

基于自回归积分滑动平均模型的猪肉价格分析研究

余浩琰，水含瑾，虞薇男

信息工程学院 22 移动应用开发专 4，22 移动应用开发专 4，22 移动应用开发专 5

摘要：

最近几年，国内生猪市场价格大幅波动，在人工成本、饲料原料价格上涨的背景下，生猪产消环节严重分离，不仅放大了市场风险和交易成本，也造成市场价格的大幅波动以及生产者及消费者福利的下降，由此，在生产端和消费端同时产生了对生猪养殖业高质量发展的现实需求。

通过建立皮尔逊积矩相关系数和斯皮尔曼等级相关系数的平均值模型，得出影响猪肉(去骨统肉)价格波动的关键性因素有活猪价格和仔猪价格。

通过建立自回归积分滑动平均模型进行猪肉价格随月份变化的变化趋势分析研究，得出猪肉(去骨统肉)价格随月份的变化趋势大致呈年中价格低，年底价格高，上半年降价，下半年涨价的趋势。

关键词：猪肉价格；分析研究；皮尔逊相关系数；斯皮尔曼相关系数；自回归积分滑动平均模型

一、 问题重述

（一） 问题背景

中国是全球最大的生猪生产国，中国猪肉产量接近全球猪肉产量的一半，2022 年，全球猪肉产量约为 1.25 亿吨，中国猪肉产量为 5541 万吨，中国猪肉产量在全球猪肉产量中所占的比重约为 44.47%。近年来，随着生活水平的提高，吃的安全、吃得健康逐渐成为主流，消费者对肉类产品的要求逐渐向高蛋白、低脂肪的方向转变，禽肉、水产品在中国肉类消费结构中的比重逐渐增加，但猪肉依然是中国居民第一大肉类消费产品。据相关学者考证研究，明代初年猪肉就成为了中国人的首选肉食，2022 年，全国人均肉类消费量为 34.6 公斤，其中，人均猪肉消费量为 26.9 公斤。最近几年，国内生猪市场价格大幅波动，在人工成本、饲料原料价格上涨的背景下，生猪产消环节严重分离，不仅放大了市场风险和交易成本，也造成市场价格的大幅波动以及生产者及消费者福利的下降，由此，在生产端和消费端同时产生了对生猪养殖业高质量发展的现实需求。

（二） 题目重述

猪肉是我国保供稳价的重要商品，因此前瞻性分析价格波动与趋势变化就很有必要。附件中的数据是我国 2010 年 1 月至 2023 年 3 月猪肉(去骨统肉) 价格(单位：元/公斤) 以及各影响因素的月度数据。请大家根据附件中给出的数据以及自己查询所获得的相关数据解决以下问题：

问题一：讨论猪肉(去骨统肉)价格与各因素之间的相关性，找出影响猪肉(去骨统肉)价格波动的关键性因素。

问题二：请建立数学模型描述猪肉(去骨统肉)价格的变化趋势。

二、 问题分析

（一） 问题一的分析

要讨论猪肉价格与各因素之间的相关性，可以利用给定的数据进行统计分析和相关性计算。附件中提供了猪肉价格以及各影响因素的月度数据，我们可以使用相关系数来衡量这些因素与猪肉价格之间的线性关系。

首先，将猪肉价格和各因素的数据进行对应匹配，然后计算它们之间的相关系数。常用的相关系数，包括皮尔逊积矩相关系数和斯皮尔曼等级相关系数。

皮尔逊相关系数衡量的是两个变量之间的线性关系，取值范围在-1 到 1 之间。相关系数为正表示正相关，为负表示负相关，接近 0 表示无线性关系。斯皮尔曼相关系数则衡量的是两个变量之间的单调关系，不要求线性关系，取值范围也在-1 到 1 之间。

通过计算相关系数，可以找出与猪肉价格波动密切相关的关键性因素^[1]。其中，相关系数的绝对值越大，表示其相关性越强。

（二） 问题二的分析

要建立猪肉价格的变化趋势数学模型，可以考虑使用时间序列分析方法^[2]。时间序列分析是研究随时间变化的数据的统计方法，可用于预测未来的趋势和周期性。

常用的时间序列模型包括移动平均模型（MA）、自回归模型（AR）、自回归滑动平均模型（ARMA）、季节性模型（SARIMA）等。

建立数学模型时，可以根据数据的特点和趋势选择适合的模型^[3]。例如，如果数据呈现明显的季节性变化^[4]，可以考虑使用季节性模型进行建模。

另外，对于长期趋势的分析，还可以使用趋势分解方法，将时间序列分解为趋势、季节性和残差成分，以更好地理解 and 描述猪肉价格的变化趋势。

需要注意的是，建立数学模型时还需要考虑数据的充分性和代表性，以及模型的合理性和准确性，在实际应用中可能需要不断进行模型验证和调整。

综上所述，通过统计分析猪肉价格与各因素之间的相关性，并建立适当的时间序列模型，可以更好地描述和预测猪肉价格的变化趋势^[5]。具体的分析和建模过程需要根据附件中的数据和具体问题要求进行。

