

```

1 local _,the = require"lib", require"the"
2 local fmt, inc,slots = _:fmt, _:inc, _:slots
3 local class,OBJ      = _:class, _:OBJ
4
5 local ABCD = class("ABCD",OBJ)
6
7 function ABCD:new(data,rx)
8     self.data, self.rx = data or "" , rx or ""
9     self.yes, self.no = 0,0
10    self.known, self.a, self.b, self.c, self.d = {}, {}, {}, {} end
11
12 function ABCD:exists(x, new)
13     new = not self.known[x]
14     inc(self.known,x)
15     if new then
16         self.a[x]=self.yes + self.no; self.b[x]=0; self.c[x]=0; self.d[x]=0 end end
17
18 function ABCD:report( p,out,a,b,c,d,pd,pf,pn,f,acc,g,prec)
19     p = function (z) return math.floor(100*z + 0.5) end
20     out= {}
21     for x,xx in pairs( self.known ) do
22         pd,pf,pn,prec,g,f,acc = 0,0,0,0,0,0,0
23         a= (self.a[x] or 0); b= (self.b[x] or 0);
24         c= (self.c[x] or 0); d= (self.d[x] or 0);
25         if b+d > 0 then pd = d / (b+d) end
26         if a+c > 0 then pf = c / (a+c) end
27         if a+c > 0 then pn = (b+d) / (a+c) end
28         if c+d > 0 then prec = d / (c+d) end
29         if 1-pf+pd > 0 then g=2*(1-pf) * pd / (1-pf+pd) end
30         if prec+pd > 0 then f=2*prec*pd / (prec + pd) end
31         if self.yes + self.no > 0 then
32             acc= self.yes / (self.yes + self.no) end
33         out[x] = {data=self.data,rx=self.rx,num=self.yes+self.no,
34                 a=a,b=b,c=c,d=d,acc=p(acc),
35                 prec=p(prec), pd=p(pd), pf=p(pf),f=p(f), g=p(g), class=x} end
36     return out end
37
38 function ABCD:pretty(t)
39     print""
40     local s1 = "%10s| %10s| %4s| %4s| %4s| %4s "
41     local s2 = "%3s| %3s| %4s| %4s| %3s| %3s|"
42     local d,s = "-----", (s1 .. s2)
43     print(fmt(s,"db","rx","a","b","c","d","acc","pd","pf","prec","f","g"))
44     print(fmt(s,d,d,d,d,d,d,d,d,d,d,d,d))
45     for key,x in pairs(slots(t)) do
46         local u = t[x]
47         print(fmt(s.." %s", u.data,u.rx,u.a, u.b, u.c, u.d,
48                 u.acc, u.pd, u.pf, u.prec, u.f, u.g, x)) end end
49
50 function ABCD:adds(gotwants, show)
51     for key,one in pairs(gotwants) do
52         self:exists(one.want)
53         self:exists(one.got)
54         if one.want == one.got then self.yes=self.yes+1 else self.no=self.no+1 end
55         for x,xx in pairs(self.known) do
56             if one.want == x
57                 then inc(one.want == one.got and self.d or self.b, x)
58             else inc(one.got == x and self.c or self.a, x) end end end
59     return show and self:pretty(self:report()) or self:report() end
60
61 return ABCD

```