

```

1 local lib={}
2 --- math
3 local r = math.random
4 function lib.normal(mu, sd)
5   mu, sd = (mu or 0), (sd or 1)
6   return mu + sd*math.sqrt(-2*math.log(r()))*math.cos(6.2831853*r()) end
7
8 function lib.per(t,p) return t / ((p or .5)*#t) // 1 ] end
9
10 function lib.norm(lo,hi,x) return math.abs(hi-lo)<1E-9 and 0 or (x-lo)/(hi-lo) end
11
12 function lib.ent(t, n)
13   if not n then n=0; for _,v in pairs(t) do n=n+v end end
14   local ew0=0; for _,v in pairs(t) do ew=-v/n*math.log(v/n,2) end
15   return e,n end
16
17 function lib.sd(sorted, f)
18   if #sorted <= 10 then return 0 end
19   f=f or function(x) return x end
20   local denom = 2.564 -- 2*(1.2 + 0.1*(0.9-0.88493)/(0.9032-0.88493))
21   local x= f(lib.per(sorted, .9))
22   local y= f(lib.per(sorted, .1))
23   return (x - y)/denom end
24
25 function lib.cosine(a,b,c)
26   return math.max(0,math.min(1, (a^2+c^2-b^2)/(2*c+1E-32))) end
27
28 --- check
29 function lib.ish(x,y,z) return math.abs(x-y) <= (z or 0.001) end
30
31 --- filtering
32 function lib.inc(f,a,n) f=f or();f[a]=(f[a] or 0) + (n or 1) return f end
33 function lib.inc2(f,a,b,n) f=f or();f[a]=lib.inc(f[a] or (),b,n); return f end
34 function lib.inc3(f,a,b,c,n) f=f or();f[a]=lib.inc2(f[a] or (),b,c,n); return f end
35
36 function lib.has(f,a) return f[a] or 0 end
37 function lib.has2(f,a,b) return f[a] and lib.has(f[a],b) or 0 end
38 function lib.has3(f,a,b,c) return f[a] and lib.has2(f[a],b,c) or 0 end
39
40 --- lists
41 lib.unpack = table.unpack
42 function lib.push(t,x) t[1 + #t] = x; return x end
43
44 function lib.powerset(s)
45   local function fun(s)
46     local t = {}
47     for i = 1, #s do
48       for j = 1, #t do
49         t[#t+1] = {s[i], lib.unpack(t[j])} end end
50     return t end
51   return lib.sort(fun(s), function(a,b) return #a < #b end) end
52
53 function lib.merge(b4, merge)
54   local j,n,tmp = 1,#b4,{}
55   while j<#n do
56     local a, b = b4[j], b4[j+1]
57     if b then
58       local c = merge(a, b) -- returns nil if merge fails
59       if c then
60         a,j = c,j+1 end end
61     tmp[#tmp+1] = a
62     j = j+1 end
63   return #tmp==#b4 and tmp or lib.merge(tmp,merge) end
64
65 --- filtering
66 function lib.map(t, f, u)
67   u={}; for k,v in pairs(t) do u[1+#u]=f(v) end; return u end
68 function lib.collect(t,f,u)
69   u={}; for k,v in pairs(t) do u[k]=f(k,v) end; return u end
70 function lib.copy(t, u)
71   if type(t) ~= "table" then return t end
72   u={}; for k,v in pairs(t) do u[lib.copy(k)] = lib.copy(v) end; return u end
73
74 --- sorting
75 function lib.sort(t,f) table.sort(t,f); return t end
76
77 function lib.upx(a,b) return a.x < b.x end
78 function lib.upl(a,b) return a[1] < b[1] end
79 function lib.downl(a,b) return a[1] > b[1] end
80
81 function lib.slots(t, u)
82   local function public(k) return tostring(k):sub(1,1) ~= "_" end
83   u={};for k,v in pairs(t) do if public(k) then u[1+#u]=k end end
84   return lib.sort(u) end
85
86 --- set array up
87
88 function lib.settings(help)
89   local d,used = {},{}
90   help:sub(1,"[+][%s%+)][%s]+(-[%s%+)][%s%+)][%s%+)]",
91     -- e.g. " -bins -b max.number of bins = 16"
92     --parses to ((-)(bins)) (-b) max number of bins = (16)
93     -- i.e. (long (key)) (short) (x)
94   function (long,key,short,x)
95     assert(not used[short], "repeated short flag ["..short.."]")
96     used[short]=short
97     for n,flag in ipairs(arg) do
98       if flag==long then
99         x = x=="false" and true or x=="true" and "false" or arg[n+1] end end
100     d[key] = lib.coerce(x) end
101   if d.help then os.exit(print(help)) end
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120

