数模第二次上机

## 一 分析系数339灵敏度

1. 带入目标函数得到表达式,设灵敏度参数为t
2. 对x1,x2求偏导数并解方程组得到x1,x2关于参数t的表达式

解方程组得：

1. 做出x1,x2关于t的图像，如下：

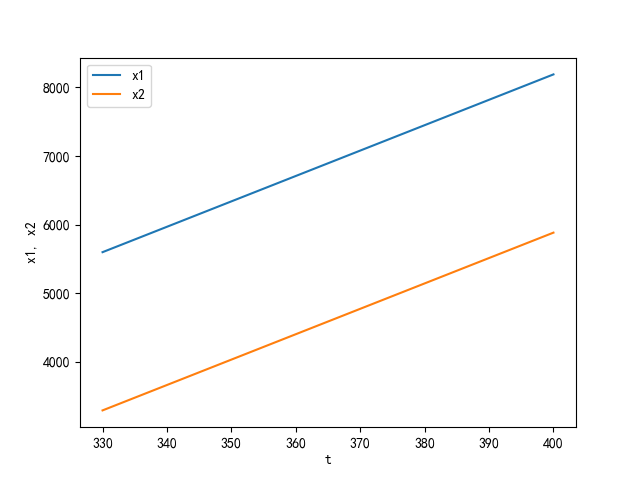


图 1x1,x2关于t的图像

可以看出系数339的提高会导致x1,x2的上升，是线性正比关系

1. 求解x1,x2关于t的灵敏度

公式为：

求得：

1. 将x1,x2带入利润表达式，做出利润y关于参数t的图像并求出灵敏度

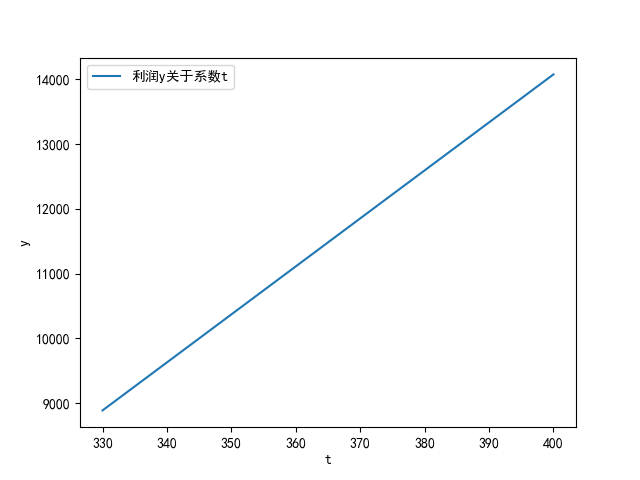


图 2 利润y关于系数t的图像

灵敏度：5.85 

可以看出该系数t与利润y成线性正比，影响较大

## 二 分析系数0.004灵敏度

1. 带入目标函数得到表达式,设灵敏度参数为t
2. 对x1,x2求偏导数并解方程组得到x1,x2关于参数t的表达式

解方程组得：

1. 做出x1,x2关于t的图像，如下：

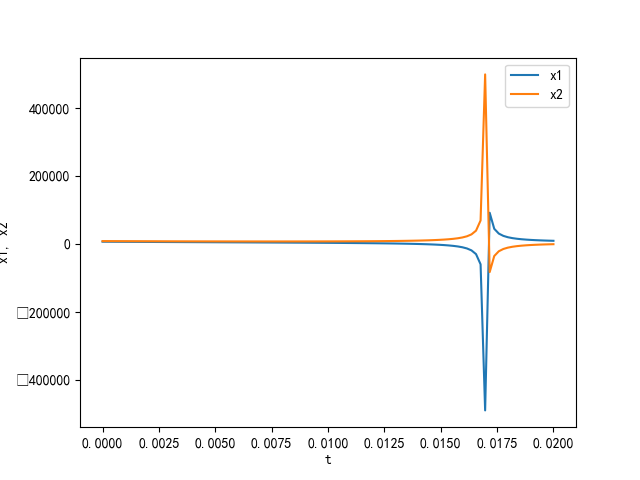


图 1x1,x2关于t的图像

可以看出系数0.004在0.017附近对x1,x2影响较大

1. 求解x1,x2关于t的灵敏度

公式为：

求得：

1. 将x1,x2带入利润表达式，做出利润y关于参数t的图像并求出灵敏度

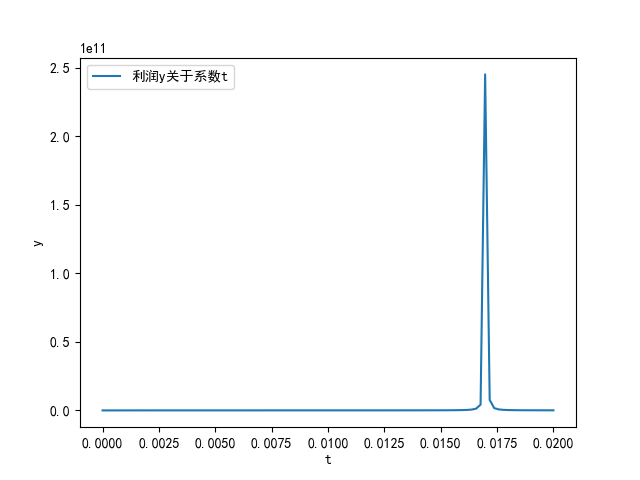


图 2 利润y关于系数t的图像

灵敏度：-0.24 

可以看出该系数0.004在0.17附近对y的影响较大

## 三 分析系数195灵敏度

1. 带入目标函数得到表达式,设灵敏度参数为t
2. 对x1,x2求偏导数并解方程组得到x1,x2关于参数t的表达式

解方程组得：

1. 做出x1,x2关于t的图像，如下：

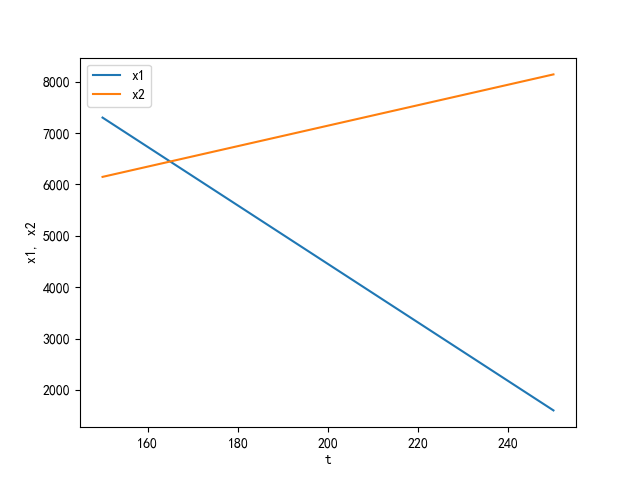


图 1x1,x2关于t的图像

可以看出系数195与x1是线性反比，与x2是线性反比

