
This is the title of the template article

Firstname Lastname, University of Examples

Here is some sample text to show the initial in the introductory paragraph of this template article. The color and lineheight of the initial can be modified in the preamble of this document.

Heading on level 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi.

Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies

$$A = \begin{bmatrix} A_{11} & A_{21} \\ A_{21} & A_{22} \end{bmatrix} \quad (1)$$

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus.

Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies

287

Heading on level 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem.

- First item in a list
- Second item in a list
- Third item in a list

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim.

Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies

```
#!/usr/bin/env python3.9
# vim: ts=2 sw=2 sts=2 et :
# autopep8: ignore E20,E401,E226,E302,E41
```

```

4 import re, sys, math, argparse, itertools
5 from argparse import ArgumentParser as parse
6 from argparse import RawTextHelpFormatter as textual
7 Float = Str = Int = Bool = lambda *l: l[0]
8 fails=0
9
10 def keys(
11     BINS : Float("bins are of size n**BINS") = .5,
12     COLS : Str("columns to use for inference") = "x",
13     DATA : Str("where to read data") = "../data/auto2.csv",
14     EPSILON: Float("small = sd**EPSILON") = .3,
15     FAR : Float("where to look for far things") = .9,
16     GOAL : Str("learning goals: best—rest—other") = "best",
17     K : Int("bayes low class frequency hack") = 2,
18     M : Int("bayes low range frequency hack") = 1,
19     P : Int("distance calculation exponent") = 2,
20     SAMPLE : Int("#samples to find far things") = 20,
21     VERBOSE: Bool("set verbose") = False,
22     TOP : Int("focus on this many") = 20,
23     XAMPLE : Str("egs: fl-x lsfl lists all, fl-x allfl runs all") = "" ):
24     """
25     (c) Tim Menzies, 2021, unlicense.org.
26     The delta between things is
27     simpler than the things.
28     """
29
30     GOAL = -flbestfl : lambda b, r: b**2/b+r,
31     flrestfl : lambda b, r: r**2/(b+r),
32     flotherfl: lambda b, r: 1/(b+r) "[GOAL]
33
34     class Col(o):
35         "Store columns in `Col`, `Skip`, `Sym`, `Num`."
36         def __init__(i, at=0, txt="", inits=[]):
37             i.n, i.at, i.txt = 0, at, txt
38             i.w = -1 if "-" in txt else 1
39             [i.add(x) for x in inits]
40
41         def add(i, x, n=1):
42             if x != "?": i.n += 1; x = i.add1(x, n)
43             return x
44
45         # -----
46         class Skip(Col):
47             def add1(i, x, n=1): return x
48
49         # -----
50         class Sym(Col):
51             def __init__(i, **kw): i.has = "-"; super().__init__(**kw)
52
53             def add1(i, x, n=1): inc(i.has, x, n); return x
54
55         def bins(i, j):
56             for k in (i.has — j.has):
57                 yield i.has.get(k, 0), True, (i.at, (k, k))
58                 yield j.has.get(k, 0), False, (j.at, (k, k))
59
60         def dist(i, x, y): return 0 if x == y else 1
61
62         def ent(i):
63             return sum(-v/i.n * math.log(v/i.n) for v in i.has.values())
64
65         def merge(i, j):
66             k = Sym(at=i.at, txt=i.txt)
67             [k.add(x, n) for has in (i.has, j.has) for x, n in has.items()]
68             return k
69
70         def merged(i, j):
71             k = i.merge(j)
72             e1, n1, e2, n2, e, n = i.ent(), i.n, j.ent(), j.n, k.ent(), k.n
