

Modeling discrete character evolution on trees

```
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function Project (None)
Untitled1
Source on Save Run Source
Environment History Connections Tutorial
Files Plots Packages Help Viewer Presentation
Zoom Export Publish
13 <source>(file="~/R/phytools.R")
14 data(sunfish.tree)
15 data(sunfish.data)
16 ## extract discrete character (feeding mode)
17 fmode<-setNames(sunfish.data$feeding.mode,
18 rownames(sunfish.data))
19 ## do stochastic mapping
20 smap.trees<-make.simmap(sunfish.tree,fmode,model="ER",
21 nsim=100)
22 ## print a summary of the stochastic mapping
23 summary(smap.trees)
24 ## plot a posterior probability distribution of states
25 cols<-setNames(c("blue","red"))
26 plot(summary(smap.trees), colors=cols, ftype="1",
27 Legend="topleft",c("non-piscivorous","piscivorous"),
28 topLevel=c("non-piscivorous","piscivorous"),
29 topLevelColor=c("blue","red"))
30 legend("topright",c("non-piscivorous","piscivorous"),
31 topLevel=c("non-piscivorous","piscivorous"),
32 topLevelColor=c("blue","red"))
33
34 ## plot a posterior probability distribution of states
35 colors<-setNames(c("lightblue","red"))
36 plot(summary(smap.trees), colors=colors, ftype="1",
37 topLevel=c("non-piscivorous","piscivorous"),
38 topLevelColor=c("lightblue","red"))
39
40 ## plot a posterior probability distribution of states
41 colors<-setNames(c("lightblue","red"))
42 plot(summary(smap.trees), colors=colors, ftype="1",
43 topLevel=c("non-piscivorous","piscivorous"),
44 topLevelColor=c("lightblue","red"))
45
46 ## plot a posterior probability distribution of states
47 colors<-setNames(c("lightblue","red"))
48 plot(summary(smap.trees), colors=colors, ftype="1",
49 topLevel=c("non-piscivorous","piscivorous"),
50 topLevelColor=c("lightblue","red"))
51
52 ## plot a posterior probability distribution of states
53 colors<-setNames(c("lightblue","red"))
54 plot(summary(smap.trees), colors=colors, ftype="1",
55 topLevel=c("non-piscivorous","piscivorous"),
56 topLevelColor=c("lightblue","red"))
57
58 ## plot a posterior probability distribution of states
59 colors<-setNames(c("lightblue","red"))
60 plot(summary(smap.trees), colors=colors, ftype="1",
61 topLevel=c("non-piscivorous","piscivorous"),
62 topLevelColor=c("lightblue","red"))
63
64 ## plot a posterior probability distribution of states
65 colors<-setNames(c("lightblue","red"))
66 plot(summary(smap.trees), colors=colors, ftype="1",
67 topLevel=c("non-piscivorous","piscivorous"),
68 topLevelColor=c("lightblue","red"))
69
70 ## plot a posterior probability distribution of states
71 colors<-setNames(c("lightblue","red"))
72 plot(summary(smap.trees), colors=colors, ftype="1",
73 topLevel=c("non-piscivorous","piscivorous"),
74 topLevelColor=c("lightblue","red"))
75
76 ## plot a posterior probability distribution of states
77 colors<-setNames(c("lightblue","red"))
78 plot(summary(smap.trees), colors=colors, ftype="1",
79 topLevel=c("non-piscivorous","piscivorous"),
80 topLevelColor=c("lightblue","red"))
81
82 ## plot a posterior probability distribution of states
83 colors<-setNames(c("lightblue","red"))
84 plot(summary(smap.trees), colors=colors, ftype="1",
85 topLevel=c("non-piscivorous","piscivorous"),
86 topLevelColor=c("lightblue","red"))
87
88 ## plot a posterior probability distribution of states
89 colors<-setNames(c("lightblue","red"))
90 plot(summary(smap.trees), colors=colors, ftype="1",
91 topLevel=c("non-piscivorous","piscivorous"),
92 topLevelColor=c("lightblue","red"))
93
94 ## plot a posterior probability distribution of states
95 colors<-setNames(c("lightblue","red"))
96 plot(summary(smap.trees), colors=colors, ftype="1",
97 topLevel=c("non-piscivorous","piscivorous"),
98 topLevelColor=c("lightblue","red"))
99
100 ## plot a posterior probability distribution of states
101 colors<-setNames(c("lightblue","red"))
102 plot(summary(smap.trees), colors=colors, ftype="1",
103 topLevel=c("non-piscivorous","piscivorous"),
104 topLevelColor=c("lightblue","red"))
105
106 ## plot a posterior probability distribution of states
107 colors<-setNames(c("lightblue","red"))
108 plot(summary(smap.trees), colors=colors, ftype="1",
109 topLevel=c("non-piscivorous","piscivorous"),
110 topLevelColor=c("lightblue","red"))
111
112 ## plot a posterior probability distribution of states
113 colors<-setNames(c("lightblue","red"))
114 plot(summary(smap.trees), colors=colors, ftype="1",
115 topLevel=c("non-piscivorous","piscivorous"),
116 topLevelColor=c("lightblue","red"))
117
118 ## plot a posterior probability distribution of states
119 colors<-setNames(c("lightblue","red"))
120 plot(summary(smap.trees), colors=colors, ftype="1",
121 topLevel=c("non-piscivorous","piscivorous"),
122 topLevelColor=c("lightblue","red"))
123
124 ## plot a posterior probability distribution of states
125 colors<-setNames(c("lightblue","red"))
126 plot(summary(smap.trees), colors=colors, ftype="1",
127 topLevel=c("non-piscivorous","piscivorous"),
128 topLevelColor=c("lightblue","red"))
129
130 ## plot a posterior probability distribution of states
131 colors<-setNames(c("lightblue","red"))
132 plot(summary(smap.trees), colors=colors, ftype="1",
133 topLevel=c("non-piscivorous","piscivorous"),
134 topLevelColor=c("lightblue","red"))
135
136 ## plot a posterior probability distribution of states
137 colors<-setNames(c("lightblue","red"))
138 plot(summary(smap.trees), colors=colors, ftype="1",
139 topLevel=c("non-piscivorous","piscivorous"),
140 topLevelColor=c("lightblue","red"))
141
142 ## plot a posterior probability distribution of states
143 colors<-setNames(c("lightblue","red"))
144 plot(summary(smap.trees), colors=colors, ftype="1",
145 topLevel=c("non-piscivorous","piscivorous"),
146 topLevelColor=c("lightblue","red"))
147
148 ## plot a posterior probability distribution of states
149 colors<-setNames(c("lightblue","red"))
150 plot(summary(smap.trees), colors=colors, ftype="1",
151 topLevel=c("non-piscivorous","piscivorous"),
152 topLevelColor=c("lightblue","red"))
153
154 ## plot a posterior probability distribution of states
155 colors<-setNames(c("lightblue","red"))
156 plot(summary(smap.trees), colors=colors, ftype="1",
157 topLevel=c("non-piscivorous","piscivorous"),
158 topLevelColor=c("lightblue","red"))
159
160 ## plot a posterior probability distribution of states
161 colors<-setNames(c("lightblue","red"))
162 plot(summary(smap.trees), colors=colors, ftype="1",
163 topLevel=c("non-piscivorous","piscivorous"),
164 topLevelColor=c("lightblue","red"))
165
166 ## plot a posterior probability distribution of states
167 colors<-setNames(c("lightblue","red"))
168 plot(summary(smap.trees), colors=colors, ftype="1",
169 topLevel=c("non-piscivorous","piscivorous"),
170 topLevelColor=c("lightblue","red"))
171
172 ## plot a posterior probability distribution of states
173 colors<-setNames(c("lightblue","red"))
174 plot(summary(smap.trees), colors=colors, ftype="1",
175 topLevel=c("non-piscivorous","piscivorous"),
176 topLevelColor=c("lightblue","red"))
177
178 ## plot a posterior probability distribution of states
179 colors<-setNames(c("lightblue","red"))
180 plot(summary(smap.trees), colors=colors, ftype="1",
181 topLevel=c("non-piscivorous","piscivorous"),
182 topLevelColor=c("lightblue","red"))
183
184 ## plot a posterior probability distribution of states
185 colors<-setNames(c("lightblue","red"))
186 plot(summary(smap.trees), colors=colors, ftype="1",
187 topLevel=c("non-piscivorous","piscivorous"),
188 topLevelColor=c("lightblue","red"))
189
190 ## plot a posterior probability distribution of states
191 colors<-setNames(c("lightblue","red"))
192 plot(summary(smap.trees), colors=colors, ftype="1",
193 topLevel=c("non-piscivorous","piscivorous"),
194 topLevelColor=c("lightblue","red"))
195
196 ## plot a posterior probability distribution of states
197 colors<-setNames(c("lightblue","red"))
198 plot(summary(smap.trees), colors=colors, ftype="1",
199 topLevel=c("non-piscivorous","piscivorous"),
200 topLevelColor=c("lightblue","red"))
201
202 ## plot a posterior probability distribution of states
203 colors<-setNames(c("lightblue","red"))
204 plot(summary(smap.trees), colors=colors, ftype="1",
205 topLevel=c("non-piscivorous","piscivorous"),
206 topLevelColor=c("lightblue","red"))
207
208 ## plot a posterior probability distribution of states
209 colors<-setNames(c("lightblue","red"))
210 plot(summary(smap.trees), colors=colors, ftype="1",
211 topLevel=c("non-piscivorous","piscivorous"),
212 topLevelColor=c("lightblue","red"))
213
214 ## plot a posterior probability distribution of states
215 colors<-setNames(c("lightblue","red"))
216 plot(summary(smap.trees), colors=colors, ftype="1",
217 topLevel=c("non-piscivorous","piscivorous"),
218 topLevelColor=c("lightblue","red"))
219
220 ## plot a posterior probability distribution of states
221 colors<-setNames(c("lightblue","red"))
222 plot(summary(smap.trees), colors=colors, ftype="1",
223 topLevel=c("non-piscivorous","piscivorous"),
224 topLevelColor=c("lightblue","red"))
225
226 ## plot a posterior probability distribution of states
227 colors<-setNames(c("lightblue","red"))
228 plot(summary(smap.trees), colors=colors, ftype="1",
229 topLevel=c("non-piscivorous","piscivorous"),
230 topLevelColor=c("lightblue","red"))
231
232 ## plot a posterior probability distribution of states
233 colors<-setNames(c("lightblue","red"))
234 plot(summary(smap.trees), colors=colors, ftype="1",
235 topLevel=c("non-piscivorous","piscivorous"),
236 topLevelColor=c("lightblue","red"))
237
238 ## plot a posterior probability distribution of states
239 colors<-setNames(c("lightblue","red"))
240 plot(summary(smap.trees), colors=colors, ftype="1",
241 topLevel=c("non-piscivorous","piscivorous"),
242 topLevelColor=c("lightblue","red"))
243
244 ## plot a posterior probability distribution of states
245 colors<-setNames(c("lightblue","red"))
246 plot(summary(smap.trees), colors=colors, ftype="1",
247 topLevel=c("non-piscivorous","piscivorous"),
248 topLevelColor=c("lightblue","red"))
249
250 ## plot a posterior probability distribution of states
251 colors<-setNames(c("lightblue","red"))
252 plot(summary(smap.trees), colors=colors, ftype="1",
253 topLevel=c("non-piscivorous","piscivorous"),
254 topLevelColor=c("lightblue","red"))
255
256 ## plot a posterior probability distribution of states
257 colors<-setNames(c("lightblue","red"))
258 plot(summary(smap.trees), colors=colors, ftype="1",
259 topLevel=c("non-piscivorous","piscivorous"),
260 topLevelColor=c("lightblue","red"))
261
262 ## plot a posterior probability distribution of states
263 colors<-setNames(c("lightblue","red"))
264 plot(summary(smap.trees), colors=colors, ftype="1",
265 topLevel=c("non-piscivorous","piscivorous"),
266 topLevelColor=c("lightblue","red"))
267
268 ## plot a posterior probability distribution of states
269 colors<-setNames(c("lightblue","red"))
270 plot(summary(smap.trees), colors=colors, ftype="1",
271 topLevel=c("non-piscivorous","piscivorous"),
272 topLevelColor=c("lightblue","red"))
273
274 ## plot a posterior probability distribution of states
275 colors<-setNames(c("lightblue","red"))
276 plot(summary(smap.trees), colors=colors, ftype="1",
277 topLevel=c("non-piscivorous","piscivorous"),
278 topLevelColor=c("lightblue","red"))
279
280 ## plot a posterior probability distribution of states
281 colors<-setNames(c("lightblue","red"))
282 plot(summary(smap.trees), colors=colors, ftype="1",
283 topLevel=c("non-piscivorous","piscivorous"),
284 topLevelColor=c("lightblue","red"))
285
286 ## plot a posterior probability distribution of states
287 colors<-setNames(c("lightblue","red"))
288 plot(summary(smap.trees), colors=colors, ftype="1",
289 topLevel=c("non-piscivorous","piscivorous"),
290 topLevelColor=c("lightblue","red"))
291
292 ## plot a posterior probability distribution of states
293 colors<-setNames(c("lightblue","red"))
294 plot(summary(smap.trees), colors=colors, ftype="1",
295 topLevel=c("non-piscivorous","piscivorous"),
296 topLevelColor=c("lightblue","red"))
297
298 ## plot a posterior probability distribution of states
299 colors<-setNames(c("lightblue","red"))
300 plot(summary(smap.trees), colors=colors, ftype="1",
301 topLevel=c("non-piscivorous","piscivorous"),
302 topLevelColor=c("lightblue","red"))
303
304 ## plot a posterior probability distribution of states
305 colors<-setNames(c("lightblue","red"))
306 plot(summary(smap.trees), colors=colors, ftype="1",
307 topLevel=c("non-piscivorous","piscivorous"),
308 topLevelColor=c("lightblue","red"))
309
310 ## plot a posterior probability distribution of states
311 colors<-setNames(c("lightblue","red"))
312 plot(summary(smap.trees), colors=colors, ftype="1",
313 topLevel=c("non-piscivorous","piscivorous"),
314 topLevelColor=c("lightblue","red"))
315
316 ## plot a posterior probability distribution of states
317 colors<-setNames(c("lightblue","red"))
318 plot(summary(smap.trees), colors=colors, ftype="1",
319 topLevel=c("non-piscivorous","piscivorous"),
320 topLevelColor=c("lightblue","red"))
321
322 ## plot a posterior probability distribution of states
323 colors<-setNames(c("lightblue","red"))
324 plot(summary(smap.trees), colors=colors, ftype="1",
325 topLevel=c("non-piscivorous","piscivorous"),
326 topLevelColor=c("lightblue","red"))
327
328 ## plot a posterior probability distribution of states
329 colors<-setNames(c("lightblue","red"))
330 plot(summary(smap.trees), colors=colors, ftype="1",
331 topLevel=c("non-piscivorous","piscivorous"),
332 topLevelColor=c("lightblue","red"))
333
334 ## plot a posterior probability distribution of states
335 colors<-setNames(c("lightblue","red"))
336 plot(summary(smap.trees), colors=colors, ftype="1",
337 topLevel=c("non-piscivorous","piscivorous"),
338 topLevelColor=c("lightblue","red"))
339
340 ## plot a posterior probability distribution of states
341 colors<-setNames(c("lightblue","red"))
342 plot(summary(smap.trees), colors=colors, ftype="1",
343 topLevel=c("non-piscivorous","piscivorous"),
344 topLevelColor=c("lightblue","red"))
345
346 ## plot a posterior probability distribution of states
347 colors<-setNames(c("lightblue","red"))
348 plot(summary(smap.trees), colors=colors, ftype="1",
349 topLevel=c("non-piscivorous","piscivorous"),
350 topLevelColor=c("lightblue","red"))
351
352 ## plot a posterior probability distribution of states
353 colors<-setNames(c("lightblue","red"))
354 plot(summary(smap.trees), colors=colors, ftype="1",
355 topLevel=c("non-piscivorous","piscivorous"),
356 topLevelColor=c("lightblue","red"))
357
358 ## plot a posterior probability distribution of states
359 colors<-setNames(c("lightblue","red"))
360 plot(summary(smap.trees), colors=colors, ftype="1",
361 topLevel=c("non-piscivorous","piscivorous"),
362 topLevelColor=c("lightblue","red"))
363
364 ## plot a posterior probability distribution of states
365 colors<-setNames(c("lightblue","red"))
366 plot(summary(smap.trees), colors=colors, ftype="1",
367 topLevel=c("non-piscivorous","piscivorous"),
368 topLevelColor=c("lightblue","red"))
369
370 ## plot a posterior probability distribution of states
371 colors<-setNames(c("lightblue","red"))
372 plot(summary(smap.trees), colors=colors, ftype="1",
373 topLevel=c("non-piscivorous","piscivorous"),
374 topLevelColor=c("lightblue","red"))
375
376 ## plot a posterior probability distribution of states
377 colors<-setNames(c("lightblue","red"))
378 plot(summary(smap.trees), colors=colors, ftype="1",
379 topLevel=c("non-piscivorous","piscivorous"),
380 topLevelColor=c("lightblue","red"))
381
382 ## plot a posterior probability distribution of states
383 colors<-setNames(c("lightblue","red"))
384 plot(summary(smap.trees), colors=colors, ftype="1",
385 topLevel=c("non-piscivorous","piscivorous"),
386 topLevelColor=c("lightblue","red"))
387
388 ## plot a posterior probability distribution of states
389 colors<-setNames(c("lightblue","red"))
390 plot(summary(smap.trees), colors=colors, ftype="1",
391 topLevel=c("non-piscivorous","piscivorous"),
392 topLevelColor=c("lightblue","red"))
393
394 ## plot a posterior probability distribution of states
395 colors<-setNames(c("lightblue","red"))
396 plot(summary(smap.trees), colors=colors, ftype="1",
397 topLevel=c("non-piscivorous","piscivorous"),
398 topLevelColor=c("lightblue","red"))
399
399
```

```
Untitled1  
1 > source("c:/Users/11am/Desktop/R Scripts/Phylogenetic tree.R")  
2 > go to function... Addins...  
3 > Project (None)
```



