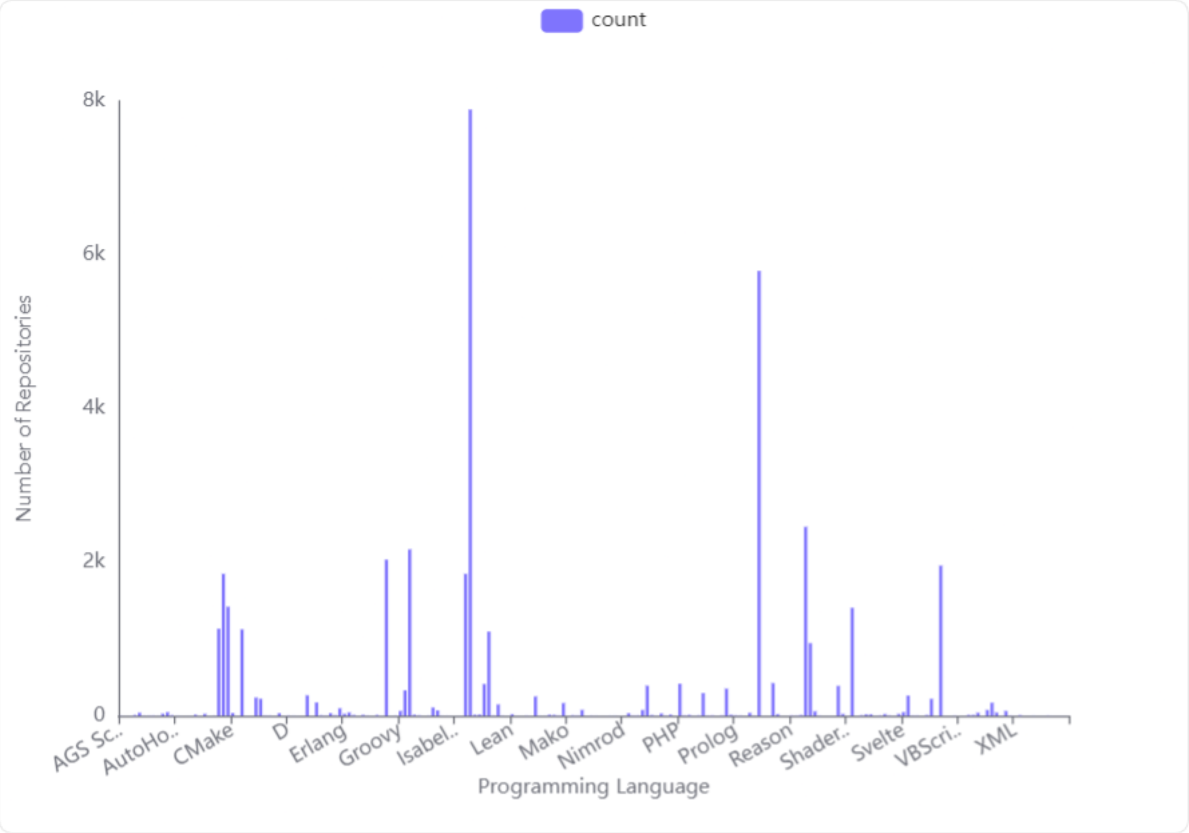
Sample Data: Licenses like 'agpl-3.0' and 'apache-2.0' have higher average stargazers counts (67.16 and 58.49, respectively), indicating potential popularity among developers.

Visual Insights: The bar chart highlights that certain license, such as 'bsd-3-clause', have notably higher average stargazers counts, suggesting they might be more attractive or widely used in popular projects.

Conclusion and Insights

- **Language Trends**: There is a significant variation in the average stargazers count across different programming languages, with some languages being more popular or having niche appeal.
- **License Trends**: Certain licenses are associated with higher average stargazers counts, possibly due to their permissiveness or popularity in open-source communities.

What are the most common programming languages used in the repositories.csv?



Language Distribution Analysis

- **Top Languages**: The most common programming languages are JavaScript, Python, Ruby, HTML, and Go.
- **JavaScript**: With 7,888 occurrences, JavaScript is the most frequently used language.
- **Python**: Python follows with 5,788 occurrences.
- **Ruby**: Ruby has 2,458 occurrences, making it the third most common language.

