专题二：统计类模型

目录

I. 线性回归部分

1 相关分析

1.1 Pearson 相关系数

Pearson 相关系数（Pearson Correlation Coefficient）用于衡量两个**连续变量**之间的**线性关系**，取值范围为[-1, 1]。其公式为：

$$r=\frac{\sum(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum(x_i-\bar{x})^2\sum(y_i-\bar{y})^2}}$$

①Pearson 相关系数是最传统的相关系数，只能适用于线性关系；

②Pearson 相关系数适用于定量数据，不适用于定序数据；

③Pearson 相关系数对正态分布较为敏感，对异常值敏感。

1.2 Spearman 相关系数

Spearman 相关系数（Spearman's Rank Correlation Coefficient）用于衡量两个变量的**单调关系（不一定是线性关系）**。它基于变量的**秩（排序）**而非原始值，公式为：

$$\rho=1-\frac{6\sum d_i^2}{n(n^2-1)}$$

其中：d_i 是两个变量秩的差值，n 是样本数量。

①Spearman 相关系数是更为创新的相关系数；

②Spearman 相关系数适用于定类数据，对数据的分布不敏感，因此结果更稳健。

1.3 相关系数的创新与注意事项

（1）注意对相关系数的解释，而不只是计算相关系数的结果

相关性解释	相关系数范围	备注
完全相关	r=1 或 r=-1	完全正相关或完全负相关
强相关	0.7<|r|<1	强正相关或强负相关
中等相关	0.3<|r|<0.7	中等正相关或中等负相关
弱相关	0<|r|<0.3	弱正相关或弱负相关
无相关	r=0	没有线性相关性

（2）Pearson 相关系数更准确，Spearman 相关系数更稳健，但在数模国赛的比赛中往往将这两种方法结合起来适用，并对比分析说明结果的合理性

（3）相关系数这一部分可以绘制热力图，增强数据可视化的效果

3. 式(14)两边同时除以 Y 的标准差σ_y：

$$\frac{Y-\overline{Y}}{\sigma_y}=\beta_1\frac{\sigma_{x1}}{\sigma_y}\frac{(X_1-\overline{X_1})}{\sigma_{x1}}+\beta_2\frac{\sigma_{x2}}{\sigma_y}\frac{(X_2-\overline{X_2})}{\sigma_{x2}}+\beta_3\frac{\sigma_{x3}}{\sigma_y}\frac{(X_3-\overline{X_3})}{\sigma_{x3}} \quad (15)$$

4. 利用最小二乘法求出式(15)各自变量线性回归系数的求解模型，在此基础上，进行一定的数量变换，则可得出如下各简单相关系数的分解方程：

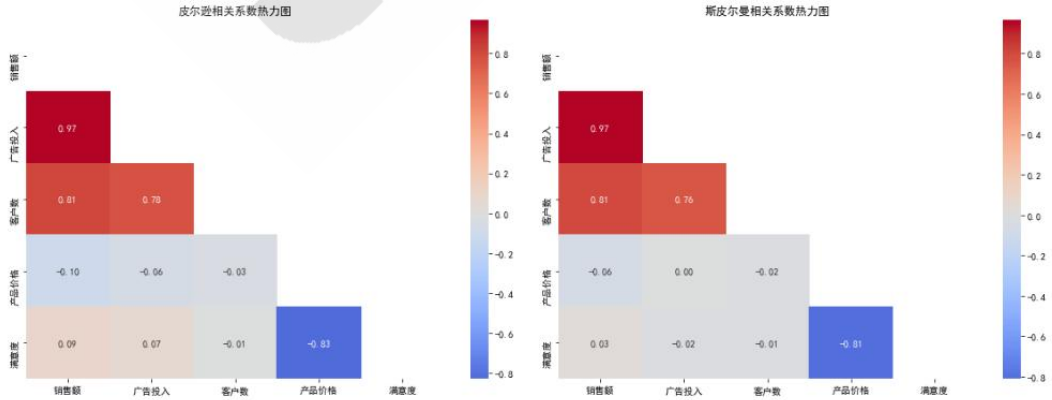
相关系数应用事例【2013 数模国赛 A 题】

首先先从总体的角度去分析不同品类蔬菜的分布规律及相互关系。分别使用 **K-S 检验、皮尔逊相关系数矩阵、ARIMA 时间序列分析**等方法，对销量数据从不同的维度进行整体分析；之后认为再从周、月、年等多个维度，利用**可视化分析和相关系数矩阵**等方法，详细分析不同品类蔬菜的分布规律及相互关系。

相关系数应用事例【2023 数模国赛 C 题】

1.4 相关系数代码

```
# 2. 计算相关系数矩阵
pearson_corr = df.corr(method='pearson')
spearman_corr = df.corr(method='spearman')

# 3. 绘制热力图
fig, axes = plt.subplots(1, 2, figsize=(16, 6))

# 皮尔逊热力图
sns.heatmap(pearson_corr,
            annot=True,
            fmt=".2f",
            cmap='coolwarm',
            center=0,
            ax=axes[0],
            mask=np.triu(np.ones_like(pearson_corr, dtype=bool)))
axes[0].set_title('皮尔逊相关系数热力图')  # 中文标题

# 斯皮尔曼热力图
sns.heatmap(spearman_corr,
            annot=True,
            fmt=".2f",
            cmap='coolwarm',
            center=0,
            ax=axes[1],
            mask=np.triu(np.ones_like(spearman_corr, dtype=bool)))
axes[1].set_title('斯皮尔曼相关系数热力图')  # 中文标题

plt.tight_layout()
plt.show()
```



2 回归分析

2.1 回归方程的设定

（1）回归方程设定

回归方程研究的是自变量 *x1* 、*x2* 、*x3*...对因变量 *y* 的影响

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p + \varepsilon \text{（变量形式）}$$

$$Y = X\beta + \varepsilon \text{（矩阵形式）}$$

其中ε代表误差项（扰动项），一般假设服从正态分布

（2）回归模型的假定

①回归模型参数线性：模型参数是以线性方式出现的，但变量本身可以是线性的或非线性的

②解释变量与扰动误差项不相关：保证解释变量不是随机的，并且在重复抽样中取固定值

③扰动项的期望或均值为零：这意味着误差项的平均值为零，不会系统性地偏向任何一方

④扰动项的方差为常数或同方差：即误差项的方差在整个样本范围内是恒定的

⑤无自相关，即两个误差项之间不相关：这意味着误差项之间没有序列相关性

⑥无完全多重共线性：对于多变量复回归模型，解释变量之间没有完全的线性关系

2.2 回归方程的评价

SST=SSE+SSR		
SST（Sum Square of Total） 总平方和	SSE（Sum Square of Explain） 可解释平方和	SSR（Sum Square of Residual） 总残差平方和
$\sum_{i=1}^{n}(y_i-\bar{y})^2$	$\sum_{i=1}^{n}(\hat{y}_i-\bar{\hat{y}})^2$	$\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$
总的变化量	回归方程可以解释的变化	回归方程不能解释的变化
$R^2=\frac{SSE}{SST}$		

