```
local the,help={},[[

lua l5.lua [OPTIONS]
L5 == a very little LUA learning lab
(c)2022, Tim Menzies, BSD 2-clause license

OPTIONS:
  -dump  -d     on error, exit after stacktrace = false
  -far   -F  F  look no further than "far"      = .9
  -seed  -S  P  random number seed              = 10019
  -file  -f  S  where to get data               = ../etc/data/auto93.csv
  -help  -h     show help                       = false
  -p     -p  P  distance calcs coefficient      = 2
  -some  -s     look only at "some" items       = 512
  -todo  -t  S  start-up action                 = nothing

KEY: S=string, P=poisint, F=float
]]
local b4={}; for k,_ in pairs(_ENV) do b4[k]=k end
-------------------------------------------------------------------------
--[[
Conventions
- "i" not "self"
- if something holds soehtimg, make the hodler cald "all"
-]]
-------------------------------------------------------------------------
local push,fmt
fmt=string.format
function push(t,x) table.insert(t,x); return x end

local thing,things,file2things
function thing(x)
  x = x:match"^%s*(.-)%s*$"
  if x=="true" then return true elseif x=="false" then return false end
  return tonumber(x) or x end

function things(x,sep,  t)
  t={}; for y in x:gmatch(sep or"([^,]+)") do push(t,thing(y)) end
  return t end

function file2things(file,      x)
  file = io.input(file)
  return function()
    x=io.read();
    if x then return things(x) else io.close(file) end end end

local last,per,any,many
function last(a)       return a[ #a ] end
function per(a,p)      return a[ (p*#a)//1 ] end
function any(a)        return a[ math.random(#a) ] end
function many(a,n,  u) u={}; for j=1,n do push(u,any(a)) end; return u end

local firsts,sort,map,slots
function firsts(a,b)   return a[1] < b[1] end
function sort(t,f)     table.sort(t,f); return t end
function map(t,f, u)   u={};for k,v in pairs(t) do push(u,f(v)) end; return u en
d
function slots(t, u,s)
  u={}
  for k,v in pairs(t) do s=tostring(k);if s:sub(1,1)~="_" then push(u,k) end end
  return sort(u) end

local oo,o
function oo(t) print(o(t)) end
function o(t,seen,         key,xseen,u)
  seen = seen or {}
  if type(t)~="table" then return tostring(t) end
  if seen[t]          then return "..." end
  seen[t] = t
  key   = function(k) return fmt(":%s %s",k,o(t[k],seen)) end
  xseen = function(x) return o(x,seen) end
  u = #t>0 and map(t,xseen) or map(slots(t),key)
  return (t.is or "")..'{'..table.concat(u," ")..'}' end

local Demo, ok = {fails=0}
function ok(test,msg)
  print(test and "PASS: "or "FAIL: ",msg or "")
  if not test then
    Demo.fails=Demo.fails+1
    if the.dump then assert(test,msg) end end end

function Demo.main(todo,seed)
   for k,one in pairs(todo=="all" and slots(Demo) or {todo}) do
     if k ~= "main" and type(Demo[one]) == "function" then
       math.randomseed(seed)
       Demo[one]() end end
   for k,v in pairs(_ENV) do if not b4[k] then print("?",k,type(v)) end end
   return Demo.fails end

local function settings(txt,  d)
  d={}
  txt:gsub("\n [-]([^%s]+))[%s]+[-[^%s]+)[^\n]*%s([^%s]+)",
    function(long,key,short,x)
      for n,flag in ipairs(arg) do
        if flag==short or flag==long then
          x = x=="false" and true or x=="true" and "false" or arg[n+1] end end
      if x=="false" then the[key]=false elseif x=="true" then the[key]=true else
      d[key] = tonumber(x) or x end end)
  if d.help then print(help) end
  return d end

-------------------------------------------------------------------------
local Sym,Num,nump add
function Sym(at,s) return {is="Sym", at=at, name=s, n=0,all={},most=0} end

function Num(at,s) return {is="Num", at=at, name=s, n=0,mu=0,m2=0,sd=0,
                           hi=-1E31, lo=1E31, w=s:find"-$" and -1 or 1} end