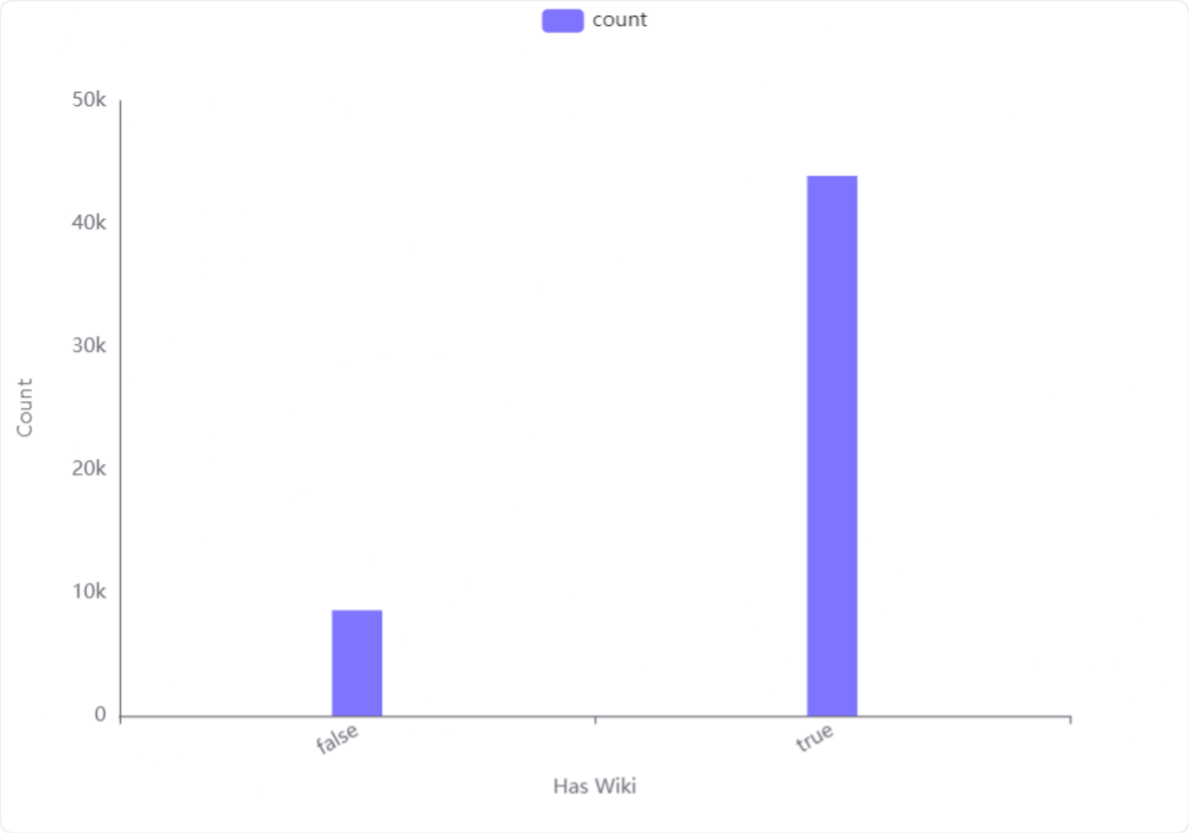
Visualization Insights

- **Bar Chart Representation**: The bar chart visually confirms that JavaScript and Python are the most dominant languages, with significant gaps compared to others.
- **Other Languages**: HTML and Go also show notable usage but are less common than the top two languages.

Conclusion and Insights

- **JavaScript and Python Dominance**: These two languages are the most prevalent in the dataset, indicating their widespread use in repositories.
- **Diverse Language Use**: While JavaScript and Python lead, a variety of other languages are also in use, reflecting diverse programming needs and preferences.

How does the distribution of repositories with a wiki compare to those without in the repositories.csv?



Analysis of 'has_wiki' Column

- **Repositories with a Wiki**: There are 43,912 repositories that have a wiki.
- **Repositories without a Wiki**: There are 8,576 repositories that do not have a wiki.

