

# github 用户行为数据分析

--利用 kimi 大模型辅助分析

## 1.如何使用大模型给出正确准确的答案



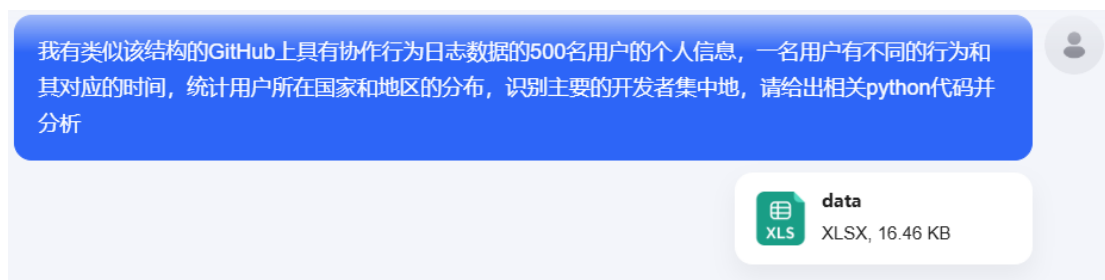
询问提示词智能体，获得模版，根据具体需求进行修改即可

## 2. 人口统计分析

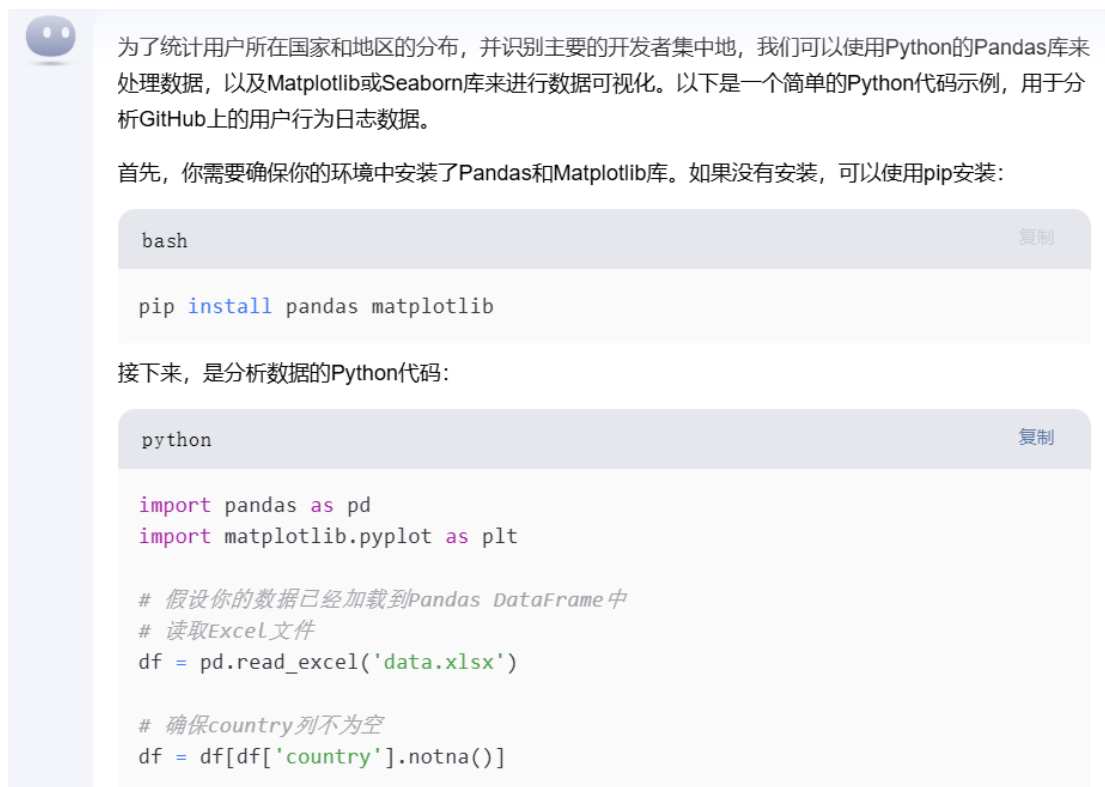
国家和地区分布: 统计用户所在国家和地区的分布, 识别主要的开发者集中地。



先输入提示词, 接着提取部分数据给 kimi 进行分析



给出结果部分如下:



