全国大学生数学建模竞赛获奖代码-《生产过程中决策问题》

作为组长，我在为期 3 天的全国大学生数学建模竞赛中，负责全程建模思路的搭建、编程实现与部分论文撰写，期间协调全流程的节奏，我们从给定的 A、B、C 三道题目中选择了考察建模逻辑和数据分析能力的 B 题。整个流程可整理为“选题与问题拆解-建立模型-编程实现-论文撰写”。经过专家评审后我们的作品最终获得了三等奖，模型具有实际应用价值且论文逻辑清晰、结果验证充分。以下为代码分享：

附录 1

1.问题一验证假设误差源代码

```
import scipy.stats as ss
import math
##第 1 问检验：如果算出实际次品率与标称值之间的误差在 5%左右，证明我们假设的误差合理，模型便合理

p=0.10 #标称值
a1=0.05 #显著性水平（1）
a2=0.10 #显著性水平（2）
##使用单侧检验
z1=ss.norm.ppf(1-a1)
z2=ss.norm.ppf(1-a2)
wucha=0.05 #尽可能大的误差，这样求出的抽样次数会尽可能少
##计算抽样结果
N1=math.ceil((z1**2*p*(1-p))/wucha**2) #（1）
N2=math.ceil((z2**2*p*(1-p))/wucha**2) #（2）
'''
因为 H0：p>p0(拒绝)；H：p<p0(接受)；所以，
将 N1 代入统计量计算式 z1=(p1-p)/math.sqrt(p(1-p)/N1)求实际次品率 p1；
将 N2 代入统计量计算式 z2=(p-p2)/math.sqrt(p(1-p)/N2)求实际次品率 p2'''
p1=z1*math.sqrt(p*(1-p)/N1)+p
p2=p-z2*math.sqrt(p*(1-p)/N2)
print(p1)
print(p2)
```

附录 2

2.问题一求解检验次数源代码

```
import scipy.stats as ss
import math
p=0.10 #标称值
a1=0.05 #显著性水平（1）
a2=0.10 #显著性水平（2）
##使用单侧检验
z1=ss.norm.ppf(1-a1)
z2=ss.norm.ppf(1-a2)
wucha=0.05 #尽可能大的误差，这样求出的抽样次数会尽可能少
##计算抽样结果
N1=math.ceil((z1**2*p*(1-p))/wucha**2) #（1）
N2=math.ceil((z2**2*p*(1-p))/wucha**2) #（2）
print(N1)
print(N2)
```

附录 3

3.问题二求解情况 6 下各决策方案利润源代码

```
import random
import scipy.stats as ss
import math
#配件 1 次品率
p1=0.05
#配件 2 次品率
p2=0.05
#检验零件的成品次品率(利用指标函数)
a=0.05
z=ss.norm.ppf(1-a)
n=52 #置信度 95%下的抽样次数
p3=1-0.056-1.89*(1-(-z*math.sqrt(p1*(1-p1)/n)+p1))*(1-(-z*math.sqrt(p2*(1-p2)/n)+p2
))/(2+z*math.sqrt(p1*(1-p1)/n)+z*math.sqrt(p2*(1-p2)/n)-p1-p2)
#未检验零件的成品次品率
```

```
p4=0.05
#随机生成的配件 1 数目
n1=random.randint(0, 1000)
print(n1)
#随机生成的配件 2 数目
n2=random.randint(0, 1000)
print(n2)
#未检测零件的成品数目
n4=min(n1,n2)
#检测零件的成品数目
n3=n4-(n1*p1+n2*p2)/2
#配件 1 检测成本
c1=2
#配件 2 检测成本
c2=3
#成品检测成本
ct=3
#成品装配成本
cp=6
#配件 1 进价
b1=4
#配件 2 进价
b2=18
#成品售价
s=56
#调换损失
e=10
#拆解费用
d=40
##八种情况的利润求解
#1.检验零件；检验成品；拆解
y1=(1-p3)*s*n3-(n1*(c1+b1)+n2*(c2+b2)+n3*(cp+ct)+n3*p3*d)
#2.检验零件；检验成品；不拆解
y2=(1-p3)*s*n3-(n1*(c1+b1)+n2*(c2+b2)+n3*(cp+ct)+n3*p3*(b1+b2+cp))
#3.不检验零件；检验成品；拆解
```

```
    y3=(1-p4)*s*n4-(n1*b1+n2*b2+n4*(cp+ct)+n4*p4*d+n4*p4*(c1+c2))
    #4.不检验零件；检验成品；不拆解
    y4=(1-p4)*s*n4-(n1*b1+n2*b2+n4*(cp+ct)+n4*p4*(b1+b2+cp))
    #5.不检验零件；不检验成品；拆解
    y5=(1-p4)*s*n4-(n1*b1+n2*b2+n4*cp+n4*p4*(e+d)+n4*p4*(c1+c2))
    #6.不检验零件；不检验成品；不拆解
    y6=(1-p4)*s*n4-(n1*b1+n2*b2+n4*cp+n4*p4*(e+b1+b2+cp))
    #7.检验零件；不检验成品；拆解
    y7=(1-p3)*s*n3-(n1*(c1+b1)+n2*(c2+b2)+n3*cp+n3*p3*(e+d))
    #8.检验零件；不检验成品；不拆解
    y8=(1-p3)*s*n3-(n1*(c1+b1)+n2*(c2+b2)+n3*cp+n3*p3*(e+b1+b2+cp))
    print(y1)
    print(y2)
    print(y3)
    print(y4)
    print(y5)
    print(y6)
    print(y7)
print(y8)
```

