《数学建模基础创新实践 1》

实验大作业

题目：为亚运代言公司的某商品定价问题

学院：卓越学院

队员(学号姓名):

1. ________22330421 于依晗________
2. ________22050626 王文杰________
3. ________22180105 徐堰涛________

基于线性回归模型的亚运商品定价分析

摘要

杭州 2022 第 19 届亚运会申办委员会在筹办过程中实施市场开发计划，某公司为竞标相关代言对某商品做出一次评估。针对在盈利最大情况下确定商品定价这一问题，拟合近年数据得到相关参数，构建了常微分方程模型中的线性回归模型，利用最小二乘法，确定不同情景下的最高定价。该问题的研究能优化代言方案，扩大利润，增加企业实力及加大对亚运会的支持力度。

针对问题一，经过简单的散点图构造，观察数据特点，引入**线性回归模型**，将近年来的销售情况代入计算，利用最小二乘法拟合求数据，解出售价一次项系数 a 为-0.2059，常数项为 b 13.15，此种情况下拟合得出直线 z_1=-0.2059x+13.15 ，经过误差分析可知拟合程度较好，能较为准确地反映售价和销售量的关系，可确定销售总收入不低于最初收入时该商品的最高定价约为 38.91 元，销售总收入达到最高时商品定价约为 31.93 元。为确认模型的稳定性，在后续处理过程中展开对一次项系数 a 的**灵敏度**分析。

针对问题二，因疫情公司实行技术革新和宣传策略创新，在固定宣传费 50 万元的基础上拟投入技术革新费 y_1，补充浮动宣传费 y_2，在销量和售价的关系基础上，得出销售总收入与销量和售价的数学关系，构建线性回归模型，结合拟投入技术革新费用和浮动宣传费与商品价格之间的关系，将得出的之后总盈利 c_2 与问题一背景下的盈利 c_1 比较，构建动态模型初步得到结果，结合实际情况进行分析，排除不合理取值范围，据此通过 Scipy 中的函数计算出合理的范围并对图像进行修正，得到所需要超过的销量和对应定价的关系为 x*z2-0.16x2-0.2x+50⩾x(ax+b) 以及取等时的图像为图 9。结合图 8，在合理范围内比较极值点与交点 x 坐标的关系，得出在合理范围内可以获得最大盈利和不能获得最大盈利的情况，并计算得到最大盈利可能性 η =2.4%，为确认模型的稳定性，在后续处理过程中展开对 k_1，k_2 进行灵敏度分析。

本文的特点在于对问题一进行了**误差分析**，提高了模型的准确性，更符合实际情况，并在问题二中抓住图像之间的关系，在清晰的**动态**分析过程中展示求解过程，并直观求解出最大盈利的可能性。

关键词： 线性回归模型 动态模型 误差分析 灵敏度

一、 问题重述

1.1 问题背景

杭州成功取得亚运会举办权，绵延了中国红的精神力量，在充盈杭州精神文化的同时，带动了杭州各行各业和各公司的共同发展，增强了社会活力。

杭州 2022 年第 19 届亚运会的申办委员会为成功举办杭州亚运会，提高杭州亚运会的品牌价值，增强社会对杭州亚运会的关注度和参与度实施市场开发计划。某公司抓住亚运契机，扩大某商品影响力，增加利润，因此该公司事先对这一进军亚运的商品做出评估，探求商品盈利与商品定价之间的关系。

1.2 问题重述

1.已知该商品初售价 25 元/件时，年销售量为 8 万件，根据近年商品售价与销量统计数据分析关系，构建数学模型，当销售总收入不低于最初收入时确定商品最高定价，并求出商品总收入最高时的商品定价。

2. 受疫情影响，公司启动一轮创新，在固定宣传费 50 万元基础上，已知拟投入技术革新费用（单位：万元）与商品定价的平方呈 $y_1=k_1x^2+b$（$k_1=0.16$，$b=-100$）的线性关系，补充浮动宣传费（单位：万元）与商品价格呈 $y_2=k_2x(k_2=0.2)$的正比例关系，构建数学模型，确定销量超出多少件能比问题一的背景更加盈利，并确定此时定价，同时分析获利最大的可能性。

二、 问题分析

本文要解决商品如何盈利最大的问题，问题一要求只从定价和销售额的关系，求得收入达到最高时的定价，问题二要求在宣传政策创新和技术革新的情况下，提高盈利，并分析获得最大盈利的可能性。

2.1 问题一的分析

问题一中给出了商品初售价和年销量的数据，由于商品售价和年销量的关系相对稳定，通过数据分析散点图可以进一步确认售价与销量存在着线性关系。由此，可以考虑构建常微分方程模型中的线性回归模型对销售总收入进行分析。

2.2 问题二的分析

问题二中设定了宣传和技术革新费用，考虑两者与售价分别存在线性关系和非线性关系，选择线性回归模型刻画最终盈利与售价的关系，并比较其与问题一背景下的盈利情况，进一步选择使用动态模型得到所需要超过的销量和对应定价的关系。

三、 模型假设

1. 假设问题一背景下售价和销量保持线性关系
2. 假设商品销售所得收入减去技术革新费用和宣传费用即为商品盈利
3. 假设销量只受定价和技术革新费用以及宣传费用的影响

