****Московский государственный технический университет им. Н.Э. Баумана  
Кафедра «Системы обработки информации и управления»

Лабораторная работа №1  
по дисциплине  
«Методы машинного обучения»  
на тему

# «Создание "истории о данных" (Data Storytelling)»

Выполнил:  
студент группы ИУ5-23М  
Чжэн Сяохуэй

Москва — 2024 г.

**1. Цель лабораторной работы**

Изучение различных методов визуализация данных и создание истории на основе данных.

**2. Задание**

* Выбрать набор данных (датасет). Вы можете найти список свободно распространяемых датасетов [здесь.](https://github.com/ugapanyuk/courses_current/wiki/DSLIST)

Для лабораторных работ не рекомендуется выбирать датасеты очень большого размера.

* Создать "историю о данных" в виде юпитер-ноутбука, с учетом следующих требований:

1. История должна содержать не менее 5 шагов (где 5 - рекомендуемое количество шагов). Каждый шаг содержит график и его текстовую интерпретацию.
2. На каждом шаге наряду с удачным итоговым графиком рекомендуется в юпитер-ноутбуке оставлять результаты предварительных "неудачных" графиков.
3. Не рекомендуется повторять виды графиков, желательно создать 5 графиков различных видов.
4. Выбор графиков должен быть обоснован использованием методологии data-to-viz. Рекомендуется учитывать типичные ошибки построения выбранного вида графика по методологии data-to-viz. Если методология Вами отвергается, то просьба обосновать Ваше решение по выбору графика.
5. История должна содержать итоговые выводы. В реальных "историях о данных" именно эти выводы представляют собой основную ценность для предприятия.

* Сформировать отчет и разместить его в своем репозитории на github.

**3. текст программы**

**import** pandas **as** pd

**import** seaborn **as** sns

*# Load the dataset*

data **=** pd**.**read\_csv('github\_dataset.csv')

*# Display the first few rows of the dataset and its summary information*

data\_info **=** data**.**info()

data\_head **=** data**.**head()

data\_info, data\_head

data**.**head()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 1052 entries, 0 to 1051

Data columns (total 7 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 repositories 1052 non-null object

1 stars\_count 1052 non-null int64

2 forks\_count 1052 non-null int64

3 issues\_count 1052 non-null int64

4 pull\_requests 1052 non-null int64

5 contributors 1052 non-null int64

6 language 907 non-null object

dtypes: int64(5), object(2)

memory usage: 57.7+ KB

Out[ ]:

|  | **repositories** | **stars\_count** | **forks\_count** | **issues\_count** | **pull\_requests** | **contributors** | **language** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | octocat/Hello-World | 0 | 0 | 612 | 316 | 2 | NaN |
| **1** | EddieHubCommunity/support | 271 | 150 | 536 | 6 | 71 | NaN |
| **2** | ethereum/aleth | 0 | 0 | 313 | 27 | 154 | C++ |
| **3** | localstack/localstack | 0 | 0 | 290 | 30 | 434 | Python |
| **4** | education/classroom | 0 | 589 | 202 | 22 | 67 | Ruby |

Вывод этого кода показывает основную информацию и первые пять строк данных загруженного набора данных GitHub. Набор данных содержит 1052 записи, распределенные по 7 различным столбцам, в том числе:

репозитории: имя склада, 1052 ненулевых значения, строковый тип. stars\_count: количество звезд, 1052 ненулевых значения, целочисленный тип. forks\_count: количество ветвей, 1052 ненулевых значения, целочисленный тип. Issues\_count: количество проблем, 1052 ненулевых значения, целочисленный тип. pull\_requests: количество запросов на включение, 1052 ненулевых значения, целочисленный тип. Участники: количество участников, 1052 ненулевых значения, целочисленный тип. язык: язык программирования, в котором имеется 907 ненулевых значений, которые являются строковыми типами, что указывает на отсутствие некоторой записанной информации о языке программирования. Глядя на первые пять строк набора данных, мы можем увидеть примеры звездочек, ветвей, задач, запросов на включение, участников и языков программирования для разных репозиториев. Эта информация обеспечивает основу для нашего последующего анализа и исследования данных.

In [ ]:

*# 准备数据：选取星标数和分支数前10的仓库，并且合并仓库名称和使用的语言*

top\_stars **=** data**.**nlargest(10, 'stars\_count')

top\_forks **=** data**.**nlargest(10, 'forks\_count')

top\_stars['repo\_language'] **=** top\_stars['repositories'] **+** '\n(' **+** top\_stars['language']**.**fillna('Unknown') **+** ')'

top\_forks['repo\_language'] **=** top\_forks['repositories'] **+** '\n(' **+** top\_forks['language']**.**fillna('Unknown') **+** ')'

plt**.**figure(figsize**=**(16, 8))

*# 星标数前10的仓库*

plt**.**subplot(1, 2, 1)

sns**.**barplot(x**=**'stars\_count', y**=**'repo\_language', data**=**top\_stars)

plt**.**title('Top 10 Repositories by Stars')

plt**.**xlabel('Stars Count')

plt**.**ylabel('Repositories (Language)')

