



广东外语外贸大学

市场调查与统计预测期末课程论文

题目 金价走势预测：基于 ARIMA 和 GM 模型的比较研究

姓名 章静怡

学号 20212005202

所在学院 数学与统计学院

年级专业 经济统计学 2021

完成时间 2024 年 6 月 10 日

摘要：随着全球经济的发展和金融市场的日益复杂化，黄金期货市场逐渐成为市场参与者关注的焦点。准确预测黄金价格不仅有助于投资者制定有效的投资策略，还有助于金融机构进行风险管理和资产配置，为金融市场的稳定发展提供重要参考和支持。本文选取印度泰米尔纳德邦 2006 至 2020 年的日度黄金汇率数据，构建 ARIMA(2,1,2)模型和 GM(1,1)模型进行黄金价格预测。研究结果显示，ARIMA 模型在捕捉长期趋势和周期性波动方面表现良好，但其预测结果可能受到滞后效应的影响。与之相比，GM 模型展现出对非线性数据的良好适应性，尤其在预测长期趋势和整体价格走势方面表现突出，但数据的准确性较差，基于此，本文建议在实际应用中采用多模型结合的方法，综合利用不同模型的优势，以提高预测的准确性和稳定性，为黄金价格预测提供更全面和准确的支持，助力投资者和金融机构在复杂多变的市场环境中做出更科学的决策。

关键词：黄金价格预测 ARIMA 模型 GM 模型

Abstract: With the global economy's development and increasing financial market complexity, the gold futures market has gained significant attention. Accurate gold price prediction aids investors in forming effective strategies and helps financial institutions in risk management and asset allocation, supporting financial market stability. This paper analyzes daily gold exchange rate data from Tamil Nadu, India, from 2006 to 2020, using ARIMA(2,1,2) and GM(1,1) models for forecasting. The ARIMA model effectively captures long-term trends and cyclical fluctuations but is susceptible to hysteresis effects. The GM model adapts well to nonlinear data and long-term trends but has lower data accuracy. Therefore, combining multiple models is recommended to leverage their strengths, enhancing prediction accuracy and stability. This approach provides better support for decision-making in a complex market.

Key words: Gold Price Forecasting ARIMA Model GM Model

目 录

摘 要:	1
Abstract:	1
一、引言	4
二、文献综述	5
三、模型设计	6
3.1 数据和样本	6
3.2 ARIMA 模型搭建	7
3.3 ARIMA 模型解释	7
3.4 GM 模型搭建	7
四、实证分析	9
4.1 数据可视化分析	9
4.1.1 数据趋势分析	9
4.1.2 数据分布分析	10
4.2 指标选取	10
4.2.1 市场流动性	10
4.2.2 价格稳定性	10
五、ARIMA 模型构建	11
5.1 平稳性和随机性	11
5.2 自相关图和偏相关图	13
5.3 残差检验	15
5.4 模拟预测	16
六、GM (1, 1) 模型构建	18
6.1 数据预处理	18
6.2 模型的有效性检验	19
6.3 模型预测	19
七、模型比较及结论建议	21
7.1 模型比较	21
7.2 结论及改进建议	23
参考文献	24
附 录一：原始及预测后数据	25
附 录二：代码	26

一、引言

随着全球经济的发展和金融市场的日益复杂化，黄金期货市场逐渐成为市场参与者关注的焦点。黄金作为一种重要的避险资产，其价格波动直接影响投资者和金融机构的决策。因此，准确预测黄金价格具有重要意义。通过准确的黄金价格预测，不仅有助于投资者制定有效的投资策略，还有助于金融机构进行风险管理和资产配置，从而为金融市场的稳定发展提供重要参考和支持。

黄金价格的预测对于投资者和金融机构具有双重意义。首先，对于投资者而言，准确的价格预测可以帮助其在黄金市场中制定更有效的投资策略，从而最大化收益并降低风险。据世界黄金协会（World Gold Council）¹指出，黄金作为避险资产和多元化投资组合的重要组成部分，能够在金融市场不确定性增加时提供保护。其次，对于金融机构来说，准确的黄金价格预测有助于进行资产配置和风险管理，保障其财务稳定性。根据国际货币基金组织（IMF）²的报告，黄金储备在国家外汇储备中起着重要的稳定作用，对国家经济安全具有重要意义。总体而言，黄金价格预测的研究不仅有助于个体决策，还对金融市场的整体稳定起到积极作用。

目前，学者们在黄金价格预测中应用了多种方法，包括传统计量经济学模型、人工智能模型和高级统计方法。尽管现有研究在黄金价格预测方面取得了显著成果，但仍存在一些不足。传统的 ARIMA 模型在处理非线性和复杂波动时其预测性能有限。机器学习和高级统计方法如 LASSO-QRNN 尽管提高了预测精度，但其复杂性和计算成本较高。灰色系统模型（GM 模型）虽然在处理小样本和高不确定性数据方面具有优势，但在处理大样本和长期预测中的表现仍需进一步验证。

本文选取 ARIMA 模型和 GM 模型进行黄金价格预测结果的比较研究。ARIMA 模型是传统计量经济学模型中的经典方法，广泛应用于时间序列预测，尤其在处理稳定的线性数据时表现良好，具有较高的预测精度。其次，GM 模型在处理小样本和高不确定性数据方面具有独特优势，其建模过程相对简单，对数据的依赖性较低，能够有效应对非线性和复杂波动的数据特征。

¹ World Gold Council. "The Relevance of Gold as a Strategic Asset."

² International Monetary Fund. "Gold in the IMF."

通过比较 ARIMA 模型和 GM 模型在黄金价格预测中的表现，可以深入理解这两种模型在不同数据特征和预测场景下的优劣势，从而为实际应用提供科学依据。具体而言，这一比较研究不仅能够验证 ARIMA 和 GM 模型在黄金价格预测中的有效性，还可以为不同市场条件下的模型选择提供指导，帮助投资者和金融机构在不同情境下选择最适合的预测工具，提高投资决策和风险管理的科学性和可靠性。

本文使用印度泰米尔纳德邦的黄金汇率 2006 至 2020 年的日度数据，通过构建 ARIMA(2,1,2)模型和 GM(1,1)模型进行黄金价格预测。预测结果显示 2020 年 10 月 10 日至 20 日印度黄金价格将持续上升。ARIMA 模型在捕捉长期趋势和周期性波动方面表现良好，但其预测结果可能受到滞后效应的影响，特别是在需要即时调整以反映最新市场变化时。与之相比，GM 模型展现出对非线性和不完全数据的良好适应性，尤其在预测长期趋势和整体价格走势方面表现突出，但数据的预测结果准确性较差。

基于此，本文建议在实际应用中，可以采用多模型结合的方法，综合利用不同模型的优势，以提高预测的准确性和稳定性。此外，未来的研究可以进一步探索其他先进模型和混合模型，结合多种数据源和变量，为黄金价格预测提供更全面和准确的支持，助力投资者和金融机构在复杂多变的市场环境中做出更科学的决策。

二、文献综述

金价预测不仅能帮助投资者制定有效的投资策略，还能帮助金融机构进行风险管理和资产配置。因此，研究金价预测模型对金融市场的稳定发展具有重要的参考和支持作用。

在研究方法方面，多数学者选择将传统的计量经济学模型与人工智能模型相结合，以提高预测准确性。Zakaria 等（2018）通过改进的多层感知器神经网络和鲸鱼优化算法对 ARIMA 模型的预测性能进行了评估，发现 ARIMA 模型在处理稳定的线性数据时表现良好。此外，ANN-GARCH 模型显著降低了金价波动的预测误差，其输入变量包括汇率波动、股票市场回报和油价波动。

机器学习和高级统计方法也在黄金价格预测中得到了广泛应用。Liu (2023) 采用 LASSO-QRNN 模型, 通过筛选出关键变量, 结合分位回归神经网络, 对黄金期货价格进行了高精度预测。Dadabada 和 Vadlamani (2020) 则运用粒子群优化训练的分位回归神经网络 (PSO-QRNN), 在处理复杂非线性数据时表现出色, 显著提高了预测性能。

灰色系统模型 (GM 模型) 在处理小样本和高不确定性数据方面具有优势。例如, 高菲和顾炜宇 (2018) 利用 SVAR 模型对国际黄金价格的影响因素进行了实证分析, 表明 GM 模型在处理不完全数据和高不确定性数据时具有较好的适应性和鲁棒性。

尽管现有研究在黄金价格预测方面取得了显著进展, 但仍存在一些不足之处。传统的 ARIMA 模型在处理非线性和复杂波动时, 其预测性能有限。机器学习和高级统计方法, 如 LASSO-QRNN, 虽然提高了预测精度, 但其复杂性和计算成本较高。GM 模型在处理大样本和进行长期预测方面的表现仍需进一步验证。

本文将通过比较 ARIMA 模型和 GM 模型在黄金价格预测中的应用效果, 探索其在不同预测场景中的优劣。通过比较两种模型的预测结果, 探讨其在不同时间段和数据特征下的表现, 总结两种模型的优缺点。本文旨在为黄金价格预测提供更加科学可靠的方法, 并为实际应用提供参考依据, 提高预测模型在金融市场中的实用性和有效性。

三、 模型设计

3.1 数据和样本

本文选取了 2006 年 1 月 2 日至 2020 年 10 月 10 日的印度泰米尔纳德邦的黄金汇率数据³, 以探究印度黄金价格的季节性变化以及长期趋势。随后本研究基于 Windows 系统, 采用 R 4.3.1 进行数据可视化。具体绘图过程使用了 R 语言的基础包`ggplot2`和`plotly`函数。`ggplot2`用于静态图形绘制, `plotly`则用

³ [金奈黄金汇率 | 金奈今日黄金汇率 | 22Kt 今日黄金汇率 \(15/Jun/2024\) - LiveChennai](#)