Varanus salvator

● *Lepomis gibbosus*
● *Acantharchus pomotis*

plites cavifrons
plites rupestris
plites ariommus
ltes interruptus
vis nigromaculatus
vis annularis
lcanthus obesus
lurchus macropterus
lterus dolomieu
lterus punctulatus
lterus floridanus
lterus salmoides
lterus treculii
lterus notius
lterus coosae
lvis cyanellus
lvis symmetricus
lvis gulosus
lvis macrochirus
lvis humilis
lvis megalotis
lvis marginatus
lvis auritus
lvis miniatus
lvis punctatus
lvis microlophus

Console Terminal ↗ Back



Bitis gabonica

- *Lepomis gibbosus*
- *Acantharchus pomotis*

A large brown gecko with a long, slightly curved tail and four toes per limb. The gecko is resting on a light-colored, textured surface, possibly a wall or rock. Its body is elongated and slightly curved, with a pattern of darker brown spots and stripes. The gecko's head is turned slightly to the left, and its eyes are visible. The background is a plain, light-colored wall.

D. Martiré

Gehyra mutilata



```
Untitled1
```

```
13  "vucirne.Rnw"
14  data(sunfish)
15  data(sunFish)
16  ## extract df
17  fmode<-setNames
18  rownames(fmode)
19  ## do stochastic simulation
20  smap.trees<-
21  nsim=100
22  ## print a summary
23  summary(smap)
24  ## plot a posterior distribution
25  cols<-setNames
26  plot(summary(smap))
27  legend("topleft",c("nsim","n"))
27.1 (topLevel) <-
```

Console Terminal Background R 4.2.2 →

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
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
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259.1
259.2
259.3
259.4
259.5
259.6
259.7
259.8
259.9
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
279.1
279.2
280
281
282
283
284
285
286
287
288
289
289.1
289.2
290
291
292
293
294
295
296
297
298
299
299.1
299.2
299.3
299.4
299.5
299.6
299.7
299.8
299.9
300
301
302
303
304
305
306
307
308
309
309.1
309.2
310
311
312
313
314
315
316
317
318
319
319.1
319.2
320
321
322
323
324
325
326
327
328
329
329.1
329.2
330
331
332
333
334
335
336
337
338
339
339.1
339.2
340
341
342
343
344
345
346
347
348
349
349.1
349.2
350
351
352
353
354
355
356
357
358
359
359.1
359.2
360
361
362
363
364
365
366
367
368
369
369.1
369.2
370
371
372
373
374
375
376
377
378
379
379.1
379.2
380
381
382
383
384
385
386
387
388
389
389.1
389.2
390
391
392
393
394
395
396
397
398
399
399.1
399.2
400
401
402
403
404
405
406
407
408
409
409.1
409.2
410
411
412
413
414
415
416
417
418
419
419.1
420
421
422
423
424
425
426
427
428
429
429.1
429.2
430
431
432
433
434
435
436
437
438
439
439.1
440
441
442
443
444
445
446
447
448
449
449.1
450
451
452
453
454
455
456
457
458
459
459.1
460
461
462
463
464
465
466
467
468
469
469.1
470
471
472
473
474
475
476
477
478
479
479.1
480
481
482
483
484
485
486
487
488
489
489.1
490
491
492
493
494
495
496
497
498
499
499.1
500
501
502
503
504
505
506
507
508
509
509.1
510
511
512
513
514
515
516
517
518
519
519.1
520
521
522
523
524
525
526
527
528
529
529.1
530
531
532
533
534
535
536
537
538
539
539.1
540
541
542
543
544
545
546
547
548
549
549.1
550
551
552
553
554
555
556
557
558
559
559.1
560
561
562
563
564
565
566
567
568
569
569.1
570
571
572
573
574
575
576
577
578
579
579.1
580
581
582
583
584
585
586
587
588
589
589.1
590
591
592
593
594
595
596
597
598
599
599.1
600
601
602
603
604
605
606
607
608
609
609.1
610
611
612
613
614
615
616
617
618
619
619.1
620
621
622
623
624
625
626
627
628
629
629.1
630
631
632
633
634
635
636
637
638
639
639.1
640
641
642
643
644
645
646
647
648
649
649.1
650
651
652
653
654
655
656
657
658
659
659.1
660
661
662
663
664
665
666
667
668
669
669.1
670
671
672
673
674
675
676
677
678
679
679.1
680
681
682
683
684
685
686
687
688
689
689.1
690
691
692
693
694
695
696
697
698
699
699.1
700
701
702
703
704
705
706
707
708
709
709.1
710
711
712
713
714
715
716
717
718
719
719.1
720
721
722
723
724
725
726
727
728
729
729.1
730
731
732
733
734
735
736
737
738
739
739.1
740
741
742
743
744
745
746
747
748
749
749.1
750
751
752
753
754
755
756
757
758
759
759.1
760
761
762
763
764
765
766
767
768
769
769.1
770
771
772
773
774
775
776
777
778
779
779.1
780
781
782
783
784
785
786
787
788
789
789.1
790
791
792
793
794
795
796
797
798
799
799.1
800
801
802
803
804
805
806
807
808
809
809.1
810
811
812
813
814
815
816
817
818
819
819.1
820
821
822
823
824
825
826
827
828
829
829.1
830
831
832
833
834
835
836
837
838
839
839.1
840
841
842
843
844
845
846
847
848
849
849.1
850
851
852
853
854
855
856
857
858
859
859.1
860
861
862
863
864
865
866
867
868
869
869.1
870
871
872
873
874
875
876
877
878
879
879.1
880
881
882
883
884
885
886
887
888
889
889.1
890
891
892
893
894
895
896
897
898
899
899.1
900
901
902
903
904
905
906
907
908
909
909.1
910
911
912
913
914
915
916
917
918
919
919.1
920
921
922
923
924
925
926
927
928
929
929.1
930
931
932
933
934
935
936
937
938
939
939.1
940
941
942
943
944
945
946
947
948
949
949.1
950
951
952
953
954
955
956
957
958
959
959.1
960
961
962
963
964
965
966
967
968
969
969.1
970
971
972
973
974
975
976
977
978
979
979.1
980
981
982
983
984
985
986
987
988
989
989.1
990
991
992
993
994
995
996
997
997.1
998
998.1
999
999.1
1000
1000.1
1000.2
1000.3
1000.4
1000.5
1000.6
1000.7
1000.8
1000.9
1000.10
1000.11
1000.12
1000.13
1000.14
1000.15
1000.16
1000.17
1000.18
1000.19
1000.20
1000.21
1000.22
1000.23
1000.24
1000.25
1000.26
1000.27
1000.28
1000.29
1000.30
1000.31
1000.32
1000.33
1000.34
1000.35
1000.36
1000.37
1000.38
1000.39
1000.40
1000.41
1000.42
1000.43
1000.44
1000.45
1000.46
1000.47
1000.48
1000.49
1000.50
1000.51
1000.52
1000.53
1000.54
1000.55
1000.56
1000.57
1000.58
1000.59
1000.60
1000.61
1000.62
1000.63
1000.64
1000.65
1000.66
1000.67
1000.68
1000.69
1000.70
1000.71
1000.72
1000.73
1000.74
1000.75
1000.76
1000.77
1000.78
1000.79
1000.80
1000.81
1000.82
1000.83
1000.84
1000.85
1000.86
1000.87
1000.88
1000.89
1000.90
1000.91
1000.92
1000.93
1000.94
1000.95
1000.96
1000.97
1000.98
1000.99
1000.100
1000.101
1000.102
1000.103
1000.104
1000.105
1000.106
1000.107
1000.108
1000.109
1000.110
1000.111
1000.112
1000.113
1000.114
1000.115
1000.116
1000.117
1000.118
1000.119
1000.120
1000.121
1000.122
1000.123
1000.124
1000.125
1000.126
1000.127
1000.128
1000.129
1000.130
1000.131
1000.132
1000.133
1000.134
1000.135
1000.136
1000.137
1000.138
1000.139
1000.140
1000.141
1000.142
1000.143
1000.144
1000.145
1000.146
1000.147
1000.148
1000.149
1000.150
1000.151
1000.152
1000.153
1000.154
1000.155
1000.156
1000.157
1000.158
1000.159
1000.160
1000.161
1000.162
1000.163
1000.164
1000.165
1000.166
1000.167
1000.168
1000.169
1000.170
1000.171
1000.172
1000.173
1000.174
1000.175
1000.176
1000.177
1000.178
1000.179
1000.180
1000.181
1000.182
1000.183
1000.184
1000.185
1000.186
1000.187
1000.188
1000.189
1000.190
1000.191
1000.192
1000.193
1000.194
1000.195
1000.196
1000.197
1000.198
1000.199
1000.199.1
1000.199.2
1000.199.3
1000.199.4
1000.199.5
1000.199.6
1000.199.7
1000.199.8
1000.199.9
1000.199.10
1000.199.11
1000.199.12
1000.199.13
1000.199.14
1000.199.15
1000.199.16
1000.199.17
1000.199.18
1000.199.19
1000.199.20
1000.199.21
1000.199.22
1000.199.23
1000.199.24
1000.199.25
1000.199.26
1000.199.27
1000.199.28
1000.199.29
1000.199.30
1000.199.31
1000.199.32
1000.199.33
1000.199.34
1000.199.35
1000.199.36
1000.199.37
1000.199.38
1000.199.39
1000.199.40
1000.199.41
1000.199.42
1000.199.43
1000.199.44
1000.199.45
1000.199.46
1000.199.47
1000.199.48
1000.199.49
1000.199.50
1000.199.51
1000.199.52
1000.199.53
1000.199.54
1000.199.55
1000.199.56
1000.199.57
1000.199.58
1000.199.59
1000.199.60
1000.199.61
1000.199.62
1000.199.63
1000.199.64
1000.199.65
1000.199.66
1000.199.67
1000.199.68
1000.199.69
1000.199.70
1000.199.71
1000.199.72
1000.199.73
1000.199.74
1000.199.75
1000.199.76
1000.199.77
1000.199.78
1000.199.79
1000.199.80
1000.199.81
1000.199.82
1000.199.83
1000.199.84
1000.199.85
1000.199.86
1000.199.87
1000.199.88
1000.199.89
1000.199.90
1000.199.91
1000.199.92
1000.199.93
1000.199.94
1000.199.95
1000.199.96
1000.199.97
1000.199.98
1000.199.99
1000.199.100
1000.199.101
1000.199.102
1000.199.103
1000.199.104
1000.199.105
1000.199.106
1000.199.107
1000.199.108
1000.199.109
1000.199.110
1000.199.111
1000.199.112
1000.199.113
1000.199.114
1000.199.115
1000.199.116
1000.199.117
1000.199.118
1000.199.119
1000.199.120
1000.199.121
1000.199.122
1000.199.123
1000.199.124
1000.199.125
1000.199.126
1000.199.127
1000.199.128
1000.199.129
1000.199.130
1000.199.131
1000.199.132
1000.199.133
1000.199.134
1000.199.135
1000.199.136
1000.199.137
1000.199.138
1000.199.139
1000.199.140
1000.199.141
1000.199.142
1000.199.143
1000.199.144
1000.199.145
1000.199.146
1000.199.147
1000.199.148
1000.199.149
1000.199.150
1000.199.151
1000.199.152
1000.199.153
1000.199.154
1000.199.155
1000.199.156
1000.199.157
1000.199.158
1000.199.159
1000.199.160
1000.199.161
1000.199.162
1000.199.163
1000.199.164
1000.199.165
1000.199.166
1000.199.167
1000.199.168
1000.199.169
1000.199.170
1000.199.171
1000.199.172
1000.199.173
1000.199.174
1000.199.175
1000.199.176
1000.199.177
1000.199.178
1000.199.179
1000.199.180
1000.199.181
1000.199.182
1000.199.183
1000.199.184
1000.199.185
1000.199.186
1000.199.187
1000.199.188
1000.199.189
1000.199.190
1000.199.191
1000.199.192
1000.199.193
1000.199.194
1000.199.195
1000.199.196
1000.199.197
1000.199.198
1000.199.199
1000.199.200
1000.199.201
1000.199.202
1000.199.203
1000.199.204
1000.199.205
1000.199.206
1000.199.207
1000.199.208
1000.199.209
1000.199.210
1000.199.211
1000.199.212
1000.199.213
1000.199.214
1000.199.215
1000.199.216
1000.199.217
1000.199.218
1000.199.219
1000.199.220
1000.199.221
1000.199.222
1000.199.223
1000.199.224
1000.199.225
1000.199.226
1000.199.227
1000.199.228
1000.199.229
1000.199.230
1000.199.231
1000.199.232
1000.199.233
1000.199.234
1000.199.235
1000.199.236
1000.199.237
1000.199.238
1000.199.239
1000.199.240
1000.199.241
1000.199.242
1000.199.243
1000.199.244
1000.199.245
1000.199.246
1000.199.247
1000.199.248
1000.199.249
1000.199.250
1000.199.251
1000.199.252
1000.199.253
1000.199.254
1000.199.255
1000.199.256
1000.199.257
1000.199.258
1000.199.259
1000.199.260
1000.199.261
1000.199.262
1000.199.263
1000.199.264
1000.199.265
1000.199.266
1000.199.267
1000.199.268
1000.199.269
1000.199.270
1000.199.271
1000.199.272
1000.199.273
1000.199.274
1000.199.275
1000.199.276
1000.199.277
1000.199.278
1000.199.279
1000.199.280
1000.199.281
1000.199.282
1000.199.283
1000.199.284
1000.199.285
1000.199.286
1000.199.287
1000.199.288
1000.199.289
1000.199.290
1000.199.291
1000.199.292
1000.199.293
1000.199.294
1000.199.295
1000.199.296
1000.199.297
1000.199.298
1000.199.299
1000.199.300
1000.199.301
1000.199.302
1000.199.303
1000.199.304
1000.199.305
1000.199.306
1000.199.307
1000.199.308
1000.199.309
1000.199.310
1000.199.311
1000.199.312
1000.199.313
1000.199.314
1000.199.315
1000.199.316
1000.199.317
1000.199.318
1000.199.319
1000.199.320
1000.199.321
1000.199.322
1000.199.323
1000.199.324
1000.199.325
1000.199.326
1000.199.327
1000.199.328
1000.199.329
1000.199.330
1000.199.331
1000.199.332
1000.199.333
1000.199.334
1000.199.335
1000.199.336
1000.199.337
1000.199.338
1000.199.339
1000.199.340
1000.199.341
1000.199.342
1000.199.343
1000.199.344
1000.199.345
1000.199.346
1000.199.347
1000.199.348
1000.199.349
1000.199.350
1000.199.351
1000.199.352
1000.199.353
1000.199.354
1000.199.355
1000.199.356
1000.199.357
1000.199.358
100
```