实际使用代码需要替换文件为使用的 csv 文件

观察发现，该代码对同一用户重复统计，向 kimi 提出要求改正

注意，一个用户可能会出现多次，且一共只有500个

了解到一个用户可能会出现多次，且一共有500个用户的情况，我们需要对代码进行一些调整。我们将使用value\_counts()方法来统计每个国家的开发者数量，但是这次我们将基于user\_id去重，以确保每个用户只被计算一次。以下是修改后的Python代码：

```
python

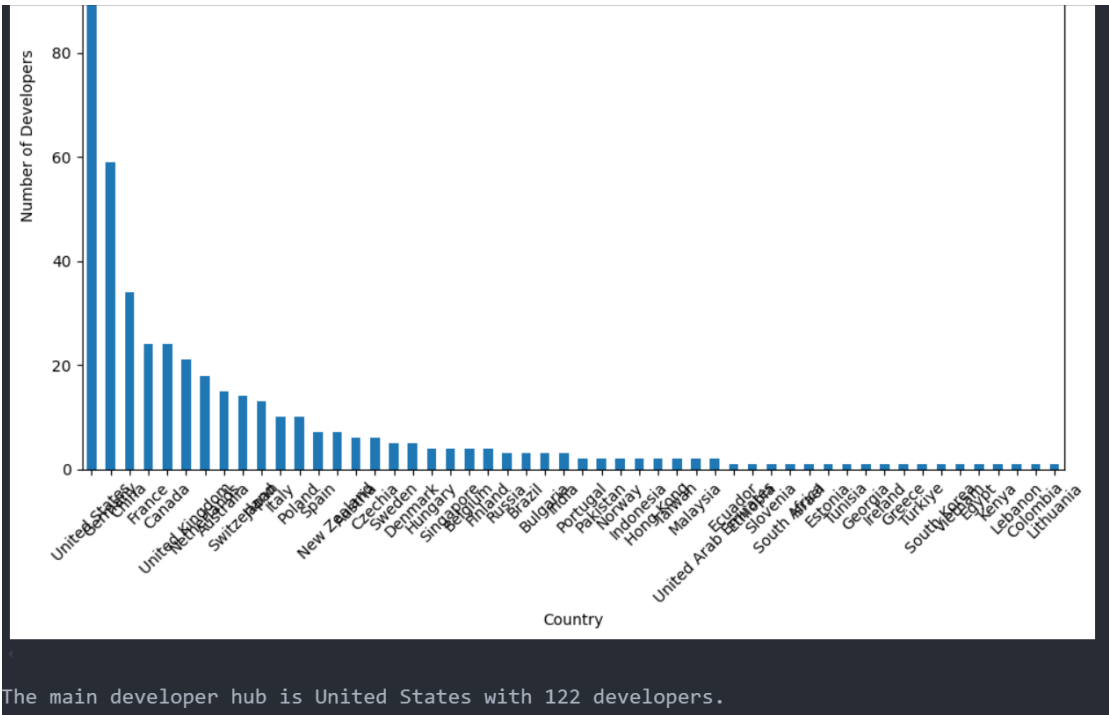
import pandas as pd
import matplotlib.pyplot as plt

# 假设你的数据已经加载到Pandas DataFrame中
# 读取Excel文件
df = pd.read_excel('data.xlsx')

# 去重，确保每个用户只被计算一次
unique_users = df.drop_duplicates(subset='user_id')

# 统计每个国家的开发者数量
```

