

```

1 local the={rations-256}
2
3 local _=require"etc"
4 local cat,cli,coerce,csv,klass = _cat, _cli, _coerce, _csv, _klass
5 local lines,push,rogues, words = _lines, _push, _rogues, _words
6
7 local Cols, Data, Num, Sym = obj"Cok", obj"Data", obj"Num", obj"Sym"
8
9 -----
10 function Num:new(at,txt)
11   txt = txt or ""
12   return {n=0,at=at or 0, txt=txt, ready=false, has={},
13          w=txt:find"-3" and -1 or 1} end
14
15 function NUM:add(x)
16   if x ~= "?" then
17     local pos
18     self.n = self.n + 1
19     if #self.has < the.ratios then pos = 1 + (#self.has)
20     elseif rand() < the.ratios/self.n then pos = rand(#self.has) end
21     if pos then
22       self.ready=false -- the 'kept' list is no longer in sorted order
23       self.has[pos]=x end end end
24
25 -----
26 function Sym:new(at,txt)
27   return {n=0,at=at or 0, txt=txt or "", ready=false, has={}} end
28
29 -----
30 function Cols:new(names)
31   self.names, self.x, self.y, self.all= names, {}, {}, {}
32   for at,txt in pairs(names) do
33     local what = txt:find"^[A-Z]" and Num or Sym
34     local col = push(self.all, what(at,txt))
35     if not txt:find"$" then
36       push(txt:find"!+-$" and self.y or self.x, col) end end end
37
38 -----
39 local function rows(src)
40   if type(src) == "table" then return src else
41     local u={}; csv(src, function(t) push(u,t) end); return u end end
42
43 function Data:new(rows)
44   for i,row in pairs(rows) do
45     if i==1
46       then i.cols = Cols(row)
47       else for cols in pairs(i.cols.x, i.cols.y) do
48         for col in pairs(cols) do col:add(row[col.at]) end end end end
49
50 -----
51 rogues()
52 return {Cols, Data, Num, Sym}

```