```lua
local help= [[
NB:
(c)2022 Tim Menzies, timm@ieee.org

OPTIONS:
  -k  --k  handle rare classes    = 1
  -m  --m  handle rare attributes = 2
  -p  --p  distance coefficient   = 2

OPTIONS (other):
  -h  --help  show help      = false
  -g  --go    start-up goal = nothing
  -s  --seed  seed          = 10019
  -f  --file  file          = ../../data/auto93.csv]]
----------------------------------------------------------------------
--
--       ._    _.  ._ _    _    _
--       | |  (_|  | | |  (/_  _>
--
local lib = require"lib"
local cli,csv,demos,is,normpdf = lib.cli, lib.csv, lib.demos, lib.is, lib.normpdf
local oo,read,str      = lib.oo, lib.read, lib.str

local THE={}
help:gsub(" [-][-]([^%s]+)[^\n]*%s([^%s]+)",function(key,x) THE[key] = read(x) end)


local NUM,SYM,COLS,ROWS = is"NUM", is"SYM", is"COLS", is"ROWS"
--------------------------------------------------------------------------------
--        _   _   |        ._ _    ._
--       (_  (_)  |  |_|  | | |  | |
--
local function add(i, x)
  for _,v in pairs(type(x)=="table" and x or {x}) do
    if v ~="?" then
      i.n = i.n + 1
      i:add(v) end end
  return x end

function NUM.new(i)        i.n,i.mu,i.m2,i.mu = 0,0,0,0 end
function NUM.mid(i,p)      return rnd(i.mu,p) end
function NUM.like(i,x,...) return normpdf(x, i.mu, i.sd) end
function NUM.add(i,v)
  d    = v - i.mu
  i.mu = i.mu + d/i.n
  i.m2 = i.m2 + d*(v - i.mu)
  i.sd = i.n<2 and 0 or (i.m2/(i.n-1))^0.5 end

function SYM.new(i)          i.n,i.syms,i.most,i.mode = 0,{},0,nil end
function SYM.mid(i,...)      return i.mode end
function SYM.like(i,x,prior) return ((i.syms[x] or 0)+THE.m*prior)/(i.n+THE.m) end
function SYM.add(i,v)
  i.syms[v] = (inc or 1) + (i.syms[v] or 0)
  if i.syms[x] > i.most then i.most,i.mode = i.syms[v],v end end
--------------------------------------------------------------------------------
--        _   _   |        ._ _    ._    _
--       (_  (_)  |  |_|  | | |  | |  _>
--
local function usep(x)   return not x:find":$" end
local function nump(x)   return x:find"^[A-Z]" end
local function goalp(x)  return x:find"[!+-]$" end
local function klassp(x) return x:find"!$"     end

local function new(at,txt)
  txt = txt or ""
  local i = (nump(txt) and NUM or SYM)()
  i.txt, i.usep, i.at, i.w = txt, usep(txt), at or 0, txt:find"-$" and -1 or 1
  return i  end

function COLS.new(i,t)
  i.all, i.xs, i.ys, i.names = {},{},{},t
  for at,x in pairs(t) do
    col = push(i.all, new(at,x))
    if col.usep  then
      if klassp(col.txt) then i.klass=col end
      push(goalp(col.txt) and i.ys or i.xs, col) end end end

 function COLS.add(i,t)
   for _,cols in pairs{i.xs,i.ys} do
     for _,col in pairs(cols) do col:add(t[col.at]) end end
   return t end
--------------------------------------------------------------------------------
--       ._   _          _
--       |   (_)  \/\/  _>
--
local function load(src, fun)
  if type(src)~="string" then for _,t in pairs(src) do fun(t) end
                         else for   t in csv(src)   do fun(t) end end end

function ROWS.new(i,t) i.cols=COLS(t); i.rows={} end
function ROWS.add(i,t) push(i.rows, i.cols:add(t)) end
function ROWS.mid(i, p)
  t={}; for k,v in pairs(i.cols.ys) do t[k]=col:mid(p) end; return t end

function ROWS.clone(i,t,  j)
  j= ROWS({i.cols.names});for _,row in pairs(t) do j:add(row) end; return j end

function ROWS.like(i,t, nklasses, nrows,    prior,like,inc,has)
  prior = (i.n + THE.k) / (nrows + THE.k * nklasses)
  like  = math.log(prior)
  for _,col in pairs(i.cols.xs) do
    x = t[col.at]
    if x and x ~= "?" then
      like = like + math.log(col:like(x,prior)) end end
  return like end
--------------------------------------------------------------------------------
--      _|_   _    _  _|_   _
--       |_  (/_  _>   |_  _>
--
local no,go = {},{}
function go.the() oo(THE); return true end
function go.csv() print(THE.file); for row in csv(THE.file)  do oo(row) end; return tr
ue end

--------------------------------------------------------------------------------
--        _  _|_   _.  ._  _|_
--       _>   |_  (_|  |    |_
--
if    pcall(debug.getlocal, 4, 1)
then  return {ROW=ROW, ROWS=ROWS, NUM=NUM, SYM=SYM, THE=THE,lib=lib}
else  THE = cli(THE,help)
      oo(THE)
      demos(THE,go) end
```