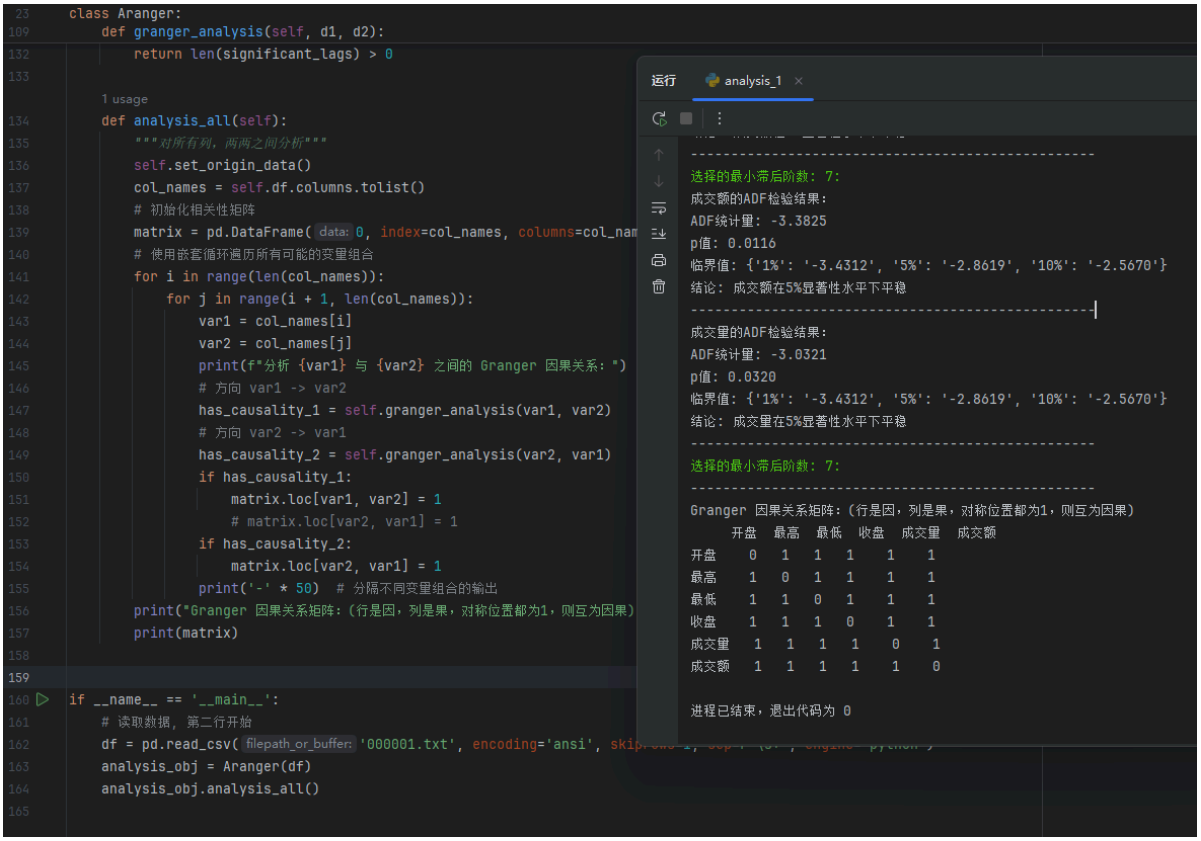
一、Granger因果关系分析

Granger 因果关系分析是一种统计方法，用于**确定一个时间序列是否可以预测另一个时间序列。**

- 基本思想是，如果一个时间序列 X 中的过去值能够提供 Y 未来值的显著信息，那么可以说 X Granger 引起了 Y。
- 重要的是，这种因果关系并不意味着真实的因果关系，而是统计学意义上的预测能力。