（1）对于回归方差来说，SSE 越大，SSE 残差越小，回归效果越好；为了便于不同变量回归效果的比较，**可以定义无量纲回归效果指数，即解释方差**，又称判决系数 R^2

（2）R^2 越大，回归效果越好，表示两者之间的关系越密切

（3）R^2 相当于相关系数的平方，回归效果取决于相关系数平方的大小

（4）统计建模中一般采用**调整自由度后的 R^2 即 Adjusted_R^2**

问题一：

基于对数据的分析，本文认为有 Data1.xls 提供的 5 组数据能确定颜色读数与物质浓度之间的关系，并建立了多元线性回归模型：

$$Y=\varepsilon+C_1R+C_2G+C_3B+C_4H+C_5S \qquad (\text{I})$$

（I）式中C_1，C_2，C_3，C_4，C_5表示方程的回归系数。

回归方程数模事例【2017 数模国赛 A 题】

为此，我们尝试了线性回归、支持向量机、随机森林等多种模型，并以拟合接近度R^2进行评估，**最终发现随机森林模型与原数据拟合度可以达到99.999%，几乎能完全拟合原数据。因此，我们选择训练该模型并作为后续数据的支撑。拟合效果如下：**

回归方程数模事例【2023 数模国赛 B 题】

针对问题二，第一问首先计算出各蔬菜品类每日成本加成定价，然后通过 **Pearson 相关系数**检验出各蔬菜品类销售总量与成本加成定价**呈负线性相关关系**，于是建立线性回归模型，利用最小二乘法求出线性回归方程。第二问首先针对不同品类蔬菜建立

回归方程数模事例【2023 数模国赛 B 题】

2.3 回归方程的参数估计与假设检验

（1）最小二乘（OLS）参数估计结果： $\hat{\beta}=(X^TX)^{-1}X^TY$

（2）针对整个回归方程的检验：F 检验用于检验回归模型整体的显著性，即判断所有自变量对因变量的影响是否显著

原假设 H0: β_i 全为 0 vs 备择假设 H1: β_i 全为 0 不全为 0

F 统计量计算方法：$\dfrac{SSR_{H_0}/Df_{H_0}}{SSR_{H_1}/Df_{H_1}}=\dfrac{SSM/k}{SSE/(n-k-1)}\sim F(k,n-k-1)$

根据F 分布查找相应的临界值，或者计算p 值

结果解释：如果p 值小于显著性水平（通常为0.05），则拒绝原假设，认为至少有一个自变量对因变量有显著影响，使得整个模型是显著的

（3）针对单个系数的检验：t 检验用于检验回归模型中每个自变量系数是否显著不同于零

原假设 H0: $\beta_i=0$ vs 备择假设 H1: $\beta_i\neq 0$

t 统计量计算方法（分母即为该系数的标准误）$\frac{(\hat{\beta}-\beta)}{\sqrt{(X^TX)^{-1}S^2}} \sim T(n-k-1)$

根据 t 分布查找相应的临界值，也可以直接计算 p 值以决定是否拒绝原假设

结果解释：如 p 值小于显著性水平（通常为 0.05），则拒绝原假设，认为系数显著不为零

（4）注意事项

①t 检验用于评估每个单独系数的影响，能够判断每个自变量的显著性；

②F 检验用于评估整体模型的有效性，能够判断模型中所有自变量的组合是否显著地解释了因变量的变化

③t 检验和 F 检验这两种检验通常一起使用， 以全面评估线性回归模型的性能和解释力量

④一般先采用 F 检验，如果 F 检验能够通过再进行 t 检验

⑤在多元线性回归中，只需要解释 t 检验显著的系数

2.4 回归方程代码

```
# 构建设计矩阵（添加截距项）
X = np.column_stack((np.ones(n_samples), X1, X2, X3))

# 使用最小二乘法求解回归系数
coefficients = np.linalg.inv(X.T @ X) @ X.T @ y

# 提取系数
intercept = coefficients[0]
coef_X1 = coefficients[1]
coef_X2 = coefficients[2]
coef_X3 = coefficients[3]

# 预测值
y_pred = X @ coefficients

# 计算评估指标
residuals = y - y_pred
mse = np.mean(residuals**2)
tss = np.sum((y - np.mean(y))**2)
rss = np.sum(residuals**2)
r_squared = 1 - (rss / tss)

# 打印结果
print("多元线性回归结果:")
print(f"回归方程: y = {intercept:.2f} + {coef_X1:.2f}*X1 + {coef_X2:.2f}*X2 + {coef_X3:.2f}*X3")
print(f"均方误差(MSE): {mse:.2f}")
print(f"R²分数: {r_squared:.2f}")
```

II. 多元统计部分

3 聚类分析

3.1 K-means 聚类

1. K-means 聚类概述

K-means 聚类是一种常用的无监督学习算法，主要用于将数据集划分为多个聚类（簇）。其基本原理可以简单概括为以下几个步骤：

Step1 选择聚类数 K：首先，需要指定希望将数据划分为多少个簇（K 的值）

Step2 初始化中心：随机选择 K 个数据点作为初始的簇中心（centroids）

Step3 分配簇：将每个数据点分配给离其最近的聚类中心，形成 K 个簇

Step4 更新中心：计算每个簇的均值，并更新聚类中心的位置

Step5 重复步骤 3 和 4：直到聚类中心不再发生显著变化（或变化小于某个阈值），即算法收敛，结束迭代

2. K-means 聚类事例

一个典型的 K-means 聚类应用实例是市场细分和用户画像。假设某公司希望根据顾客的购买行为将客户进行分类，以便更有针对性地进行市场营销

注意：对于聚类算法，不同方法可能有不同结果，关键不是方法，而是对于结果的解释

针对问题一，第一问首先计算出蔬菜品类的标准差、偏度系数、峰度系数等描述统计量。然后进行数据可视化处理，分析各蔬菜品类、单品蔬菜的销售量分布规律。结果可得各单品和各蔬菜品类的日销售量**都呈现出不同程度的季节性波动**，其中花叶类和辣椒类蔬菜日销售量的波动性较大。第二问引入 Spearman 相关系数，以各蔬菜品类及单品蔬菜的日销售量为指标，进行相关性分析，求解得出**除茄类外，其它五种品类蔬菜之间都呈现出显著的正相关关系。**然后以单品的总销售量、每日最大销售量和日均销售量为指标，通过 **K-means++聚类算法**将单品划分为热销、畅销、平销和滞销四大类，进行相关性分析，结果可得**热销单品较其他单品呈现出高总销售量，高每日最大销售量和高日均销售量特点。**