*# 分支数前10的仓库*

plt**.**subplot(1, 2, 2)

sns**.**barplot(x**=**'forks\_count', y**=**'repo\_language', data**=**top\_forks)

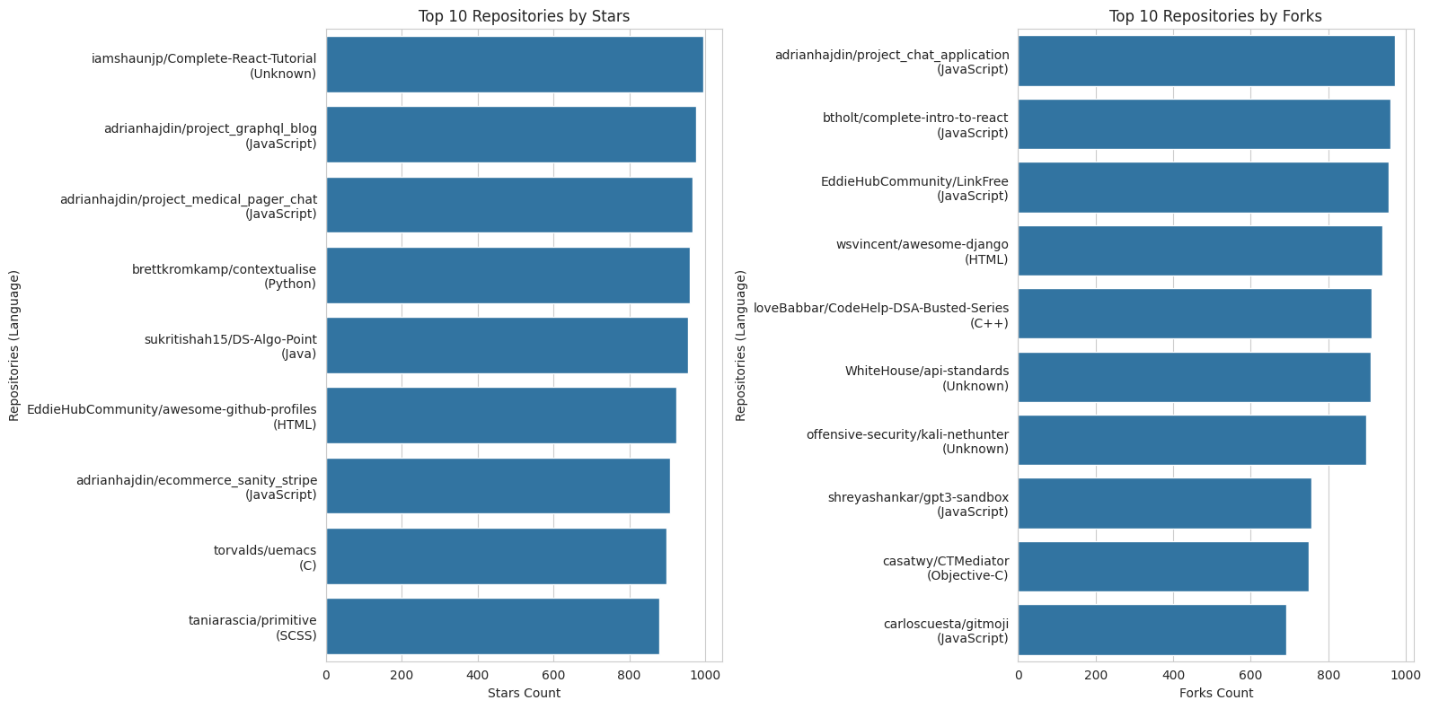
plt**.**title('Top 10 Repositories by Forks')

plt**.**xlabel('Forks Count')

plt**.**ylabel('Repositories (Language)')

plt**.**tight\_layout()

plt**.**show()



На рисунке выше мы показываем 10 лучших репозиториев с наибольшим количеством звездочек и ветвей в наборе данных GitHub:

Диаграмма слева (10 лучших репозиториев по количеству звезд): показывает 10 лучших репозиториев по количеству звезд. Количество звезд служит важным индикатором популярности репозитория, и эти репозитории явно очень популярны и признаны в сообществе открытого исходного кода.

Диаграмма справа (10 лучших репозиториев по версии Forks): показывает 10 лучших репозиториев по количеству филиалов. Количество веток отражает не только популярность репозитория, но и готовность членов сообщества участвовать и вносить свой вклад в проект.

