

```

1 function [Result] = SOSDConstBoundsv3(DIST1,DIST2)
2 % Finds the min number where DIST1 (the comparison distribution) SOSD DIST2 (the base distribution) and
3 % the min number where DIST2 SOSD DIST1 in addition to a variety of
4 % descriptive statistics for DIST1 and DIST2
5
6 if size(DIST1,2) > 1
7     error('DIST1 must be a column vector!')
8 end
9
10 if size(DIST2,2) > 1
11     error('DIST2 must be a column vector!')
12 end
13
14 % Output Result row definitions
15 MEANDIST1      = 1;
16 SDDIST1        = 2;
17 CVDIST1        = 3;
18 MAXDIST1       = 4;
19 MINDIST1       = 5;
20 PRCROFFAILDIST1 = 6;
21 DIST1MINPROP   = 7;
22 MEANDIST2      = 8;
23 SDDIST2        = 9;
24 CVDIST2        = 10;
25 MAXDIST2       = 11;
26 MINDIST2       = 12;
27 PRCROFFAILDIST2 = 13;
28 DIST2MINPROP   = 14;
29 MEANDIFF       = 15;
30 SDDIFF         = 16;
31 CVDIFF         = 17;
32 CFDIFF        = 18;
33 RELCOMPSOSDBASE = 19;
34 RELBASESOSDCOMP = 20;
35 DELTARISK      = 21;
36
37 Result          = -999999 * ones(21,1); % Initialize Results
38
39 % Calculate the mean, standard deviation, coefficient of variation, minimum
40 % and maximum for DIST1
41 Result(MEANDIST1,1) = mean(DIST1);
42 Result(SDDIST1,1)   = std(DIST1);
43 if Result(MEANDIST1,1) > 0
44     Result(CVDIST1,1) = Result(SDDIST1,1) / Result(MEANDIST1,1);
45 end
46 Result(MAXDIST1,1) = max(DIST1);
47 Result(MINDIST1,1) = min(DIST1);
48
49 ldist1          = length(DIST1);
50
51 % Calculate the mean, standard deviation, coefficient of variation, minimum
52 % and maximum for DIST2
53 Result(MEANDIST2,1) = mean(DIST2);
54 Result(SDDIST2,1)   = std(DIST2);
55 if Result(MEANDIST2,1) > 0
56     Result(CVDIST2,1) = Result(SDDIST2,1) / Result(MEANDIST2,1);
57 end
58 Result(MAXDIST2,1) = max(DIST2);
59 Result(MINDIST2,1) = min(DIST2);
60
61 ldist2          = length(DIST2);
62
63 % Calculate differences in descriptive statistics between DIST2 and DIST1
64 if Result(MEANDIST1,1) == 0 && Result(MEANDIST2,1) == 0
65     Result(MEANDIFF,1) = -999999;
66 else
67     Result(MEANDIFF,1) = Result(MEANDIST1,1) - Result(MEANDIST2,1);
68 end
69
70 if Result(SDDIST1,1) == 0 && Result(SDDIST2,1) == 0
71     Result(SDDIFF,1) = -999999;
72 else
73     Result(SDDIFF,1) = Result(SDDIST1,1) - Result(SDDIST2,1);
74 end
75
76 if Result(CVDIST1,1) >= 0 && Result(CVDIST2,1) >= 0
77     Result(CVDIFF,1) = Result(CVDIST1,1) - Result(CVDIST2,1);
78 else
79     Result(CVDIFF,1) = -999999;
80 end
81
82 % Initialize Thresholds for Stopping Golden Section Search
83 thresh      = 0.00001;
84 bailthresh  = 1000000;
85