聚类算法事例【2023 年数模国赛 A 题】

3.K-means 聚类代码

```
# 2. 肘部法则确定最佳K值
wcss = []  # 保存每个K值的WCSS(组内平方和)
silhouette_scores = []  # 保存轮廓系数

K_range = range(2, 11)
for k in K_range:
    kmeans = KMeans(n_clusters=k, init='k-means++', n_init=10, random_state=42)
    kmeans.fit(X_scaled)
    wcss.append(kmeans.inertia_)
    silhouette_scores.append(silhouette_score(X_scaled, kmeans.labels_))

# 绘制肘部法则图
plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
plt.plot(K_range, wcss, 'bo-')
plt.xlabel('聚类数(K)', fontsize=12)
plt.ylabel('WCSS (组内平方和)', fontsize=12)
plt.title('肘部法则', fontsize=14)
plt.grid(True)

# 绘制轮廓系数图
plt.subplot(1, 2, 2)
plt.plot(K_range, silhouette_scores, 'go-')
plt.xlabel('聚类数(K)', fontsize=12)
plt.ylabel('轮廓系数', fontsize=12)
plt.title('轮廓系数分析', fontsize=14)
plt.grid(True)

plt.tight_layout()
plt.show()
```

3.2 系统聚类

1. 系统聚类概述

系统聚类（Hierarchical Clustering）是一种将数据逐层聚类的无监督学习方法。它可以生成一个树状图（dendrogram），通过图形的高度可以了解数据之间的相似性。系统聚类的 原理主要包括以下步骤：

（1）选择距离度量：

确定数据点之间的相似性或距离的计算方法，常用的有欧氏距离、曼哈顿距离等

（2）构建聚类树：

- **自下而上的层次聚类（凝聚型，Agglomerative Clustering）：**从每个数据点开始，将最 近的两个聚类合并，直到所有点都被包含在一个大聚类中。
- **自上而下的层次聚类（分裂型，Divisive Clustering）：**从一个整体聚类开始，逐步将 其分裂为更小的聚类，直到每个数据点成为一个单独的聚类。

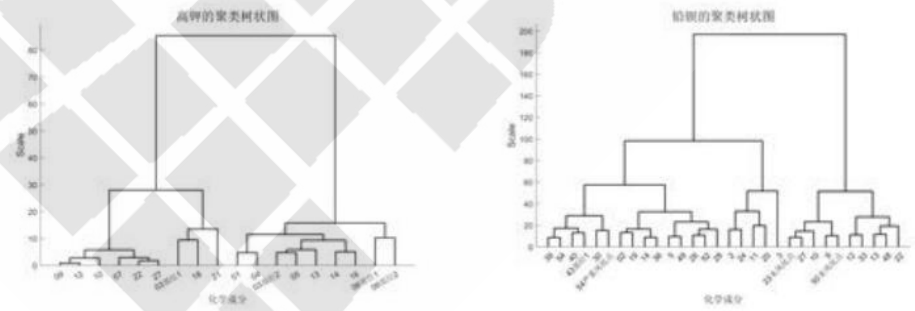
（3）确定最终聚类数：

通过设置阈值或观察树状图中聚类之间的距离，决定最终的聚类数。

肘部法则是一种用来确定聚类数（K 值）的方法，用于系统聚类。在画出不同K 值下的聚类效果时，查看误差平方和（SSE）或聚类间距。

针对问题二，第一小问要求分析铅钡玻璃以及高钾玻璃的分类规律，本文引入监督学习进行分类，采用**决策树法**对数据进行分类，求解结果发现该模型在精确率、召回率、准确率以及 F1 系数上均为 1，说明模型性能良好。第二小问要求对不同的玻璃类型选择适合的化学成分进行亚类分类，本文利用 **R 型聚类法**得到的特征变量为基础进行 **Q 型聚类**，相比于直接使用 Q 型聚类法具有更高的合理性，并通过在扰动范围内随意重新赋值，扰动范围处于 [0.1,0.2] 范围内，由此说明模型敏感性良好。

系统聚类事例【2022 年数模国赛 C 题】



(a) 高钾玻璃　　(b) 铅钡玻璃

系统聚类事例【2022 年数模国赛 C 题】

2. 系统聚类的注意事项

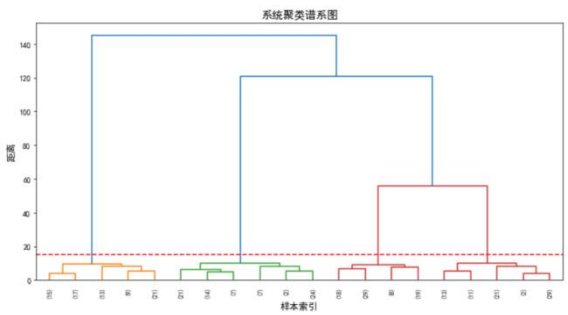
①在数模国赛中优先采用系统聚类，因为系统聚类的结果更容易解释，且可视化图片比其他聚类方法更为出彩；

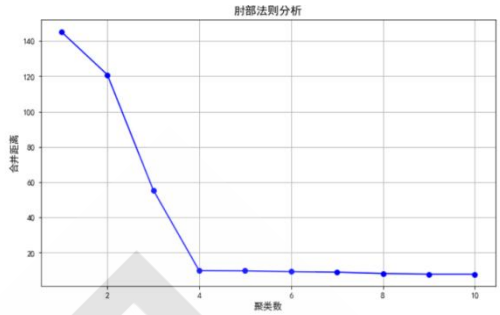
②系统聚类一般情况下是 Q 型聚类（针对样本聚类），极少采用 R 型聚类（针对变量聚类）；

③系统聚类的结果需要将**聚类结果表格单独整理好，连同聚类谱系图、肘部法则的碎石图**呈现在论文中，如果表格或者图片过长也应当放在附录中，这个比较重要最好不要漏放，否则

会被扣掉步骤分；

④系统聚类的聚类数量的判定要结合实际情况和肘部法则，不能一概而论。





通常情况下，随着K 值的增加，SSE 会下降。肘部法则的关键在于找到一个“肘部”点，通常在此点之后SSE 下降的幅度减小，表明增加K 值的收益减小，因此这个点对应的K 值可作为最佳聚类数