于交互式图形分析。通过这两种工具的结合，实现了数据结果的多维展示和互动分析，提升了结果的可视化效果。

3.2 ARIMA 模型搭建

ARIMA 模型，即自回归积分滑动模型，是一种结合自回归（Auto-Regressive, AR）、差分（Differencing, I）和滑动（Moving Average, MA）过程的时间序列分析方法。其数学表达式为：

$$Y_t = \mu + \sum_{i=1}^p \varphi_i Y_{t-i} + \sum_{j=1}^q \theta_j \varepsilon_{t-j} + \varepsilon_t \quad (1)$$

其中： Y_t 是当前值， μ 是常数项， φ_i 是自回归系数， ε_t 是随机误差项， θ_j 是滑动系数。

3.3 ARIMA 模型解释

p （自回归系数）表示序列值滞后 p 阶的自相关程度。 d （差分次数）表示使非平稳时间序列变成平稳序列所需的最少差分次数。 q （滑动系数）表示误差项滞后 q 阶的移动程度。

对时间序列数据进行自相关图（ACF）和偏自相关图（PACF）的平稳性检验。若序列为非平稳序列，则需要对其进行差分处理，并再次进行平稳性检验，直到满足平稳性条件。满足条件的差分次数即为 d 值。

通过上述方法确定 ARIMA 模型的参数后，采用 SAS 中的 `ARIMA` 库进行模型拟合和验证。

3.4 GM 模型搭建

本文首先将原始数据记为：

$$X^{(0)} = \{x^{(0)}, x^{(0)}(2), \dots, x^{(0)}(n)\} \quad (2)$$

式(2)中： $X^{(0)}$ 表示初始代数据集（第 0 代）， $X^{(0)}(n)$ 表示初始代数据中的第 n 个元素。在初代数据构建完成后，通过对初代数据进行一次累加得到了新的时间序列 $X^{(1)}$ ， $X^{(1)}$ 又称为 $X^{(0)}$ 的 1-AGO 序列，如下：

$$X^{(1)} = \{x^{(1)}(1), x^{(1)}(2), \dots, x^{(1)}(n)\} \quad (3)$$

$$x^{(1)}(k) = \sum_{i=1}^k x^{(0)}(i) \quad (4)$$

然后，通过初始数据序列 $X^{(0)}$ 和 1-AGO 序列 $X^{(1)}$ ，可以计算出 $X^{(1)}$ 的紧邻均值生成序列 $Z^{(1)}$ ，如下：

$$Z^{(1)} = \{z^{(1)}(2), z^{(1)}(3), \dots, z^{(1)}(n)\} \quad (5)$$

$$z^{(1)}(k) = \frac{1}{2}(x^{(1)}(k) + x^{(1)}(k-1)), k = 2, 3, \dots, n \quad (6)$$

于是就有：

$$x^{(0)}(k) + az^{(1)}(k) = b, k = 2, 3, \dots, n \quad (7)$$

我们称式(7)为 GM(1,1)模型的基本形式。其中， $-a$ 表示发展系数， b 表示灰作用量。对参数列进行最小二乘估计，设 $\hat{a} = (a, b)^T$ 为参数列，且满足下式：

$$Y = \begin{bmatrix} x^{(0)}(2) \\ x^{(0)}(3) \\ \dots \\ x^{(0)}(n) \end{bmatrix}, B = \begin{bmatrix} -z^{(1)}(2) & 1 \\ -z^{(1)}(3) & 1 \\ \dots & \dots \\ -z^{(1)}(n) & 1 \end{bmatrix} \quad (8)$$

$$\hat{a} = (a, b)^T = (B^T B)^{-1} B^T Y \quad (9)$$

即最小二乘估计参数列 $\hat{a} = (B^T B)^{-1} B^T Y$ ，由此得出式子(7)的白化方程为

$$\frac{dx^{(1)}(t)}{dt} = -\hat{a}x^{(1)}(t) + \hat{b} \quad (10)$$

及其时间响应方程：

$$\hat{x}^{(1)}(t) = \left(x^{(0)}(1) - \frac{b}{a} \right) e^{-a(t-1)} + \frac{b}{a}, t = 2, 3, \dots, n \quad (11)$$

最后对 $X^{(1)}(t)$ 作一阶累减，即可还原得到 $X^{(0)}$ 的模拟值：

$$\hat{x}^{(0)}(t) = \hat{x}^{(1)}(t) - \hat{x}^{(1)}(t-1), t = 2, 3, \dots, n \quad (12)$$

通过上述方法确定 GM(1,1)模型的参数后，在 R 语言中编写 GM 函数进行模型拟合和验证。

四、实证分析

4.1 数据可视化分析

4.1.1 数据趋势分析

图 1 展示了从 2007 年至 2020 年期间标准金（22K）和纯金（24K）的价格变化趋势。整体来看，金价在这段时间内呈现出显著的上升趋势，尤其是在 2008 年金融危机后和 2019 年新冠疫情期间，金价出现了明显的上涨。标准金和纯金的价格走势高度相似，但纯金价格通常略高于标准金，反映出纯金的市场价值更高。



图 1 印度金价趋势图（2007-2020）

4.1.2 数据分布分析

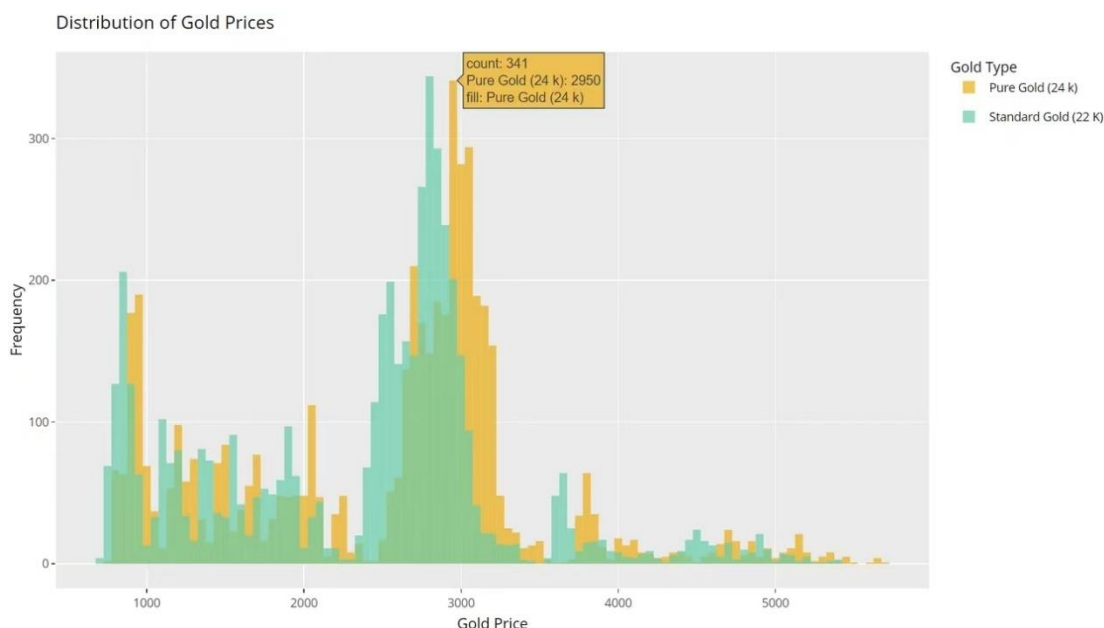


图 2 印度金价分布图（2007-2020）

图 2 展示了同一时间范围内标准金（22K）和纯金（24K）价格的频率分布情况。大部分金价集中在 2000 至 3000 的区间内，特别是在 2500 至 3000 的区间，频率最高。标准金和纯金的价格分布形状相似，但纯金的价格分布略偏右，说明其整体价格水平高于标准金，这与时序图中的观察结果一致。

4.2 指标选取

4.2.1 市场流动性

图 1 和图 2 均显示出标准金（22K）和纯金（24K）的价格走势和分布形态高度相似，但标准金的市场交易量通常更大，流动性更强。标准金在市场中的广泛应用使其价格数据更加丰富和可靠，这有助于提高研究结果的准确性和适用性。选择标准金能更好地反映市场的整体情况，减少由于数据稀疏性带来的误差。

4.2.2 价格稳定性

从图 2 的价格分布图可以看出，尽管纯金的价格通常略高于标准金，但其价格波动性较大，分布略偏右，表明纯金价格更容易受到短期市场波动的影响。

响。标准金的价格波动性较小，能够更稳定地反映长期市场趋势，这对于建立和验证价格预测模型是有利的。因此，选择标准金有助于减少短期波动对预测结果的干扰，提供更可靠的长期预测。

综上所述，选取标准金作为研究对象不仅能够更好地反映市场整体趋势，提高研究结果的可靠性和应用价值，还能为实际投资和行业应用提供重要的市场信息和决策支持。

五、ARIMA 模型构建

5.1 平稳性和随机性

R 语言具备强大的图形功能和高度灵活性，能够高效处理大规模数据集。其丰富的绘图包（如 ggplot2）可以定制高分辨率图表，清晰展示标准金（22K）和纯金（24K）的价格变化趋势和分布情况。这确保了数据的完整性和准确性，同时使得图表更加直观和符合学术论文的格式要求。⁴

根据原始历史的数据画出的序列图可以看出印度黄金价格呈现上涨趋势，初步判断为不平稳序列，如图 3 所示。



图 3 原时序图

⁴ 在本研究中，由于日度数据量过大，使用 SAS 软件绘制时序图时，图表呈现不清晰，难以满足学术论文的格式和精度要求。为了解决这一问题并提高图表的清晰度和可读性，本文选取了 R 语言进行数据可视化。

