数据洞察报告

姓名:贾馨雨　　学号:10235501437

准备工作

数据处理，将7个csv文件合并为一个

```
#将git中的7个文件信息合并到一起
import pandas as pd
import os
folder_path = "D:\\导论data"
csv_files = [os.path.join(folder_path, f) for f in os.listdir(folder_path) if
f.endswith('.csv')]
combined_data = pd.DataFrame()
for csv_file in csv_files:
    df = pd.read_csv(csv_file)
    combined_data = pd.concat([combined_data, df], ignore_index=True)
# 定义合并后文件在原文件夹内的保存路径
save_path = os.path.join(folder_path, "合并后的data.csv")
combined_data.to_csv(save_path, index=False)
```

1. 国家和地区分布

分析结果

- 用户主要集中在少数几个国家，其中美国用户数量占据显著比例。
- 英国和印度是排名第二和第三的国家，但数量上与美国有明显差距。

洞察

- 数据表明，用户分布呈现显著的地域性差异，可能与地区的技术发展水平和市场需求相关。
- 在用户密集的国家，可以进一步挖掘区域特点，为后续推广和资源分配提供依据。

可视化说明

- 以 country 字段为核心依据，运用高效的数据分组算法对用户数据进行分组操作，然后借助统计分析工具精确统计每个国家的用户数量使用柱状图展示了各国用户数量，按用户数量从高到低排序。

  (柱状图因数据量差异过大，导致数量过少的国家看不明显，所以采用了取对数的方法)
- 图表直观地显示了各国用户分布的不均衡性，重点国家尤为突出。同时，结合先进的地理信息可视化库，将统计结果以直观的世界地图形式呈现，清晰展示用户在全球范围内的分布态势。

```
# 绘制柱状图
ax = major_countries.plot(kind='bar', figsize=(12, 8),color='lightblue')
ax.set_title('Developer Distribution by Country')
ax.set_xlabel('Country')
ax.set_ylabel('Number of Developers')
# 数据量级差异太大，取对数
if (major_countries.max() / major_countries.min()) > 100:
```

```
    ax.set_yscale('log')
# 旋转x轴标签角度，这里旋转45度，水平对齐方式设为右对齐（ha='right'），让标签更清晰
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
# 绘制国家分布的地图
fig = px.choropleth(
    country_dist,
    locations="country",
    locationmode="country names",
    color="user_count",
    color_continuous_scale=px.colors.sequential.Plasma,
    title="Global User Distribution by Country"
)
fig.update_layout(
    geo=dict(
        showframe=False,
        showcoastlines=True,
        projection_type="equirectangular"
    )
)

fig.show()
```

2. 城市级别分布

分析结果

运用字符串处理与数据提取技术，从 location 字段中精准提取城市信息。然后结合人口 数据计算开发者密度，通过数据排序与筛选，识别出开发者密度显著高于其他地区的技术热点城市。

- 用户分布在多个城市，但集中在少数技术热点区域。
- 排名前三的城市分别为旧金山、纽约和伦敦，这些城市均为全球重要的技术中心。

洞察

- 技术热点区域用户的集中可能反映了城市在技术产业中的领先地位。
- 对于用户数量较多的城市，可以针对性地设计本地化服务和推广策略。

可视化说明

- 图表显示了用户数量最多的前20个城市，并解决了城市标签重合问题。
- 各城市的用户数量以对齐的柱状图展示，使得视觉呈现更清晰。

```
# 2. 城市级别分布
city_dist = data['location'].value_counts().head(20)  # 显示前20个城市
plt.figure(figsize=(12, 6))
plt.bar(range(len(city_dist)), city_dist.values, color='lightblue',
align='center')
plt.xticks(range(len(city_dist)), city_dist.index, rotation=45, ha='right')
plt.title('Top 20 Cities by User Count')
plt.xlabel('City')
plt.ylabel('Number of Users')
plt.tight_layout()
plt.show()
```

3. 时区分布

分析结果

- 用户活动时间集中在白天工作时段，特别是上午10点到下午3点，因为题目要求为时区，所以就要自己创建一个字典，把每个国家和对应城市的时区写出来，便于查找，绘制图表时是每个国家的不同时间。UTC - 8（美国西海岸） 用户活跃于北京时间凌晨（当地时间上午 9 点 - 下午 3 点）； UTC + 8（东亚） 用户活跃于北京时间早上 9 点 - 下午 6 点； UTC + 5:30（印度） 用户活跃于北京时间中午至晚上。

洞察

- 数据显示用户活动时间与典型的工作日时间高度吻合，这为协作工具的优化提供了参考。
- 不同区域用户的协作可能受到时区差异的限制，建议进一步探索跨时区协作的优化方案。

