This is the title of the template article

Firstname Lastname, University of Examples

ere 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, consectetuer 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

$$A = \begin{bmatrix} A_{11} & A_{21} \\ A_{21} & A_{22} \end{bmatrix} \tag{1}$$

Lorem ipsum dolor sit amet, consectetuer 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 ??

• •

Heading on level 2

Lorem ipsum dolor sit amet, consectetuer 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, consectetuer 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

```
import re, sys, math, argparse, itertools
                                                                                                                                                                                                                             def all(i):
 from argparse import ArgumentParser as parse
                                                                                                                                                                                                                                 if not i.ok: i.ok = True; i. all = sorted(i. all)
 from argparse import RawTextHelpFormatter as textual
                                                                                                                                                                                                   89
Float = Str = Int = Bool = lambda *I: I[0]
                                                                                                                                                                                                   90
                                                                                                                                                                                                                                  \begin{aligned} & \mathsf{x} \mathsf{y} = [(\mathsf{z}, \mathsf{True}) \; \mathsf{for} \; \mathsf{z} \; \mathsf{in} \; \mathsf{i}.\; \mathsf{all}] + [(\mathsf{z}, \; \mathsf{False}) \; \mathsf{for} \; \mathsf{z} \; \mathsf{in} \; \mathsf{j}.\; \mathsf{all}] \\ & \mathsf{eps} = \mathsf{EPSILON} \; * \; (\mathsf{i}.\mathsf{n} * \mathsf{i}.\mathsf{sd}() + \mathsf{j}.\mathsf{n} * \mathsf{j}.\mathsf{sd}()) \; / \; (\mathsf{i}.\mathsf{n} + \mathsf{j}.\mathsf{n}) \end{aligned} 
 def keys(
                                                                                                                                                                                                   93
    BINS : Float("bins are of size n**BINS") = .5,

COLS : Str("columns to use for inference") = "x",

DATA : Str("where to read data") = "../data/auto2.csv",

EPSILON: Float("small = sd**EPSILON") = .3,
                                                                                                                                                                                                                                 for ((lo, hi), ) in bins(xy,epsilon=eps,size=len(xy)**BINS): yield i.at, (lo, hi)
                                                                                                                                                                                                   94
                                                                                                                                                                                                   95
                                                                                                                                                                                                   96
                                                                                                                                                                                                                            \begin{array}{ll} \text{def dist}(i,\,\mathsf{x},\,\mathsf{y}) \colon \\ \text{if} \quad \mathsf{x} == \text{"?"} \colon \mathsf{y} = i.\mathsf{norm}(\mathsf{y});\, \mathsf{x} = 1 \text{ if } \mathsf{y} \mid 0.5 \text{ else } 0 \\ \text{elif } \mathsf{y} == \text{"?"} \colon \mathsf{x} = i.\mathsf{norm}(\mathsf{x}); \mathsf{y} = 1 \text{ if } \mathsf{x} \mid 0.5 \text{ else } 0 \\ \text{else} \qquad \vdots \, \mathsf{x},\, \mathsf{y} = i.\mathsf{norm}(\mathsf{x}),\, \mathsf{y}.\mathsf{norm}(\mathsf{y}) \end{array}
                                                                                                                                                                                                   97
                        : Float("where to look for far things") = .9,
: Str("learning goals: best—rest—other") = "best",
                                                                                                                                                                                                   98
                                                                                                                                                                                                   99
    GOAL: Str( learning goals: best—rest—other ) = K: Int("bayes low class frequency hack") = 2, M: Int("bayes low range frequency hack") = 1, P: Int("distance calculation exponent") = 2, SAMPLE: Int("#samples to find far things?") = 20, VERBOSE: Bool("set verbose") = False, TOP: Int("focus on this many") = 20, VAMPLE: Str("set if the life its ellife transfer.")
                                                                                                                                                                                                 100
                                                                                                                                                                                                                                 return abs(x-y)
                                                                                                                                                                                                 101
                                                                                                                                                                                                 102
                                                                                                                                                                                                                            \begin{array}{l} \text{def } \underset{\text{if } \times == \text{"?"}}{\text{norm}(i, \times)}: \\ \text{if } \times == \text{"?"}: \text{return } \times \end{array}
                                                                                                                                                                                                 103
                                                                                                                                                                                                 104
                                                                                                                                                                                                                                  a = i.all()
     XAMPLE : Str("egs: fl-x | sfl | lists | all, fl-x | allfl | runs | all") = "" ): 106
                                                                                                                                                                                                                                  return max(0, min(1, (x-first(a))/(last(a)-first(a)+1E-32)))
                                                                                                                                                                                                                            \begin{array}{l} \text{def sd(i): return (per(i.all(), .9) - per(i.all(), .1))/2.56} \\ \text{def span(i): return (first(i.all()), last(i.all()))} \end{array}
                                      (c) Tim Menzies, 2021, unlicense.org.
                " The delta between things is
                 .'* simpler than the things.
                                                                                                                                                                                                 109
                                                                                                                                                                                                                             def wide(i, n=0): return last(i.all()) - first(i.all()) \xi = n
                                                                                                                                                                                                 110
                                                                                                                                                                                                 111
                                                                                                                                                                                                112
     \begin{split} \mathsf{GOAL} &= -\mathsf{flbestfl} : \mathsf{lambda} \ \mathsf{b}, \ \mathsf{r} : \ \mathsf{b}^{**2}/\mathsf{b} + \mathsf{r}, \\ & \mathsf{flrestfl} : \mathsf{lambda} \ \mathsf{b}, \ \mathsf{r} : \ \mathsf{r}^{**2}/(\mathsf{b} + \mathsf{r}), \\ & \mathsf{flotherfl} : \mathsf{lambda} \ \mathsf{b}, \ \mathsf{r} : \ \mathbf{1}/(\mathsf{b} + \mathsf{r}) \quad \text{``[GOAL]} \end{split}
                                                                                                                                                                                                                             "Data is in `Row`s which, in turn, are in `Table`s."
                                                                                                                                                                                                113
                                                                                                                                                                                                                             def "init"(i, lst, tab=None): i.tab, i.cells = tab, lst
                                                                                                                                                                                                114
                                                                                                                                                                                                 115
                                                                                                                                                                                                                             def dist(i, j):