```
File Edit Code View Plots Session Build Debug Profile Tools Help  
Untitled1  
Source  
13  "uci.rme.R"  
14  data(sunfish)  
15  data(sunfish)  
16  ## extract diff  
17  fmode<-setNames  
18  rownames  
19  ## do stochastic  
20  smap.trees<-m  
21  nsim<100  
22  ## print a sum  
23  summary(smap.  
24  ## plot a pos  
25  cols<-setNames  
26  plot(summary  
27  Legend("tople  
28  ->  
27.1 (topLevel) <-  
Console Terminal Back  
R 4.2.2 -->  
x  
y  
p  
T  
d  
1  
t  
f  
f  
f  
f  
f  
f  
f  
f  
f  
<bytecode: 0x0000002  
<environment: names  
> getwd()  
[1] "C:/Users/l1amj/Desktop/R/phylogenetic analysis"  
> plot(summary(smap.trees),colors=cols,ftype="1")  
● Lepomis gibbosus  
● Acantharchus pomotis
```

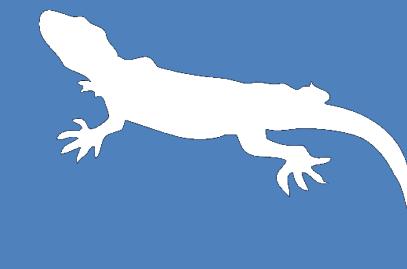
Plestiodon fasciatus

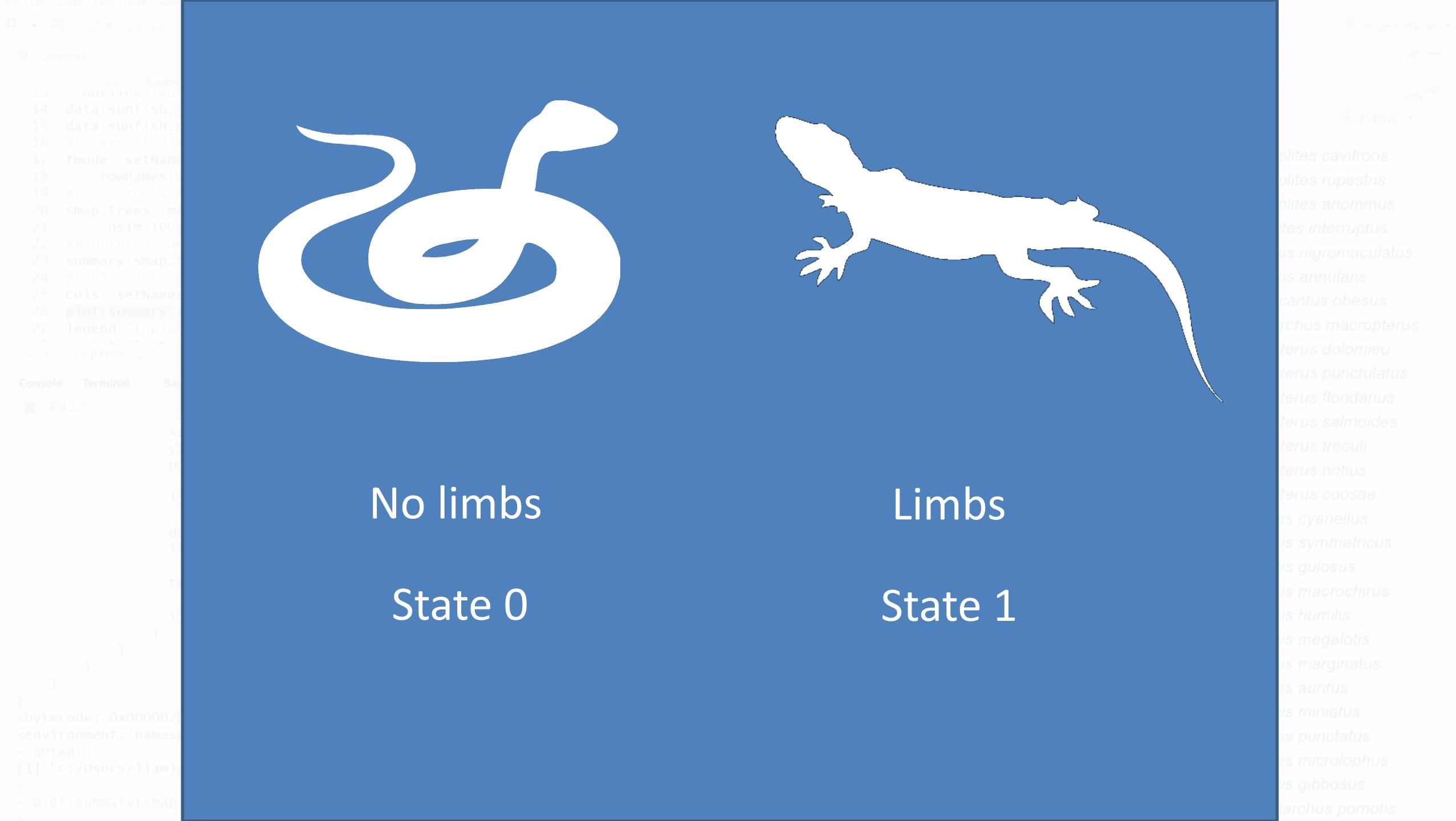
**Armand Kok
2009**



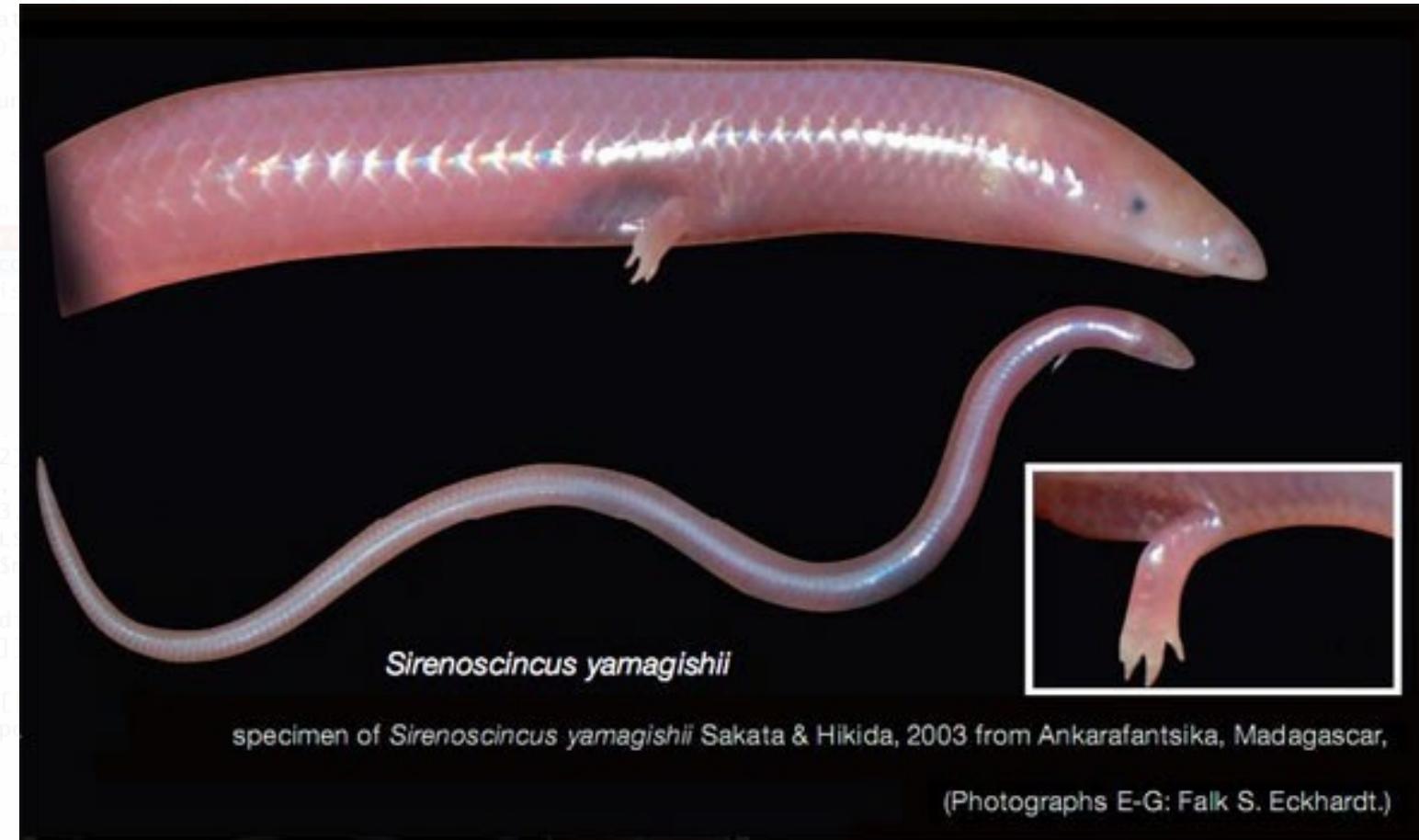
Acontias meleagris

How do we model the process by which squamates (snakes and lizards) lose their limbs?





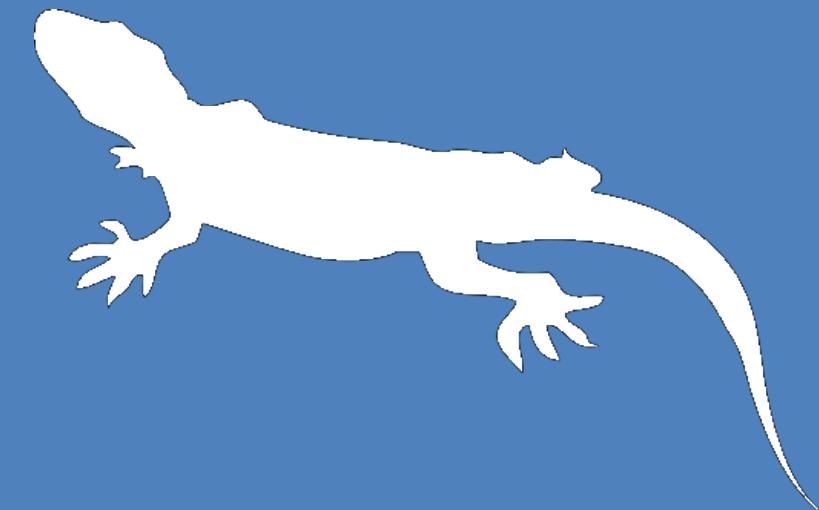
```
13 ## Load source on Save outcome (RUC)
14 data(sunfish.tree)
15 data(sunfish.data)
16 ## extract discrete character breeding mode
17 fmode<-setNames(sunfish.da
18 rownames(sunfish.data)
19 ## do stochastic mapping
20 smap.trees<-make.simmap(sunfish,
21 nsim=100)
22 ## print a summary of the trees
23 summary(smap.trees)
24 ## plot a posterior probability map
25 cols<-setNames(c("blue","red"))
26 plot(summary(smap.trees),cols=cols)
27 legend("topleft",c("non-pi
28 (Top Level) <
29
```





No limbs

State O

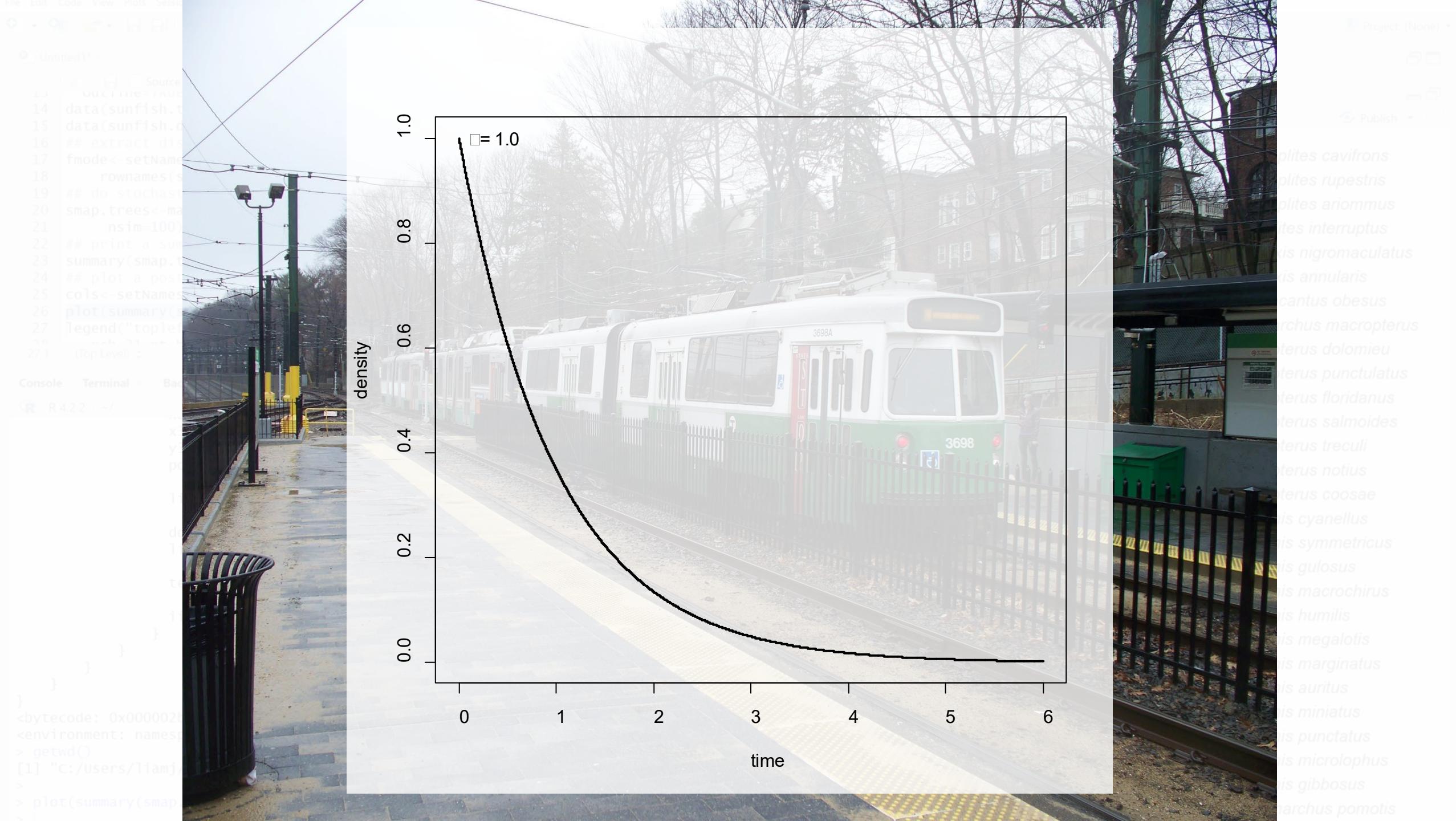


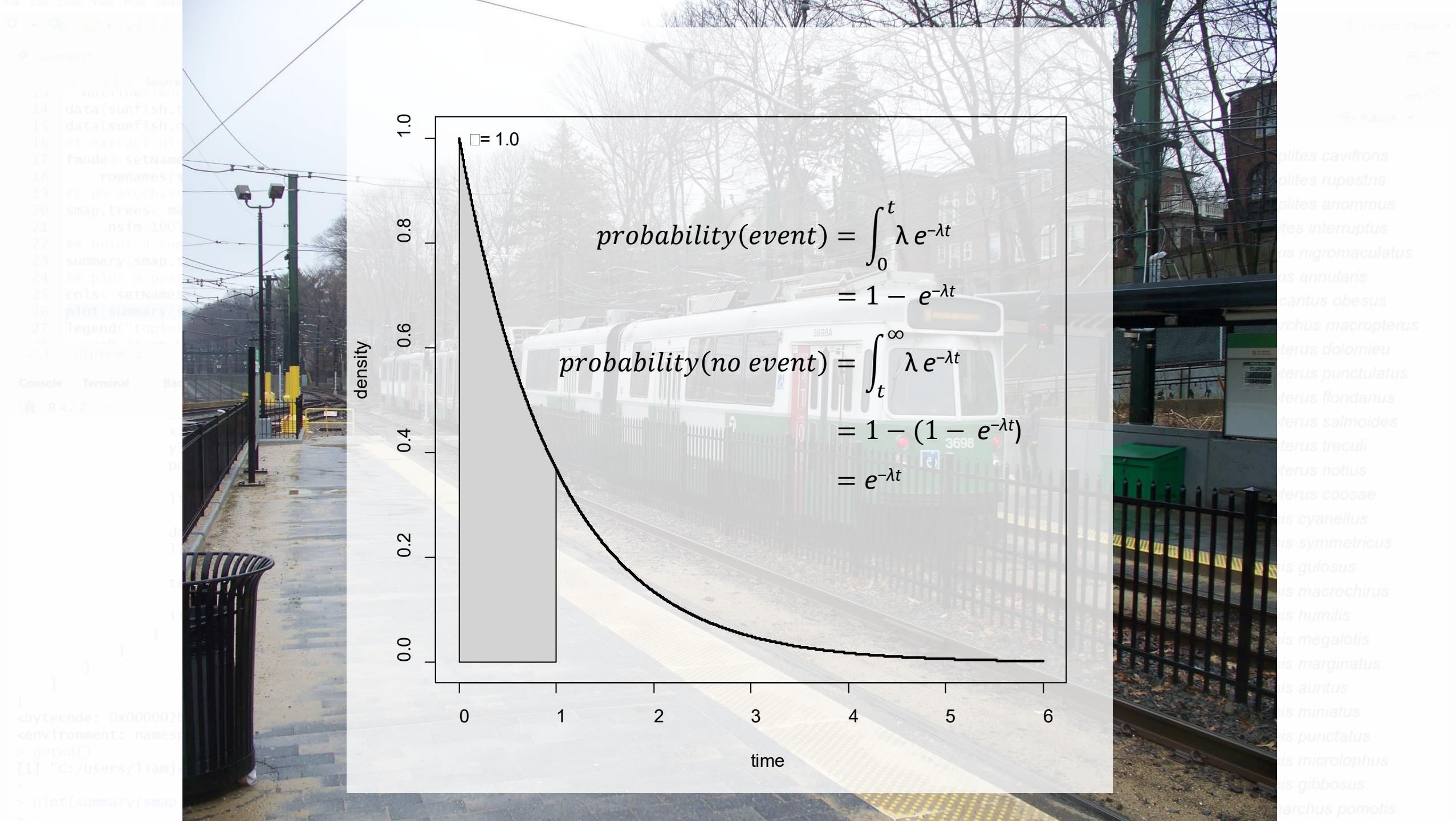
Limbs

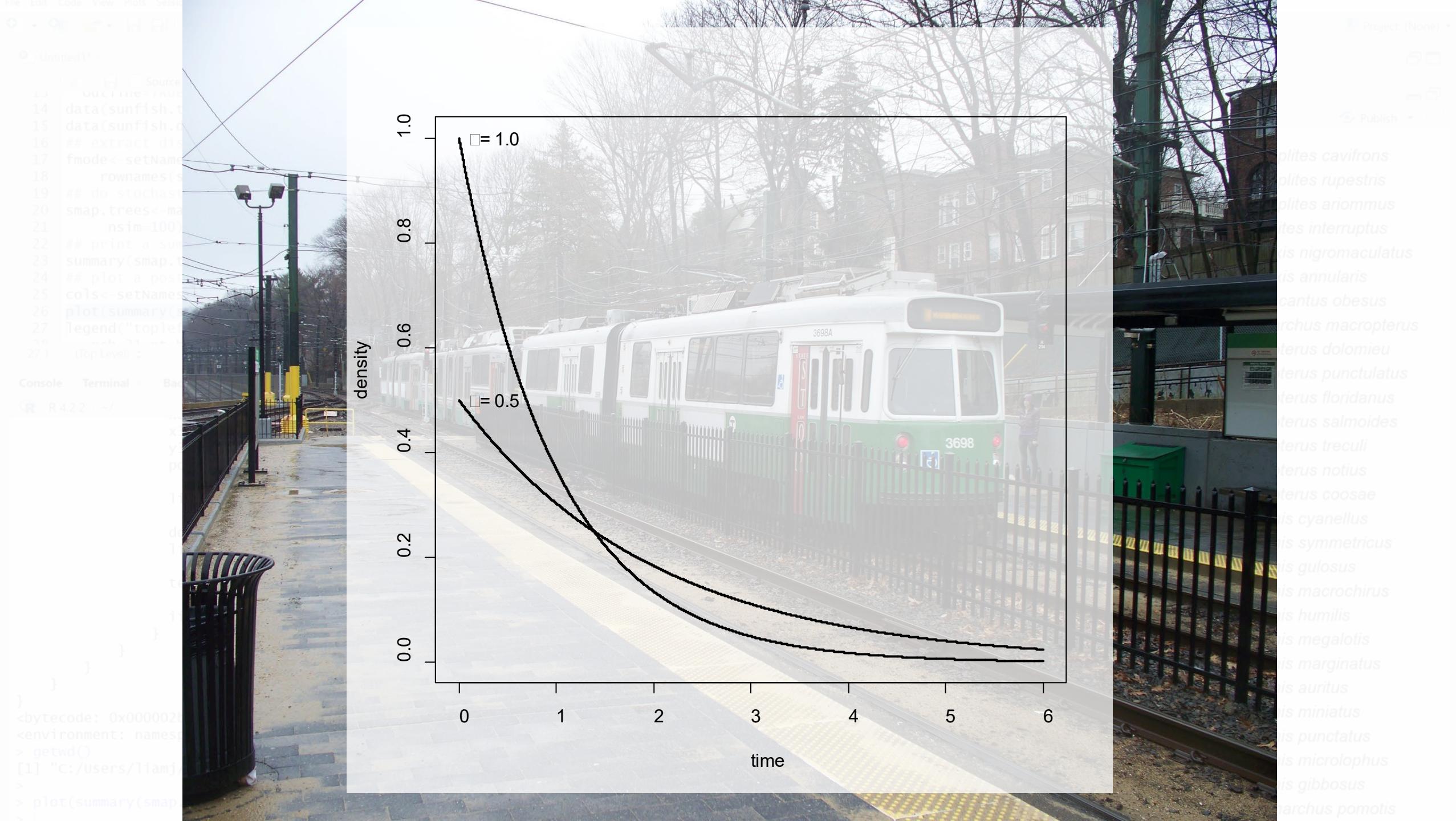
State 1

Discrete character evolution

- Mk model: continuous-time discrete-state Markov process.
- Changes between states occur with instantaneous rate q .
- After some time t the probability that a change has occurred is calculated by computing the integral of an exponential distribution







- There is an important way in which waiting for the train differs from waiting for character change.
- That is, a train that has arrived can't 'un-arrive.' By contrast, a character that has changed from 0 to 1, can change back to 0 before time t has elapsed.
- To take that into consideration, we compute:

$$P[0,1] = \frac{1}{2} - \frac{1}{2}e^{-2qt}$$

$$P[0,0] = \frac{1}{2} + \frac{1}{2}e^{-2qt}$$

for the probabilities that the character has or has not changed in state after time t , respectively (in which q is substituted for λ).



$$P(r, b) = P(b|r)_1 P(r|r)_2 + P(b|b)_1 P(r|b)_2$$

* assuming r & b are equiprobable at the root.

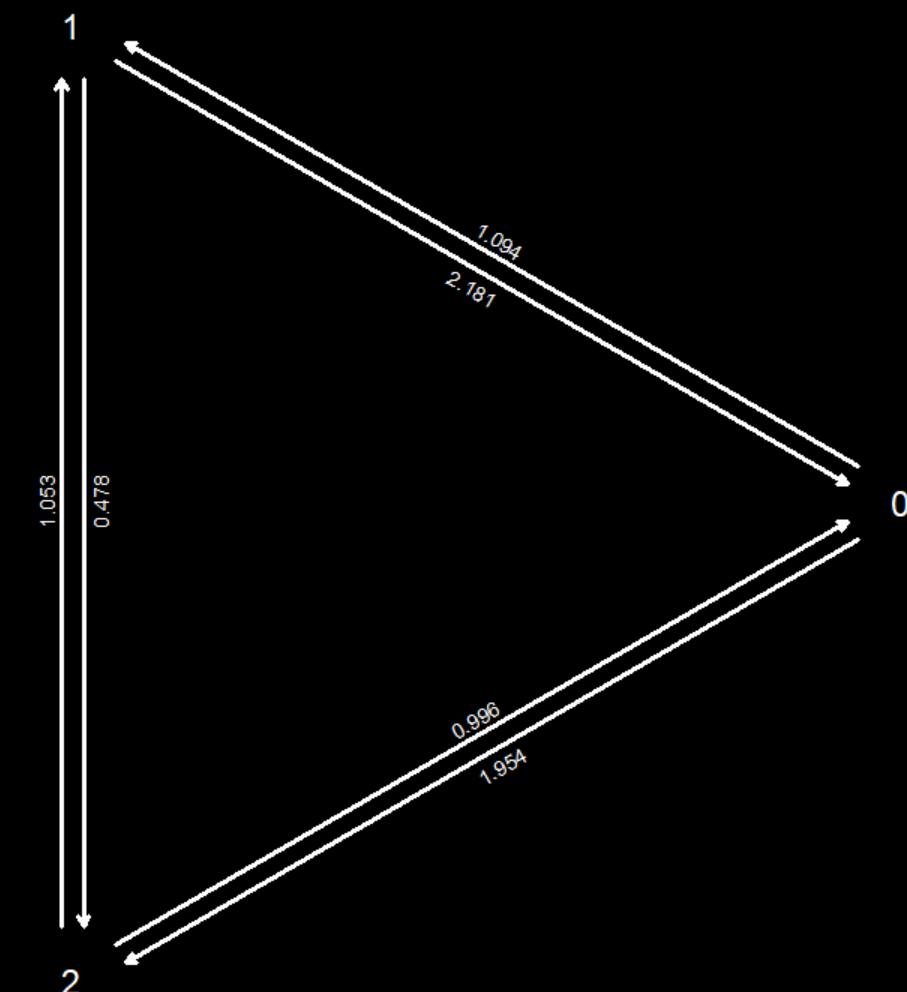
$$Q = \begin{bmatrix} -q_{01} & q_{01} \\ q_{10} & -q_{10} \end{bmatrix}$$

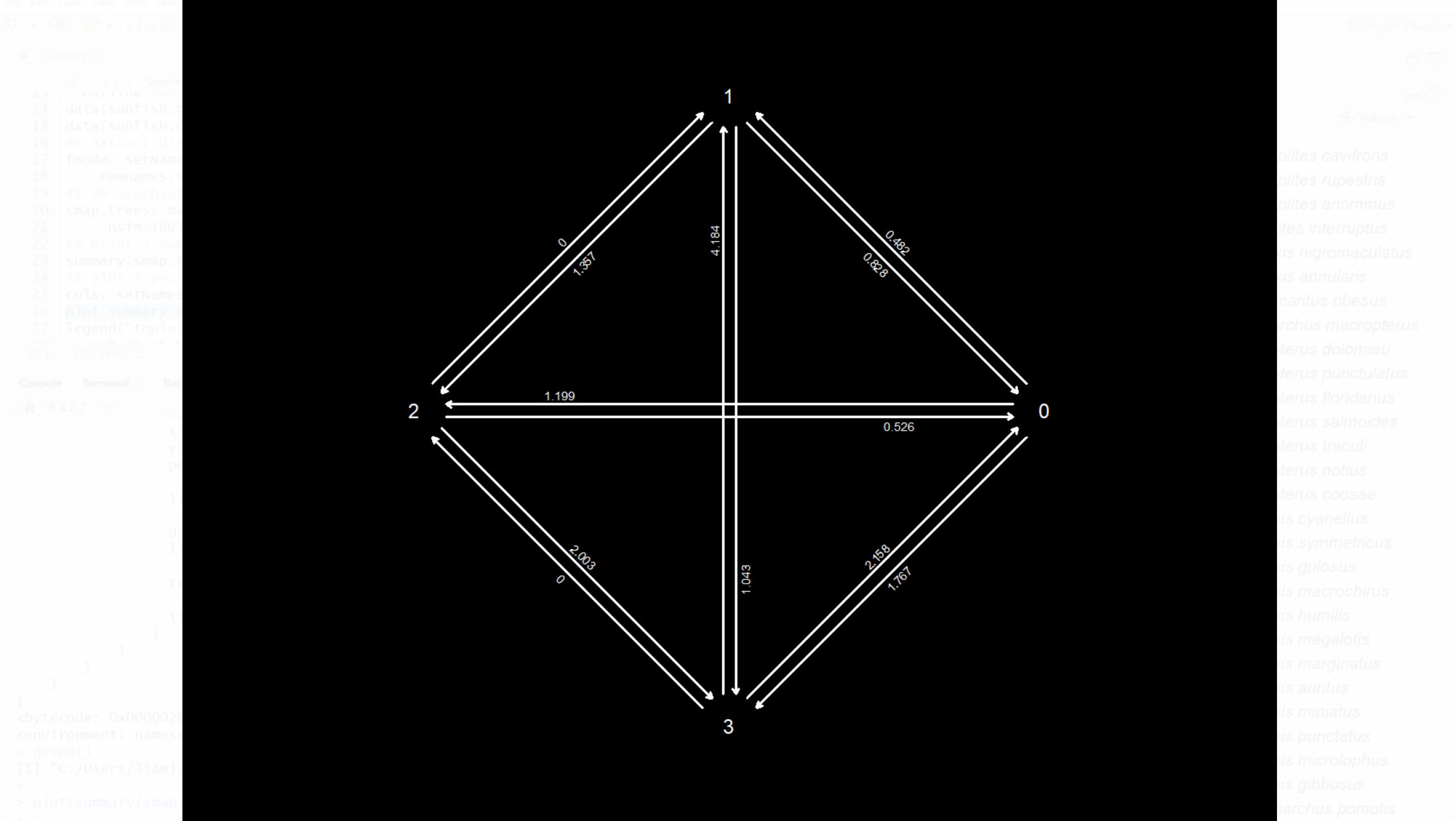
$$P(t) = \exp(Qt)$$



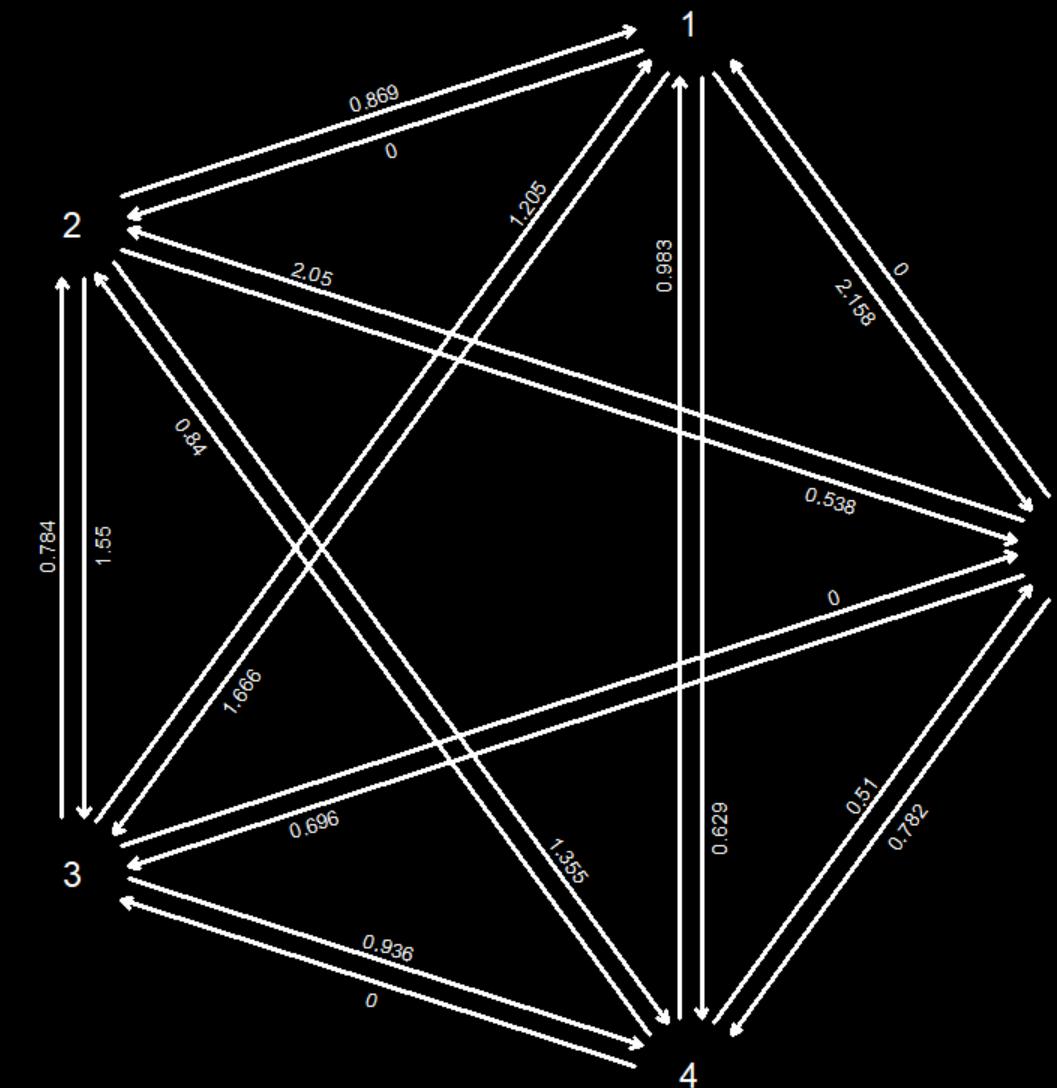
$$Q = \begin{bmatrix} -(q_{01} + q_{02}) & q_{01} & q_{02} \\ q_{10} & -(q_{10} + q_{12}) & q_{12} \\ q_{20} & q_{21} & -(q_{20} + q_{21}) \end{bmatrix}$$

$$P(t) = \exp(Qt)$$



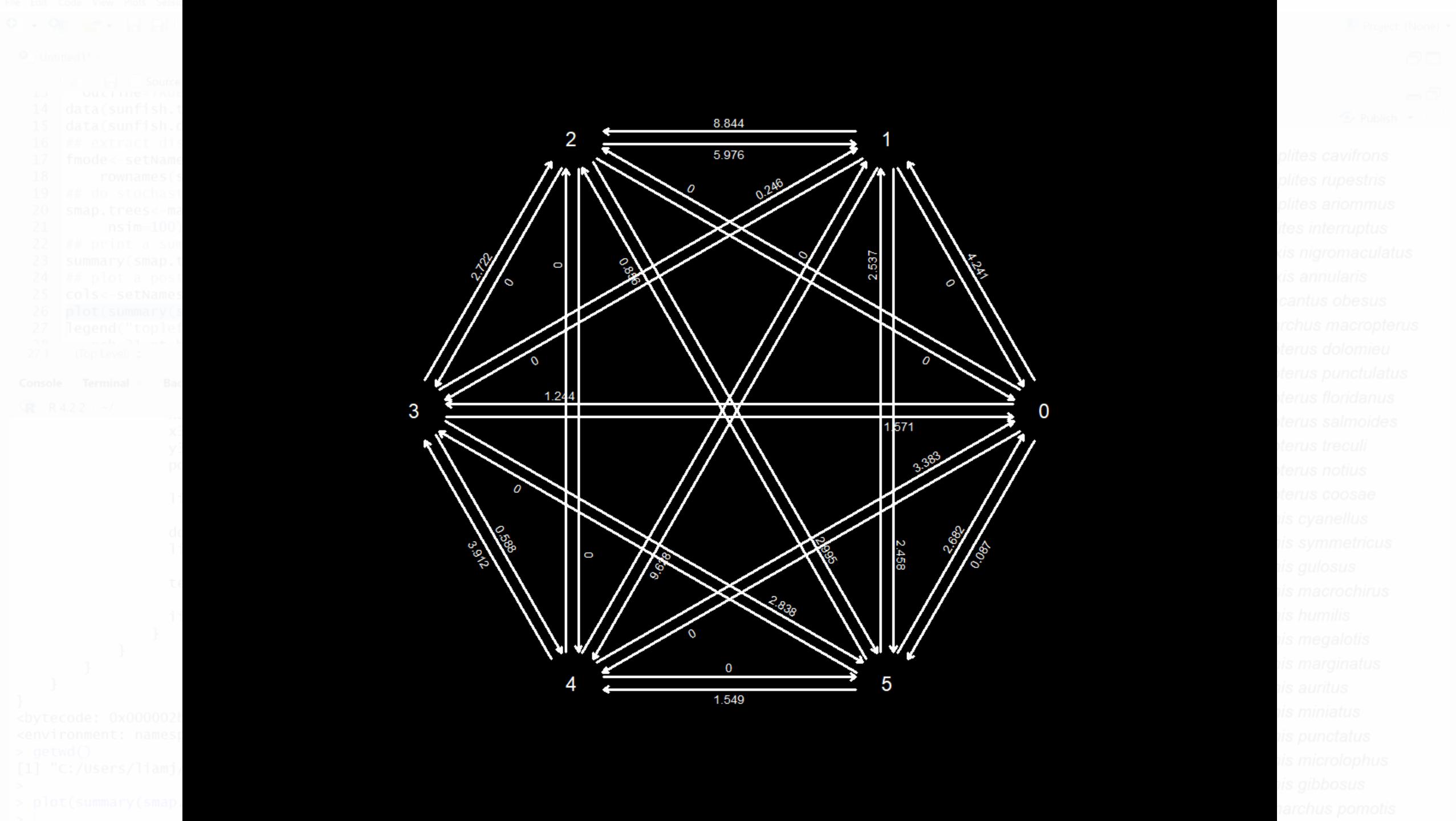


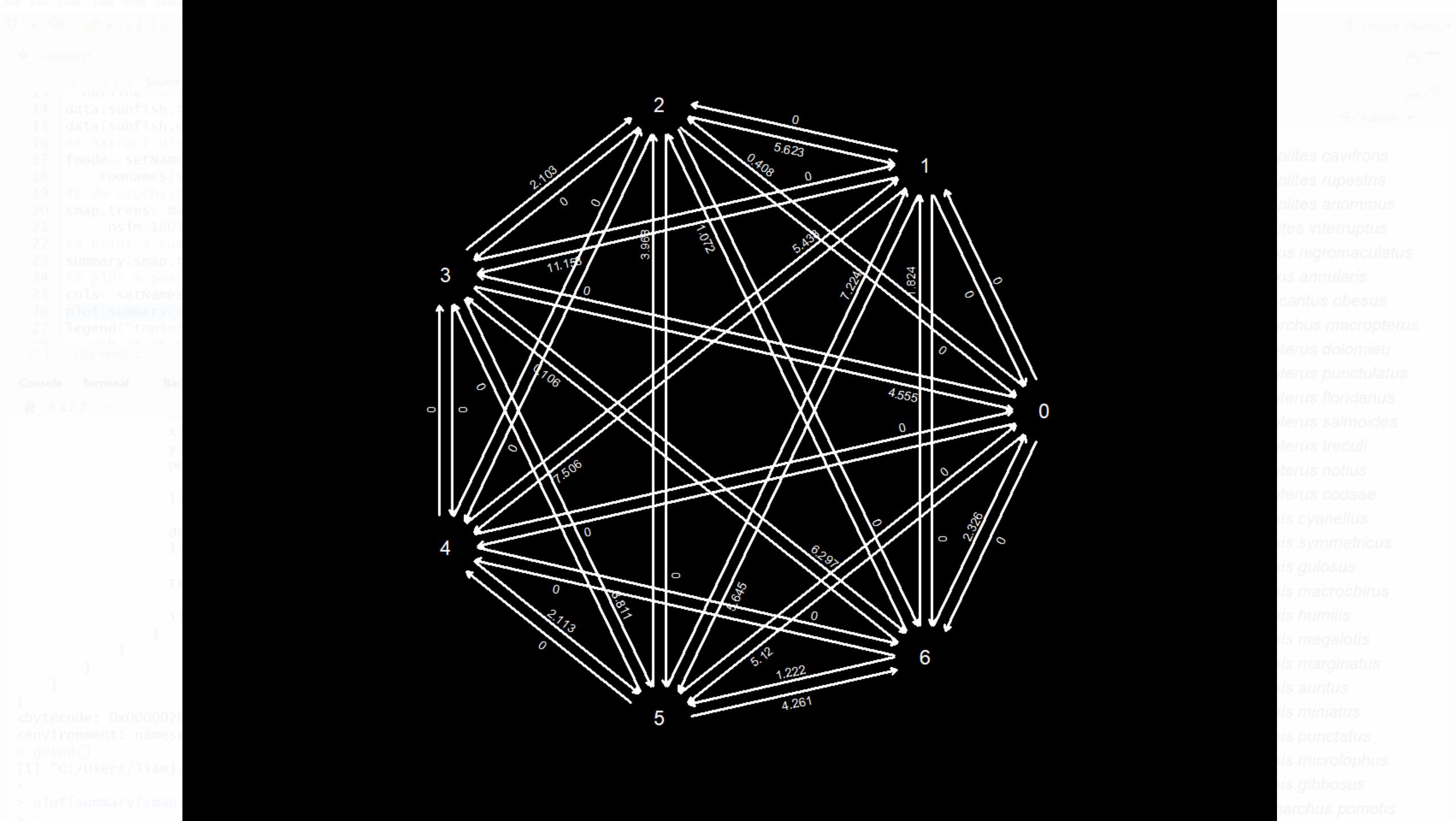
```
File Edit Code View Plot Help  
Untitled1  
1 > ## Source  
2 > "vaccine.RW"  
3 > library(RW)  
4 > data(sunfish)  
5 > data(sunfish)  
6 > ## extract df  
7 > fmode<-setNames  
8 > rownames(fmode)  
9 > ## do stochastic  
10 > smap.trees<-m  
11 > nsim=100  
12 > ## print a su  
13 > summary(smap.  
14 > ## plot a pos  
15 > cols<-setNames  
16 > plot(summary(s  
17 > Legend("tople  
18 > ncol=3, na  
19 > (topLevel <-  
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 >  
45 >  
46 >  
47 >  
48 >  
49 >  
50 >  
51 >  
52 >  
53 >  
54 >  
55 >  
56 >  
57 >  
58 >  
59 >  
60 >  
61 >  
62 >  
63 >  
64 >  
65 >  
66 >  
67 >  
68 >  
69 >  
70 >  
71 >  
72 >  
73 >  
74 >  
75 >  
76 >  
77 >  
78 >  
79 >  
80 >  
81 >  
82 >  
83 >  
84 >  
85 >  
86 >  
87 >  
88 >  
89 >  
90 >  
