

```

1  -- vim: ts=2 sw=2 et :
2  -- LOOK.LUA: landscape analysis
3  -- (c) 2022 Tim Menzies, timm@ieee.org, BSD-2 license
4  local l,L = require"lib", require"look"
5  local any,cli,csv,main,map = l.any, l.cli, l.csv, l.map
6  local o, oo,shuffle,sort = l.o, l.oo, l.shuffle, l.sort
7  local ROW,ROWS = L.ROW, L.ROWS
8  local the = cli(L.the,L.help)
9  -----
10 local go,no={},{} -- place to store enabled and disabled tests
11
12 function go.the()
13   if the.lood then oo(the) end; return type(the.seed)=="number" end
14
15 function go.row( n)
16   n=0
17   for r in csv(the.file) do n=n+#r; if the.lood then o(r) end end
18   return n == 3192 end
19
20 function go.egs( rows)
21   rows= ROWS(the.file)
22   if the.lood then map(rows.nums,oo) end
23   return rows.nums[1].hi==8 end
24
25 function go.clone( rows)
26   rows= ROWS(the.file)
27   oo(rows:mid()) end
28
29 function go.dist( r1,rows,ok)
30   ok,rows= true, ROWS(the.file);
31   r1 = rows.rows[1]
32   for _,r2 in pairs(rows.rows) do
33     ok = ok and r2:dist(r1)==0
34     ok = ok and r1:dist(r2) == r2:dist(r1) end
35   return ok end
36
37 function go.around( r1,rows, order)
38   rows = ROWS(the.file);
39   r1 = rows.rows[1]
40   order = rows:around(r1)
41   return order[#order//3].dist < order[#order//2].dist end
42
43 function go.far( rows,r1,r2,ok)
44   ok = true
45   rows = ROWS(the.file);
46   for k=1,50 do
47     r1 = rows:far(any(rows.rows))
48     r2 = rows:far(r1)
49     ok = ok and r1:dist(r2) > .5 end
50   return ok end
51
52 function go.betters( t,n1)
53   t=sort(ROWS(the.file).rows)
54   n1=10
55   for k =1,n1 do oo(t[k].cells) end; print""
56   for k =#t-n1, #t do oo(t[k].cells) end
57   return t[1] < t[#t]
58 end
59
60 function go.look( rs,best,bests,rests,n)
61   for i=1,20 do
62     print("")
63     rs = ROWS(the.file)
64     rs.rows = shuffle(rows.rows)
65     best,bests,rests = rs:look()
66     for n,r in pairs(sort(rs.rows)) do r.rank = n // (#rows.rows // (6/.35)) end
67     for _,r in pairs(bests) do print(r.rank) end
68     n=0
69     for _,r in pairs(rs.rows) do if r.evaluated then n=n+1 end end
70     oo(bests=#bests,rests=#rests,n=n) end
71   return true end
72   -----
73 main(go, the)

```