                                                                                                                                                                                                116
                                                                                                                                                                                                                                 d = n = 1E-32
    class Col(o):
                                                                                                                                                                                                117
         "Store columns in `Col`, `Skip`, `Sym`, `Num`."

def "init"(i, at=0, txt="", inits=[]):
                                                                                                                                                                                                                                  for col in i.tab.cols[COLS]:
                                                                                                                                                                                                118
                                                                                                                                                                                                                                      n += 1
                                                                                                                                                                                                119
                                                                                                                                                                                                                                     x, y = i.cells[at], j.cells[at]

d + 1 \text{ if } x = 2000 \text{ and } y = 2000 \text{ else col.dist}(x, y) ** P
             i.n, i.at, i.txt = \mathbf{0}, at, txt
                                                                                                                                                                                                120
              i.w = -1 if "-" in txt else 1
                                                                                                                                                                                                 121
                                                                                                                                                                                                                                  return (d/n) ** (1/P)
              i.klass=False
                                                                                                                                                                                                122
              [i.add(x) for x in inits]
                                                                                                                                                                                                123
                                                                                                                                                                                                                             def far(i, rows):
                                                                                                                                                                                                124
         \begin{array}{l} \text{def add(i, x, n=1):} \\ \text{if x != "?": i.n += n; x = i.add1(x, n)} \end{array}
                                                                                                                                                                                                                                 \mathsf{tmp} = [(\mathsf{dist}(\mathsf{i},\,\mathsf{j}),\,\mathsf{j})\;\mathsf{for}\;\;\mathsf{in}\;\mathsf{range}(\mathsf{SAMPLE})]
                                                                                                                                                                                                125
                                                                                                                                                                                                126
                                                                                                                                                                                                                                  return per(sorted(tmp, key=first), FAR)
              return x
                                                                                                                                                                                                 127
                                                                                                                                                                                                                         class Table(o):
                                                                                                                                                                                                128
                                                                                                                                                                                                                             def "init"(i, inits=[]):
    class Skip(Col):
                                                                                                                                                                                                 129
         def add1(i, \times, n=1): return \times
                                                                                                                                                                                                                                 i.rows = []
                                                                                                                                                                                                 130
                                                                                                                                                                                                                                  i.cols = o(all=[], names=[], x=[], y=[], klass=None)
                                                                                                                                                                                                 131
    class Sym(Col):
    def ''init''(i, **kw): i.has = -"; super().''init''(**kw)
                                                                                                                                                                                                 132
                                                                                                                                                                                                                                  [i.add(x) for x in inits]
                                                                                                                                                                                                 133
                                                                                                                                                                                                                             def add(i, a): i.data(a) if i.cols.names else i.header(a)
                                                                                                                                                                                                 134
         def add1(i, x, n=1): inc(i.has, x, n); return x
                                                                                                                                                                                                                             def clone(i, inits=[]): return Table([i.cols.names] + inits)
                                                                                                                                                                                                 135
                                                                                                                                                                                                 136
         def bins(i, j):
                                                                                                                                                                                                                             def data(i, a):
                                                                                                                                                                                                 137
              for k in (i.has — j.has):
                                                                                                                                                                                                                                 a = a.cells if type(a) == Row else a
                                                                                                                                                                                                 138
                   yield i has get(k, 0), True, (i.at, (k, k))
                                                                                                                                                                                                                                  a = [col.add(a[col.at]) for col in i.cols.all]
                                                                                                                                                                                                 139
                   yield j.has.get(k, 0), False, (j.at, (k, k))
                                                                                                                                                                                                                                  i.rows += [Row(a, tab=i)]
                                                                                                                                                                                                 140
                                                                                                                                                                                                141
         def dist(i, x, y): return 0 if x == y else 1