1. 求解x1,x2关于t的灵敏度

公式为：

求得：

1. 将x1,x2带入利润表达式，做出利润y关于参数t的图像并求出灵敏度

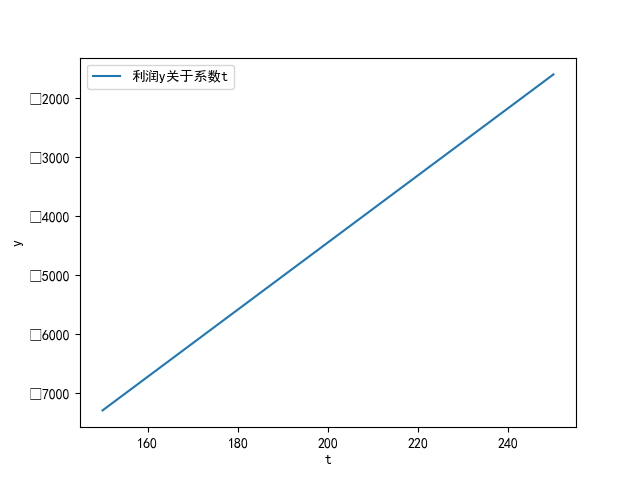


图 2 利润y关于系数t的图像

灵敏度：-1.66 

可以看出该系数t与利润y成线性正比，影响较大

## 四 计算代码

from sympy import symbols, diff, solve  
import numpy as np  
import matplotlib.pyplot as plt  
  
  
plt.rcParams['font.family'] = 'SimHei'  
  
def plot\_curves(solution, t\_range):  
 t\_values = np.linspace(t\_range[0], t\_range[1], 100)  
  
 # 初始化空列表来存储x1和x2的值  
 x1\_values = []  
 x2\_values = []  
  
 # 计算每个t值对应的x1和x2值  
 for t\_val in t\_values:  
 x1\_val = solution[x1].subs(t, t\_val)  
 x2\_val = solution[x2].subs(t, t\_val)  
 x1\_values.append(x1\_val)  
 x2\_values.append(x2\_val)  
  
 plt.plot(t\_values, x1\_values, label='x1')  
 plt.plot(t\_values, x2\_values, label='x2')  
 plt.xlabel('t')  
 plt.ylabel('x1, x2')  
 plt.legend()  
 plt.show()  
  
def plot\_curves\_2(expr, t\_range):  
 t\_values = np.linspace(t\_range[0], t\_range[1], 100)  
  
