# 一、数据探索

## 1.1 查看数据行

print(f"行数: {len(data)}")

行数: 464243

## 1.2 查看数据列

Index(['id', 'content\_id', 'page\_path', 'username', 'userid', 'sessionid',

'ip', 'country', 'area', 'browser\_type', 'browser\_version',

'platform\_type', 'platform\_series', 'platform\_version', 'date\_time',

'mobile\_type', 'agent', 'uniqueVisitorId', 'key\_word', 'source',

'operate'],

dtype='object')

|  |  |  |
| --- | --- | --- |
| 字段名 | 宇段解释 | 可能用途 |
| id | 日志记录的唯一标识符 | 数据记录的索引或主键 |
| content\_id | 内容的唯一标识符 | 用于分析具体内容的访问量或受欢迎程 度 |
| page\_path | 页面路径 (URL 路径) | 用于识别具体的访问页面 |
| username | 用户名 (如果有登录系统) | 分析特定用户的行为 |
| userid | 用户的唯一标识符 | 区分具体用户，便于用户行为分析 |
| sessionid | 会话的唯一标识符 | 跟踪同一会话中的行为 |
| ip | 用户访问的IP地址 | 用于地理位置分析或安全审计 |
| country | 用户所在的国家 | 地理位置统计与流量来源分析 |
| area | 用户所在的区域 (省、市等) | 更精细的地理位置分析 |
| browser\_type | 用户使用的浏览器类型 | 浏览器使用趋势分析 |
| browser\_version | 浏览器版本 | 分析不同版本的兼容性与使用率 |
| platform\_type | 用户设备平台类型（如移动设备或桌面设 备) | 区分不同平台的访问趋势 |
| platform\_series | 用户设备的平台系列(如Android、iOS) | 设备分布统计 |
| platform\_version | 用户设备的平台版本 | 平台版本使用分布 |
| date\_time | 访问的日期和时间 | 构建时间序列，分析访问趋势 |
| mobile\_type | 移动设备型号(如iPhone、三星等) | 分析移动设备分布 |
| agent | 用户代理信息 | 判断用户设备和浏览器的详细信息 |
| uniqueVisitorId | 唯一访问者标识符 (UV) | 统计独立访客数量 |
| key\_word | 搜索关键词 | 分析用户的搜索行为 |
| source | 流量来源 (如搜索引擎、社交媒体) | 流量来源分析 |
| operate | 用户的操作类型 | 分析用户行为模式 |

## 1.3 数据检查

# 查看每一列是否有空值

no\_null\_columns = data.columns[data.notnull().all()]

print(f"没有空值的列: {no\_null\_columns.tolist()}")

没有空值的列: ['id', 'page\_path', 'sessionid', 'ip', 'date\_time']

## 1.3 数据提取与去重

根据任务要求：利用Arima对时间序列进行预测。只需要构造每日访问量的时间序列，即可满足模型要求，因此我对原始数据进行提取，去掉无关的数据。根据字段的解释，我取id、Ip、date\_time构成新的数据集，同时去重。

# 数据筛选

seriesdata = data[['id', 'ip', 'date\_time']]

# 去重

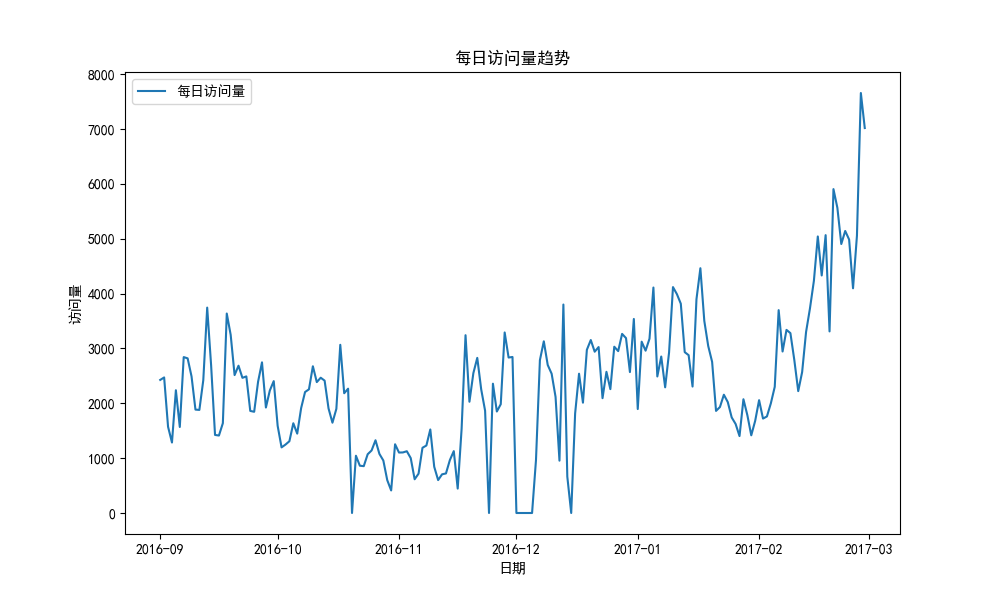
seiresdata = seriesdata.drop\_duplicates(subset=['ip', 'date\_time'])

print(seriesdata.shape)