                                                                                                                                                                                                                             def header(i, a):
                                                                                                                                                                                                 142
                                                                                                                                                                                                                                 i.cols.names \stackrel{\checkmark}{=} a
                                                                                                                                                                                                 143
         def ent(i):
                                                                                                                                                                                                                                  for at, \times in enumerate(a):
                                                                                                                                                                                                  144
                                                                                                                                                                                                                                      new = Skip if i.skipp(x) else (Num if i.nump(x) else Sym)
              return sum(-v/i.n * math.log(v/i.n) for v in i.has.values())
                                                                                                                                                                                                145
                                                                                                                                                                                                                                      new = new(at=at, txt=x)
                                                                                                                                                                                                 146
         def merge(i, j):
                                                                                                                                                                                                                                      i.cols.all += [new]
                                                                                                                                                                                                 147
              k = Sym(at=i.at, txt=i.txt)
                                                                                                                                                                                                                                      if not i.skipp(x):
                                                                                                                                                                                                  148
                                                                                                                                                                                                                                          i.cols["y"] if i.yp(x) else "x"] += [new]
              [k.add(x, n) for has in (i.has, j.has) for x, n in has.items()]
                                                                                                                                                                                                149
                                                                                                                                                                                                                                           if i.klassp(x):
              return k
                                                                                                                                                                                                 150
                                                                                                                                                                                                                                               new.klass=True
         def merged(i, j):
                                                                                                                                                                                                                                               i.cols.klass = new
                                                                                                                                                                                                 152
              k = i.merge(j)
                                                                                                                                                                                                  153
              e1,n1, e2,n2, \vec{v}, \vec{v},
                                                                                                                                                                                                                              \begin{tabular}{ll} \be
                                                                                                                                                                                                 154
                                                                                                                                                                                                                            \begin{array}{l} \text{def nump(i, x): return x[0].isupper()} \\ \text{def skipp(i, x): return "?" in x} \\ \text{def yp(i, x): return "-" in x or "+" in x or i.klassp(x)} \end{array}
                                                                                                                                                                                                 156
              if e1 + e2 + 0.01 or e * .95 + tmp:
                                                                                                                                                                                                 157
                  return k
                                                                                                                                                                                                 158
                                                                                                                                                                                                 159
    class Num(Col):
    def 'init''(i, **kw):
        i.'all, i.ok = [], False
        super().''init''(**kw)
                                                                                                                                                                                                                         def stratify(src):
                                                                                                                                                                                                 160
                                                                                                                                                                                                                             all, klass \stackrel{\frown}{=} None,–"
                                                                                                                                                                                                 161
                                                                                                                                                                                                                             for n,row in enumerate(src):
                                                                                                                                                                                                163
                                                                                                                                                                                                                                         kl = row[all.cols.klass.at]
                                                                                                                                                                                                164
         def add1(i, x, n):
                                                                                                                                                                                                                                         here = klass[kl] = klass.get(kl,None) or all.clone()
                                                                                                                                                                                                165
              x, i.ok \stackrel{\cdot}{=} float(x), False
                                                                                                                                                                                                166
                                                                                                                                                                                                                                        here.add(row)
              for in range(n): i. all += [x]
                                                                                                                                                                                                                                         all.add(row)
                                                                                                                                                                                                167
                                                                                                                                                                                                168
                                                                                                                                                                                                                                         all = Table([row])
                                                                                                                                                                                                 169
```