91 >  
92 >  
93 >  
94 >  
95 >  
96 >  
97 >  
98 >  
99 >  
100 >  
101 >  
102 >  
103 >  
104 >  
105 >  
106 >  
107 >  
108 >  
109 >  
110 >  
111 >  
112 >  
113 >  
114 >  
115 >  
116 >  
117 >  
118 >  
119 >  
120 >  
121 >  
122 >  
123 >  
124 >  
125 >  
126 >  
127 >  
128 >  
129 >  
130 >  
131 >  
132 >  
133 >  
134 >  
135 >  
136 >  
137 >  
138 >  
139 >  
140 >  
141 >  
142 >  
143 >  
144 >  
145 >  
146 >  
147 >  
148 >  
149 >  
150 >  
151 >  
152 >  
153 >  
154 >  
155 >  
156 >  
157 >  
158 >  
159 >  
160 >  
161 >  
162 >  
163 >  
164 >  
165 >  
166 >  
167 >  
168 >  
169 >  
170 >  
171 >  
172 >  
173 >  
174 >  
175 >  
176 >  
177 >  
178 >  
179 >  
180 >  
181 >  
182 >  
183 >  
184 >  
185 >  
186 >  
187 >  
188 >  
189 >  
190 >  
191 >  
192 >  
193 >  
194 >  
195 >  
196 >  
197 >  
198 >  
199 >  
200 >  
201 >  
202 >  
203 >  
204 >  
205 >  
206 >  
207 >  
208 >  
209 >  
210 >  
211 >  
212 >  
213 >  
214 >  
215 >  
216 >  
217 >  
218 >  
219 >  
220 >  
221 >  
222 >  
223 >  
224 >  
225 >  
226 >  
227 >  
228 >  
229 >  
230 >  
231 >  
232 >  
233 >  
234 >  
235 >  
236 >  
237 >  
238 >  
239 >  
240 >  
241 >  
242 >  
243 >  
244 >  
245 >  
246 >  
247 >  
248 >  
249 >  
250 >  
251 >  
252 >  
253 >  
254 >  
255 >  
256 >  
257 >  
258 >  
259 >  
260 >  
261 >  
262 >  
263 >  
264 >  
265 >  
266 >  
267 >  
268 >  
269 >  
270 >  
271 >  
272 >  
273 >  
274 >  
275 >  
276 >  
277 >  
278 >  
279 >  
280 >  
281 >  
282 >  
283 >  
284 >  
285 >  
286 >  
287 >  
288 >  
289 >  
290 >  
291 >  
292 >  
293 >  
294 >  
295 >  
296 >  
297 >  
298 >  
299 >  
300 >  
301 >  
302 >  
303 >  
304 >  
305 >  
306 >  
307 >  
308 >  
309 >  
310 >  
311 >  
312 >  
313 >  
314 >  
315 >  
316 >  
317 >  
318 >  
319 >  
320 >  
321 >  
322 >  
323 >  
324 >  
325 >  
326 >  
327 >  
328 >  
329 >  
330 >  
331 >  
332 >  
333 >  
334 >  
335 >  
336 >  
337 >  
338 >  
339 >  
340 >  
341 >  
342 >  
343 >  
344 >  
345 >  
346 >  
347 >  
348 >  
349 >  
350 >  
351 >  
352 >  
353 >  
354 >  
355 >  
356 >  
357 >  
358 >  
359 >  
360 >  
361 >  
362 >  
363 >  
364 >  
365 >  
366 >  
367 >  
368 >  
369 >  
370 >  
371 >  
372 >  
373 >  
374 >  
375 >  
376 >  
377 >  
378 >  
379 >  
380 >  
381 >  
382 >  
383 >  
384 >  
385 >  
386 >  
387 >  
388 >  
389 >  
390 >  
391 >  
392 >  
393 >  
394 >  
395 >  
396 >  
397 >  
398 >  
399 >  
400 >  
401 >  
402 >  
403 >  
404 >  
405 >  
406 >  
407 >  
408 >  
409 >  
410 >  
411 >  
412 >  
413 >  
414 >  
415 >  
416 >  
417 >  
418 >  
419 >  
420 >  
421 >  
422 >  
423 >  
424 >  
425 >  
426 >  
427 >  
428 >  
429 >  
430 >  
431 >  
432 >  
433 >  
434 >  
435 >  
436 >  
437 >  
438 >  
439 >  
440 >  
441 >  
442 >  
443 >  
444 >  
445 >  
446 >  
447 >  
448 >  
449 >  
450 >  
451 >  
452 >  
453 >  
454 >  
455 >  
456 >  
457 >  
458 >  
459 >  
460 >  
461 >  
462 >  
463 >  
464 >  
465 >  
466 >  
467 >  
468 >  
469 >  
470 >  
471 >  
472 >  
473 >  
474 >  
475 >  
476 >  
477 >  
478 >  
479 >  
480 >  
481 >  
482 >  
483 >  
484 >  
485 >  
486 >  
487 >  
488 >  
489 >  
490 >  
491 >  
492 >  
493 >  
494 >  
495 >  
496 >  
497 >  
498 >  
499 >  
500 >  
501 >  
502 >  
503 >  
504 >  
505 >  
506 >  
507 >  
508 >  
509 >  
510 >  
511 >  
512 >  
513 >  
514 >  
515 >  
516 >  
517 >  
518 >  
519 >  
520 >  
521 >  
522 >  
523 >  
524 >  
525 >  
526 >  
527 >  
528 >  
529 >  
530 >  
531 >  
532 >  
533 >  
534 >  
535 >  
536 >  
537 >  
538 >  
539 >  
540 >  
541 >  
542 >  
543 >  
544 >  
545 >  
546 >  
547 >  
548 >  
549 >  
550 >  
551 >  
552 >  
553 >  
554 >  
555 >  
556 >  
557 >  
558 >  
559 >  
560 >  
561 >  
562 >  
563 >  
564 >  
565 >  
566 >  
567 >  
568 >  
569 >  
570 >  
571 >  
572 >  
573 >  
574 >  
575 >  
576 >  
577 >  
578 >  
579 >  
580 >  
581 >  
582 >  
583 >  