故对原序列就行一阶差分，差分后结果如图 4 所示。一阶差分后去除全序列的趋势性，基本可以认定为平稳序列。

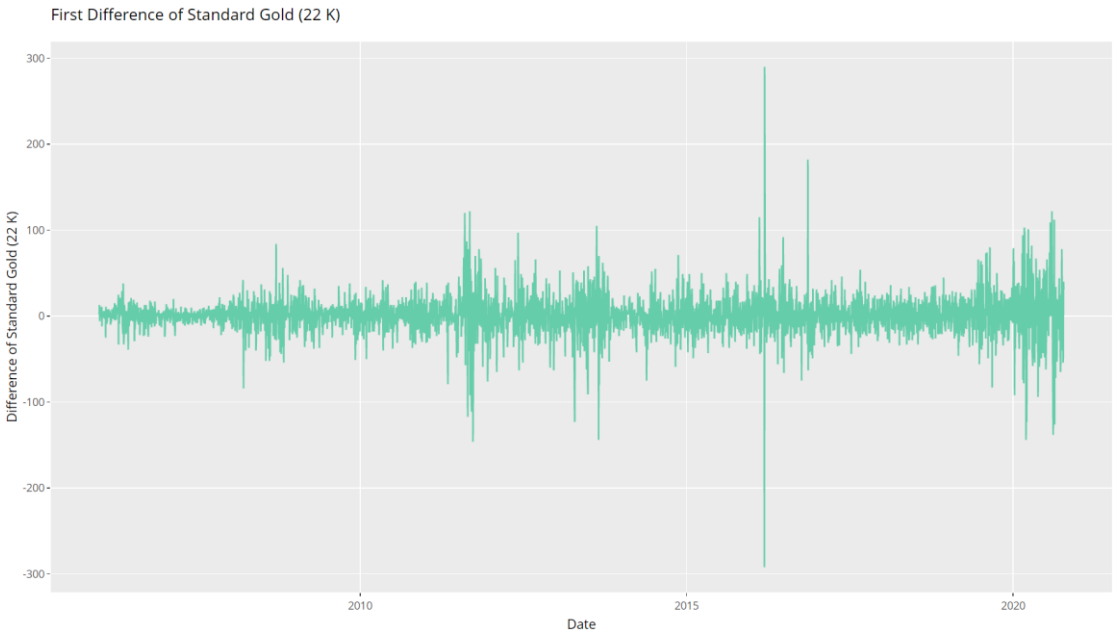


图 4 差分后的时序图

进一步使用 ADF 检验，检验该序列的平稳性，结果如下表 1 所示：

表 1 单位根检验结果表

增广 Dickey-Fuller 单位根检验							
类型	滞后	Rho	Pr < Rho	Tau	Pr < Tau	F	Pr > F
Zero Mean	0	-5159.84	0.0001	-73.23	<.0001		
	1	-4983.27	0.0001	-49.87	<.0001		
	2	-4746.96	0.0001	-40.02	<.0001		
Single Mean	0	-5168.63	0.0001	-73.35	<.0001	2690.20	0.0010
	1	-5009.66	0.0001	-50.00	<.0001	1249.94	0.0010
	2	-4796.29	0.0001	-40.16	<.0001	806.50	0.0010
Trend	0	-5169.72	0.0001	-73.36	<.0001	2690.88	0.0010
	1	-5012.94	0.0001	-50.01	<.0001	1250.58	0.0010
	2	-4802.41	0.0001	-40.18	<.0001	807.10	0.0010

表 2 纯随机性检验结果表

白噪声的自相关检查									
至滞后	卡方	自由度	Pr > 卡方	自相关					
6	15.20	6	0.0187	-0.040	0.017	0.013	0.030	0.009	-0.003
12	24.05	12	0.0200	-0.030	0.025	-0.002	0.004	-0.006	-0.013
18	32.94	18	0.0170	0.012	0.013	-0.007	-0.026	0.027	0.005
24	42.62	24	0.0110	-0.040	-0.001	0.000	-0.011	0.001	-0.016

结果显示（表 1）， p 值均小于 0.05，故可以认为是平稳序列。纯随机检验结果（表 2）显示该序列为非白噪声序列，故可以使用 ARIMA 模型拟合该序列。

5.2 自相关图和偏相关图

下面考察该序列的自相关图和偏相关图的特征，给 ARIMA 模型定阶。

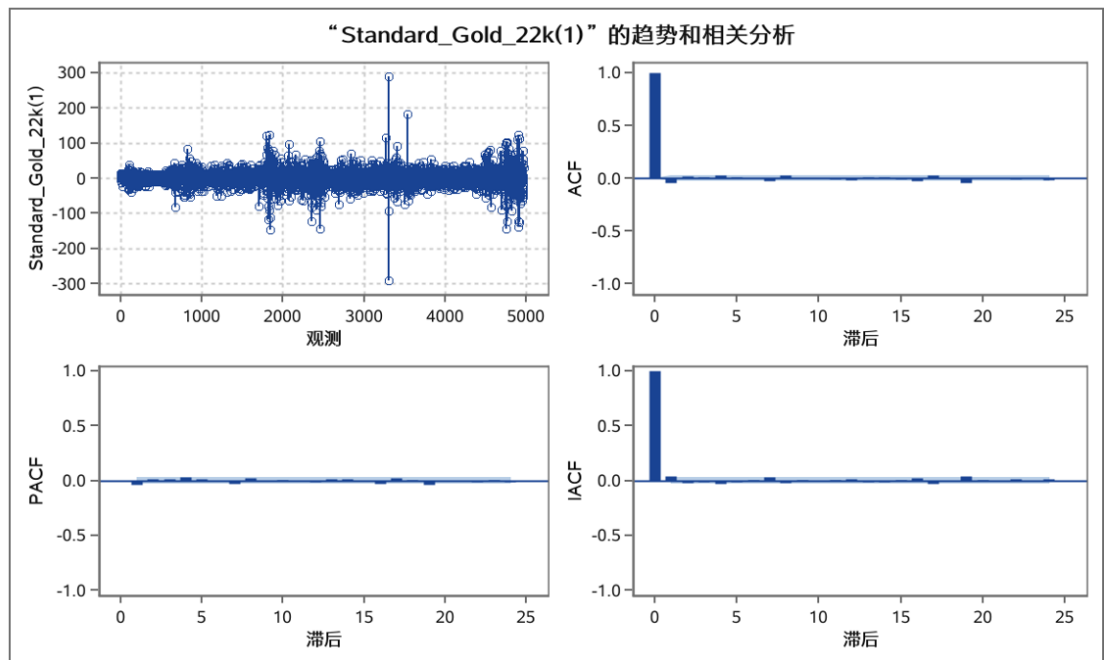


图 5 “Standard_Gold_22K（1）” 的趋势和相关系数图

由图 5 可知，原始序列进行一阶差分后 ACF、PACF 图均为不规则衰减拖尾，故本文可以采用 ARIMA 模型来进行预测。

表 3 最小信息准则结果

滞后	最小信息准则										
	MA 0	MA 1	MA 2	MA 3	MA 4	MA 5	MA 6	MA 7	MA 8	MA 9	MA 10
AR 0	6.016242	6.016421	6.017751	6.019281	6.020114	6.021748	6.023439	6.024268	6.025344	6.027041	6.028734
AR 1	6.016341	6.017964	6.019439	6.020922	6.021632	6.023116	6.024819	6.025587	6.026947	6.028659	6.030357
AR 2	6.017807	6.019512	6.020501	6.022188	6.022434	6.023612	6.025111	6.026114	6.027668	6.02927	6.030644
AR 3	6.01931	6.020991	6.022198	6.023778	6.024134	6.02528	6.02665	6.027826	6.029379	6.030982	6.032285
AR 4	6.020058	6.021638	6.022468	6.024165	6.025837	6.026971	6.028345	6.029486	6.031084	6.032692	6.033992
AR 5	6.021651	6.023073	6.023589	6.025271	6.026979	6.028683	6.030026	6.030842	6.032548	6.033967	6.035136
AR 6	6.023345	6.024778	6.025045	6.026601	6.02827	6.029968	6.031546	6.032532	6.034242	6.035516	6.036836
AR 7	6.024047	6.025427	6.026005	6.027716	6.029401	6.030863	6.032565	6.034231	6.035942	6.036953	6.038479
AR 8	6.025275	6.0269	6.027628	6.029333	6.031043	6.032572	6.034276	6.03594	6.037605	6.038665	6.040135
AR 9	6.026987	6.0286	6.029253	6.030961	6.032673	6.034036	6.035632	6.037023	6.038736	6.040293	6.041632
AR 10	6.028683	6.030296	6.030688	6.032303	6.034012	6.035247	6.036959	6.038571	6.040242	6.041637	6.043344

表 4 ARIMA(2,1,2)模型结果

条件最小二乘估计					
参数	估计	标准 误差	t 值	近似 Pr > t	滞后
MU	0.83941	0.28758	2.92	0.0035	0
MA1,1	0.87015	0.19095	4.56	<.0001	1
MA1,2	-0.58489	0.17549	-3.33	0.0009	2
AR1,1	0.83056	0.19657	4.23	<.0001	1
AR1,2	-0.54386	0.18242	-2.98	0.0029	2

根据最小信息准则表 3，得知 ARIMA(1,1,0)为最适宜模型，但在模型实际拟合过程中出现残差白噪音检验不通过的情况，最终确定最优模型为带常数项 ARIMA(2,1,2)模型，参数检验结果如表 4 所示模型拟合结果为：

$$\nabla x_t = 0.83941 + \frac{1+0.87015B-0.58489B^2}{1-0.83056B+0.54386B^2} \varepsilon_t$$

