

```

1  #!/usr/bin/env lua
2  local l=require("lib")
3  local the=l.settings[[
4
5  L5 : a lean little learning library, in LUA
6  (c) 2022 Tim Menzies <timmeeee.org> BSD-2 license
7
8  USAGE: lua l5.lua [OPTIONS]
9
10 OPTIONS:
11 -e --eg      start-up example          = nothing
12 -b --bins    max number of bins        = 16
13 -d --dump    on test failure, exit with stack dump = false
14 -f --file    file with csv data        = ../data/auto93.csv
15 -F --Far     how far to look for poles (max=1) = .95
16 -h --help    show help                  = false
17 -m --min     min size. If<1 then t*min else min. = 10
18 -n --nums    number of nums to keep     = 512
19 -p --p       distance calculation coefficient = 2
20 -r --rest    size of "rest" set         = 3
21 -s --seed    random number seed        = 10019
22 -S --Sample  how many numbers to keep   = 10000 ]]
23
24 local copy, csv, map, obj, oo = l.copy, l.csv, l.map, l.obj, l.oo
25 local push, sort = l.push, l.sort
26 local Data, Num, Row, Skip, Sym
27
28 -----
29 Skip=obj"Skip"
30 function Skip:new(c,x) return (at=c,txt=x) end
31 function Skip:add(x) return x end
32 function Skip:dist(v1,v2) return 0,0 end
33
34 -----
35 Sym=obj"Sym"
36 function Sym:new(c,x) return (at=c or 0,txt=x or "",has={}) end
37 function Sym:add(x) if x=="?" then self.has[x]=1+(self.has[x] or 0) end end
38 function Sym:dist(v1,v2)
39   return (v1=="?" and v2=="?" and 1 or v1==v2 and 0 or 1),1 end
40
41 -----
42 Num=obj"Num"
43 function Num:new(c,x)
44   return (at=c or 0,txt=x or "",lo=-1E32,hi=-1E32, has={},
45         w=(x or ""):find"$" and -1 and 1) end
46 function Num:add(x)
47   if x=="?" then self.lo = math.min(x,self.lo)
48     self.hi = math.max(x,self.hi)
49   push(self.has,x) end end
50 function Num:norm(n)
51   return n=="?" and n or (n-self.lo)/(self.hi-self.lo + 1E-32) end
52
53 function Num:dist(v1,v2)
54   if v1=="?" and v2=="?" then return 1,1 end
55   v1,v2 = self:norm(v1), self:norm(v2)
56   if v1=="?" then v1 = v2<.5 and 1 or 0 end
57   if v2=="?" then v2 = v1<.5 and 1 or 0 end
58   return math.abs(v1-v2),1 end
59
60 -----
61 Row=obj"Row"
62 function Row:new(data,t) return (_data=data,cells=t) end
63 function Row:around(rows)
64   return sort(map(rows, function(r) return (row=r,d=self-r) end),lt="d") end
65 function Row:far(rows)
66   return per(self:around(rows),the.far).row end
67
68 function Row:__sub(row, d,n,dl,nl)
69   d,n = 0,0
70   for i,col in pairs(self._data.cols.x) do
71     dl,nl= col:dist(self.cells[col.at], row.cells[col.at])
72     n = n + nl
73     d = d + dl*the.p end end
74   return (d/n)^(1/the.p) end
75
76 function Row:__lt(row)
77   self.evald, row.evald = true,true
78   local s1,s2,d,n,x,y=0,0,0,0,0
79   local ys = self._data.cols.y
80   for _,col in pairs(ys) do
81     x,y= self.cells[col.at], row.cells[col.at]
82     x,y= col:norm(x), col:norm(y)
83     s1 = s1 - 2.71828*(col.w * (x-y)/#ys)
84     s2 = s2 - 2.71828*(col.w * (y-x)/#ys) end
85   return s1/#ys < s2/#ys end
86
87 -----
88 Data=obj"Data"
89 function Data:new(src)
90   self.rows, self.cols = {}, {all={},x={},y={}}
91   if type(src)=="string"
92   then csv(src, function(row) self:add(row) end)
93   else map(src or {}, function(row) self:add(row) end) end end
94
95 function Data:add(row, what)
96   function what(x)
97     return x:find"$" and Skip or (x:find"[A-Z]" and Num or Sym) end
98   if #self.cols.all==0
99   then for c,x in pairs(row) do
100     local col = push(self.cols.all, what(x)(c,x))
101     push(x:find"[-+]" and self.cols.y or self.cols.x, col) end
102   else row = row.cells and row or Row(self,row)
103     for c,col in pairs(self.cols.all) do col:add(row.cells[c]) end
104     push(self.rows, row) end end
105
106 function Data:cheat( m,n)
107   for i,row in pairs(sort(self.rows)) do
108     row.rank = math.floor(100*i/#self.rows)
109     row.evald = false end
110   self.rows = l.shuffle(self.rows) end
111
112

```

```

112 -----
113 local eg = {}
114 local function run( fails,old)
115   fails=0
116   the = l.cli(the)
117   old = copy(the)
118   for k,fun in pairs(eg) do
119     if the.eg == "all" or the.eg == k then
120       for k,v in pairs(old) do the[k]=v end
121       math.randomseed(the.seed)
122       print("\n>>>>>",k)
123       if not fun() then fails = fails+1 end end end
124   l.rogues()
125   os.exit(fails) end
126
127 function eg.the() oo(the); return true end
128
129 function eg.num( z)
130   z=Num(); for i=1,100 do z:add(i) end; print(z); return true end
131
132 function eg.sym( z)
133   z=Sym(); for _,x in pairs{1,1,1,1,2,2,3} do z:add(x) end;
134   print(z); return true end
135
136 function eg.data( d)
137   d=Data(the.file); map(d.cols.x,print) return true end
138
139 function eg.dist( num,d,r1,r2)
140   d=Data(the.file)
141   num=Num()
142   for i=1,100 do
143     r1=l.any(d.rows)
144     r2=l.any(d.rows)
145     r3=r1:far(self.rows)
146     print(r3-r1)
147     num:add(r2-r1) end
148   oo(sort(num.has))
149   return true end
150
151 run()

```