改正后运行结果为：



可以明显发现美国，中国等科技发展大国开发者数量更多

城市级别分布：分析主要城市的开发者密度，发现技术热点区域。

城市级别分布：分析主要城市的开发者密度，发现技术热点区域

为了分析城市级别的开发者密度并发现技术热点区域，我们需要对数据集中的城市字段进行分析。我们将统计每个城市的开发者数量，并使用可视化技术来展示结果。以下是进行这种分析的Python代码：

```
python

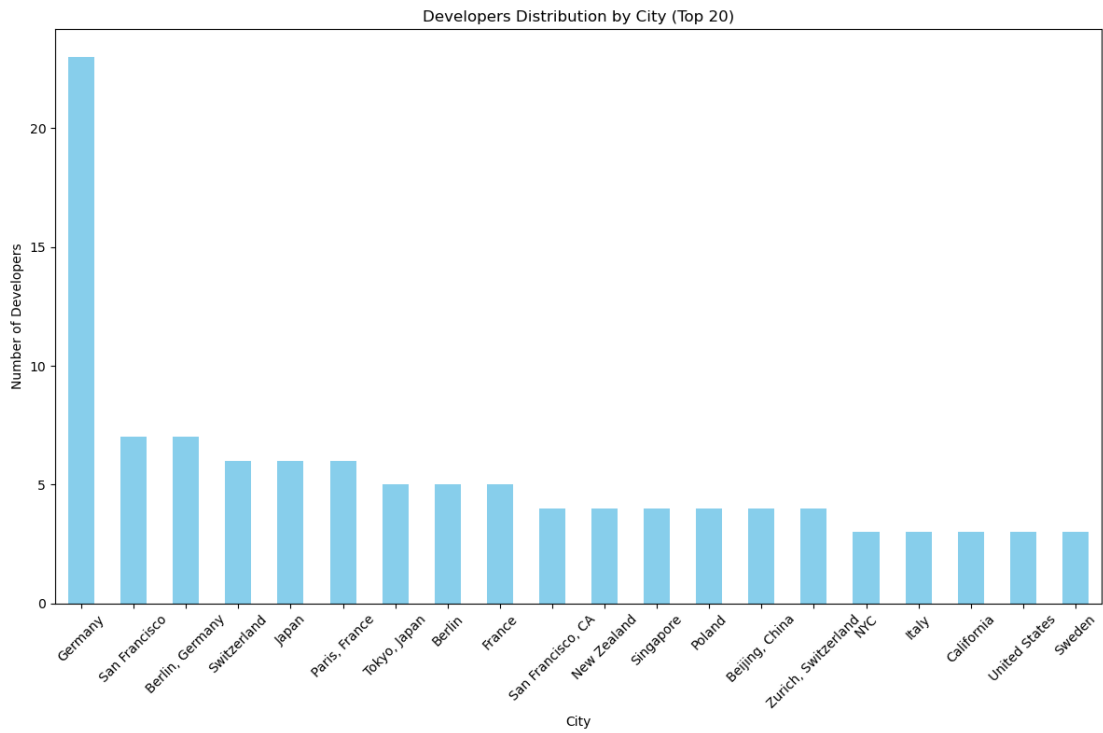
import pandas as pd
import matplotlib.pyplot as plt

# 假设你的数据已经加载到Pandas DataFrame中
# 读取Excel文件
df = pd.read_excel('data.xlsx')

# 去重，确保每个用户只被计算一次
unique_users = df.drop_duplicates(subset='user_id')

# 统计每个城市的开发者数量
city_distribution = unique_users['location'].value_counts()
```

