DST implementation Full C183085 – Mahir Shadid

```
# mahir_shadid_C183085
print("Enter two weights: ")
w1 = float(input())
w2 = float(input())
Sum = w1 + w2;
if Sum <= 1:
   b11=float(input("Enter belief of Poor accuracy: "))
   b21=float(input("Enter belief of Average accuracy: "))
   b31=float(input("Enter belief of Good accuracy: "))

   m11=w1*b11
   m21=w1*b21
   m31=w1*b31

   print("\nProbability Mass m(1,1),m(2,1),m(3,1) is:")
   print("%.2f" % m11)
   print("%.3f" % m21)
   print(m31)

   b12=float(input("\nEnter belief of Poor reputation: "))
   b22=float(input("Enter belief of Average reputation: "))
   b32=float(input("Enter belief of Good reputation: "))

   m12=w2*b12
   m22=w2*b22
   m32=w2*b32

   print("\nProbability Mass m(1,2),m(2,2),m(3,2) is:")
   print(m12)
   print("%.4f" % m22)
   print(m32)

   bmh1=1-w1
   bmh2=1-w2
   sb1=b11+b21+b31
```

```
    msb1=1-sb1
    fm11=w1*msb1
    mh1=bmh1+fm11


    sb2=b12+b22+b32
    msb2=1-sb2
    fm12=w2*msb2
    mh2=bmh2+fm12

    print("\nProbability Mass
bar_m(H,1),bar_m(H,2),tilda_m(H,1),tilda_m(H,2),m(h,1),m(h,2) is:")
    print(bmh1)
    print(bmh2)
    print("%.3f" % fm11)
    print(fm12)
    print(mh1)
    print(mh2)

    a=m11*m22
    b=m11*m32
    c=m21*m12
    d=m21*m32
    e=m31*m12
    f=m31*m22

    sum2=a+b+c+d+e+f
    subtr=1-sum2
    div=1/subtr

    print("\nKi2 constant is ", end=" ")
    print("%.5f"%div)

    g=m11*m12
    h=mh1*m12
    i=m11*mh2
```

```
    sum3=g+h+i
    m1i2=div*sum3

    j=m21*m22
    k=mh1*m22
    l=m21*mh2

    sum4=j+k+l
    m2i2=div*sum4

    m=m31*m32
    n=mh1*m32
    o=m31*mh2

    sum5=m+n+o
    m3i2=div*sum5

    print("\nProbability mass aggregations
m(1,i2),m(2,i2),m(3,i2),tilda_m(H,i2),bar_m(H,i2),m(H,i2): \n")
    print("%.4f"%m1i2)
    print("%.4f"%m2i2)
    print("%.4f"%m3i2)

    p=fm11*fm12
    q=bmh1*fm12
    r=fm11*bmh2

    sum6=p+q+r
    fmhi2=div*sum6
    print("%.4f"%fmhi2)

    bmhi2=bmh1*bmh2*div
    print("%.4f"%bmhi2)

    mhi2=bmhi2+fmhi2
```

```
    print("%.4f"%mhi2)

    print("\nCombined degrees of belief for intrinsic data quality")
    print("beta(1,1),beta(2,1),beta(3,1),beta(H):")

    s=1-bmhi2

    ib11=m1i2/s
    ib21=m2i2/s
    ib31=m3i2/s
    ibh=fmhi2/s

    print("%.4f"%ib11)
    print("%.4f"%ib21)
    print("%.4f"%ib31)
    print("%.4f"%ibh)

    print("\nThe evaluation grades utilities are:\n")

    h1u=float(input("Enter evaluation grade of Poor utility: "))
    h2u=float(input("Enter evaluation grade of Average utility: "))
    h3u=float(input("Enter evaluation grade of Good utility: "))

    print("\nThe beliefs for each assessment grade for the general property data
quality:\n")

    b1u=float(input("Beta1: "))
    b2u=float(input("Beta2: "))
    b3u=float(input("Beta3: "))

    u1=h1u*b1u
    u2=h2u*b2u
    u3=h3u*b3u

    u=u1+u2+u3
```

```
    print("\nThe total expected utility of this (complete) assessment is thus:
",end=" ")
    print(u)

    bhu=0.0336

    u11=(b3u+bhu)*h3u
    u22=(b1u+bhu)*h1u

    umax=u1+u2+u11
    umin=u22+u2+u3

    uavg=(umax+umin)/2

    print("\nUtility intervals for the assessment of data quality: ")
    print("umax: ",end=" ")
    print(umax)
    print("umin: ",end=" ")
    print(umin)
    print("uavg: ",end=" ")
    print(uavg)


else:
    print("Weight limit exceeded!")
```

Output:

```
Enter two weights:
0.35
0.65
Enter belief of Poor accuracy: 0.4
Enter belief of Average accuracy: 0.5
Enter belief of Good accuracy: 0

Probability Mass m(1,1),m(2,1),m(3,1) is:
0.14
0.175
0.0
```

```
Enter belief of Poor reputation: 0.1
Enter belief of Average reputation: 0.75
Enter belief of Good reputation: 0.15

Probability Mass m(1,2),m(2,2),m(3,2) is:
0.065
0.4875
0.0975

Probability Mass bar_m(H,1),bar_m(H,2),tilda_m(H,1),tilda_m(H,2),m(h,1),m(
h,2) is:
0.65
0.35
0.035
0.0
0.685
0.35

Ki2 constant is  1.12402

Probability mass aggregations m(1,i2),m(2,i2),m(3,i2),tilda_m(H,i2),bar_m(
H,i2),m(H,i2):

0.1154
0.5401
0.0751
0.0138
0.2557
0.2695

Combined degrees of belief for intrinsic data quality
beta(1,1),beta(2,1),beta(3,1),beta(H):
0.1550
0.7257
0.1009
0.0185

The evaluation grades utilities are:

Enter evaluation grade of Poor utility: 0
Enter evaluation grade of Average utility: 0.5
Enter evaluation grade of Good utility: 1

The beliefs for each assessment grade for the general property data qualit
y:

Beta1: 0.2695
Beta2: 0.6097
Beta3: 0.0872

The total expected utility of this (complete) assessment is thus:  0.39205

Utility intervals for the assessment of data quality:
```

```
umax:  0.42565
umin:  0.39205
uavg:  0.40885
```