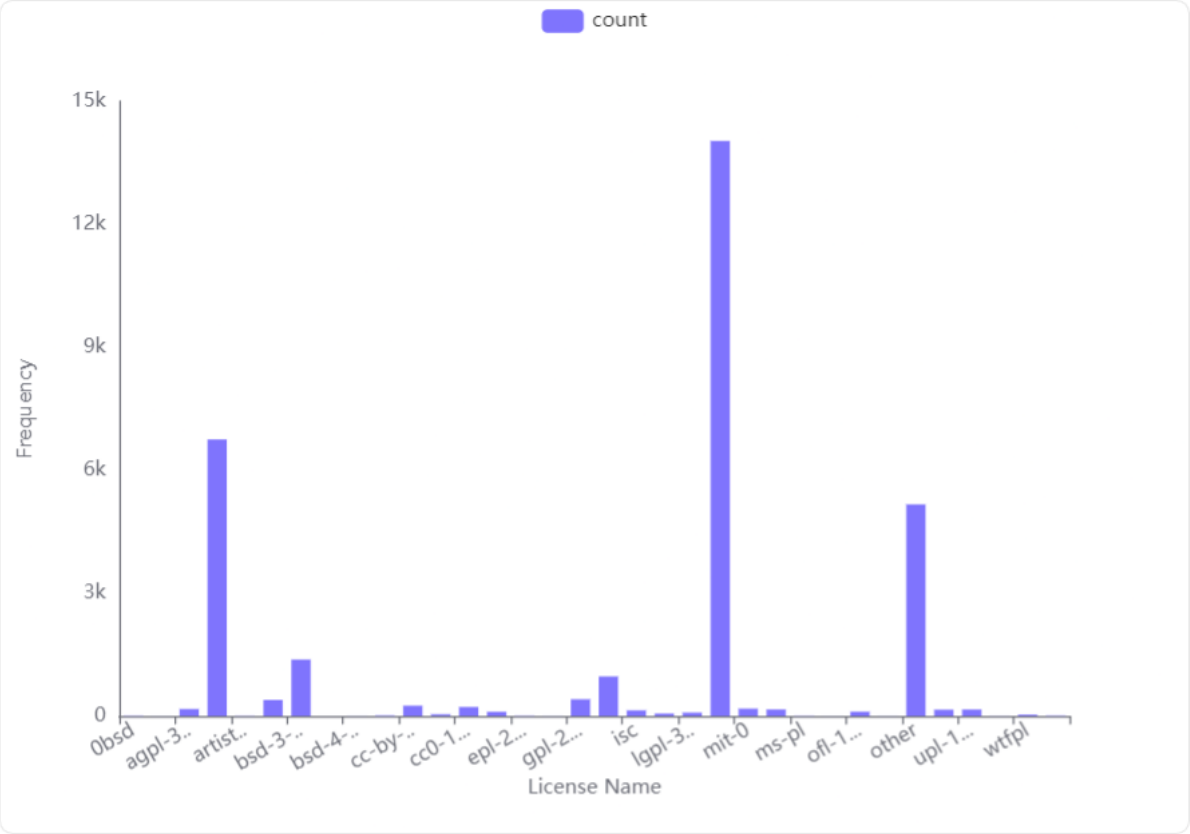
Visualization of Distribution

- **Bar Chart Insights**: The bar chart clearly shows that the number of repositories with a wiki is significantly higher than those without. The count for repositories with a wiki is more than five times greater than those without.

Conclusion and Insights

- **Dominance of Wiki Feature**: The majority of repositories include a wiki, indicating its importance or popularity among users.
- **Potential for Further Analysis**: Understanding why certain repositories opt out of using a wiki could provide insights into user preferences or the nature of the projects.

What is the distribution of license types in the repositories.csv?



License Data Analysis

- **Total License Types**: There are 34 different license types in the dataset.
- **Count Statistics**: The average count of repositories per license type is 912.12, with a standard deviation of 2720.71. The counts range from a minimum of 1 to a maximum of 14,033.

