

```

1 local help = {}
2
3 bore: best or rest. u show me a good lower and i'll show u a loser.
4 (c) 2022, Tim Menzies, timm@ieee.org, opensource.org/licenses/Fair
5
6 USAGE:
7   alias twk="lua tweak.lua "
8   twk [OPTIONS]
9
10 OPTIONS:
11   --bins      -b max bins          = 16
12
13 OPTIONS (other):
14   --seed      -s random number seed = 10019
15   --file      -f where to find data  = ../etc/data/auto2.csv
16   --dump      -d dump stack+exit on error = false
17   --help      -h show help          = false
18   --go        -g start up action     = nothing
19 }}
20
21 local function thing(x)
22   x = x:match("^%s*(~)%s*$"
23   if x=="true" then return true elseif x=="false" then return false end
24   return math.tointeger(x) or tonumber(x) or x end
25
26 local the={}
27 help:gsub("u ([~!-])([^%s+])[%s]+(-[^%s+])[^u]%s+([~%s+])",function(f1,k,f2,x)
28   for n,flag in ipairs(arg) do if flag==f1 or flag==f2 then
29     x = x=="false" and true or x=="true" and false or arg[n+1] end end
30   the[k] = thing(x) end)
31
32 local atom, csv, map, o, obj, ok, push, rows, with
33 local _, GO, BIN, NUM, SYM, COLS, ROW, EGS
34 local R, Big
35
36 R=math.random
37 Big=math.huge
38
39 function map(t,f, u) u={}; for k,v in pairs(t) do u[1+#u]=f(v) end;return u end
40 function push(t,x) t[1+#t]=x; return x end
41
42 function with(i,mew,defaults)
43   for k,v in pairs(defaults) do i[k] = v end
44   for k,v in pairs(new) do assert(i[k]~=nil,"missing key"..k); i[k] = v end end
45
46 function csv(f)
47   f = io.input(f)
48   return function(t, u)
49     t=io.read()
50     if not t then io.close(f) else
51       u={}; for x in t:gmatch("[^,]+") do u[1+#u]=thing(x) end
52       return u end end
53
54 function o(t, u)
55   u={}; for k,v in pairs(t) do u[1+#u] = string.format("%s %s",k,v) end
56   return (t.is or "").."["..table.concat(sort(u)," " ).."]" end
57
58 function obj(name, t,new)
59   function new(kl,...)
60     local x=setmetatable({},kl); kl.new(x,...); return x end
61   t = {__tostring=o, is=name or ""}; t.__index=t
62   _=t
63   Return setmetatable(t, {__call=new}) end

```

```

64 SYM=obj"SYM"
65
66 function _..new(i,t) with(i,t,{at=0,txt=""}) end
67 function _..add(i,x) if x=="*" then i.has[x] = 1+(i.has[x] or 0) end end
68 function _..addxy(i,x,y) if x=="*" then i.bins[x]=y+(i.bins[x] or 0) end end
69
70 _..mid=function(i, m,x)
71   m=0; for y,n in pairs(i.has) do if n>m then m,x=y,n end end; return x end
72
73 _..div=function(i, n,e)
74   n=0; for k,m in pairs(i.has) do n = n + m end
75   e=0; for k,m in pairs(i.has) do e = e - m/n*math.log(m/n,2) end
76   return e end
77
78 --init with fields
79
80 BIN=obj"BIN"
81 function _..new(i,t) i.pos,i.txt,i.lo,i.hi,i.y = t.pos,t.txt,t.lo,t.hi,t.y end
82 function _..of(i,x) return i.y.has[x] or 0 end
83
84 function _..select(i,t, x)
85   t = t.cells and t.cells or t
86   x = t[i.pos]
87   return x=="*" or i.lo == i.hi and i.lo == x or i.lo <= x and x < i.hi end
88
89 function _..tostring(i)
90   local x,lo,hi,big = i.txt, i.lo, i.hi, Big
91   if lo == hi then return fmt("%s==%s",x, lo)
92   elseif hi == big then return fmt("%s>=%s",x, lo)
93   elseif lo == -big then return fmt("%s<=%s", x, hi)
94   else return fmt("%s<=%s<%s",lo,x,hi) end end
95
96 NUM=obj"NUM"
97 function _..new(i,n,s) i.at,i.txt, i.lo,i.hi,i.bins=n,s,Big,-Big,{} end
98 function _..norm(i,x) return x=="*" and x or (x-i.lo)/(i.hi - i.lo) end
99
100 function _..addx(i,x)
101   if x=="*" then return x end
102   if x > i.hi then i.hi=x elseif x<i.lo then i.lo=x end end
103
104 function _..addxy(i,x,y)
105   if x=="*" then return x end
106   x = math.max(1, math.min(the.bins, the.bins*i:norm(x) // 1))
107   i.bins[x] = i.bins[x] or Sym()
108   i.bins[x]:add(y) end
109
110 ROW=obj"ROW"
111 function _..new(i,egs,t) i.cells,i.data = t,egs end
112
113 COLS=obj"COLS"
114 function _..new(i,names, col)
115   i.all,i.x,i.y,i.names = {}, {}, {}, names
116   for at,txt in pairs(names) do
117     col = push(i.all, (txt:find("[A-Z]+") and Num or Sym) (at,txt))
118     if not txt:find"["..!$%* and i.y or i.x,col) end end end
119     push(txt:find"["..!$%* and i.y or i.x,col) end end end
120
121 EGS=obj"EGS"
122 function _..new(i) i.rows,i.cols= {},nil end
123 function _..file(i,f) for row in csv(f) do i.add(row) end end
124 function _..add(i,t)
125   if i.cols
126     then t = push(i.rows, t.cells and t or ROW(i,t)).cells
127     for k,col in pairs(i.cols.all) do col:add(t[col.pos]) end
128   else i.cols = COLS(t) end end

```

```

129 GO=obj"GO"
130
131 function ok(test,msg)
132   print("", test and "PASS" or "FAIL", msg or "")
133   if not test then
134     GO.fail= GO.fail+1
135     if the.dump then assert(test,msg) end end end
136
137 function _..new(i,todo, b4,go)
138   b4={}; for k,v in pairs(the) do b4[k]=v end
139   go={}; for k,_ in pairs(GO) do
140     if k=="new" and type(GO[k])=="function" then go[1+#go]=k end end
141   GO.fail= 0
142   for _,x in pairs(todo=="all" and sort(go) or {todo}) do
143     for k,v in pairs(b4) do the[k]=v end
144     math.randomseed(the.seed)
145     if GO[x] then print(x); GO[x]() end end
146   GO.rogue()
147   os.exit(fails) end
148
149 function GO.rogue( t )
150   t={}; for _,k in pairs{ "G", "_VERSION", "arg", "assert", "collectgarbage",
151     "coroutine", "debug", "dofile", "error", "getmetatable", "io", "ipairs",
152     "load", "loadfile", "math", "next", "os", "package", "pairs", "pcall",
153     "print", "rawequal", "rawget", "rawlen", "rawset", "require", "select",
154     "setmetatable", "string", "table", "tonumber", "tostring", "type", "utf8",
155     "warn", "xpcall" } do t[k]=true end
156   for k,v in pairs(_ENV) do if not t[k] then print("?",k, type(v)) end end end
157
158 if the.help then print(help) else GO(the.go) end

```