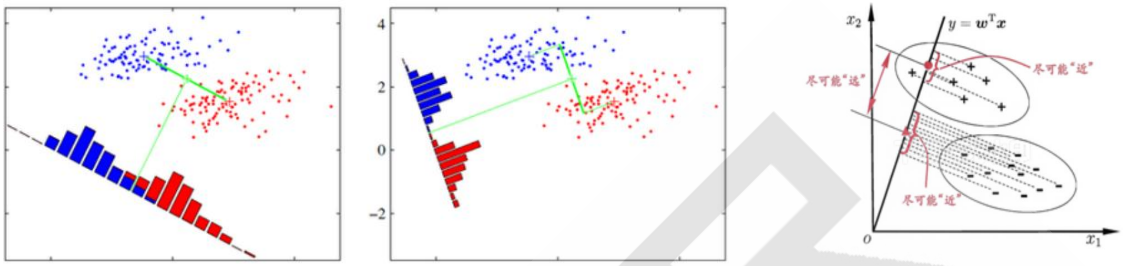
3. 系统聚类代码

```
# 2. 计算距离矩阵和链接矩阵
distance_matrix = pdist(X, metric='euclidean')
Z = linkage(distance_matrix, method='ward')

# 3. 绘制谱系图（树状图）
plt.figure(figsize=(12, 6))
plt.title('系统聚类谱系图', fontsize=14)
plt.xlabel('样本索引', fontsize=12)
plt.ylabel('距离', fontsize=12)
dendrogram(Z, truncate_mode='lastp', p=20, show_leaf_counts=True, leaf_rotation=90, leaf_font_size=8)
plt.axhline(y=15, color='r', linestyle='--')  # 示例切割线
plt.show()

# 4. 肘部法则分析（确定最佳聚类数）
last = Z[-10:, 2]  # 最后10次合并的距离
last_rev = last[::-1]  # 反转顺序
idxs = np.arange(1, len(last)+1)

plt.figure(figsize=(10, 6))
plt.plot(idxs, last_rev, 'b-', marker='o')
plt.title('肘部法则分析', fontsize=14)
plt.xlabel('聚类数', fontsize=12)
plt.ylabel('合并距离', fontsize=12)
plt.grid(True)
plt.show()

# 5. 轮廓系数分析（另一种确定最佳聚类数的方法）
range_n_clusters = range(2, 8)
silhouette_avg = []
for n_clusters in range_n_clusters:
    cluster_labels = fcluster(Z, n_clusters, criterion='maxclust')
    silhouette_avg.append(silhouette_score(X, cluster_labels))

plt.figure(figsize=(10, 6))
plt.plot(range_n_clusters, silhouette_avg, 'b-', marker='o')
plt.title('轮廓系数分析', fontsize=14)
plt.xlabel('聚类数', fontsize=12)
plt.ylabel('平均轮廓系数', fontsize=12)
plt.grid(True)
plt.show()
```

4 判别分析

4.1 判别分析原理

判别分析是一种经典的分类方法，旨在通过将数据投影到低维空间，最大化类间差异并最小化类内差异，从而实现对数据的分类。常用的判别分析方法包括线性判别分析（LDA）和二次判别分析（QDA）



判别分析原理展示【2022 数模国赛 A 题】

以上特征，我们对成分数据进行了中心对数比变换 (Centered logratio transformation, CLR)，变换后的各指标反映了某成分相对其他成分的含量大小（相对重要性），并且一定程度上改善了成分数据集的一些不良性质。

针对问题三，我们将问题二中的分类器应用于未分类数据。结果显示，除 A5 样品以外，其他文物样品均能被分类器完美区分，尤其是已风化样品 A2、A6、A7，经过还原其原始相对成分含量数据以后（即考虑了风化对其相关化学成分的相对含量的影响趋势），依然能够被分类器分开。并且模型对 A5 的“跨类行为”给出了合理解释。第三问中针对未分类样品的分类结果表明，该分类器可以有效地对风化后的玻璃文物做亚类区分，并且结果具有较好的合理性与可解释性。

判别分析原理展示【2022 数模国赛 C 题】

4.2 判别分析与聚类分析的区别

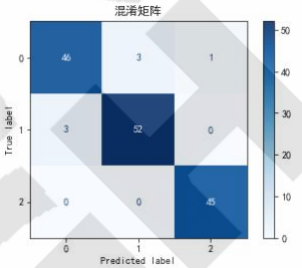
对比维度	聚类分析	判别分析
学习类型	无监督学习	有监督学习
输入数据	仅需特征变量 (X)	需要特征变量 (X) 和类别标签 (y)
核心目标	发现数据内在分组结构	建立分类规则预测新样本类别
典型算法	K-means、层次聚类、DBSCAN	LDA（线性判别）、QDA（二次判别）
输出结果	样本的聚类标签（新类别）	分类函数 + 新样本的预测类别
评估指标	轮廓系数、肘部法则	分类准确率、混淆矩阵、ROC 曲线
是否需要先验知识	不需要已知分类	依赖已知分类标签
可视化重点	簇分布（散点图/树状图）	分类边界（决策超平面）

聚类分析是"探索未知分组"，判别分析是"预测已知分类"

4.3 判别分析代码

```
# 2. 划分训练集和测试集
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# 3. 创建并训练LDA模型
lda = LinearDiscriminantAnalysis()
lda.fit(X_train, y_train)

# 4. 预测和评估
y_pred = lda.predict(X_test)

print("分类报告:")
print(classification_report(y_test, y_pred))
```

```
分类报告:
              precision    recall  f1-score   support

           0       0.94      0.92      0.93        50
           1       0.95      0.95      0.95        55
           2       0.98      1.00      0.99        45

    accuracy                           0.95       150
   macro avg       0.95      0.96      0.95       150
weighted avg       0.95      0.95      0.95       150
```



III. 时间序列部分

5 ARIMA 模型

5.1 时间序列的预处理

（1）时间序列的定义：按时间进行排序的**随机变量**

（2）时间序列的数字特征：

①均值函数：对于一个时间序列而言，任意时刻的序列值 X_t 都是一个随机变量，就一定存在某个常数 μ_t，**使得随机变量 X_t 总是围绕在常数 μ_t 附近做随机波动**。我们称 μ_t 为序列在 t 时刻的均值函数。

②方差函数：定义时间序列的方差函数以描述序列值围绕**其均值做随机波动时的平均波动**程度。当 t 取遍所有的观察时刻时，得到一个方差函数序列 σ_t

③自协方差函数：定义 $\gamma(t, s)$为序列的自协方差函数：$\gamma(t,s) = E(X_t - \mu_t)(X_s - \mu_s)$

④自相关系数（ACF）：定义 $\rho(t, s)$为时间序列的自相关系数：$\rho(t,s) = \dfrac{\gamma(t,s)}{\sqrt{DX_t \cdot DX_s}}$

