


```

213 -----
214 == SUPER RANGES
215
216
217
218 function nb3(file, log)
219     local tmp, i, create, update, discretizel, discretize = {}
220     i = {h={}, nh=0, e={}, names=nil, n=0, wait=the.wait,
221         bests=0, rests=0, best={}, rest={}, log=log or {},
222         nums={}}
223
224 function create(t)
225     for j,txt in pairs(t) do
226         if ako.num(txt) then i.nums[j] = {} end end; return t end
227
228 function update(t, x)
229     for j,n in pairs(i.nums) do
230         x=t[j]
231         if x~="?" then push(n, {x=x, y= t[#t]}) end end; return t end
232
233 function discretizel(t,x)
234     if x == "?" then return x end
235     for j,b in pairs(t) do if b.lo <= x and x < b.hi then return j end end end
236
237 function discretize(t, x)
238     for j,bins in pairs(i.nums) do t[j] = discretizel(bins,t[j]) end
239     return t end
240
241 tmp={}
242 for row in lines(file) do
243     if not i.names then i.names = create(row) else push(tmp,update(row)) end end
244 for j,xys in pairs(i.nums) do i.nums[j] = bins(xys) end
245 for _,row in pairs(tmp) do
246     discretize(row);
247     test(i,row); train(i,row) end
248 return i end
249
250 == kind bins
251
252
253 function bins(xys)
254     xys = sort(xys, upx)
255     local cohen = the.cohen * (per(xys,.9).x - per(xys, .1).x) / 2.54
256     local minItems = #xys / the.bins
257     local out, b4 = {}, -math.huge
258     local function add(f,z) f[z] = (f[z] or 0) + 1 end
259     local function sub(f,z) f[z] = f[z] - 1 end
260     local function argmin(lo,hi)
261         local lhs, rhs, cut, div, xpect, xy = {},{}
262         for j=lo,hi do add(rhs, xys[j].y) end
263         div = ent(rhs)
264         if hi-lo+1 > 2*minItems
265             then
266             for j=lo,hi - minItems do
267                 add(lhs, xys[j].y)
268                 sub(rhs, xys[j].y)
269                 local n1,n2 = j - lo +1, hi-j
270                 if n1 > minItems and
271                     xys[j].x ~ xys[j+1].x and -- enough items (on left)
272                     xys[j].x - xys[lo].x > cohen and -- there is a break here
273                     xys[hi].x - xys[j].x > cohen and -- not trivially small (on left)
274                     then xpect = (n1*ent(lhs) + n2*ent(rhs)) / (n1+n2) -- not trivially small (on right)
275                         if xpect < div then -- cutting here simplifies things
276                             cut, div = j, xpect --end for
277                         end -- end if
278                     if cut
279                         then argmin(lo, cut)
280                             argmin(cut+1, hi )
281                         else b4 = push(out, {lo=b4, hi=xys[hi].x, n=hi-lo+1, div=div}).hi end
282                     end
283                 argmin(1,#xys)
284                 out[#out].hi = math.huge
285             return out end