5.3 残差检验

表 5 残差的自相关系数检验

残差的自相关检查									
至滞后	卡方	自由度	Pr > 卡方	自相关					
6	2.38	2	0.3048	-0.001	0.009	-0.013	0.011	0.008	0.006
12	8.98	8	0.3441	-0.020	0.026	-0.005	-0.000	-0.008	-0.012
18	17.51	14	0.2301	0.013	0.013	-0.007	-0.025	0.026	0.005
24	26.49	20	0.1503	-0.038	-0.003	-0.001	-0.011	0.001	-0.014
30	34.98	26	0.1122	0.007	-0.021	-0.016	-0.026	0.009	-0.014
36	38.88	32	0.1874	0.003	-0.018	-0.011	0.010	-0.000	-0.016
42	46.47	38	0.1628	0.001	-0.023	0.011	0.021	-0.020	0.005
48	50.85	44	0.2219	-0.011	0.003	-0.016	0.019	-0.011	-0.000

表 5 展示了残差的自相关性检验结果。该图表中列出了不同滞后期下的卡方统计量、自由度、p 值以及相应的自相关系数。图表中卡方统计量随着滞后期的增加而增加，但都未达到显著水平。在所有滞后期下，p 值均显著大于 0.1。这表明在 95% 的置信水平下，我们不能拒绝残差是白噪声的假设，即残差没有显著的自相关性。

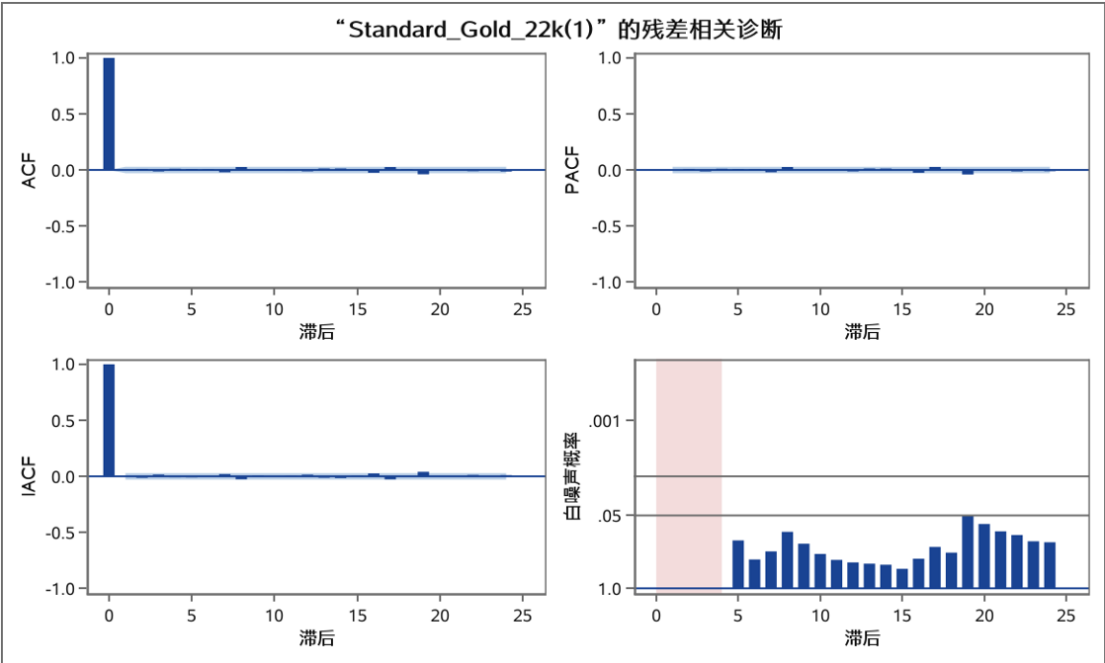


图 6 “Standard_Gold_22K（1）” 的残差相关诊断图

从图 6 中可以看到，ACF 在大部分滞后期内都接近于零，并且大部分点落在置信区间内。这表明残差没有显著的自相关性。PACF 在大部分滞后期内也

接近于零，并且大部分点落在置信区间内。图中显示标准化残差在零附近波动，没有明显的趋势或模式。

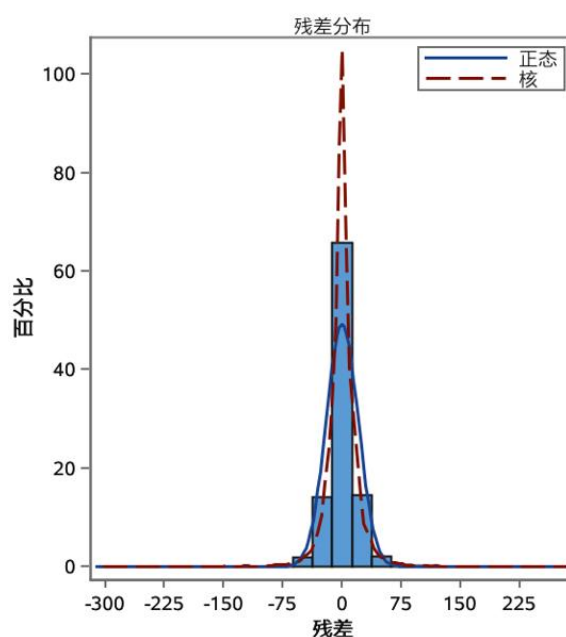


图 7 “Standard_Gold_22K (1)” 的残差正态诊断图

残差分布直方图显示了残差的频率分布情况。图 7 中蓝色实线代表正态分布，红色虚线表示核密度估计。可以看出，残差大致呈现正态分布，中心位置较为集中，两侧对称性良好，基本符合正态分布的特点。这表明残差没有明显的偏态或峰态，符合模型残差应为随机分布的假设。

综合以上分析，残差状态基本符合正态性假设，且没有显著的自相关性，表明模型拟合情况良好。残差的正态分布和随机性表明模型对数据的解释力较强，没有显著的系统性偏差，为模型的预测能力提供了良好的基础，保证了预测结果的可靠性和有效性。

5.4 模拟预测

采用 ARIMA(2,1,2)模型对 2020 年 10 月 10 日-2020 年 10 月 18 日印度黄金价格变化进行短期预测，预测结果如表 6 所示。

表 6 印度黄金价格预测表（2020 年 10 月 10 日-10 月 18 日）

变量 “Standard_Gold_22k” 的预测				
观测	预测	标准误差	95% 置信限	
4972	4880.5092	20.2325	4840.8542	4920.1643
4973	4882.7389	28.0524	4827.7572	4937.7206
4974	4885.4565	34.2192	4818.3882	4952.5248
4975	4887.0997	39.7207	4809.2486	4964.9508
4976	4887.5853	44.7239	4799.9280	4975.2425
4977	4887.6936	49.2250	4791.2144	4984.1728
4978	4888.1183	53.2700	4783.7110	4992.5256
4979	4889.0108	56.9673	4777.3570	5000.6646

根据表 6 和图 8 数据显示，从 2020 年 10 月 10 日到 10 月 18 日，印度黄金价格呈现逐渐上升的趋势。具体来说，黄金价格从 4880 元上涨到 4889 元，尽管涨幅不大，但持续的上升趋势显示出市场对黄金需求的增长和对未来价格上涨的预期。