附录 4

4.问题二以情况 2 为例验证模型稳定性源代码

```
    import random
    import scipy.stats as ss
    import math
    #具有扰动的配件 1 次品率
    p1=0.2+0.0498
    #具有扰动的配件 2 次品率
    p2=0.2+0.0498
    #检验零件的成品次品率(利用指标函数)
    a=0.05
    z=ss.norm.ppf(1-a)
    n=174 #置信度 95%下标称值为 20%的抽样次数
    p3=1-0.056-1.89*(1-(-z*math.sqrt(p1*(1-p1)/n)+p1))*(1-(-z*math.sqrt(p2*(1-p2)/n)+p2
))/(2+z*math.sqrt(p1*(1-p1)/n)+z*math.sqrt(p2*(1-p2)/n)-p1-p2)
```

```
#未检验零件的成品次品率
p4=0.2
#随机生成的配件 1 数目
n1=random.randint(0, 1000)
print(n1)
#随机生成的配件 2 数目
n2=random.randint(0, 1000)
print(n2)
#未检测零件的成品数目
n4=min(n1,n2)
#检测零件的成品数目
n3=n4-(n1*p1+n2*p2)/2
#配件 1 检测成本
c1=2
#配件 2 检测成本
c2=3
#成品检测成本
ct=3
#成品装配成本
cp=6
#配件 1 进价
b1=4
#配件 2 进价
b2=18
#成品售价
s=56
#调换损失
e=6
#拆解费用
d=5
##八种情况的利润求解
#1.检验零件；检验成品；拆解
y1=(1-p3)*s*n3-(n1*(c1+b1)+n2*(c2+b2)+n3*(cp+ct)+n3*p3*d)
#2.检验零件；检验成品；不拆解
y2=(1-p3)*s*n3-(n1*(c1+b1)+n2*(c2+b2)+n3*(cp+ct)+n3*p3*(b1+b2+cp))
```

```
    #3.不检验零件；检验成品；拆解
    y3=(1-p4)*s*n4-(n1*b1+n2*b2+n4*(cp+ct)+n4*p4*d+n4*p4*(c1+c2))
    #4.不检验零件；检验成品；不拆解
    y4=(1-p4)*s*n4-(n1*b1+n2*b2+n4*(cp+ct)+n4*p4*(b1+b2+cp))
    #5.不检验零件；不检验成品；拆解
    y5=(1-p4)*s*n4-(n1*b1+n2*b2+n4*cp+n4*p4*(e+d)+n4*p4*(c1+c2))
    #6.不检验零件；不检验成品；不拆解
    y6=(1-p4)*s*n4-(n1*b1+n2*b2+n4*cp+n4*p4*(e+b1+b2+cp))
    #7.检验零件；不检验成品；拆解
    y7=(1-p3)*s*n3-(n1*(c1+b1)+n2*(c2+b2)+n3*cp+n3*p3*(e+d))
    #8.检验零件；不检验成品；不拆解
    y8=(1-p3)*s*n3-(n1*(c1+b1)+n2*(c2+b2)+n3*cp+n3*p3*(e+b1+b2+cp))
    print(y1)
    print(y2)
    print(y3)
    print(y4)
    print(y5)
    print(y6)
print(y7)
print(y8)
```

附录 5

5.问题三求解决策方案总成本源代码

```
    import random
    #随机生成各配件的数目
    n=[]
    for i in range(8):
        ni=random.randint(0, 1000)
        n.append(ni)
    print(n)
    #零配件购买单价
    b=[2,8,12,2,8,12,8,12]
    #零配件的检测费用
    cp=[1,1,2,1,1,2,1,2]
    #半成品的检测费用
    cB=[4,4,4]
```

```
#成品的检测费用
ct=6
#零配件的次品率
p1=0.1
#半成品的次品率
p2=0.1
#成品的次品率
p3=0.1
#半成品的数量
B=[min(n[0]*p1,n[1]*p1,n[2]*p1),min(n[3]*p1,n[4]*p1,n[5]*p1),min(n[6]*p1,n[7]*p1)]
#成品数
m=min(B[0]*p2,B[1]*p2,B[2]*p2)
#成品的拆解费用
d1=10
#半成品的拆解费用
d2=[6,6,6]
#总成本
y=sum(n[i]*(cp[i]+b[i]) for i in range(8))+sum(B[j]*cB[j] for j in range(3))+m*ct+p3*m*d1+sum(B[j]*d2[j] for j in range(3))
print(y)
```