```

```

286 -----
287 == MISC
288
289
290
291 == maths
292
293
294 min = math.min
295 max = math.max
296
297 function per(t,p) return t[ (p or .5)*#t//1 ] end
298
299 function ent(t)
300     local n=0; for _,m in pairs(t) do n = n+m end
301     local e=0; for _,m in pairs(t) do if m>0 then e = e+m/n*math.log(m/n,2) end end
302     return -e end
303
304 == check
305
306 function rogues()
307     for k,v in pairs(_ENV) do if not b4[k] then print("??",k,type(v)) end end end
308
309
310 == count
311
312
313 function inc(f,a,n) f=f or {}; f[a]=(f[a] or 0) + (n or 1) return f end
314 function inc2(f,a,b,n) f=f or {}; f[a]=inc( f[a] or {},b,n); return f end
315 function inc3(f,a,b,c,n) f=f or {}; f[a]=inc2(f[a] or {},b,c,n); return f end
316
317 function has(f,a) return f[a] or 0 end
318 function has2(f,a,b) return f[a] and has( f[a],b) or 0 end
319 function has3(f,a,b,c) return f[a] and has2(f[a],b,c) or 0 end
320
321 == lists
322
323
324 function push(t,x) t[1 + #t] = x; return x end
325
326 function map(t,f, u) u={}; for k,v in pairs(t) do u[1+#u]=f(v) end; return u end
327
328 function sort(t,f) table.sort(t,f); return t end
329
330 function upx(a,b) return a.x < b.x end
331 function upl(a,b) return a[1] < b[1] end
332 function downl(a,b) return a[1] > b[1] end
333
334
335 function slots(t, u)
336     local function public(k) return tostring(k):sub(1,1) ~= "-" end
337     u={}; for k,v in pairs(t) do if public(k) then u[1+#u]=k end end
338     return sort(u) end
339
340 == string '2 things
341
342
343 function words(s,sep, t)
344     sep="([^\n .. (sep or ",") .. "]+)"
345     t={}; for y in s:gmatch(sep) do t[1+#t] = y end; return t end
346
347 function things(s) return map(words(s), thing) end
348
349 function thing(x)
350     x = x:gmatch("%s*(-)%s*$")
351     if x=="true" then return true elseif x=="false" then return false end
352     return tonumber(x) or x end
353
354
355 function lines(file,f, x)
356     file = io.input(file)
357     f = f or things
358     return function() x=io.read(); if x then return f(x) else io.close(file) end end
359
360 == things '2 string
361
362
363
364 fmt = string.format
365
366 function oo(t) print(o(t)) end
367
368 function o(t, seen, u)
369     if type(t)~="table" then return tostring(t) end
370     seen = seen or {}
371     if seen[t] then return "..." end
372     seen[t] = t
373     local function show1(x) return o(x, seen) end
374     local function show2(k) return fmt("%.8s",k, o(t[k],seen)) end
375     u = #t>0 and map(t,show1) or map(slots(t),show2)
376     return (t.s or "").."{"..table.concat(u, " ").."}" end
377
378
379 == cli
380
381 function cli(help)
382     local d,used = {},{}
383     help:gsub("(--[^(%s+)])(%s+)(--[^(%s+)](\\n)%s(^(%s+))",
384         function(long,key,short,x)
385             assert(not used[short], "repeated short flag ["..short.."]")
386             used[short]=short
387             for n,flag in ipairs(arg) do
388                 if flag==short or flag==long then
389                     x = x=="false" and true or x=="true" and "false" or arg[n+1] end end
390             d[key] = x==true and true or thing(x) end)
391     if d.help then os.exit(print(help)) end
392     return d end
393

```

```

393 -----
394 --- DEMOS
395 ---
396 ---
397
398 function eg.ent()
399     print(ent{a=9,b=7}) end
400
401 function eg.nb1()
402     local i = nb1(the.file);
403     local acc, out = score(i); print(acc); map(out,oo) end
404
405 function eg.nb2()
406     local i = nb2(the.file);
407     local acc, out = score(i); print(acc); map(out,oo) end
408
409 function eg.nb2a()
410     local i = nb2(the.file);
411     local acc, out = score(i)
412     abcd(i.log, true)
413     map(out,oo) end
414
415 function eg.bins( t)
416     local t,n = {},30
417     for j=1,n do push(t, {x=j, y=j<.6*n and 1 or j<.8*n and 2 or 3}) end
418     map(bins(t),oo)
419 end
420
421 function eg.nb3( i)
422     print(20)
423     i=nb3("/etc/data/diabetes.csv")
424     for n,bins in pairs(i.nums) do
425         print(n,#bins) end
426     local acc, out = score(i) -- XXX
427     print(#out)
428     print(acc)
429     map(out,oo)
430     end
431

```

```

431 -----
432 --- START
433 ---
434 ---
435
436 the=cli(help)
437 math.randomseed( the.seed or 10019 )
438 if eg[the.todo] then eg[the.todo]() end
439 rogues()

```