```lua
1   local the, help = {}, [[
2
3   ween.lua [OPTIONS]
4   ween (vb), archaic. To think or imagine.
5
6   A small sample multi-objective optimizer / data miner.
7   (c)2021 Tim Menzies <timm@ieee.org> unlicense.org
8
9   OPTIONS:
10   -best    X  Best end  of the examples.         = .5
11   -debug   X  Run one test, show stack dumps on fail. = ing
12   -file    X  Read data from files.               = ../../data/auto93.csv
13   -h          Show help.                          = false
14   -hints   X  How many  to evaluate each iteration. = 4
15   -p       X  Coefficient on distance calculation. = 2
16   -seed    X  Random number seed.                 = 10019
17   -todo    X  Demos to run at start-up. 'all'=run all. = ing]]
18
19   local b4={}; for k,_ in pairs(_ENV) do b4[k]=k end
20   local function rogues()
21     for k,v in pairs(_ENV) do if not b4[k] then print("?:",k,type(v)) end end end
22   ---------------------------------------------------------------------------------
23
24   local randi,rand,Seed -- remember to set seed before using this
25   function randi(lo,hi) return math.floor(0.5 + rand(lo,hi)) end
26   function rand(lo,hi)
27     lo, hi = lo or 0, hi or 1
28     Seed = (16807 * Seed) % 2147483647
29     return lo + (hi-lo) * Seed / 2147483647 end
30
31   local pop,csv,fmt,map,keys,sort,copy,norm,push
32   local color,first,firsts,coerce,second22,shuffle,bchop
33   fmt = string.format
34   function coerce(x)        return tonumber(x) or x end
35   function color(n,s)       return fmt("\27[1m\27[%sm%s\27[0m",n,s) end
36   function copy(t,  u)      u={};for k,v in pairs(t) do u[k]=v    end; return u end
37   function keys(t,u)        u={};for k,_ in pairs(t) do u[1+#u]=k end; return sort(u)end
38   function first(x)         return x[1] end
39   function firsts(x,y)      return x[1] < y[1] end
40   function norm(lo,hi,x)    return math.abs(lo-hi)<1E-9 and 0 or (x-lo)/(hi-lo) end
41   function pop(t)           return table.remove(t) end
42   function push(t,x)        table.insert(t,x); return x end
43   function second22(_,t)      return t[2] end
44   function sort(t,f)        table.sort(t,f); return t end
45   function shuffle(t,   j)
46     for i=#t,2,-1 do j=randi(1,i); t[i],t[j]=t[j],t[i] end; return t end
47   function map(t,f,   u)
48     u,f = {},f or same; for k,v in pairs(t) do push(u, f(k,v)) end; return u end
49
50   function csv(file)
51     file = io.input(file)
52     return function(    t,x)
53       x = io.read()
54       if x then
55         t={};for y in x:gsub("%s*","") :gmatch("([^,]+)") do push(t,coerce(y)) end
56         if #t>0 then return t end
57       else io.close(file)  end end end
58
59   local shout,out
60   function shout(x) print(out(x)) end
61   function out(t,     u,key,val)
62     function key(_,k) return string.format(":%s %s", k, out(t[k])) end
63     function val(_,v) return out(v) end
64     if type(t) ~= "table" then return tostring(t) end
65     u = #t>0 and map(t, val) or map(keys(t), key)
66     return "{".. table.concat(u,"").. "}" end
67
68   function bchop(t,val,policy,      lo,hi,mid)
69     lt = lt or function(x,y) return x < y end
70     lo,hi = 1,#t
71     while lo <= hi do
72       mid = (lo+hi) // 2
73       if policy(t[mid],val) then lo=mid+1 else hi= mid-1 end end
74     return math.min(lo,#t)   end
75   ---------------------------------------------------------------------------------
76   local slurp,sample,dist,ordered,hint,left_is_best
77   function slurp( i) for eg in csv(the.file) do i=sample(i,eg) end; return i end
78
79   function sample(i,eg)
80     local numeric,independent,dependent,head,data,datum
81     i = i or {n=0,xs={},nys=0,ys={},lo={},hi={},w={},egs={},heads={},divs={}}
82     function head(n,x)
83       function numeric()      i.lo[n]= math.huge; i.hi[n]= -i.lo[n] end
84       function independent() i.xs[n]= x end
85       function dependent()
86         i.w[n]  = x:find"-" and -1 or 1
87         i.ys[n] = x
88         i.nys   = i.nys+1 end
89       if not x:find":" then
90         if x:match"^[A-Z]" then numeric() end
91         if x:find"-" or x:find"+" then dependent() else independent() end end
92       return x end
93     function datum(n,x)
94       if x ~= "?" then
95         if i.lo[n] then
96           i.lo[n] = math.min(i.lo[n],x)
97           i.hi[n] = math.max(i.hi[n],x) end end
98       return x end
99     if #i.heads==0 then i.heads=map(eg,head) else push(i.egs,map(eg,datum)) end
100    i.n = i.n + 1
101    return i end
102
103  function left_is_best(i,left,right,      a,b,lefts,rights)
104    lefts,rights=0,0
