


```

256 --      -|_  (7_ > -|_ > -----
257 --
258 local no,go = {},{}
259 function go.the() print(1); print(1); return type(1); end
260 function go.argmax(t,fun)
261   fun=function(x) return -x end
262   t={50,40,0,40,50}
263   return 3 == argmax(t,fun) end
264
265 function go.num(n) n=NUM(); for x=1,100 do n:add(x) end; return n.mu==50.5 end
266
267 function go.sym(s)
268   s=SYM(); for _,x in pairs{"a","a","a","a","b","b","c"} do s:add(x) end
269   return s.mode=="a" end
270
271 function go.csv(n,s)
272   n,s=0,0; for row in csv(1) do n=n+1; if n>1 then s=s+row[1] end end
273   return rnd(s/n,3) == 5.441 end
274
275 function go.rows(rows)
276   data(1,1,rows,function(t) if rows then rows:add(t) else rows=ROWS(t) end end)
277   return rnd(rows.cols.ys[1].sd,0)==847 end
278
279 function go.nb()
280   return 268 == #NB("./data/diabetes.csv").dict["positive"].rows end
281
282 local function _classify(file)
283   local abcd=require"abcd"
284   local abcd=Abcd()
285   NB(file, function(got,want) abcd:add(got,want) end)
286   abcd:pretty(abcd:report())
287   return true end
288
289 function go.soybean() return _classify("./data/soybean.csv") end
290 function go.diabetes() return _classify("./data/diabetes.csv") end
291
292 --      -|_  (7_ > -|_ > -----
293 --
294 if pcall(debug.getlocal, 4, 1)
295 then return (ROW=ROW, ROWS=ROWS, NUM=NUM, SYM=SYM, THE=THE,lib=lib)
296 else THE = cli(THE,help)
297   demos(THE,go) end
298
299 ---
300 ---
301 ---
302 ---
303 ---
304 ---
305 ---
306 ---
307 ---
308 ---
309 ---

```