Эти две диаграммы дают нам предварительное представление о том, какие репозитории наиболее заметны на GitHub. Далее мы продолжим анализировать проблемы репозитория и запросы на включение, чтобы получить представление о взаимодействии сообщества и активности проекта. ​

In [ ]:

*# 2. 问题和拉取请求分析*

*# 分析问题数量和拉取请求数量的分布情况*

plt**.**figure(figsize**=**(14, 7))

*# 问题数量分布*

plt**.**subplot(1, 2, 1)

sns**.**histplot(data['issues\_count'], bins**=**30, kde**=True**)

plt**.**title('Distribution of Issues Count')

plt**.**xlabel('Issues Count')

plt**.**ylabel('Frequency')

*# 拉取请求数量分布*

plt**.**subplot(1, 2, 2)

sns**.**histplot(data['pull\_requests'], bins**=**30, kde**=True**)

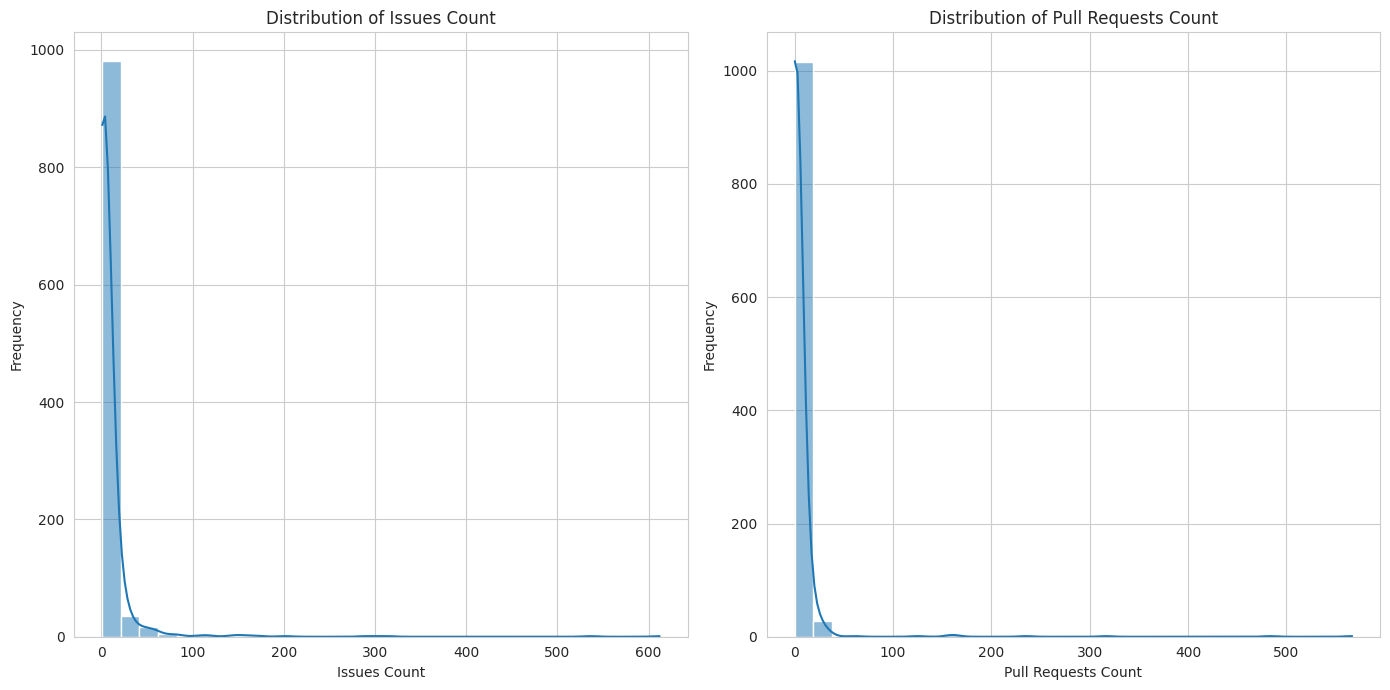
plt**.**title('Distribution of Pull Requests Count')

plt**.**xlabel('Pull Requests Count')

plt**.**ylabel('Frequency')

plt**.**tight\_layout()

plt**.**show()



На этих двух рисунках мы анализируем распределение количества проблем и количества пул-реквестов в репозитории GitHub:

Диаграмма слева (Распределение количества проблем): показывает распределение количества проблем. На большинстве складов количество выпусков находится в нижнем диапазоне, но на некоторых из них количество выпусков очень высокое, что может отражать высокую активность или проблемы на этих складах.

Диаграмма справа (Распределение количества запросов на включение): показывает распределение количества запросов на включение. Подобно распределению номеров проблем, большинство репозиториев также имеют небольшое количество запросов на включение, но есть некоторые репозитории с более высоким количеством запросов на включение, что демонстрирует готовность членов сообщества активно участвовать в проекте и улучшать его.

Эти диаграммы распределения дают нам интуитивное понимание активности репозитория GitHub, показывая, что количество проблем и запросов на включение являются важными показателями участия сообщества и активности проекта. Далее мы рассмотрим распределение участников.