二、方向性条件概率

方向性条件概率（Directional Conditional Probability）通常是指在给定方向上的某个条件下事件发生的概率。

它涉及概率分布的方向性或时间顺序，这在时间序列分析、因果关系分析或贝叶斯网络中可能会遇到。

- 在**时间序列分析**中，方向性条件概率考虑时间顺序。例如，给定过去的观测值，预测未来观测值的条件概率：

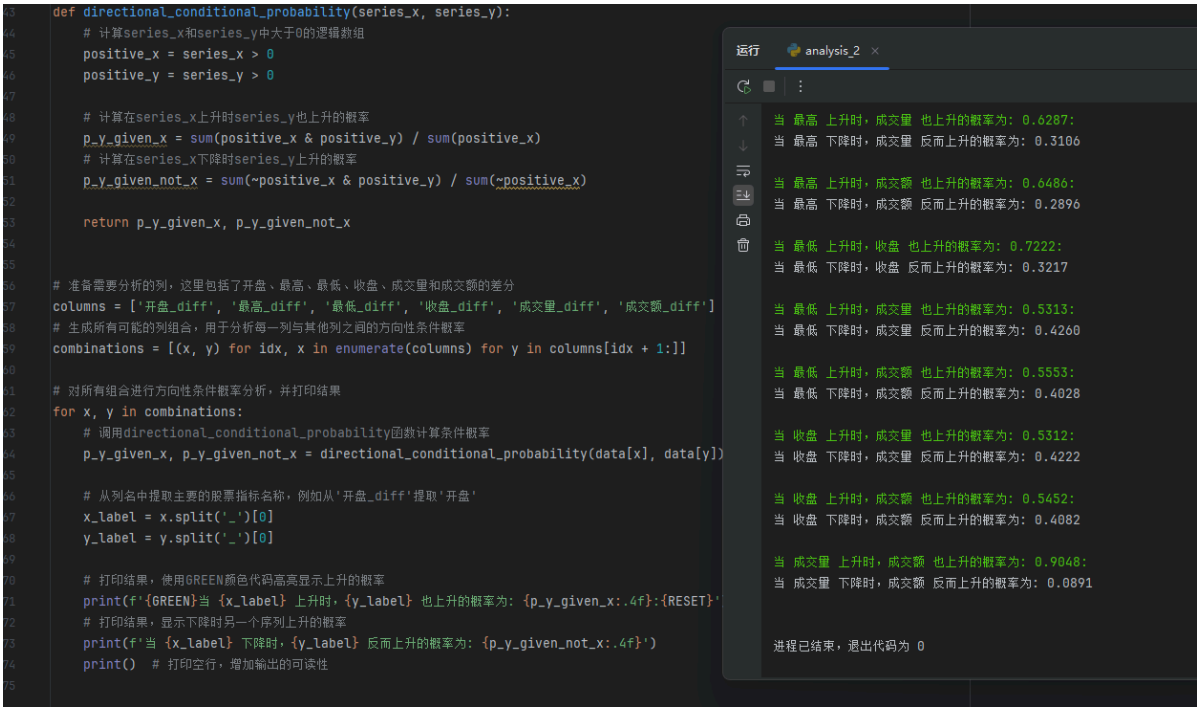
$$P(Y_{t+1}|Y_t, Y_{t-1}, \ldots, Y_{t-k})$$

条件概率是基于之前的 k个观测值，预测时间序列在 t+1时刻的值。

- 在**因果关系分析**中，方向性条件概率用于表示因果关系的方向。例如，假设我们有两个变量 X 和 Y，我们可能会计算：

$$P(Y|X)$$

表示在 X发生的条件下 Y发生的概率。如果 P(Y|X)和 P(X|Y)显著不同，这可能暗示一个方向性的因果关系。



三、交互延迟互信息

交互延迟互信息（Interaction Delay Mutual Information，**IDMI**）是一种用于衡量两个时间序列之间的相互作用和延迟关系的统计方法。

它结合了**互信息**和**时间延迟分析**，用于识别和量化在时间序列数据中存在的非线性依赖关系。

- 互信息（Mutual Information）是用来衡量两个随机变量之间的非线性依赖关系的量度。它**表示两个变量之间共享的信息量。互信息越大，说明两个变量之间的关系越强。**互信息可以计算为：

