Modeling the pork price cycle in China based on the age structure

Peng Shicheng 120090636

November 29^{th}, 2022

Contents

1 Abstract

Pork price is essential in economic operation. Pork price prediction advises governing the macroeconomy. Pork price shows an explicitly periodic feature. Previous researchers used fancy functions to fit the price cycle and predict it. In this paper, we use the most straightforward optimization and age structure models to simulate the production process in pig farms and the pork price trend. We construct our model based on the actual economic behavior, which is more understandable in the economic sense. Our results match the claim that the price cycle is around five years and do well in empirical testing. We find the model is sensitive to forage price in the sensitivity analysis. Finally, we give some advice on controlling inflation based on our prediction.

2 Introduction

Pork price is significant in our daily life. More than 60% of the family's total meat consumption in China is pork (Chen et al., 2014). So, pork prices can significantly impact real households' disposable income. When pork prices are too high, daily consumption of other products will reduce, slowing economic growth. Besides, pork price is selected as one of the critical components in the Consumer Price Index (CPI) that indicates the economy's inflation level. If the pork price can be predicted, the government can estimate the economic situation in the future and regulate the macro-economy to prevent an inflation outbreak or economic recession (Wang et al., 2009).

Mordecai Ezekiel (1938) introduced the cobweb theorem to the world. The theorem claimed that if the commodity production is determined by the producers' response to the price, then the price of this commodity will show strong periodicity. The intrinsic logic is that if the producer notices that the price of the commodity is going up, he will increase production and try to sell as much as possible to maximize his profit. However, this will lead to a supply surplus and cause the price to drop immediately. After knowing the price decline, the producer reduces the supply, and the commodity shortage will raise the price again.

Previous researchers found that pork price is the key factor influencing pork supply and demand (Jeremić et al., 2018). So, pork is a typical agricultural product that follows the cobweb theorem. Based on the theorem, the trend of pork prices is in a periodic pattern. This pattern is proven to be existing in both US and China. The price cycle in China is confirmed to be around 4 to 5 years (Gale & Hu, 2012). Many researchers contributed a lot to studying this periodic cycle. Hovav Talpaz (1974) used the cosine function to simulate the seasonal change in pork supply and demand. The SETAR model and machine learning combination precisely fit the pork price in the Chinese market (Zhao & Wu, 2015). Many fancy functions and solving techniques were used to match the periodic pattern of pork prices. However, few researchers have tried to solve the pork cycle based on the real production of hogs.

The main contributions of this paper:

- We try to simulate the pork price in China by modeling hogs' production process and age structure.
- Instead of using fancy functions to fit the price pattern, we construct the model based on the interaction between pig farmers and pork prices.
- Since our model follows the economy's operation, it is highly understandable.
- Our model applies the cobweb theorem to reality.

3 Data and data preprocessing

3.1 Raw data

In order to make the model as close to reality as possible, it is preferred to build the model based on several historical data. The following process uses the data shown in *Table 1.*

Table 1: **Data used in the model**
This table presents the data, the data frequency and the Chinese name of the data file.

Data	Unit	Frequency	File name in Chinese
Pork price	￥ per kilogramgram	weekly	22个省市 平均价 猪肉
CPI	-	monthly	CPI 所有项目 中国
Amount of sow	ten thousand	monthly	生猪存栏 能繁母猪
Amount of hogs supply	ten thousand	yearly	肉猪出栏头数
Piglet price	￥ per kilogramgram	monthly	22个省市 平均价 仔猪
Forage price	￥ per kilogramgram	weekly	平均价 生猪饲料

All data are downloaded from the Wind Financial Terminal, China's most professional financial database. Wind Financial Terminal collects these data from the Ministry of Agriculture and the Rural Affairs and National Bureau of Statistics, PRC.

Besides, the survival rates of the female pig are cited from the paper *Within-farm variability in the age structure of breeding-female pigs and reproductive performance on commercial swine breeding farms*(Koketsu, 2004). The survival rates for hog are quoted from the paper *Causes of mortality in Yorkshire pigs from birth to 20 weeks of age* (Fahmy & Bernard, 1971). These raw data are shown in *Table 2 and 3.*

We obtain the birth information from the paper *Simple calculation method of pig group structures, groups of pig population and turnover in intensive pig farm: example as a large scale pig farm for 10000 commercial fatting pigs annually*(Gao & Gu, 2015) (*Table 4*).

Finally, we need to find the quantity of pork produced by a hog to make our model more precise. It is commonly accepted that if the hog is weighted at about 95.9 kilograms, it is mature and can go to the market.

Table 2: **Survival rates used for female group**
All these data are observed within 2 years. We need additional operation in the data preprocessing.

The number of parity	Survival rate (%)
0	96.2
1	92.2
2	93.9
3 to 5	92.7
>6	90.01

Table 3: **Survival rates used for hogs group**
All these data can be used directly to calculate the survival of newborn hogs.

Stage	Survival rate (%)
weaned	82.4
alive after 20 weeks	97.3

Table 4: **Birth information for pigs**
All these data can be used directly to calculate the number of newborn hogs.

Item	Value
Fertilized rate (per pig)	80%
Successful birth (per pig)	95%
Average piglets in a parity	10
The reserved female piglets per parity	0.16

Most hogs will produce pork at 73% of their own weight (Huang et al., 2022). Thus, we can assume the pork produced by each hog is about 70 kilograms.

3.2 Data preprocessing

First, we need to calculate the adjusted pork price for each month, which is the real pork price, after kicking out the influence of inflation. China experienced high economic growth in the past several years. The Chinese yuan's purchasing power has declined rapidly and affected the prices of products a lot (Zhao & zhao, 2018). The pork price cycle will be more explicit after eliminating the influence of purchase power. The formula for adjusted pork price is as follows:

$$P_{adjusted} = \frac{P_{nominal}}{CPI} \times 100$$

$P_{adjusted}$ is the adjusted pork price.

$P_{nominal}$ is the original pork price.

CPI is the value of CPI corresponding to the same month with the pork price.

Second, all data should be converted into yearly frequency. We need to ensure the *hog supply-pork price transmission mechanism* is close to real life. This mechanism is one of the most critical relationships in the model set. The relationship can be found if these two data sets have the same frequency. However, the supply of hogs is yearly, while the pork price is monthly. Because the hog supply differs each month, we cannot accurately distribute the number of hogs supplied monthly. So, all information should be adjusted into yearly to adapt to the frequency of the hog supply data. The way to change the frequency is to take the average of all monthly data within the same year as the yearly data of that year. The converted formula is the following:

$$variable\ value_{yearly} = \frac{1}{number\ of\ data\ in\ a\ year} \sum variable\ value_{monthly\ or\ weekly}$$

As mentioned in the raw data part, the survival information for female pigs is observed within two years. We first assume that the annual survival rates remain the same in these two years to gain the annual data. Then, we can transform each 2-year survival rate s_{2-year} into an annual basis s_{1-year} based on the following formula.

$$s_{1-year} = (s_{2-year})^{\frac{1}{2}}$$

After calculation, the survival information for female pigs is in *Table 5*.

Table 5: **Survival rates used for female group**
All these data are converted into one year, and we can use them directly in the modeling.

The number of parity	**Survival rate (%)**
0	98.08
1	96.02
2	96.9
3 to 5	96.28
>6	94.87

3.3 Data analysis

3.3.1 Price adjustment

We compare the pork price before and after the adjustment (*Figure 1*). Prices after the adjustment have a smaller variance than the original ones. Thus, the preprocessing is meaningful. The adjusted price's mean value and standard deviation are ￥27.374 per kilogram and 7.154.

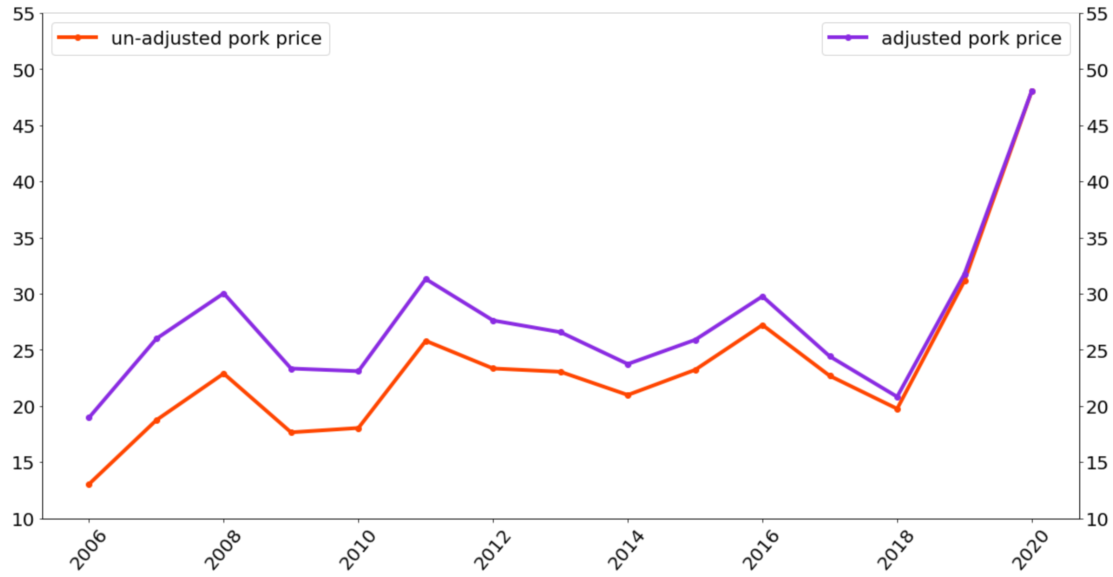


Figure 1: The effect of eliminating CPI on pork prices.

3.3.2 Forage price

Then, we draw the forage and adjusted pork prices into one graph (*Figure 2*). We find that these two patterns are not relative according to the picture. To justify our findings, we calculate the correlation between them. The result is only -0.002, which is very low. Thus, the forage price is not a key factor in pork pricing. In the model set, we can assume that the forage price is fixed for each year, and the fixed value is equal to the mean forage price (￥2.71 per kilogram). The standard deviation of the forage price is 0.45. We will use this value in the sensitivity analysis.

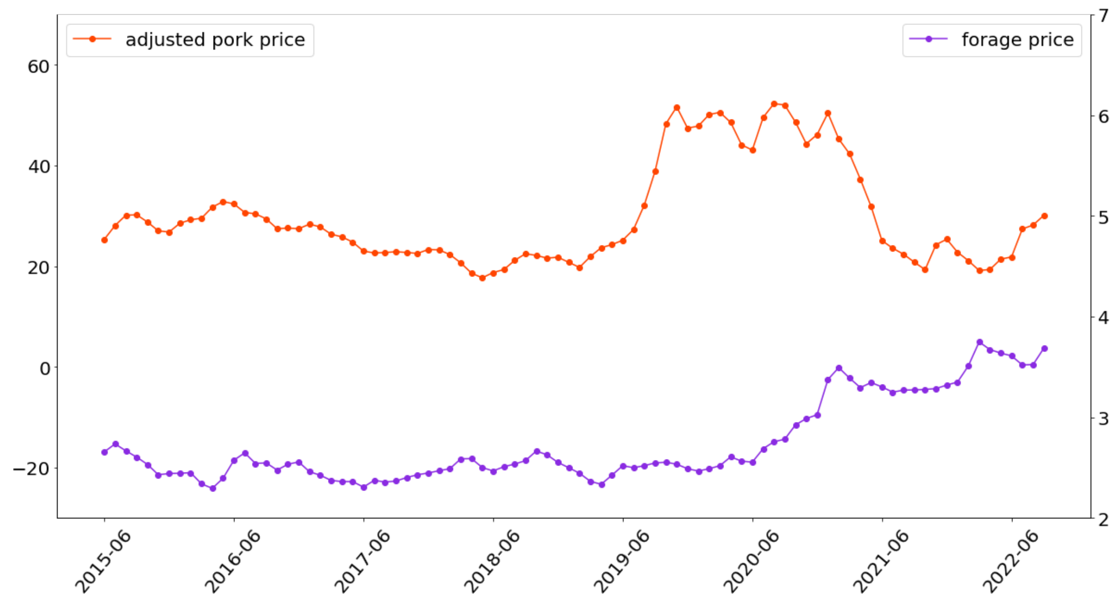


Figure 2: The comparison between the adjusted price and the forage price.