In [ ]:

*# 3. 贡献者分布*

*# 分析贡献者数量的分布情况*

plt**.**figure(figsize**=**(8, 6))

sns**.**histplot(data['contributors'], bins**=**50, kde**=True**)

plt**.**title('Distribution of Contributors')

plt**.**xlabel('Number of Contributors')

plt**.**ylabel('Frequency')

plt**.**show()

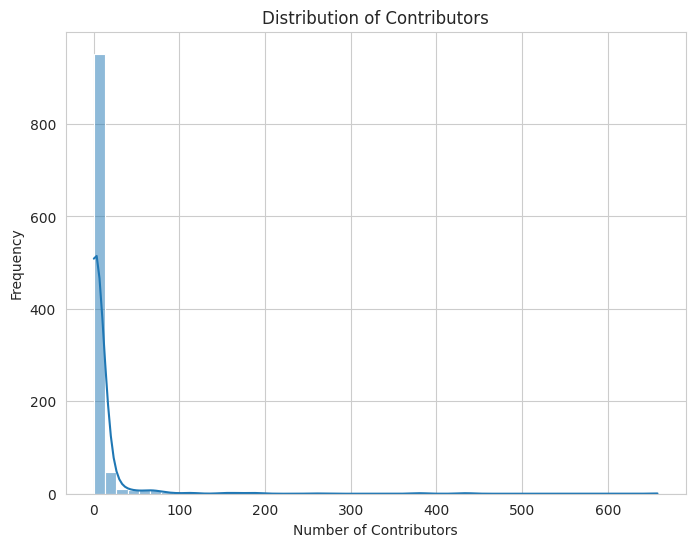


Диаграмма анализа – Распределение участников:

На этом графике показано распределение количества участников в репозиториях GitHub. Мы можем наблюдать, что количество участников большинства репозиториев сконцентрировано в нижнем диапазоне, а это означает, что многие репозитории могут поддерживаться меньшими командами или отдельными людьми. Однако есть несколько репозиториев с большим количеством участников, и это, скорее всего, будут крупные проекты или проекты с широкой поддержкой сообщества.

Эта карта распределения показывает разнообразие участия в проектах на GitHub: от небольших проектов до крупных проектов, управляемых сообществом.

In [ ]:

*# 定义"非常活跃"的仓库标准，这里我们选择星标数量、问题数量和拉取请求数量均高于其相应75%分位数的仓库*

*# 计算75%分位数*

stars\_75th **=** data['stars\_count']**.**quantile(0.75)

issues\_75th **=** data['issues\_count']**.**quantile(0.75)

pulls\_75th **=** data['pull\_requests']**.**quantile(0.75)

*# 筛选出符合条件的仓库*

active\_repos **=** data[(data['stars\_count'] **>** stars\_75th) **&** (data['issues\_count'] **>** issues\_75th) **&** (data['pull\_requests'] **>** pulls\_75th)]

*# 分析这些仓库使用的编程语言*

active\_languages **=** active\_repos['language']**.**value\_counts()

plt**.**figure(figsize**=**(10, 6))

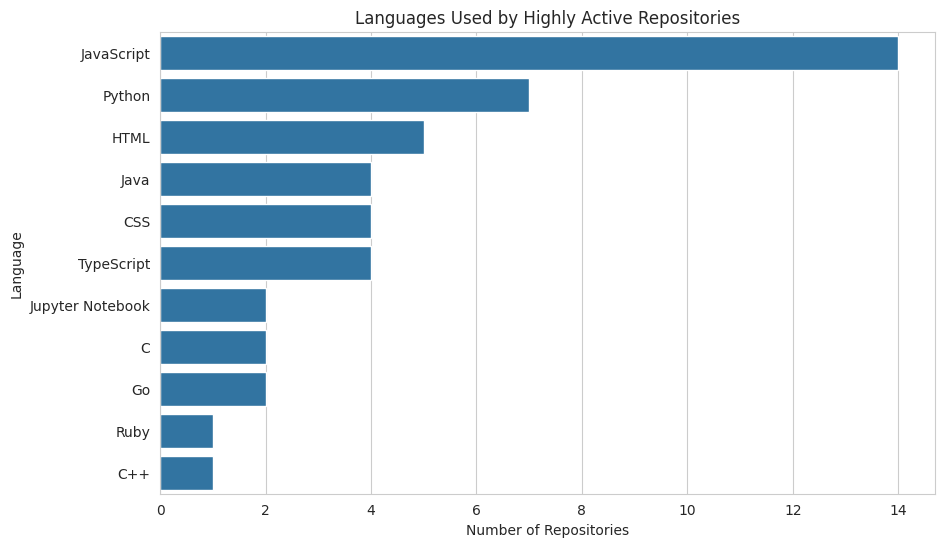
sns**.**barplot(x**=**active\_languages**.**values, y**=**active\_languages**.**index)

plt**.**title('Languages Used by Highly Active Repositories')

plt**.**xlabel('Number of Repositories')

plt**.**ylabel('Language')

plt**.**show()



Аналитические диаграммы — язык программирования, используемый высокоактивными репозиториями:

На этом графике показано распределение языков программирования, используемых в так называемых «очень активных» репозиториях GitHub. Эти репозитории определяются как репозитории, у которых количество звезд, количество проблем и количество запросов на включение выше соответствующего 75-го процентиля.