584 >  
585 >  
586 >  
587 >  
588 >  
589 >  
590 >  
591 >  
592 >  
593 >  
594 >  
595 >  
596 >  
597 >  
598 >  
599 >  
600 >  
601 >  
602 >  
603 >  
604 >  
605 >  
606 >  
607 >  
608 >  
609 >  
610 >  
611 >  
612 >  
613 >  
614 >  
615 >  
616 >  
617 >  
618 >  
619 >  
620 >  
621 >  
622 >  
623 >  
624 >  
625 >  
626 >  
627 >  
628 >  
629 >  
630 >  
631 >  
632 >  
633 >  
634 >  
635 >  
636 >  
637 >  
638 >  
639 >  
640 >  
641 >  
642 >  
643 >  
644 >  
645 >  
646 >  
647 >  
648 >  
649 >  
650 >  
651 >  
652 >  
653 >  
654 >  
655 >  
656 >  
657 >  
658 >  
659 >  
660 >  
661 >  
662 >  
663 >  
664 >  
665 >  
666 >  
667 >  
668 >  
669 >  
670 >  
671 >  
672 >  
673 >  
674 >  
675 >  
676 >  
677 >  
678 >  
679 >  
680 >  
681 >  
682 >  
683 >  
684 >  
685 >  
686 >  
687 >  
688 >  
689 >  
690 >  
691 >  
692 >  
693 >  
694 >  
695 >  
696 >  
697 >  
698 >  
699 >  
700 >  
701 >  
702 >  
703 >  
704 >  
705 >  
706 >  
707 >  
708 >  
709 >  
710 >  
711 >  
712 >  
713 >  
714 >  
715 >  
716 >  
717 >  
718 >  
719 >  
720 >  
721 >  
722 >  
723 >  
724 >  
725 >  
726 >  
727 >  
728 >  
729 >  
730 >  
731 >  
732 >  
733 >  
734 >  
735 >  
736 >  
737 >  
738 >  
739 >  
740 >  
741 >  
742 >  
743 >  
744 >  
745 >  
746 >  
747 >  
748 >  
749 >  
750 >  
751 >  
752 >  
753 >  
754 >  
755 >  
756 >  
757 >  
758 >  
759 >  
760 >  
761 >  
762 >  
763 >  
764 >  
765 >  
766 >  
767 >  
768 >  
769 >  
770 >  
771 >  
772 >  
773 >  
774 >  
775 >  
776 >  
777 >  
778 >  
779 >  
780 >  
781 >  
782 >  
783 >  
784 >  
785 >  
786 >  
787 >  
788 >  
789 >  
790 >  
791 >  
792 >  
793 >  
794 >  
795 >  
796 >  
797 >  
798 >  
799 >  
800 >  
801 >  
802 >  
803 >  
804 >  
805 >  
806 >  
807 >  
808 >  
809 >  
8010 >  
8011 >  
8012 >  
8013 >  
8014 >  
8015 >  
8016 >  
8017 >  
8018 >  
8019 >  
8020 >  
8021 >  
8022 >  
8023 >  
8024 >  
8025 >  
8026 >  
8027 >  
8028 >  
8029 >  
8030 >  
8031 >  
8032 >  
8033 >  
8034 >  
8035 >  
8036 >  
8037 >  
8038 >  
8039 >  
8040 >  
8041 >  
8042 >  
8043 >  
8044 >  
8045 >  
8046 >  
8047 >  
8048 >  
8049 >  
8050 >  
8051 >  
8052 >  
8053 >  
8054 >  
8055 >  
8056 >  
8057 >  
8058 >  
8059 >  
8060 >  
8061 >  
8062 >  
8063 >  
8064 >  
8065 >  
8066 >  
8067 >  
8068 >  
8069 >  
8070 >  
8071 >  
8072 >  
8073 >  
8074 >  
8075 >  
8076 >  
8077 >  
8078 >  
8079 >  
8080 >  
8081 >  
8082 >  
8083 >  
8084 >  
8085 >  
8086 >  
8087 >  
8088 >  
8089 >  
8090 >  
8091 >  
8092 >  
8093 >  
8094 >  
8095 >  
8096 >  
8097 >  
8098 >  
8099 >  
80100 >
```

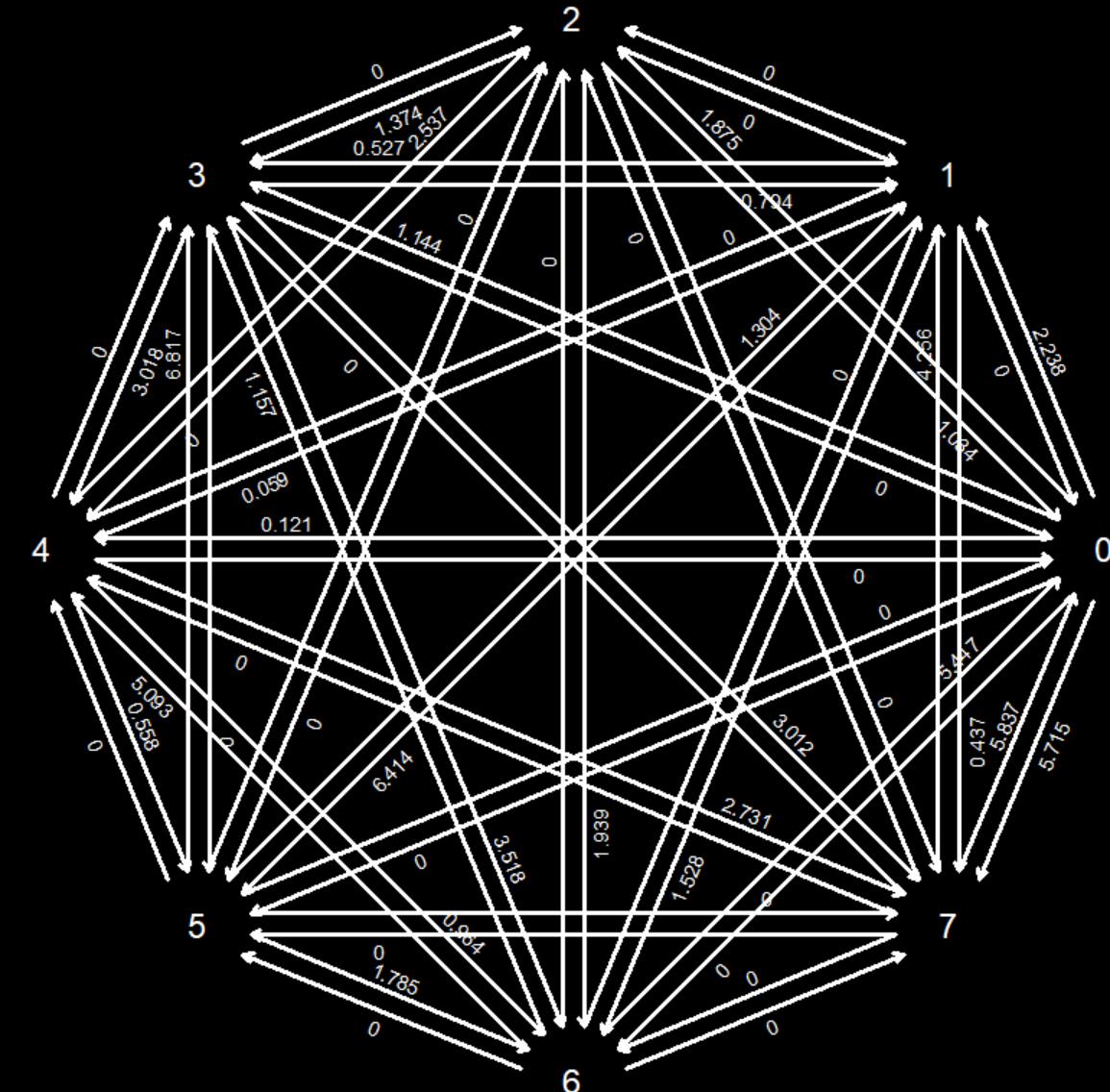


Project: (None) ▾

oplites cavifrons
oplites rupestris
oplites ariommus
lutes interruptus
lutes nigromaculatus
lutes annularis
lutes obesus
archus macropterus
terus dolomieu
terus punctulatus
terus floridanus
terus salmoides
terus treculi
terus notius
terus coosae
lutes cyanellus
lutes symmetricus
lutes gulosus
lutes macrochirus
lutes humilis
lutes megalotis
lutes marginatus
lutes auritus
lutes miniatus
lutes punctulatus
lutes microlophus
lutes gibbosus
archus pomotis







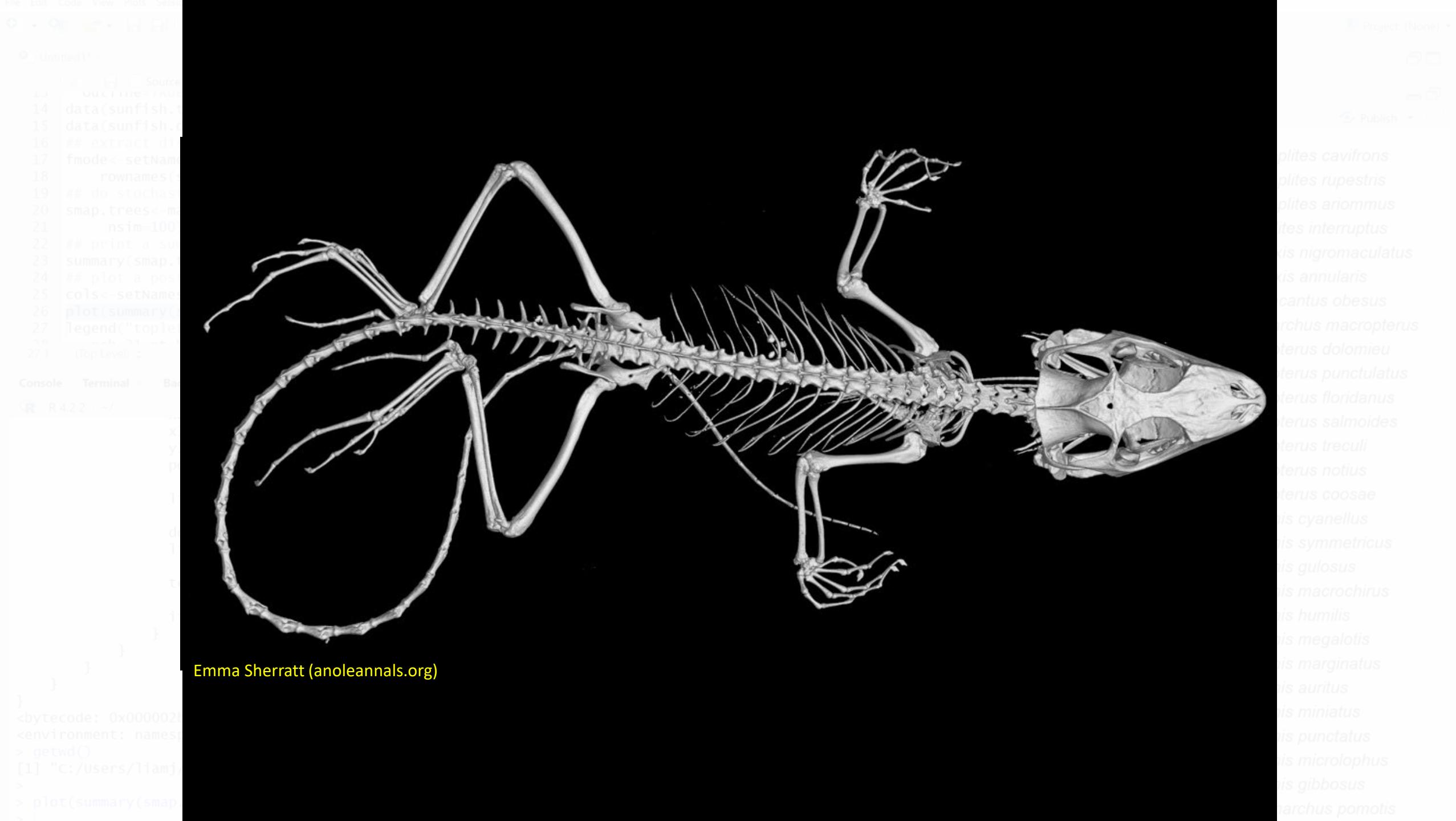
```
① Untitled1  
1 > #> sunfis  
2 > sunfis  
3 > sunfis  
4 > data(sunfis)  
5 > data(sunfis)  
6 ## extract  
7 fmode<-setS  
8 rowname  
9 ## do stoc  
10 smap.trees<  
11 nsim=1  
12 ## print a  
13 summary(Sma  
14 ## plot a t  
15 cols<-setNa  
16 plot(suma  
17 Legend("top  
18 nlevels(Sm  
19 (topLevel, c  
20 )
```

Console Terminal

R 4.2.2 -->



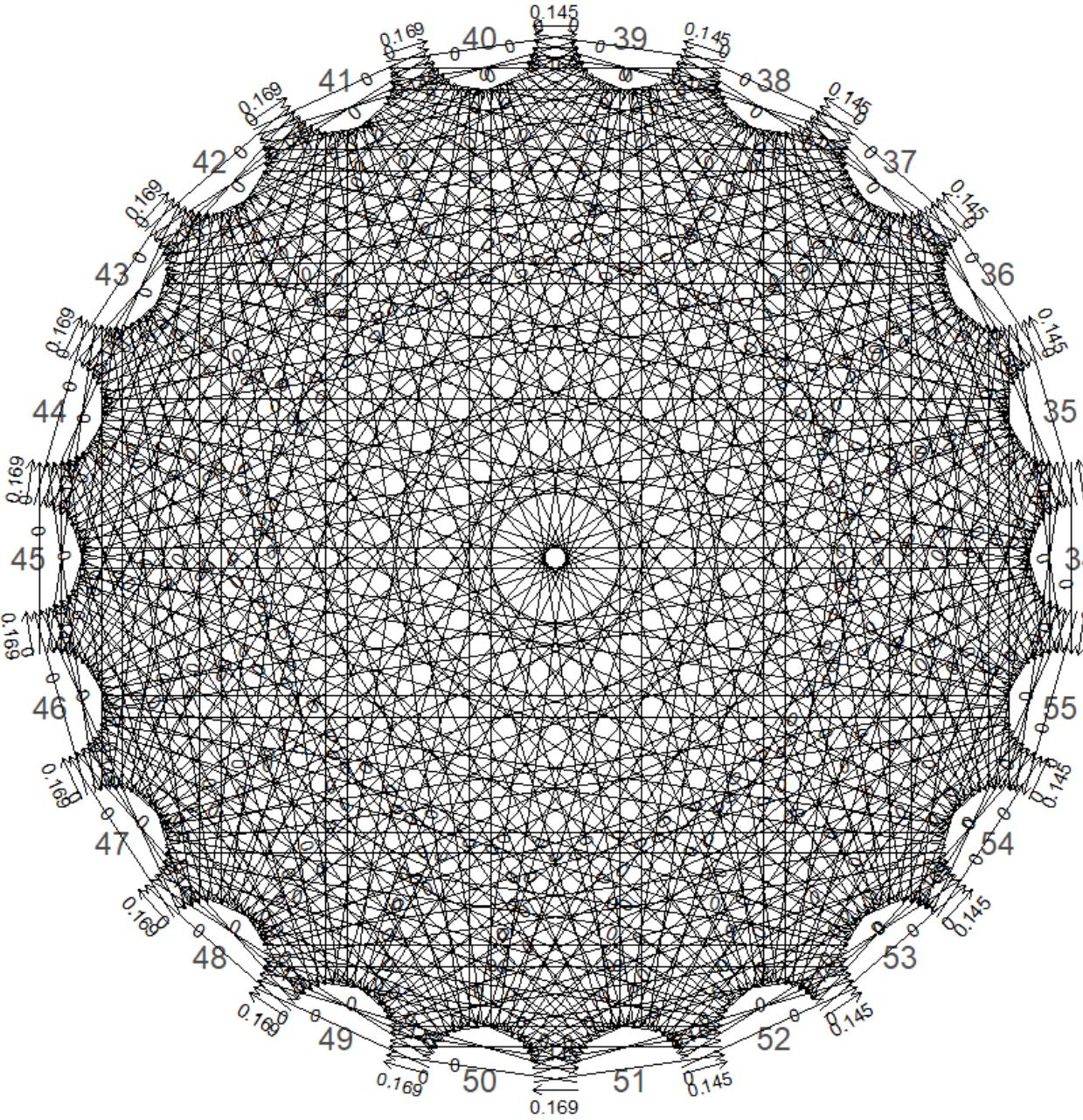
s cavifrons
s rupestris
s ariommus
interruptus
igromaculatus
nnularis
tus obesus
is macropterus
is dolomieu
is punctulatus
is floridanus
is salmoides
is treculi
is notius
is coosae
yanellus
ymmetricus
ulosus
macrochirus
umilis
negalotis
marginatus
unitus
iniatus
unctatus
microlophus
ibbosus
hus pomotis



Emma Sherratt (anoleannals.org)