3.3.3 Piglet price

The mean and the standard deviation value of piglet price are ￥37.4 per kilogram and 23.1.

3.3.4 Amount of hogs supply

The annual supply of hogs has a high correlation (-0.531) with the adjusted price (*Figure 4*). The periodic alternation patterns also follow the cobweb theorem. It proves that our idea about modeling the price simulating the pork supply is reasonable.

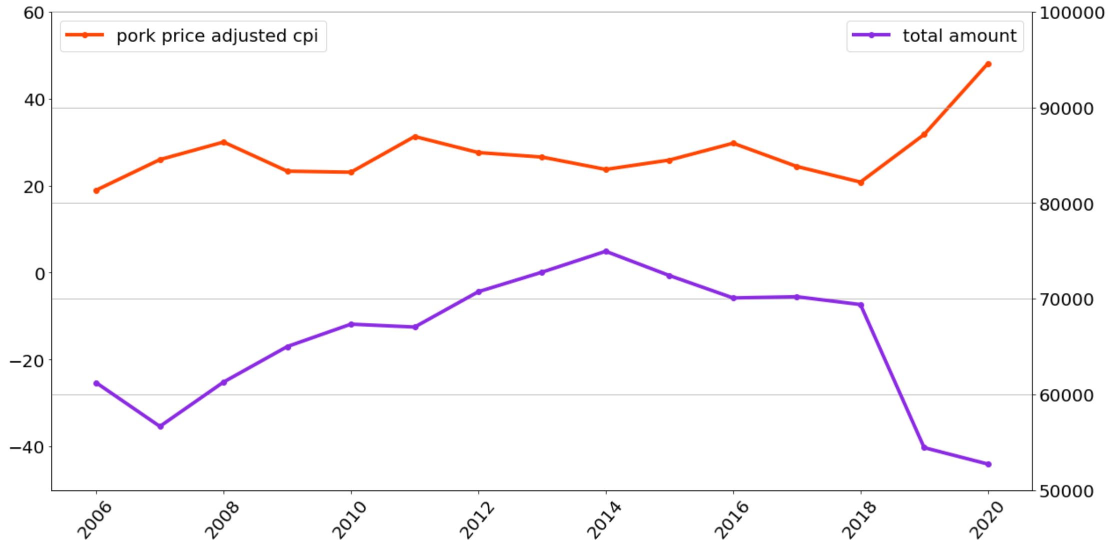


Figure 3: The comparison between the adjusted price and the annual hogs supply.

4 Model construction and analysis

4.1 Model construction

4.1.1 Model assumptions

Basic assumptions

1. Our model only considers the interaction between the pig farmers and the market. We will not consider external actions (e.g., government regulations).

2. No swine epidemic happens.

3. There are 53 weeks in a year.

4. All female pigs get pregnant by artificial insemination. Pig farmers do not need to raise any male pigs.

5. Pig farmers make the production decision based on the same logic each year.

6. The birth rates and survival rates will not change.

Model assumptions

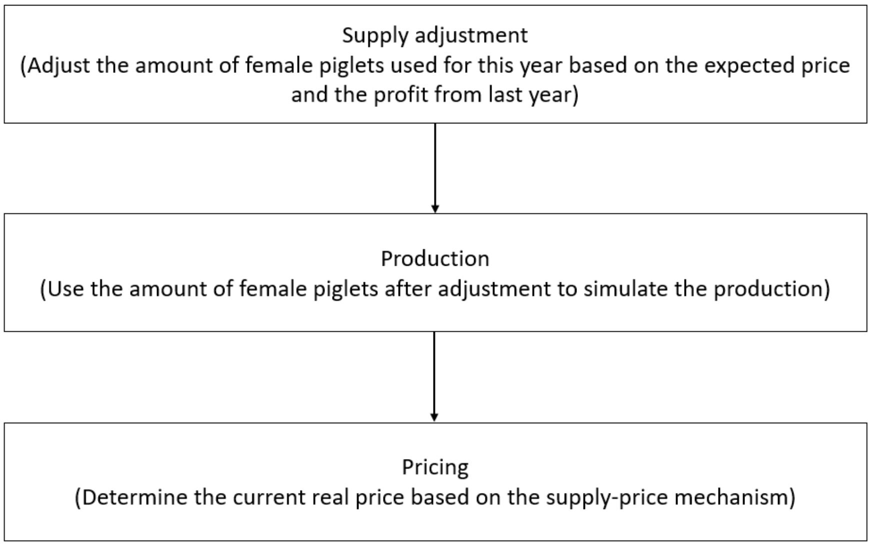


Figure 4: The whole production process for each year

The model is divided into three parts, supply adjustment, production, and pork pricing (*Figure 4*). At the beginning of each year, pig farmers will use the supply adjustment model to compute the adjusted amount of female piglets. After that, they buy or sell the corresponding amount of piglets and let the pigs produce. At the end of the year, all hogs go to the market, and the farmer earns money by selling pork at the current price. The supply-price mechanism in the pricing part decides the current pork price. These construct the whole production process in a year.

In the following, we will introduce the age structure in production first. This part can help readers learn about the whole process more clearly.

Age structure in production

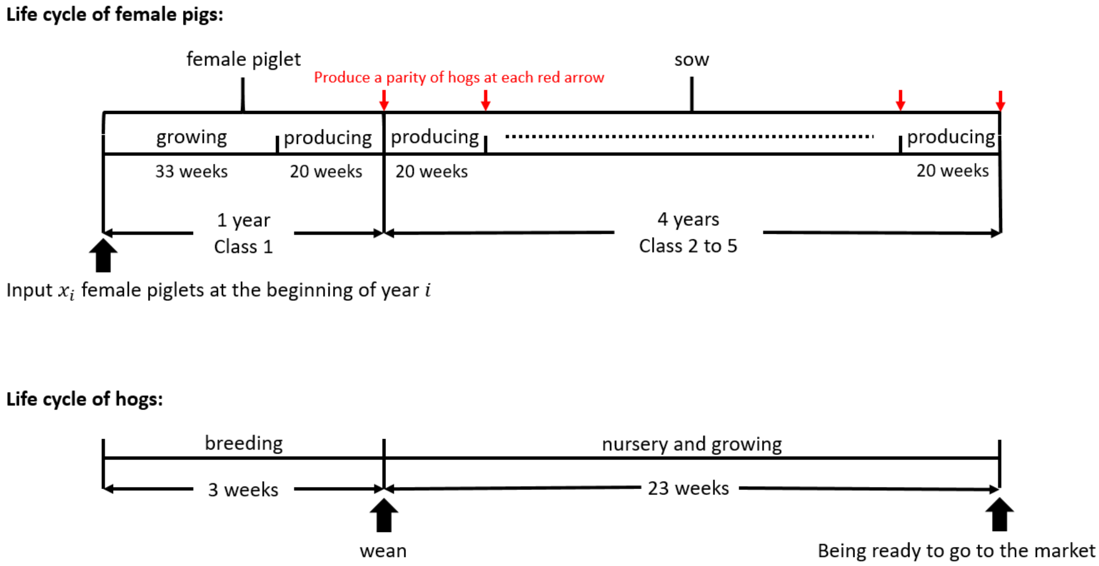


Figure 5: The life cycle of female pig and hog

In production, pigs are divided into **hogs (肉猪)** and **female pigs (繁育母猪)**. Hogs are pigs that will be slaughtered. Their meat will go to the market and be served as pork. **Female pigs** are used to produce baby pigs. We need to claim that not all pigs with female sex are female pigs. In each newborn parity, only the female baby with high fertility can be selected as a female pig. There are two kinds of female pigs. They are different due to the number of parity they have given birth to. **Female piglets (繁育母猪仔猪)** have not or only given birth to one parity. **Sows (能繁母猪)** produce more than one parities from their birth to now. We mention sows and female piglets here because we can only collect historical data on sows instead of the whole female pigs. The following will introduce more details about these pigs' life cycle and modeling.

Female pigs are responsible for producing baby pigs. Thirty-three weeks after birth, female baby piglets are sexually mature. Then, they start their first production. They will be fertilized in 1 week. After that, 80% of piglets will be pregnant successfully. About 15 weeks later, 95% of pregnant female piglets will successfully give birth to one parity. In the following four weeks, the female piglet nurses its babies. We assume there are ten baby pigs in each parity. Only 0.16 female babies are selected as the new female piglets. The other babies are chosen to be hogs. This is the whole process of production. It takes female piglets one year from birth to finish the first production. After their first production, these female piglets become sows. In the following years, sows will repeat the production again and again. Since each production takes only 20 weeks, we assume each sow can produce $\frac{53}{20}$ times a year. We will select 0.08 female piglets from each parity produced by sows (this number is 0.16 for female piglets). Most pig farms will hold a sow for five years. After five years, the sows' fertility reduces, and they will be disposed of. Thus, we assume the number of age classes belonging to female pigs is 5.

The life cycle of hogs is quite simple. Newborn hogs wean after three weeks. Then, they experience the nursery stage and growth stage in the next 23 weeks. After six months from birth, hogs mature and are ready to be sold. The total number of hogs produced each year is the sum of all hogs produced by each age class of female pigs. So, the model of hogs is correlated to the female pigs.

Supply adjustment

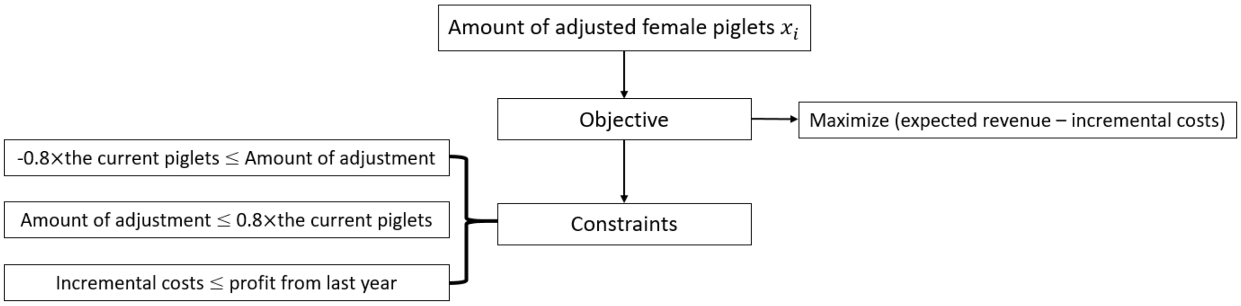


Figure 6: The supply adjustment model

$$\begin{aligned}&Total\ revenue\\ &= hogs\ supply\ after\ adjustment \times the\ amount\ of\ pork\ per\ hog \times expected\ price\ corresponding\ to\ the\ pork\ supply\end{aligned}$$

$$Incremental\ costs = incremental\ feed\ costs + purchase\ expenses$$

$$\begin{aligned}&Incremental\ feed\ costs\\ &= (survived\ amount\ of\ incremental\ femlae\ piglets + their\ babies) \times 365 \times forage\ consumed\ each\ day\ per\ pig\\ &\times forage\ price\end{aligned}$$

$$Purchase\ expenses = price\ of\ female\ piglet \times weight\ of\ a\ female\ piglet \times adjusted\ amount\ of\ female\ piglet\ x_i$$

$$Last\ year\ profit = total\ revenue\ of\ last\ year - total\ costs\ of\ last\ year$$

$$Total\ revenue\ of\ last\ year = hogs\ supply\ last\ year \times the\ amount\ of\ pork\ per\ hog \times pork\ price\ last\ year$$

$$Total\ cost\ last\ year = total\ feed\ costs\ last\ year + purchase\ expenses\ last\ year$$

$$\begin{aligned}&Total\ feed\ cost\ last\ year\\ &= (amount\ of\ female\ pig\ last\ year + amount\ of\ hog\ last\ year) \times 365 \times forage\ consumed\ each\ day\ per\ pig\\ &\times forage\ price\end{aligned}$$

$$\begin{aligned}&Purchase\ expenses\ last\ year\\ &= price\ of\ female\ piglet \times weight\ of\ a\ female\ piglet \times adjusted\ amount\ of\ female\ piglet\ last\ year\ x_{i-1}\end{aligned}$$

Figure 7: Formulas used in supply adjustment model

In the supply adjustment part, we try to simulate the production logic of pig farmers. Pig farmers will adjust the production plan of hogs based on the current expectation of pork price at the beginning of each year. The ultimate goal for pig farmers is to maximize their profit the following year.