运行结果如下：

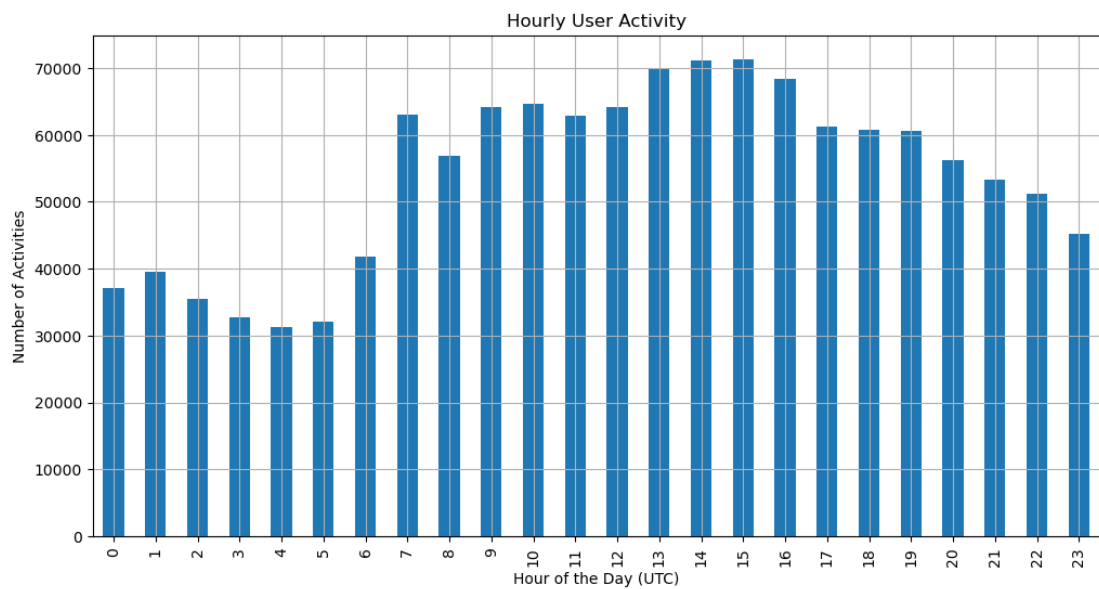


```
Top 10 Tech Hotspots by Developer Density:
location
Germany                23
San Francisco           7
Berlin, Germany         7
Switzerland             6
Japan                   6
Paris, France           6
Tokyo, Japan            5
Berlin                  5
France                  5
San Francisco, CA       4
Name: count, dtype: int64
```

德国，日本等城市开发者较为密集

时区分布:了解用户的时区分布，分析不同地区用户的协作时间模式

运 行 结 果 :



大部分用户在白天进行开发协作，在下午一点到四点活跃度最高，此后逐渐降低

### 3. 协作行为分析

提交频率：统计每个用户的提交次数，识别高活跃用户和低活跃用户。



为了统计每个用户的提交次数并识别高活跃用户和低活跃用户，我们可以使用Pandas库来处理数据。以下是相应的Python代码：

```
python

import pandas as pd

# 读取Excel文件
df = pd.read_excel('data.xlsx')

# 确保event_time列是datetime类型
df['event_time'] = pd.to_datetime(df['event_time'])

# 筛选出表示提交事件的行，例如：CreateEvent, PullRequestEvent, IssuesEvent等
submit_events = df[df['event_type'].isin(['CreateEvent',
'PullRequestEvent', 'IssuesEvent'])]

# 统计每个用户的提交次数
user_submit_counts =
```

```
python

print(high_active_users)

print("\nLow active users:")
print(low_active_users)
```

这段代码执行以下操作：

1. 读取Excel文件并将event\_time列转换为datetime类型。
2. 筛选出表示提交事件的行，例如：CreateEvent, PullRequestEvent, IssuesEvent等。
3. 统计每个用户的提交次数，并按降序排序。
4. 打印每个用户的提交次数。
5. 识别高活跃用户和低活跃用户。这里我们简单地将提交次数最多的前10%用户视为高活跃用户，提交次数最少的后10%用户视为低活跃用户。根据实际需要，这个百分比可以调整。

请注意，这个代码示例假设user\_id是唯一的用户标识符，并且event\_type列包含了事件的类型。如果你的数据集中使用了不同的列名或者有不同的事件类型，请相应地修改代码。