```
File Edit Code View Plot Help  
Untitled1  
Source  
uci.mine.rav  
data(sunfish)  
data(sunfish)  
## extract df  
fmode<-setNames  
rownames(df)  
## do stochastic  
smap.trees<-mcmc  
nsim=100  
## print a summary  
summary(smap)  
## plot a posterior  
cols<-setNames  
plot(summary(smap))  
Legend("topleft")  
## (top level) z  
Console Terminal Batch  
R 4.2.2 -->  
x  
y  
p  
1  
d  
1  
t  
1  
f  
1  
i  
1  
}  
}  
}  
}  
<bytecode: 0x0000002  
<environment: names  
> getwd()  
[1] "C:/Users/11amj"  
> plot(summary(smap))  
>
```

plites cavifrons
plites rupestris
plites ariommus
tes interruptus
is nigromaculatus
is annularis
cantus obesus
archus macropterus
terus dolomieu
terus punctulatus
terus floridanus
terus salmoides
terus treculii
terus notius
terus coosae
is cyanellus
is symmetricus
is gulosus
is macrochirus
is humilis
is megalotis
is marginatus
is auritus
is miniatus
is punctatus
is microlophus
is gibbosus
archus pomotis

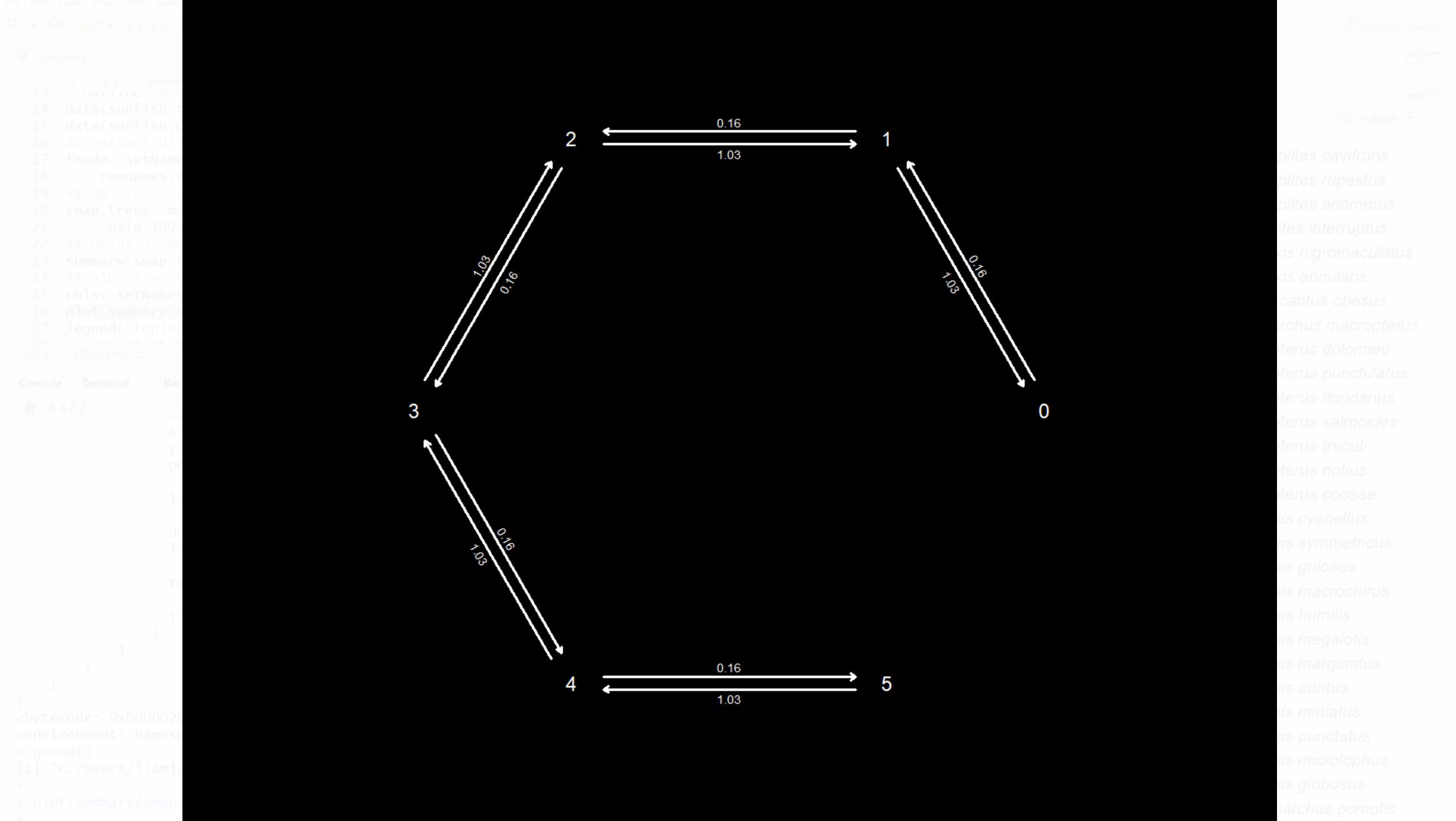


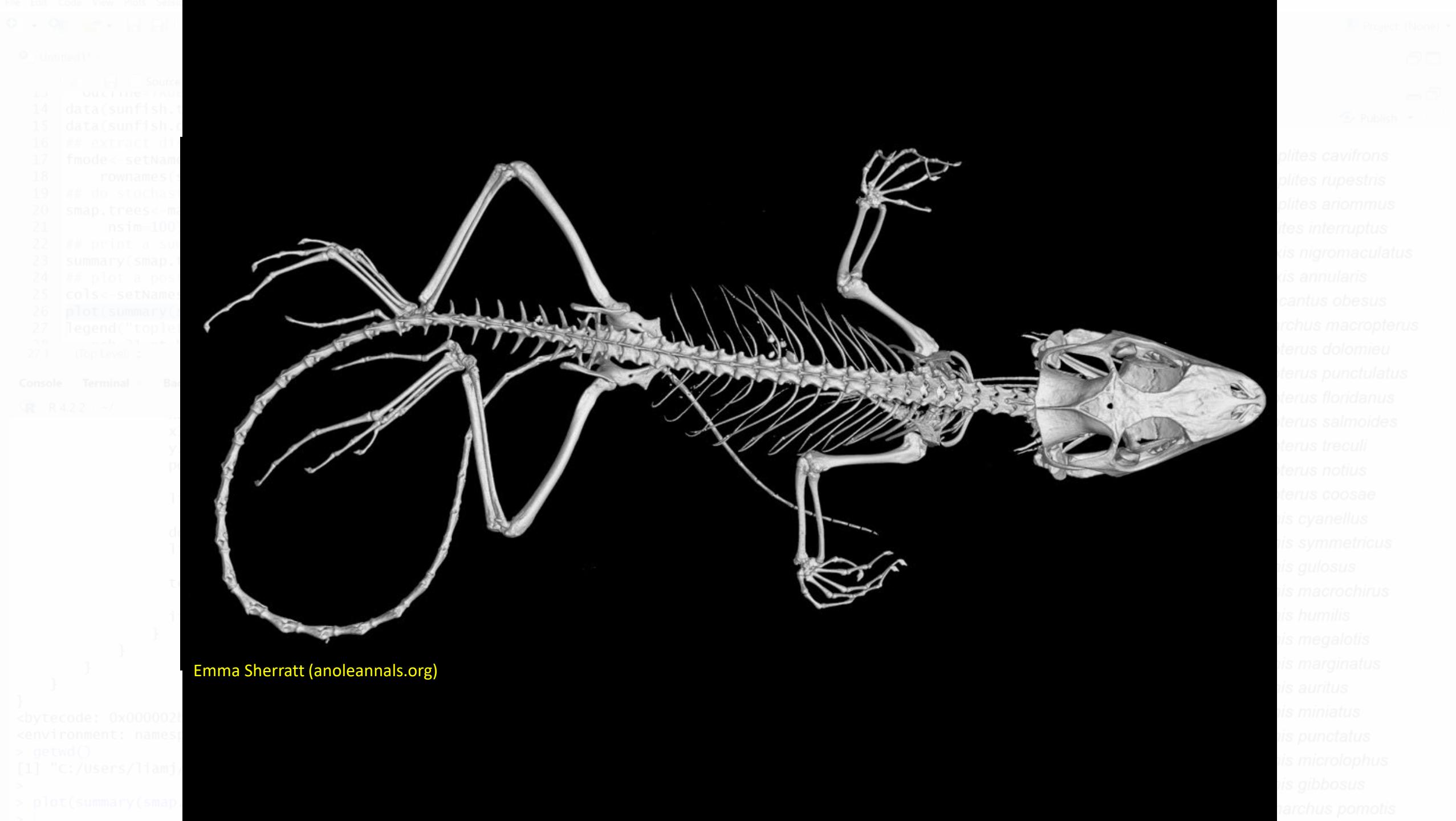
$$Q = \begin{bmatrix} -(q_{01} + q_{02}) & q_{01} & q_{02} \\ q_{10} & -(q_{10} + q_{12}) & q_{12} \\ q_{20} & q_{21} & -(q_{20} + q_{21}) \end{bmatrix}$$

$$P(t) = \exp(Qt)$$

$$Q = \begin{bmatrix} -(q_{01} + q_{02}) & q_{01} & 0 \\ q_{10} & -(q_{10} + q_{12}) & q_{12} \\ 0 & q_{21} & -(q_{20} + q_{21}) \end{bmatrix}$$

$$P(t) = \exp(Qt)$$

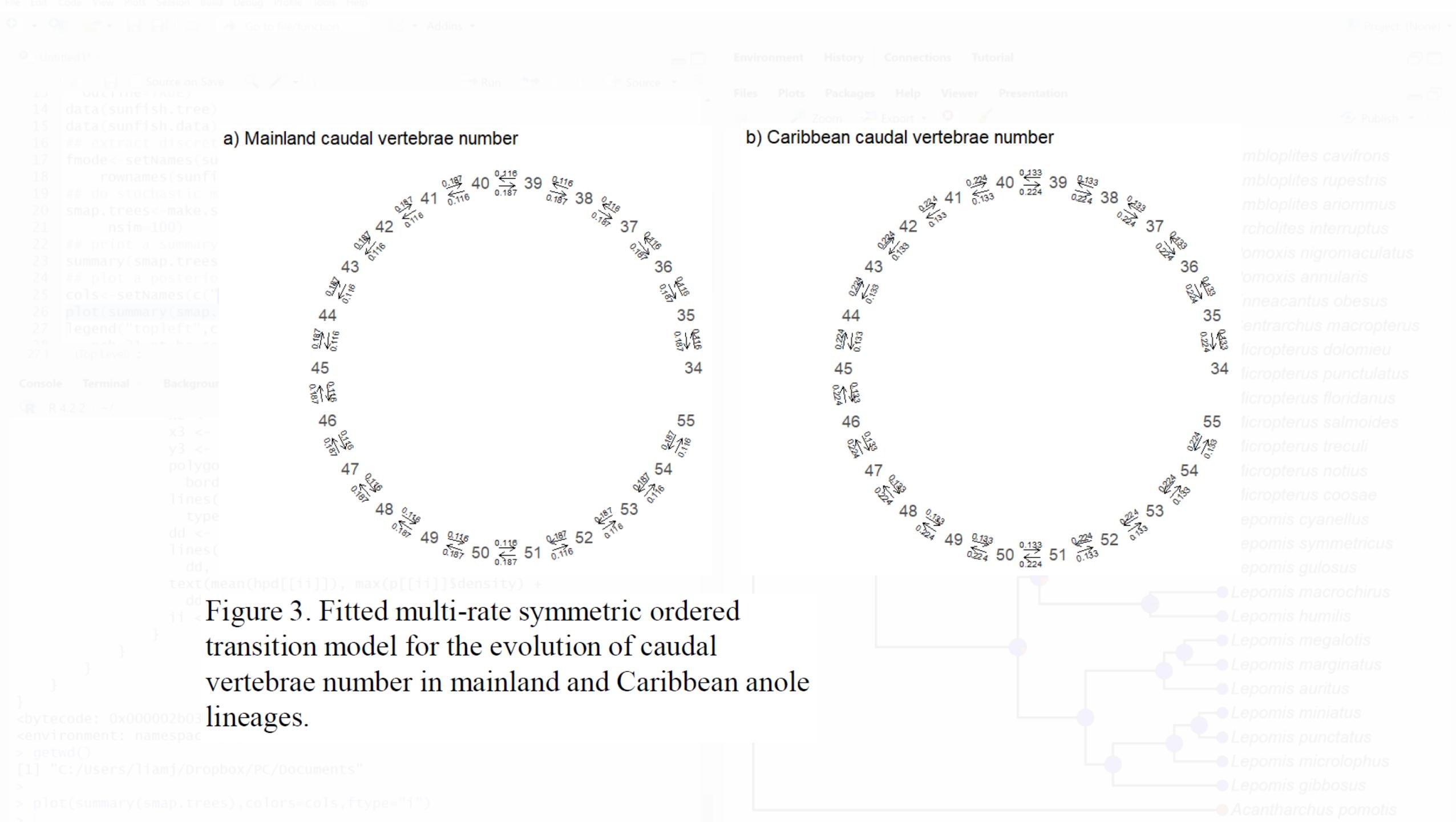




Emma Sherratt (anoleannals.org)