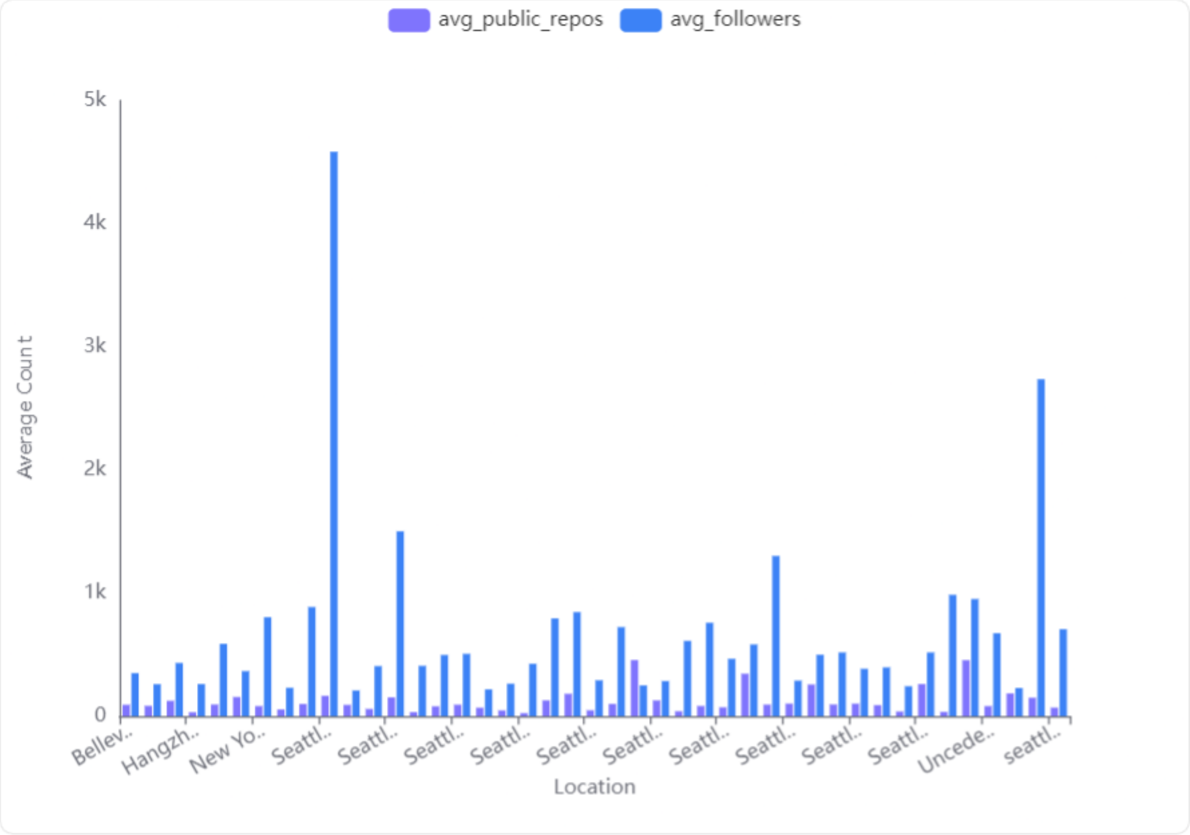
Visualization Insights

- **Most Common Licenses**: The bar chart shows that "apache-2.0" and "lgpl-3.0" are among the most frequently used licenses.
- **Other Notable Licenses**: "agpl-3.0" and "other" also have significant counts, indicating their popularity.
- **Less Common Licenses**: Many licenses, such as "0bsd" and "afl-3.0," have relatively low counts.

Conclusion and Insights

- **Dominant Licenses**: The distribution is heavily skewed towards a few licenses, with "apache-2.0" and "lgpl-3.0" being the most prevalent.
- **Diversity in Usage**: Despite the dominance of a few licenses, there is a wide variety of licenses in use, reflecting diverse licensing preferences.

What are the most common locations listed in the users.csv, and how do they correlate with the number of public repositories or followers?



Common Locations and Averages

- **Average Public Repositories**: The mean number of public repositories across the locations is approximately 120.79, with a standard deviation of 99.07. The range is from 24 to 455 repositories.

- **Average Followers**: The mean number of followers is about 679.26, with a standard deviation of 754.22, ranging from 208 to 4584.5 followers.

Visualization of Correlations

- **Seattle Dominance**: Seattle appears frequently and shows a high number of followers, indicating a strong correlation between this location and user popularity.

