

```

1  -- SHORTR.lua (c) 2022, Tim Menzies <timm@ieee.org>, BSD2 license.
2  -- Semi-supervised multi-objective optimization XAI. From N
3  -- items, find and explain the best ones, using just log(N)
4  -- evals. All in a few hundreds lines of LUA.
5
6
7  --
8  --
9  --
10 --
11 --
12 --
13
14 local lib = {}
15
16 -----
17 --> lint() :nil -> report rogue globals
18 local b4={} ; for x,_ in pairs(_ENV) do b4[x]=x end
19 function lib:lint() --> {}
20   for x,v in pairs(_ENV) do if not b4[x] then print("'"..x,type(v)) end end end
21
22 -----
23 --> big :num -> large number
24 --> fmt(:str, :str) :str -> emulate printf
25 --> R( max:num=1) :num -> return random number 0..max
26
27 lib.big = math.huge
28 lib.fmt = string.format
29 lib.R = math.random
30
31 -----
32 --> obj(name :str) :klass -> class creator
33 local _id = 0
34 function lib.obj(name, t,new)
35   function new(kl,...)
36     _id = _id + 1
37     local x=setmetatable(({id=_id,kl}); kl.new(x,...); return x end
38     t = {__tostring=lib.o, is=name or ""}; t.__index=t
39     return setmetatable(t, {__call=new}) end
40
41 -----
42 --> thing(x :str) :any -> coerce x to some LUA type
43 function lib.thing(x)
44   if type(x)~="string" then return x end
45   x = x:match"%s*(-)%s*"
46   if x=="true" then return true elseif x=="false" then return false end
47   return math.tointeger(x) or tonumber(x) or x end
48
49 -----
50 --> help(x :str) :tab -> For lines with '--', pull keys:defaults.
51 -- Look for updates for "key" on command-line. Things with boolean defaults
52 -- are negated via '--flag'. Other keys need '--flag value'.
53 function lib.help(str)
54   local t = {}
55   str:gsub("%n ([^-]|(^[^s]+))([s]+)(-[^s]+|^n)^s([^[s]+)",function(f1,k,f2,x)
56     for n,flag in ipairs(arg) do if flag==f1 or flag==f2 then
57       x = x=="false" and"true" or x=="true" and"false" or arg[n+1] end end
58     t[k] = lib.thing(x) end
59   if t.help then print(str:gsub("%u[%u%d]+","%27[1:32m%1\27[0m)","") end
60   return t end
61
62 -----
63 --> csv(arc :str, fun :function) :nil -> for file lines, split on ",", pass to fun
64 function lib.csv(file, fun, line,t)
65   file = io.input(file)
66   line = io.read()
67   while line do
68     t={} ; for x in line:gmatch("(^[^,]+)") do t[1+#t]=lib.thing(x) end; fun(t)
69     line = io.read() end
70   io.close(file) end

```