4. 假设不存在经济危机、生产事故和其他同类商品竞争影响销售量

四、 符号说明

表 1 符号系统

变量	变量描述	单位
x	售价	元
z_1	之前销量	万件
c_1	销量总收入	万元
x_{max}	最高定价	元
x_0	总收入最高时定价	元
y	总投入	万元
z_2	后来销量	万件
z_q	合理销量左边界	万元
z_h	合理销量右边界	万元
c_2	总盈利	万元
η	最大盈利可能性	
x_i	交点横坐标	元
x_j	对称轴横坐标	元
z_0	可否取得最大盈利的临界值	万元

五、 模型的建立与求解

5.1 问题一的建模与求解

5.1.1 模型准备

我们利用 Matplotlib 绘制销售量与售价之间的散点图如下：

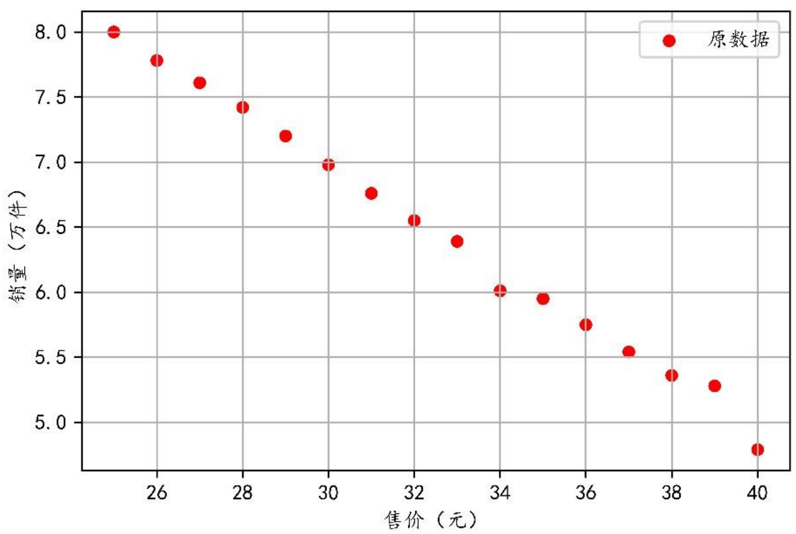

图 1 商品售价和销量关系散点图

分析以上散点图可以得知，商品销量与售价符合线性关系，在此我们可以使用一次

函数拟合数据并进一步建立线性回归模型。

5.1.2 线性回归模型的建立

假设售价与销量的关系为

$$z_1=ax+b \tag{1}$$

其中 a 为一次项系数，b 为常数项，x 为售价，z_1 为销量

根据最小二乘法通过多项式拟合，并使用 Matplotlib 绘制图像得到：

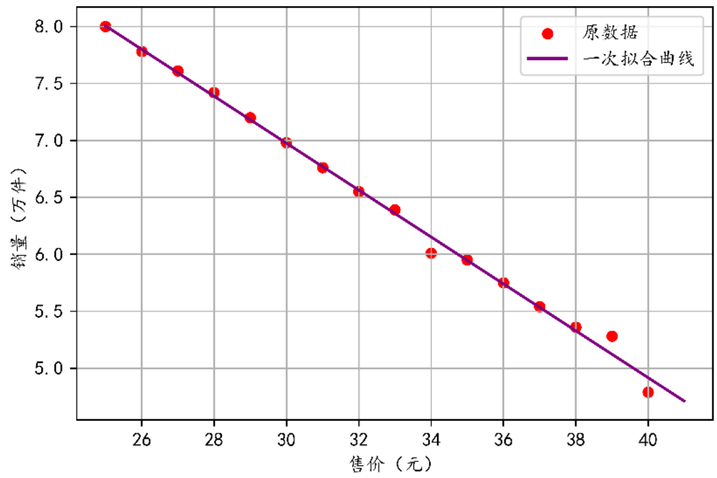

图 2 商品售价和销售量拟合曲线

求得 a、b 的估算值为：

$$a= -0.2059 \quad b= 13.15 \tag{2}$$

将（2）代入（1），所得售价和销量关系为

$$z_1=-0.2059x+13.15 \tag{3}$$

假设销售总收入为 c_1，结合（3）可得

$$c_1=（-0.2059x+13.15）x \tag{4}$$

绘制图像如下：

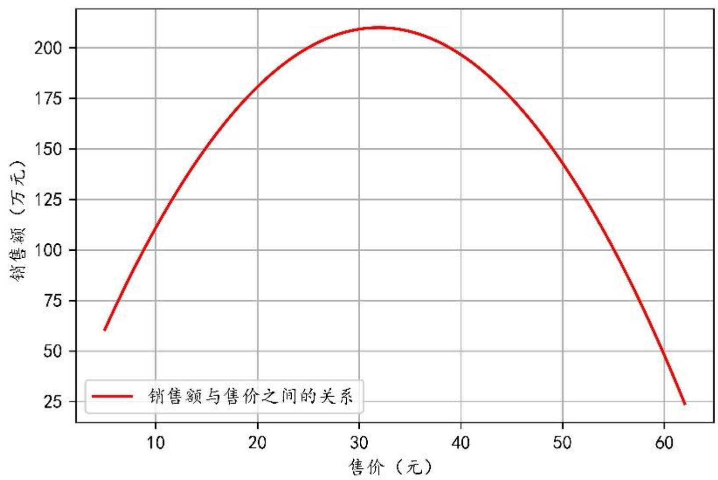

图 3 销售额与售价的关系

5.1.3 线性回归模型的求解

已知最初销售额为 200 万元，需确定销售总收入不低于最初销售额的取值范围

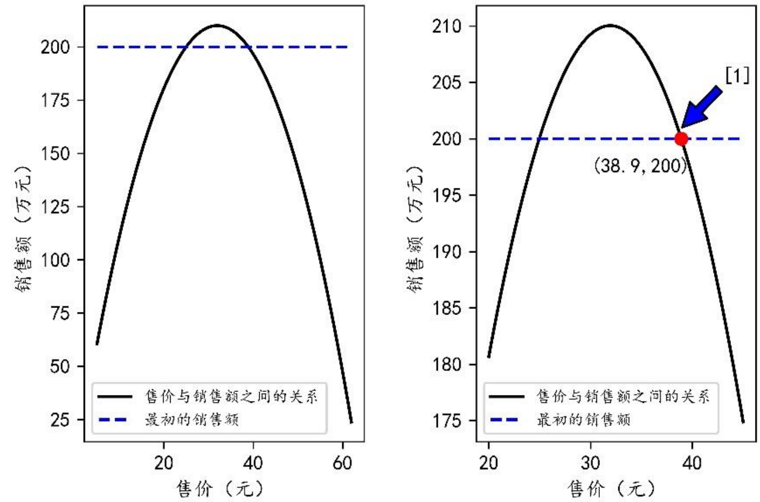

图 4 对比图

为便于观察，放大左图得右图，分析右图可知，满足条件的最高定价为[1]