三、 模型假设

(一) 问题一的模型假设

1. 建立皮尔逊积矩相关系数模型

皮尔逊积矩相关系数用于衡量两个变量之间的线性相关性。建立皮尔逊积矩相关系数模型可以通过以下步骤进行：

收集数据：收集包含两个变量的数据集，每个变量的取值要对应。

计算均值：对每个变量的取值计算均值。假设变量 X 的均值为 \bar{X} ，变量 Y 的均值为 \bar{Y} 。

计算差值：对每个数据点，分别减去对应变量的均值，得到差值。对于变量 X 的第 i 个数据点，记作 $X_i - \bar{X}$ ；对于变量 Y 的第 i 个数据点，记作 $Y_i - \bar{Y}$ 。

计算标准差：对每个变量的差值计算标准差。对于变量 X 的差值，记作 σ_X ；对于变量 Y 的差值，记作 σ_Y 。

计算协方差：对于每对差值，将两个差值相乘得到乘积，然后求平均。这个平均值称为协方差。记变量 X 和 Y 的协方差为 $Cov(X, Y)$ 。

计算相关系数：使用协方差和标准差计算皮尔逊积矩相关系数。相关系数的计算公式见公式(1)。

$$r = \frac{Cov(X, Y)}{\sigma_X \times \sigma_Y} \quad (1)$$

进行模型分析：根据相关系数的取值范围进行模型分析。相关系数 r 的取值范围为 $[-1, 1]$ ，其中相关系数为 -1 表示完全负相关，相关系数为 1 表示完全正相关，相关系数为 0 表示无相关性。

建立斯皮尔曼等级相关系数模型

斯皮尔曼等级相关系数用于衡量两个变量之间的单调相关性，不要求变量呈线性关系。建立斯皮尔曼等级相关系数模型可以按照以下步骤进行：

收集数据：收集包含两个变量的数据集，每个变量的取值要对应。

对变量进行排序：对每个变量的取值进行排序，从最小到最大分配一个等级，相同的值使用平均等级。

计算等级差值：对于每个数据点，计算两个变量的等级差值。对于变量 X 的第 i 个数据点，记作 diX ；对于变量 Y 的第 i 个数据点，记作 diY 。

计算等级差值的平方：对每个等级差值进行平方，得到平方差。

计算斯皮尔曼等级相关系数：使用平方差计算斯皮尔曼等级相关系数。相关系数的计算公式见公式(2)。

$$r = 1 - \frac{6 \times \sum diX^2}{n \times (n^2 - 1)} \quad (2)$$

其中， Σ 表示求和符号， diX 表示变量 X 的等级差值， n 表示数据点的数量。

进行模型分析：根据秩相关系数的取值范围进行模型分析。秩相关系数 r 的取值范围为 $[-1,1]$ ，其中秩相关系数为 -1 表示完全负相关，秩相关系数为 1 表示完全正相关，秩相关系数为 0 表示无相关性。

（二） 问题二的模型假设

1. 建立自回归积分滑动平均模型

自回归积分滑动平均模型是一种经典的时间序列分析模型，用于预测时间序列数据的未来趋势。ARIMA 模型结合了自回归、差分和滑动平均三个组成部分。

建立 ARIMA 模型的一般步骤：

数据预处理：收集并准备时间序列数据。确保数据的平稳性，即均值和方差在时间上是稳定的。如果数据不稳定，可以进行差分操作来实现稳定性。

确定参数：通过观察自相关图（ACF）和偏自相关图（PACF），确定 ARIMA 模型的参数。自相关图可以显示时间序列与自身在不同滞后时间点上的相关性，偏自相关图则显示了在移除其他滞后项的影响后，特定滞后项与时间序列之间的相关性。