6

9

10

11

12

13

14

17

18

19

20

21

22

23 24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

46

47

48

49

50

51

52

53

54

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

73

74

75

76

77

78

79

80

81

82

83

84

85

```
return o(all=all, klass=klass)
170
                                                                                      253
                                                                                                 def bins(file="../data/diabetes.csv"
171
                                                                                      254
                                                                                                          k1= "positive", k2= "negative"):
         def bins(xy, epsilon=0, size=30):
172
                                                                                      255
                                                                                                    "discretize some data"
            'Use `bins` to divide numeric data into ranges."
173
                                                                                      256
           def merge(b4):
                                                                                                   ts = stratify(csv(file))
174
                                                                                      257
             j, tmp, n = 0, [], len(b4)
                                                                                                   goods, bads = ts.klass[k1], ts.klass[k2]
175
                                                                                      258
                                                                                                   for good,bad in zip(goods.cols.all, bads.cols.all):
             while j | n:
                                                                                      259
               ((lo, '), ay) = a = b4[j]
if j \mid n - 1:
((', hi), by) = b4[j + 1]
                                                                                                     \mathsf{bins} = \mathsf{list}(\mathsf{good}.\mathsf{bins}(\mathsf{bad}))
177
                                                                                      260
                                                                                                     if len(bins) ¿ 1 and not good.klass:
178
                                                                                      261
                                                                                                       print(good.txt,bins)
                                                                                      262
170
                 if cy := ay.merged(by):
180
                                                                                      263
                  a = ((lo, hi), cy)
                                                                                                # main program for keys
181
                                                                                      264
                                                                                                Eg.bins()
                  i += 1
182
                                                                                      265
                                                                                                if XAMPLE == "all":
               tmp += [a]
183
                                                                                      266
                                                                                                 for k, f in vars(Eg).items():
    if k[0] != """: run(f)
               i += 1
184
                                                                                      267
             return merge(tmp) if len(tmp) i len(b4) else b4
185
                                                                                      268
                                                                                      269
186
                                                                                                else:
           def divide(xy):
                                                                                                 if XAMPLE and XAMPLE in vars(Eg):
187
                                                                                      270
188
             bin = o(x=Num(), y=Sym())
                                                                                                   f = vars(Eg)[XAMPLE]
                                                                                      271
                                                                                                   f() if XAMPLE=="Is" else run(f)
             bins = [bin]
189
190
             for i, (x, y) in enumerate(xy):
                                                                                      273
               if bin.x.n \ i = size:
                                                                                              191
                                                                                      274
                 if x != b4 and i | len(xy)-size and bin.x.wide(epsilon):
                                                                                              # things that donflt use the config vars
                                                                                      275
192
                   bin = o(x=Num(), y=Sym())
                                                                                      276
193
                   bins += [bin]
194
                                                                                      277
                                                                                              def color(end=" "n", **kw):

s, a, z = "", " "u001b[", ";1m"
               bin.x.add(x)
                                                                                      278
195
               bin.y.add(y)
196
                                                                                      279
                                                                                               c = dict(black=30, red=31, green=32, yellow=33, purple=34, pink=35, blue=36, white=37)
               b4 = x
                                                                                      280
197