为进一步确定商品的销售总收入达到最高值的定价，通过销售额对售价取微分得到销售额关于售价的导函数，该函数零点即为销售总收入达到最高时的商品定价，如下：

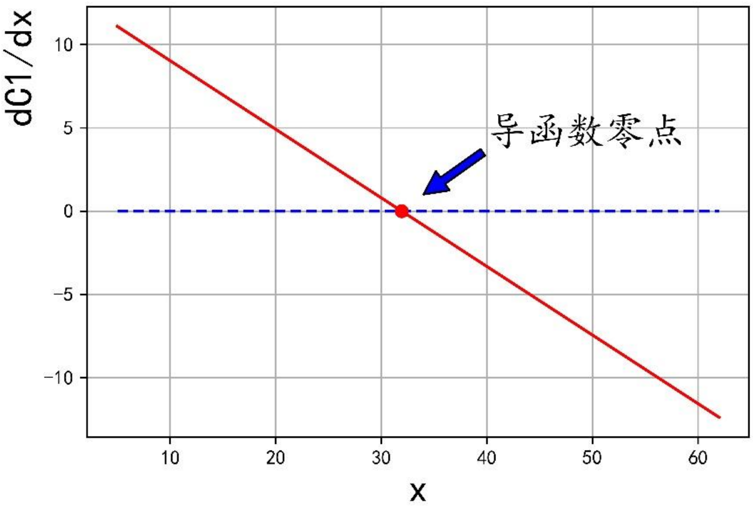

图 5 销售额关于售价的导函数图

5.1.4 结果分析

根据图 4，利用 Scipy 求得商品的最高定价：

$$x_{max}=38.91 \tag{5}$$

根据图 5，当销售总收入达到最高时，商品定价为：

$$x_0= 31.93 \tag{6}$$

5.1.5 模型检验

根据（4），可计算 x=25，30，35 时的销售额：

当 x=25 时

$$c_1（25）=（-0.2059*25+13.15）*25=200.1 \tag{7}$$

当 x=30 时

$$c_1（30）=（-0.2059*30+13.15）*30=209.3 \tag{8}$$

当 x=35 时

$$c_1（35）=（-0.2059*35+13.15）*35=208.1 \tag{9}$$

其相对误差分别为

$$\left|\frac{200.1-200}{200}\right|=0.05\% \tag{10}$$

$$\left|\frac{209.3-30*6.98}{30*6.98}\right|=0.048\% \tag{11}$$

$$\left|\frac{208.1-35*5.95}{35*5.95}\right|=0.072\% \tag{12}$$

观察数据发现，误差较小，说明模型较符合实际情况，从而证明计算结果的准确性。

5.2 问题二的建模与求解

5.2.1 模型准备

假设 c_2 为公司启动技术革新和宣传时的总盈利，z_2 为此时的销售量，y 为投入的总宣传费和技术革新费的总和，结合问题一的思路，构建线性回归模型，可得：

$$c_2=x*z_2-y_1-y_2-50 \tag{13}$$

由题意可得，拟投入技术革新费用跟商品价格平方的关系为：

$$y_1=k_1x_2^2+b \tag{14}$$

已知 $k_1=0.16$，b=-100

补充浮动宣传费和商品价格的关系为：

$$y_2=k_2x \tag{15}$$

已知 $k_2=0.2$

因此时较问题一背景下的情况更盈利，即在售价相同的情况下：

$$c_2>c_1 \tag{16}$$

5.2.2 模型建立

结合（3）（4）（13）（14）（15）可得:

$$x*z_2-0.16x^2-0.2x+50\geqslant x(ax+b) \tag{17}$$

分析可得，（17）取等时为临界条件，据此画图得：

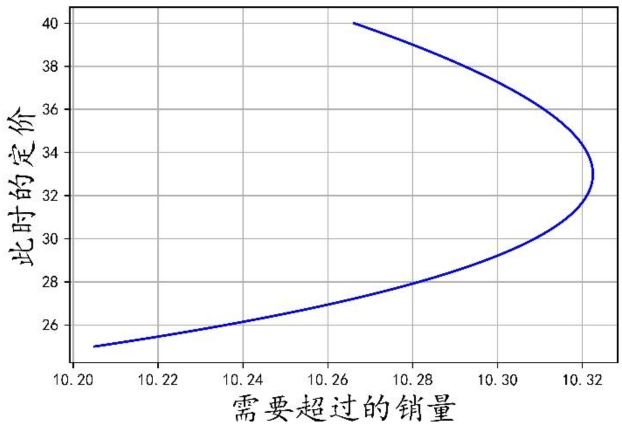

图 6 超过的销量和定价的关系图

观察图像发现，图像中存在不合理区间，结合实际情况分析，假设销售量极少时，盈利难度极大，反之，则盈利难度较小，在两种极端下模型的盈利与否是确定的，与图像展示内容不符，故为提高模型的准确性，需将销量控制在合理范围内考虑问题，根据（4）（13）构建动态模型并令 z_2 为整数且 $z_2 \in [5,14]$，绘制图像如下：

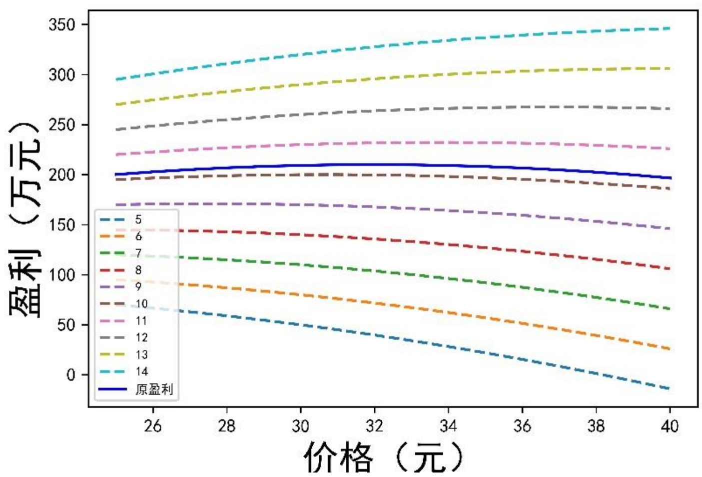

图 7 价格和盈利关系图

观察图像可知在销量取 10-11 万之间，计算得出销量的合理范围，绘制图像如下：

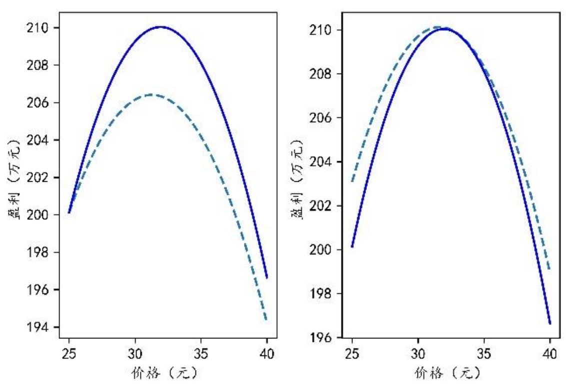

图 8 合理范围的两个临界图

5.2.3 模型求解及结果分析

通过反复迭代，求得此时的临界销量分别为 z_q=10.21 万件，z_q=10.32 万件，据此分析，销量低于 10.21 万件时不可能比之前更加盈利，当其超过 10.32 万件时任何合理定价都会更加盈利，由此将图 6 修正到合理范围，绘制图像如下：

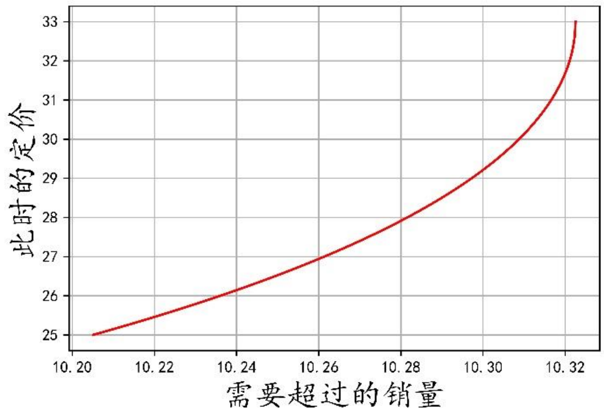

图 9 对图 6 的修正图

结合（17）和图 9，可以确定所需要超过的销量和此时的商品定价之间的关系。

结合图 8，进一步分析最大盈利的可能性。在图 8 的合理范围内，两图像存在唯一交点，在交点左侧满足比之前更加盈利，而在交点右侧不满足更加盈利，因此问题的可行域为交点左侧的部分；最大盈利的 x 值表现为函数极值点，根据二次函数的性质，函数的极值点在对称轴处取得。为求得最大盈利的可能性，需比较不同销售量下的对称轴和极值点的大小。

联立(4)(13)式，设交点横坐标为 x_i：

$$\begin{cases} c_1=(-0.2059x+13.15)x \\ c_2=x*z_2-y_1-y_2-50 \end{cases} \tag{18}$$