此外，识别高活跃和低活跃用户的方法可能需要根据实际业务需求进行调整。例如，你可以根据具体的业务目标定义不同的活跃度阈值，或者使用其他方法来识别活跃用户。

复制 再试一次 分享



运行结果：

High active users:

user\_id

158862	15762
40306929	11252
28706372	7900
1580956	5850
46537034	5519
43724913	4824
50149701	4089
9824526	4020
195327	3342
663432	3272
6270921	2775
20182680	2646
19890894	2424
2119212	2394

Low active users:

user\_id

29464607	259
5098874	258
1688225	256
15078782	252
197404	248
24123821	245
7599318	244
2230599	244
22235437	241
1365881	237

可以发现不同开发者提交频率差异巨大

#### 4. 事件类型分布

还有什么好的数据洞察角度，以我的数据

基于您提供的GitHub协作行为日志数据，可以从多个角度进行数据洞察，以下是一些可能的分析角度：

**1. 事件类型分布：**

- 统计不同类型事件（如CreateEvent、PullRequestEvent、IssuesEvent等）的数量，了解哪些类型的事件最常见。

**2. 用户活跃时间段：**

- 分析用户在一天中哪个时间段最活跃，这可以帮助理解全球用户的协作模式。

**3. 地理分布：**

- 除了国家和城市级别的分布，还可以分析州或省份级别的分布，如果数据足够详细的话。

**4. 用户参与度：**

- 分析用户在不同项目或仓库中的参与度，识别核心贡献者。

向 kimi 提问是否有其他的洞察角度

接着先从事件类型分布进行分析

那先分析这个：统计不同类型事件（如CreateEvent、PullRequestEvent、IssuesEvent等）的数量，了解哪些类型的事件最常见。

好的，我们可以通过对数据集中的事件类型进行计数来分析不同类型事件的数量。以下是进行这种分析的Python代码：

python

```
import pandas as pd

# 读取Excel文件
df = pd.read_excel('data.xlsx')

# 确保event_type列存在
if 'event_type' in df.columns:
    # 统计不同类型事件的数量
    event_type_counts = df['event_type'].value_counts()

    # 打印事件类型统计结果
    print("Event Type Counts:")
    print(event_type_counts)
```



```

event_type_counts.plot(kind='bar', title='Event Type Distribution')
plt.xlabel('Event Type')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
else:
    print("The 'event_type' column is not found in the data.")

```

这段代码执行以下操作：

1. 读取Excel文件。
2. 检查event\_type列是否存在于DataFrame中。
3. 使用value\_counts()方法统计每种事件类型的数量。
4. 打印每种事件类型的计数。
5. 使用Matplotlib绘制条形图，展示不同事件类型的分布。