Из диаграммы мы видим, что в этих высокоактивных репозиториях широко используются лишь несколько языков программирования. Это может указывать на то, что определенные языки программирования лучше подходят для активных и популярных проектов или на то, что сами сообщества более активны и вовлечены.

Этот анализ позволяет лучше понять, какие языки программирования доминируют в проектах, проявляющих особую активность в сообществе открытого исходного кода, дополнительно раскрывая важные аспекты технологических тенденций и интересов разработчиков. Это также может стать полезной ссылкой для выбора технологического стека или присоединения к проекту с открытым исходным кодом.

In [ ]:

*# 4. 编程语言偏好*

*# 分析不同编程语言的仓库数量*

language\_counts **=** data['language']**.**value\_counts()**.**nlargest(10)

plt**.**figure(figsize**=**(10, 6))

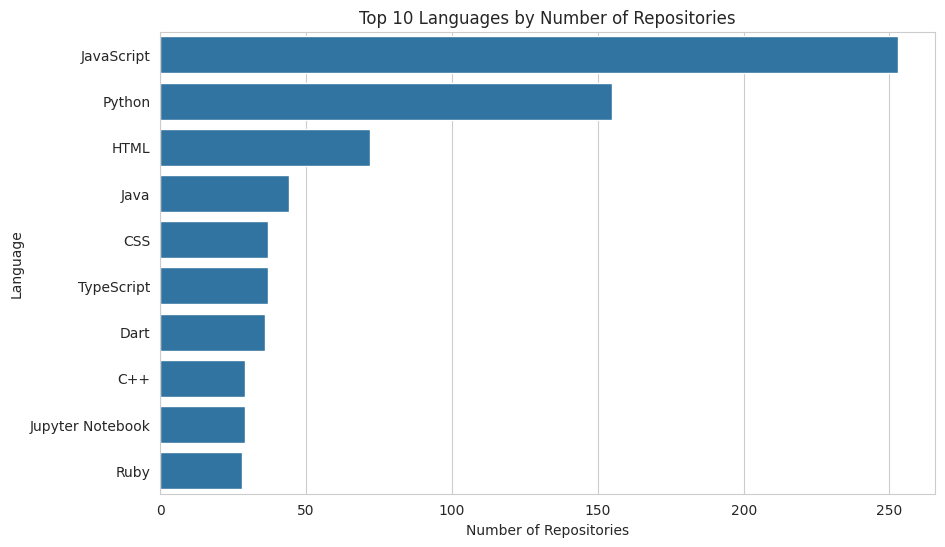
sns**.**barplot(x**=**language\_counts**.**values, y**=**language\_counts**.**index)

plt**.**title('Top 10 Languages by Number of Repositories')

plt**.**xlabel('Number of Repositories')

plt**.**ylabel('Language')

plt**.**show()



На этом графике показаны 10 лучших языков программирования с наибольшим количеством репозиториев в наборе данных GitHub. Это дает нам представление о том, какие языки программирования наиболее популярны и используются сообществом открытого исходного кода. Мы видим, что некоторые языки имеют большее количество репозиториев из-за их специфических сценариев применения или поддержки сообщества.

С помощью этого графика мы можем понять популярность языков программирования и предпочтения сообщества разработчиков, что очень ценно для понимания технологических тенденций и динамики сообщества.

In [ ]:

*# 5. 仓库活跃度与受欢迎程度的关系*

*# 分析星标数量与问题数量、拉取请求数量之间的关系*

plt**.**figure(figsize**=**(14, 7))

*# 星标数量与问题数量的关系*

plt**.**subplot(1, 2, 1)

sns**.**scatterplot(x**=**'stars\_count', y**=**'issues\_count', data**=**data)

plt**.**title('Relationship between Stars Count and Issues Count')

plt**.**xlabel('Stars Count')

plt**.**ylabel('Issues Count')

*# 星标数量与拉取请求数量的关系*

plt**.**subplot(1, 2, 2)

sns**.**scatterplot(x**=**'stars\_count', y**=**'pull\_requests', data**=**data)