设 c_2 关于 x 的函数图像对称轴为 x_j，比较 x_i 和 x_j 得：

$$xi-xj \tag{19}$$

分析可得，(19)式是关于 z_2 的函数，设其零点为 z_0，当(19)式不小于零时，即 $z_2 \in [z_0, z_h]$可以取得最大盈利；当(19)式小于零时，即 $z_2 \in [z_q, z_0]$不能取得最大盈利。结合合理范围，设 η 为最大盈利可能性。

$$\eta = \left|\frac{zh-z0}{zh-zq}\right| *100\% \tag{20}$$

根据附录 2 可知：

$$\eta = 2.4\% \tag{21}$$

综上所述，

$$\eta = \begin{cases} 0, z2 < zq \\ 2.4\%, z2 < zq < zh \\ 100\%, z2 > zh \end{cases} \tag{22}$$

六、 灵敏度分析

为了进一步确定模型的稳定性，我们对问题一、二分别进行灵敏度分析。

6.1 问题一分析

a 的变化会对模型结果产生影响，因此对系数 a 进行灵敏度分析。

首先假设 x=25 不变，

$$S(c_1,a)=25*25*\frac{a}{25(a*25+b)} \tag{22}$$

由此可得，

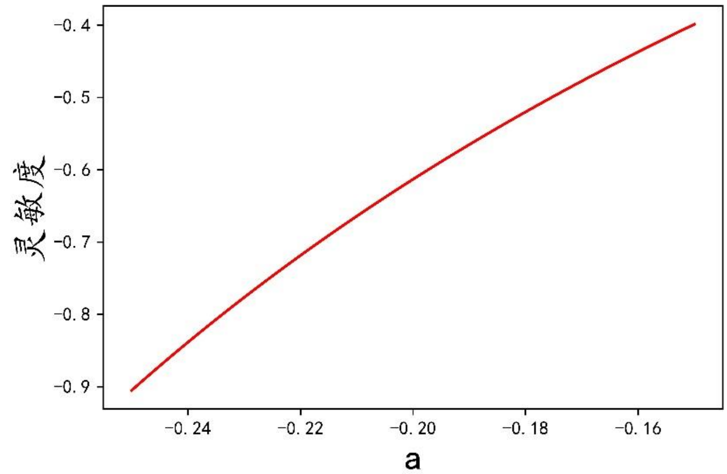

图 10 参数 a 灵敏度的分析图

其中，当 a=-0.2059 时，灵敏度为-0.61；同时观察图像可知，灵敏度普遍较小，模型较稳定。

6.2 问题二的分析

k_1 , k_2 的变化会对模型结果产生影响，因此对其进行灵敏度分析。

再次假设 x=25，y=10.25 不变，

$$S(c_2,k_1)=\frac{-625*k1}{25*10.25-(625k1-100)-5-50} \tag{23}$$

$$S(c_2,k_2)=\frac{-0.2k2}{25*10.25-0.2k2-50} \tag{24}$$

由此可得，

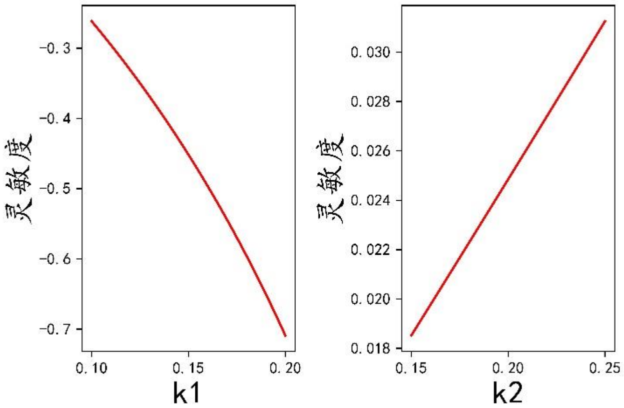

图 11 k_1，k_2 灵敏度分析图

其中当 k_1=0.16 时，灵敏度为-0.49，当 k_2=0.2 时，灵敏度为 0.024，观察图像可知，灵敏度普遍较小，模型较稳定。

通过一系列分析，发现模型关于系数的灵敏度极小，模型较稳定。

七、 模型的评价与改进

7.1 模型的优点

1. 图像的直观性：解题过程伴随图像展示，更为客观、直接的展现本组解题思路。
2. 建模的科学性：本文基于线性回归模型和动态模型，与已知数据拟合，得到了较稳定的模型和准确的结果。
3. 求解的严谨性：求解过程注重误差和灵敏度分析，并最大程度与实际相结合。

7.2 模型的缺点与改进

1. 问题一假设过于理想，对销量与售价关系的稳定性要求较高，在研究盈利问题时忽略了商品成本这一因素，并忽视较多现实因素产生的影响。
2. 模型仅适用于题目的局限背景下，应当考虑更多现实因素优化模型，使模型更具实际意义。