附录 6

6.基于抽样检测求解问题二情况 6 下各决策方案利润源代码

```
import random
import scipy.stats as ss
import math
#配件 1 的实际次品率
p1=0.00028
#配件 2 的实际次品率
p2=0.00028
#检验零件的成品次品率(利用指标函数)
a=0.05
z=ss.norm.ppf(1-a)
n=52 #置信度 95%下的抽样次数
```

```
    p3=1-0.056-1.89*(1-(-z*math.sqrt(p1*(1-p1)/n)+p1))*(1-(-z*math.sqrt(p2*(1-p2)/n)+p2
))/(2+z*math.sqrt(p1*(1-p1)/n)+z*math.sqrt(p2*(1-p2)/n)-p1-p2)
    #未检验零件的实际成品次品率
    p4=0.00028
    #随机生成的配件 1 数目
    n1=random.randint(0, 1000)
    print(n1)
    #随机生成的配件 2 数目
    n2=random.randint(0, 1000)
    print(n2)
    #未检测零件的成品数目
    n4=min(n1,n2)
    #检测零件的成品数目
    n3=n4-(n1*p1+n2*p2)/2
    #配件 1 检测成本
    c1=2
    #配件 2 检测成本
    c2=3
    #成品检测成本
    ct=3
    #成品装配成本
    cp=6
    #配件 1 进价
    b1=4
    #配件 2 进价
    b2=18
    #成品售价
    s=56
    #调换损失
    e=10
    #拆解费用
    d=40
    ##八种情况的利润求解
    #1.检验零件；检验成品；拆解
    y1=(1-p3)*s*n3-(n1*(c1+b1)+n2*(c2+b2)+n3*(cp+ct)+n3*p3*d)
```

```
#2.检验零件；检验成品；不拆解
y2=(1-p3)*s*n3-(n1*(c1+b1)+n2*(c2+b2)+n3*(cp+ct)+n3*p3*(b1+b2+cp))
#3.不检验零件；检验成品；拆解
y3=(1-p4)*s*n4-(n1*b1+n2*b2+n4*(cp+ct)+n4*p4*d+n4*p4*(c1+c2))
#4.不检验零件；检验成品；不拆解
y4=(1-p4)*s*n4-(n1*b1+n2*b2+n4*(cp+ct)+n4*p4*(b1+b2+cp))
#5.不检验零件；不检验成品；拆解
y5=(1-p4)*s*n4-(n1*b1+n2*b2+n4*cp+n4*p4*(e+d)+n4*p4*(c1+c2))
#6.不检验零件；不检验成品；不拆解
y6=(1-p4)*s*n4-(n1*b1+n2*b2+n4*cp+n4*p4*(e+b1+b2+cp))
#7.检验零件；不检验成品；拆解
y7=(1-p3)*s*n3-(n1*(c1+b1)+n2*(c2+b2)+n3*cp+n3*p3*(e+d))
#8.检验零件；不检验成品；不拆解
y8=(1-p3)*s*n3-(n1*(c1+b1)+n2*(c2+b2)+n3*cp+n3*p3*(e+b1+b2+cp))
print(y1)
print(y2)
print(y3)
print(y4)
print(y5)
print(y6)
print(y7)
print(y8)
```

附录 7

7.基于抽样检测求解问题三决策方案总成本源代码

```
#问题三中生成各配件的数目
n=[385, 725, 960, 575, 160, 300, 923, 761]
#零配件购买单价
b=[2,8,12,2,8,12,8,12]
#零配件的检测费用
cp=[1,1,2,1,1,2,1,2]
#半成品的检测费用
cB=[4,4,4]
#成品的检测费用
```

```
    ct=6
    #零配件的次品率
    p1=0.05015
    #半成品的次品率
    p2=0.05015
    #成品的次品率
    p3=0.05015
    #半成品的数量
    B=[min(n[0]*p1,n[1]*p1,n[2]*p1),min(n[3]*p1,n[4]*p1,n[5]*p1),min(n[6]*p1,n[7]*p1)]
    #成品数
    m=min(B[0]*p2,B[1]*p2,B[2]*p2)
    #成品的拆解费用
    d1=10
    #半成品的拆解费用
    d2=[6,6,6]
    #总成本
    y=sum(n[i]*(cp[i]+b[i]) for i in range(8))+sum(B[j]*cB[j] for j in range(3))+m*ct+p3*m*d1+sum(B[j]*d2[j] for j in range(3))
    print(y)
```