拟合模型：使用确定的参数拟合 ARIMA 模型。这可以通过最大似然估计方法来实现。

模型诊断：对拟合的 ARIMA 模型进行诊断，检查模型的残差序列是否满足平稳性、白噪声和独立性等假设。可以使用残差的自相关图和偏自相关图来检查模型的拟合情况。

预测：使用已拟合的 ARIMA 模型进行未来值的预测。可以使用模型的预测误差来估计预测的置信区间。

四、模型建立与求解

（一）问题一的模型建立

1. 建立皮尔逊积矩相关系数模型

依据公式(3)，其中， X 和 Y 表示两个变量的取值， n 表示样本数量， x_i 和 y_i 表示第 i 个样本的 X 和 Y 的取值， \bar{x} 和 \bar{y} 表示 X 和 Y 的均值。使用 Python 进行皮尔逊积矩相关系数模型的建立¹，将得出的相关系数降序排列，得到表 4-1。

$$\rho_{X,Y} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (3)$$

表 4-1 猪肉(去骨统肉)价格与各因素的皮尔逊积矩相关系数

因素	相关系数
活猪（中等）价格	0.998
仔猪（普通）价格	0.936
时间	0.501
粳米（中等）价格	0.416
籼米（中等）价格	0.415
大白菜（中等）价格	0.364
籼稻（中等）价格	0.361
大豆（中等）价格	0.322
小麦（中等）价格	0.317
黄瓜（中等）价格	0.294
粳稻（中等）价格	0.293
玉米（中等）价格	0.094

使用表 4-1 中的数据，使用 Python 进行绘图²，得到图 4-1。

¹ 代码见七、附录（一）PPMCC.py

² 代码见七、附录（一）PPMCC.py

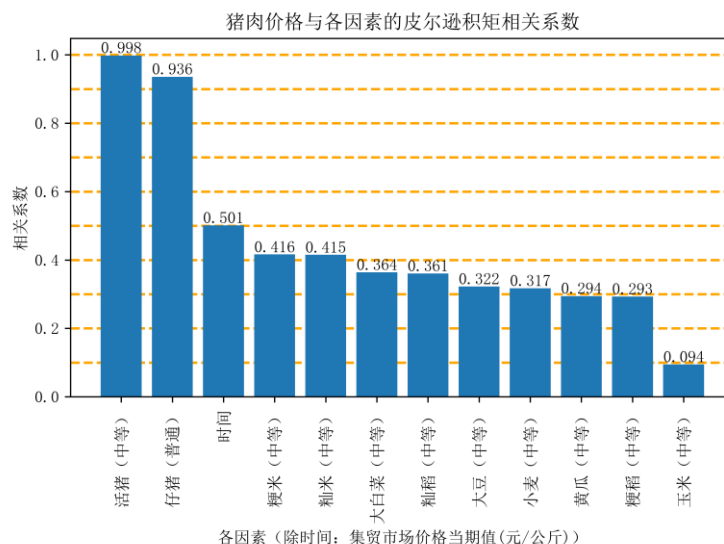


图 4-1 猪肉(去骨统肉)价格与各因素的皮尔逊积矩相关系数

2. 建立斯皮尔曼等级相关系数模型

依据公式(4)，其中， X 和 Y 表示两个变量的取值， n 表示样本数量， d_i 表示在排序后的数据中，第 i 个样本的 X 和 Y 的等级差。使用 Python 进行斯皮尔曼等级相关系数模型的建立³，将得出的秩相关系数降序排列，得到表 4-2。

$$\rho_{X,Y} = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \quad (4)$$

表 4-2 猪肉(去骨统肉)价格与各因素的斯皮尔曼等级相关系数

因素	秩相关系数
活猪（中等）价格	0.993
仔猪（普通）价格	0.886
时间	0.500
籼米（中等）价格	0.488
粳米（中等）价格	0.469
大豆（中等）价格	0.380
籼稻（中等）价格	0.370
大白菜（中等）价格	0.335
小麦（中等）价格	0.324
黄瓜（中等）价格	0.305
粳稻（中等）价格	0.233

