

2000 Public Use Microdata Sample Data for Oregon MPO and Non-MPO Travel Demand Models

Ben Stabler
Transportation Planning Analysis Unit
Oregon Department of Transportation
555 13th Street NE
Salem, Oregon 97301-4178
benjamin.stabler@odot.state.or.us

September 30, 2003

This document summarizes the 2000 Census Public Use Microdata Sample (PUMS) data for Oregon MPO and non-MPO models. PUMS files represent five percent and one percent samples of housing and person records from the 2000 Census. Individual weights are applied to each housing and person record so the sample can be expanded to represent the total relevant population, which in this case is Oregon. Microdata are individual records that can be cross classified to create custom summary tables not in standard Census reports.

PUMS data is summarized at the Public Use Microdata Area (PUMA), which are areas with greater than 100,000 persons. There are also Super PUMAs, which represent areas of 400,000 or more persons. The PUMAs offer a greater spatial precision and therefore represent a better fit for urban travel demand models. PUMS boundaries follow county boundaries, city boundaries and census blocks. For the most part, PUMA boundaries are Oregon county lines. The Salem (01101) and Eugene (00702) PUMAs are a bit larger than their respective urban growth boundaries, while Metro is comprised of about 11 PUMAs. PUMAs are delineated by the Population Research Center at Portland State University. See the attached map for PUMA definitions.

For MPO travel demand models, a three-way classification of household size by household income by age of head of household is needed. For non-MPO models, a two-way classification of household size by household income is needed. These cross classification tables are then used as seeds in an iterative proportional fitting process to generate a joint distribution for each TAZ in the model. The three variables above are classified as follows:

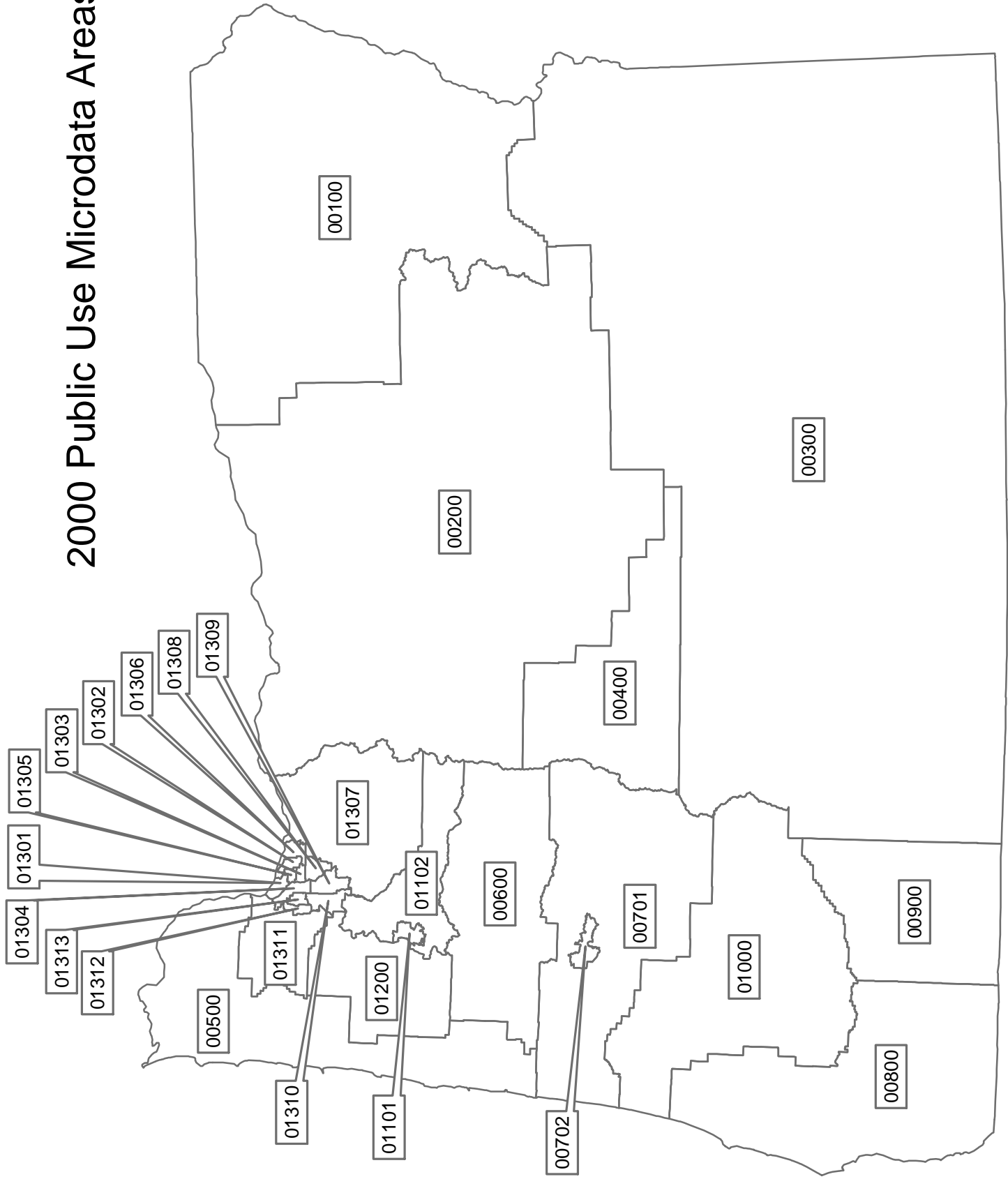
Variable	Class	1	2	3	4
(h) Household Size		1	2	3	4+
(i) Household Income (Non-MPO)		0-15k	15-25k	25-40k	40k+
(i) Household Income (MPO)		0-15k	15-30k	30-60k	60k+
(a) Age of Head of Household		0-24	25-54	55-64	65+

In order to find the correct two-dimensional or three-dimensional table in the following pages, one first needs to identify the appropriate PUMA code for their model. This can be found from the map of PUMAs or the description of PUMAs. Depending on whether the model is an MPO model or a non-MPO model, a three-way or two-way table is needed. The two-way tables are household size by household income by PUMA. The rows and columns of the table are labeled with the variable code and the class code. The three-way tables are household size by household income by age of head of household. Age of head of household is the third dimension and is represented as four tables (one for each age class) of household size by household income.

At the end of this document is the R code used to create the PUMS tables.

If there are any questions or concerns please email me at benjamin.stabler@odot.state.or.us