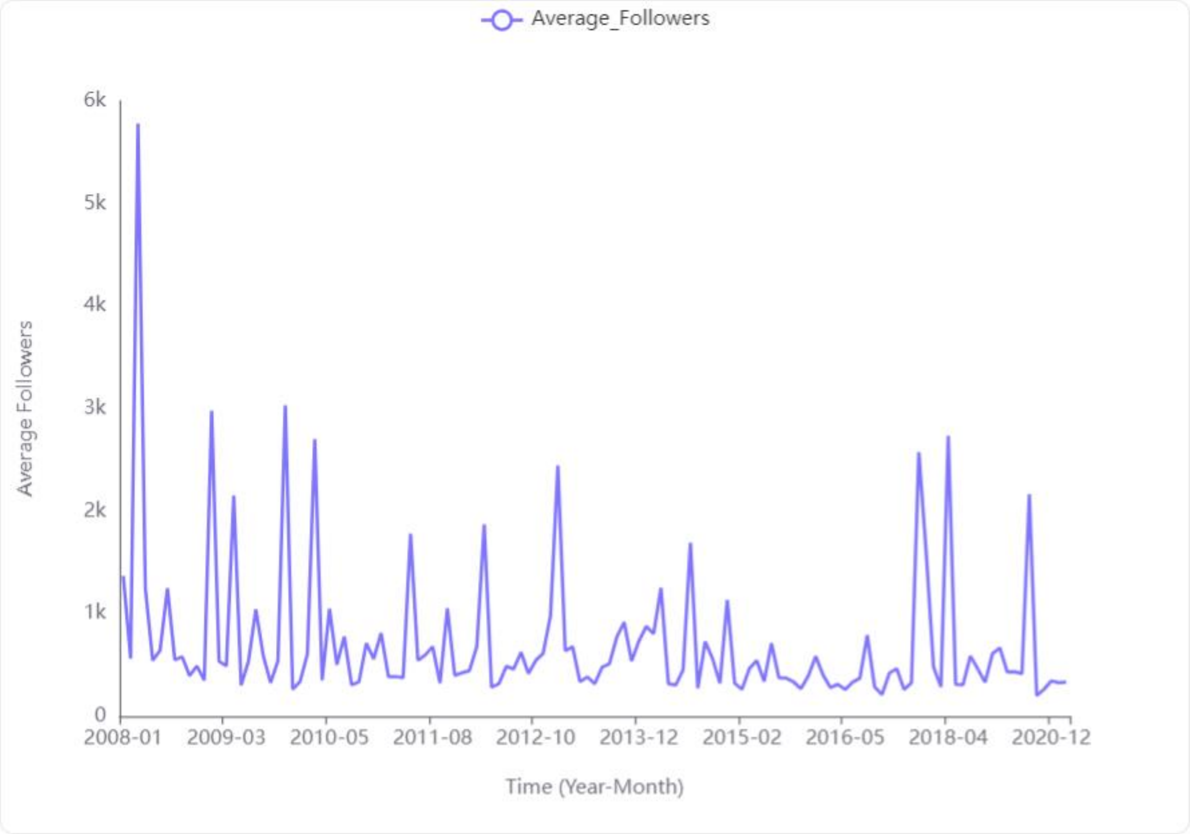
- **Variability Across Locations**: There is significant variability in both public repositories and followers across different locations, with some locations like Seattle showing much higher averages.

Conclusion and Insights

- **Seattle as a Hub**: Seattle is a prominent location with a high average number of followers, suggesting it may be a significant hub for influential users.

- **Diverse Distribution**: The distribution of public repositories and followers varies widely across locations, indicating that certain areas may have more active or popular users.

How does the number of followers change over time for users in the users.csv?



Data Analysis

- **Time Intervals**: The data is aggregated by month and year, showing changes in followers over time.
- **Average Followers**: The average number of followers per month ranges from 201 to 5780.75, with a mean of 722.53 and a standard deviation of 742.54.