（3）时间序列的平稳性：

- **严平稳：**序列所有的统计性质都不会随着时间的推移而发生变化

- **宽平稳：**认为序列的统计性质主要由它的低阶矩决定，所以只要保证序列低阶（二阶）矩平稳，就能保证序列的主要性质近似稳定。

保证时间序列平稳性的意义在于：只有平稳时间序列满足 Wold 分解定理

我们只能对平稳的时间序列进行建模，如果时间序列不平稳，要将其变为平稳序列

- **检验方法：**ADF 检验

（4）时间序列的纯随机性：

- **纯随机序列的定义：**如果时间序列满足如下性质：

任取 t∈T，有 $EX_t=\mu$;

任取 t，s ∈ T ，有 t 时刻和 s 时刻的序列不相关

称序列$\{X_t\}$为纯随机序列，也称白噪声序列，X_t ~ WN(μ, σ^2)

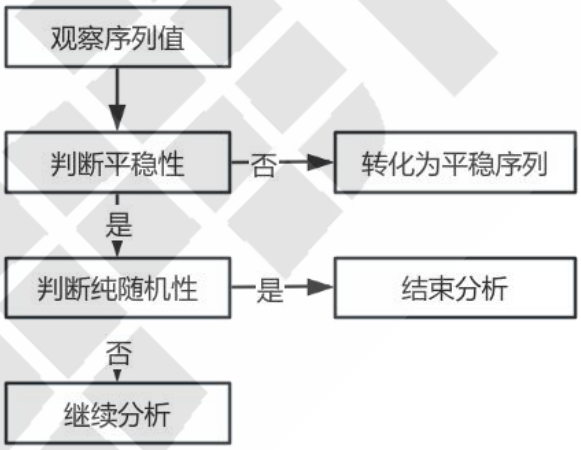
时间序列是纯随机的，这意味着我们没有办法从时间序列中提取任何有用的信息

因此，我们只能对非纯随机时间序列进行建模

- **检验方法：**Bartlett 检验

（5）时间序列的建模逻辑：

我们只能对**平稳、非纯随机**序列进行建模



（6）延迟算子与差分算子

B 表示对 X_t 滞后一项变为 X_{t-1}

一阶差分	$\nabla X_t=(1-B)X_t$
***p* 阶差分**	$\nabla^p X_t=(1-B)^p X_t$
***k* 步差分**	$\nabla_k X_t=(1-B^k)X_t$

5.2 AR 模型

$$\begin{cases} x_t = \phi_0 + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \ldots + \phi_p x_{t-p} + \varepsilon_t \\ \phi_p \neq 0 \\ E(\varepsilon_t) = 0, Var(\varepsilon_t) = \sigma_\varepsilon^2, E(\varepsilon_t \varepsilon_s) = 0, s \neq t \\ E(\varepsilon_t \varepsilon_s) = 0, \forall s < t \end{cases}$$

AR 模型的实质：用过去能够观测到的数据来预测未来的数据

①要保证最高阶数为 p

②要求随机干扰序列是零均值白噪声序列；当期的随机干扰与过去的序列值无关

③基于延迟算子和差分运算的表达：$\boldsymbol{\varphi}(B) = 1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p$

5.3 MA 模型

$$\begin{cases} x_t = \mu + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \ldots - \theta_q \varepsilon_{t-q} \\ \theta_p \neq 0 \\ E(\varepsilon_t) = 0, Var(\varepsilon_t) = \sigma_\varepsilon^2, E(\varepsilon_t \varepsilon_s) = 0, s \neq t \end{cases}$$

MA 模型的实质：用过去无法观测到的数据来预测未来的数据

①要保证最高阶数为 q

②要求随机干扰序列是零均值白噪声序列

③基于延迟算子的表达：$\boldsymbol{\theta}(B) = 1 - \theta_1 B - \theta_2 B^2 - \ldots - \theta_p B^q$

5.4 ARMA 模型与 ARIMA 模型

ARMA 模型的实质：综合使用过去能够观测到和无法观测到的数据来预测未来的数据

因此：ARMA 模型=AR 模型+MA 模型

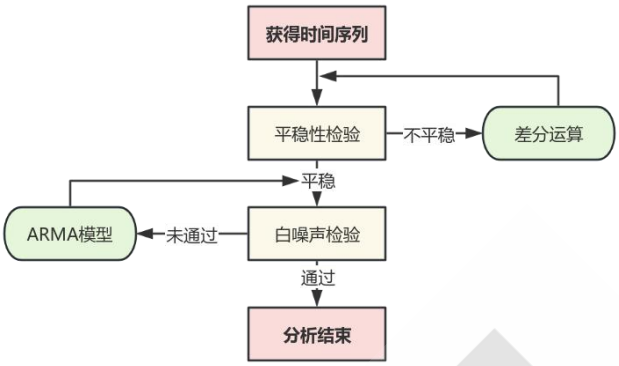
$$\begin{cases} x_t = \phi_0 + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \ldots + \phi_p x_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \ldots - \theta_q \varepsilon_{t-q} \\ \phi_p \neq 0, \theta_q \neq 0 \\ E(\varepsilon_t) = 0, Var(\varepsilon_t) = \sigma_\varepsilon^2, E(\varepsilon_t \varepsilon_s) = 0, s \neq t \\ E(\varepsilon_t \varepsilon_s) = 0, \forall s < t \end{cases}$$

ARIMA 模型的实质：对不平稳的时间序列差分后进行再通过 ARMA 模型进行建模

因此：ARIMA 模型=AR 模型+MA 模型+差分 I

$$\begin{cases} \Psi(B)\nabla^d x_t = \Theta(B)\varepsilon_t \\ E(\varepsilon_t) = 0, Var(\varepsilon_t) = \sigma_\varepsilon^2, E(\varepsilon_t \varepsilon_s) = 0, s \neq t \\ E(\varepsilon_t \varepsilon_s) = 0, \forall s < t \end{cases}$$