# 二、时序图、自相关图、偏自相关图与平稳性检验

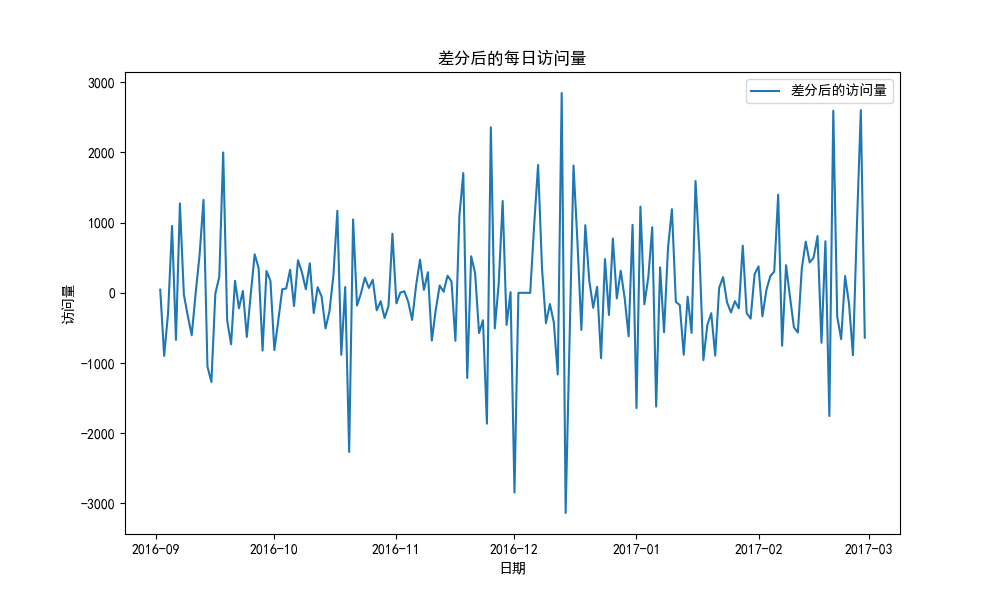
## 2.1 时序图

# 绘制原始时间序列  
plt.figure(figsize=(10, 6))  
plt.plot(daily\_visits['visits'], label='每日访问量')  
plt.title('每日访问量趋势')  
plt.xlabel('日期')  
plt.ylabel('访问量')  
plt.legend()  
plt.savefig('./img/每日访问量趋势.png')   
plt.show()