八、 附录

附录名称	问题一	代码语言	Python

```
#一次拟合曲线


import numpy as np
import matplotlib.pyplot as plt
from scipy import optimize
import pylab
```

```
def func(x, a, b):   # 需要拟合的函数
    return a*x+b
# 拟合点
plt.figure()
plt.rcParams['font.sans-serif'] = ['KaiTi'] # 指定字体 KaiTi（楷体）
plt.rcParams['axes.unicode_minus'] = False
x = [i for i in range(25, 41, 1)]#售价
y = [8.00,7.78,7.61,7.42,7.20,6.98,6.76,6.55,6.39,6.01,5.95,5.75,5.54,5.36,5.28,4.79]#销量
a1, b1 = optimize.curve_fit(func, x, y)[0]
x1 = np.arange(25, 41, 0.01)
y1 = a1*x1+b1
print(x1.shape,y1.shape)
print(a1,b1)
z=func(35,a1,b1)
print(z)
plt.scatter(x[:], y[:], 25, "red")
plt.grid()
plt.plot(x1, y1, "purple")
plt.legend(['原数据','一次拟合曲线'])
plt.xlabel('售价（元）')
plt.ylabel('销量（万件）')
plt.savefig('售价和销量的关系.png', dpi=1000)
plt.show()



#二次拟合曲线

import math

import numpy as np
import matplotlib.pyplot as plt
from scipy import optimize
import pylab
import scipy

def func(x):   # 需要拟合的函数
    return (-0.2059264703592322*x+13.153235286675047)*x
# 拟合点
a = -0.2059264703592322
b = 13.153235286675047
x0 = (-b-math.sqrt(b*b-4*a*(-200)))/(2*a)
```

```
plt.figure()
plt.rcParams['font.sans-serif'] = ['KaiTi'] # 指定字体 KaiTi（楷体）
plt.rcParams['axes.unicode_minus'] = False
plt.subplot(121)
x = np.linspace(5,62,1000)#售价
R0 = np.ones(1000)
R0 = R0*200
R = func(x)
plt.xlabel('售价（元）')
plt.ylabel('销售额（万元）')
plt.plot(x,R,'k-',x,R0,'b--')
plt.legend(['销售额与售价之间的关系','最初的销售额'],fontsize=8)
print(x0)
print((func(x0)))

plt.subplot(122)
x = np.linspace(20,45,1000)#售价
R0 = np.ones(1000)
R0 = R0*200
R = func(x)
plt.annotate('[1]',fontsize=10,xy=(39,201),xytext=(43,205),arrowprops=dict(facecolor='blu
e',edgecolor='black'))
plt.text(30,197,'(38.9,200)',fontsize=10)
plt.plot(x,R,'k-',x,R0,'b--',x0,200,'ro')

plt.xlabel('售价（元）')
plt.ylabel('销售额（万元）')
plt.legend(['销售额与售价之间的关系','最初的销售额'],fontsize=8)

plt.subplots_adjust(wspace =0.4)#调整子图间距

plt.savefig('销售额和售价的关系.jpg', dpi=1000)
plt.show()


#导函数曲线

import math
import sympy as sp
import numpy as np
import matplotlib.pyplot as plt
from scipy import optimize
```

```
import pylab
import scipy
import sympy as sp

x = sp.Symbol('x')
f = -0.2059264703592322*2*x+13.153235286675047
x = sp.solve(f)
x0=x
print(x)

def func(x):    # 需要拟合的函数
    return 2*(-0.2059264703592322)*x+13.153235286675047

plt.figure()
plt.rcParams['font.sans-serif'] = ['KaiTi'] # 指定字体 KaiTi（楷体）
plt.rcParams['axes.unicode_minus'] = False
x = np.linspace(5,62,1000)
y = func(x)
y0 = np.zeros(1000)
plt.plot(x,y,'r-',x,y0,'b--',x0,0,'ro')
plt.xlabel('x',fontsize=20)
plt.ylabel('dC1/dx',fontsize=20,loc='top')
plt.grid()
plt.annotate('导函数零点
',fontsize=20,xy=(34,1),xytext=(40,4),arrowprops=dict(facecolor='blue',edgecolor='black'))

plt.savefig('关于销售额的导函数图像.jpg',dpi=1000)
plt.show()


#图 10

import math
import numpy as np
import matplotlib.pyplot as plt
from scipy import optimize
import pylab
import scipy
import sympy
from scipy.optimize import fsolve
from mpl_toolkits.mplot3d import Axes3D
from sympy import symbols, diff
```

```
from tqdm import tqdm


a = np.linspace(-0.15,-0.25,1000)
k1 = np.linspace(0.1,0.2,1000)
k2 = np.linspace(0.15,0.25,1000)
fig = plt.figure()
plt.rcParams['font.sans-serif'] = ['KaiTi'] # 指定字体 KaiTi（楷体）
plt.rcParams['axes.unicode_minus'] = False
b = 13.153235286675047
def func1(a):
    return 25*25*a/(25*(25*a+b))
plt.xlabel('a',fontsize=20)
plt.ylabel('灵敏度',fontsize=20)
print(func1(-0.2))
plt.plot(a,func1(a),'r-')
plt.savefig('a 灵敏度.jpg',dpi=1000)
plt.show()
```

