```lua
local b4={}; for k,_ in pairs(_ENV) do b4[k]=k end
local all,any,firsts,new,many,map,o,push
local rows,seconds,slots,sort,thing,things
local EGS, NUM, SYM = {},{},{}
-- ------------------------------------------------------------------------
function NUM.new(i,at,s)
  return new(i,{at=at,txt=s,w=s:find"-" and -1 or 1,all={},
                ok=false, lo=math.huge, hi=-math.huge}) end

function NUM.add(i,x)
  if x ~= "?" then
    i.ok = false
    i.all[1 + #i.all] = x
    if x < i.lo then i.lo = x end
    if x > i.hi then i.hi = x end end end
  return x end

function NUM.dist(i,a,b)
  if     a=="?" and  b=="?" then a,b=1,0
  elseif a=="?" then b=norm(num,b); a=b>.5 and 0 or 1
  elseif b=="?" then a=norm(num,a); b=a>.5 and 0 or 1
  else               a, b=norm(num,a), norm(num,b) end
  return math.abs(a-b) end

function NUM.norm(i,x)
  return i.hi - i.lo<1E-9 and 0 or (x - i.lo)/(i.hi - i.lo) end
-- ------------------------------------------------------------------------
function SYM.new(i,at,s)
  return new(i,{at=at,txt=s,_all={}}) end

function SYM.add(i,x)
  if x ~= "?" then i._all[x] = 1+(i._all[x] or 0) end
  return x end

function SYM.all(i)
  if not i.ok then sort(i._all); i.ok=true end; return i._all end

function SYM.dist(i,a,b)
  return  a=="?" and b=="?" and 1 or a==b and 0 or 1 end
-- ------------------------------------------------------------------------
function EGS.new(i)
  return new(i,{rows={}, head=nil, all={}, x={},  y={}}) end

function EGS.add(i,t)
  local add,now = function(col) return col:add(t[col.at]) end
  if   i.head
  then i.rows[1+#i.rows] = map(i.all,add)
  else i.head=t
       for n,x in pairs(t) do
         now = push(i.all, (x:find"^[A-Z]" and NUM or SYM):new(n,x))
         if not x:find":" then
            push((x:find"+" or x:find"-") and i.y or i.x,now) end end end end

function EGS.clone(i,inits,    j)
  j = EGS:new()
  j:add(i.head)
  for _,row in pairs(inits or {}) do j = egs1(j, row) end
  return j end

function EGS.dist(i,r1,r2,    d,n,norm)
  d,n = 0, (#i.x)+1E-31
  for _,col in pairs(i.x) do
    inc = col:dist(r1[col.at], r2[col.at])
    d   = d + inc^2 end
  return (d/n)^.5 end

function EGS.far(i,r1,rows,         fun,tmp)
  fun = function(r2) return {r2, i:dist(r1,r2)} end
  tmp = sort(map(rows,fun), seconds)
  print(1)
  print(o(tmp))
  return table.unpack(tmp[#tmp*.9//1] ) end

function EGS.half(i,rows)
  local some,nth,sth,c,cosine,ls,rs
  rows  = rows or i.rows
  some  = #rows > 512 and many(rows,512) or rows
  nth   = i:far(any(rows), some)
  sth,c = i:far(nth,       some)
  function cosine(r,     a,b)
    a,b = i:dist(r,nth),i:dist(r,sth);return {(a^2+c^2-b^2)/(2*c),r} end
  ls,rs = i:clone(), i:clone()
  for n,pair in pairs(sort(map(rows,cosine), firsts)) do
    egs1(n <= #rows//2 and ls or rs, pair[2]) end
  return ls,rs,l,r,c end

-- ------------------------------------------------------------------------
function any(t)       return t[math.random(#t)] end

function firsts(a,b)  return a[1] < b[1] end

function many(t,n, u) u={};for j=1,n do t[1+#t]=any(t) end; return u end

function map(t,f,u)   u={};for _,v in pairs(t) do u[1+#u]=f(v) end; return u end

function new(k,t) k.__index=k; return setmetatable(t,k) end

function o(t,    u)
  if type(t)~="table" then return tostring(t) end
  local key=function(k) return string.format(":%s %s",k,t[k]) end
  u = #t>0 and  map(t,o) or map(sort(slots(t)),key)
  return '{'..table.concat(u," ")..'}' end

function push(t,x) table.insert(t,x); return x end

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

function slots(t, u) u={};for k,_ in pairs(t) do u[1+#u]=k    end; return u end

function sort(t,f)    table.sort(t,f); return t end

function seconds(a,b) return a[2] < b[2] end

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 t[1+#t]=thing(y) end; return t end
-- ------------------------------------------------------------------------
--for row in rows("../../data/auto93.csv") do print(o(row)) end
local i=EGS:new()
i:half()
for row in rows("../../data/auto93.csv") do i:add(row)  end
for k,v in pairs(_ENV) do if not b4[k] then print("?",k,type(v)) end end
```