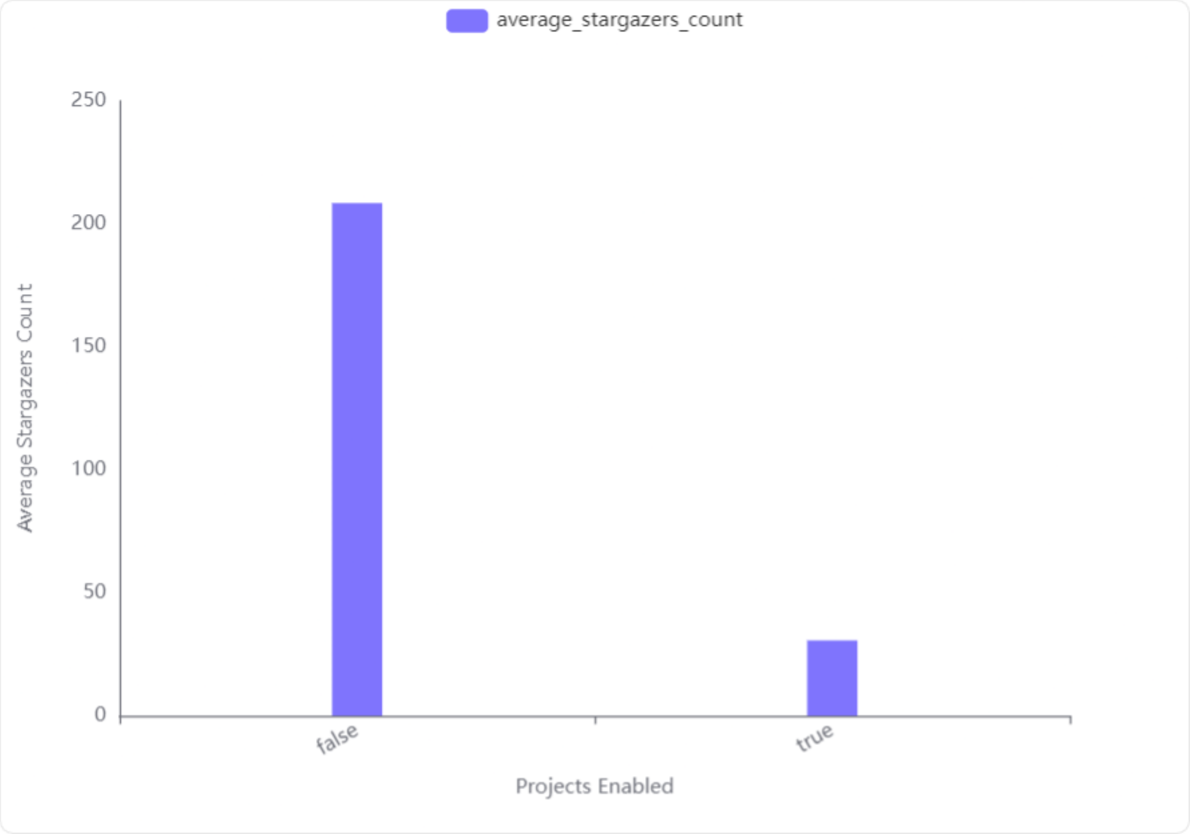
Visualization Insights

- **Trend Overview**: The line chart illustrates fluctuations in the average number of followers over time.
- **Peaks and Valleys**: There are noticeable peaks, particularly in early 2008 and sporadically in later years, indicating periods of increased follower activity.

Conclusion and Insights

- **Fluctuating Trends**: The number of followers shows significant variability over time, with some months experiencing sharp increases.
- **Potential Influences**: These fluctuations could be influenced by external factors such as events or changes in user activity. Further analysis could explore correlations with specific events or user behaviours.

Analyze the correlation between the number of stargazers and whether projects are enabled in the repositories.csv. Does having projects enabled influence the popularity of a repository?



Average Stargazers by Project Status

- **Repositories without projects enabled** have an average of **208.50 stargazers**.
- **Repositories with projects enabled** have an average of **30.63 stargazers**.