附录名称	问题二	代码语言	Python

```
#图 6
def func1(x):
    return (0.16+a) * x + (-100)/x + (0.2+b) + 50/x
x = np.linspace(25,40,1000)
z2 = func1(xl)
plt.xlabel('需要超过的销量',fontsize=20)
plt.ylabel('此时的定价',fontsize=20)
plt.plot(z2,x,'b-')
plt.grid()
plt.savefig('需要超过的销量与此时的定价关系.jpg',dpi=1000)
plt.show()

#图 7
import matplotlib.pyplot as plt
#设销售量与售价的关系为未知，建立 P 的表达式
def func(x,y):
    return x * y - 0.16 * x ** 2 - (-100) - 0.2 * x - 50
#对比原来的销售量与售价的关系对应的 P
def func0(x):
    return (-0.2059264703592322*x+13.153235286675047)*x
for yi in range(5,15):
```

```
    P = func(x,yi)
    plt.plot(x,P,'--',label=yi)
P0 = func0(x)
plt.plot(x,P0,'b-',label='原盈利')

plt.xlabel('价格（元）',fontsize=20)
plt.legend(loc='lower left',fontsize=7)
plt.ylabel('盈利（万元）',fontsize=20)
plt.savefig('遍历图.jpg',dpi=1000)
plt.show()

#图 8
fig = plt.figure()
plt.rcParams['font.sans-serif'] = ['KaiTi'] # 指定字体 KaiTi（楷体）
plt.rcParams['axes.unicode_minus'] = False
a = -0.2059264703592322
b = 13.153235286675047
x = np.linspace(25,40,1000)
y0 = a*x+b
#设销售量与售价的关系为未知，建立 P 的表达式
def func(x,y):
    return x * y - 0.16 * x ** 2 - (-100) - 0.2 * x - 50
#对比原来的销售量与售价的关系对应的 P
def func0(x):
    return (-0.2059264703592322*x+13.153235286675047)*x
for yi in range(5,15):
    P = func(x,yi)
    plt.plot(x,P,'--')
P0 = func0(x)
plt.plot(x,P0,'b-')
plt.xlabel('价格（元）')
plt.ylabel('盈利（万元）')
#由图可知原来的关系在 yi 对应 10--11,先粗略求出一直大于原函数保持盈利的近似临界位置
tr = np.linspace(10,11,1000)
sign = 1
for i in tr:
    P = func(x,i)
    k = 0
    for j in range(0,1000) :
        if P[j]>=P0[j] :
            k+=1
```

```
            if k==1000 :
                print("i1 = ",i )
                sign=0
        if sign==0 :
            break

plt.subplot(122)
ym = func(x,i)
plt.plot(x,ym,'--')
plt.plot(x,P0,'b-')
plt.xlabel('价格（元）')
plt.ylabel('盈利（万元）')
'''plt.legend([])'''

tr = np.linspace(10,11,1000)
n = 1
t = 0
temp = []
for i in tr :
    P = func(x,i)
    k = 0
    sign = 0
    for j in range(0,1000) :
        if P[j]>P0[j] :
            k+=1
        if k>=n :
            t = i
            temp.append(i)
            sign = 1
            break
    if sign==0 :
        temp.append(t)
    if temp==1 :
        n+=1

temp1 = np.array(temp)
tr = np.linspace(10.10,10.30,1000)
sign = 0
for i in tr :
    P = func(x,i)
    k=0
    for j in range(0,1000) :
```

```
            if P[j]>=P0[j] :
                k+=1
            if k==1 :
                print("i2 = ",i)
                sign = 1
                break
        if sign ==1 :
            break

y1 = func(x,i)
plt.subplot(121)
plt.plot(x,y1,'--')
plt.plot(x,P0,'b-')
plt.xlabel('价格（元）')
plt.ylabel('盈利（万元）')
plt.savefig('临界情况图.jpg',dpi=1000)
plt.subplots_adjust(wspace=0.3)
plt.show()


#图 9
r=[]
tr2=np.linspace(10.205105105105105,10.322512512512512,1000)
for q in tr2:

    f6=lambda x:x*(args[0]*x+args[1])-(x*q-0.16*x**2-(-100)-0.2*x-50)
    a=fsolve(f6,[1])
    if a>=25 and a<=40 :
        r.append(a)

plt.plot(tr2,r,'r-')
plt.xlabel('需要超过的销量',fontsize=20)
plt.ylabel('此时的定价',fontsize=20)
plt.grid()
plt.savefig('修正图.jpg',dpi=1000)

plt.show()
```

```
def func1(x):
    return (0.16+a) * x + (-100)/x + (0.2+b) + 50/x
x = np.linspace(25,40,1000)
z2 = func1(xl)
plt.xlabel('需要超过的销量',fontsize=20)
plt.ylabel('此时的定价',fontsize=20)
plt.plot(z2,x,'b-')
plt.grid()
plt.savefig('需要超过的销量与此时的定价关系.jpg',dpi=1000)
plt.show()




import matplotlib.pyplot as plt
#设销售量与售价的关系为未知，建立 P 的表达式
def func(x,y):
    return x * y - 0.16 * x ** 2 - (-100) - 0.2 * x - 50
#对比原来的销售量与售价的关系对应的 P
def func0(x):
    return (-0.2059264703592322*x+13.153235286675047)*x
for yi in range(5,15):
    P = func(x,yi)
    plt.plot(x,P,'--',label=yi)
P0 = func0(x)
plt.plot(x,P0,'b-',label='原盈利')

plt.xlabel('价格（元）',fontsize=20)
plt.legend(loc='lower left',fontsize=7)
plt.ylabel('盈利（万元）',fontsize=20)
plt.savefig('遍历图.jpg',dpi=1000)
plt.show()



fig = plt.figure()
plt.rcParams['font.sans-serif'] = ['KaiTi'] # 指定字体 KaiTi（楷体）
plt.rcParams['axes.unicode_minus'] = False
a = -0.2059264703592322
b = 13.153235286675047
x = np.linspace(25,40,1000)
```

```
y0 = a*x+b
#设销售量与售价的关系为未知，建立 P 的表达式
def func(x,y):
    return x * y - 0.16 * x ** 2 - (-100) - 0.2 * x - 50
#对比原来的销售量与售价的关系对应的 P
def func0(x):
    return (-0.2059264703592322*x+13.153235286675047)*x
for yi in range(5,15):
    P = func(x,yi)
    plt.plot(x,P,'--')
P0 = func0(x)
plt.plot(x,P0,'b-')
plt.xlabel('价格（元）')
plt.ylabel('盈利（万元）')
#由图可知原来的关系在 yi 对应 10--11,先粗略求出一直大于原函数保持盈利的近似临界位置
tr = np.linspace(10,11,1000)
sign = 1
for i in tr:
    P = func(x,i)
    k = 0
    for j in range(0,1000) :
        if P[j]>=P0[j] :
            k+=1
        if k==1000 :
            print("i1 = ",i )
            sign=0
    if sign==0 :
        break

plt.subplot(122)
ym = func(x,i)
plt.plot(x,ym,'--')
plt.plot(x,P0,'b-')
plt.xlabel('价格（元）')
plt.ylabel('盈利（万元）')
'''plt.legend([])'''

tr = np.linspace(10,11,1000)
n = 1
t = 0
temp = []
```

```
for i in tr :
        P = func(x,i)
        k = 0
        sign = 0
        for j in range(0,1000) :
                if P[j]>P0[j] :
                        k+=1
                if k>=n :
                        t = i
                        temp.append(i)
                        sign = 1
                        break
        if sign==0 :
                temp.append(t)
        if temp==1 :
                n+=1

temp1 = np.array(temp)
tr = np.linspace(10.10,10.30,1000)
sign = 0
for i in tr :
        P = func(x,i)
        k=0
        for j in range(0,1000) :
                if P[j]>=P0[j] :
                        k+=1
                if k==1 :
                        print("i2 = ",i)
                        sign = 1
                        break
        if sign ==1 :
                break

y1 = func(x,i)
plt.subplot(121)
plt.plot(x,y1,'--')
plt.plot(x,P0,'b-')
plt.xlabel('价格（元）')
plt.ylabel('盈利（万元）')
plt.savefig('临界情况图.jpg',dpi=1000)
plt.subplots_adjust(wspace=0.3)
plt.show()
```

