```lua
local R = require
local _, the, ABCD  = R"lib", R"the", R"ABCD"
local NUM, SYM, BIN,EGS,COLS = R"num", R"sym", R"bin", R"egs", R"cols"
--local num, sym                    = R"num", R"sym"
--local ako, egs, seen, cluster      = R"ako", R"egs", R"seen", R"cluster"
--local learn101, learn201, learn301 = R"learn101", R"learn201", R"learn301"
local per,map,dent = _.per, _.map, _.dent

local ish,copy,items,o,oo,powerset = _.ish,_.copy,_.items,_.o,_.oo,_.powerset
local map,fmt,rnds, rnd,push       = _.map,_.fmt,_.rnds, _.rnd,_.push
local class,Obj = _.class, _.Obj
local go,ok = _.go,_.ok

function go.class()
  local EMP=class("EMP",Obj)
  function EMP:show() return {"name", "age", "_id"} end
  function EMP:new(name) self._id=1;  self.name=name; self.age=0 end
  local fred = EMP("tim")
  local MANAGER=class("MANAGER",EMP)
  local jane = MANAGER("janc")
  print(jane) end

function go.copy(     t,u)
  t={a={b={c=10},d={e=200}}, f=300}
  u= copy(t)
  t.a.b.c= 20
  ok(u.a.b.c ~= 20,"copy") end

function go.rnd()
  ok("23.11" == rnds({23.11111})[1],"rounds") end

function go.collect()
  local function aux(x,y) return x*y end
  oo(_.collect({10,20,30},aux)) end

function go.items()
  for  x in items{10,20,30} do oo(x) end
  local n=0
  for x in items(the.file) do n=n+1; if n<=5 then oo(x) end end end

function go.powerset()
  for _,x in pairs(powerset{10,20,30,40,50}) do oo(x) end end

 function go.many( t)
  local o,many=lib.o,lib.many
  t={};for j = 1,1000 do t[#t+1] = j end
  print(900,"+", o(many(t, 10, 900)))
  print(1,100,    o(many(t, 10,    1, 100)))
  print(300,700, o(many(t, 10, 300, 700))) end

function go.some( n)
  the.some=512
  n=NUM()
  for i=1,999 do n:add( i//100) end
  for k,v in pairs(SYM():adds(n:all()).has) do print(k,v) end end

function go.ent()
  local n = SYM()
  n:add("a",9)
  n:add("b",7)
  ok(ish(n:div(), .98886), "entropy")  end

function go.normal( n)
  n=NUM()
  for i=1,10^3 do n:add( _.normal(10,2) //1) end
  for n,k in pairs(SYM():adds(n:all()).has) do print(n,k) end end

function go.nums( n)
  n=NUM()
  for i=1,10^2 do n:add(_.normal(8,1)) end
  oo(rnds{n:mid(), n:div()}) end

function go.bins(   n1,n2)
  n1,n2 = NUM(),NUM()
  for i=1,100 do n1:add(_.normal(-4,1)) end
  for i=1,100 do n2:add(_.normal( 0,1)) end
  for i=1,100 do n1:add(_.normal( 4,1)) end
  map(n1:bins(n2, BIN),
      function(b)
        print(b.ys.n, rnd(b.lo), rnd(b.hi), o(b.ys.has)) end) end

function go.cols()
  _.dent(COLS{"Name","Age:","gender","Weight-"}) end

function go.egs(  i)
  i= EGS():adds(the.file)
  ok(7==i.cols.x[2].has["lt40"], "counts")
  ok(286 == #i.rows,"egs") end

function go.mid(  i)
  i= EGS():adds("../etc/data/auto93.csv")
  j,k=i:bestRest()
  j=i:clone(j)
  k=i:clone(k)

  oo(i.mid())
  oo(j:mid())
  oo(k:mid())

  end

function go.bestRest(  i)
  i= EGS():adds("../etc/data/auto93.csv") end

local function _dist(file,  i,all)
  local any= _.any
  i= EGS():adds(file)
  local yes=true
  all=NUM()
  for j=1,1000 do
    if (j % 50)==0 then io.write(".") end
    local a,b,c = any(i.rows), any(i.rows), any(i.rows)
    local aa = i:dist(a,a)
    local ba = i:dist(b,a)
    local ab = i:dist(a,b)
    local bc = i:dist(b,c)
    local ac = i:dist(a,c)
    all:adds{aa,ba,ab,bc,ac}
    yes = yes and aa==0 and ab == ba and ab+bc >= ac
    yes = yes and aa>=0 and aa<=1 and ba>=0 and ba<=1 and ab>=0 and ab<=1 and
                  bc>=0 and bc <=1 and ac >= 0 and ac <= 1 end
  oo(rnds(all:all()))
  ok(yes, "dist") end

function go.dist1() _dist(the.file) end
function go.dist2() _dist("../etc/data/diabetes.csv") end

function go.half(  i)
  the.file = "../etc/data/diabetes.csv"
  i = egs.Init(the.file)
  local lefts,rights,left,right,border,c= cluster.half(i)
  print("rows",#i.rows)
  ok(384 == #lefts.rows,   "left")
  ok(384 == #rights.rows, "rights") end

function go.cluster(  i)
  the.file = "../etc/data/diabetes.csv"
  i = egs.Init(the.file)
  cluster.show(cluster.new(i)) end

function go.abcd()
  local t={}
  for _ = 1,6 do push(t,{want="yes",got="yes"}) end
  for _ = 1,2 do push(t,{want="no",got="no"}) end
  for _ = 1,6 do push(t,{want="maybe",got="maybe"}) end
  for _ = 1,1 do push(t,{want="maybe", got="no"}) end
  ABCD():adds(t,true) end

local function qq(i,q)
  print(q[1], fmt("%15s = %-8s best= %s/%s rest= %s/%s",
                  i.cols[q[2]].name, q[3],q[4],q[5],q[6],q[7])) end

local function gonb1(file)
  local i = require"learn101"(file)
  local _, out = i:score()
  local cnt={}
  for _,one in pairs(out) do local k=one.got.."."..one.want; cnt[k] = 1+ (cnt[k] or 0) end
  for k,n in pairs(cnt) do print(n,o(k)) end
  ABCD():adds(i.log,true) end

function go.nb1a() gonb1(the.file) end
function go.nb1b() gonb1("../etc/data/diabetes.csv") end

function go.nb2()
  the.file = "../etc/data/diabetes.csv"
  the.goal = "positive"
  local i = require("learn201")(the.file);
  ABCD():adds(i.log,true) end

function go.nb2a()
  the.file = "../etc/data/diabetes.csv"
  the.goal = "positive"
  for _,bins in pairs{2,5,9} do
    the.bins = bins
    local i = nb2(the.file);
    abcd(i.log,true) end end

function go.nb3()
  the.file = "../etc/data/diabetes.csv"
  the.goal = "positive"
  the.bins = 16
  local i = nb3(the.file);
  abcd(i.log,true)
  local acc, out = score(i);  map(out,function(q) qq(i,q) end) end

return go
```