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} = P(X) + P(Y) - P(X,Y)$$

其中：

p(x, y) 是 X 和 Y 的**联合概率分布。**

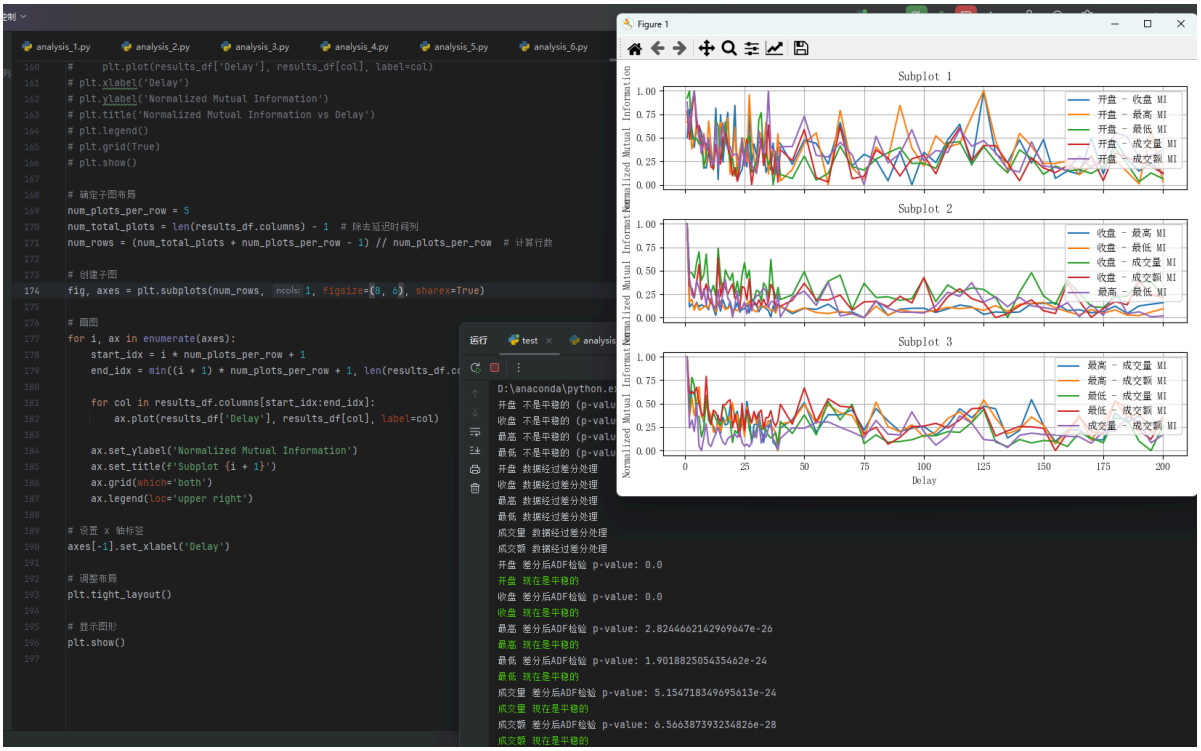
p(x) 和 p(y) 是 X 和 Y 的**边际概率分布。**

- 交互延迟互信息通过将 **时间序列的延迟（滞后）**与互信息结合，用于衡量一个时间序列的过去状态对另一个时间序列当前状态的影响。其步骤通常包括：

计算互信息：计算两个时间序列在不同延迟下的互信息，以量化它们的相互依赖。

引入延迟：考虑时间序列中的滞后或延迟。例如，对于时间序列 X 和 Y，计算X(t-k) 对 Y(t) 的互信息，其中 k 是滞后期。

计算交互延迟互信息：在不同的**延迟期**上计算互信息，得到关于时间序列相互作用的延迟信息。这可以揭示出时间序列之间的复杂非线性关系和交互模式。



四、时间滞后相干性分析

时间滞后相干性分析（Time-Lagged Coherence Analysis）是一种用于研究时间序列之间在不同时间滞后下的线性关系和相干性的技术。

它特别适用于探究两个时间序列之间在不同时间延迟上的相互依赖关系。

1. **相干性（Coherence）**：
   相干性是衡量两个时间序列之间**在频域上的线性相关性**。相干性值范围从0到1，0表示无相干性，**1表示完全的相干性**。高相干性表明两个时间序列在相同频率下有强的线性关系。
2. **时间滞后（Time Lag）**：
   时间滞后是指在一个时间序列的观测值与另一个时间序列的观测值之间的时间差。在分析中，我们可能会考虑不同的滞后期，以了解两个时间序列在这些滞后期下的关系如何变化。