可视化说明

- 活动分布以小时为单位绘制了折线图，直观展示了全天活动量的变化趋势。
- 折线图中的高峰和低谷明确反映了用户的活跃时间段。

```
valid_data['timezone'] = valid_data['country'].map(country_timezone_dict)
# 选取的十个国家对应的时区列表
timezones = list(country_timezone_dict.values())
# 循环处理每个时区的数据，展示其不同小时的事件分布情况
for timezone in timezones:
    timezone_data = valid_data[valid_data['timezone'] == timezone]
    hourly_dist =
timezone_data['event_time'].dt.hour.value_counts().sort_index()
    plt.figure(figsize=(10, 5))
    plt.plot(hourly_dist.index, hourly_dist.values, marker='o',
color='orange')
    plt.title(f'Activity Distribution by Hour in {timezone}')
    plt.xlabel('Hour of Day')
    plt.ylabel('Number of Events')
    plt.grid(True)
    plt.tight_layout()
    plt.show()
```

4. 提交频率

分析结果

依据每个 user_id，运用数据统计函数精确统计其所有提交次数。然后，根据 预先设定的活跃度分类标准：高活跃用户（提交次数 > 100）、中等活跃用户（提 交次数在 20 - 100 之间）、低活跃用户（提交次数 < 20），对用户进行分类划分

- 提交频率分布显示出显著的差异性，少数高活跃用户的提交量远超其他用户。
- 为了避免极值影响分析效果，对数据进行过滤，仅保留95%分位数以内的数据。

洞察

- 高活跃用户的行为特征值得深入分析，可能是推动平台发展的关键力量。
- 低活跃用户的参与动机和障碍需要进一步探讨，以提高整体参与率。

可视化说明

- 提交频率分布直方图在过滤极值后清晰展示了大部分用户的提交行为。
- 图表有效减少了极端值对整体趋势的干扰，便于分析主流用户行为。

```
# 4. 提交频率
user_activity = data.groupby('user_id')
['event_action'].count().sort_values(ascending=False)
user_activity = user_activity.replace([float('inf'), -float('inf')],
pd.NA).dropna()
plt.figure(figsize=(12, 6))
sns.histplot(user_activity[user_activity < user_activity.quantile(0.95)],
bins=30, kde=False, color='lightblue')
plt.title('Submission Frequency Distribution (Filtered)')
plt.xlabel('Number of Submissions')
plt.ylabel('Number of Users')
plt.tight_layout()
plt.show()
```

5. 其他有趣的洞察

(1) 用户影响力分布

分析结果

- 用户总影响力呈现出典型的长尾分布，少数用户的影响力显著高于大多数用户。
- 使用核密度估计（KDE）方法更细致地展示了影响力的分布特征。

洞察

- 高影响力用户可能是平台的核心推动者，可以针对这些用户设计专属的激励机制。
- 普通用户的影响力分布提供了平台公平性和用户增长空间的视角。

可视化说明

- 核密度估计曲线直观显示了用户影响力的集中与分散情况。
- 图表细致呈现了数据分布的整体趋势及局部特性。

```
# (1) 用户影响力分布
plt.figure(figsize=(10, 6))
data['total_influence'] = data['total_influence'].replace([float('inf'), -
float('inf')], pd.NA).dropna()
sns.histplot(data['total_influence'], bins=30, kde=True, color='orange')
plt.title('User Influence Distribution')
plt.xlabel('Total Influence')
plt.ylabel('Frequency')
plt.tight_layout()
plt.show()
```

(2) 事件类型分布

分析结果

- 平台事件类型主要集中在少数几类，`CreateEvent` 占比最大。
- 数据存在长尾分布现象，小众事件类型的数量显著较少。

洞察

- 主流事件类型的集中可能反映了用户的主要使用需求，可重点优化相关功能。
- 小众事件类型的分析有助于发现潜在的增长点。

可视化说明

- 对事件类型分布数据进行了平方根平滑处理，使得小众事件的分布更清晰。
- 平滑后的柱状图展示了各类事件的相对频率，突出了细节。

```
# (2) 事件类型分布
event_type_dist = data['event_type'].value_counts()
plt.figure(figsize=(10, 6))
sqrt_event_type_dist = event_type_dist.apply(lambda x: max(x, 1)**0.5)  # 平
滑处理
sqrt_event_type_dist.plot(kind='bar', color='lightblue')
plt.title('Event Type Distribution (Square Root Scale)')
plt.xlabel('Event Type')
plt.ylabel('Transformed Frequency')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

总结

综合洞察

- 用户的地理分布和行为模式揭示了平台在不同区域和人群中的影响力。
- 活跃用户和高影响力用户是平台发展的重要资源，应优先关注。