## 2.2 平稳性检验（ADF检验）

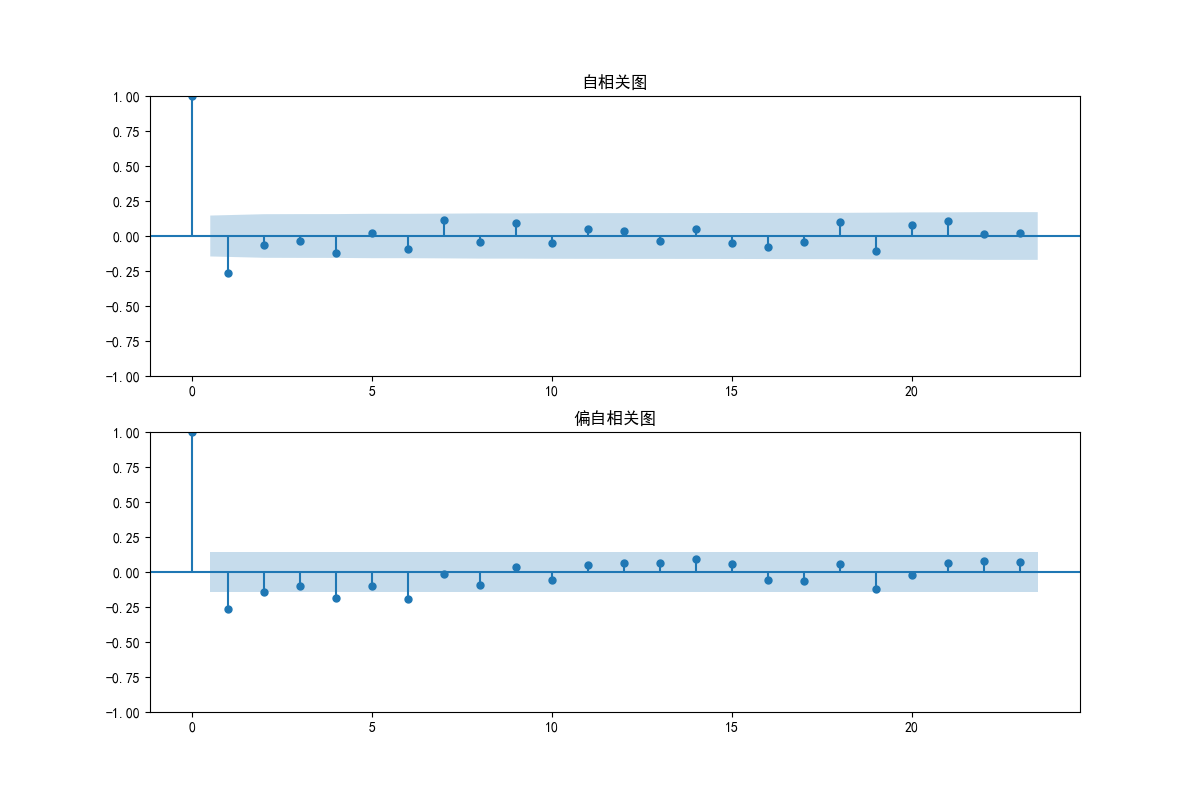
# 检验数据平稳性并确定差分次数（d）  
def test\_stationarity(series, max\_diff=5):  
 d = 0  
 while d < max\_diff:  
 adf\_test = adfuller(series.dropna())  
 p\_value = adf\_test[1]  
 print(f"ADF检验 p-value: {p\_value}")  
 if p\_value > 0.05:  
 print(f"数据非平稳，进行第 {d + 1} 次差分处理")  
 series = series.diff().dropna()  
 d += 1  
 else:  
 print("数据已平稳")  
 return series, d  
 raise ValueError(f"数据在最大差分次数 {max\_diff} 次后仍未平稳，请检查数据或采用其他方法处理。")



通过ADF检验，我们发现原始时间序列非平稳，因此采用差分处理来消除趋势性和非平稳性。在每次差分后对序列重新进行ADF检验，直到p-value小于0.05，最终确定差分次数d=1。差分后的序列显示出波动围绕0水平线，振幅较稳定，证明其已达到平稳状态，为后续ARIMA模型的拟合提供了良好的基础。

## 2.3 自相关图与偏相关图

# 自相关图与偏自相关图  
fig, ax = plt.subplots(2, 1, figsize=(12, 8))  
plot\_acf(daily\_visits['visits\_diff'].dropna(), ax=ax[0], title='自相关图')  
plot\_pacf(daily\_visits['visits\_diff'].dropna(), ax=ax[1], title='偏自相关图')  
plt.savefig('./img/自相关与偏自相关图.png') # 中文命名  
plt.show()



通过对差分后的时间序列进行自相关函数（ACF）和偏自相关函数（PACF）分析，我们绘制了自相关图和偏自相关图（如图所示）。自相关图显示，在滞后1后的自相关迅速衰减并趋于随机，表明滑动平均（MA）项可能的阶数q=1。偏自相关图显示，在滞后1后的偏自相关系数显著下降并趋于零，表明自回归（AR）项可能的阶数p=1。基于这些结果，我们初步判断ARIMA模型的参数组合可能为(p,d,q)=(1,1,1)。这种图像法提供了直观的统计依据，为后续模型的参数优化奠定了基础。

# 三、白噪声检验（Ljung-Box检验）

# 白噪声检验  
lb\_test = acorr\_ljungbox(daily\_visits['visits\_diff'].dropna(), lags=[10])  
print(f"Ljung-Box检验 p-value: {lb\_test['lb\_pvalue'].values[0]}")



在差分处理后的时间序列上，我们对其进行了 Ljung-Box 检验，以评估其是否为白噪声序列。Ljung-Box检验的p-value为**0.011**，低于显著性水平0.05，表明差分后的序列仍存在显著的自相关性，不是纯白噪声。这说明时间序列中可能仍然包含一些结构性信息，如趋势或周期性，尚未被完全移除。为了进一步改善模型拟合效果，需要进一步调整ARIMA模型的参数（如p、q）