73             if e1 + e2 > 0.01 or e * .95 > n1 / n * e1 + n2 / n * e2:
74                 return k
75
76         # -----
77         class Num(Col):
78             def __init__(i, **kw):
79                 i.all, i.ok = [], False
80                 super().__init__(**kw)
81
82             def add1(i, x, n):
83                 x, i.ok = float(x), False
84                 for i in range(n): i.all += [x]
85                 return x
86
87             def all(i):
88                 if not i.ok: i.ok = True; i.all = sorted(i.all)
89                 return i.all
90
91 def bins(i, j):
92     xy = [(z, True) for z in i.all] + [(z, False) for z in j.all]
93     eps = EPSILON * (i.n*i.sd() + j.n*j.sd()) / (i.n + j.n)
94     for ((lo, hi), s) in bins(xy, epsilon=eps, size=len(xy)*BINS):
95         for klass, n in s.has.items():
96             yield n, klass, (i.at, (lo, hi))
97
98 def dist(i, x, y):
99     if x == "?": y = i.norm(y); x = 1 if y > 0.5 else 0
100     elif y == "?": x = i.norm(x); y = 1 if x > 0.5 else 0
101     else : x, y = i.norm(x), i.norm(y)
102     return abs(x-y)
103
104 def norm(i, x):
105     if x == "?": return x
106     a = i.all()
107     return max(0, min(1, (x-first(a))/(last(a)-first(a)+1E-32)))
108
109 def sd(i) : return (per(i.all(), .9) - per(i.all(), .1))/2.56
110 def span(i) : return (first(i.all()), last(i.all()))
111 def wide(i, n=0): return last(i.all()) - first(i.all()) i= n
112
113 # -----
114 class Row(o):
115     "Data is in `Row`s which, in turn, are in `Table`s."
116     def __init__(i, lst, tab=None): i.tab, i.cells = tab, lst
117
118 def dist(i, j):
119     d = n = 1E-32
120     for col in i.tab.cols[COLS]:
121         n += 1
122         x, y = i.cells[at], j.cells[at]
123         d += 1 if x == "?" and y == "?" else col.dist(x, y) ** P
124     return (d/n) ** (1/P)
125
126 def far(i, rows):
127     tmp = [(dist(i, j), j) for j in range(SAMPLE)]
128     return per(sorted(tmp, key=first), FAR)
129
130 # -----
131 class Table(o):
132     def __init__(i, inits=[]):
133         i.rows = []
134         i.cols = o(all=[], names=[], x=[], y=[], klass=None)
135         [i.add(x) for x in inits]
136
137     def add(i, a): i.data(a) if i.cols.names else i.header(a)
138     def clone(i, inits=[]): return Table([i.cols.names] + inits)
139
140     def data(i, a):
141         a = a.cells if type(a) == Row else a
142         a = [col.add(a[col.at]) for col in i.cols.all]
143         i.rows += [Row(a, tab=i)]
144
145     def header(i, a):
146         i.cols.names = a
147         for at, x in enumerate(a):
148             new = Skip if i.skipp(x) else (Num if i.nump(x) else Sym)
149             new = new(at=at, txt=x)
150             i.cols.all += [new]
151             if not i.skipp(x):
152                 i.cols["y" if i.y(x) else "x"] += [new]
153                 if i.klass(x):
154                     i.cols.klass = new
155
156     def klassp(i, x): return "!" in x
157     def nump(i, x): return x[0].isupper()
158     def skipp(i, x): return "?" in x
159     def yp(i, x): return "-" in x or "+" in x or i.klassp(x)
160
161 # -----
162 def stratify(src):
163     all, klass = None, "-"
164     for n, row in enumerate(src):
165         if all:
166             kl = row[all.cols.klass.at]
167             here = klass[kl] = klass.get(kl, None) or all.clone()
168             here.add(row)
169             all.add(row)
170         else:
171             all = Table([row])
172     return o(all=all, klass=klass)
173
174 # -----
175 def bins(xy, epsilon=0, size=30):