³ 代码见七、附录（二）Rho.py

玉米（中等）价格	0.145
----------	-------

使用表 4-2 中的数据，使用 Python 进行绘图⁴，得到图 4-2。

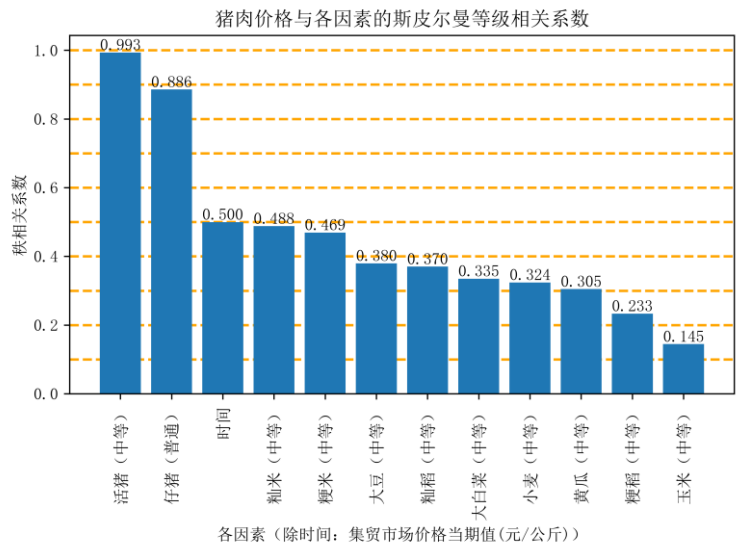


图 4-2 猪肉(去骨统肉)价格与各因素的斯皮尔曼等级相关系数

3. 建立平均相关系数模型

为了排除因为模型的差异性导致其系数的差异性，故利用表 4-1 和表 4-2 中的数据，使用 Python 进行皮尔逊积矩相关系数和斯皮尔曼等级相关系数的平均值模型的建立⁵，将得出的平均相关系数降序排列，得到表 4-3。

表 4-3 猪肉(去骨统肉)价格与各因素的平均相关系数

因素	相关系数	秩相关系数	平均相关系数
活猪（中等）价格	0.998	0.993	0.996
仔猪（普通）价格	0.936	0.886	0.911
时间	0.501	0.500	0.501
籼米（中等）价格	0.415	0.488	0.451
粳米（中等）价格	0.416	0.469	0.443
籼稻（中等）价格	0.361	0.370	0.365
大豆（中等）价格	0.322	0.380	0.351
大白菜（中等）价格	0.364	0.335	0.349

⁴ 代码见七、附录（二）Rho.py

⁵ 代码见七、附录（三）average_PPMCC_add_rho.py

小麦（中等）价格	0.317	0.324	0.321
黄瓜（中等）价格	0.294	0.305	0.299
粳稻（中等）价格	0.293	0.233	0.263
玉米（中等）价格	0.094	0.145	0.120