Profit is the difference between revenue and cost. Revenue is the gain from selling porks produced by hogs this year. Cost includes the annual feed cost, purchasing expenses, and fixed costs. Feed cost is the total forage consumption of all pigs. Purchasing expenses are the amount of money pig farmers spend buying new female piglets. If pig farmers decide to sell some female piglets to the market and gain some profit, the purchasing cost will become purchasing income. Fixed cost is the amount of money used to expand pigsties and maintain the daily operation of pig farms. In the following model, we assume the fixed cost is 20

To maximize the revenue, pig farmers will probably put more female piglets into production if the expected pork price is high. It will increase the number of hogs produced in the following year. As a result, the farmer can earn as much as possible in the bullish pork market. If the expected price decreases, they will sell some female piglets to the market at the current piglet price. It can help pig farmers lock up some revenue in the bearish pork market.

In order to prevent pig farmers from increasing or decreasing the number of female piglets dramatically, we add some constraints. First, we assume that the inflow and the outflow of female piglets should be less or equal to 0.8 times the number of female piglets at the beginning of each year. Second, pig farmers invest their profits from last year to support the expansion of production. It means that the incremental total cost due to the incremental female piglets should not exceed last year's profits. The incremental total cost includes two parts, feed cost, and purchasing expenses. The incremental feed cost is the total amount of forage that new female piglets and their babies eat in one year. Purchasing expenses have been explained previously.

To obtain the adjustment of female piglets each year, we build up a maximizing linear programming problem. This model will be conducted at the beginning of each year. It can help farmers decide how many female piglets should be bought or sold that year.

Hog supply-pork price transmission mechanism

The transformation between the supply of hogs and pork prices is a complicated economic problem. Because we only aim to find the corresponding pork price for a given supply value, it is unnecessary to create a complex model to simulate the detailed supply-price mechanism. In previous data analysis (*Figure 3*), we find that these two are explicitly negatively correlated. We assume the relationship between supply and price follows a linear function.

4.1.2 Variables description

Variables are shown in *Table 6*:

Table 6: **Variables**
All these variables are on yearly basis, and we can use them directly in the following model.

Variable	Meaning	Unit
u_i	the total number of female pigs at year i	ten thousand
h_i	the total number of hogs at year i	ten thousand
$u_{i,j}$	the number of female pigs in the j^{th} age class at year i	ten thousand
x_i	the number of newly-purchased female piglets at year i	ten thousand
b_j	expected number of offspring from a female pig in age class j	1 per female pig
$b_{f,j}$	expected number of female piglet produced by a female pig in age class j	1 per female pig
$b_{h,j}$	expected number of hogs produced by a female pig in age class j	1 per female pig
$s_{w,n}$	fraction of hogs surviving from their birth to mature	
$s_{f,j}$	fraction of female pigs surviving from the $j-1^{th}$ class to the j^{th} class	
p_i	the pork price at year i	￥ per kilogram
$\hat{p}_i$	the expected pork price at beginning of year i	￥ per kilogram
q_i	the amount of hog supply at year i	ten thousand
f	the amount of forage a pig eats in one day	kilogram per day
pf	the price of forage	￥ per kilogram
r	the amount of pork a hog produces	kilogram per hog
c	the price of female piglets	￥ per kilogram
w	the weight of a female piglet	kilogram per piglet
$profit_i$	the profit earned in year i	￥

4.1.3 Model formulations

Supply adjustment model:

$$\begin{aligned} \max_{x_i} \quad & (h_i + s_{w,n}b_{h,1} \times x_i) \times r \times \hat{p}_i - [x_i \times (s_{w,n}b_{h,1} + s_{f,1}b_{f,1}) \times 365 \times f \times pf + c \times w \times x_i] \\ s.t. \quad & -0.8u_{i,1} \leq x_i \\ & x_i \leq 0.8u_{i,1} \\ & x_i \times (s_{w,n}b_{h,1} + s_{f,1}b_{f,1}) \times 365 \times f \times pf + c \times w \times x_i \leq profit_{i-1} \end{aligned} \tag{1}$$

The predicted price formula is same as the hog supply-pork price transmission mechanism in the following (*Equation 4*).

$$\hat{p}_i = n + m \times (h_i + s_{w,n}b_{h,1} \times x_i)$$

The profit part is calculated by:

$$profit_{i-1} = (0.8 \times h_{i-1} \times r \times p_{i-1}) - (h_{i-1} + u_{i-1}) \times f \times pf - x_{i-1} \times w \times c$$

Age structure of female pigs:

$$
\begin{aligned}
u_{i,1} &= s_{f,1}[b_{f,1}(u_{i-1,1} + x_{i-1}) + b_{f,2}u_{i-1,2} + ... + b_{f,5}u_{i-1,5}] \\
u_{i,2} &= s_{f,2}(u_{i-1,1} + x_{i-1}) \\
&\vdots \\
u_{i,5} &= s_{f,5}u_{i-1,4}
\end{aligned}
$$

The formula with Leslie matrix:

$$
\begin{bmatrix} u_{i,1} \\ u_{i,2} \\ ... \\ u_{i,5} \end{bmatrix} = \begin{bmatrix} s_{f,1}b_{f,1} & s_{f,1}b_{f,2} & ... & s_{f,1}b_{f,4} & s_{f,1}b_{f,5} \\ s_{f,2} & 0 & ... & 0 & 0 \\ & & ... & & \\ 0 & 0 & ... & s_{f,5} & 0 \end{bmatrix} \left(\begin{bmatrix} u_{i-1,1} \\ u_{i-1,2} \\ ... \\ u_{i-1,5} \end{bmatrix} + \begin{bmatrix} x_{i-1} \\ 0 \\ ... \\ 0 \end{bmatrix} \right) \tag{2}
$$

The amount of female pigs:

$$u_i = Lu_{i-1},\ L\ is\ the\ Leslie\ matrix$$

$$u_i = \sum_{j=1}^{5} u_{i,j}$$

The amount of sows:

$$sow_i = \sum_{j=2}^{5} u_{i,j}$$

The amount of female piglets:

$$female\ piglet_i = u_{i,1}$$

Supply of hogs:

The birth rate of hogs in the j^{th} group of female pigs ($b_{h,j}$) is calculated by:

$$b_{h,j} = b_j - b_{f,j}\ ,\ j = 1, 2, 3, 4, 5$$

$$h_i = s_{w,n}[(b_{h,1} \times u_{i-1,1} + b_{h,2} \times u_{i-1,2} + b_{h,3} \times u_{i-1,3} + b_{h,4} \times u_{i-1,4} + b_{h,5} \times u_{i-1,5}) + b_{h,1} \times x_{i-1}]$$

We write the model into a vector multiplication form:

$$\begin{bmatrix} h_{i,1} \\ h_{i,2} \\ \dots \\ h_{i,5} \end{bmatrix} = \begin{bmatrix} s_{w,n}b_{h,1} & s_{w,n}b_{h,2} & \dots & s_{w,n}b_{h,4} & s_{w,n}b_{h,5} \end{bmatrix} \left(\begin{bmatrix} u_{i-1,1} \\ u_{i-1,2} \\ \dots \\ u_{i-1,5} \end{bmatrix} + \begin{bmatrix} x_{i-1} \\ 0 \\ \dots \\ 0 \end{bmatrix} \right) \tag{3}$$

The annual hogs supply:

$$h_i = \sum_{j=1}^{5} h_{i,j}$$

Hog supply-pork price transmission mechanism:

$$p_i = n + m \times q_i \tag{4}$$

4.1.4 Solving the model

Hyper-parameter:

We give values to the hyper-parameters so that we can solve the model (*Table 7*).

Table 7: **Hyper-parameter**
All these values are mentioned in the data part.

Hyper-parameter	**Value**	**Unit**
b_1	7.6	1 per female pig
$b_{2,3,4,5}$	20.14	1 per female pig
$b_{f,1}$	0.16	1 per female pig
$b_{f,2,3,4,5}$	0.212	1 per female pig
$s_{w,n}$	0.801752	
$s_{f,1}$	0.974385	
$s_{f,2,3,4,5}$	0.967479	
f	1.5	kilogram per day
pf	2.71	￥ per kilogram
r	70	kilogram per hog
c	37.4	￥ per kilogram
w	30	kilogram per piglet

Supply adjustment model:

This is a simple linear programming problem. We try to use the K.K.T condition to solve the problem.

We rewrite the objective function into a minimization problem so that we can derive the K.K.T. condition easily:

$$-mr(s_{w,n}b_{h,1})^2x_i^2 - [2rmh_is_{w,n}b_{h,1} + rns_{w,n}b_{h,1} - (s_{w,n}b_{h,1} + s_{f,1}b_{f,1}) \times 365 \times f \times fp - cw]x_i - (rmh_i^2 + rnh_i)$$

The Lagrange formula of the problem is:

$$L(x_i, \mu, \lambda, \varphi) =$$

$$\begin{aligned}
&- mr(s_{w,n}b_{h,1})^2x_i^2 - [2rmh_is_{w,n}b_{h,1} + rns_{w,n}b_{h,1} - (s_{w,n}b_{h,1} + s_{f,1}b_{f,1}) \times 365 \times f \times fp - cw]x_i \\
&- (rmh_i^2 + rnh_i) \\
&+ \mu(x_i + 0.8u_{i,1}) \\
&+ \lambda(-x_i + 0.8u_{i,1}) \\
&+ \varphi(profit_{i-1} - x_i \times (s_{w,n}b_{h,1} + s_{f,1}b_{f,1}) \times 365 \times f \times pf - c \times w \times x_i)
\end{aligned} \tag{5}$$

The stationary condition:

$$\frac{\partial L}{\partial x_i} = 0$$

$$x_i = -h - is_{w,n}b_{h,1} - \frac{rns_{w,n}b_{h,1}}{2mr} + \frac{1+\varphi}{2mr}[(s_{w,n}b_{h,1} + s_{f,1}b_{f,1}) \times 365 \times f \times pf - c \times w] - \frac{\mu - \lambda}{2mr}$$

Primal feasibility:

$$-0.8u_{i,1} \leq x_i$$

$$x_i \leq 0.8u_{i,1}$$

$$x_i \times (s_{w,n}b_{h,1} + s_{f,1}b_{f,1}) \times 365 \times f \times pf + c \times w \times x_i \leq profit_{i-1}$$

Dual feasibility:

$$\mu, \lambda, \varphi \leq 0$$

Complementarity conditions:

$$\mu(x_i + 0.8u_{i,1}) = 0$$

$$\lambda(-x_i + 0.8u_{i,1}) = 0$$

$$\varphi(profit_{i-1} - x_i \times (s_{w,n}b_{h,1} + s_{f,1}b_{f,1}) \times 365 \times f \times pf - c \times w \times x_i) = 0$$

Since the K.K.T. condition is too long, we do not fill values of hyper-parameters into it and rewrite it again.