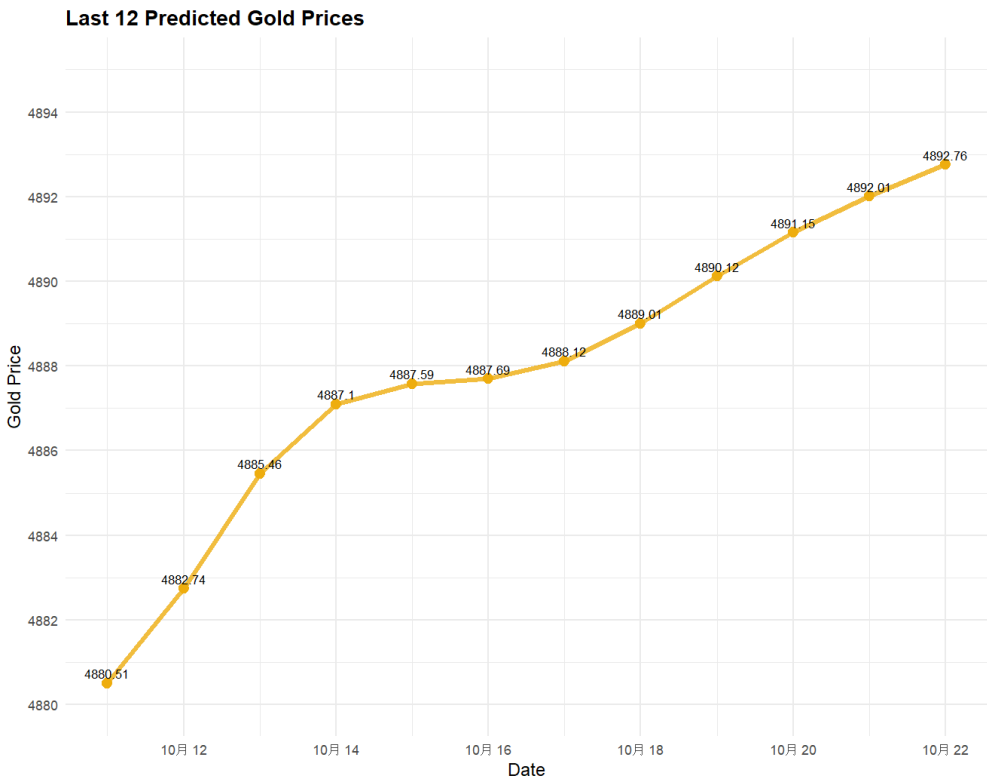


图 8 印度黄金价格预测图（2020 年 10 月 10 日-10 月 18 日）

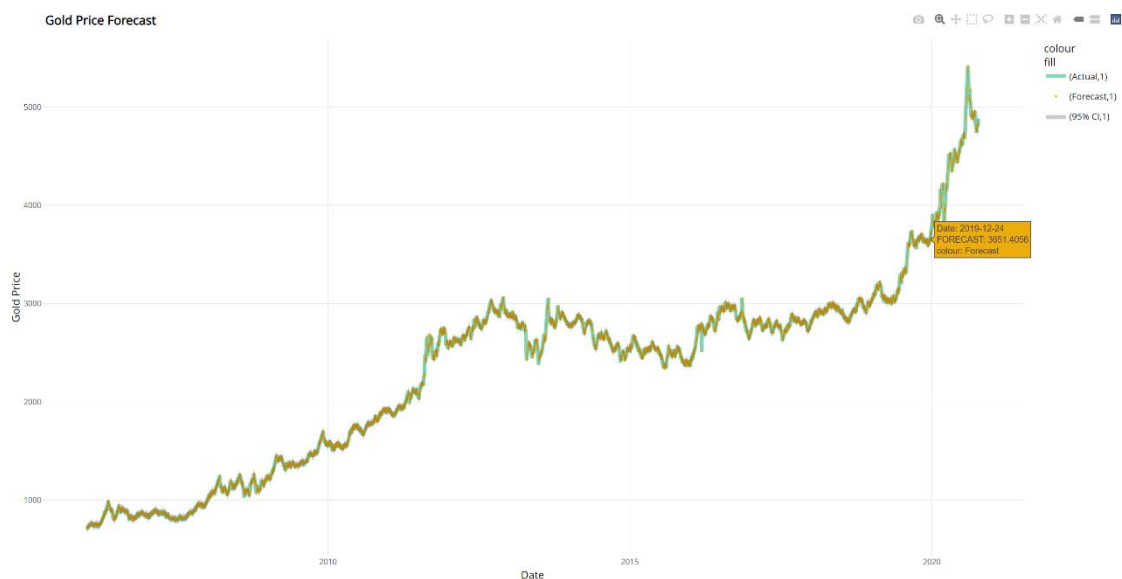


图 9 印度黄金价格整体预测图（2016 年 1 月 2 日-2020 年 10 月 18 日）

根据图 9 的数据显示，图中显示了印度黄金价格在所选时间段内的整体上升趋势。特别是在接近 2020 年的时间段内，黄金价格有显著的上升。从图中可以看出，预测的黄金价格走势与实际价格走势高度吻合，表明所使用的预测模型在这段时间内具有较好的拟合效果。虽然存在一些偏差，但整体趋势和价格波动基本被模型准确捕捉。

六、GM（1，1）模型构建

6.1 数据预处理

选取 2020 年 9 月 28 日-2020 年 10 月 10 日这 13 天的黄金价格作为拟合 GM 模型使用的数据，具体如下：

$$X^{(0)} = \{4740.4818, 4809.4815, 4869.4850, 4850.4821, 4860.4806, 4810.4850, 4881\}$$

根据式(4)可求出 1-AGO 序列，即：

$$X^{(1)} = \{4740, 9558, 14367, 19182, 24051, 28901, 33751, 38572, 43432, 48238, 53048, 57898, 62779\}$$

根据式（6）可求其紧邻均值生成序列，即：

$$Z^{(1)} = \begin{Bmatrix} -7149.0, -11962.5, -16774.5, -21616.5, -26476.0, -31326.0, \\ -36161.5, -41002.0, -45835.0, -50643.0, -55473.0, -60338.5 \end{Bmatrix}$$

由式（8）、（9）即可求出发展系数 $-a = -0.0005154462$ ，灰作用量 $b = 4819.197$ 。

6.2 模型的有效性检验

首先由式（15）可计算出原始数据的级比，如表 7 所示：

表 7 级比

K	2	3	4	5	6	7	8	9	10	11	12	13
级比	0.984	1.002	0.999	0.989	1.004	1.000	1.006	0.992	1.011	0.999	0.992	0.994

由于我们已经求得发展系数 $-a = -0.0005154462$ ，求出级比偏差，如表 8 所示：

表 8 级比偏差

K	2	3	4	5	6	7	8	9	10	11	12	13
级比偏差	0.015	-0.003	0	0.01	-0.005	-0.001	-0.007	0.007	-0.012	0	0.007	0.005

有了级比偏差之后，求出级比偏差为：

$$\eta = \sum_{k=2}^n \frac{\eta(k)}{n-1} = 0.001333$$

由于级比偏差 $\eta < 0.2$ ，故该模型可用于检测。

6.3 模型预测

最终 GM 模型所预测的 2020 年 9 月 28 至 10 月 10 日的结果如下：

$$\hat{X}^{(0)} = \left\{ \begin{array}{l} 4740, 4822.9, 4825.4, 4827.9, 4830.3, 4832.3, 4835.3, \\ 4837.8, 4840.3, 4842.8, 4845.3, 4847.8, 4850.3 \end{array} \right\}$$

计算得出原数据和拟合数据的相对残差，如表 9 所示：

表 9 相对残差表

K	2	3	4	5	6	7	8	9	10	11	12	13
相对残差	0.001	0.003	0.003	0.008	0.004	0.003	0.004	0.004	0.008	0.007	0.001	0.006

相对残差是一个用来衡量模型预测值与实际值之间差异的指标。在 GM（灰色模型）拟合中，相对残差的计算方法通常为：相对残差越小，表示模型预测的结果越接近实际值，说明模型的拟合效果越好。由表 9 可知，模型拟合效果较好，具体如图 10 所示。

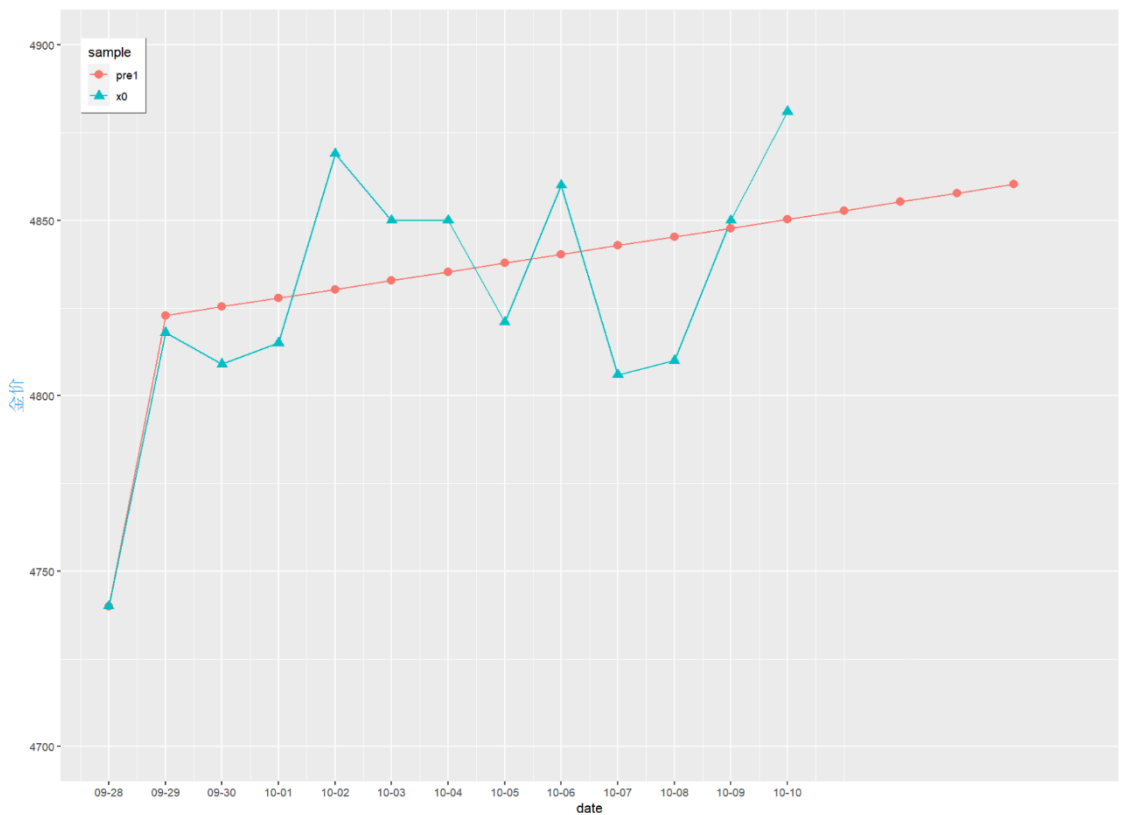


图 10 GM 模型预测值与真实值的折现对比图

针对金价数据的 GM 模型拟合结果，由图 10 发现，尽管相对残差很小，但从图表上来看模型的拟合效果偏差较大，这可能是因为金价本身的价格较高。即使预测值和实际值之间的绝对偏差在 10 或 20 左右，相对误差依然很小。然而，当将这些误差绘制成图表时，误差的累积效应和可视化效果使得这些差异变得明显。

七、模型比较及结论建议

7.1 模型比较

两类模型的预测结果如表 10 所示：

表 10 两种模型的金价预测真实值与预测拟合值对比表

date	Standard_Gold_22k	ARIMA	GM
2020-09-28	4740	4781.43	4740
2020-09-29	4818	4743.86	4822.884
2020-09-30	4809	4816.39	4825.37
2020-10-01	4815	4809.5	4827.858
2020-10-02	4869	4816.37	4830.347
2020-10-03	4850	4868.6	4832.838
2020-10-04	4850	4852.42	4835.329
2020-10-05	4821	4852.16	4837.822
2020-10-06	4860	4823.21	4840.317
2020-10-07	4806	4858.52	4842.812
2020-10-08	4810	4807.76	4845.309
2020-10-09	4850	4810.62	4847.807
2020-10-10	4881	4848.69	4850.306

分别计算出 2 种模型实际值与拟合值的均方误差（MSE）、均方根误差（RMSE）、平均绝对误差（MAE）、平均绝对百分比误差（MAPE），通过比较上述预测评价指标可知，其中拟合精度与预测精度较好的为 GM（1，1）模型，结果见表 11。

表 11 两模型的预测效果评价指标

预测评价指标	ARIMA (2, 1, 2) 模型	GM(1, 1) 模型
均方误差 (MSE)	1392.486	513.8589
均方根误差 (RMSE)	37.31603	22.66846
平均绝对误差 (MAE)	30.50077	18.93162
平均绝对百分比误差(MARE)	0.63%	0.39%

ARIMA 模型的误差指标显示，预测误差相对较大，特别是 RMSE 和 MAE 比较高，表明模型的预测值与实际观测值之间存在较大的偏差。相比之下，GM 模型的误差指标明显优于 ARIMA 模型。GM 模型的 RMSE 和 MAE 较低，表明其预测值与实际观测值之间的偏差较小，预测精度较高。但从两模型预测拟

合图中发现，ARIMA 模型虽然误差比较大，主要是由于其预测具有滞后性（见图 11）。

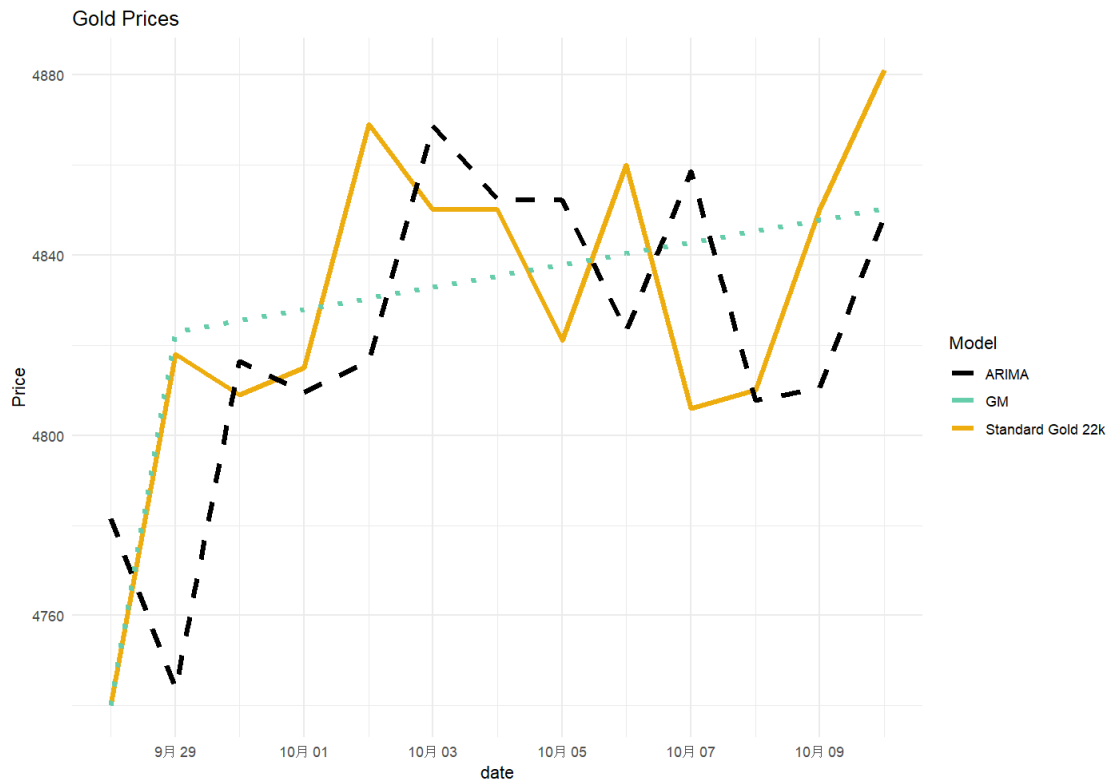


图 11 两种模型的金价预测真实值与预测拟合值对比图

在时间序列预测中，ARIMA（自回归积分移动平均）模型和灰色系统模型（GM 模型）各有其优缺点，导致它们在预测金价时表现出不同的特点和误差模式。

ARIMA 模型在金价预测中常表现出一定的滞后效应，这意味着其预测结果可能在反映市场实际变化时稍显迟缓。然而，ARIMA 模型对于长期趋势和稳定市场情况下的周期性波动预测较为准确，这使其在长期投资决策和风险管理中仍具有重要价值。

相对而言，GM 模型在捕捉整体价格趋势和短期波动性方面表现较为优越。其能够有效处理非线性和高不确定性数据，适用于复杂市场环境下的预测需求。然而，GM 模型在极端市场波动和快速价格变动的情况下，可能出现预测误差增加的情况，这是其在短期预测精度上的一种局限性。

7.2 结论及改进建议

综合研究结果显示，选择适当的预测模型应根据具体市场环境和数据特性而定。建议在实际应用中，可以考虑采用模型集成的方法，利用 ARIMA 模型和 GM 模型各自的优势进行组合预测，以提高预测精度和鲁棒性。此外，对 ARIMA 模型进行参数调优和 GM 模型的结构优化，以适应不同市场条件和数据变化，是提高预测准确性的关键步骤。

进一步的研究可以探索更复杂的混合模型或者引入其他高级统计方法和机器学习技术，以进一步提升金价预测的效果。这些结论和建议不仅对金融市场参与者的投资决策具有指导意义，还为未来相关研究提供了有价值的参考和方向。

综上所述，ARIMA 模型和 GM 模型在金价预测中的表现差异主要源自它们对数据特性和波动模式的不同处理方式。理解每种模型的优势和局限性，可以帮助选择合适的模型来满足具体的预测需求。进一步的研究可以探索更复杂的混合模型或者引入其他高级统计方法和机器学习技术，以进一步提升金价预测的效果。这些结论和建议不仅对金融市场参与者的投资决策具有指导意义，还为未来相关研究提供了有价值的参考和方向。

参考文献

- [1] 高菲,顾炜宇. “一般商品-保值手段” 双重属性视角下的国际黄金价格影响因素研究[J].中国软科学,2018,(05):160-170.
- [2] 刘丞宇.基于 LASSO-QRNN 的黄金期货价格预测研究[D].中南大学,2023.
- [3] 王庆荣,张慈仁.基于新陈代谢 GM(1,1)++的疫情应急物资需求量预测研究[J].商丘师范学院学报,2024,40(06):1-5.
- [4] 杨楠,柳预才.基于分形分析的国际金价波动长记忆性识别与预测研究[J].数理统计与管理,2013,32(05):931-940.
- [5] 张明玉,周钰博,吴建军,等.ARIMA 模型与 GM(1,1)模型对丙型病毒性肝炎发病的预测效果比较研究[J/OL].预防医学情报杂志,1-6[2024-06-20].
- [6] 邹子昂,彭啸帆,皮俊.国际黄金现货市场的避险能力研究——基于 DCC-GARCH 模型[J].财经理论与实践,2018,39(06):44-50.
- [7] Dadabada, A., & Vadlamani, A. (2020). Performance evaluation of particle swarm optimized quantile regression neural network for non-linear gold price forecasting. Expert Systems with Applications, 146, 113180.
- [8] Gao, F., & Gu, W. (2018). Empirical analysis of international gold price influencing factors using SVAR model. Journal of Finance and Economics, 14(4), 112-121.
- [9] Liu, Y. (2023). High-precision forecasting of gold futures prices using LASSO-QRNN model. International Journal of Financial Studies, 11(2), 231-245.
- [10] Zakaria, R., Ismail, Z., & Isa, Z. (2018). Evaluation of ARIMA model with improved multi-layer perceptron neural network and whale optimization algorithm for gold price prediction. Journal of Applied Economic Sciences, 13(3), 748-756.

附录一：原始及预测后数据⁵

观测	Date	Standard _Gold_2 2k	FORECAS T	STD	L95	U95	RESIDUAL
1	2006/1/2	711					
2	2006/1/3	713	711.8264079	20.1988352	672.2374184	751.4153974	1.173592121
3	2006/1/4	726	713.7814139	20.1988352	674.1924244	753.3704034	12.21858613
4	2006/1/5	725	726.3664632	20.1988352	686.7774737	765.9554527	-1.366463178
5	2006/1/6	719	725.9999895	20.1988352	686.411	765.588979	-6.999989533
6	2006/1/7	729	720.4480118	20.1988352	680.8590223	760.0370013	8.551988172
7	2006/1/9	732	729.6469499	20.1988352	690.0579604	769.2359394	2.353050105
8	2006/1/10	732	732.5704251	20.1988352	692.9814356	772.1594147	-0.570425139
9	2006/1/11	730	732.8274905	20.1988352	693.238501	772.4164801	-2.827490547
10	2006/1/12	731	731.0584343	20.1988352	691.4694447	770.6474238	-0.058434251
11	2006/1/13	732	731.9117157	20.1988352	692.3227262	771.5007052	0.088284322
12	2006/1/16	743	732.7609151	20.1988352	693.1719256	772.3499046	10.23908492
13	2006/1/17	752	743.333752	20.1988352	703.7447625	782.9227416	8.666247952
14	2006/1/18	740	752.5211396	20.1988352	712.9321501	792.1101291	-12.52113957
15	2006/1/19	744	741.6715377	20.1988352	702.0825482	781.2605272	2.328462311
16	2006/1/20	750	745.1272954	20.1988352	705.5383059	784.7162849	4.872704585
17	2006/1/21	749	750.5055825	20.1988352	710.916593	790.094572	-1.505582483
18	2006/1/23	749	749.6357556	20.1988352	710.046766	789.2247451	-0.635755557
19	2006/1/24	750	749.8126739	20.1988352	710.2236844	789.4016634	0.187326094
20	2006/1/25	752	750.8907077	20.1988352	711.3017182	790.4796972	1.109292277
21	2006/1/27	754	752.8532576	20.1988352	713.2642681	792.4422471	1.146742386
22	2006/1/28	753	754.8124435	20.1988352	715.2234539	794.401433	-1.812443462
23	2006/1/30	755	753.9142402	20.1988352	714.3252507	793.5032297	1.085759806
24	2006/1/31	761	755.7950361	20.1988352	716.2060466	795.3840256	5.204963933
25	2006/2/1	758	761.5935159	20.1988352	722.0045264	801.1825055	-3.593515948
26	2006/2/2	763	758.9915895	20.1988352	719.4026	798.580579	4.008410487
27	2006/2/3	768	763.8019643	20.1988352	724.2129748	803.3909538	4.198035677
28	2006/2/4	767	768.7096402	20.1988352	729.1206507	808.2986297	-1.709640212
29	2006/2/6	769	767.9734548	20.1988352	728.3844652	807.5624443	1.026545249
30	2006/2/7	769	769.9096043	20.1988352	730.3206148	809.4985938	-0.909604281
31	2006/2/8	744	769.8885257	20.1988352	730.2995362	809.4775152	-25.88852571
32	2006/2/9	755	745.7880941	20.1988352	706.1991046	785.3770836	9.211905871
33	2006/2/10	754	755.2244869	20.1988352	715.6354974	794.8134764	-1.224486921
34	2006/2/11	745	754.1847327	20.1988352	714.5957432	793.7737222	-9.184732687
35	2006/2/13	739	745.9310528	20.1988352	706.3420633	785.5200423	-6.931052832