 # 初始化空列表来存储y的值  
 y\_values = []  
  
 # 计算每个t值对应的x1和x2值  
 for t\_val in t\_values:  
 y\_values.append(expr.subs(t , t\_val))  
  
 plt.plot(t\_values, y\_values, label='利润y关于系数t')  
 plt.xlabel('t')  
 plt.ylabel('y')  
 plt.legend()  
 plt.show()  
  
def cal\_s(expr , t\_value , best\_value):#求解灵敏度  
 prime = expr.subs(t, t\_value)  
 return prime\*(t\_value/best\_value)  
  
  
#定义三个需要分析灵敏度的系数  
a\_value = 339 #取值范围[330 , 400]  
b\_value = 0.004 #取值范围[0 , 0.02]  
c\_value = 195 #取值范围[150 , 250]  
  
  
# 定义变量  
x1, x2, t= symbols('x1 x2 t')#x1取值范围[0 , 50000] x2取值范围[-10000 , 7500]  
  
  
# 定义函数  
f1 = -0.01\*(x1\*\*2+x2\*\*2)-0.007\*x1\*x2+(t-195)\*x1+(t-225)\*x2-400000 #分析339  
f2 = -0.01\*(x1\*\*2+x2\*\*2)-(t+0.003)\*x1\*x2+144\*x1+174\*x2-400000 #分析0.004  
f3 = -0.01\*(x1\*\*2+x2\*\*2)-0.007\*x1\*x2+(339-t)\*x1+174\*x2-400000 #分析195  
  
# 求偏导数  
f1\_prime\_x1 = diff(f1, x1) # 对x1求偏导数  
f1\_prime\_x2 = diff(f1, x2) # 对x2求偏导数  
# print(f1\_prime\_x1)  
# print(f1\_prime\_x2)  
  
f2\_prime\_x1 = diff(f2, x1) # 对x1求偏导数  
f2\_prime\_x2 = diff(f2, x2) # 对x2求偏导数  
# print(f2\_prime\_x1)  
# print(f2\_prime\_x2)  
  
f3\_prime\_x1 = diff(f3, x1) # 对x1求偏导数  
f3\_prime\_x2 = diff(f3, x2) # 对x2求偏导数  
# print(f3\_prime\_x1)  
# print(f3\_prime\_x2)  
  
# 解方程组  
solution1 = solve((f1\_prime\_x1, f1\_prime\_x2), (x1, x2))  
# print(solution1[x1])  
# print(solution1[x2])  
#对参数t求导  
x1\_prime\_t = solution1[x1].subs(t, a\_value)  
x2\_prime\_t = solution1[x2].subs(t, a\_value)  
#求出灵敏度值  
# print(cal\_s(solution1[x1], a\_value, 4735))  
# print(cal\_s(solution1[x2], a\_value, 7043))  
  
solution2 = solve((f2\_prime\_x1, f2\_prime\_x2), (x1, x2))  
# print(solution2[x1])  
# print(solution2[x2])  
#求出灵敏度值  
# print(cal\_s(solution2[x1], b\_value, 4735))  
# print(cal\_s(solution2[x2], b\_value, 7043))  
  
solution3 = solve((f3\_prime\_x1, f3\_prime\_x2), (x1, x2))  
# print(solution3[x1])  
# print(solution3[x2])  
#求出灵敏度值  
# print(cal\_s(solution3[x1], c\_value, 4735))  
# print(cal\_s(solution3[x2], c\_value, 7043))  
  
#绘制x1,x2关于t的曲线  
# plot\_curves(solution1, [330, 400])  
# plot\_curves(solution2, [0, 0.02])  
# plot\_curves(solution3, [150, 250])  
  
# 将x1, x2的表达式代入f中  
f1\_expr\_a = f1.subs({x1: solution1[x1], x2: solution1[x2]})  
f2\_expr\_a = f2.subs({x1: solution2[x1], x2: solution2[x2]})  
f3\_expr\_a = f3.subs({x1: solution3[x1], x2: solution3[x2]})  
  
#对参数t求导数  
f1\_prime\_a = diff(f1\_expr\_a, t)  
f2\_prime\_a = diff(f2\_expr\_a, t)  
f3\_prime\_a = diff(f3\_expr\_a, t)  
  
# plot\_curves\_2(f1\_prime\_a , [330, 400])  
# plot\_curves\_2(f2\_prime\_a , [0, 0.02])  
# plot\_curves\_2(f3\_prime\_a , [150, 250])  
  
# 打印结果  
print(cal\_s(f1\_prime\_a, a\_value, 553641))  
print(cal\_s(f2\_prime\_a, b\_value, 553641))  
print(cal\_s(f3\_prime\_a, c\_value, 553641))