Age structure of female pigs:

We fill values of hyper-parameters into the Leslie matrix of female pigs.

$$L = \begin{bmatrix} 0.974385 \times 0.16 & 0.974385 \times 0.212 & ... & 0.974385 \times 0.212 & 0.974385 \times 0.212 \\ 0.967479 & 0 & ... & 0 & 0 \\ & & ... & & \\ 0 & 0 & ... & 0.967479 & 0 \end{bmatrix} \quad (6)$$

Because the population of female pigs follows the formula $u_i = Lu_{i-1}$, we can use matrix decomposition to represent the value of u_i:

$$u_i = L^i u_1 = \sum_{j=1}^{5} c_j \lambda_j^i v_j, \ \lambda \ is \ the \ j^t h \ eigenvalue, \ v_j \ is \ the \ corresponding \ eigenvector$$

When i goes to infinity, $u_i = c_1 \lambda_1^n v_1$. λ_1 is the largest eigenvalue, which is 0.972199 in the Leslie matrix. Thus, the ratio of population between adjacent age classes can be calculated by:

$$\frac{class\ (j+1)}{class\ j} = \frac{s_{f,j+1}}{0.972199} = \frac{0.967479}{0.972199} = 0.99514489, \ j = 1, 2, 3, 4$$

Suppose the population in the first group is 1, the age structure under the natural condition is like this:

$$\begin{bmatrix} u_{i,1} \\ u_{i,2} \\ ... \\ u_{i,5} \end{bmatrix} = \begin{bmatrix} 0.99514489^0 \\ 0.99514489^1 \\ ... \\ 0.99514489^4 \end{bmatrix} \quad (7)$$

In order to generate a simulation process, we assume the female pigs' age structure in 2006 follows the natural condition (2006 is the earliest year when the price data is recorded). Then, we can infer the population in each age class. In 2006, the number of sows (class 2 to 5) was 4887.583333 (actual value). We plug in the proportion for each group. The estimated population of each group is:

$$\begin{bmatrix} u_{2006,1} \\ u_{2006,2} \\ u_{2006,3} \\ u_{2006,4} \\ u_{2006,5} \end{bmatrix} = \begin{bmatrix} 1236.83555468 \\ 1230.83058201 \\ 1224.85476415 \\ 1218.90795953 \\ 1212.99002731 \end{bmatrix} \quad (8)$$

We will use this population as the beginning data in the following simulation.

Supply of hogs:

After filling in the value of hyper-parameters, the production vector of the hog is like this:

$$\begin{bmatrix} 7.44 & 19.928 & 19.928 & 19.928 & 19.928 \end{bmatrix} \tag{9}$$

Hog supply-pork price transmission mechanism:

We put the historical data of pork price and hog supply into a linear regression. Then, the result is:

$$p_i = 61.5770624 - 0.00051957 \times q_i$$

The fitted graph is like:

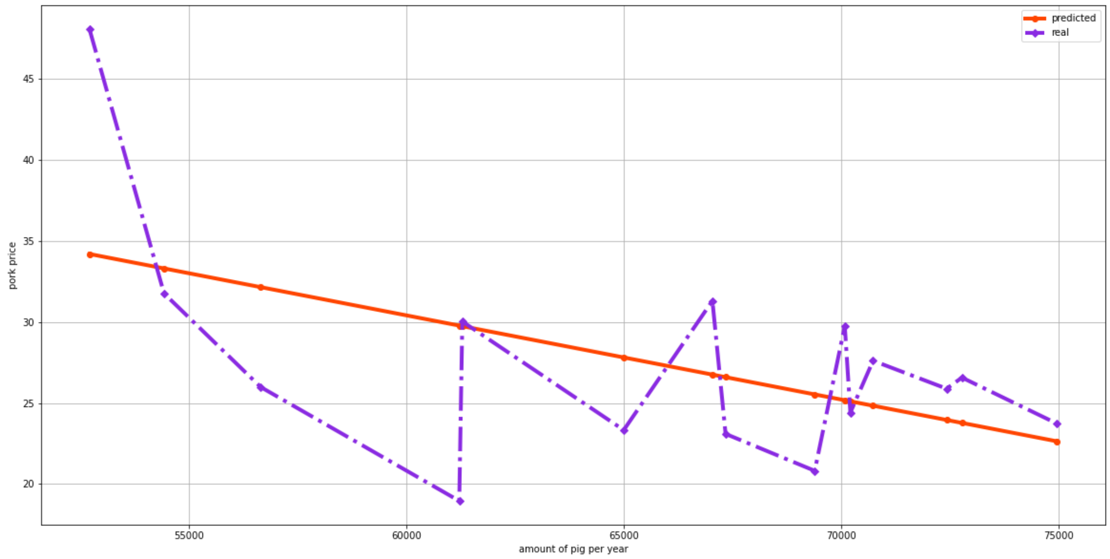


Figure 8: The fitted model of the supply-price mechanism

The result is statistically significant, with the p-value of each parameter being less than 0.05. We can use this model in our following simulation.

4.2 Results

We use all models above to simulate the pork price trend from 2007 to 2034.

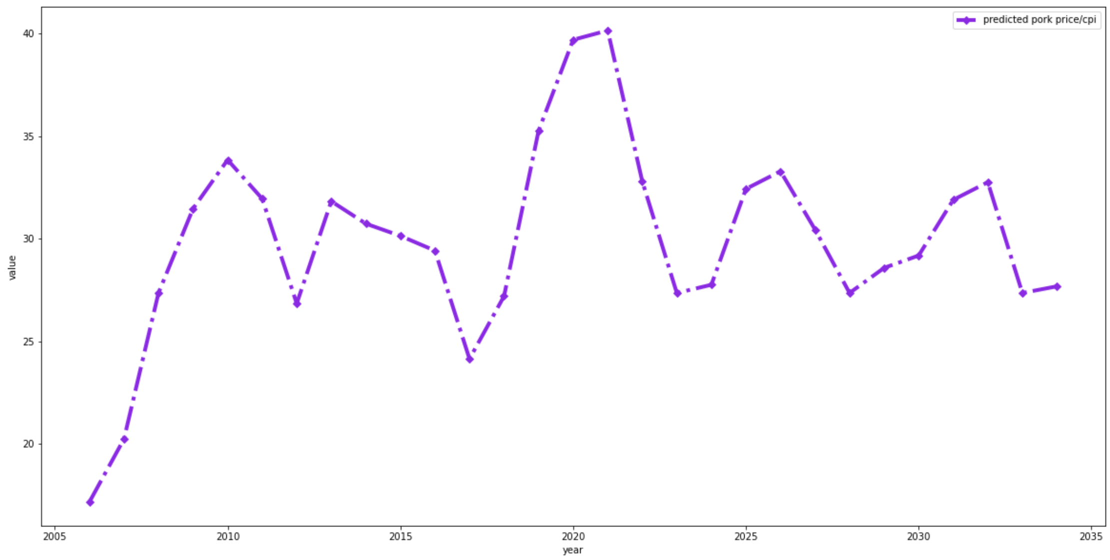


Figure 9: The simulation until year 2034

The pork price shows a 5-year periodic cycle. It matches the finding given by Gale and Hu in 2012.

5 Model validation

5.1 Empirical testing

Combining the real adjusted pork price and the predicted one from our model, we find that our model gives a proper fit. The correlation between these two data sets is 0.68.

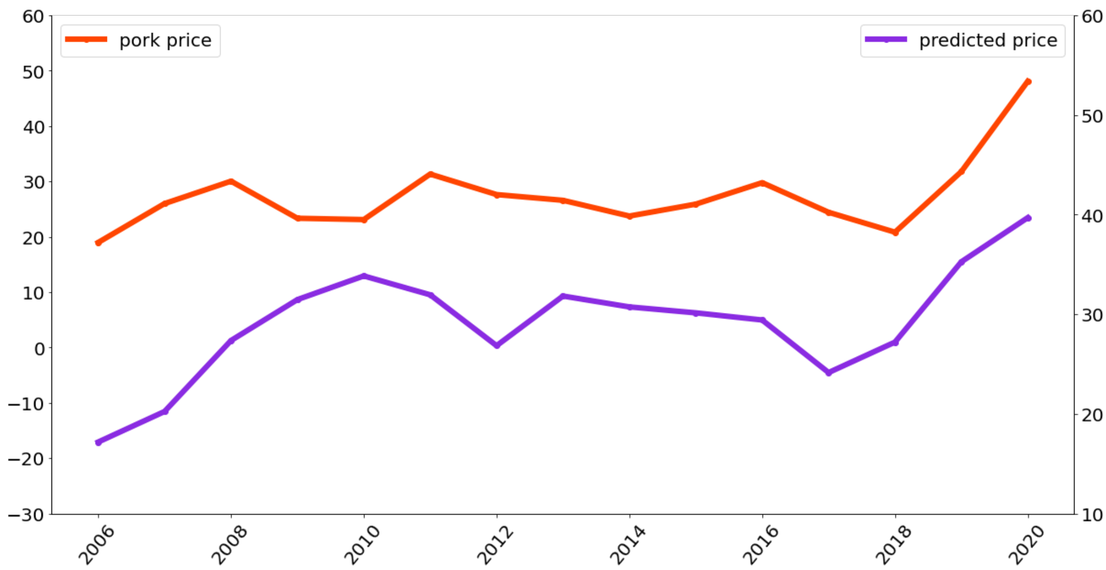


Figure 10: The comparison between the real historical data and the predicted value.

However, we find that this model needs to be better. Here are some drawbacks and their corresponding analysis.

1. Predicted price does not match the trend between 2008 and 2013. The correlation during this period is only -0.32.

The contrary trend from 2008 to 2013 might be due to our assumption about the age structure in 2006 (*Equation 8*). Since the adjustment on hog production has been conducted for decades, we cannot assume the age structure in 2006 follows the natural condition. From 2006 to 2013, the model adjusts itself to find the proper age structure fitting the periodic cycle. Thus, the model matches the real data well some years later while experiencing a bad fit initially.

2. The predicted price's mean value and standard deviation are around 29.7 and 4.69. These values differ from the true prices (27.42 and 6.82).

We think the supply-price transmission mechanism causes the difference in standard deviation. In *Figure 8*, the speed of price increase accelerates when the price decreases. Our linear function can't reflect this feature precisely. The bias on mean value might be owed to the hyper-parameter setting on the production adjustment model. These hyper-parameters cause the profit to break even when the price reaches around ￥29.7 per kilogram.

5.2 Sensitivity analysis

We did the sensitivity test on two hyper-parameters separately, the forage price pf and the price of female piglets c.

Forage price pf:

We add in another two conditions of forage price. One is the mean value minus standard (2.32), and the other is the mean value plus standard deviation (3.21). Here is the result:

Figure 11: The comparison of predicted price between three conditions of forage price

Patterns in different conditions of forage price are quite different. We find that the length of the price cycle increases with the increase in forage price—the line with the lowest forage price peaks earlier than the other two. The model is very sensitive to the change in forage price.

Price of female piglets c:

As the same manipulation on the above, we add in two conditions of piglet prices. The first one is 12.6, and the second one is 58.8. They are values calculated by adding or subtracting one standard deviation from the mean value of the female piglet price.

Figure 12: The comparison of predicted price between three conditions of piglet price

Compared to patterns in different forage prices, price patterns with different piglet prices are tidier. From 2006 to 2023, predicted prices present similar cycles among these three conditions. However, the cycle length of the green line (piglet price=58.8) gets shorter after 2023. The change on the orange line (piglet price=12.6) is contrary to that of the green line. Our model generally has lower sensitivity to the change in female piglet price.

We also notice that price trends between 2006 and 2013 are similar among these five conditions. Thus, the inverse trend between the actual and predicted prices from 2008 to 2013 can be caused by other factors in the model.

5.3 Directions of improvement

Based on the bias between our prediction and the actual situation, we propose the following improvement advice:

1. Since the production process of parity and the growth of a hog need half a year to complete, we can make the model more precise by simulating the process semiannually. We failed to find semiannual data in

this research, so we cannot build the model at semiannual frequency. We hope other researchers can manage this problem.

2. Using a polynomial function to model the hog supply-pork price mechanism is better. We tried this method in our modeling. However, this will cause an over-fitting problem, so we used the linear function instead. The over-fitting model violates the economic sense of this mechanism. This problem is due to our lack of a large amount of data. If other researchers can find more data on the amount of hogs supply and the corresponding pork price, they can try our original thought.

3. Making more suitable assumptions on the starting age structure. We used the age structure in the natural condition to infer the situation in 2006. We failed to collect any age structure data, so we had to make risky guessing. This realistic guessing causes the failure of our prediction between the years 2007 to 2013.