function nump(col) return col.w end

function add(i,x,inc,      sym,num)
  inc=inc or 1
  function sym()
    i.all[x] = inc + (i.all[x] or 0)
    if i.all[x] > i.most then i.most, i.model = i.all[x], x end
  end ----------
  function num1()
    for j=1,inc do
      d     = x - i.mu
      i.mu  = i.mu + d/i.n
      i._m2 = i.m2 + d*(x - i.mu)
      i.sd  = (i.m2<0 or i.n<2) and 0 or ((i.m2/(i.n-1))^0.5)
      i.lo  = math.min(x, i.lo)
      i.hi  = math.max(x, i.hi)  end
  end ----------
 if x ~= "?" then
    i.n = i.n+1
    (nump(x) and num1 or sym1)() end
  return x end

-------------------------------------------------------------------------
function Egs(names,   i,col)
  i = {is="egs",all={},names=names,all={},x={},y={}}
  for at,name in pairs(names) do
    col = push(i.all, (name:find"^[A-Z]" and Num or Sym)(at,name))
    if not name:find":$" then
      push(name:find"[-+]$" and i.y or i.x, col) end end
  return i end

function data(i,row)
  push(i.all,row)
  for _,c in pairs(i.all) do add(c, row[c.at]) end
  return i end

function file2Egs(file, egs)
  for row in file2things(file) do
    if egs then data(egs,row) else egs=Egs(row) end end
  return egs end

-------------------------------------------------------------------------
local dist,far,furthest,neighbors
function dist(i,row1,row2,    d,n,norm,dist1,lo,hi)
  function norm(x,lo,hi,    y)
    return  ((hi-lo)<1E-9) and 0 or (x-lo)/(hi-lo)
  end --------------------
  function dist1(col,a,b)
    if a=="?" and b=="?" then return 1 end
    if not nump(col) then return a==b and 0 or 1 end
    lo,hi=col.lo, col.hi
    if     a=="?" then b=norm(b,lo,hi); a=b<.5 and 1 or 0
    elseif b=="?" then a=norm(a,lo,hi); b=a<.5 and 1 or 0
    else   a,b = norm(a,lo,hi), norm(b,lo,hi)  end
    return math.abs(a - b)
  end ----------------------
  d,n = 0,0
  for _,col in pairs(i.x) do
    d = d + dist1(col, row1[col.at], row2[col.at])^the.p
    n = n + 1 end
  return (d/n)^(1/the.p) end

function far(      i,r1,rows,far)
  return per(neighbors(i,r1,rows),far or the.far)[2] end

function furthest( i,r1,rows)
  return last(neighbors(i,r1,rows))[2] end

function neighbors(i,r1,rows)
  return sort(map(rows, function(r2) return {dist(i,r1,r2),r2} end), firsts) end

local half
function half(i, rows,    project,row,some,east,west,easts,wests,c,mid)
  function project(row,a,b)
    a= dist(i,east,row)
    b= dist(i,west,row)
    return {(a^2 + c^2 - b^2)/(2*c), row}
  end ----------------------
  some = many(rows, the.some)
  east = furthest(i,any(some), some)
  west = furthest(i,east,         some)
  c    = dist(i,east,west)
  easts,wests = {},{}
  for n, xrow in pairs(sort(map(rows,project),firsts)) do
    row = xrow[2]
    if n==#rows//2 then mid=row end
    push(n <= #rows//2 and easts or wests, row) end
  return easts, wests, east, west, mid  end

local mid,div
```

```lua
-------------------------------------------------------------------------------
function Demo.the() oo(the) end

function Demo.many(a)
  a={1,2,3,4,5,6,7,8,9,10}; ok("{10 2 3}" == o(many(a,3)), "manys") end

function Demo.egs()
  ok(5140==file2Egs(the.file).y[1].hi,"reading") end

function Demo.dist(i)
  i = file2Egs(the.file)
  for n,row in pairs(i.all) do print(n,dist(i, i.all[1], row)) end end

function Demo.far(  i,j,row1,row2,row3,d3,d9)
  i = file2Egs(the.file)
  for j=1,10 do
    row1 = any(i.all)
    row2 = far(i,row1, i.all, .9)
    d9   = dist(i,row1,row2)
    row3 = far(i,row1, i.all, .3)
    d3   = dist(i,row1,row3)
    ok(d3 < d9, "closer far") end end

function Demo.half(  i,east,west)
  i = file2Egs(the.file)
  east,west = half(i, i.all)
  print(#east,#west) end

the=settings(help)
Demo.main(the.todo, the.seed)
```