5.5 ARIMA 模型的定阶和参数估计



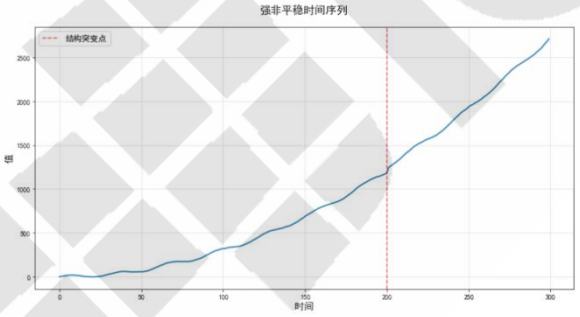
（1）应当率先确定差分 I 的阶数 d

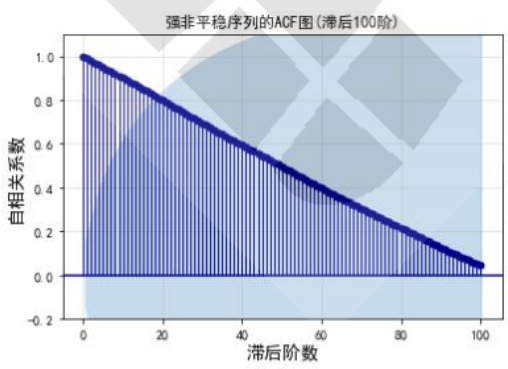
（2）再确定 AR 和 MA 模型的阶数 p 和 q

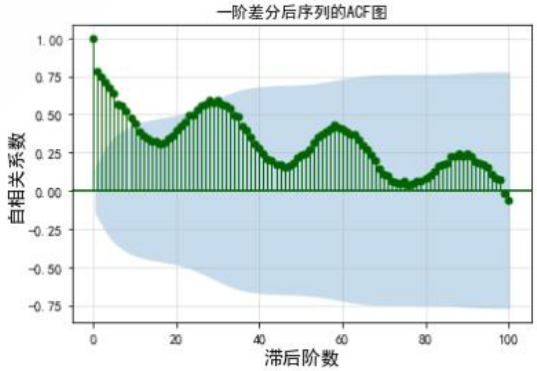
方法：绘制 ACF 和 PACF 图+AIC/BIC 的判定

ACF 图	PACF 图	适合拟合的模型
q 阶截尾	拖尾	MA(q)
拖尾	p 阶截尾	AR(p)
拖尾	拖尾	ARMA(p,q)

①平稳时间序列和非平稳时间序列的 ACF 图

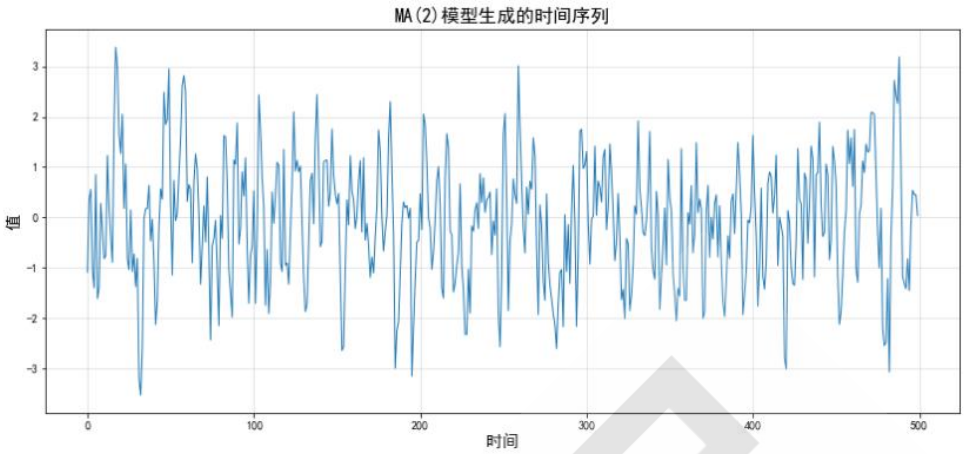


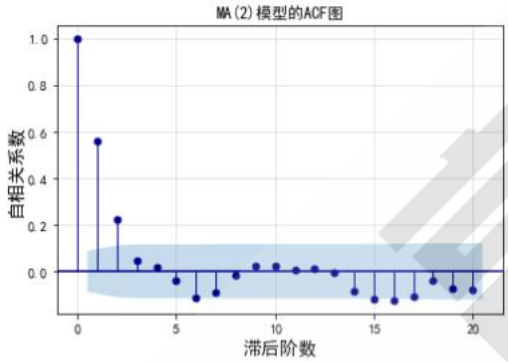


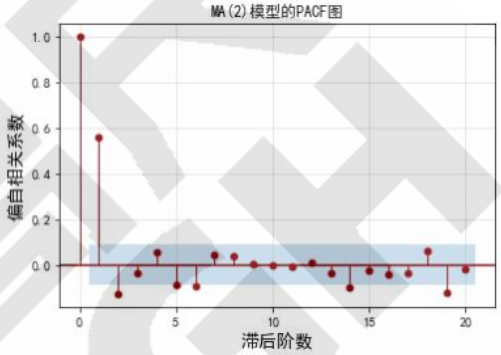


自相关系数衰减极其缓慢，即使在大滞后(如 100 阶)仍保持高位，季节性模式在 ACF 图中表现为周期性峰值(约 30 阶滞后处)

