```lua
-- vim: ts=2 sw=2 et :
-- LOOK.LUA: landscape analysis
-- (c) 2022 Tim Menzies, timm@ieee.org, BSD-2 license
local l,L  = require"lib", require"look"
local any,cli,csv,fmt              = l.any, l.cli, l.csv, l.fmt
local lt, main, many, map          = l.lt, l.main, l.many,l.map
local o, oo,per,shuffle,sort,splice= l.o,l.oo,l.per,l.shuffle,l.sort,l.splice
local NUM,ROW,ROWS                 = L.NUM, L.ROW, L.ROWS
local the                          = cli(L.the,L.help)
-------------------------------------------------------------------------------
local go,no={},{} -- place to store enabled and disabled tests

function go.the()
  if the.loud then oo(the) end; return type(the.seed)=="number" end

function go.row(    n)
  n=0
  for r in csv(the.file) do n=n+#r; if the.loud then oo(r) end end
  return n == 3192 end

function go.egs(    rows)
  rows= ROWS(the.file)
  if the.loud then map(rows.ys,oo) end
  return rows.ys[1].hi==5140 and rows.ys[1].lo==1613 end

function go.clone(    rows)
  rows= ROWS(the.file)
  return rows:mid()[3]==20 end

function go.dist(    r1,rows,ok)
  ok,rows= true, ROWS(the.file);
  r1 = rows.all[1]
  for _,r2 in pairs(rows.all) do
    ok = ok and r2:dist(r2)==0
    ok = ok and r1:dist(r2) == r2:dist(r1) end
  return ok end

function go.around(    r1,rows, order)
  rows = ROWS(the.file);
  r1 = rows.all[1]
  order = rows:around(r1)
  return order[#order//3].dist < order[#order//2].dist  end

function go.far(    rows,r1,r2,ok)
  ok = true
  rows = ROWS(the.file);
  for k=1,10 do
    r1 = rows:far(any(rows.all))
    r2 = rows:far(r1)
    ok = ok and r1:dist(r2) > .5 end
  return ok end

function go.betters(  t,n1)
  t=sort(ROWS(the.file).all)
  n1=10
  for k =1,n1 do oo(t[k].cells) end; print""
  for k =#t-n1, #t do oo(t[k].cells) end
  return t[1] < t[#t]
end

function go.how(  t,n,bests,rests)
  t     = ROWS(the.file)
  t.all = sort(t.all)
  n     = (#t.all)^.5 // 1
  bests = splice(t.all, 1,  n)
  rests = splice(t.all, n+1, #t.all)
  t:how(bests,rests)
  oo(t:clone(bests):mid())
  oo(t:clone(rests):mid())
  return true
end

function go.look(   rows,best,bests,rests,n,names,b4,guess,b,g)
  rows = ROWS(the.file)
  names=map(rows.ys,function(col) return col.txt end)
  b=NUM()
  g=NUM()
  b4=rows:mid()
  for i=1,10 do
    rows = ROWS(the.file)
    rows.all = shuffle(rows.all)
    best,bests,rests = rows:look()
    for n,r in pairs(sort(rows.all)) do r.rank = math.floor(100*n/#rows.all //1) e
nd
    n=0;for _,r in pairs(rows.all) do if r.evaluated then n=n+1 end end
    guess=rows:clone(many(rows.all,n))
    for _,rank in pairs(map(sort(bests,lt"rank"),function(r) return r.rank end)) do
 b:add(rank) end
    for _,rank in pairs(map(sort(guess.all,lt"rank"),function(r) return r.rank end)
) do g:add(rank) end
    print(fmt("%20s %20s %20s",
          o(names),o(b4),
          o(rows:clone(bests):mid()),
          o{bests=#bests,rests=#rests,evalled=n})) end
  for _,p in pairs{0,.2,.4,.6,.8}  do io.write(per(b:has(),p)," ") end; print""
  for _,p in pairs{0,.2,.4,.6,.8}  do io.write(per(g:has(),p)," ") end; print""
  return true end
-------------------------------------------------------------------------------
main(go, the)
```