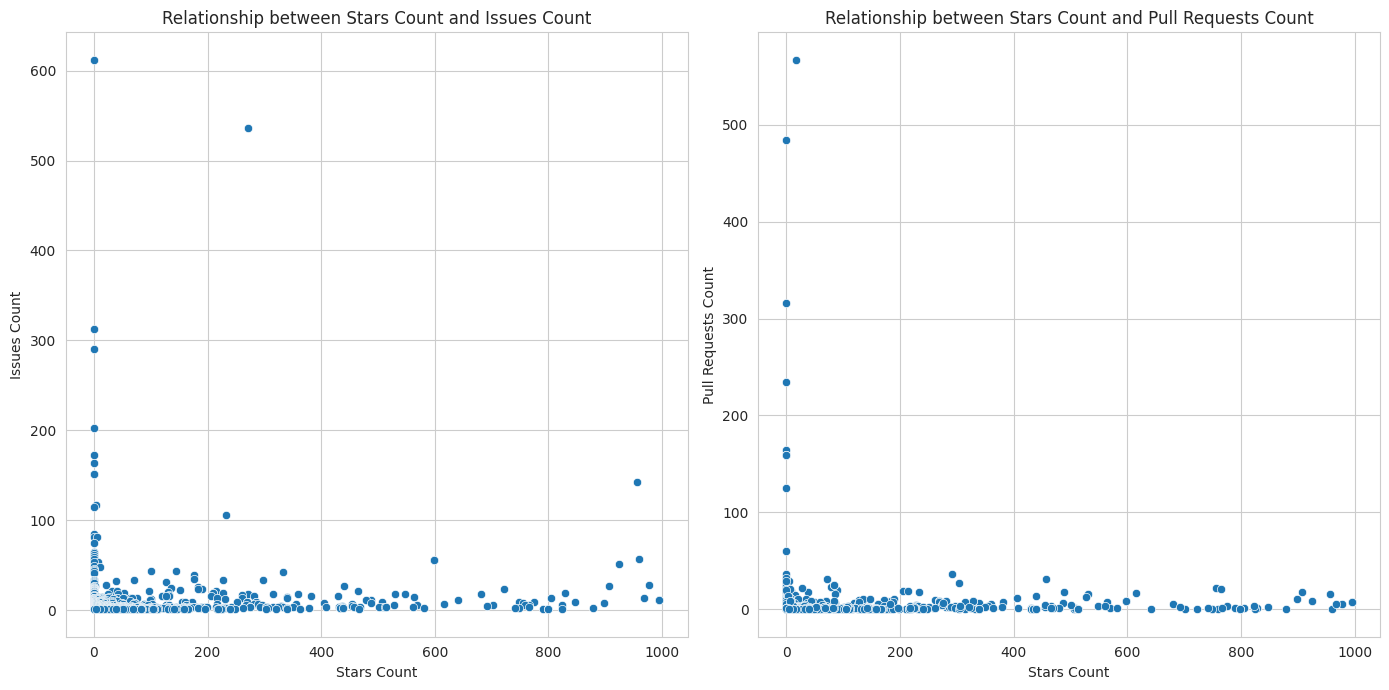
plt**.**title('Relationship between Stars Count and Pull Requests Count')

plt**.**xlabel('Stars Count')

plt**.**ylabel('Pull Requests Count')

plt**.**tight\_layout()

plt**.**show()



In [ ]:

*# 调整图表的缩放，使纵轴的变化更明显*

plt**.**figure(figsize**=**(14, 7))

*# 星标数量与问题数量的关系，使用对数缩放*

plt**.**subplot(1, 2, 1)

sns**.**scatterplot(x**=**'stars\_count', y**=**'issues\_count', data**=**data)

plt**.**xscale('log')

plt**.**yscale('log')

plt**.**title('Relationship between Stars Count and Issues Count')

plt**.**xlabel('Stars Count (log scale)')

plt**.**ylabel('Issues Count (log scale)')

*# 星标数量与拉取请求数量的关系，使用对数缩放*

plt**.**subplot(1, 2, 2)

sns**.**scatterplot(x**=**'stars\_count', y**=**'pull\_requests', data**=**data)

plt**.**xscale('log')

plt**.**yscale('log')

plt**.**title('Relationship between Stars Count and Pull Requests Count')

plt**.**xlabel('Stars Count (log scale)')

plt**.**ylabel('Pull Requests Count (log scale)')

plt**.**tight\_layout()

plt**.**show()