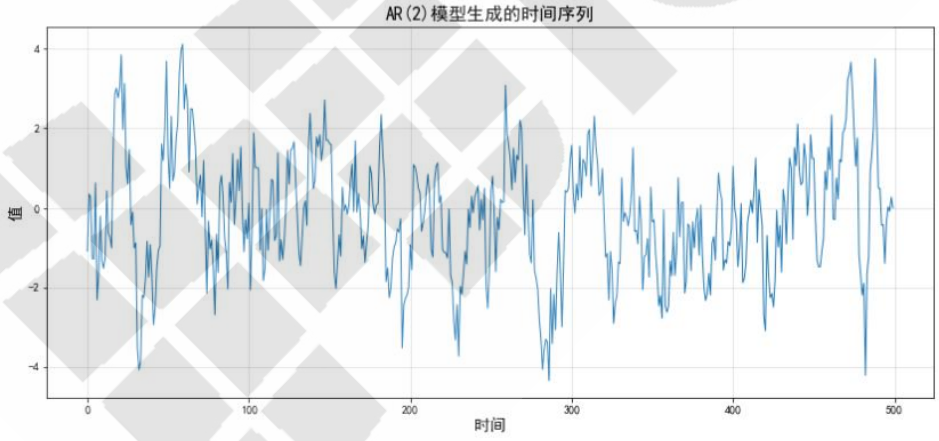
②MA 模型的 ACF 图与 PACF 图

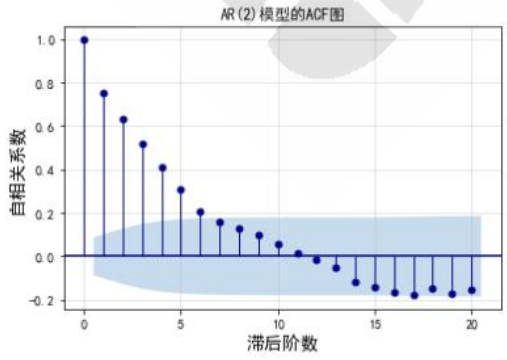


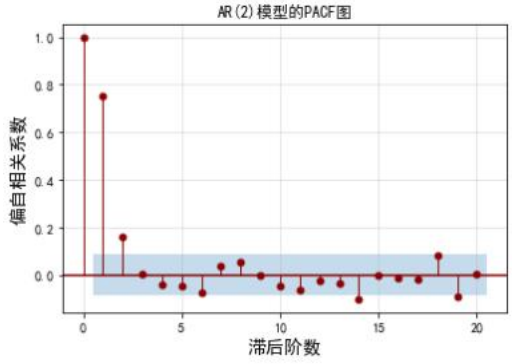




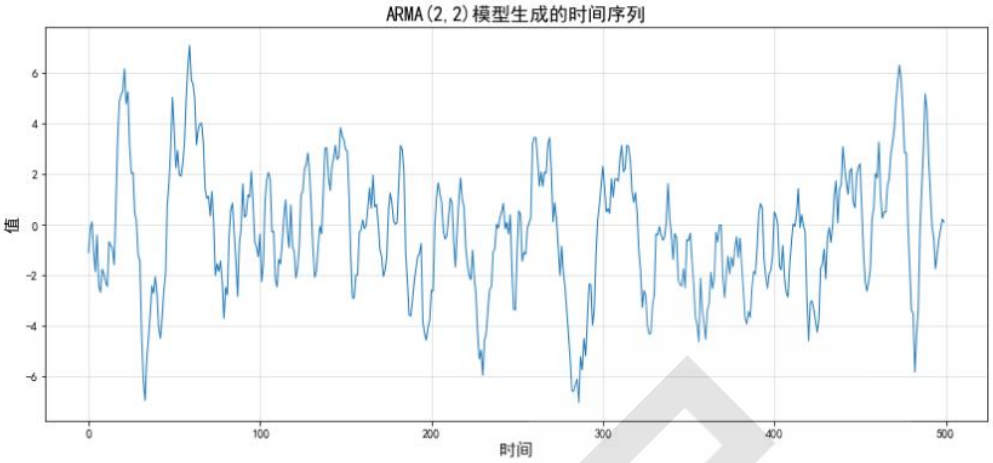
③AR 模型的 ACF 图与 PACF 图

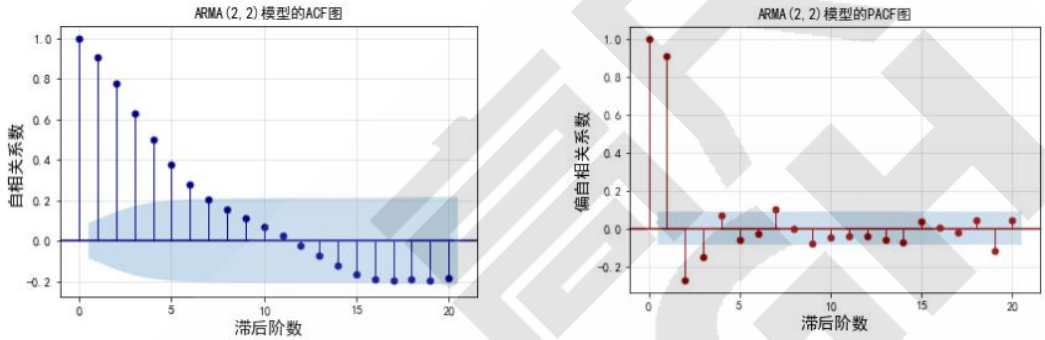






④ARMA 模型的 ACF 图与 PACF 图





（3）结合信息准则进行判断

AIC = -2 ln(模型的极大似然函数值) + 2（模型中未知参数个数）

BIC = -2 ln(模型的极大似然函数值) + ln n（模型中未知参数个数）

一般参考 BIC 的取值，BIC 越小越好（原因：贝叶斯结合了先验信息）

（4）综合考虑 AR、MA、差分 I 以及信息准则的情况：最终确定所选定的 ARIMA（p,d,q）的阶数**【注意：如果 ARIMA 的阶数判断不够清楚，可以多尝试几个不同的结束组合，一般为 p 和 q 的阶数，选择拟合和预测效果更好的那一个】**

因此拟合曲线为

$$y_t = -2.0232 + \epsilon_t - 0.7243\epsilon_{t-1} - 0.2757\epsilon_{t-2}$$

结合视频对图像进行分析，横断面的实际通行能力呈现上下波动的不稳定状态，但整体都在一条直线上下波动。在第二个时间点下降是因为车祸发生的时间刚好是上一次绿灯通过大量车到达车祸截面，由于人们原本还在自己选择的车道上，但是当发现车祸后，均会转移到右转车道，因此右转车道在开始短时间内可以顺利通行，当其他车道挤过来时，会产生排队效应，降低该车道的实际通行能力，因此第二个时间点的实际通行能力会比第一个时间点低。加上上游路口红绿灯是以60S为周期，所以整个截面通行能力呈现上下波动的情况。校园咖客收集整理（www.campustars.com）

ARIMA 模型实例【2013 年数模国赛 A 题】

对于问题一，为更好地分析，本文将问题聚焦为探究销售量的时间分布规律、销售分布规律以及相关关系，并**从多个时间颗粒维度分析总体、类别、个体的综合信息**。其一，我们使用 **ACF 自相关函数**与**时间序列分解**，发现了各销量在各时间颗粒维度上的波动规律和趋势，并结合生活与经济学理论完成了解释和检验。其二，分析销售规律时，