2000 Public Use Microdata Areas



**Two-Way Tables for Non-MPO models
(household size by household income)**

```
1 "$100"
2      i1      i2      i3      i4
3 h1 5420 2719 1973 1318
4 h2 2325 2532 3877 6879
5 h3 859 706 1340 3770
6 h4 793 1205 2856 6288
7
8 "$200"
9      i1      i2      i3      i4
10 h1 4012 1983 1781 1395
11 h2 1767 2239 3986 7405
12 h3 632 781 1548 3229
13 h4 826 967 2407 5885
14
15 "$300"
16      i1      i2      i3      i4
17 h1 5133 2450 1703 1067
18 h2 2187 2808 3638 6626
19 h3 917 1089 1285 2967
20 h4 989 1434 2389 5087
21
22 "$400"
23      i1      i2      i3      i4
24 h1 3346 2020 2460 2115
25 h2 1285 2097 3958 9796
26 h3 513 657 1552 4752
27 h4 496 750 2008 7137
28
29 "$500"
30      i1      i2      i3      i4
31 h1 6885 4100 3058 2588
32 h2 2412 3485 5548 11918
33 h3 996 916 1652 5321
34 h4 807 1103 2074 7888
35
36 "$600"
37      i1      i2      i3      i4
38 h1 7090 4099 3395 2902
39 h2 2910 3549 5314 14015
40 h3 1225 1027 1738 6472
41 h4 955 1402 3153 10760
42
43 "$701"
44      i1      i2      i3      i4
45 h1 4046 1960 2182 1688
46 h2 1648 2714 4303 8761
47 h3 412 689 1347 3618
48 h4 764 775 1780 6302
49
50 "$702"
51      i1      i2      i3      i4
52 h1 10128 5651 5229 4050
53 h2 5011 4888 7044 14963
54 h3 1447 1782 2623 7531
55 h4 1402 1645 2876 11108
56
57 "$800"
58      i1      i2      i3      i4
59 h1 9150 3941 3081 2166
60 h2 3830 5117 6756 11439
61 h3 1147 1420 2114 4225
62 h4 1480 1547 2374 6445
63
64 "$900"
65      i1      i2      i3      i4
66 h1 7265 4580 3428 2872
67 h2 2841 3998 6822 13110
```

```
68 h3 1400 1320 2083 6429
69 h4 981 1444 3398 9408
70
71 $"1000"
72 i1 i2 i3 i4
73 h1 4643 2138 1538 1151
74 h2 1951 2877 4428 6545
75 h3 587 863 1655 3248
76 h4 572 1031 2037 5364
77
78 $"1101"
79 i1 i2 i3 i4
80 h1 5565 3826 4407 3596
81 h2 2394 2659 4661 10902
82 h3 1053 942 2069 5485
83 h4 1469 1722 2848 9304
84
85 $"1102"
86 i1 i2 i3 i4
87 h1 2798 1613 1605 1424
88 h2 1252 1214 2888 8465
89 h3 489 695 881 3944
90 h4 542 848 2500 7539
91
92 $"1200"
93 i1 i2 i3 i4
94 h1 4314 1968 2638 2049
95 h2 1683 2435 3876 10760
96 h3 978 946 1422 5459
97 h4 673 1074 2297 10157
98
99 $"1301"
100 i1 i2 i3 i4
101 h1 4005 2443 2775 2129
102 h2 1216 1467 2749 7738
103 h3 666 812 1058 3925
104 h4 1038 1082 1896 4797
105
106 $"1302"
107 i1 i2 i3 i4
108 h1 3247 2734 2940 1963
109 h2 1387 1769 3418 6822
110 h3 583 803 1320 3491
111 h4 702 658 1633 5927
112
113 $"1303"
114 i1 i2 i3 i4
115 h1 4666 3124 3018 2367
116 h2 1105 1576 3670 8245
117 h3 377 601 1354 3949
118 h4 473 615 1388 5122
119
120 $"1304"
121 i1 i2 i3 i4
122 h1 8677 4025 4768 6753
123 h2 1308 1119 2413 12081
124 h3 282 554 508 4069
125 h4 104 107 363 5403
126
127 $"1305"
128 i1 i2 i3 i4
129 h1 5433 3527 5045 4085
130 h2 1140 1836 3488 10578
131 h3 432 466 1216 4734
132 h4 457 568 732 5785
133
134 $"1306"
```

```
135      i1      i2      i3      i4
136 h1 2739 2482 2381 2487
137 h2 1016 1330 2955 9017
138 h3 453 628 1559 4807
139 h4 787 1095 2072 7444
140
141 $"1307"
142      i1      i2      i3      i4
143 h1 2044 1096 1496 1341
144 h2 616 967 3005 9341
145 h3 346 383 862 4604