             return bins
                                                                                      281
198
                                                                                                for col,txt in kw.items(): s = s+a + str(c[col]) + z+txt+" "033[0m"
                                                                                      282
199
           xy = sorted(xy, key=first)
                                                                                               print(s, end=end)
200
                                                                                      283
           return merge([(bin.x.span(), bin.y) for bin in divide(xy)])
                                                                                      284
201
                                                                                      285
202
         def contrasts(here, there, t):
                                                                                              # dictionary stuff
                                                                                      286
203
           "Report ranges that are most different in two classes."
                                                                                              def has(d, k): return d.get(k, 0)
204
                                                                                      287
                                                                                              def \ inc(d,\ k,\ n=1):\ tmp=\ d[k]=n+d.get(k,\ 0);\ return\ tmp
           def like(d, kl):
                                                                                      288
205
             \mathsf{out} = \mathsf{prior} = (\mathsf{hs}[\mathsf{kl}] + \mathsf{K}) \; / \; (\mathsf{n} + \mathsf{K*2})
206
                                                                                      280
             for at, span in d.items():
                                                                                              def dict'product(d):
207
                                                                                      290
               f = has.get((kl, (at, span)), 0)
208
                                                                                      291
                                                                                               keys = d.keys()
               out *= (f + M*prior) / (hs[kl] + M)
                                                                                                for p in itertools.product(*d.values()):
209
                                                                                      292
             return out
                                                                                                 yield dict(zip(keys, p))
210
                                                                                      293
211
                                                                                      294
           def val(d): return GOAL(like(d, True), like(d, False)), d
212
                                                                                      295
           def top(a): return sorted(a,reversed=True,key=first)[:TOP]
                                                                                              # list stuff
213
                                                                                      296
214
                                                                                              def first(a): return a[0]
                                                                                      297
                                                                                              def last(a): return a[-1]
           has = -(kl, (at, (lo, hi))): f
                                                                                      298
                  for col1, col2 in zip(here.cols.x, there.cols.x)
                                                                                              def per(a, p=.5): return a[int(p*len(a))]
216
                                                                                      299
                  for f, kl, (at, (lo, hi)) in col1.bins(col2)
217
                                                                                      300
218
           n = len(here.rows, there.rows)
                                                                                      301
219
           hs = -True: len(here.rows), False: len(there.rows)"
                                                                                      302
                                                                                              # object stuff
                                                                                             class o(object):

def "init"(i, **k): i."dict".update(**k)

def "getitem"(i, k): return i."dict"[k]

def "repr"(i): return i."class"."name"+str(

-k:v for k, v in i."dict".items() if k[0]!=""")
           solos = [val(dict(at=x)) \text{ for at, } x \text{ in } set([z \text{ for ', } z \text{ in } has])]
220
                                                                                      303
221
                                                                                      304
           for , d in top(solos):
                                                                                      305
222
             for k in d:
                                                                                      306
223
               ranges[k] = ranges.get(k, set()).add(d[k])
                                                                                      307
224
           for rule in top([val(d) for d in dict product(ranges)]):
                                                                                                def "setitem"(i, k, v): i. "dict"[k] = v
225
                                                                                      308
             print(rule)
226
                                                                                      309
227
                                                                                      310
          # Unit tests.
                                                                                              # file stuff
228
                                                                                      311