# 四、使用 BIC 准则定阶，辅助人工使用图像法定阶

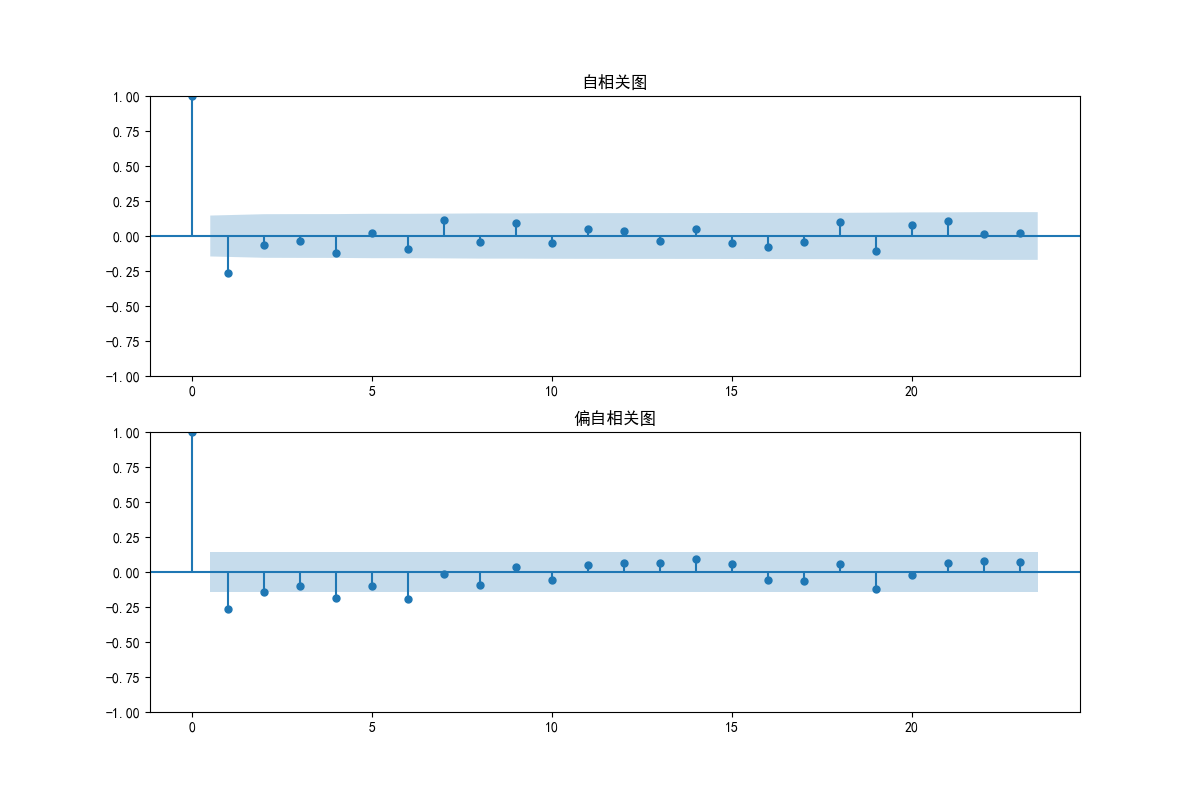
## 4.1 循环尝试不同阶数

遍历 ARIMA 模型的参数组合 (p, q)，计算每种情况的 BIC 值

# 使用BIC准则定阶  
bic\_values = []  
max\_p = 5 # 限制最大阶数  
max\_q = 5  
for p in range(max\_p):  
 for q in range(max\_q):  
 try:  
 model = ARIMA(daily\_visits['visits'], order=(p, 1, q)).fit()  
 bic\_values.append((p, q, model.bic))  
 except Exception as e:  
 print(f"模型阶数 p={p}, q={q} 拟合失败: {e}")  
 continue  
best\_p, best\_q = min(bic\_values, key=lambda x: x[2])[:2]  
print(f"最佳阶数 p, q: {best\_p}, {best\_q}")



## 4.2 图像定阶法



通过绘制差分后时间序列的自相关图（ACF）和偏自相关图（PACF），我们可以观察其特性。从自相关图（上图）可以看出，在滞后1处有显著的自相关系数，之后的自相关性迅速减弱并趋于随机，表明滑动平均（MA）部分的阶数可能为q=1。从偏自相关图（下图）来看，滞后1处的偏自相关系数显著，其余滞后值逐渐趋于零，表明自回归（AR）部分的阶数可能为p=1。这些特性表明，差分后的序列适合拟合ARIMA(1,1,1)模型，与BIC准则得出来的一致。

# 五、预测未来7天