146 h4 432 408 1376 8790
147
148 $"1308"
149      i1      i2      i3      i4
150 h1 3222 2811 3360 3014
151 h2 976 1517 3606 9903
152 h3 543 497 1460 5593
153 h4 654 377 1674 8542
154
155 $"1309"
156      i1      i2      i3      i4
157 h1 2141 1671 2592 2828
158 h2 542 1120 2294 11209
159 h3 329 346 658 4917
160 h4 122 422 1084 9458
161
162 $"1310"
163      i1      i2      i3      i4
164 h1 2550 2367 3430 3156
165 h2 852 1220 2286 10930
166 h3 200 471 966 5333
167 h4 152 544 1073 8276
168
169 $"1311"
170      i1      i2      i3      i4
171 h1 3040 2099 3004 2890
172 h2 923 1485 2537 11767
173 h3 423 455 1144 6123
174 h4 689 1119 2202 10831
175
176 $"1312"
177      i1      i2      i3      i4
178 h1 1554 1509 2814 2774
179 h2 736 896 2035 9448
180 h3 293 501 916 5956
181 h4 328 541 970 9803
182
183 $"1313"
184      i1      i2      i3      i4
185 h1 2664 2923 3367 3934
186 h2 624 1262 2840 10246
187 h3 372 333 802 4254
188 h4 393 796 1434 6308
189
```

**Three Way Tables for MPO models
(household size by household income by
age of head of household)**


```
1 $"100"
2 , , a1
3
4     i1    i2    i3    i4
5 h1  543  152   39    0
6 h2  404  188  167    0
7 h3  212  263  250   37
8 h4   30  182  221   85
9
10 , , a2
11
12     i1    i2    i3    i4
13 h1 1431 1316 1233  145
14 h2  854 1161 2247 1615
15 h3  508  550 1712 1662
16 h4  679 1750 4264 3114
17
18 , , a3
19
20     i1    i2    i3    i4
21 h1  813  427  369  107
22 h2  373  551 1588 1159
23 h3   81  166  293  378
24 h4   48   87  197  252
25
26 , , a4
27
28     i1    i2    i3    i4
29 h1 2633 1670  442  110
30 h2  694 2118 1904  590
31 h3   58  149  228  128
32 h4   36   76   76   45
33
34
35 $"200"
36 , , a1
37
38     i1    i2    i3    i4
39 h1  154   78   13    0
40 h2  256  145  398   40
41 h3  125  160  272   24
42 h4   39  118  153   59
43
44 , , a2
45
46     i1    i2    i3    i4
47 h1 1021 1176  968  135
48 h2  520 1136 2877 1592
49 h3  449  913 1753 1195
50 h4  708 1599 4017 2495
51
52 , , a3
53
54     i1    i2    i3    i4
55 h1  530  513  318  171
56 h2  420  574 1378 1173
57 h3    6  116  206  341
58 h4   55   39  175  199
59
60 , , a4
61
62     i1    i2    i3    i4
63 h1 2307 1018  584  185
64 h2  571 1632 2030  655
65 h3   52  115  281  182
66 h4   24   54  225  126
67
```

```
68
69 $"300"
70 , , a1
71
72     i1    i2    i3    i4
73 h1  320    60    27     6
74 h2  207   293   180     0
75 h3  143   134    70    57
76 h4   58    53   205    34
77
78 , , a2
79
80     i1    i2    i3    i4
81 h1 1347 1375   999   113
82 h2  864 1150 2328 1411
83 h3  660 1203 1846 1056
84 h4  907 2098 3609 2297
85
86 , , a3
87
88     i1    i2    i3    i4
89 h1  866   534   124   105
90 h2  492   782  1124  1145
91 h3   90    94   280   233
92 h4   12    55   157   216
93
94 , , a4
95
96     i1    i2    i3    i4
97 h1 2600 1309   412   156
98 h2  624 1950 1877   832
99 h3   24    98   208    62
100 h4   12    16   111    59
101
102
103 $"400"
104 , , a1
105
106     i1    i2    i3    i4
107 h1  330   148    30     0
108 h2   67   391   300    28
109 h3   85   147   202    61
110 h4   28   141   174     0
111
112 , , a2
113