```
File Edit Code View Plot Help  
Untitled1  
Source  
uci.mle.R  
data(sunfish)  
data(sunfish)  
## extract df  
fmode<-setNames  
rownames(f)  
## do stochastic  
smap.trees<-m  
nsim=100  
## print a summary  
summary(smap)  
## plot a posterior  
cols<-setNames  
plot(summary(smap))  
Legend("topleft")  
## (top level) z  
Console Terminal Batch  
R 4.2.2 -->  
x  
y  
p  
1  
d  
1  
t  
f  
i  
f  
}  
}  
}  
<bytecode: 0x0000002  
<environment: names  
> getwd()  
[1] "C:/Users/11amj"  
> plot(summary(smap))  
>
```

olites cavifrons
olites rupestris
olites ariommus
olites interruptus
olis nigromaculatus
olis annularis
lantus obesus
archus macropterus
terus dolomieu
terus punctulatus
terus floridanus
terus salmoides
terus treculi
terus notius
terus coosae
lis cyanellus
lis symmetricus
lis gulosus
lis macrochirus
lis humilis
lis megalotis
lis marginatus
lis auritus
lis miniatus
lis punctatus
lis microlophus
lis gibbosus
larchus pomotis



```
x3 <- c(min(x2),
y3 <- c(0, y2,
polygon(x3, y3,
border = FALSE)
lines(p[[ii]]$x,
      type = "s")
dd <- 0.03 * d
lines(hpd[[ii]]$x,
      dd, 2))
text(mean(hpd[[ii]]$x,
          dd, "HPD", p
ii <- ii + 1