```

```

170 "Use `bins` to divide numeric data into ranges."
171 def merge(b4):
172     j, tmp, n = 0, [], len(b4)
173     while j < n:
174         a = b4[j]
175         if j < n - 1:
176             b = b4[j + 1]
177             print("na", a[1])
178             print("b", b[1])
179             if cy := a[1].merged(b[1]):
180                 print("c", cy)
181                 a = ((a[0][0], b[0][1]), cy)
182                 j += 1
183             tmp += [a]
184             j += 1
185     return merge(tmp) if len(tmp) < len(b4) else b4
186
187 def divide(xy):
188     bin = o(x=Num(), y=Sym())
189     bins = [bin]
190     for i, (x, y) in enumerate(xy):
191         if bin.x.n < size:
192             if x != b4 and i < len(xy)-size and bin.x.wide(epsilon):
193                 bin = o(x=Num(), y=Sym())
194                 bins += [bin]
195                 bin.x.add(x)
196                 bin.y.add(y)
197                 b4 = x
198     return bins
199
200 return merge([(bin.x.span(), bin.y)
201               for bin in divide(sorted(xy, key=first))])
202
203 #
204 def contrasts(here, there, t):
205     "Report ranges that are most different in two classes."
206     def like(d, kl):
207         out = prior = (hs[kl] + K) / (n + K*2)
208         for at, span in d.items():
209             f = has.get((kl, (at, span)), 0)
210             out *= (f + M*prior) / (hs[kl] + M)
211         return out
212
213 def val(d): return GOAL(like(d, True), like(d, False)), d
214 def top(a): return sorted(a, reversed=True, key=first)[-TOP]
215
216 has = -(kl, (at, (lo, hi))): f
217     for col1, col2 in zip(here.cols.x, there.cols.x)
218     for f, kl, (at, (lo, hi)) in col1.bins(col2)
219 n = len(here.rows, there.rows)
220 hs = -True: len(here.rows), False: len(there.rows)
221 solos = [val(dict(at=x)) for at, x in set([z for `z` in has])]
222 ranges = "-"
223 for `d` in top(solos):
224     for k in d:
225         ranges[k] = ranges.get(k, set()).add(d[k])
226 for rule in top([val(d) for d in dict`product(ranges)]):
227     print(rule)
228
229 #
230 # Unit tests.
231 class Eg:
232     def ls():
233         "list all examples."
234         print("nexamples:")
235         for k, f in vars(Eg).items():
236             if k[0] != "f":
237                 print(f"-k:{k} -f: {f.__doc__}")
238
239 def data(file="~/data/vote.csv"):
240     "simple load of data into a table"
241     t = Table(csv(file))
242     assert(435 == len(t.rows))
243     assert(195 == t.cols.all[1].has[flyfl])
244
245 def nclasses(file="~/data/diabetes.csv", kl="positive"):
246     ts = stratify(csv(file))
247     assert(2 == len(ts.klass))
248     assert(268 == len(ts.klass[kl].rows))
249     assert(768 == len(ts.all.rows))
250
251 def bins(file="~/data/diabetes.csv",
252         kl="positive", k2="negative"):
253     ts = stratify(csv(file))
254     goods, bads = ts.klass[k1], ts.klass[k2]
255
256     for good, bad in zip(goods.cols.all, bads.cols.all):
257         print(f"n=good.at")
258         [print(f"t=x") for x in good.bins(bad)]
259
260 #
261 # main program for keys
262 if XAMPLE == "all":
263     for k, f in vars(Eg).items():
264         if k[0] != "f": print("n"+k); f()
265 else:
266     if XAMPLE and XAMPLE in vars(Eg): vars(Eg)[XAMPLE]()
267
268 #####
269 # things that don't use the config vars
270 # string stuff
271 def color(end="n", **kw):
272     s, a, z = "", "u001b[", ";1m"
273     c = dict(red="31", green="32", yellow="33", purple="34",
274             pink="35", blue="36", reset="033[0m")
275     for col, txt in kw.items(): s = s + a + c[col] + z + txt + c["reset"]
276     print(s, end=end)
277
278 # dictionary stuff
279 def has(d, k): return d.get(k, 0)
280 def inc(d, k, n=1): tmp = d[k] = n + d.get(k, 0); return tmp
281
282 def dict`product(d):
283     keys = d.keys()
284     for p in itertools.product(*d.values()):
285         yield dict(zip(keys, p))
286
287 # list stuff
288 def first(a): return a[0]
289 def last(a): return a[-1] #
290 def per(a, p=.5): return a[int(p*len(a))]
291
292 # object stuff
293 class o(object):
294     def __init__(i, **k): i.__dict__.update(**k)
295     def __getitem__(i, k): return i.__dict__[k]
296     def __repr__(i): return i.__class__.__name__ + str(
297         -k:v for k, v in i.__dict__.items() if k[0] != "f")
298     def __setitem__(i, k, v): i.__dict__[k] = v
299
300 # file stuff
301 def csv(f=None, sep=","):
302     def prep(s): return re.sub(rf[["n" "t" "r "] -#.*]fl, flfl, s)
303     if f:
304         with open(f) as fp:
305             for s in fp:
306                 if s := prep(s): yield s.split(sep)
307     else:
308         for s in sys.stdin:
309             if s := prep(s): yield s.split(sep)
310
311 # command-line stuff
312 def cli(f):
313     used, p = "-", parse(prog="."+f.__name__, description=f.__doc__,
314                          formatter`class=textual)
315     for (k, h), b4 in zip(list(f.__annotations__.items()), f.__defaults__):
316         k0 = k[0]
317         used[k0] = c = k0 if k0 in used else k0.lower()
318         if b4==False:
319             p.add`argument("-"+c, dest=k, help=h, default=False,
320                           action="store`true")
321         else: p.add`argument("-"+c, dest=k, default=b4,
322                             help=h+"["+str(b4)+"]", type=type(b4),
323                             metavar=k)
324     f(**p.parse`args()).__dict__
325
326 # start up stuff
327 if __name__ == "__main__": cli(keys)

```

Heading on level 1 again

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec,

Table 1: *Random table*

Name		
First name	Last Name	Grade
John	Doe	7.5
Richard	Miles	2

pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies

Heading on level 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo.

First This is the first item

Last This is the last item

Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies