

```

1 local b4={}; for k,_ in pairs(_ENV) do b4[k]=k end
2 local big,copy,cli, csv, egs, fmt, is, lt, map
3 local o, oo, push, splice, sort, string2thing
4 local the = {
5     bins = 16,
6     cohen = .35,
7     file = "etc/data/auto93.csv",
8     how = "up",
9     min = .5
10 }
11
12 big = 1E32
13 fmt = string.format
14 lt = function(x) return function(a,b) return a[x] < b[x] end end
15 map = function(t,f, u) u={}; for k,v in pairs(t) do u[l+#u]=f(v) end; return u end
16 push=function(t,x); t[l+#t]=x; return x end
17 sort=function(t,f) table.sort(t,f); return t end
18
19 function cli(d)
20     for slot,x in pairs(d) do
21         x = tostring(x)
22         for n,flag in ipairs(arg) do
23             if flag=="-"..slot or flag=="-"..slot:sub(1,1) then
24                 x = x=="false" and true or x=="true" and false or arg[n+1] end end
25             d[slot] = string2thing(x) end
26         return d end
27
28 function copy(t, u)
29     if type(t) ~= "table" then return t end
30     u={}; for k,v in pairs(t) do u[copy(k)]=copy(v) end
31     return setmetatable(u, getmetatable(t)) end
32
33 function csv(csvfile)
34     csvfile = io.input(csvfile)
35     return function(line, row)
36         line=io.read()
37         if not line then io.close(csvfile) else
38             row={}; for x in line:gmatch("[^,]+") do push(row, string2thing(x)) end
39             return row end end
40
41 function oo(t) print(o(t)) end
42 function o(t, u)
43     if #t>0 then return {"..table.concat(map(t, tostring), " " ..)" } else
44         u={}; for k,v in pairs(t) do u[l+#u] = fmt("%.%s",k,v) end
45         return (t.is or "").."{"..table.concat(sort(u), " " ..)" } end end
46
47 function splice(t,i,j,k, u)
48     u={}; for n=(i or 1), (j or #t), (k or 1) do u[l+#u] = t[n] end return u end
49
50 function string2thing(x)
51     x = x:match("^%s*(~)%s*$")
52     if x=="true" then return true elseif x=="false" then return false end
53     return math.tointeger(x) or tonumber(x) or x end
54
55 function is(name, t, new)
56     function new(kl,...) local x=setmetatable({}, kl); kl.new(x,...); return x end
57     t = {__tostring=o, is=name or ""}; t.__index=t
58     return setmetatable(t, {__call=new}) end
59
60 local NUM, SYM, EGS = is"NUM", is"SYM", is"EGS"
61
62 function SYM.new(i, at, names)
63     i.n, i.txt, i.at, i.all = 0, txt or "", at or 0, {} end
64
65 function SYM.add(i, x)
66     if x=="?" then i.n = i.n+1; i.all[x] = 1+(i.all[x] or 0) end end
67
68 function NUM.new(i, at, txt)
69     i.n, i.mu, i.txt, i.at = 0, 0, txt or "", at or 0
70     i.w, i.lo, i.hi = i.txt:find"-$" and -1 or 1, big, -big end
71
72 function NUM.add(i, x, d)
73     if x=="?" then
74         i.n = i.n+1
75         d = i.mu - x
76         i.mu = i.mu+d/i.n
77         i.lo = math.min(x, i.lo)
78         i.hi = math.max(x, i.hi) end end
79
80 function NUM.norm(i, x)
81     return x=="?" and (i.hi-i.lo<1E-9 and 0 or (x-i.lo)/(i.hi-i.lo)) end
82
83 function EGS.new(i, names)
84     i.rows, i.names, i.all, i.y = {}, names, {}, {}
85     for at,txt in pairs(names) do
86         local col = push(i.all, (txt:find"^[A-Z]" and NUM or SYM) (at,txt))
87         if txt:find"[+-]$" then push(i.y, col) end end end
88
89 function EGS.add(i, row)
90     push(i.rows, row)
91     for _, col in pairs(i.all) do col:add(row[col.at]) end end
92
93 function EGS.order(i)
94     sort(i.rows, function(r1,r2)
95         local s1,s2,e,y,a,b = 0,0,math.exp(1),i.y
96         for _, col in pairs(y) do
97             a,b = col:norm(r1[col.at]), col:norm(r2[col.at])
98             s1 = s1 - e^(col.w * (a - b) / #y)
99             s2 = s2 - e^(col.w * (b - a) / #y) end
100         return s1/#y < s2/#y end)
101     return i end
102
103 function egs(f, i)
104     for row in csv(f or the.file) do
105         if i then i:add(row) else i=EGS(row) end end
106         return i:order() end
107
108 the = cli(the)
109 local x=egs(i)
110 for i=1,5 do oo(x.rows[i]) end; print""
111 for i=#x.rows-5, #x.rows do oo(x.rows[i]) end
112 for k,v in pairs(_ENV) do if not b4[k] then print("?", k, type(v)) end end

```