附录名称	灵敏度分析	代码语言	Python

```
#可能性
#原始方程
def func(x,y):
    return x * y - 0.16 * x ** 2 - (-100) - 0.2 * x - 50
a = -0.2059264703592322
b = 13.153235286675047
#由先前方程得出的销售量的估计取值范围
y = np.linspace(10.205105105,10.3225125125,1000)
temp1 = []
temp2 = []
n = 0
n0 = 0
#计算在变化的利润的函数图像的对称轴以及此函数图像和原来的函数图像的交点来估算可能性的大小
for i in tqdm(y) :
    try:

        x = sympy.Symbol('x')
        n0 +=1
        temp1.append(sympy.solve(diff(func(x,i),x),[x])[0])
        temp2.append(min(sympy.solve((a+0.16)*x**2+(b-i+0.2)*x-50,[x])))
        if(temp2[n0-1]<temp1[n0-1]) :
            n+=1
            #通过 try 函数来排除因为误差导致的极少数虚数情况
    except TypeError :
        break
print(temp1)
print(temp2)
print(n/1000*100)
#问题一

import math
import numpy as np
import matplotlib.pyplot as plt
from scipy import optimize
import pylab
import scipy
import sympy
from scipy.optimize import fsolve
from mpl_toolkits.mplot3d import Axes3D
```

```
from sympy import symbols, diff
from tqdm import tqdm


a = np.linspace(-0.15,-0.25,1000)
k1 = np.linspace(0.1,0.2,1000)
k2 = np.linspace(0.15,0.25,1000)
fig = plt.figure()
plt.rcParams['font.sans-serif'] = ['KaiTi'] # 指定字体 KaiTi（楷体）
plt.rcParams['axes.unicode_minus'] = False
b = 13.153235286675047
def func1(a):
	return 25*25*a/(25*(25*a+b))
plt.xlabel('a')
plt.ylabel('灵敏度')
print(func1(-0.2))
plt.plot(a,func1(a),'r-')
plt.show()


#问题二

import math
import numpy as np
import matplotlib.pyplot as plt
from scipy import optimize
import pylab
import scipy
import sympy
from scipy.optimize import fsolve
from mpl_toolkits.mplot3d import Axes3D
from sympy import symbols, diff
from tqdm import tqdm


a = np.linspace(-0.15,-0.25,1000)
k1 = np.linspace(0.1,0.2,1000)
k2 = np.linspace(0.15,0.25,1000)
fig = plt.figure()
plt.rcParams['font.sans-serif'] = ['KaiTi'] # 指定字体 KaiTi（楷体）
plt.rcParams['axes.unicode_minus'] = False
b = 13.153235286675047
```

```
plt.subplot(121)
def func2(k1):
    return -625*k1/(25*10.25-(625*k1-100)-5-50)
plt.xlabel('k1')
plt.ylabel('灵敏度')
print(func2(0.16))
plt.plot(k1,func2(k1),'r-')
plt.subplot(122)
def func3(k3):
    return 25*k3/(25*10.25-25*k3-50)
plt.xlabel('k2')
plt.ylabel('灵敏度')
print(func3(0.2))
plt.plot(k2,func3(k2),'r-')
plt.subplots_adjust(wspace=0.5,hspace=0.5)

#a -61%
#k1 -49%
#k2 2.4%
plt.show()
```

后记：这是大学以来第一次接触大作业，也是第一次书写论文，体验很新奇，收获很丰富。