⁵ 由于数据共 4971 条，为节省篇幅，此处只节选前 35 条数据，其中表中第 1，2 列为原始数据，剩余为 ARIMA 模型预测后的数据。

附录二：代码

1. 可视化（R 语言）

```
# 加载必要的包
library(dplyr)
library(ggplot2)
library(forecast)
library(greybox)
library(readr)
library(plotly)
library(lubridate)
library(tseries)
# 加载数据
file_path <- "C:\\Users\\立早\\Desktop\\gold_rate_history.csv"
df <- read_csv(file_path)
# 转换日期格式
df$Date <- as.Date(df$Date, format = "%m/%d/%Y")
# 筛选所需的列
df_selected <- df %>% select(Date, `Pure Gold (24 k)`, `Standard Gold (22 K)`)
# 查看数据的基本统计信息
summary(df_selected)
# 画出金价的时间序列图
p_time <- ggplot(df_selected, aes(x = Date)) +
  geom_line(aes(y = `Pure Gold (24 k)`, color = "Pure Gold (24 k)")) +
  geom_line(aes(y = `Standard Gold (22 K)`, color = "Standard Gold (22 K)")) +
  labs(title = "Gold Prices Over Time", x = "Date", y = "Gold Price") +
  scale_color_manual(name = "Gold Type", values = c("Pure Gold (24 k)" =
"darkgoldenrod2", "Standard Gold (22 K)" = "mediumaquamarine")) +
  theme(legend.title = element_blank())
p_time
ggplotly(p_time)
# 画出金价的分布图
p_distribution <- ggplot(df_selected) +
  geom_histogram(aes(x = `Pure Gold (24 k)`, fill = "Pure Gold (24 k)"), binwidth =
50, alpha = 0.7, position = "identity") +
  geom_histogram(aes(x = `Standard Gold (22 K)`, fill = "Standard Gold (22 K)"),
binwidth = 50, alpha = 0.7, position = "identity") +
  labs(title = "Distribution of Gold Prices", x = "Gold Price", y = "Frequency") +
  scale_fill_manual(name = "Gold Type", values = c("Pure Gold (24 k)" =
"darkgoldenrod2", "Standard Gold (22 K)" = "mediumaquamarine")) +
  theme(legend.title = element_blank())
p_distribution
```

```

ggplotly(p_distribution)
# 转换为时间序列对象
ts_pure_gold <- ts(df_selected$`Pure Gold (24 k)`, start =
c(year(min(df_selected$Date)), month(min(df_selected$Date))), frequency = 12)
ts_standard_gold <- ts(df_selected$`Standard Gold (22 K)`, start =
c(year(min(df_selected$Date)), month(min(df_selected$Date))), frequency = 12)
# 绘制标准金的原始时间序列图
ggplot(df_selected, aes(x = Date, y = `Standard Gold (22 K)`) +
  geom_line(color = "mediumaquamarine") +
  labs(title = "Original Time Series of Standard Gold (22 K)", x = "Date", y =
"Standard Gold (22 K)"))
# 计算一阶差分
df_selected <- df_selected %>%
  arrange(Date) %>%
  mutate(Diff_Standard_Gold = c(NA, diff(`Standard Gold (22 K)`)))
# 绘制一阶差分后的标准金时间序列图
p_diff <- ggplot(df_selected, aes(x = Date, y = Diff_Standard_Gold)) +
  geom_line(color = "mediumaquamarine") +
  labs(title = "First Difference of Standard Gold (22 K)", x = "Date", y = "Difference
of Standard Gold (22 K)"))
p_diff
ggplotly(p_diff)
# 保留原始日期
df_ts_standard_gold <- df_selected %>% select(Date, `Standard Gold (22 K)`)
# 导出时间序列数据
write.csv(df_ts_standard_gold, file = "C:\\Users\\立早
\\Desktop\\ts_standard_gold.csv", row.names = FALSE)

```

2. ARIMA 模型拟合（SAS 语言）

```

/* 删除现有的 WORK.IMPORT 表 */
proc datasets lib=work nolist;
  delete IMPORT;
quit;
/* 定义文件路径 */
FILENAME REFFILE '/home/u63877733/sasuser.v94/mylib/ts_standard_gold.csv';
/* 新建一个逻辑库 mylib, 并保存数据 */
libname mylib '/home/u63877733/sasuser.v94/mylib';
/* 导入 CSV 文件到 SAS 数据集中 */
proc import datafile=REFFILE
  dbms=csv
  out=work.IMPORT
  replace;
  getnames=yes;

```

```

run;
/* 检查导入数据的变量名 */
proc contents data=work.IMPORT; run;
/* 保存导入的数据到 mylib 库 */
data mylib.GoldPrices;
    set work.IMPORT;
run;
/* 检查导入的数据 */
proc print data=mylib.GoldPrices (obs=10); run;
/* 确保变量名正确 */
data mylib.GoldPrices;
    set mylib.GoldPrices;
    rename Standard_Gold_22_K=Standard_Gold_22k;
run;
/* 计算一阶差分 */
data mylib.GoldPrices_diff;
    set mylib.GoldPrices;
    if _N_ > 1 then Diff_Standard_Gold = dif(Standard_Gold_22k);
run;
/* ODS PDF 和 Word 输出设置 */
ods pdf file="/home/u63877733/sasuser.v94/mylib/gold_price_analysis.pdf"
style=journal;
ods word file="/home/u63877733/sasuser.v94/mylib/gold_price_analysis.docx";
/* 调整绘图区域以适应较多的数据点 */
ods graphics / reset=all height=6in width=10in;
/* 绘制原始金价数据的时序图 */
proc sgplot data=mylib.GoldPrices;
    series x=Date y=Standard_Gold_22k / markers lineattrs=(color=red);
    xaxis label="Date";
    yaxis label="Standard Gold (22 K)";
run;
/* 绘制差分后的时序图 */
proc sgplot data=mylib.GoldPrices_diff;
    where not missing(Diff_Standard_Gold);
    series x=Date y=Diff_Standard_Gold / markers lineattrs=(color=red);
    xaxis label="Date";
    yaxis label="Difference of Standard Gold (22 K)";
run;
/* 重新检查原始数据的平稳性和纯随机性 */
proc arima data=mylib.GoldPrices;
    identify var=Standard_Gold_22k(1) nlag=24 stationarity=(adf) minic p=(0:10)
q=(0:10);
run;
/* 预测未来 5 天的金价 */

```

```

proc arima data=mylib.GoldPrices;
    identify var=Standard_Gold_22k(1) nlag=24;
    estimate p=2 q=2;
    forecast lead=5 id=Date out=results;
run;
/* 检查预测结果 */
proc print data=results; run;
/* 绘制预测结果 */
proc sgplot data=results;
    series x=Date y=Standard_Gold_22k / markers lineattrs=(color=black);
    series x=Date y=Forecast / markers lineattrs=(color=red);
    band x=Date lower=L95 upper=U95 / transparency=0.5 fillattrs=(color=green);
    xaxis label="Date";
    yaxis label="Standard Gold (22 K)";
run;
/* 导出预测结果到 Excel 文件 */
proc export data=results
    outfile="/home/u63877733/sasuser.v94/mylib/predicted_gold_prices.xlsx"
    dbms=xlsx
    replace;
    sheet="Forecast";
run;
/* 结束 ODS PDF 和 Word 输出 */
ods pdf close;
ods word close;

```

3. ARIMA 模型的预测图（R 语言）

```

# 加载必要的包
library(readxl)
library(ggplot2)
library(dplyr)
library(plotly)
# 读取 Excel 文件
file_path <- "C:\\Users\\立早\\Downloads\\predicted_gold_prices.xlsx"
gold_data <- read_excel(file_path)
# 检查数据结构，确保包含'日期', '真实值', '预测值', '置信下限', '置信上限'
print(colnames(gold_data))
# 转换日期格式（如果需要）
gold_data$Date <- as.Date(gold_data$Date, format = "%Y-%m-%d") # 格式根据实际情况调整
# 绘图代码
p <- ggplot(gold_data, aes(x = Date)) +

