

```

1 local R = require
2 local _, the, ABCD = R"lib", R"the", R"ABCD"
3 local NUM, SYM, BIN, EGS, COLS, RULE = R"num", R"sym", R"bin", R"egs", R"cols", R"rule"
4 local per, map, dent = _, per, _, map, _, dent
5
6 local ish, copy, items, o, oo, powerset = _, ish, _, copy, _, items, _, o, _, oo, _, powerset
7 local map, fmt, rnds, _, rnd, push = _, map, _, fmt, _, rnds, _, rnd, _, push
8 local class, Obj = _, class, _, Obj
9 local no, go, ok = _, no, _, go, _, ok
10
11 function go.class()
12   local EMP=class("EMP", Obj)
13   function EMP:show() return {"name", "age", "id"} end
14   function EMP:new(name) self.id=1; self.name=name; self.age=0 end
15   local fred = EMP("tim")
16   local MANAGER=class("MANAGER", EMP)
17   local jane = MANAGER("jane")
18   print(jane) end
19
20 function go.copy( t,u)
21   t={a={b={c=10},d={e=200}}, f=300}
22   u= copy(t)
23   t.a.b.c= 20
24   ok(u.a.b.c ~= 20, "copy") end
25
26 function go.rnd()
27   ok("23.11" == rnds({(23.11111)})[1], "rounds") end
28
29 function go.collect()
30   local function aux(x,y) return x*y end
31   oo( _.collect({(10,20,30),aux}) end
32
33 function go.items()
34   for x in items(10,20,30) do oo(x) end
35   local n=0
36   for x in items(the.file) do n=n+1; if n<=5 then oo(x) end end end
37
38 function go.powerset()
39   for _,x in pairs(powerset(10,20,30,40,50)) do oo(x) end end
40
41 function go.many( t)
42   local o,many=_,o,_.many
43   t={};for j = 1,1000 do t[#t+1] = j end
44   print(900,"+", o(many(t, 10, 900)))
45   print(1,100, o(many(t, 10, 1, 100)))
46   print(300,700, o(many(t, 10, 300, 700))) end
47
48 function go.some( n)
49   the.some=512
50   n=NUM()
51   for i=1,999 do n:add( i//100) end
52   for k,v in pairs(SYM()):adds(n:all()) .has do print(k,v) end end
53
54 function go.ent()
55   local n = SYM()
56   n:add("a",9)
57   n:add("b",7)
58   ok(ish(n:div(), .98886), "entropy") end
59
60 function go.normal( n)
61   n=NUM()
62   for i=1,10^3 do n:add( _.normal(i,0,2) //1) end
63   for n,k in pairs(SYM()):adds(n:all()) .has do print(n,k) end end
64
65 function go.nums( n)
66   n=NUM()
67   for i=1,10^2 do n:add( _.normal(8,1)) end
68   oo(rnds(n:mid(), n:div())) end
69
70 function go.cols()
71   _,dent(COLS["Name", "Age", "gender", "Weight="]) end
72
73 function go.egs( i)
74   i= EGS():adds(the.file)
75   ok(7 == i.cols.x[2].has["i40"], "counts")
76   ok(286 == #i.rows, "egs") end
77
78 function go.clone( i,j)
79   i= EGS():adds("._etc/data/auto93.csv")
80   j= i:clone(i.rows)
81   local flag = true
82   for k,n in pairs(i.cols.y[1]:all()) do
83     flag=flag and n==j.cols.y[1]:all()[k] end
84   ok(flag, "clone") end
85
86 function go.mid( all,best,rest)
87   all = EGS():adds("._etc/data/auto93.csv")
88   best,rest = all:bestRest()
89   best = all:clone(best)
90   rest = all:clone(rest)
91   print("all",o(all:mid()))
92   print("best",o(best:mid()))
93   print("rest",o(rest:mid())) end
94
95 function go.bins( all,best,rest,b4)
96   all = EGS():adds("._etc/data/auto93.csv")
97   best,rest = all:bestRest()
98   best = all:clone(best)
99   rest = all:clone(rest)
100   for _,bin in pairs(best:bins(rest)) do
101     if bin.at ~= b4 then print("") end
102     print(bin.name, bin.at, bin.lo, bin.hi,
103           bin.ys.has["left"] or 0,
104           bin.ys.has["right"] or 0)
105     b4 = bin.at end end
106
107 local function _rules(file, all,bests,rests,left,right,b4,bins, rules,h)
108   all = EGS():adds(file)
109   print(o(rnds(all:mid()))), o(rnds(map(all:div(), function(x) return x*the.cohen end
110   )), "<= before\n")
111   bests,rests = all:bestRest()
112   left = all:clone(bests)
113   right = all:clone(rests)
114   h = (left=#bests, right=#rests)
115   rules = RULE.fromBins(left:bins(right), all,h,bests,rests)
116   end
117
117 function go.rules1() _rules("._etc/data/auto93.csv") end
118 function go.rules2() _rules("._etc/data/china.csv") end
119 function go.rules3() _rules("._etc/data/nasa93dem.csv") end

```

```

120 function go.rules4() _rules("._etc/data/pom.csv") end
121 function go.rules5() _rules("._etc/data/coc10000.csv") end
122 function go.rules6() _rules("._etc/data/auto2.csv") end
123
124
125 local function _dist(file, i,all)
126   local any= _any
127   i= EGS():adds(file)
128   local yes=true
129   all=NUM()
130   for j=1,1000 do
131     if (j % 50)==0 then io.write(".") end
132     local a,b,c = any(i.rows), any(i.rows), any(i.rows)
133     local aa = i:dist(a,a)
134     local ba = i:dist(b,a)
135     local ab = i:dist(a,b)
136     local bc = i:dist(b,c)
137     local ac = i:dist(a,c)
138     all:adds(aa,ba,ab,bc,ac)
139     yes = yes and aa==0 and ab == ba and ab+bc >= ac
140     yes = yes and aa==0 and aa<=1 and ba==0 and ba<=1 and ab>=0 and ab<=1 and
141           bc>=0 and bc <=1 and ac >= 0 and ac <= 1 end
142     oo(rnds(all:all()))
143     ok(yes, "dist") end
144
145 function go.dist1() _dist(the.file) end
146 function go.dist2() _dist("._etc/data/diabetes.csv") end
147
148 function no.half( i)
149   the.file = "._etc/data/diabetes.csv"
150   i = egs.Init(the.file)
151   local lefts,rights,left,right,border,c= cluster.half(i)
152   print("rows", #i.rows)
153   ok(384 == #lefts.rows, "left")
154   ok(384 == #rights.rows, "right") end
155
156 function no.cluster( i)
157   the.file = "._etc/data/diabetes.csv"
158   i = egs.Init(the.file)
159   cluster.show(cluster.new(i)) end
160
161 function go.abcd()
162   local t={}
163   for _ = 1,6 do push(t,{want="yes",got="yes"}) end
164   for _ = 1,2 do push(t,{want="no",got="no"}) end
165   for _ = 1,6 do push(t,{want="maybe",got="maybe"}) end
166   for _ = 1,1 do push(t,{want="maybe", got="no"}) end
167   ABCD():adds(t,true) end
168
169 local function qq(i,q)
170   print(q[1], fmt ("%15s = %8s best= %s/%s rest= %s/%s",
171           i.cols[q[2]].name, q[3],q[4],q[5],q[6],q[7])) end
172
173 local function gonbl(file)
174   local i = require"learn101"(file)
175   local _, out = i:score()
176   local cnt={}
177   for _,one in pairs(out) do local k=one.got..".".one.want; cnt[k] = 1+ (cnt[k] o
178   r 0) end
179   for k,n in pairs(cnt) do print(n,o(k)) end
180   ABCD():adds(i.log,true) end
181
182 function go.nbla() gonbl(the.file) end
183 function go.nblb() gonbl("._etc/data/diabetes.csv") end
184
185 function go.nb2()
186   the.file = "._etc/data/diabetes.csv"
187   the.goal = "positive"
188   local i = require("learn201")(the.file);
189   ABCD():adds(i.log,true) end
190
191 function no.nb2a()
192   the.file = "._etc/data/diabetes.csv"
193   the.goal = "positive"
194   for _,bins in pairs(2,5,9) do
195     the.bins = bins
196     local i = require("learn201")(the.file);
197     ABCD():adds(i.log,true) end end
198
199 function no.nb3()
200   the.file = "._etc/data/diabetes.csv"
201   the.goal = "positive"
202   the.bins = 16
203   local i = nb3(the.file);
204   abcd(i.log,true)
205   local acc, out = score(i); map(out,function(q) qq(i,q) end) end
206
207 return go

```