使用表 4-3 中的数据，使用 Python 进行绘图⁶，得到图 4-3。

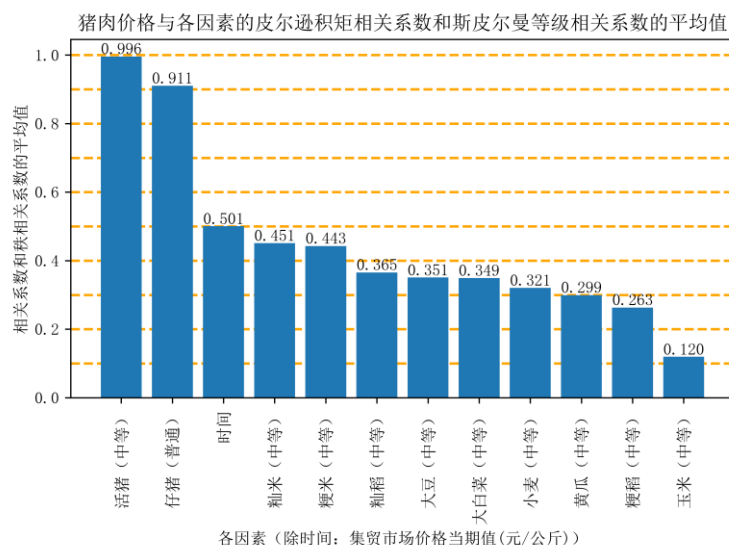


图 4-3 猪肉(去骨统肉)价格与各因素的斯皮尔曼等级相关系数

（二） 问题一的求解

通过分析图 4-3，可以得出如下结论：

影响猪肉(去骨统肉)价格波动的关键性因素有：活猪价格和仔猪价格；

影响猪肉(去骨统肉)价格波动的主要因素有：籼米价格、粳米价格和籼稻价格；

影响猪肉(去骨统肉)价格波动的次要因素有：大豆价格、大白菜价格、小麦价格、黄瓜价格和籼稻价格；

影响猪肉(去骨统肉)价格波动的无关因素有：玉米价格。

（三） 问题二的模型建立与求解

1. 绘制每年猪肉(去骨统肉)价格随月份变化的趋势图

利用 python 绘制每年猪肉(去骨统肉)价格随月份变化的趋势图⁷，绘制结果见图 4-4。

⁶ 代码见七、附录（三）average_PPMCC_add_rho.py

⁷ 代码见七、附录（四）yearly_pork.py

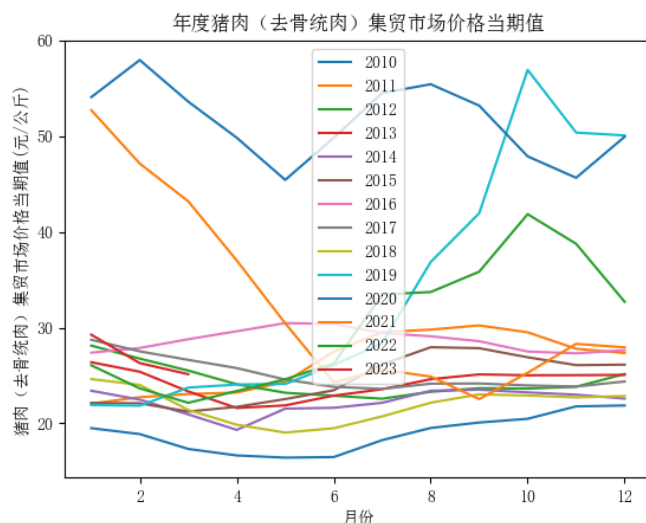


图 4-4 每年猪肉(去骨统肉)价格随月份变化的趋势图

由图 4-4 不难看出，猪肉(去骨统肉)价格随月份的变化趋势大致呈年中价格低，年底价格高，上半年降价，下半年涨价的趋势。

2. 建立自回归积分滑动平均模型

ARIMA(p, d, q)模型的表示为公式(5)。

$$(1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p)(1 - B)^d(X_t - \mu) = (1 + \theta_1 B + \theta_2 B^2 + \dots + \theta_q B^q)Z_t \quad (5)$$

其中， B 是滞后算子， X_t 是时间序列的观测值， μ 是均值， Z_t 是白噪声序列。参数 p 表示自回归阶数， d 表示差分阶数， q 表示移动平均阶数。 $\phi_1, \phi_2, \dots, \phi_p$ 是自回归系数， $\theta_1, \theta_2, \dots, \theta_q$ 是移动平均系数。

依据公式(5)，使用 python 进行自回归积分滑动平均模型的建立⁸，使用附件.xlsx 中的数据训练模型进行拟合，并预测 2023 年下半年猪肉(去骨统肉)价格的变化趋势，进行趋势图的绘制，绘制结果见图 4-5。

⁸ 代码见七、附录（五）ARIMA.py

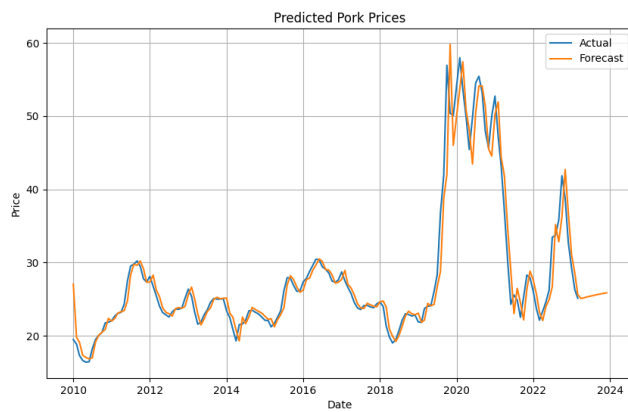


图 4-5 猪肉(去骨统肉)价格变化的趋势图

由图 4-5 不难看出，2023 年下半年猪肉(去骨统肉)价格的变化趋势呈现上涨趋势，符合四、（三）1.中的分析结果。

五、 模型评价

（一） 对自回归积分滑动平均模型的评价

使用 python 进行自回归积分滑动平均模型的结果分析和变化趋势分析（即模型拟合结果的计算）⁹，得到表 5-1。

表 5-1 SARIMAX Results

Dep. Variable	猪肉（去骨统肉）		No. Observations	159		
Model	ARIMA(1, 0, 1)		Log Likelihood	-366.493		
Date:	Fri, 30 Jun 2023		AIC	740.986		
Time	15:44:43		BIC	753.261		
Sample	01-01-2010-03-01-2023		HQIC	745.971		
Covariance Type	opg					
	coef	std err	z	P> z	[0.025	0.975]
const	27.0644	6.084	4.448	0.000	15.139	38.989
ar.L1	0.9387	0.025	38.292	0.000	0.891	0.987
ma.L1	0.3142	0.045	6.928	0.000	0.225	0.403
sigma2	5.7822	0.307	18.804	0.000	5.180	6.385
Ljung-Box (L1) (Q)	0.40		Jarque-Bera (JB)	773.11		
Prob(Q)	0.53		Prob(JB)	0.00		
Heteroskedasticity (H)	15.52		Skew	1.56		
Prob(H) (two-sided)	0.00		Kurtosis	13.34		

通过对表 5-1 中的数据分析，得出以下结论：

模型选择：根据您提供的模型阶数（ARIMA(1, 0, 1)），模型假设时间序列的自回归阶数为 1（AR=1），没有差分（I=0），滑动平均阶数为 1（MA=1）。

系数解释：模型结果中给出了各个参数的估计值（coef）、标准误差（std err）、z 值、P 值（P>|z|）以及置信区间（[0.025, 0.975]）。例如，自回归系数（ar.L1）的估计值为 0.9387，标准误差为 0.025，z 值为 38.292，P 值小于 0.001，置信区间为[0.891, 0.987]。这些参数用于描述模型的结构和影响。

常数项（const）：模型结果中的常数项（const）的估计值为 27.0644，标准误差为 6.084。这表示在模型中考虑了一个常数偏移量。

sigma2：模型结果中的 sigma2 表示残差的方差估计值，为 5.7822。较小的方差值通常表示模型对

⁹ 代码见七、附录（五）ARIMA.py

数据的拟合较好。

模型适应度：AIC（赤池信息准则）为 740.986，BIC（贝叶斯信息准则）为 753.261，较低的 AIC 和 BIC 值通常表示模型的拟合较好。

诊断统计：模型结果中提供了 Ljung-Box 统计量（Q）和 Jarque-Bera 统计量（JB）。Ljung-Box 统计量用于检验模型的残差序列中是否存在自相关性，P 值为 0.53，表明在滞后阶数为 1 时没有统计显著的自相关。Jarque-Bera 统计量用于检验残差序列的正态性，P 值小于 0.001，表明残差序列的分布不服从正态分布。

总体而言，根据表 5-1 的模型结果，该自回归积分滑动平均模型对猪肉价格数据的拟合较好，残差序列中没有明显的自相关性，并且在一定程度上符合正态分布。可以进一步研究和分析模型结果，探索猪肉价格的变化趋势和其他统计特征。

六、 参考文献

- [1] 何秋红. 《中国猪肉价格波动分析》[J]. 《中国农业大学硕士学位论文》, 2007.
- [2] 王阳. 《针对河北省猪肉价格波动规律的数学模型研究》[J]. 《河北科技大学硕士学位论文》, 2011.
- [3] 刘翠霞. 《基于关系聚类的面板数据向量自回归模型及其应用研究》[J]. 《西南财经大学博士学位论文》, 2019.
- [4] 白华艳. 《产业链视角下中国猪肉价格波动机制研究》[J]. 《华南农业大学博士学位论文》, 2018.
- [5] 张瑛瑛. 《黑龙江省猪肉市场周期波动规律的研究》[J]. 《黑龙江八一农垦大学硕士学位论文》, 2008.

七、 附录

(一) PPMCC.py

```
import pandas as pd
import matplotlib.pyplot as plt
from scipy.stats import pearsonr

def calculate_correlations(filename):
    # 读取 Excel 文件
    data = pd.read_excel(filename)

    # 提取所需的列数据
    column_names = data.columns
    pork_prices = data[column_names[-1]] # 猪肉价格列

    # 转换时间列为数值型数据
    time_column = data[column_names[0]]
    time_values = pd.to_numeric(time_column, errors='coerce')

    # 创建空列表存储相关系数和列名
    correlations = []

    # 计算最后一列与前面每一列的相关系数
    for i in range(1, len(column_names) - 1):
        column_name = column_names[i]
        column_data = data[column_name]

        # 处理时间列的数据类型
        if pd.api.types.is_datetime64_any_dtype(column_data):
            column_data = pd.to_numeric(column_data, errors='coerce')

        correlation, _ = pearsonr(pork_prices, column_data)
        correlations.append((column_name, correlation))

    # 计算时间列与猪肉价格的皮尔逊相关系数
```

```
time_corr, _ = pearsonr(time_values, pork_prices)
```

```
correlations.append(("时间", time_corr))
```

```
# 根据相关系数降序排序
```

```
correlations.sort(key=lambda x: x[1], reverse=True)
```

```
return correlations
```

```
def plot_correlations(correlations):
```

```
    # 提取列名和相关系数
```

```
    labels = [x[0] for x in correlations]
```

```
    coefficients = [x[1] for x in correlations]
```

```
# 输出相关系数数据
```

```
for column, correlation in correlations:
```

```
    print(f'猪肉价格与{column} 的皮尔逊积矩相关系数: {format(correlation, '.20f')}")
```

```
# 输出相关系数数据并添加到柱形图上
```

```
plt.rcParams['font.sans-serif'] = ['SimSun'] # 设置宋体作为默认字体
```

```
plt.figure(dpi=600)
```

```
plt.bar(labels, coefficients)
```

```
plt.xticks(rotation=90)
```

```
plt.xlabel("各因素（除时间：集贸市场价格当期值(元/公斤)）")
```

```
plt.ylabel("相关系数")
```

```
plt.title("猪肉价格与各因素的皮尔逊积矩相关系数")
```

```
# 添加柱形上的数据标签
```

```
for i in range(len(labels)):
```

```
    plt.text(i, coefficients[i], format(coefficients[i], '.3f'), ha='center', va='bottom')
```

```
# 添加水平虚线
```

```
for i in range(1, int(max(coefficients) * 10) + 2):
```

```
    plt.axhline(i * 0.1, color='orange', linestyle='--', zorder=0)
```



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

```
def main():
    filename = '附件.xlsx'
    correlations = calculate_correlations(filename)
    plot_correlations(correlations)

if __name__ == "__main__":
    main()
```

(二) Rho.py

```
import pandas as pd
import matplotlib.pyplot as plt
from scipy.stats import spearmanr
```

```
def calculate_correlations(filename):
    # 读取 Excel 文件
    data = pd.read_excel(filename)

    # 提取所需的列数据
    column_names = data.columns
    pork_prices = data[column_names[-1]] # 猪肉价格列

    # 转换时间列为数值型数据
    time_column = data[column_names[0]]
    time_values = pd.to_numeric(time_column, errors='coerce')

    # 创建空列表存储相关系数和列名
    correlations = []

    # 计算最后一列与前面每一列的斯皮尔曼相关系数
```

```

for i in range(1, len(column_names) - 1):
    column_name = column_names[i]
    column_data = data[column_name]

    # 处理时间列的数据类型
    if pd.api.types.is_datetime64_any_dtype(column_data):
        column_data = pd.to_numeric(column_data, errors='coerce')

    correlation, _ = spearmanr(pork_prices, column_data)
    correlations.append((column_name, correlation))

# 计算时间列与猪肉价格的斯皮尔曼相关系数
time_corr, _ = spearmanr(time_values, pork_prices)
correlations.append(("时间", time_corr))

# 根据相关系数降序排序
correlations.sort(key=lambda x: x[1], reverse=True)

return correlations

```

```

def plot_correlations(correlations):
    # 提取列名和相关系数
    labels = [x[0] for x in correlations]
    coefficients = [x[1] for x in correlations]

    # 输出相关系数数据
    for column, correlation in correlations:
        print(f'猪肉价格与{column} 的斯皮尔曼等级相关系数: {format(correlation, '.20f)}')

    # 输出相关系数数据并添加到柱形图上
    plt.rcParams['font.sans-serif'] = ['SimSun'] # 设置宋体作为默认字体
    plt.figure(dpi=600)
    plt.bar(labels, coefficients)
    plt.xticks(rotation=90)

```

```

plt.xlabel("各因素（除时间：集贸市场价格当期值(元/公斤)）")
plt.ylabel("秩相关系数")
plt.title("猪肉价格与各因素的斯皮尔曼等级相关系数")

# 添加柱形上的数据标签
for i in range(len(labels)):
    plt.text(i, coefficients[i], format(coefficients[i], '.3f'), ha='center', va='bottom')

# 添加水平虚线
for i in range(1, int(max(coefficients) * 10) + 2):
    plt.axhline(i * 0.1, color='orange', linestyle='--', zorder=0)

plt.tight_layout()
plt.show()

def main():
    filename = '附件.xlsx'
    correlations = calculate_correlations(filename)
    plot_correlations(correlations)

if __name__ == "__main__":
    main()

```

（三） average_PPMCC_add_rho.py

```

import pandas as pd
import matplotlib.pyplot as plt
from scipy.stats import pearsonr
from scipy.stats import spearmanr

def calculate_correlations(filename):
    # 读取 Excel 文件
    data = pd.read_excel(filename)

```

```

# 提取所需的列数据
column_names = data.columns
pork_prices = data[column_names[-1]] # 猪肉价格列

# 转换时间列为数值型数据
time_column = data[column_names[0]]
time_values = pd.to_numeric(time_column, errors='coerce')

# 创建空列表存储相关系数和列名
correlations = []

# 计算最后一列与前面每一列的相关系数
for i in range(1, len(column_names) - 1):
    column_name = column_names[i]
    column_data = data[column_name]

    # 处理时间列的数据类型
    if pd.api.types.is_datetime64_any_dtype(column_data):
        column_data = pd.to_numeric(column_data, errors='coerce')

    correlation_pearsonr, _ = pearsonr(pork_prices, column_data)
    correlation_spearmanr, _ = spearmanr(pork_prices, column_data)
    correlation = (correlation_pearsonr + correlation_spearmanr) / 2
    correlations.append((column_name, correlation))

# 计算时间列与猪肉价格的皮尔逊相关系数和斯皮尔曼相关系数的平均值
time_corr_pearsonr, _ = pearsonr(time_values, pork_prices)
time_corr_spearmanr, _ = spearmanr(time_values, pork_prices)
time_corr = (time_corr_pearsonr + time_corr_spearmanr) / 2
correlations.append(("时间", time_corr))

# 根据相关系数降序排序
correlations.sort(key=lambda x: x[1], reverse=True)