6 Conclusions

Generally speaking, our prediction fits the actual pork price to some degree. The result shows that predicting pork prices by simulating the behavior of pig farmers and the age structure of female pigs is possible. However, there are some drawbacks needed to be improved. We hope other researchers can dig deeper and construct a better model through our idea.

Our prediction also shows that the pork price will peak in 2026 and 2032. Because pork price is an indicator of inflation, there might be inflation in these two years. The government can take action in 2025 and 2031 to prevent this, like increasing the interest rate and releasing pork stocks. Our results can help govern the macroeconomy.

References

[1] Chen, G., Sui, Y., & Chen, S. (2014). Detection of flavor compounds in longissimus muscle from four hybrid pig breeds of Sus scrofa, Bamei pig, and Large White, *Bioscience Biotechnology and Biochemistry, 78* (11), 1910-1916

[2] Ezekiel, M. (1938). The cobweb theorem. *The quarterly journal of economics, 52*(2), 255-280.

[3] Fahmy, M. H., & Bernard, C. (1971). Causes of mortality in Yorkshire pigs from birth to 20 weeks of age. *Canadian Journal of Animal Science, 51*(2), 351-359.

[4] Gale, F., Hu, D., & Marti, D. (2012). China's volatile pork industry.

[5] Gao, Y., & Gu, Z. (2015). Simple calculation method of pig group structures, groups of pig population and turnover in intensive pig farm : example as a large scale pig farm for 10000 commercial fatting pigs annually. *Chinese Journal of Animal Science, 51*(16), 65-68.

[6] Huang, H., Wang, Y., Liu, Y., & Tao, H. (2022). Review of Chinese pork import and export market in 2021 and prospect in 2022. *China Swine Industry, 17*(1), 38-49.

[7] Jeremić, M., Lovre, K., & Matkovski, B. (2018). Serbian pork market analysis. Економика пољопривреде*, 65*(4), 1449-1460.

[8] Koketsu, Y. (2005). Within-farm variability in age structure of breeding-female pigs and reproductive performance on commercial swine breeding farms. *Theriogenology, 63*(5), 1256-1265.

[9] Talpaz, H. (1974). Multi-Frequency cobweb model: decomposition of the hog cycle. *American Journal of Agricultural Economics, 56*(1), 38-49.

[10] Wang, W., Wang, T., & Shi, Y. (2009, September). Factor analysis on consumer price index rising in China from 2005 to 2008. In *2009 International Conference on Management and Service Science* (pp. 1-4). IEEE.

[11] Zhao, G. Q., & Qiong, W. U. (2015). Nonlinear dynamics of pork price in China. *Journal of Integrative Agriculture, 14*(6), 1115-1121.

[12] Zhao, L., & Zhao, Y. (2018). Purchasing power parity and price fluctuations in China before July 1937. *Frontiers of Economics in China, 13*(3), 458-483.

Appendix

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import datetime as dt
import matplotlib.pyplot as plt
from matplotlib.pyplot import MultipleLocator
from sklearn.linear_model import LinearRegression
import statsmodels.api as sm
from scipy.optimize import minimize
import matplotlib.ticker as ticker
import scipy.stats
```

Data loading

In [2]:
```
#load in the pork price and calculate the monthly pork price
pork_price_original = pd.read_excel('22个省市_平均价_猪肉.xlsx')
pork_price_original.columns = ['date','pork_price']
pork_price_monthly = pork_price_original
pork_price_monthly['date'] = pd.to_datetime(pork_price_monthly['date'])
pork_price_monthly['date'] = pork_price_monthly['date'].dt.strftime('%Y-%m')
pork_price_monthly = pork_price_monthly.groupby(['date']).mean()
pork_price_monthly['date'] = pork_price_monthly.index
pork_price_monthly = pork_price_monthly.reset_index(drop=True)
pork_price_monthly
```

Out[2]:

	pork_price	date
0	11.159000	2006-07
1	13.000000	2006-09
2	12.791333	2006-10
3	13.370000	2006-11
4	14.845000	2006-12
...	...	...
191	28.266000	2022-07
192	29.047500	2022-08
193	31.128000	2022-09
194	35.675000	2022-10
195	35.050000	2022-11

196 rows × 2 columns

In [3]:
```
#load in the monthly cpi value
cpi = pd.read_excel('CPI_所有项目_中国.xlsx')
cpi.columns = ['date','cpi_value']
cpi['date'] = pd.to_datetime(cpi['date'])
cpi['date'] = cpi['date'].dt.strftime('%Y-%m')
cpi
```

Out[3]:

	date	cpi_value
0	1993-01	37.06709
1	1993-02	37.27705
2	1993-03	37.96701
3	1993-04	39.06402
4	1993-05	39.64885
...	...	...
352	2022-05	102.70000
353	2022-06	102.70000
354	2022-07	103.20000
355	2022-08	103.10000
356	2022-09	103.40000

357 rows × 2 columns

Here we use the monthly data because the cpi value is monthly. We want to use a more precise adjusted pork price. So, we first calculate the monthly adjusted pork price and then convert it to the yearly basis.

In [4]:
```
#calculate the annually average amount of sow
amount_female = pd.read_excel('生猪存栏_能繁母猪.xlsx')
amount_female.columns = ['date','total_amount_female']
amount_female['date'] = pd.to_datetime(amount_female['date'])
amount_female['date'] = amount_female['date'].dt.strftime('%Y')
amount_female = amount_female.groupby(['date']).mean()
amount_female['date'] = amount_female.index
amount_female = amount_female.reset_index(drop=True)
amount_female
```

Out[4]:

	total_amount_female	date
0	4887.583333	2009
1	4729.166667	2010
2	4788.666667	2011
3	4990.000000	2012
4	5010.083333	2013
5	4590.250000	2014
6	3932.666667	2015
7	3736.500000	2016
8	3557.500000	2017
9	3228.833333	2018
10	2316.333333	2019
11	3128.714286	2020
12	4419.375000	2021
13	4263.711111	2022

In [5]:
```
#the number of hog supplied per year
amount_sale = pd.read_excel('肉猪出栏头数.xlsx')
amount_sale.columns = ['date','total_amount_sale']
amount_sale['date'] = pd.to_datetime(amount_sale['date'])
amount_sale['date'] = amount_sale['date'].dt.strftime('%Y')
amount_sale
```

Out[5]:

	date	total_amount_sale
0	1996	41225.2
1	1997	46483.7
2	1998	50215.1
3	1999	51977.2
4	2000	51862.3
5	2001	53281.1
6	2002	54143.9
7	2003	55701.8
8	2004	57278.5
9	2005	60367.4
10	2006	61209.0
11	2007	56640.9
12	2008	61278.9
13	2009	64990.9
14	2010	67332.7
15	2011	67030.0
16	2012	70724.5
17	2013	72768.0
18	2014	74951.5
19	2015	72415.6
20	2016	70073.9
21	2017	70202.1
22	2018	69382.4
23	2019	54419.2
24	2020	52704.1

In [6]:

```
#load in the female piglet price
piggy_price = pd.read_excel('22个省市_平均价_仔猪.xlsx')
piggy_price.columns = ['date','piggy_price']
piggy_price['date'] = pd.to_datetime(piggy_price['date'])
piggy_price['date'] = piggy_price['date'].dt.strftime('%Y')
piggy_price = piggy_price.groupby(['date']).mean()
piggy_price['date'] = piggy_price.index
piggy_price = piggy_price.reset_index(drop=True)
piggy_price
```

Out[6]:

	piggy_price	date
0	12.715889	2006
1	23.605917	2007
2	32.865490	2008
3	20.802760	2009
4	19.436538	2010
5	31.899000	2011
6	30.310400	2012
7	27.057959	2013
8	24.240400	2014
9	31.729167	2015
10	51.050588	2016
11	40.002800	2017
12	25.521400	2018
13	56.680000	2019
14	107.994082	2020
15	62.001698	2021
16	38.596667	2022

In [7]:
```
#load in the monthly forage price
#forage_price = pd.read_excel('平均价_育肥猪配合饲料.xlsx')
forage_price = pd.read_excel('平均价_生猪饲料.xlsx')
forage_price.columns = ['date','forage_price']
forage_price['date'] = pd.to_datetime(forage_price['date'])
forage_price['date'] = forage_price['date'].dt.strftime('%Y-%m')
forage_price = forage_price.groupby(['date']).mean()
forage_price['date'] = forage_price.index
forage_price = forage_price.reset_index(drop=True)
forage_price
```

Out[7]:

	forage_price	date
0	2.656667	2015-06
1	2.736000	2015-07
2	2.667500	2015-08
3	2.605000	2015-09
4	2.532500	2015-10
...	...	...
85	3.520000	2022-07
86	3.524000	2022-08
87	3.687500	2022-09
88	3.860000	2022-10
89	3.900000	2022-11

90 rows × 2 columns

Data preprocessing and analysis

In [8]:
```
# calculate the monthly adjusted pork price
de_cpi_price = pd.merge(cpi,pork_price_monthly,how='inner')
de_cpi_price['de_price'] = (de_cpi_price['pork_price']/de_cpi_price['cpi_valu
e'])*100
de_cpi_price
```

Out[8]:

	date	cpi_value	pork_price	de_price
0	2006-07	67.95898	11.159000	16.420199
1	2006-09	68.50367	13.000000	18.977085
2	2006-10	68.57217	12.791333	18.653826
3	2006-11	68.77789	13.370000	19.439387
4	2006-12	69.74078	14.845000	21.285968
...	...	...	...	...
189	2022-05	102.70000	21.972500	21.394839
190	2022-06	102.70000	22.425000	21.835443
191	2022-07	103.20000	28.266000	27.389535
192	2022-08	103.10000	29.047500	28.174103
193	2022-09	103.40000	31.128000	30.104449

194 rows × 4 columns

In [9]:
```
#compare trends of adjusted price and the forage price
de_price_monthly = de_cpi_price[['date','de_price']]
forage_price = pd.merge(de_price_monthly,forage_price,how='inner')
forage_price
```

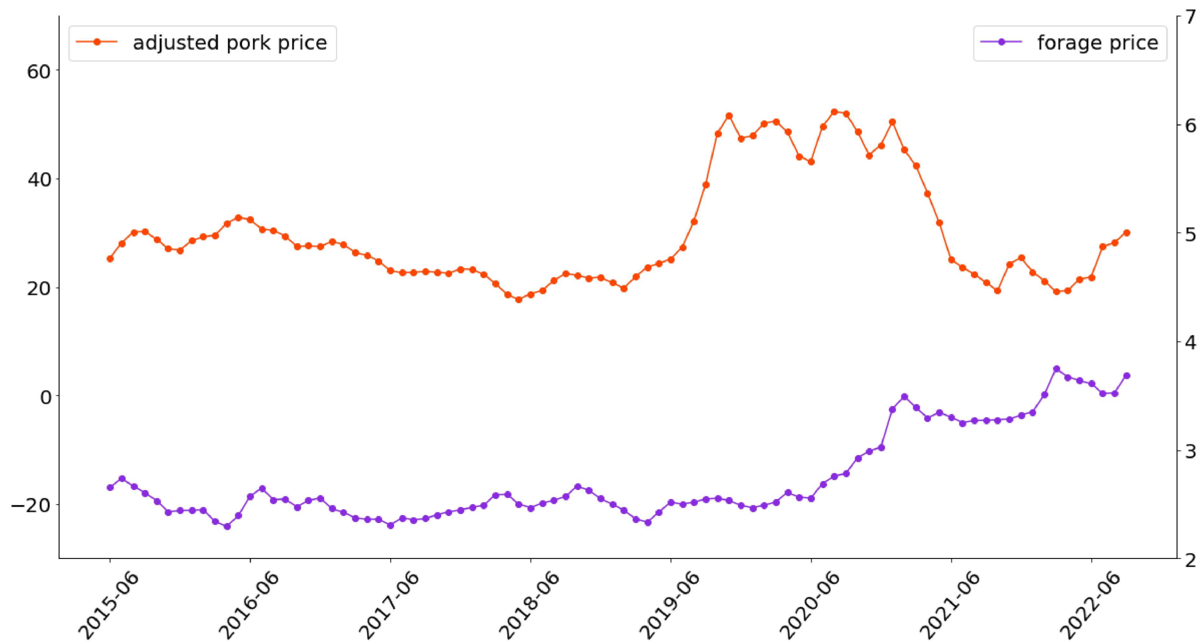
Out[9]:

	date	de_price	forage_price
0	2015-06	25.319280	2.656667
1	2015-07	28.075225	2.736000
2	2015-08	30.122443	2.667500
3	2015-09	30.200686	2.605000
4	2015-10	28.769642	2.532500
...	...	...	...
83	2022-05	21.394839	3.637500
84	2022-06	21.835443	3.610000
85	2022-07	27.389535	3.520000
86	2022-08	28.174103	3.524000
87	2022-09	30.104449	3.687500

88 rows × 3 columns

In [10]:
```
plt.figure(figsize=(20,10))
plt.ylim(-30,70)
plt.xticks(rotation=50)
ax=plt.gca()
ax.xaxis.set_major_locator(ticker.MultipleLocator(base=12))
ax.spines['top'].set_color('none')
plt.tick_params(labelsize=20)
ax=plt.gca()
ax.xaxis.set_major_locator(ticker.MultipleLocator(base=12))
ax.spines['top'].set_color('none')
plt.plot(forage_price['date'], forage_price['de_price'], color='orangered', ma
rker='o', linestyle='-', label='adjusted pork price')
l1 = plt.legend(loc=2,prop={'size':20})
ax2=ax.twinx()
ax2.plot(forage_price['date'], forage_price['forage_price'], color='blueviole
t', marker='o', linestyle='-', label='forage price')
ax2.set_ylim([2,7])
ax2.tick_params(labelsize=20)
ax=plt.gca()
ax.xaxis.set_major_locator(ticker.MultipleLocator(base=12))
ax.spines['top'].set_color('none')
plt.legend(loc=1,prop={'size':20})
```

Out[10]: <matplotlib.legend.Legend at 0x1c1670d8730>



In [11]:
```
forage_price['forage_price'].mean()
```

Out[11]: 2.7426704545454546

In [12]:
```
forage_price['forage_price'].std()
```

Out[12]: 0.4171817345022973

In [13]:
```
forage_price['de_price'].corr(forage_price['forage_price'])
```

Out[13]:
```
-0.002308801039743526
```

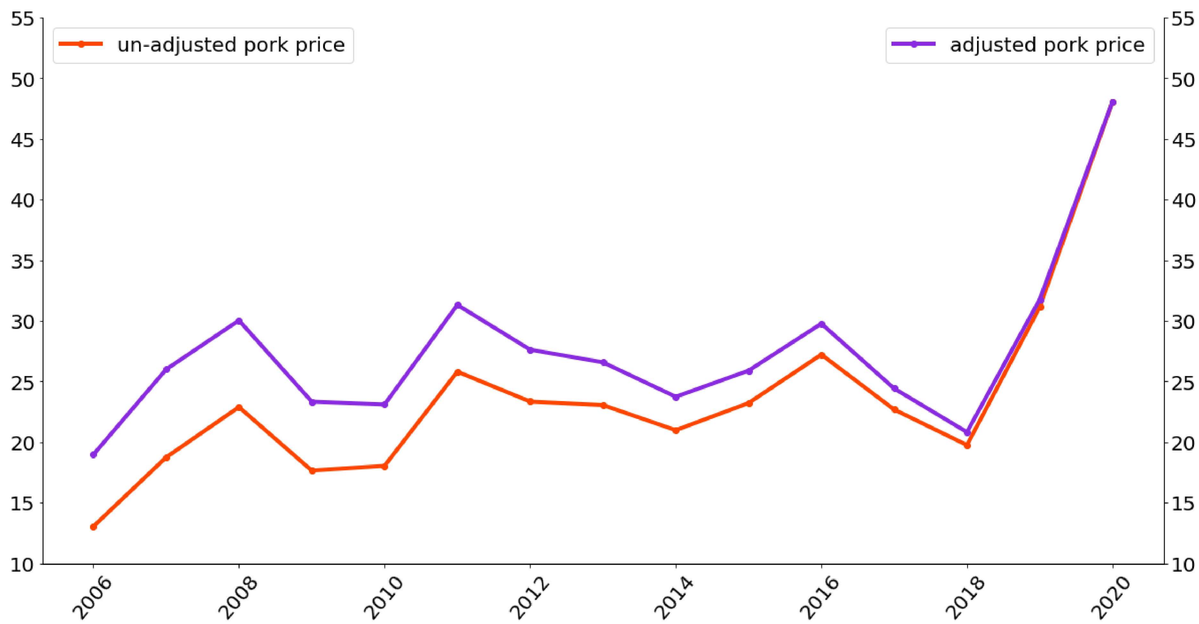
In [14]:
```
# calculate the annually adjusted pork price and put the number of hog supply
 to the table
de_cpi_price['date'] = pd.to_datetime(de_cpi_price['date'])
de_cpi_price['date'] = de_cpi_price['date'].dt.strftime('%Y')
de_cpi_price = de_cpi_price.groupby(['date']).mean()
de_cpi_price['date'] = de_cpi_price.index
de_cpi_price = de_cpi_price.reset_index(drop=True)
de_cpi_price = pd.merge(de_cpi_price,amount_sale,how='inner')
de_cpi_price
```

Out[14]:

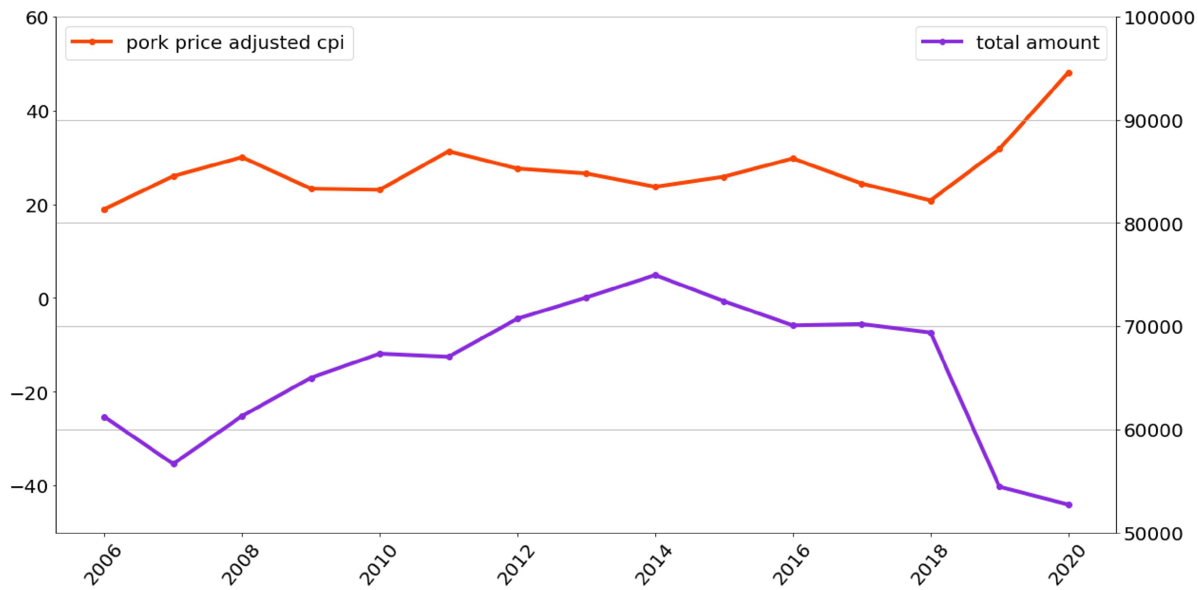
	cpi_value	pork_price	de_price	date	total_amount_sale
0	68.710698	13.033067	18.955293	2006	61209.0
1	71.901943	18.733004	25.984452	2007	56640.9
2	76.162313	22.879333	30.026348	2008	61278.9
3	75.607726	17.642937	23.327972	2009	64990.9
4	78.008517	18.031181	23.091733	2010	67332.7
5	82.341031	25.795333	31.303684	2011	67030.0
6	84.497974	23.331000	27.608342	2012	70724.5
7	86.712708	23.040417	26.566778	2013	72768.0
8	88.379016	20.972514	23.728157	2014	74951.5
9	89.649043	23.207708	25.881365	2015	72415.6
10	91.442026	27.207111	29.758112	2016	70073.9
11	92.898822	22.672861	24.408658	2017	70202.1
12	94.826278	19.744819	20.813744	2018	69382.4
13	97.575513	31.133708	31.761495	2019	54419.2
14	99.936277	48.061333	48.078896	2020	52704.1

In [15]:
```
# visualize the effect of cpi adjustment on the pork price
plt.figure(figsize=(20,10))
plt.ylim(10,55)
x_major_locator=MultipleLocator(2)
plt.xticks(rotation=50)
ax=plt.gca()
ax.xaxis.set_major_locator(x_major_locator)
ax.spines['top'].set_color('none')
plt.tick_params(labelsize=20)
ax=plt.gca()
ax.xaxis.set_major_locator(x_major_locator)
ax.spines['top'].set_color('none')
plt.plot(de_cpi_price['date'], de_cpi_price['pork_price'], color='orangered',
marker='o', linestyle='-', linewidth=4.0, label='un-adjusted pork price')
l1 = plt.legend(loc=2,prop={'size':20})
ax2=ax.twinx()
ax2.plot(de_cpi_price['date'], de_cpi_price['de_price'], color='blueviolet', m
arker='o', linestyle='-', linewidth=4.0, label='adjusted pork price')
ax2.set_ylim([10,55])
ax2.tick_params(labelsize=20)
ax=plt.gca()
ax.xaxis.set_major_locator(x_major_locator)
ax.spines['top'].set_color('none')
plt.legend(loc=1,prop={'size':20})
```

Out[15]: <matplotlib.legend.Legend at 0x1c1671b0fa0>



In [16]:
```
# visualize the relationship between number of supply and the price
plt.figure(figsize=(20,10))
plt.ylim(-50,60)
x_major_locator=MultipleLocator(2)
plt.xticks(rotation=50)
ax=plt.gca()
ax.xaxis.set_major_locator(x_major_locator)
ax.spines['top'].set_color('none')
plt.tick_params(labelsize=20)
ax=plt.gca()
ax.xaxis.set_major_locator(x_major_locator)
ax.spines['top'].set_color('none')
plt.plot(de_cpi_price['date'], de_cpi_price['de_price'], color='orangered', ma
rker='o', linestyle='-', linewidth=4.0, label='pork price adjusted cpi')
l1 = plt.legend(loc=2,prop={'size':20})
ax2=ax.twinx()
ax2.plot(de_cpi_price['date'], de_cpi_price['total_amount_sale'], color='bluev
iolet', marker='o', linestyle='-', linewidth=4.0, label='total amount')
ax2.set_ylim([50000,100000])
ax2.tick_params(labelsize=20)
ax=plt.gca()
ax.xaxis.set_major_locator(x_major_locator)
ax.spines['top'].set_color('none')
plt.legend(loc=1,prop={'size':20})
plt.grid()
```



In [17]:
```
# obtain the statistic of piglet price
piggy_price['piggy_price'].mean()
```

Out[17]: 37.44180907712788

In [18]:
```
piggy_price['piggy_price'].std()
```

Out[18]: 22.510019598469743

Solving the model

The hog supply-pork price transmission mechanism

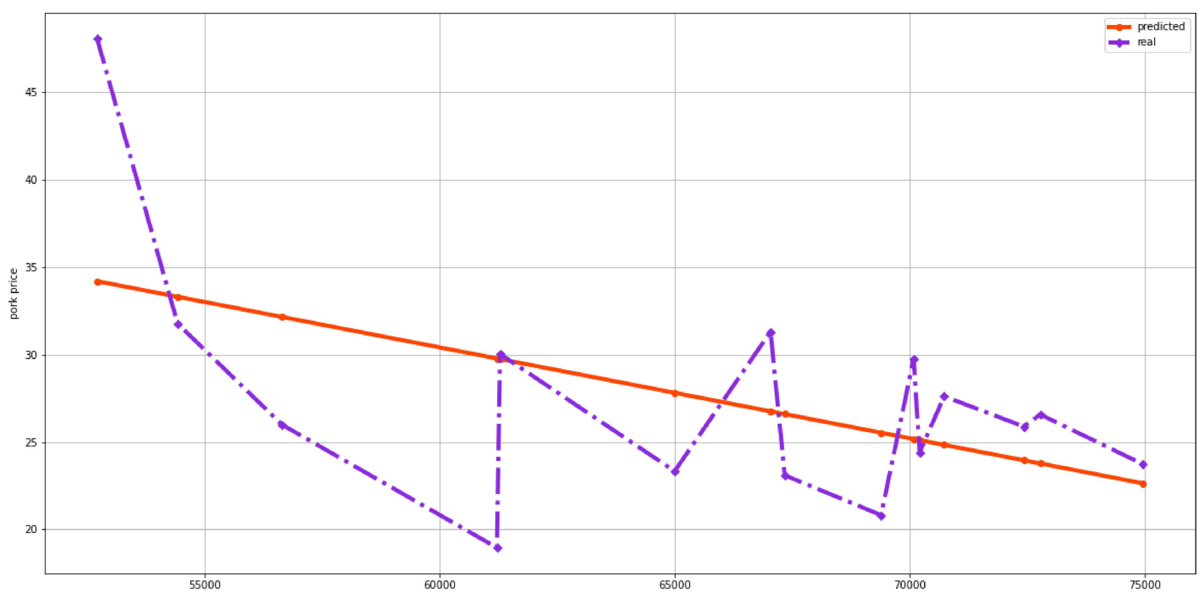
In [19]:
```
#regression to find the price function
de_cpi_price = de_cpi_price.sort_values(by='total_amount_sale')
x = de_cpi_price[['total_amount_sale']].values
y = de_cpi_price['de_price']

