

```

1  -----
2  --
3  --
4  --
5  --
6  --
7  --
8  --
9  -----
10 local help={
11
12 bore == best or rest
13 (c) 2022, Tim Menzies, BSD 2-clause license.
14
15 USAGE:
16   lua bore.lua [OPTIONS]
17
18 OPTIONS:
19   -Dump          stack dump on error = false
20   -Format S      format string          = %5.2f
21   -best F        best space              = .15
22   -cohen F       Cohen's delta           = .35
23   -data N        data file               = etc/data/auto93.csv
24   -furthest F    far                     = .9
25   -help          show help               = false
26   -seed I        random seed             = 10019
27   -todo S        start-up action         = nothing
28
29 }
30 -----
31 --
32 --
33 --
34 --
35 --
36 --
37 local b4={}; for k,_ in pairs(_ENV) do b4[k]=k end
38 local big  = 1E32
39 local tiny = 1E-32
40 local the  = {}
41
42 local function atom(x)
43   if type(x)~="string" then return x end
44   x = x:match("^%s*(-)%s*$")
45   if x=="true" then return true else if x=="false" then return false end
46   return tonumber(x) or x end
47
48 local function atoms(x, t)
49   t={}; for y in x:gmatch("or"((^)+)) do t[1+#t]=atom(y) end; return t end
50
51 local function cli(txt, t)
52   t={}
53   txt:gsub("\n [^-]([^\s+][^\n]*%s([^\s+])",function(key,x)
54     for n,flag in ipairs(arg) do
55       if flag:sub(1,1)=="-" and key:find("^"..flag:sub(2)..".*") then
56         x = x=="false" and true or x=="true" and "false" or arg[n+1] end end
57       t[key] = atom(x) end
58   return t end
59
60 local fmt = string.format
61
62 local function sort(t,f) table.sort(t,f); return t end
63
64 local function slots(t, u)
65   u={}; for k,v in pairs(t) do l=tostring(k); if l:sub(1,1)~="-" then u[1+#u]=k
66   end end;
67   return sort(u) end
68
69 local function main(the, help, demos)
70   if the.help then print(help) else
71     for _,todo in pairs(the.todo=="all" and slots(demos) or {the.todo}) do
72       math.randomseed(the.seed)
73       if type(demos[todo])=="function" then demos[todo]() end end end
74     os.exit(demos.fails) end
75
76 local function map(t,f, u)
77   u={};for k,v in pairs(t) do u[1+#u]=f(v) end; return u end
78
79 local function tablep(t) return type(t)=="table" end
80
81 local function o(t, seen)
82   seen = seen or {}
83   if not tablep(t) then return tostring(t) end
84   if seen[t] then return "..." end
85   seen[t]=t
86   local key=function(k) return fmt("%s %s",k,o(t[k],seen)) end
87   local u= #t>0 and map(t,function(x) return o(x,seen) end) or map(slots(t),key)
88
89   return ' { '..table.concat(u, " " ..")" end
90
91 local function oo(t) print(o(t)) end
92
93 local function rows(file, x,prep)
94   file = io.input(file)
95   return function()
96     x=io.read(); if x then return atoms(x) else io.close(file) end end end
97
98 local function sum(t,f, n)
99   n=0; for _,v in pairs(t) do n=n+f(v) end; return n end
100
101 local function tree(t, seen, pre, txt, v)
102   pre, seen = pre or "", seen or {}
103   if not tablep(t) then return print(fmt("%s%s",pre,t)) end
104   if seen[t] then return print(fmt("%s...",pre)) end
105   seen[t]=t
106   for _,k in pairs(slots(t)) do
107     v= t[k]
108     if tablep(v)
109       then print(fmt("%s%s", pre,k)); tree(v,seen,pre .. " " )
110     else print(fmt("%s%s %s",pre,k,v)) end end end

```

## Functions

```

109 -----
110 --
111 --
112 --
113 --
114 --
115 --
116 local as=setmetatable
117 local function obj( t)
118   t={__tostring=0}; t.__index=t
119   return as(t, {__call=function(_,...) return t.new(_,...) end}) end
120
121 --
122 --
123 --
124 --
125 --
126 --
127 --
128 --
129 --
130 --
131 --
132 --
133 --
134 --
135 --
136 --
137 --
138 --
139 --
140 --
141 --
142 --
143 --
144 --
145 --
146 --
147 --
148 --
149 --
150 --
151 --
152 --
153 --
154 --
155 --
156 --
157 --
158 --
159 --
160 --
161 --
162 --
163 --
164 --
165 --
166 --
167 --
168 --
169 --
170 --
171 --
172 --
173 --
174 --
175 --
176 --
177 --
178 --
179 --
180 --
181 --
182 --
183 --
184 --
185 --
186 --
187 --
188 --
189 --
190 --
191 --
192 --
193 --
194 --
195 --
196 --
197 --
198 --
199 --
200 --
201 --
202 --
203 --
204 --
205 --
206 --
207 --
208 --
209 --
210 --
211 --
212 --
213 --
214 --
215 --
216 --
217 --
218 --
219 --
220 --
221 --
222 --
223 --
224 --
225 --
226 --
227 --