105    for n,_ in pairs(i.ys) do
106      a  = norm(i.lo[n], i.hi[n], left[n])
107      b  = norm(i.lo[n], i.hi[n], right[n])
108      lefts  = lefts  - 2.71828^(i.w[n] * (a-b)/i.nys)
109      rights = rights - 2.71828^(i.w[n] * (b-a)/i.nys) end
110    return lefts/i.nys < rights/i.nys end
111
112  function ordered(i,egs)
113    return sort(egs or i.egs, function(a,b) return  left_is_best(i,a,b) end) end
114
115  function dist(i,eg1,eg2)
116    local function dist1(lo,hi,a,b)
117      if  lo
118      then if      a=="?" then b=norm(lo,hi,b); a = b>.5 and 0 or 1
119           elseif b=="?" then a=norm(lo,hi,a); b = a>.5 and 0 or 1
120           else          a,b = norm(lo,hi,a), norm(lo,hi,b) end
121           return math.abs(a-b)
122      else return a==b and 0 or 1 end
123    end --------------------
124    local d,n = 0,0
125    local a,b,inc
126    for col,_ in pairs(i.xs) do
127      a,b = eg1[col], eg2[col]
128      inc = a=="?" and b=="?" and 1 or dist1(i.lo[col],i.hi[col],a,b)
129      d   = d + inc^the.p
130      n   = n + 1 end
131    return (d/n)^(1/the.p) end
132
133  function hint(i,egs)
134    local function hint1(egs, all, min, evals,lvl)
135      local scoreds,nearest,best = {}
136      function nearest(_,eg, tmp)
137        return sort(map(scoreds, function(rank,scored)
138                       return {rank+dist(i,eg,scored)/10^6, eg} end),firsts)[1] end
139      if   #egs <= 2*min
140      then scoreds = egs
141           return map(sort(map(all,nearest),firsts), second22)
142      else for j=1,the.hints do push(scoreds, pop(egs)) end
143           best, scoreds = {}, ordered(i,scoreds)
144           egs = sort(map(egs,nearest),firsts)
145           for j=1,(#egs)//2 do push(best, egs[j][2]) end
146           return hint1(best, all, min, evals+the.hints,lvl.."|.. ") end
147    end ---------------
148    egs = egs or i.egs
149    return hint1( copy(shuffle(egs)), egs, (#egs)^the.best, 0,"") end
150
```

```lua
151  ---------------------------------------------------------------------------------
152  local go={}
153  function go.ing() return true end
154  function go.the() shout(the) end
155  function go.csv() for eg in csv(the.file) do shout(eg) end end
156  function go.more(   u,t)
157    t= {10,20,30,40}
158    u= copy(shuffle(t))
159    t[1]=100
160    assert(u[1] ~= t[1])
161    assert(u[1] ~= 100) end
162
163  function go.sample(  s)
164    s=slurp()
165    assert(398 == #s.egs)
166    assert(3 == s.lo[1]) end
167
168  function go.ordered(  _,i,egs)
169    egs,i = ordered(slurp())
170    shout(i.heads)
171    for j=1,5 do shout(egs[j]) end
172    print("#")
173    for j=#egs-5,#egs do shout(egs[j]) end end
174
175  function go.dist(  i,dist1,t)
176    function dist1(_,eg) return {dist(i,i.egs[1],eg), eg} end
177    i = slurp()
178    t=map(i.egs,dist1)
179    for j=1,5 do print(j,fmt("%5.3f",t[j][1]),out(t[j][2])) end
180    print("#")
181    for j=(#t)-5,#t do print(j,fmt("%5.3f",t[j][1]),out(t[j][2])) end end
182
183  function go.hint(  i,sort1,sort2,s,lt)
184    function lt(a,b)  return left_is_best(i,a,b) end
185    i=slurp()
186    sort1= ordered(i)
187    sort2= hint(i)
188    for m,eg in pairs(sort2) do
189      if m < 20 then shout(eg) end
190      if m > (#sort2)-20 then shout(eg) end end end
191
192  -- Run demos, each time resetting random seed and the global config options.
193  -- Return to the operating system then number of failing demos.
194  local function main()
195    local no,defaults,reset = 0,copy(the)
196    function reset(x)  Seed=the.seed or 10019; the= copy(defaults) end
197    reset()
198    go[ the.debug ]()
199    for _,it in pairs(the.todo=="all" and keys(go) or {the.todo}) do
200      if type(go[it]) ~= "function" then return print("NOFUN:",it) end
201      reset()
202      local ok,msg = pcall( go[it] )
203      if ok then print(color("PASS ",31)..it)
204               else print(color("FAIL ",31)..it,msg); no=no+1 end end
205    rogues()
206    os.exit(no)  end
207
208  ---------------------------------------------------------------------------------
209  -- Make 'the' options array from help string and any updates from command line.
210  (help or ""):gsub("^.*OPTIONS:","") :gsub("\n%s*-([^%s]+)[^\n]*%s([^%s]+)",
211        function(flag,x)
212          for n,word in ipairs(arg) do if word=="-"..flag) then
213            x = x=="false" and "true" or tonumber(arg[n+1]) or arg[n+1] end end
214          if x=="false" then x=false elseif x=="true" then x=true end
215          the[flag]=x end)
216
217  if the.h then print(help) else main() end
```