emmaredfoot / AHP

Branch: master ▾ AHP / AHP_Buckley.py Find file Copy path


emmaredfoot Small add 4b0ad67 on Mar 16

1 contributor

202 lines (160 sloc) | 8.35 KB

```
1  import numpy
2  import csv
3  import matplotlib.pyplot as plt
4
5  #Functions
6  def GeometricMean(LineList, Spot):
7      values = [x[Spot] for x in LineList]
8      GeoMean=1
9      for i in range(len(LineList)):
10         GeoMean=GeoMean*values[i]
11     return(GeoMean**(1/len(LineList)))
12
13 def PerformanceScores(a, b, c, d):
14     # The number of rows in each matrix will always be three, so I am hard coding it in
15     PS=[0]*3
16     for x in range(3):
17         PS[x]=[a[x]/sum(d), b[x]/sum(c), c[x]/sum(b), d[x]/sum(a)]
18     return(PS)
19
20 #Weight all of the performance score values
21 #Add all of the values in the individual locations together
22 #Return the utility function for safety, ability to fluctuate, and profitability
23 def WeightPS(r_ij, weight):
24     weightedPS=[0]*len(r_ij)
25     for i in range(len(r_ij)):
26         weightedPS[i]=[a*b for a,b in zip(r_ij[i],weight[i])]
27         print(i,r_ij[i])
28         #L_1[i]=r_ij[i]
29     Desal=weightedPS[0]
30     HProd=weightedPS[1]
31     SynFuel=weightedPS[2]
32     #Find the right and left sides of each of the lines
33     # for i in range(len(Desal)):
34     #     L_1=(Desal[1]-weight[0])/(weight[1]-weight[0])
35     #     R_1=(Desal[3]-Desal[2])/(Weight[3]-Weight[2])
36     #     L_2=Weight[0]*(Desal[1]-Desal[0])+Desal[0]*(Weight[1]-Weight[1])
37     #     R_2=-1*(Weight[3]*(Desal[3]-Desal[2])+Desal[0]*(Weight[3]-Weight[2]))
38     #     return(L_1, L_2, R_1, R_2)
39     return(Desal, HProd, SynFuel)
40
41 def Utility(A, B, C):
42     utility=[0]*len(C)
43     for x in range(len(C)):
44         utility[x] = A[x]+B[x]+C[x]
45     return(utility)
46
47
48
49
50 #CONSTANTS AND INPUT
51 #1 corresponds to equally important
52 #2-3 correspond to weakly more important
53 #4-5 corresponds to strongly more important
54 #6-7 correspond to very strongly more important
55 #8-9 correspond to absolutely more important
```