114     i1    i2    i3    i4
115 h1 1179 1210 1505   342
116 h2  466   928  3076  2836
117 h3  360   917  2062  2340
118 h4  391  1039  3689  3991
119
120 , , a3
121
122     i1    i2    i3    i4
123 h1  333   423   451   148
124 h2  415   525  1236  1589
125 h3   68    18   300   355
126 h4   52    63   190   325
127
128 , , a4
129
130     i1    i2    i3    i4
131 h1 1504 1411   728   199
132 h2  337  1735  1824  1383
133 h3    0    97   268   194
134 h4   25    64    44   175
```

```
135
136
137 $"500"
138 , , a1
139
140     i1     i2     i3     i4
141 h1  313   256    61    32
142 h2  227   239   298    42
143 h3  231   139   317    27
144 h4   94   116   140    18
145
146 , , a2
147
148     i1     i2     i3     i4
149 h1 2027  2184  1678   409
150 h2 1096  1592  3168  3175
151 h3  695   967  2455  2171
152 h4   704  1436  4387  4020
153
154 , , a3
155
156     i1     i2     i3     i4
157 h1 1116   538   700   138
158 h2  402  1070  2546  2029
159 h3   61    91   374   514
160 h4    9    19   197   387
161
162 , , a4
163
164     i1     i2     i3     i4
165 h1 3429  2494   998   258
166 h2  687  2484  2768  1540
167 h3    9   179   316   339
168 h4    0    83    75   187
169
170
171 $"600"
172 , , a1
173
174     i1     i2     i3     i4
175 h1 1259   128    33     0
176 h2 1106   917   568    24
177 h3  379   266   469    50
178 h4  123   270   475    59
179
180 , , a2
181
182     i1     i2     i3     i4
183 h1 1873  2370  2438   796
184 h2 1021  1613  4623  3914
185 h3  735  1009  2217  3327
186 h4  820  1895  5599  5885
187
188 , , a3
189
190     i1     i2     i3     i4
191 h1  781   625   401   203
192 h2  271   571  1795  2059
193 h3   86   118   251   789
194 h4   12   190    87   375
195
196 , , a4
197
198     i1     i2     i3     i4
199 h1 3177  2435   606   361
200 h2  512  2289  2755  1750
201 h3   25   175   320   246
```

```
202 h4      0    33  112  335
203
204
205 $"701"
206 , , a1
207
208      i1    i2    i3    i4
209 h1  154    46     0     0
210 h2   74   243   268    27
211 h3   49    78   132    12
212 h4   43     0    99    40
213
214 , , a2
215
216      i1    i2    i3    i4
217 h1 1137   962 1309   198
218 h2  533   816 2725  2229
219 h3  330   850 1655  1589
220 h4  657  1219 3488  3541
221
222 , , a3
223
224      i1    i2    i3    i4
225 h1  633   488   375   165
226 h2  298   748  1632  1572
227 h3    0    82   312   285
228 h4   49    34    36   197
229
230 , , a4
231
232      i1    i2    i3    i4
233 h1 2122  1378   664   245
234 h2  743  2278  2443   797
235 h3   33    79   442   138
236 h4   15     0    82   121
237
238
239 $"702"
240 , , a1
241
242      i1    i2    i3    i4
243 h1 2200   521   137     0
244 h2 2022  1124   736   119
245 h3  672   849   615    47
246 h4  270   402   437   164
247
248 , , a2
249
250      i1    i2    i3    i4
251 h1 3871  3424  3551   700
252 h2 2082  2963  6267  4001
253 h3  647  1722  3519  3275
254 h4 1082  1893  5517  6255
255
256 , , a3
257
258      i1    i2    i3    i4
259 h1  981   843   831   572
260 h2  403   845  1795  1967
261 h3   31   177   309   607
262 h4   50    61   372   310
263
264 , , a4
265