请确保您的数据文件位于正确的路径上，并且列名与代码中使用的列名相匹配。如果列名不同，请相应地调整代码中的列名引用。

执行上述代码后，您将能够看到不同事件类型的计数和分布图，从而了解哪些类型的事件在数据集中最为常见。

运行结果：

## 5. 单个开发者分析

我想对单个开发者进行分析，分析其开发时间和时间类型，比如先分析id为158862的用户



要对单个开发者进行分析，我们可以筛选出特定用户ID的数据，然后分析其事件类型和事件时间。以下是进行这种分析的Python代码：

```

python

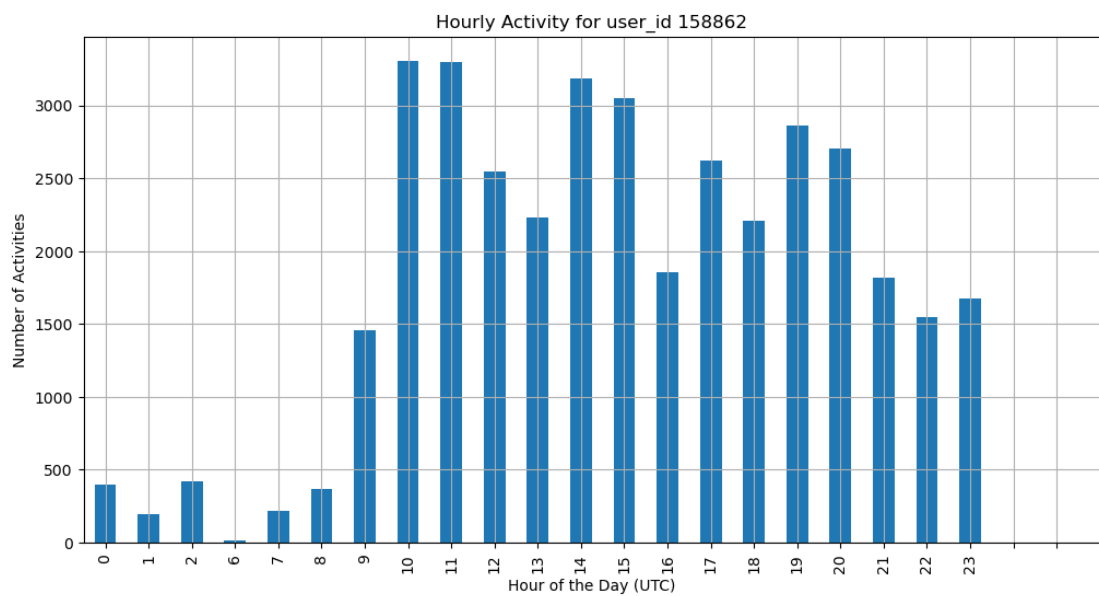
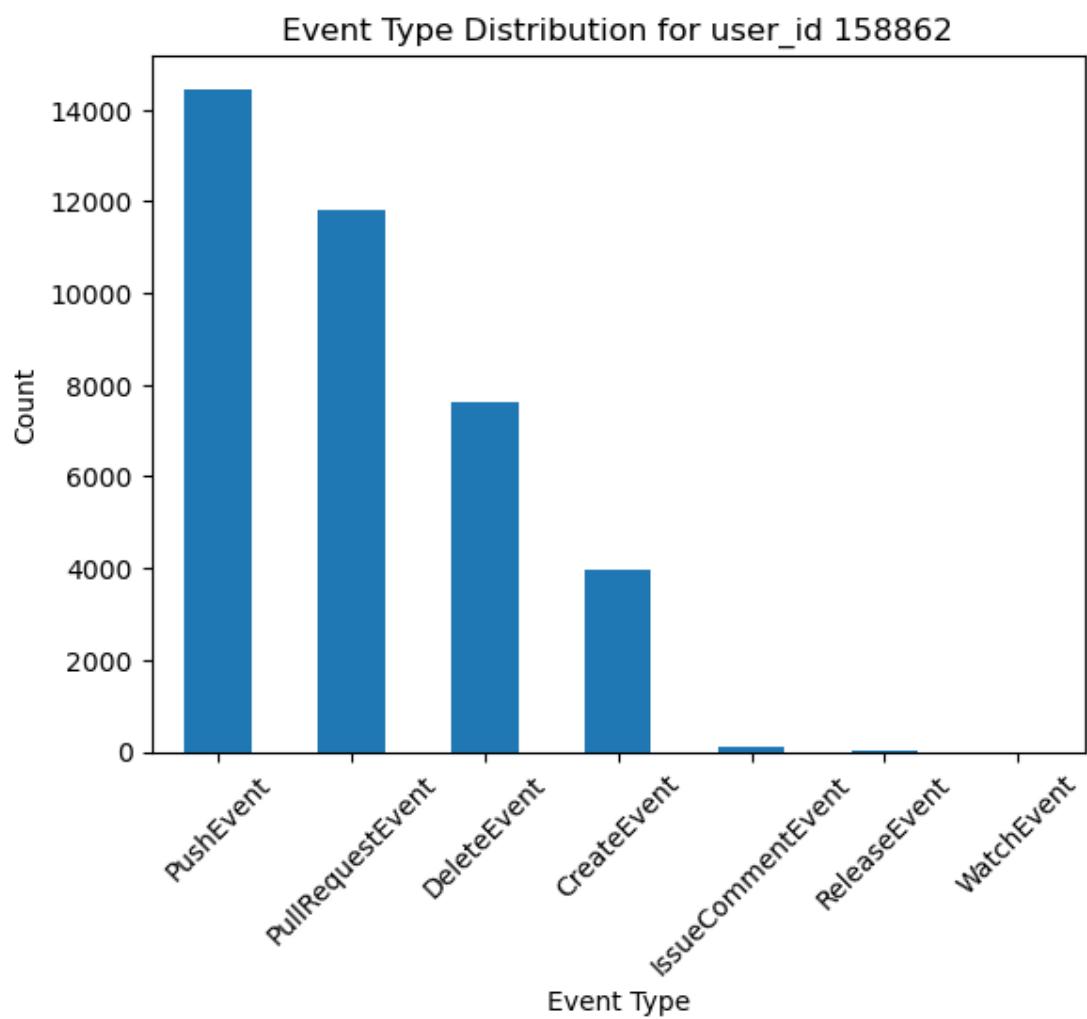