```

```

121 return d end
122
123 lib.go = {_fails=0}
124 lib.no = {}
125 function lib.ok(test,msg)
126   print("", test and "PASS" or "FAIL",msg or "")
127   if not test then
128     lib.go._fails= lib.go._fails+1
129     if the and the.dump then assert(test,msg) end end end
130
131 function lib.main(the,go,b4, resets,todos)
132   todos = the.todo == "all" and lib.slots(go) or {the.todo}
133   resets={}; for k,v in pairs(the) do resets[k]=v end
134   go._fails = 0
135   for _,todo in pairs(todos) do
136     math.randomseed(the.seed or 10019)
137     if go[todo] then print("M"..todo); go[todo]() end
138     for k,v in pairs(resets) do the[k]=v end
139     for k,v in pairs(_ENV) do
140       if b4 and not b4[k] then print("?",k,type(v)) end end
141     os.exit(go._fails) end
142
143 --- solution
144 function lib.any(a,lo,hi)
145   lo,hi = lo or 1, hi or #a; return a[ (lo+(hi-lo)*math.random())/1 ] end
146
147 function lib.many(a,n,lo,hi, u)
148   u={}; for j=1,n do lib.push(u, lib.any(a,lo,hi)) end; return u end
149
150 function lib.slice(a,lo,hi, u)
151   u, lo, hi = {}, lo or 1, hi or #a
152   hi = math.min(hi,#a)
153   for j=lo,hi do u[1+#u]=a[j] end; return u end
154
155 --- string '2 thing
156 function lib.words(s,sep, t)
157   sep="["..(sep or " ") .. "]"
158   t={}; for y in s:gmatch(sep) do t[1+#t] = y end; return t end
159
160 function lib.coerces(s)
161   return lib.map(lib.words(s), lib.coerce) end
162
163 function lib.coerce(x)
164   if type(x) ~= "string" then return x end
165   x = x:match("%(%-)?%s%")
166   if x=="true" then return true elseif x=="false" then return false end
167   return math.tointeger(x) or tonumber(x) or x end
168
169 function lib.items(src,f)
170   local function file(f)
171     src,f = io.input(src),(f or lib.coerces)
172     return function(x) x=io.read()
173       if x then return f(x) else io.close(src) end end end
174   local function tbl( x)
175     x,f = 0, f or function(z) return z end
176     return function() if x< #src then x=x+1; return f(src[x]) end end end
177   if src then return type(src) == "string" and file(f) or tbl() end end
178
179 --- things '2 string
180 lib.fmt = string.format
181
182 function lib.oo(t, slots) print(lib.o(t,slots)) end
183
184 function lib.o(t,slots, seen, u)
185   if type(t) ~= "table" then return tostring(t) end
186   seen = seen or {}
187   if seen[t] then return "..." end
188   seen[t] = t
189   local function show1(x) return lib.o(x, nil, seen) end
190   local function show2(k) return lib.fmt("%s%s",k, lib.o(t[k], nil, seen)) end
191   u = #t>0 and lib.map(t,show1) or lib.map(slots or lib.slots(t),show2)
192   return (t._is or "").. ("["..table.concat(u," ")]") end
193
194 function lib.dent(t, seen,pre)
195   pre,seen = pre or "", seen or {}
196   if seen[t] then t = "..." end
197   if type(t) ~= "table" then return print(pre .. tostring(t)) end
198   seen[t]=t
199   for key,k in pairs(lib.slots(t)) do
200     local v = t[k]
201     io.write(lib.fmt ("%s%s%s",pre,k, type(v)=="table" and "\n" or " "))
202     if type(v)=="table" then lib.dent(v,seen,"|" ..pre)
203     else print(v) end end end
204
205 function lib.rnds(t,f)
206   return lib.map(t, function(x) return lib.rnd(x,f) end) end
207
208 function lib.rnd(x,f)
209   return lib.fmt(type(x)=="number" and (x-x//1 and f or "%5.2f") or "%s",x) end
210
211 --- o b j e c t
212
213 local _id=0
214 function lib.id() _id=_id+1; return _id end
215
216 function lib.class(name,base)
217   local klass,base_ctor = {}
218   if base then
219     for k,v in pairs(base) do klass[k] = v end
220     klass._base = base
221     base_ctor = rawget(base,'new') end
222   klass.__index = klass
223   klass._is = name
224   klass._class = klass
225   return setmetatable(klass,{
226     __call = function(klass,...)
227       local obj = setmetatable({},klass)
228       if rawget(klass,'new')
229
230
231
232
233
234
235
236
237
238
239
240

```

```

241 then klass.super = base_ctor
242   local res = klass.new(obj,...)
243   if res then obj = setmetatable(res,klass) end
244   elseif base_ctor then base_ctor(obj,...) end
245   return obj end }) end
246
247 lib.Obj = lib.class("Obj")
248
249 function lib.Obj:show( t)
250   t={}
251   for k,v in pairs(self) do if tostring(k):sub(1,1) ~= "." then t[1+#t]=k end end
252   return lib.sort(t) end
253
254 function lib.Obj:__tostring( u) return lib.o(self,self:show()) end
255
256 --u={}; for _,k in pairs(self:show()) do u[1+#u]=lib.fmt("%s %s",k,self[k]) end
257 -- return self._is .. ("["..table.concat(u," ")]") end
258
259 return lib

```