```

## Classes

### COL

```

124 local function col(at,txt, i)
125   i = {n=0, at=at or 0, txt=txt or "", has={}}
126   i.w = i.txt:find"$" and -1 or 1
127   return i end

```

```

128 local function add(self,x,inc)
129   if x=="?" then
130     inc = inc or 1
131     self.n = self.n + inc
132     self:add(x,inc) end
133   return self end

```

### NUM

```

139 local Num=obj{}
140 function Num:new(at,x, new)
141   new = as(col(at,x),Num)
142   new.mu, new.m2, new.lo, new.hi = 0, 0, big, -big
143   return new end

```

```

145 function Num:add(x,_, d)
146   d = x - self.mu
147   self.mu = self.mu + d/self.n
148   self.m2 = self.m2 + d*(x - self.mu)
149   self.sd = (self.n<2 or self.m2<0) and 0 or (self.m2/(self.n-1))^.5
150   if x > self.hi then self.hi = x end
151   if x < self.lo then self.lo = x end end

```

```

153 function Num:norm(x)
154   return self.hi-self.lo<tiny and 0 or (x-self.lo)/(self.hi-self.lo) end
155
156 function Num:heaven(x, heaven)
157   return ((self.w>0 and 1 or 0) - self:norm(x))^the.p end

```

### SYM

```

162 local Sym=obj{}
163 function Sym:new(at,x,inc, new)
164   new=as(col(at,x),Sym); new.most=0; return new end
165
166 function Sym:add(x,inc)
167   self.has[x] = inc + (self.has[x] or 0)
168   if self.has[x] > self.most then self.most,self.mode=self.has[x],x end end
169
170 function Sym:div()
171   local function plopg(n, p) p=n/self.n; return p*math.log(p,2) end
172   return -sum(self.has, plopg) end

```

### SKIP

```

177 local Skip=obj{}
178 function Skip:new(at,x) return as(col(at,x),Skip) end
179 function Skip:add(x,inc) return x end

```

### COLS

```

184 local Cols=obj{}
185 function Cols:new(headers, self,col,here)
186   self = as({all={}, x={}, y={}}, Cols)
187   for at,x in pairs(headers) do
188     if x:find"$" then self.all[at] = Skip(at,x) else
189       col = {x:find"[A-Z]" and Num or Sym}(at,x)
190       self.all[at] = col
191       here = x:find"[+-$]" and self.y or self.x
192       here[1+#here] = col end end
193   return self end

```

```

194 function Cols:add(t)
195   for _,col in pairs(self.all) do col:add(t[col.at]) end
196   return t end

```

```

199 function Cols:clone(rows, new)
200   new = new or Cols(map(self.cols.all, function(x) return x.txt end))
201   for _,row in pairs(rows or {}) do new:add(row) end
202   return {rows=rows,cols=new} end

```

### DATA

```

207 local Data=obj{}
208 function Data:new(inits, new)
209   new = as({rows={}, heavens=Num()},Data)
210   if type(inits)=="string" then for row in csv(inits) do new:add(row) end end
211   if type(inits)=="table" then for _,row in pairs(inits) do new:add(row) end end
212   return new end

```

```

214 function Data:add(t, n)
215   if self.cols then self:addData(t) else
216     self.cols = Cols(t)
217     self.best = self.cols:clone()
218     self.rest = self.cols:clone() end end

```

```

220 function Data:addData(t, n)
221   self.rows[1+#self.rows] = self.cols:add(t)
222   n = self.heavens.norm( self.heavens.add(self.heaven(t)))
223   (n>=the.best and self.best or self.rest):add(t) end

```

```

225 function Data:heaven(t)
226   heaven = function(col) return col:heaven(t[col.at]) end
227   return (sum(self.cols.y,heaven)/#self.cols.y)^(1/the.p) end

```

```

228 -----
229 --
230 --
231 --
232 --
233 --
234 --
235 local Demos = {fails=0}
236
237 local function asserts(test, msg)
238     print(test and "PASS: " or "FAIL: ", msg or "")
239     if not test then
240         Demos.fails = Demos.fails+1
241         if the.Dump then assert(test, msg) end end end
242
243 function Demos.the()      oo(the) end
244 function Demos.col()      oo(col(10, "Mpg-")) end
245 function Demos.num( n)    n=Num();
246     for x=1,1000 do add(n,x) end; print(n) end
247
248 function Demos.sym( s)
249     s=Sym(); for _,x in pairs{1,1,1,1,2,2,3} do add(s,x) end
250     asserts(s:div() - 1.376 < 0.005, "entropy") end
251
252 function Demos.cols( c)
253     print(Cols({{"Clnrs", "Weight", "Hp:", "Lbs-",
254         "Acc+", "Model", "origin", "Mpg+"}}))
255 end
256
257 the = cli(help)
258 main(the, help, Demos)

```