```

```

return correlations

def plot_correlations(correlations):
    # 提取列名和相关系数
    labels = [x[0] for x in correlations]
    coefficients = [x[1] for x in correlations]

    # 输出相关系数数据
    for column, correlation in correlations:
        print(f"猪肉价格与 {column} 的皮尔逊积矩相关系数和斯皮尔曼等级相关系数的平均值:
{format(correlation, '.20f')}")

    # 输出相关系数数据并添加到柱形图上
    plt.rcParams['font.sans-serif'] = ['SimSun'] # 设置宋体作为默认字体
    plt.figure(dpi=600)
    plt.bar(labels, coefficients)
    plt.xticks(rotation=90)
    plt.xlabel("各因素（除时间：集贸市场价格当期值(元/公斤)）")
    plt.ylabel("相关系数和秩相关系数的平均值")
    plt.title("猪肉价格与各因素的皮尔逊积矩相关系数和斯皮尔曼等级相关系数的平均值")

    # 添加柱形上的数据标签
    for i in range(len(labels)):
        plt.text(i, coefficients[i], format(coefficients[i], '.3f'), ha='center', va='bottom')

    # 添加水平虚线
    for i in range(1, int(max(coefficients) * 10) + 2):
        plt.axhline(i * 0.1, color='orange', linestyle='--', zorder=0)

    plt.tight_layout()
    plt.show()

def main():

```

```
filename = '附件.xlsx'
correlations = calculate_correlations(filename)
plot_correlations(correlations)
```

```
if __name__ == "__main__":
    main()
```

(四) yearly_pork.py

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
def plot_pork_prices(dataframe, time_column, pork_column):
    # 提取所需列数据
    data = dataframe[[time_column, pork_column]]

    # 将时间列设置为索引
    data.set_index(time_column, inplace=True)

    # 将时间列转换为日期时间类型
    data.index = pd.to_datetime(data.index)

    # 按年份对数据进行分组和月度重采样
    data_grouped = data.resample('M').sum()

    # 提取每年的数据
    years = data_grouped.index.year.unique()

    # 绘制每年的折线图
    for year in years:
        year_data = data_grouped[data_grouped.index.year == year]
        plt.plot(year_data.index.month, year_data[pork_column], label=str(year))

    # 设置图形标题和标签
```

```

plt.rcParams['font.sans-serif'] = ['SimSun'] # 设置宋体作为默认字体
plt.title('年度猪肉（去骨统肉）集贸市场价格当期值')
plt.xlabel('月份')
plt.ylabel('猪肉（去骨统肉）集贸市场价格当期值(元/公斤)')
plt.legend()

# 显示图形
plt.show()

# 主函数
def main():
    # 读取 Excel 文件
    df = pd.read_excel('附件.xlsx')

    # 定义时间列和猪肉列的名称
    time_column = '时间'
    pork_column = '猪肉（去骨统肉）'

    # 调用封装的函数进行绘图
    plot_pork_prices(df, time_column, pork_column)

# 调用主函数
if __name__ == '__main__':
    main()

```

（五） ARIMA.py

```

import pandas as pd
import numpy as np
import statsmodels.api as sm
import matplotlib.pyplot as plt

def run_arima_analysis():

```

```

# 读取 Excel 文件中的数据
df = pd.read_excel('附件.xlsx', parse_dates=['时间'])

# 过滤数据，仅保留在指定时间范围内的数据
start_date = pd.to_datetime('2010-01-01')
end_date = pd.to_datetime('2023-03-01')
df = df[(df['时间'] >= start_date) & (df['时间'] <= end_date)]

# 设置时间列作为索引，并将其转换为每月一号的频率
df = df.set_index('时间')
df = df.resample('MS').mean()

# 构建自回归积分滑动平均模型（ARIMA）
model = sm.tsa.ARIMA(df['猪肉（去骨统肉）'], order=(1, 0, 1))

# 训练模型并进行拟合
model_fit = model.fit()

# 预测未来的猪肉价格
start_date = pd.to_datetime('2010-01-01')
end_date = pd.to_datetime('2023-12-01')
forecast = model_fit.predict(start=start_date, end=end_date)

# 绘制原始数据和预测结果的图表
plt.figure(figsize=(10, 6))
plt.plot(df.index, df['猪肉（去骨统肉）'], label='Actual')
plt.plot(forecast.index, forecast, label='Forecast')
plt.title('Predicted Pork Prices')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.grid(True)
plt.show()

# 分析模型的结果和变化趋势

```



```
print(model_fit.summary())
```

```
if __name__ == '__main__':
```

```
    run_arima_analysis()
```