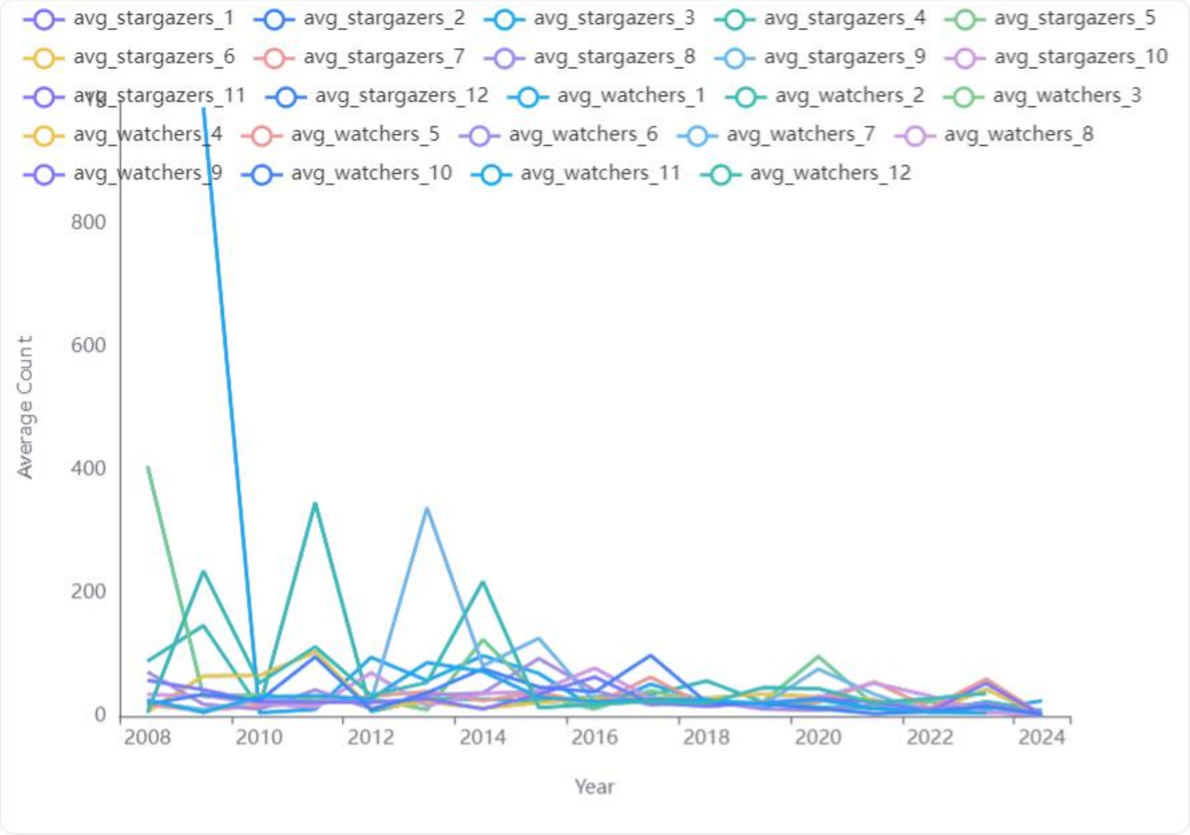
Visualization of Average Stargazers

- The bar chart clearly shows that repositories without projects enabled have significantly more stargazers compared to those with projects enabled.

Conclusion and Insights

- **Influence on Popularity**: Repositories without projects enabled tend to be more popular, as indicated by a higher average number of stargazers.
- **Potential Factors**: This could suggest that enabling projects does not necessarily correlate with increased popularity, or other factors might be influencing these results. Further investigation into additional variables could provide more insights.

Analyze the distribution of stargazers_count and watchers_count to identify any patterns or trends. What do these patterns suggest about the popularity of repositories over time?



Distribution Analysis

- **Mean and Standard Deviation**: Both the stargazers and watchers have a mean of approximately 34.60 and a high standard deviation of 535.64, indicating a wide spread in the data.

- **Quartiles**: The 25th, 50th, and 75th percentiles are all at 0, 0, and 2 respectively, suggesting that most repositories have very few stargazers and watchers.

- **Maximum Values**: The maximum count for both metrics is 54,583, showing that a few repositories are extremely popular.

Visualization of Distribution

- **Histogram Insights**: The histogram shows that the majority of repositories have low counts, with a few having very high counts, confirming the skewed distribution.

Popularity Over Time

- **Yearly Trends**: The average stargazers and watchers per month show a decreasing trend over the years, with peaks in early years like 2008 and 2009.

- **Monthly Variations**: There are fluctuations within each year, but the overall trend is a decline in average counts.

Visualization of Trends

- **Line Chart Observations**: The line chart illustrates that the popularity of repositories, as measured by stargazers and watchers, peaked around 2009 and has generally declined since then.

Conclusion and Insights

- **Skewed Distribution**: The data is heavily skewed, with most repositories having few stargazers and watchers, while a small number have very high counts.
- **Declining Popularity**: Over time, the average popularity of repositories has decreased, suggesting changes in user engagement or repository creation trends.