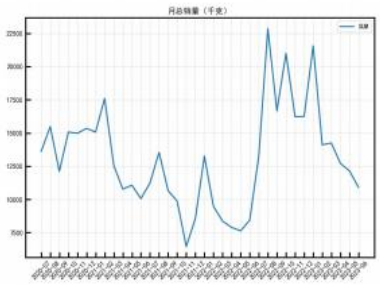


图 4　三年间蔬菜总销量曲线图

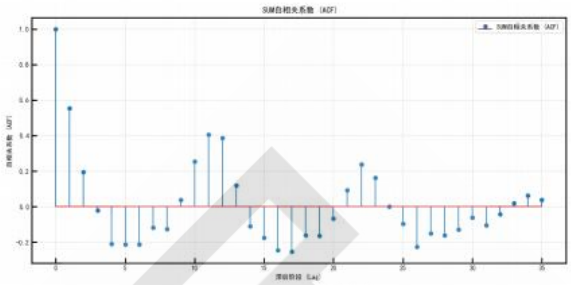


图 5　月总销售量自相关函数图

ARIMA 模型实例【2023 年数模国赛 C 题】

5.6 ARIMA 模型代码

```
# 创建ARMA(2,2)过程
arma_process = ArmaProcess.from_coeffs(ar_coeffs, ma_coeffs)

# 检查过程是否平稳
print("过程是否平稳:", arma_process.isstationary)

# 生成500个样本点
arma_series = arma_process.generate_sample(nsample=500)

# 绘制时间序列
plt.figure(figsize=(14, 6))
plt.plot(arma_series, linewidth=1)
plt.title("ARMA(2,2)模型生成的时间序列", fontsize=16)
plt.xlabel("时间", fontsize=14)
plt.ylabel("值", fontsize=14)
plt.grid(True, alpha=0.4)
plt.show()

# 绘制ACF图(展示20阶滞后)
plt.figure(figsize=(14, 6))
plot_acf(arma_series, lags=20, alpha=0.05,
         title="ARMA(2,2)模型的ACF图",
         color='darkblue', vlines_kwargs={"colors": 'darkblue', "linewidths": 1})
plt.xlabel("滞后阶数", fontsize=14)
plt.ylabel("自相关系数", fontsize=14)
plt.grid(True, alpha=0.4)
plt.show()

# 绘制PACF图(展示20阶滞后)
plt.figure(figsize=(14, 6))
plot_pacf(arma_series, lags=20, alpha=0.05,
          title="ARMA(2,2)模型的PACF图",
          color='darkred', vlines_kwargs={"colors": 'darkred', "linewidths": 1})
plt.xlabel("滞后阶数", fontsize=14)
plt.ylabel("偏自相关系数", fontsize=14)
plt.grid(True, alpha=0.4)
plt.show()
```

6 灰色预测模型

6.1 灰色预测原理

灰色预测模型（Grey Prediction Model）是一种基于少量、不完全信息进行预测的方法，适用于**数据量少、信息不完全的系统**。其核心思想是通过对原始数据进行累加生成，弱化数据的随机性，挖掘数据的内在规律，从而建立微分方程模型进行预测。

灰色预测模型中最常用的是 GM(1, 1) 模型，其特点如下：

（1）数据累加生成：通过对原始数据进行一次累加生成，使数据呈现指数增长趋势

（2）建立微分方程：基于累加生成序列，建立一阶线性微分方程

（3）参数估计：通过最小二乘法估计微分方程的参数

（4）预测与还原：利用模型预测累加生成序列，再通过累减生成还原原始序列的预测值

6.2 灰色预测步骤

1.原始数据：

$$X^{(0)}=\{x^{(0)}(1),x^{(0)}(2),...,x^{(0)}(n)\}$$

2.累加生成序列：

$$X^{(1)}(k)=\sum_{i=1}^{k}x^{(0)}(i),k=1,2,...,n$$

3.建立微分方程：

GM(1, 1) 模型的一阶微分方程为：$\frac{dx^{(1)}}{dt}+ax^{(1)}=b$

其中，a 和 b 是待估计的参数

4.求解微分方程：

微分方程的解为：

$$\hat{x}^{(1)}(k+1)=(x^{(0)}(1)-\frac{b}{a})e^{-ak}+\frac{b}{a}$$

5.累减还原（IAGO）：

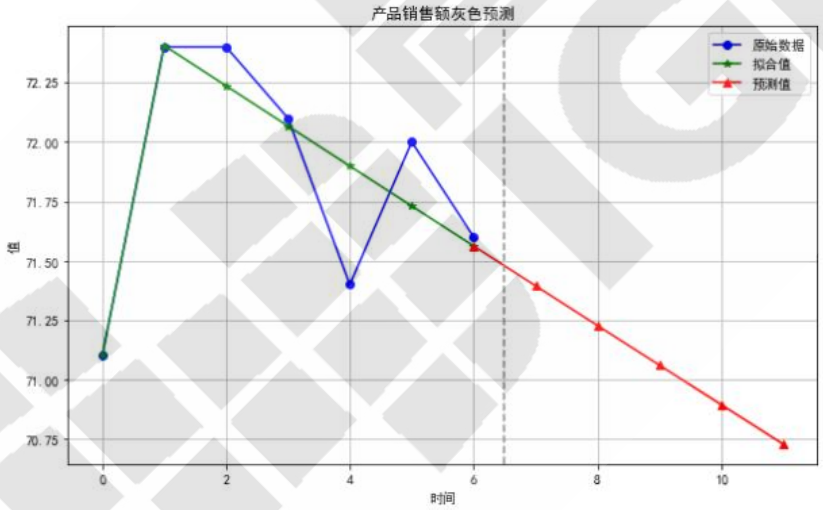
将累加生成序列的预测值还原为原始序列的预测值：

$$\hat{x}^{(0)}(k+1)=\hat{x}^{(1)}(k+1)-\hat{x}^{(1)}(k)$$

6.3 灰色预测代码

```python
def __init__(self):
    self.a = None  # 发展系数
    self.b = None  # 灰色作用量
    self.x0 = None  # 原始序列
    self.x1 = None  # 累加生成序列
    self.pred_values = None  # 预测值

def fit(self, data: np.ndarray) -> Tuple[float, float]:
    """拟合GM(1,1)模型"""
    self.x0 = data
    # 1. 累加生成(1-AGO)
    self.x1 = np.cumsum(self.x0)

    # 2. 构造数据矩阵B和Y
    B = np.zeros((len(self.x0)-1, 2))
    for i in range(len(self.x0)-1):
        B[i, 0] = -0.5 * (self.x1[i] + self.x1[i+1])
        B[i, 1] = 1
    Y = self.x0[1:].reshape(-1, 1)

    # 3. 计算参数a和b
    BT = B.T
    BTB_inv = np.linalg.inv(np.dot(BT, B))
    self.a, self.b = np.dot(np.dot(BTB_inv, BT), Y).flatten()

    return self.a, self.b
```

```
模型参数: a(发展系数)=0.0023, b(灰色作用量)=72.6573
预测结果: [71.1  72.41 72.24 72.07 71.9  71.73 71.56 71.39 71.23 71.06 70.89 70.73]
模型评估: MAPE=0.20%, RMSE=0.2238
```



首先，我们基于层次分析法建立了模型一。模型一以五个要素，即教育市场供求关系、全国家庭支付承受力、国家财政及相关社会捐助、个人收益率、教育成本为方案层。对于教育市场的供求关系我们用灰色预测 GM（1，1）模型预测出未来几年的招生人数，用蛛网模型求解稳定的价格点为 3225.51 元；对于国家财政及相关社会捐助，我们用回归分析得出其效应关系。模型一以效率和公平两个标准作为准则层，应用极差归一化思想，构造指标函数，综合建立成对比较矩阵。我们定义学费合理化指数为目标层，经准则层，得出五个要素对学费合理化指数的组合权重向量。考虑到成对比较矩阵仍有一定主观因素，我们用熵值取权法修正组合权重向量。最后，拟合出最佳学费曲线及其波动区间，其中 2007 年的结论值为 3370.75 元。模型一的突出优点是客观可信，美中不足的是结论为一个平均最优值，没有考虑其他变因的影响，使用的局限性较大。

灰色预测事例【2008 年数模国赛 A 题】