```

```

87 if Result(MEANDIST1,1) > 0 || Result(MEANDIST2,1) > 0
88     % Calculate Probability of crop failiure for DIST1
89     CropFailDIST1 = DIST1;
90     CropFailDIST1cond = DIST1(:,1) ~= 0;
91     CropFailDIST1(CropFailDIST1cond,:) = [];
92     Result(PRCROPFAILDIST1,1) = length(CropFailDIST1) / ldist1;
93
94     % Calculate Probability of crop failiure for DIST1
95     CropFailDIST2 = DIST2;
96     CropFailDIST2cond = DIST2(:,1) ~= 0;
97     CropFailDIST2(CropFailDIST2cond,:) = [];
98     Result(PRCROPFAILDIST2,1) = length(CropFailDIST2) / ldist2;
99
100 tcomp = Result(MAXDIST2,1) - Result(MINDIST1,1); % Maximum amount that DIST1 can be shifted to ensure it is SOSD
101 tbase = Result(MINDIST2,1) - Result(MAXDIST1,1); % Maximum amount that DIST1 can be shifted back to ensure
DIST2 is SOSD
102
103 % Initialize Golden Section Search upper and lower starting points
104 upper = 0;
105 lower = 0;
106 if tcomp > tbase
107     upper = tcomp;
108     lower = tbase;
109 elseif tcomp < tbase
110     upper = tbase;
111     lower = tcomp;
112 else
113     upper = tbase + 10;
114     lower = tbase - 10;
115 end
116
117 % Initialize Flags to Test for Convergence
118 bail = 0;
119 converge = 0;
120 while converge ~= 1 && bail < bailthresh % Find the minimum proportion that makes DIST1 SOSD DIST2
121     middle = (lower + upper) / 2;
122     if SOSDIntegralTestv3(DIST1 + middle, DIST2) == 1
123         upper = middle;
124     else
125         lower = middle;
126     end
127     if lower > upper
128         error('lower > upper!')
129     end
130     if (upper - lower) <= thresh % Convergence acheived when upper and lower are within thresh tolerance
131         converge = 1;
132     end
133     bail = bail + 1;
134 end
135
136 if converge == 1
137     Result(DIST1MINPROP,1) = upper;
138     if Result(MEANDIST2,1) > 0
139         Result(RELCOMPSOSDBASE,1) = upper / Result(MEANDIST2,1);
140     end
141 else % Golden Section Search Failed to converge because Threshold not met
142     Result(DIST1MINPROP,1) = -777777;
143     Result(RELCOMPSOSDBASE,1) = -777777;
144 end
145
146 % Initialize Golden Section Search upper and lower starting points
147 if tcomp > tbase
148     upper = tcomp;
149     lower = tbase;
150 elseif tcomp < tbase
151     upper = tbase;
152     lower = tcomp;
153 else
154     upper = tbase + 10;
155     lower = tbase - 10;
156 end
157
158 % Initialize Flags to Test for Convergence
159 bail = 0;
160 converge = 0;
161 while converge ~= 1 && bail < bailthresh % Find the minimum proportion that makes DIST2 SOSD DIST1
162     middle = (lower + upper) / 2;
163     if SOSDIntegralTestv3(DIST2, DIST1 + middle) == 1
164         lower = middle;
165     else
166         upper = middle;
167     end
168     if lower > upper
169         error('lower > upper!')

```

```

170     end
171     if (upper - lower) <= thresh % Convergence acheived when upper and lower are within thresh tolerance
172         converge = 1;
173     end
174     bail = bail + 1;
175 end
176
177 if converge == 1
178     Result(DIST2MINPROP,1) = lower;
179     if Result(MEANDIST2,1) > 0
180         Result(RELBASESOSDCOMP,1) = lower / Result(MEANDIST2,1);
181     end
182 else
183     Result(DIST2MINPROP,1) = -777777;
184     Result(RELBASESOSDCOMP,1) = -777777;
185 end
186
187 if Result(PCROPFAILDIST1,1) >= 0 && Result(PCROPFAILDIST2,1) >= 0
188     Result(CFDIFF,1) = Result(PCROPFAILDIST1,1) - Result(PCROPFAILDIST2,1);
189 else
190     Result(CFDIFF,1) = -999999;
191 end
192
193 % Categorize Risk: DIST1 More Risky (-1)/less Risky (1)/ Indeterminant (0) compared to DIST2
194 if Result(DIST1MINPROP,1) <= -777777 || Result(DIST2MINPROP,1) <= -777777
195     Result(DELTARISK,1) = -999999;
196 elseif Result(DIST1MINPROP,1) > 0 && Result(DIST2MINPROP,1) > 0
197     Result(DELTARISK,1) = -1;
198 elseif Result(DIST1MINPROP,1) < 0 && Result(DIST2MINPROP,1) < 0
199     Result(DELTARISK,1) = 1;
200 else
201     Result(DELTARISK,1) = 0;
202 end
203 end
204
205
206

```