266      i1    i2    i3    i4
267 h1 3076  2604  1276   471
268 h2  504  2304  3075  1699
```

```
269 h3 97 204 303 309
270 h4 0 24 86 108
271
272
273 $"800"
274 , , a1
275
276 i1 i2 i3 i4
277 h1 370 233 24 0
278 h2 321 386 124 70
279 h3 241 101 178 27
280 h4 172 106 132 0
281
282 , , a2
283
284 i1 i2 i3 i4
285 h1 2128 1636 1535 339
286 h2 1326 1671 3814 2508
287 h3 823 1469 2333 1584
288 h4 1240 2012 4260 2678
289
290 , , a3
291
292 i1 i2 i3 i4
293 h1 1408 827 684 158
294 h2 839 1109 2287 1764
295 h3 6 151 261 382
296 h4 25 194 273 202
297
298 , , a4
299
300 i1 i2 i3 i4
301 h1 5244 2316 1072 364
302 h2 1344 4038 4108 1433
303 h3 77 319 713 241
304 h4 43 91 278 140
305
306
307 $"900"
308 , , a1
309
310 i1 i2 i3 i4
311 h1 515 152 85 25
312 h2 447 581 572 88
313 h3 184 113 493 0
314 h4 30 273 225 77
315
316 , , a2
317
318 i1 i2 i3 i4
319 h1 2110 2243 1799 390
320 h2 1083 2130 3393 3278
321 h3 1029 1719 2609 2713
322 h4 871 2113 5335 5020
323
324 , , a3
325
326 i1 i2 i3 i4
327 h1 1125 990 512 232
328 h2 488 1017 2304 1978
329 h3 130 113 521 735
330 h4 80 103 223 344
331
332 , , a4
333
334 i1 i2 i3 i4
335 h1 3515 2666 1297 489
```

```
336 h2 823 2652 3844 2093
337 h3 57 108 341 367
338 h4 0 110 168 259
339
340
341 $"1000"
342 , , a1
343
344 i1 i2 i3 i4
345 h1 185 105 27 0
346 h2 167 114 225 30
347 h3 128 188 213 45
348 h4 6 88 165 27
349
350 , , a2
351
352 i1 i2 i3 i4
353 h1 1283 894 793 90
354 h2 674 1050 2311 1453
355 h3 359 869 1674 1300
356 h4 506 1271 3249 2625
357
358 , , a3
359
360 i1 i2 i3 i4
361 h1 574 504 395 113
362 h2 417 1001 1581 1022
363 h3 42 172 282 316
364 h4 60 126 268 202
365
366 , , a4
367
368 i1 i2 i3 i4
369 h1 2601 1252 427 227
370 h2 693 2170 2269 624
371 h3 58 154 357 196
372 h4 0 78 162 171
373
374
375 $"1101"
376 , , a1
377
378 i1 i2 i3 i4
379 h1 580 217 27 0
380 h2 850 476 653 83
381 h3 366 243 432 95
382 h4 165 436 367 129
383
384 , , a2
385
386 i1 i2 i3 i4
387 h1 1797 2616 2530 601
388 h2 960 1646 3534 3208
389 h3 638 1075 2918 2397
390 h4 1200 2219 5227 4789
391
392 , , a3
393
394 i1 i2 i3 i4
395 h1 704 813 911 218
396 h2 215 438 1408 1543
397 h3 24 153 123 557
398 h4 85 48 183 317
399
400 , , a4
401
402 i1 i2 i3 i4
```

```
403 h1 2484 2390 1168 338
404 h2 369 1768 2340 1125
405 h3 25 167 102 234
406 h4 19 0 88 71
407
408
409 $"1102"
410 , , a1
411
412 i1 i2 i3 i4
413 h1 189 177 0 0
414 h2 143 88 175 80
415 h3 51 284 71 69
416 h4 61 180 127 36
417
418 , , a2
419
420 i1 i2 i3 i4
421 h1 388 702 1005 228
422 h2 336 518 1782 2393
423 h3 376 545 1584 1886
424 h4 394 1474 4007 4360
425
426 , , a3
427
428 i1 i2 i3 i4
429 h1 349 370 350 142
430 h2 199 457 1223 1332
431 h3 37 43 167 371
432 h4 81 28 94 250
433
434 , , a4
435
436 i1 i2 i3 i4
437 h1 1872 1072 471 125
438 h2 574 1062 2232 1225
439 h3 25 73 298 129
440 h4 6 6 176 149
441
442
443 $"1200"
444 , , a1
445
446 i1 i2 i3 i4
447 h1 403 68 37 0
448 h2 485 461 458 19
449 h3 317 256 261 33
450 h4 60 91 220 44
451
452 , , a2
453
454 i1 i2 i3 i4
455 h1 1044 1131 1644 454
456 h2 675 1139 3225 3116
457 h3 627 575 2571 2427
458 h4 580 1697 4834 5509
459
460 , , a3
461
462 i1 i2 i3 i4
463 h1 529 421 574 103