02b03146c8f0>
espace:phytools>
mj/Dropbox/PC/Documents">
ap.trees), colors=cols, ttype=
```

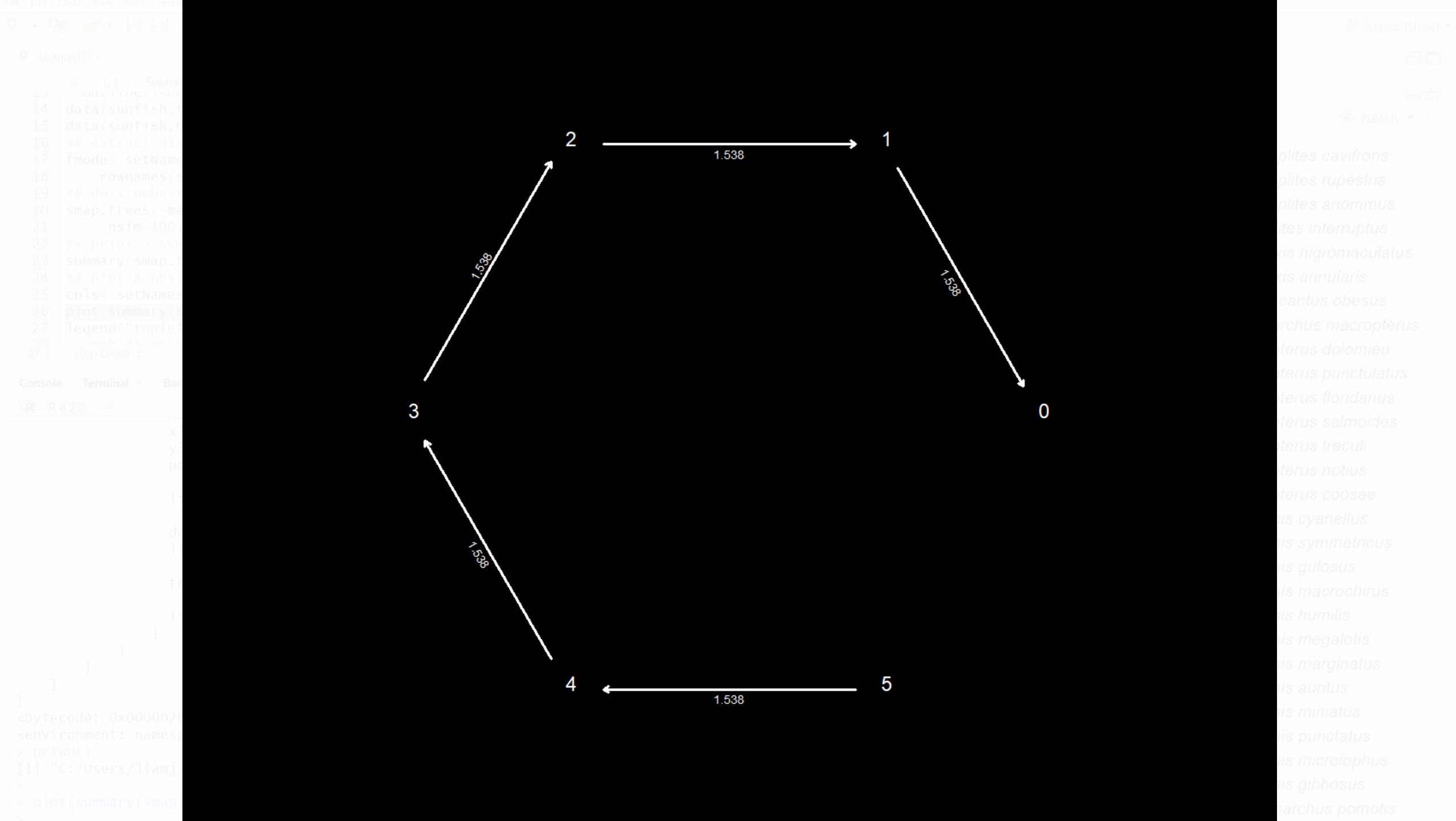


$$Q = \begin{bmatrix} -(q_{01} + q_{02}) & q_{01} & q_{02} \\ q_{10} & -(q_{10} + q_{12}) & q_{12} \\ q_{20} & q_{21} & -(q_{20} + q_{21}) \end{bmatrix}$$

$$P(t) = \exp(Qt)$$

$$Q = \begin{bmatrix} -(q_{01} + q_{02}) & 0 & 0 \\ q_{10} & -(q_{10} + q_{12}) & 0 \\ 0 & q_{21} & -(q_{20} + q_{21}) \end{bmatrix}$$

$$P(t) = \exp(Qt)$$



Discrete character evolution

- The Mk model is the most common model for discrete character evolution on the tree – but it is not the only one.
- (Much like all tractable models) the model contains some fairly unrealistic assumptions.
- Some alternatives exist. Possibly the most important of these is a model called the *threshold model*.