分析过程

1. **计算相干性**:
   相干性分析通常在频域中进行。首先将时间序列转换到频域，这可以通过**傅里叶变换**实现。然后，计算两个时间序列的**功率谱密度**（Power Spectral Density, PSD）以及它们的**互谱**（Cross-Spectrum）。

   相干性可以通过以下公式计算:

   $$C_{XY}(f) = \frac{|\mathrm{S}_{XY}(f)|^2}{\mathrm{S}_{XX}(f) \cdot \mathrm{S}_{YY}(f)}$$

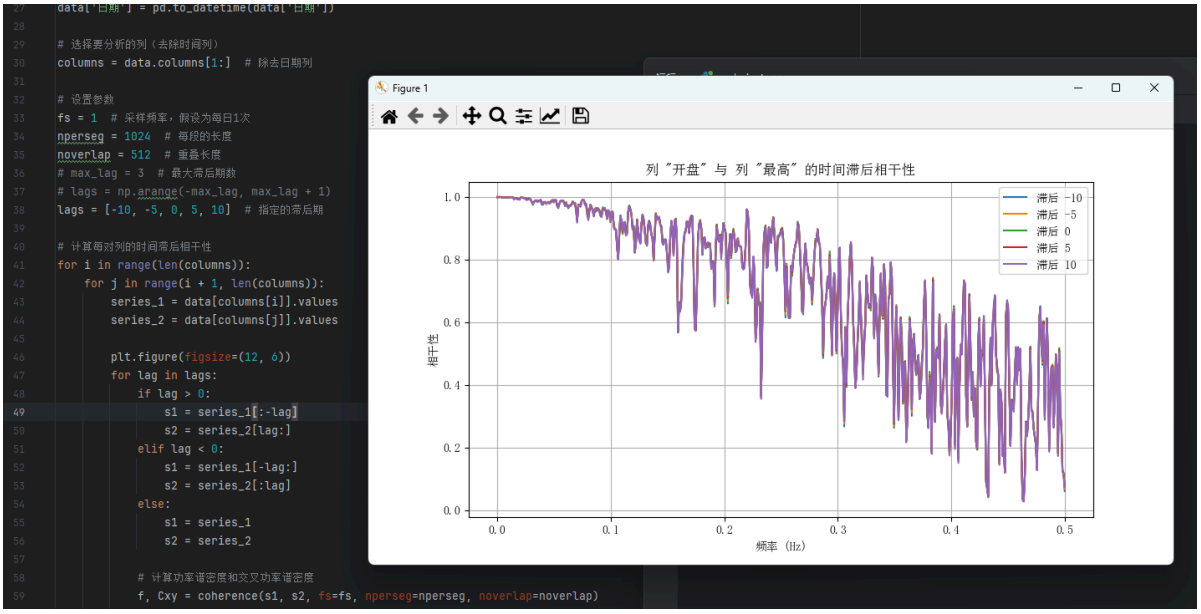
   其中:

   $\mathrm{S}_{XY}(f)$是 (X) 和 (Y) 的互谱。

   $\mathrm{S}_{XX}(f))$ 和$\mathrm{S}_{YY}(f)$ 是 (X) 和 (Y) 的功率谱密度。

2. **考虑不同时间滞后**：
   在时间滞后相干性分析中，分析不同时间滞后下的相干性可以揭示时间序列之间的复杂依赖关系。这通常涉及到将时间序列在不同滞后期进行处理，并计算相应的相干性。

3. **分析结果**：
   分析结果通常以图形方式展示，其中相干性随时间滞后变化的图表可以揭示不同滞后期下的相干性模式。高相干性值表示两个时间序列在特定频率和滞后下有强的线性关系，而低相干性值则表示较弱的关系。



五、向量自回归条件异方差

向量自回归条件异方差模型（Vector Autoregressive Conditional Heteroskedasticity，VARCH）是一种扩展的模型，用于处理多变量时间序列数据中存在的条件异方差性。

它结合了向量自回归（VAR）模型和条件异方差（ARCH/GARCH）模型的特点，适用于分析时间序列中多个变量的相互影响及其波动性。

向量自回归（VAR）模型:

向量自回归（VAR）模型是一种用于**分析多个时间序列变量之间的线性动态关系**的模型。

在 VAR 模型中，**每个时间序列的当前值都是其过去值的线性函数，同时也可能受到其他时间序列过去值的影响。**V

条件异方差（ARCH/GARCH）模型:

条件异方差（ARCH）模型用于处理时间序列数据中的**波动性聚集现象。**于1986年扩展为 GARCH（Generalized ARCH）模型。

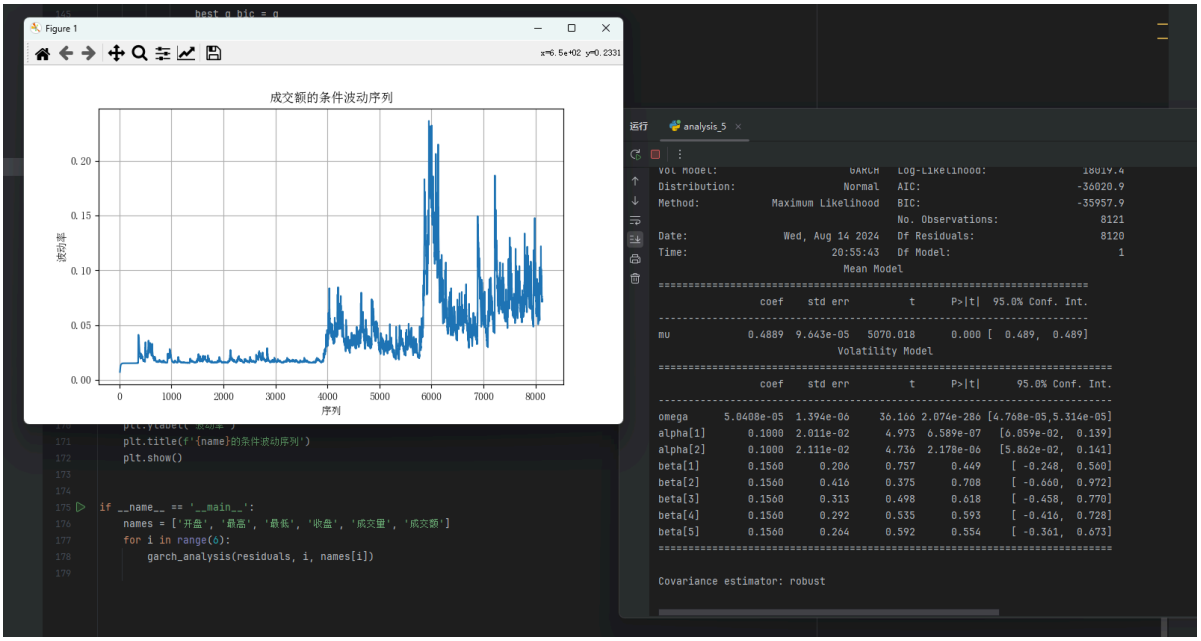
- **ARCH 模型**：假设时间序列的**条件方差依赖于过去误差的平方。**
- **GARCH 模型**：在 ARCH 模型的基础上，引入了**过去方差的滞后项**，使得条件方差依赖于**过去误差和过去方差。**

向量自回归条件异方差（VARCH）模型:

VARCH 模型结合了 VAR 模型和 ARCH/GARCH 模型，处理多变量时间序列中的**波动性和依赖关系。**VARCH 模型的主要特点包括：

1. **向量自回归**：模型的基础是 VAR 模型，用于描述时间序列变量之间的线性动态关系。
2. **条件异方差**：在 VAR 模型的基础上，引入条件异方差模型（如 GARCH），用于建模误差项的条件方差。

确定滞后阶数常用的方法包括 **信息准则**（AIC、BIC、HQIC、FPE）和**交叉相关函数**（CCF）。



六、递归量化分析

第1步：相空间重构

1. **确定嵌入维度**：
    - **方法**：使用FNN（False Nearest Neighbors）方法来确定嵌入维度。
    - **步骤**：
        1. 计算不同嵌入维度下的假最近邻数量。
        2. 选择假最近邻数量最小的维度作为最佳嵌入维度。
2. **确定延迟时间**：
    - **方法**：使用最小二乘法（或自相关函数）来确定延迟时间。
    - **步骤**：
        1. 计算自相关函数或互相关函数。
        2. 找到自相关函数首次显著下降的位置，作为延迟时间。

第2步：距离矩阵确定

1. **计算距离矩阵**：
    - **互信息距离**：
        1. 计算每对点的互信息量。

2. 使用互信息量作为距离度量。

- **曼哈顿距离**：
  1. 对于每对点，计算曼哈顿距离。
  2. 曼哈顿距离 = ∑|x_i - y_i|。
- **余弦相似度**：
  1. 计算每对点的余弦相似度。
  2. 余弦相似度 = (x·y) / (||x|| * ||y||)。
- **同时运用**：
  1. 将上述三种距离计算结果结合，形成一个综合距离矩阵。

2. **阈值确定**：
   - **K-最近邻阈值法**：
     1. 计算不同K值的K-最近邻距离。
     2. 选择合适的K值，使得距离矩阵能有效反映时间序列中的动态结构。
   - **确定K值的方法**：
     - **MSE（均方误差）**：
       1. 对不同K值，计算训练集和测试集上的均方误差。
       2. 选择MSE最小的K值。
     - **邻近误差**：
       1. 计算每个K值下的邻近误差。
       2. 选择邻近误差最小的K值。
     - **准确率-距离平衡**：
       1. 计算不同K值下的准确率和距离平衡。
       2. 选择准确率和距离平衡最优的K值。
     - **AIC和BIC（赤池信息准则和贝叶斯信息准则）**：
       1. 计算不同K值下的AIC和BIC值。
       2. 选择AIC和BIC值最小的K值。

后续步骤：RQA分析

1. **计算递归图**：
   - 使用前述的距离矩阵，生成递归图（recurrence plot）。
2. **量化指标**：
   - 计算以下RQA量化指标：
     - **Recurrence Rate (RR)**：递归图中点的比例。
     - **Determinism (DET)**：测量递归图中的确定性。
     - **Average Diagonal Line Length (L)**：测量递归图中的平均对角线长度。
     - **Entropy (ENT)**：递归图的熵值。
3. **分析结果**：
   - 通过量化指标分析时间序列的动态相关性。
   - 比较不同时间序列的RQA指标，以检测它们之间的相关性。

```python
def start_analysis(filename, select_col):
    df = df[-365:]

    result = adfuller(df[select_col].values)
    print(f'ADF 统计量: {result[0]:.4f}')
    print(f'p 值: {result[1]:.4f}')
    if result[1] > 0.05:
        print(f"{RED}序列不平稳,将进行差分。{RESET}")
        df[select_col] = df[select_col].diff().dropna()
    else:
        print(f'{GREEN}序列平稳{RESET}')

    df = df.dropna(subset=[select_col])

    # 数据标准化处理
    scaler = MinMaxScaler()
    # 对df进行拟合和转换
    df_scaled = scaler.fit_transform(df)
    df = pd.DataFrame(df_scaled, columns=df.columns)

    # 最佳延迟时间
    delay_time = delay_time_selection_mi(df[select_col].values)
    # 获取最佳嵌入维度
    embedding_dim = fnn_method(df[select_col].values, max_delay=delay_time)
    # 执行RQA分析
    rqa_analysis(df[select_col].values, embedding_dim, delay_time)


if __name__ == '__main__':
    file = '000001.txt'
    col = '成交量'

    print(f"{GREEN}=========分析{col}========{RESET}")
    start_analysis(file, col)
    print(f"{GREEN}=========================={RESET}")
```

```
D:\anaconda\python.exe E:\Analysis\序列相关性检测\analysis_6.py
=========分析成交量========
ADF 统计量: -3.9457
p 值: 0.0017
序列平稳
最佳延迟时间为: 4
最佳嵌入维度为: 4
最佳K值: 10
阈值为:0.2469
递归度RR: 0.2476
平均对角线长度: 3.6194
最大对角线长度: 350.0000
平均垂直线长度: 4.9929
最大垂直线长度: 332.0000
递归点百分比: 24.76%
```