regr = LinearRegression()
regr.fit(x,y)

regr.coef_
```

Out[19]:
```
array([-0.00051957])
```

In [20]:
```
regr.intercept_
```

Out[20]:
```
61.577062399581195
```

In [21]:
```
#plot the regression
plt.figure(figsize=(20,10))
plt.plot(de_cpi_price['total_amount_sale'], regr.predict(x), color='orangered'
, marker='o', linestyle='-', linewidth=4.0, label='predicted')
plt.plot(de_cpi_price['total_amount_sale'], de_cpi_price['de_price'], color='b
lueviolet', marker='D', linestyle='-.', linewidth=4.0, label='real')
plt.legend()  # 显示图例
plt.xlabel("amount of pig per year")  # X轴标签
plt.ylabel("pork price")  # Y轴标签
plt.grid()
plt.show()
```



In [22]:
```
#evalutaing the regression
x2 = sm.add_constant(x)
est = sm.OLS(y,x2).fit()
est.summary()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\scipy\stats\stats.py:1603: UserWar
ning: kurtosistest only valid for n>=20 ... continuing anyway, n=15
  warnings.warn("kurtosistest only valid for n>=20 ... continuing "
```

Out[22]:

OLS Regression Results

Dep. Variable:	de_price	**R-squared:**	0.282
Model:	OLS	**Adj. R-squared:**	0.227
Method:	Least Squares	**F-statistic:**	5.116
Date:	Sun, 04 Dec 2022	**Prob (F-statistic):**	0.0415
Time:	19:24:45	**Log-Likelihood:**	-47.081
No. Observations:	15	**AIC:**	98.16
Df Residuals:	13	**BIC:**	99.58
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	61.5771	15.181	4.056	0.001	28.781	94.373
x1	-0.0005	0.000	-2.262	0.041	-0.001	-2.33e-05

Omnibus:	2.729	**Durbin-Watson:**	1.510
Prob(Omnibus):	0.256	**Jarque-Bera (JB):**	0.842
Skew:	0.451	**Prob(JB):**	0.656
Kurtosis:	3.730	**Cond. No.**	6.48e+05

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 6.48e+05. This might indicate that there are
strong multicollinearity or other numerical problems.

The age structure of female pigs and hogs

The age structure of female pigs

In [23]:
```python
#we obtain the survival rates and birth rates for all classes of female pigs
death_f_1 = (33/53)*0.9808+(20/53)*0.9638
birth_f_1 = 0.8/5
death_f_2 = (40/53)*0.969+(13/53)*0.9628
birth_f_2 = (53/20)*birth_f_1/2

fem_s_annual = [death_f_1]
fem_b_annual = [birth_f_1]

i = 2
while i<= 5:
    fem_s_annual.append(death_f_2)
    fem_b_annual.append(birth_f_2)
    i+=1

#the survival and birth matrix L of female pigs
fem_b_annual_c = np.array([fem_b_annual])*death_f_1
diagnal_f_annual = np.diag(fem_s_annual[1:])
zero_column_f_annual = np.zeros((diagnal_f_annual.shape[0],1))
lower_annual = np.c_[diagnal_f_annual,zero_column_f_annual]
L_female_annual = np.r_[fem_b_annual_c,lower_annual]
L_female_annual
```

Out[23]:
```
array([[0.15590158, 0.2065696 , 0.2065696 , 0.2065696 , 0.2065696 ],
       [0.96747925, 0.        , 0.        , 0.        , 0.        ],
       [0.        , 0.96747925, 0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.96747925, 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.96747925, 0.        ]])
```

In [24]:
```python
# we analyze the age structure of female pigs during the natural condition
eigen_value = np.linalg.eig(L_female_annual)
structure = fem_s_annual/eigen_value[0][0]
structure
```

Out[24]:
```
array([1.00224802+0.j, 0.99514489+0.j, 0.99514489+0.j, 0.99514489+0.j,
       0.99514489+0.j])
```

In [25]:
```python
# we assume the age structure of female pigs in 2006 follows the natural condi
tion
average_female_second_end = 4887.583333
amount_weighted = []
for i in range(5):
    amount_weighted.append(0.99514489**i)
amount_weighted = np.array(amount_weighted)
amount_mature = sum(amount_weighted[1:])
factor = average_female_second_end/amount_mature
initial_female = amount_weighted*factor
initial_female
# this is the population of each age class in 2005
```

Out[25]:
```
array([1236.83555468, 1230.83058201, 1224.85476415, 1218.90795953,
       1212.99002731])
```

In [26]:
```
# calculate the population of female pigs in 2006
initial_female_2006 = L_female_annual@initial_female
initial_female_2006
```

Out[26]:
```
array([1202.45075731, 1196.61272898, 1190.80304256, 1185.0215628 ,
       1179.26815276])
```

The age structure of hogs

In [27]:
```
# calculate the age vector of hogs
survival_hog = 0.824*0.973
birth_s_1 = 0.8*0.95*10
birth_s_2 = birth_s_1*(53/20)
birth_hog = np.array([birth_s_1,birth_s_2,birth_s_2,birth_s_2,birth_s_2])
birth_hog = birth_hog-np.array(fem_b_annual)
birth_hog
```

Out[27]:
```
array([ 7.44 , 19.928, 19.928, 19.928, 19.928])
```

In [28]:
```
#calculate the amount of hogs in 2006
hog_initial_2006 = sum(birth_hog*initial_female)*survival_hog
hog_initial_2006
```

Out[28]:
```
85468.2201332038
```

The supply adjustment

In [29]:
```
#define the function to calculate the profit
def profit(hog_n,female_n,last_price):
    revenue = hog_n*70*last_price
    cost = (hog_n+sum(female_n))*2.71*1.5*365+0.2*revenue
    profit = revenue - cost
    return profit
```

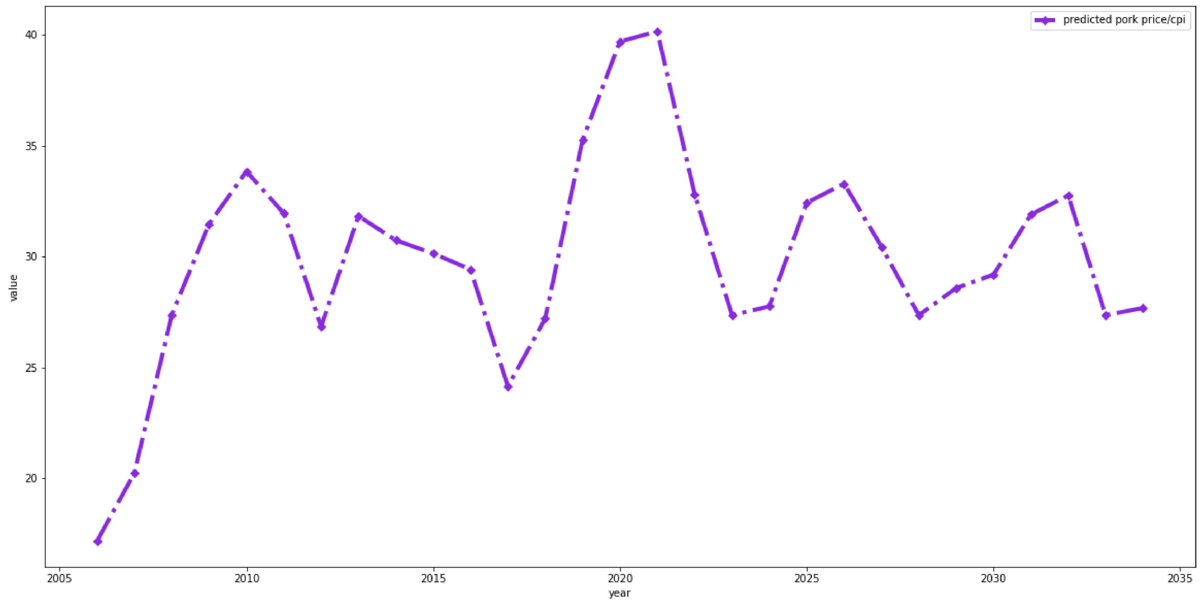
In [30]:
```
#optimal problem to find the adjustment for each year
def supply(female_begin,last_pr,last_profit):

    fun = lambda x : (-(sum(birth_hog*female_begin)+birth_hog[0]*x)*70*regr.pr
edict(np.array([(sum(birth_hog*female_begin)+x*birth_hog[0])]))-((x*birth_hog[
0]+x*L_female_annual[0,0])*2.71*1.5*365+x*30*37.4)) # 目标函数
    cons = (
            {'type': 'ineq', 'fun': lambda x: x+0.8*female_begin[0]},
            {'type': 'ineq', 'fun': lambda x: -x+0.8*female_begin[0]},
            {'type': 'ineq', 'fun': lambda x: last_profit-((x*birth_hog[0]+x*L
_female_annual[0,0])*2.71*1.5*365+x*30*37.4)},
           )

    x0 = 500 # 设置初始值(随机设置即可)

    res = minimize(fun, x0, method='SLSQP', constraints=cons) # 调用最小值模块
    solution = res.x
    return solution
```

The simulation

In [31]:
```
#simulate the whole process from 2009 to 2022

price_2006 = regr.predict(np.array([[hog_initial_2006]]))[0]
profit_2006 = profit(hog_initial_2006.flatten(order='A'),initial_female.flatte
n(order='A'),price_2006)

year = [2006]
female = [initial_female.flatten(order='A')]
hog = [hog_initial_2006]
price = [price_2006]
profit_value = [profit_2006]
input_year = []

for i in range(2007,2035):
    new_input = supply(female[i-2007],price[i-2007],profit_value[i-2007])
    female_begin = female[i-2007]
    female_begin[0] = female_begin[0]+new_input

    female_this_year = L_female_annual@female_begin
    hog_this_year = sum(birth_hog*female_begin)*survival_hog
    price_this_year = regr.predict(np.array([[hog_this_year]]))[0]
    profit_this_year = profit(hog_this_year,female_this_year,price_this_year)-
new_input*30*3.5

    input_year.append(new_input)
    female.append(female_this_year)
    hog.append(hog_this_year)
    price.append(price_this_year)
    profit_value.append(profit_this_year)
    year.append(i)
```

In [32]:

```
# draw the simulation result
plt.figure(figsize=(20,10))
plt.plot(year, price, color='blueviolet', marker='D', linestyle='-.', linewidt
h=4.0, label='predicted pork price/cpi')
plt.legend()  # 显示图例
plt.xlabel("year")  # X轴标签
plt.ylabel("value")  # Y轴标签
plt.show()
```



In [33]:

```
# statistic of the prediction
np.array(price).mean()
```

Out[33]: 29.869599634469196

In [34]:

```
np.array(price).std()
```

Out[34]: 4.689777002702038

In [35]:
```
# create table containing prediction and actual data
predict_price = pd.DataFrame({'date':year,'predict price':price})
de_cpi_price['date'] = de_cpi_price['date'].astype(int)
result = pd.merge(de_cpi_price,predict_price,how='inner')
result = result.sort_values(by='date')
result = result.reset_index(drop=True)
result
```

Out[35]:

	cpi_value	pork_price	de_price	date	total_amount_sale	predict price
0	68.710698	13.033067	18.955293	2006	61209.0	17.170286
1	71.901943	18.733004	25.984452	2007	56640.9	20.236903
2	76.162313	22.879333	30.026348	2008	61278.9	27.356648
3	75.607726	17.642937	23.327972	2009	64990.9	31.463961
4	78.008517	18.031181	23.091733	2010	67332.7	33.838838
5	82.341031	25.795333	31.303684	2011	67030.0	31.946000
6	84.497974	23.331000	27.608342	2012	70724.5	26.839736
7	86.712708	23.040417	26.566778	2013	72768.0	31.820656
8	88.379016	20.972514	23.728157	2014	74951.5	30.730221
9	89.649043	23.207708	25.881365	2015	72415.6	30.134670
10	91.442026	27.207111	29.758112	2016	70073.9	29.416102
11	92.898822	22.672861	24.408658	2017	70202.1	24.142018
12	94.826278	19.744819	20.813744	2018	69382.4	27.197893
13	97.575513	31.133708	31.761495	2019	54419.2	35.259044
14	99.936277	48.061333	48.078896	2020	52704.1	39.688606

In [36]:
```
# calculate the correlation of prediction and the actual value
result['de_price'].corr(result['predict price'])
```

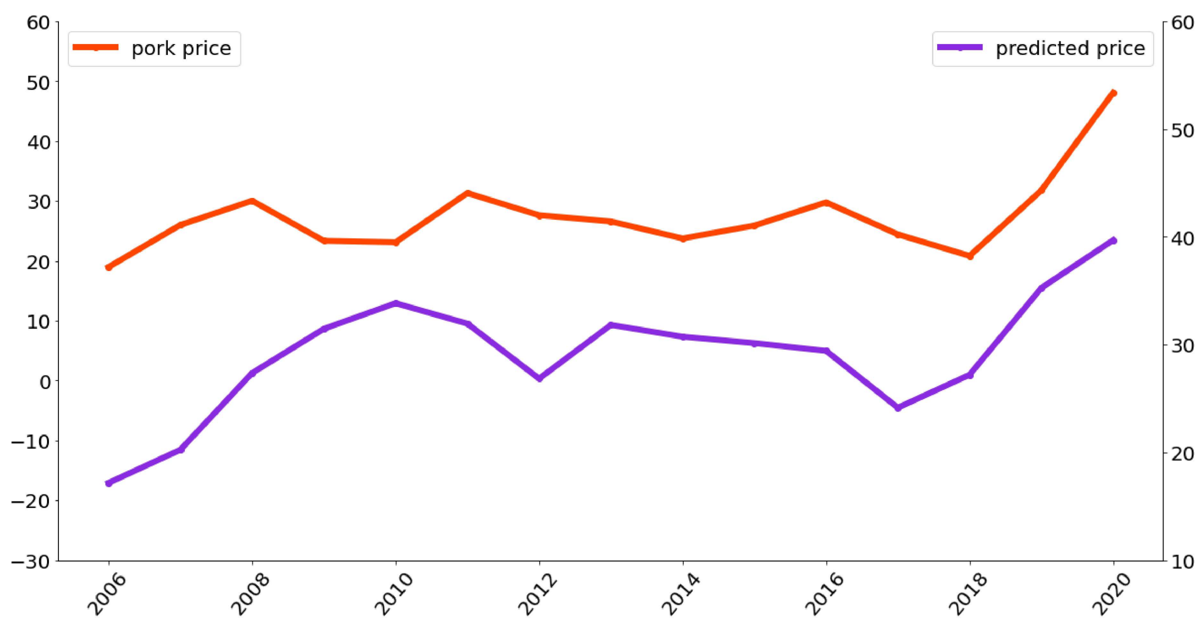
Out[36]: 0.64928706319606

In [37]:
```
# calculate the correlation during the bad fitting
result['de_price'][3:7].corr(result['predict price'][3:7])
```

Out[37]: -0.32136671719143484

In [38]:
```
# visualizing the result
plt.figure(figsize=(20,10))
plt.ylim(-30,60)
x_major_locator=MultipleLocator(2)
plt.xticks(rotation=50)
ax=plt.gca()
ax.xaxis.set_major_locator(x_major_locator)
ax.spines['top'].set_color('none')
plt.tick_params(labelsize=20)
ax=plt.gca()
ax.xaxis.set_major_locator(x_major_locator)
ax.spines['top'].set_color('none')
plt.plot(result['date'], result['de_price'], color='orangered', marker='o', li
nestyle='-', linewidth=6.0, label='pork price')
l1 = plt.legend(loc=2,prop={'size':20})
ax2=ax.twinx()
ax2.plot(result['date'], result['predict price'], color='blueviolet', marker=
'o', linestyle='-', linewidth=6.0, label='predicted price')
ax2.set_ylim([10,60])
ax2.tick_params(labelsize=20)
ax=plt.gca()
ax.xaxis.set_major_locator(x_major_locator)
ax.spines['top'].set_color('none')
plt.legend(loc=1,prop={'size':20})
```

Out[38]: <matplotlib.legend.Legend at 0x1c1684a4d90>



Sensitivity analysis

In [39]:

```
# set the sensitivity testing funtion on the forage price
def supply_forage_test(female_begin,last_pr,last_profit, forage_price):

    fun = lambda x : (-(sum(birth_hog*female_begin)+birth_hog[0]*x)*70*regr.pr
edict(np.array([(sum(birth_hog*female_begin)+x*birth_hog[0])]))-((x*birth_hog[
0]+x*L_female_annual[0,0])*forage_price*1.5*365+x*30*37.4)) # 目标函数
    cons = (
            {'type': 'ineq', 'fun': lambda x: x+0.8*female_begin[0]},
            {'type': 'ineq', 'fun': lambda x: -x+0.8*female_begin[0]},
            {'type': 'ineq', 'fun': lambda x: last_profit-((x*birth_hog[0]+x*L
_female_annual[0,0])*forage_price*1.5*365+x*30*37.4)},
           )

    x0 = 500 # 设置初始值(随机设置即可)

    res = minimize(fun, x0, method='SLSQP', constraints=cons) # 调用最小值模块
    solution = res.x
    return solution


#sensitivity analysis on forage price
forage_test = [2.32, 3.21]

for m in forage_test:
    price_2006 = regr.predict(np.array([[hog_initial_2006]]))[0]
    profit_2006 = profit(hog_initial_2006.flatten(order='A'),initial_female.fl
atten(order='A'),price_2006)

    year = [2006]
    female = [initial_female.flatten(order='A')]
    hog = [hog_initial_2006]
    price = [price_2006]
    profit_value = [profit_2006]
    input_year = []

    for i in range(2007,2035):
        new_input = supply_forage_test(female[i-2007],price[i-2007],profit_val
ue[i-2007],m)
        female_begin = female[i-2007]
        female_begin[0] = female_begin[0]+new_input

        female_this_year = L_female_annual@female_begin
        hog_this_year = sum(birth_hog*female_begin)*survival_hog
        price_this_year = regr.predict(np.array([[hog_this_year]]))[0]
        profit_this_year = profit(hog_this_year,female_this_year,price_this_ye
ar)-new_input*30*3.5

        input_year.append(new_input)
        female.append(female_this_year)
        hog.append(hog_this_year)
        price.append(price_this_year)
        profit_value.append(profit_this_year)
        year.append(i)

    title = 'forage' + str(m)
    predict_price[title] = price
```

```python
In [40]: # set the sensitivity testing funtion on the female piglet price
def supply_piggy_test(female_begin,last_pr,last_profit, piggy_price):

    fun = lambda x : (-(sum(birth_hog*female_begin)+birth_hog[0]*x)*70*regr.pr
edict(np.array([(sum(birth_hog*female_begin)+x*birth_hog[0])]))-((x*birth_hog[
0]+x*L_female_annual[0,0])*2.71*1.5*365+x*30*piggy_price)) # 目标函数
    cons = (
            {'type': 'ineq', 'fun': lambda x: x+0.8*female_begin[0]},
            {'type': 'ineq', 'fun': lambda x: -x+0.8*female_begin[0]},
            {'type': 'ineq', 'fun': lambda x: last_profit-((x*birth_hog[0]+x*L
_female_annual[0,0])*2.71*1.5*365+x*30*piggy_price)},
           )

    x0 = 500 # 设置初始值(随机设置即可)

    res = minimize(fun, x0, method='SLSQP', constraints=cons) # 调用最小值模块
    solution = res.x
    return solution

#sensitivity analysis on piglet price
piggy_test = [12.6, 58.8]

for m in piggy_test:
    price_2006 = regr.predict(np.array([[hog_initial_2006]]))[0]
    profit_2006 = profit(hog_initial_2006.flatten(order='A'),initial_female.fl
atten(order='A'),price_2006)

    year = [2006]
    female = [initial_female.flatten(order='A')]
    hog = [hog_initial_2006]
    price = [price_2006]
    profit_value = [profit_2006]
    input_year = []

    for i in range(2007,2035):
        new_input = supply_piggy_test(female[i-2007],price[i-2007],profit_valu
e[i-2007],m)
        female_begin = female[i-2007]
        female_begin[0] = female_begin[0]+new_input

        female_this_year = L_female_annual@female_begin
        hog_this_year = sum(birth_hog*female_begin)*survival_hog
        price_this_year = regr.predict(np.array([[hog_this_year]]))[0]
        profit_this_year = profit(hog_this_year,female_this_year,price_this_ye
ar)-new_input*30*3.5

        input_year.append(new_input)
        female.append(female_this_year)
        hog.append(hog_this_year)
        price.append(price_this_year)
        profit_value.append(profit_this_year)
        year.append(i)

    title = 'piggy' + str(m)
    predict_price[title] = price
```

In [41]:
```
# visualize the sensitivity test on the forage price
plt.figure(figsize=(20,10))
plt.plot(predict_price['date'], predict_price['predict price'], marker='D', linewidth=4.0, label='initial condition: forage price=2.71')
plt.plot(predict_price['date'], predict_price['forage2.32'], marker='D', linewidth=4.0, label='forage price=2.32')
plt.plot(predict_price['date'], predict_price['forage3.21'], marker='D', linewidth=4.0, label='forage price=3.21')
plt.legend()  # 显示图例
plt.xlabel("year")  # X轴标签
plt.ylabel("value")  # Y轴标签
plt.show()
```

In [42]:

```
# visualize the sensitivity test on the female piglet price
plt.figure(figsize=(20,10))
plt.plot(predict_price['date'], predict_price['predict price'], marker='D', linewidth=4.0, label='initial condition: piglet price=37.4')
plt.plot(predict_price['date'], predict_price['piggy12.6'], marker='D', linewidth=4.0, label='piglet price=12.6')
plt.plot(predict_price['date'], predict_price['piggy58.8'], marker='D', linewidth=4.0, label='piglet price=58.8')
plt.legend()  # 显示图例
plt.xlabel("year")  # X轴标签
plt.ylabel("value")  # Y轴标签
plt.show()
```