

Saturday August 27, 2022

 $\frac{1}{2}$

```

211 -- -----
212 -- ## Test Engine
213 local eg, fails = {},0
214
215
216 -- [1] reset random number seed before running something.
217 -- [2] Cache the defaults settings, and [3] restore them after the test
218 -- [4] Print error messages or stack dumps as required.
219 -- Return true if this all went well.
220 local function runs(k, old, status, out, msg)
221     if not eg[k] then return end
222     math.randomseed(the.seed) -- reset seed [1]
223     old={}; for k,v in pairs(the) do old[k]=v end -- [2]
224     if the.dump then
225         status,out = true, eg[k]()
226     else
227         status,out = pcall(eg[k]) -- pcall means we do not crash and dump on error
228     end
229     for k,v in pairs(old) do the[k]=v end -- restore old settings [3]
230     msg = status and ((out==true and "PASS") or "FAIL") or "CRASH" -- [4]
231     print("!!!!!!", msg, k, status)
232     return out or err end
233
234 -- -----
235 -- ## Tests
236 -- Test that the test happens when something crashes?
237 function eg.BAD() print(eg.dont.have.this.field) end
238
239 -- Sort all test names.
240 function eg.LIST( t )
241     t={}; for k,_ in pairs(eg) do t[1+#t]=k end; table.sort(t); return t end
242
243 -- List test names.
244 function eg.LS()
245     print("\nExamples lua csv -e...")
246     for _,k in pairs(eg.LIST()) do print(string.format("%s",k)) end
247     return true end
248
249 -- Run all tests
250 function eg.ALL()
251     for _,k in pairs(eg.LIST()) do
252         if k ~= "ALL" then
253             print("\n-----")
254             if not runs(k) then fails=fails+ 1 end end end
255     return true end
256
257 -- Settings come from big string top of "sam.lua"
258 -- (maybe updated from comamnd line)
259 function eg.the() oo(the); return true end
260
261 -- The middle and diversity of a set of symbols is called "mode"
262 -- and "entropy" (and the latter is zero when all the symbols
263 -- are the same).
264 function eg.sym( sym,entropy,mode)
265     sym= adds(Sym(), {"a","a","a","a","b","b","c"})
266     mode, entropy = mid(sym), div(sym)
267     entropy = (1000*entropy)//1/1000
268     oo({mid=mode, div=entropy})
269     return mode=="a" and 1.37 <= entropy and entropy <=1.38 end
270
271 -- The middle and diversity of a set of numbers is called "median"
272 -- and "standard deviation" (and the latter is zero when all the nums
273 -- are the same).
274 function eg.num( num)
275     num=Num()
276     for i=1,100 do add(num,i) end
277     local med,ent = mid(num), div(num)
278     print(mid(num),div(num))
279     return 50<= med and med<= 52 and 30.5 <ent and ent <32 end
280
281 -- Nums store only a sample of the numbers added to it (and that storage
282 -- is done such that the kept numbers span the range of inputs).
283 function eg.bignum( num)
284     num=Num()
285     the.nums = 32
286     for i=1,1000 do add(num,i) end
287     oo(nums(num))
288     return 32==#num._has; end
289
290 -- Show we can read csv files.
291 function eg.csv()
292     local n=0
293     csv("../data/aut093.csv",function(row)
294         n=n+1; if n> 10 then return else oo(row) end end); return true end
295
296 -- Print some stats on columns.
297 function eg.stats()
298     oo(stats(records("../data/aut093.csv"))); return true end
299
300 -- -----
301 the = cli(the)
302 runs(the.eg)
303 rogues()
304 os.exit(fails)

```