import pandas as pd
import matplotlib.pyplot as plt

# 读取Excel文件
df = pd.read_excel('data.xlsx')

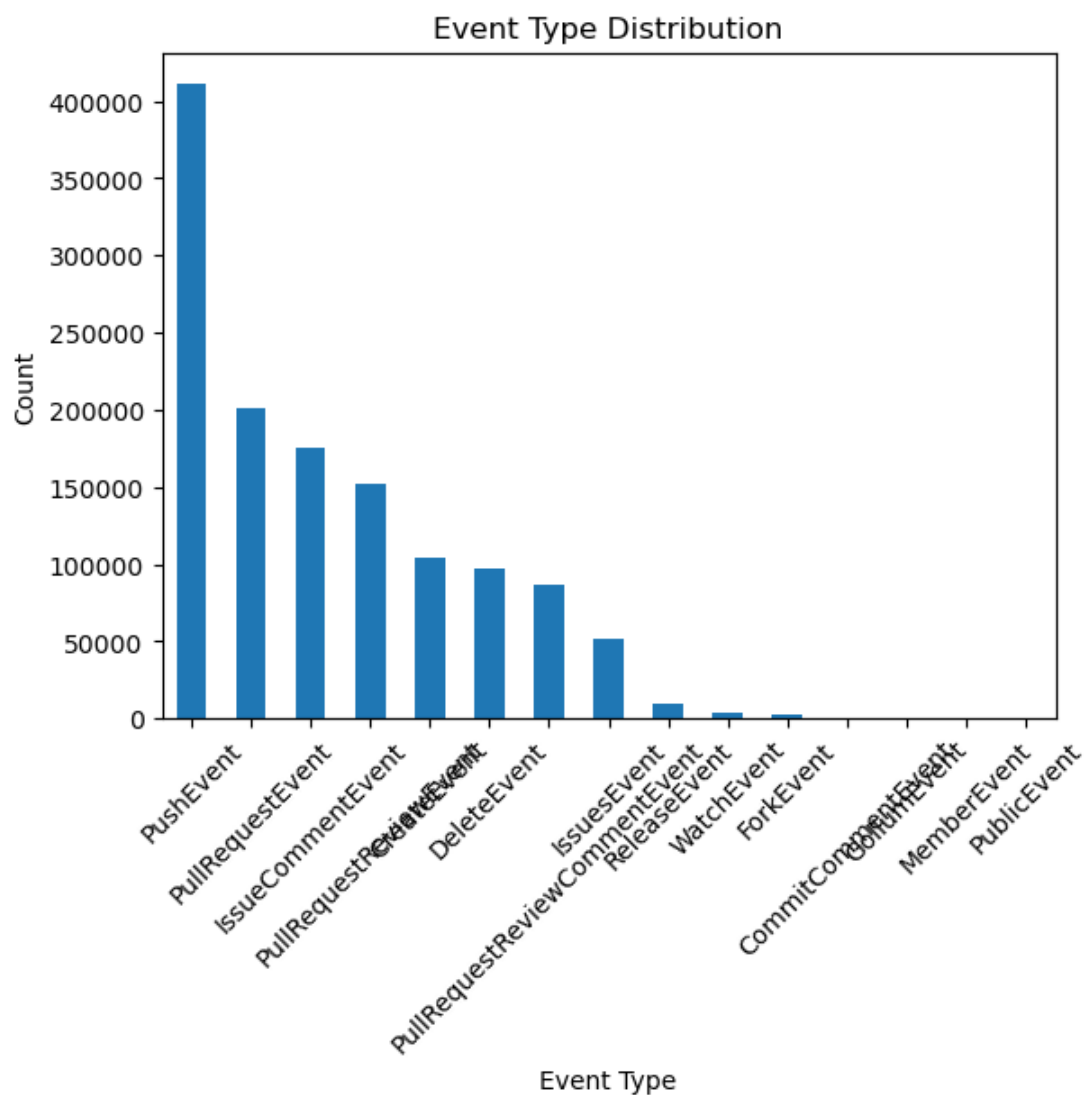
# 确保event_time列是datetime类型
df['event_time'] = pd.to_datetime(df['event_time'])

# 筛选特定用户ID的数据，这里以user_id为158862为例
user_id = 158862
user_data = df[df['user_id'] == user_id]

```



这里我分析了 id 为 158862 的用户，可以从图中得到他的提交频率和开发时间



由运行结果，提交文件 `request`，`issues` 是最多的，也是最常见的行为

## 6. 总结

上述分析的意义和方法概述如下：

### 分析的意义：

### 1. \*\*理解用户行为\*\*：

- 通过分析特定用户的事件类型和时间分布，我们可以了解用户在 **GitHub** 上的行为模式，比如他们更倾向于在一天中的哪些时段进行开发活动，以及他们参与哪些类型的事件（如创建、提交、问题讨论等）。

### 2. \*\*识别活跃度和参与度\*\*：

- 统计不同事件类型的频率可以帮助我们识别用户在项目中的活跃度和参与度。高活跃用户可能对项目的成功至关重要。

### 3. **\*\*优化协作和沟通\*\***:

- 了解用户的活动时间可以帮助团队优化协作和沟通的时间，确保在团队成员最活跃的时候进行重要的讨论和决策。

### 4. **\*\*个性化支持和资源分配\*\***:

- 根据用户的活动模式，项目管理者可以提供更个性化的支持和资源分配，比如在用户最活跃的时段提供帮助或更新信息。

### 5. **\*\*风险管理和异常检测\*\***:

- 分析用户行为还可以帮助识别潜在的风险和异常行为，比如频繁的删除操作或不寻常的活动模式。

## ### 分析的方法:

### 1. **\*\*数据筛选\*\***:

- 使用 **Pandas** 库读取和处理数据，筛选出特定用户 ID 的记录。

### 2. **\*\*事件类型统计\*\***:

- 利用 `value_counts()` 方法统计不同事件类型的频次，了解用户参与各种类型事件的活跃度。

### 3. **\*\*时间分布分析\*\***:

- 将事件时间转换为小时，并统计每个小时内的活动数量，以分析用户在一天中的活动模式。

### 4. **\*\*数据可视化\*\***:

- 使用 **Matplotlib** 库绘制条形图，直观展示事件类型分布和时间分布，使得分析结果更易于理解和交流。

### 5. **\*\*模式识别\*\***:

- 通过观察统计结果和图表，识别用户的活动模式，比如最活跃的时间段和最常参与的事件类型。

通过这些方法，我们不仅能够对单个用户的行为模式进行深入分析，还能够为项目管理和团队协作提供数据支持，从而提高项目的效率和成功率。