464 h2 139 255 1201 1882
465 h3 28 92 372 450
466 h4 0 114 221 393
467
468 , , a4
469
```

```
470      i1      i2      i3      i4
471 h1 2338 1318   736   169
472 h2   384 1943 2366 1006
473 h3     6   302   382   106
474 h4    33    54   193   158
475
476
477 $"1301"
478 , , a1
479
480      i1      i2      i3      i4
481 h1  226   128    19     0
482 h2  211   238   134    61
483 h3  112   229   259    15
484 h4  103    83   165    52
485
486 , , a2
487
488      i1      i2      i3      i4
489 h1 1610 1660 2388   453
490 h2  603 1188 3508 2694
491 h3  539   524 1872 1650
492 h4  850 1386 3045 2316
493
494 , , a3
495
496      i1      i2      i3      i4
497 h1  590   556   490   150
498 h2  167   359   738   734
499 h3   15   119   190   243
500 h4   24   172   162    96
501
502 , , a4
503
504      i1      i2      i3      i4
505 h1 1579 1027   369   107
506 h2  235   681 1069   550
507 h3    0   218   238   238
508 h4   61    74   111   113
509
510
511 $"1302"
512 , , a1
513
514      i1      i2      i3      i4
515 h1  312   149    65     0
516 h2  162   276   223    75
517 h3  149   289   158    99
518 h4   46    82   138    46
519
520 , , a2
521
522      i1      i2      i3      i4
523 h1 1057 1612 1842   410
524 h2  759 1161 2302 1967
525 h3  358   771 1722 1455
526 h4  589 1104 3528 2674
527
528 , , a3
529
530      i1      i2      i3      i4
531 h1  472   506   446    36
532 h2  148   223 1009   523
533 h3    0    64   277   187
534 h4   43    39    79   259
535
536 , , a4
```



```
537
538      i1      i2      i3      i4
539 h1 1406 1807 581 183
540 h2 318 1025 2109 1116
541 h3 76 154 265 173
542 h4 24 30 158 81
543
544
545 $"1303"
546 , , a1
547
548      i1      i2      i3      i4
549 h1 327 172 0 0
550 h2 238 438 433 40
551 h3 108 83 217 66
552 h4 134 69 209 52
553
554 , , a2
555
556      i1      i2      i3      i4
557 h1 1601 2146 2609 633
558 h2 426 1450 4045 3138
559 h3 269 746 2190 1519
560 h4 300 836 2552 2665
561
562 , , a3
563
564      i1      i2      i3      i4
565 h1 543 467 503 117
566 h2 136 243 525 614
567 h3 0 67 264 227
568 h4 24 0 279 268
569
570 , , a4
571
572      i1      i2      i3      i4
573 h1 2195 1323 481 58
574 h2 305 777 1283 505
575 h3 0 127 176 222
576 h4 15 0 68 127
577
578
579 $"1304"
580 , , a1
581
582      i1      i2      i3      i4
583 h1 1993 679 181 0
584 h2 522 615 382 74
585 h3 36 154 135 115
586 h4 22 21 81 102
587
588 , , a2
589
590      i1      i2      i3      i4
591 h1 3919 3698 3670 2377
592 h2 467 868 3035 5098
593 h3 209 409 960 2489
594 h4 82 171 836 4352
595
596 , , a3
597
598      i1      i2      i3      i4
599 h1 881 501 799 488
600 h2 56 278 377 2165
601 h3 0 81 86 398
602 h4 0 0 65 170
603
```

```
604 , , a4
605
606      i1      i2      i3      i4
607 h1 1884 1090 1342 721
608 h2 263 250 894 1577
609 h3 37 0 34 270
610 h4 0 0 21 54
611
612
613 $"1305"
614 , , a1
615
616      i1      i2      i3      i4
617 h1 506 278 39 0
618 h2 278 381 558 91
619 h3 90 168 223 140
620 h4 49 80 231 54
621
622 , , a2
623
624      i1      i2      i3      i4
625 h1 2359 2987 4091 1255
626 h2 478 1397 3929 4655