                                                                                              def csv(f=None, sep=","):
         class Eg
220
                                                                                      312
                                                                                               def prep(s): return re.sub(rfl(["n"t"r]-#.*)fl, flfl, s)
           def Is():
230
                                                                                      313
             "list all examples."
print(""nexamples:")
for k, f in vars(Eg).items():
    if k[0] != "": print(f" -k:i13" -f. "doc"")
231
                                                                                      314
                                                                                                 with open(f) as fp:
                                                                                      315
232
                                                                                      316
                                                                                                   for s in fp:
233
                                                                                                     if s := prep(s): yield s.split(sep)
234
                                                                                      317
                                                                                               else:
                                                                                      318
235
           def 'fail():
                                                                                                 for s in sys.stdin:
236
                                                                                      319
             "testing failure"
assert False, "failing"
                                                                                                   if s := prep(s): yield s.split(sep)
237
                                                                                      320
                                                                                      321
238
239
                                                                                      322
           def data(file="../data/vote.csv"):
                                                                                              # command-line stuff
240
                                                                                      323
             "simple load of data into a table"
                                                                                              def cli(f):
241
                                                                                      324
                                                                                                used, p' = -r', parse(prog="./"+f."name", description=f."doc",
             t = Table(csv(file))
                                                                                      325
             assert 435 == len(t.rows)
                                                                                                                  formatter class=textual)
243
                                                                                      326
             assert 195 == t.cols.all[1].has[flyfl]
                                                                                      327
                                                                                                for (k, h),b4 in zip(list(f. annotations items()),f. defaults:):
244
                                                                                      328
                                                                                                 k0 = k[0]
245
                                                                                                 used[k0] = c = k0 if k0 in used else k0.lower() if b4==False:
           def nclasses(file="../data/diabetes.csv", kl="positive"):
246
                                                                                      329
             "read data with nclasses"
247
                                                                                      330
             ts = stratify(csv(file))
                                                                                                       p.add argument("-"+c, dest=k, default=False,
248
                                                                                      331
                             == len(ts.klass)
                                                                                                                     help=h,
249
                                                                                      332
                                                                                                 \begin{array}{c} \text{neip=ii,} \\ \text{action="store'true"}) \\ \text{else: p.add'argument("-"+c, dest=k, default=b4,} \\ \text{help=h} + \frac{\text{l"}}{\text{l"}} + \text{str(b4)} + \frac{\text{"]}^{\text{l"}}}{\text{l}}, \end{array}
             assert 268
                             == len(ts.klass[kl].rows)
250
                                                                                      333
                              == len(ts.all.rows)
251
                                                                                      334
             assert 3.90625 == ts.klass[kl].cols.all[0].sd()
252
                                                                                      335
```

Table 1: Random table

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

```
type=type(b4), metavar=k)
336
        f( **p.parse args(). "dict"
337
338
339
       # tests
340
      fails=0
341
      def run(fun):
342
        global fails
343
        s= f" -fun."name":;12""
344
345
346
         color(green=(chr(10003)+ s),white=fun."doc")
347
348
        except Exception as err:
349
         fails +=1
         color(red=(chr(10007)+ s), white= str(err))
350
351
352
353
       # start up stuff
         "name" == ""main"":
      if
354
        cli(keys)
355
        sys.exit(fails)
356
```

Heading on level 1 again

Lorem ipsum dolor sit amet, consectetuer 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

Heading on level 2

Lorem ipsum dolor sit amet, consectetuer 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, consectetuer 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