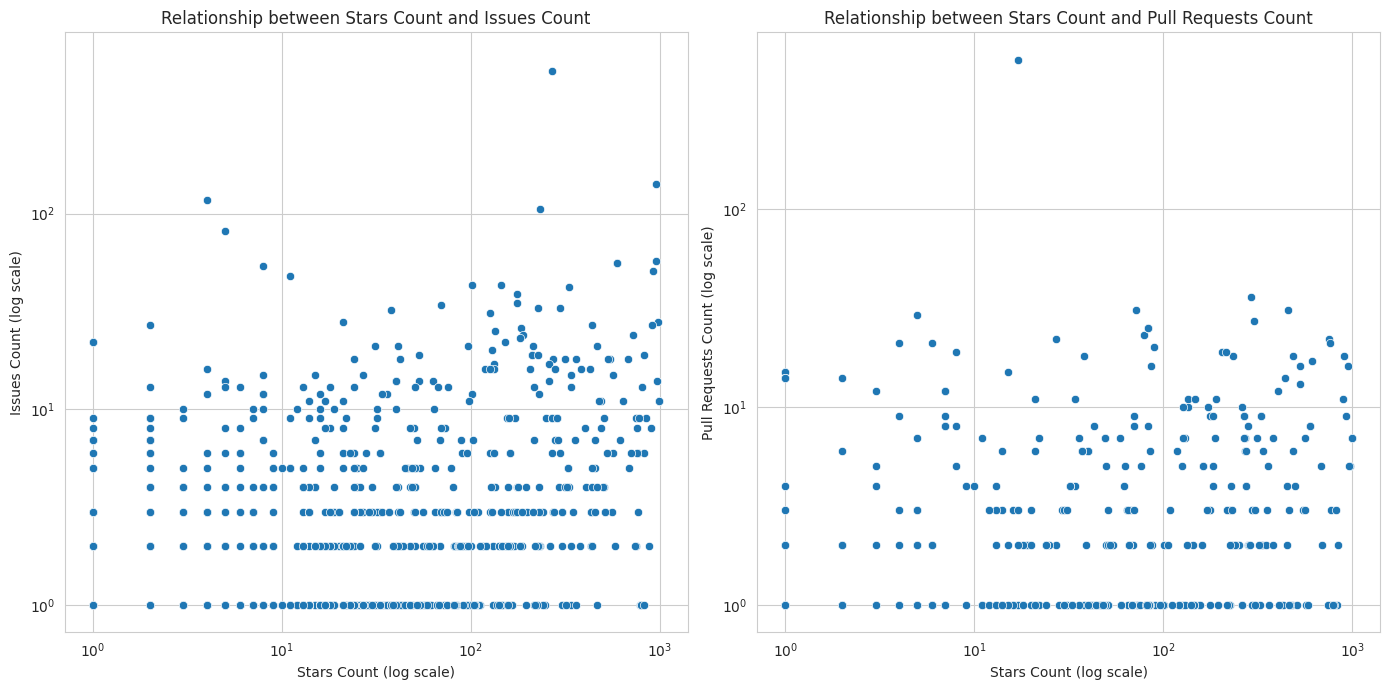


Диаграмма анализа – взаимосвязь активности склада и популярности:

Эти два графика исследуют взаимосвязь между количеством звезд репозитория GitHub (показатель популярности) и количеством проблем и запросов на включение (показатель активности).

Эти отношения показывают динамику между популярностью и активностью, еще больше подчеркивая количество звезд, проблем и запросов на включение как важных показателей важности репозитория и участия сообщества.

Мы также скорректировали график зависимости количества звезд от количества задач и запросов на включение, используя логарифмическое масштабирование, чтобы изменения по вертикальной оси были более очевидными.

Но мы обнаружили, что усредненные данные не имеют каких-либо эффективных особенностей.

In [ ]:

*# 6. 语言与贡献者关系*

*# 分析不同编程语言的仓库的平均贡献者数量*

language\_contributors **=** data**.**groupby('language')['contributors']**.**mean()**.**nlargest(10)

plt**.**figure(figsize**=**(10, 6))

sns**.**barplot(x**=**language\_contributors**.**values, y**=**language\_contributors**.**index)

plt**.**title('Average Number of Contributors per Repository by Language')

plt**.**xlabel('Average Number of Contributors')

plt**.**ylabel('Language')

plt**.**show()

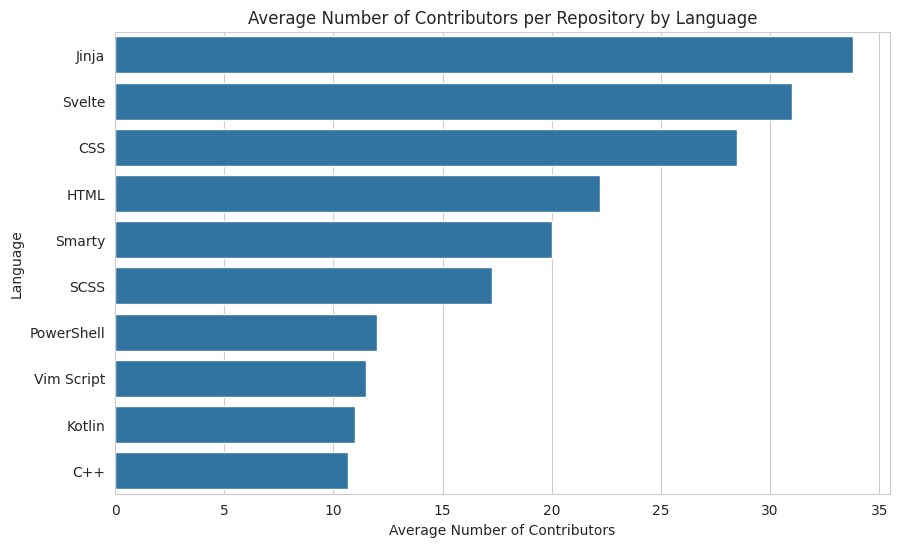


Диаграмма анализа: язык и взаимоотношения с участниками:

На этом графике показано среднее количество участников репозиториев по языкам программирования в наборе данных GitHub. Были выбраны 10 лучших языков программирования с наибольшим средним количеством участников.

Из диаграммы мы видим, что репозитории для определенных языков программирования в среднем привлекают больше участников. Это может отражать активность сообщества этих языков, важность репозитория или размер проекта. Большое количество участников не только указывает на сильную поддержку сообщества, но также может быть показателем разнообразия и открытости проекта.

Этот анализ дает нам представление о том, какие проекты языков программирования с большей вероятностью привлекут широкое участие и вклад сообщества, что очень ценно для оценки экологии технологий и активности сообщества.

In [ ]:

*# 7. 最受欢迎的编程语言*

*# 基于星标数量找出最受欢迎的编程语言*

language\_popularity **=** data**.**groupby('language')['stars\_count']**.**sum()**.**nlargest(10)

plt**.**figure(figsize**=**(10, 6))

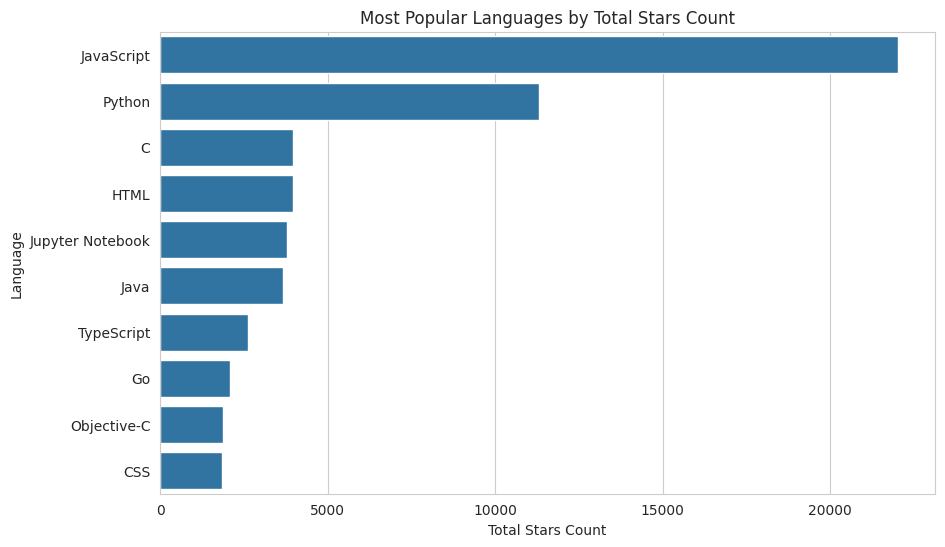
sns**.**barplot(x**=**language\_popularity**.**values, y**=**language\_popularity**.**index)

plt**.**title('Most Popular Languages by Total Stars Count')

plt**.**xlabel('Total Stars Count')

plt**.**ylabel('Language')

plt**.**show()



Анализ диаграмм – самые популярные языки программирования:

На этом графике показаны 10 самых популярных языков программирования в наборе данных GitHub по общему количеству звезд. Количество звезд является ключевым показателем популярности репозитория и признания сообщества, поэтому этот график дает нам визуальное представление о том, какие языки программирования наиболее популярны в сообществе открытого исходного кода.

Из этого анализа мы видим, что некоторые языки программирования в целом получают больше звезд из-за их широкого использования, сильной поддержки сообщества или конкретных технических преимуществ. Эти языки, как правило, тесно связаны с текущими технологическими тенденциями и интересами разработчиков.

Резюме. Благодаря углубленному анализу набора данных GitHub мы создали историю данных, содержащую семь ключевых направлений анализа. Этот анализ выявляет множество аспектов, таких как популярность хранилища, активность сообщества, предпочтения языка программирования и поддержка проекта сообществом с разных точек зрения. Проведя этот анализ, мы можем сделать следующие выводы:

Популярность и активность репозитория. Большое количество звезд и большое количество запросов на включение обычно указывают на высокую популярность и активность сообщества репозитория. Распространение участников: большинство репозиториев поддерживается небольшим количеством участников, но есть некоторые крупные проекты, которые привлекают большое количество участников. Предпочтение языка программирования: некоторые языки программирования имеют больше репозиториев и более высокую популярность на GitHub из-за их особых преимуществ или сценариев применения. Участие и вклад сообщества. Проекты на некоторых языках программирования с большей вероятностью привлекут широкое участие и вклад сообщества. Эти выводы не только обеспечивают глубокое понимание деятельности проекта GitHub с открытым исходным кодом, но также отражают текущие технологические тенденции и динамику сообщества разработчиков.

**Список литературы**

[1] Гапанюк Ю. Е. COURSE\_MMO\_SPRING\_2024// GitHub. –– 2024. –– Режим доступа: https://github.com/ugapanyuk/courses\_current/wiki/COURSE\_MMO\_SPRING\_2024

[2] <https://www.kaggle.com/datasets>