627 h3 309 528 1883 2412
628 h4 377 607 1818 3706
629
630 , , a3
631
632      i1      i2      i3      i4
633 h1 713 522 506 177
634 h2 162 226 555 1138
635 h3 33 48 191 383
636 h4 31 118 119 148
637
638 , , a4
639
640      i1      i2      i3      i4
641 h1 1855 1655 845 302
642 h2 222 802 1458 712
643 h3 0 67 136 237
644 h4 0 30 64 110
645
646
647 $"1306"
648 , , a1
649
650      i1      i2      i3      i4
651 h1 225 251 18 0
652 h2 165 365 537 45
653 h3 113 156 384 70
654 h4 59 250 274 36
655
656 , , a2
657
658      i1      i2      i3      i4
659 h1 846 1488 2487 500
660 h2 592 868 3012 3366
661 h3 340 788 2244 2347
662 h4 695 1423 3047 4898
663
664 , , a3
665
666      i1      i2      i3      i4
667 h1 241 524 364 147
668 h2 129 204 915 936
669 h3 0 94 201 252
670 h4 18 24 131 264
```

```
671
672 , , a4
673
674      i1      i2      i3      i4
675 h1 1427 1096 354 121
676 h2 130 705 1652 697
677 h3 0 105 212 141
678 h4 15 64 110 90
679
680
681 $"1307"
682 , , a1
683
684      i1      i2      i3      i4
685 h1 96 93 66 0
686 h2 49 116 175 58
687 h3 54 63 42 79
688 h4 0 27 71 31
689
690 , , a2
691
692      i1      i2      i3      i4
693 h1 667 497 1128 392
694 h2 175 549 2200 3265
695 h3 258 404 1706 2405
696 h4 325 684 3369 5249
697
698 , , a3
699
700      i1      i2      i3      i4
701 h1 310 274 198 105
702 h2 111 480 1528 1879
703 h3 0 43 290 516
704 h4 107 68 180 400
705
706 , , a4
707
708      i1      i2      i3      i4
709 h1 971 735 403 42
710 h2 281 924 1475 664
711 h3 34 108 74 119
712 h4 0 66 276 153
713
714
715 $"1308"
716 , , a1
717
718      i1      i2      i3      i4
719 h1 169 305 56 28
720 h2 147 338 540 146
721 h3 103 46 184 75
722 h4 119 53 186 0
723
724 , , a2
725
726      i1      i2      i3      i4
727 h1 896 1609 2238 745
728 h2 420 763 2747 3478
729 h3 337 664 2063 3041
730 h4 516 922 3517 4992
731
732 , , a3
733
734      i1      i2      i3      i4
735 h1 417 631 496 72
736 h2 219 403 1260 1345
737 h3 18 33 546 359
```

```
738 h4 19 0 237 358
739
740 , , a4
741
742 i1 i2 i3 i4
743 h1 1740 1621 1048 336
744 h2 190 1303 1841 862
745 h3 85 33 287 219
746 h4 0 15 161 152
747
748
749 $"1309"
750 , , a1
751
752 i1 i2 i3 i4
753 h1 67 155 65 0
754 h2 157 183 304 105
755 h3 21 130 85 36
756 h4 19 81 126 13
757
758 , , a2
759
760 i1 i2 i3 i4
761 h1 675 932 2086 693
762 h2 198 625 2150 4874
763 h3 219 443 912 3189
764 h4 103 536 2395 7192
765
766 , , a3
767
768 i1 i2 i3 i4
769 h1 287 211 305 200
770 h2 109 174 1211 1812
771 h3 37 0 123 724
772 h4 0 61 57 284
773
130 , , a4
775
776 i1 i2 i3 i4
777 h1 1112 1445 762 237
778 h2 78 793 1137 1255
779 h3 52 0 113 166
780 h4 0 16 147 56
781
782
783 $"1310"
784 , , a1
785
786 i1 i2 i3 i4
787 h1 212 227 42 31
788 h2 193 269 580 110
789 h3 52 103 178 27
790 h4 27 125 59 0
791
792 , , a2
793
794 i1 i2 i3 i4
795 h1 698 1547 2176 780
796 h2 439 652 2493 4322
797 h3 148 558 1496 3467
798 h4 125 762 2420 5848
799
800 , , a3
801
802 i1 i2 i3 i4
803 h1 295 400 540 264
804 h2 52 232 684 2011
```

```
805 h3      0      0      81  511
806 h4      0      28      30  509
807
808 , , a4
809
810      i1      i2      i3      i4