```

```

    geom_line(aes(y = Standard_Gold_22k, color = "Actual"), size = 1.5, alpha = 0.8)
+ # 绘制实际数据
    geom_point(aes(y = FORECAST, color = "Forecast"), size = 0.25, alpha = 0.8) +
# 绘制预测数据
    geom_ribbon(aes(ymin = L95, ymax = U95, fill = "95% CI"), alpha = 0.2) + # 绘制置信区间
    labs(
      title = "Gold Price Forecast",
      x = "Date",
      y = "Gold Price"
    ) +
    scale_color_manual(values = c("Actual" = "mediumaquamarine", "Forecast" =
"darkgoldenrod2")) +
    scale_fill_manual(values = c("95% CI" = "black")) +
    theme_minimal() +
    theme(
      plot.title = element_text(size = 14, face = "bold"),
      axis.title = element_text(size = 12),
      legend.position = "bottom",
      legend.title = element_text(size = 12, face = "bold"),
      legend.text = element_text(size = 10)
    ) +
    guides(
      color = guide_legend(override.aes = list(linetype = c("solid", "blank"),
                                                    size = c(1.5, 3))),
      fill = guide_legend(override.aes = list(alpha = 0.2))
    )
# 使用 plotly 转换为交互式图表
p_interactive <- ggplotly(p, tooltip = c("x", "y", "color"))
# 显示交互式图表
p_interactive

```

4. GM 模型 (R 语言)

```

# 加载必要的包
library(dplyr)
library(ggplot2)
library(tidyr)
library(DT)
# 数据
x0 <- c(4740,
        4818,
        4809,
        4815,

```

```

4869,
4850,
4850,
4821,
4860,
4806,
4810,
4850,
4881)
n <- length(x0)
# 级比检验
test <- function(x0) {
  n <- length(x0)
  lamda <- x0[1:(n-1)] / x0[2:n]
  min_lamda <- min(lamda)
  max_lamda <- max(lamda)
  if (exp(-2 / (n + 1)) < min_lamda && max_lamda < exp(2 / (n + 1))) {
    print("可直接预测")
  } else {
    print("需做变换")
  }
}
test(x0)
# 级比检验
test <- function(x0) {
  n <- length(x0)
  lamda <- x0[1:(n-1)] / x0[2:n]
  min_lamda <- min(lamda)
  max_lamda <- max(lamda)

  if (exp(-2 / (n + 1)) < min_lamda && max_lamda < exp(2 / (n + 1))) {
    result <- "可直接预测"
  } else {
    result <- "需做变换"
  }
  return(list(result = result, lamda = lamda, min_lamda = min_lamda, max_lamda =
max_lamda))
}
result <- test(x0)
print(result)
# 输出表格形式的级比
lamda_values <- result$lamda
k_values <- 1:length(lamda_values)
data.frame(级比 = lamda_values)

```

```

# 累加数据
x1 <- cumsum(x0)
print(x1)
# 均值数据
z1 <- c()
for (i in 2:n) {
  z1[i-1] <- -(0.5 * x1[i] + 0.5 * x1[i-1])
}
print(z1)
# 构造矩阵 B
one <- rep(1, n-1)
B <- cbind(z1, one)
print(B)
# 构造 Y
Y <- x0[-1]
print(Y)
# 求解参数 a 和 b
u <- solve(crossprod(B, B)) %*% crossprod(B, Y)
a <- u[1]
b <- u[2]
print(a)
print(b)
# 模型求解
pre <- c()
for (i in 0:(n + 3)) { # 改为预测未来 5 个月
  pre[i+1] <- (x0[1] - b / a) * exp(-a * i) + b / a
}
print(pre)
# 还原预测值
pre1 <- c()
pre1[1] <- x0[1]
for (i in 2:length(pre)) {
  pre1[i] <- pre[i] - pre[i-1]
}
print(pre1)

# 确保长度一致
time <- seq(1, 18) # 时间扩展到预测结束
x0_extended <- c(x0, rep(NA, length(pre1) - length(x0)+1))
pre1 <- pre1[1:length(time)] # 确保预测值和时间长度一致
# 确保所有数据列的长度一致
print(length(time)) # 18
print(length(pre1)) # 18
print(length(x0_extended)) # 18

```



```

# 准备数据用于绘图
data <- data.frame(time, pre1, x0 = x0_extended)
mydata <- pivot_longer(data, cols = c(pre1, x0), names_to = "sample", values_to =
"value")
# 将时间范围调整为 9 月 28 日到 10 月 10 日
date_range <- seq(as.Date("2020-09-28"), as.Date("2020-10-10"), by = "day")
# 绘制预测值和原始值的图
ggplot(mydata, aes(x = time, y = value, group = sample, shape = sample, col =
sample)) +
  geom_line() +
  geom_point(size = 3) +
  xlab("date") +
  ylab("金价") +
  scale_x_continuous(breaks = seq(1, 13), labels = format(date_range, "%m-%d")) +
# 修改 x 轴标签
  scale_y_continuous(limits = c(4700, 4900)) +
  theme(legend.position = c(0.05, 0.915),
        legend.box.background = element_rect(color = "black"),
        axis.title.y = element_text(size = 14, color = 4))
# 进一步检验
day <- seq(1, 13) # 修改年份序列, 使其长度与 x0 匹配
predict <- round(pre1[1:length(x0)], 3) # 修改预测值, 使其长度与 x0 匹配
res <- round(x0 - predict, 4) # 计算残差, 保留四位小数
error <- round(abs(res / x0), 4) # 计算相对误差, 保留四位小数
lamda <- round(x0[1:(n-1)] / x0[2:n], 3) # 计算级比偏差
perror <- round(1 - (1 - 0.5 * a) / (1 + 0.5 * a) * lamda, 3)
perror <- append(NA, perror)
# 确保所有列长度一致
print(length(day)) # 13
print(length(predict))
print(length(x0))
print(length(error))
print(length(perror))
print(length(res))
# 创建最终数据框并显示
blank <- data.frame(day = day, x0, predict, res, error, perror)
colnames(blank) <- c("date", "原始值", "预测值", "残差", "相对误差", "级比偏差")
# 将年份修改为 2020 年 9 月 28 日到 2020 年 10 月 10 日
blank$date <- seq(as.Date("2020-09-28"), as.Date("2020-10-10"), by = "day")
datatable(blank)
# 将 blank 导出为 CSV 文件
write.csv(blank, "C:/Users/立早/Downloads/predicted_gold_prices_blank.csv",
row.names = TRUE)

```

5. 模型对比图 (R 语言)

```
# 读取数据
data <- read.csv("C:/Users/立早/Desktop/monthlyaverage_gold_prices.csv",
colClasses = c("Month" = "character"))
# 加载必要的包
library(ggplot2)
# 读取数据，并将日期列转换为日期类型
data <- read.csv("C:/Users/立早/Desktop/monthlyaverage_gold_prices.csv",
colClasses = c("date" = "Date"))
# 绘制图表
ggplot(data, aes(x = date)) +
  geom_line(aes(y = Standard_Gold_22k, color = "Standard Gold 22k"), size = 1.5,
group = 1) +
  geom_line(aes(y = ARIMA, color = "ARIMA"), linetype = "dashed", size = 1.5,
group = 1) +
  geom_line(aes(y = GM, color = "GM"), linetype = "dotted", size = 1.5, group = 1)
+
  labs(title = "Gold Prices",
        x = "date",
        y = "Price",
        color = "Model") +
  scale_color_manual(values = c("Standard Gold 22k" = "darkgoldenrod2",
"ARIMA" = "black", "GM" = "mediumaquamarine")) +
  theme_minimal()

# 查看数据的结构和前几行
str(data)
head(data)
```

6. 模型对比指标计算 (R 语言)

```
# 原始数据
data <- data.frame(
  date = as.Date(c("2020-09-28", "2020-09-29", "2020-09-30", "2020-10-01", "2020-
10-02", "2020-10-03",
                    "2020-10-04", "2020-10-05", "2020-10-06", "2020-10-07",
"2020-10-08", "2020-10-09",
                    "2020-10-10")),
  Standard_Gold_22k = c(4740, 4818, 4809, 4815, 4869, 4850, 4850, 4821, 4860,
4806, 4810, 4850, 4881),
  ARIMA = c(4781.43, 4743.86, 4816.39, 4809.5, 4816.37, 4868.6, 4852.42,
4852.16, 4823.21, 4858.52,
            4807.76, 4810.62, 4848.69),
```

```

GM = c(4740, 4822.884, 4825.37, 4827.858, 4830.347, 4832.838, 4835.329,
4837.822, 4840.317, 4842.812,
4845.309, 4847.807, 4850.306)
)
install.packages("Metrics")
# 计算各个模型的误差指标
library(Metrics)
# 均方误差 (MSE)
mse_arima <- mse(data$ARIMA, data$Standard_Gold_22k)
mse_gm <- mse(data$GM, data$Standard_Gold_22k)
# 均方根误差 (RMSE)
rmse_arima <- rmse(data$ARIMA, data$Standard_Gold_22k)
rmse_gm <- rmse(data$GM, data$Standard_Gold_22k)
# 平均绝对误差 (MAE)
mae_arima <- mae(data$ARIMA, data$Standard_Gold_22k)
mae_gm <- mae(data$GM, data$Standard_Gold_22k)
# 平均绝对百分比误差 (MAPE)
mape_arima <- mape(data$ARIMA, data$Standard_Gold_22k)
mape_gm <- mape(data$GM, data$Standard_Gold_22k)
# 打印结果
cat("ARIMA 模型的误差指标: \n")
cat("MSE:", mse_arima, "\n")
cat("RMSE:", rmse_arima, "\n")
cat("MAE:", mae_arima, "\n")
cat("MAPE:", mape_arima, "\n\n")
cat("GM 模型的误差指标: \n")
cat("MSE:", mse_gm, "\n")
cat("RMSE:", rmse_gm, "\n")
cat("MAE:", mae_gm, "\n")
cat("MAPE:", mape_gm, "\n")

```