811 h1 1345 1822  792  332
812 h2  168  852 1148 1083
813 h3      0      22  199  128
814 h4      0      0      65   47
815
816
817 $"1311"
818 , , a1
819
820      i1      i2      i3      i4
821 h1  303  439   80   37
822 h2   99  264  265  166
823 h3  116  140  337   41
824 h4   81  163  218  118
825
826 , , a2
827
828      i1      i2      i3      i4
829 h1 1009 1579 2388  983
830 h2  376 1020 2371 4961
831 h3  289  461 2248 3213
832 h4  568 1315 4961 6497
833
834 , , a3
835
836      i1      i2      i3      i4
837 h1  332  205  539  114
838 h2  231  372  963 2078
839 h3      0      33  266  411
840 h4   15      33  156  408
841
842 , , a4
843
844      i1      i2      i3      i4
845 h1 1396 1050  432  147
846 h2  217  856 1442 1031
847 h3   18   78  299  195
848 h4   25   21   67  195
849
850
851 $"1312"
852 , , a1
853
854      i1      i2      i3      i4
855 h1  189  317   97   34
856 h2  143  346  530  116
857 h3   72  206  115   36
858 h4      0  103  133   99
859
860 , , a2
861
862      i1      i2      i3      i4
863 h1  580 1176 2412  824
864 h2  393  924 2060 4004
865 h3  182  550 1858 3710
866 h4  292  736 2560 7220
867
868 , , a3
869
870      i1      i2      i3      i4
871 h1  239  288  476  124
```

```
872 h2 30 100 707 1740
873 h3 0 63 84 469
874 h4 36 0 59 302
875
876 , , a4
877
878 i1 i2 i3 i4
879 h1 546 780 454 115
880 h2 170 328 877 647
881 h3 39 62 106 114
882 h4 0 28 24 50
883
884
885 $"1313"
886 , , a1
887
888 i1 i2 i3 i4
889 h1 324 195 78 49
890 h2 207 402 636 51
891 h3 40 123 45 154
892 h4 0 63 155 18
893
894 , , a2
895
896 i1 i2 i3 i4
897 h1 1110 2174 2766 920
898 h2 311 1084 2726 3989
899 h3 291 494 1142 2457
900 h4 393 1213 1904 4696
901
902 , , a3
903
904 i1 i2 i3 i4
905 h1 328 719 651 216
906 h2 15 158 719 1298
907 h3 25 24 195 321
908 h4 0 45 103 147
909
910 , , a4
911
912 i1 i2 i3 i4
913 h1 902 1222 886 348
914 h2 91 696 1117 1472
915 h3 16 25 209 200
916 h4 0 0 131 63
917
918
```

R Code

```

1 # pums5pct.R
2 # This script creates three way classifications of PUMS 2000 Data (5% small areas)
3 # for household size by household income by age of head of household.
4 # Ben Stabler benjamin.stabler@odot.state.or.us 9/29/03
5
6
7 # Set working directory to PUMS folder
8 setwd("F:\\_ben\\census\\pums2000")
9
10 #-----
11 # PUBLIC USE MICRODATA AREA
12 # Output list of PUM Areas by County Name
13 #-----
14 # Read in Public Use Microdata Area Definitions
15 pumasDf <- read.fwf("PUMEQ5-OR.txt", skip=42, widths=c(4,3,6,6,4,6,6,2,7,5,5,9,62))
16
17 # Grab County Names Only
18 pumasCountyDf <- pumasDf[pumasDf[,1]==781,]
19 pumasCountyDf <- pumasCountyDf[!is.na(pumasCountyDf[,1]),]
20
21 # Grab PUMA, Population, and Name
22 pumasCountyDf <- pumasCountyDf[,c(4,12,13)]
23 colnames(pumasCountyDf) <- c("PUMA","Population","Name")
24 pumasCountyDf[,3] <- gsub(" +$", "", as.character(pumasCountyDf[,3]))
25
26 # Create Tables by PUMA
27 pumaTables <- by(pumasCountyDf, pumasCountyDf$PUMA, function(x) x)
28
29 # Output Table to File
30 sink("pumas.txt")
31 pumaTables
32 sink()
33
34 #-----
35 # PUMS HOUSEHOLD DATA
36 # Create a data frame of the select fields from the HOUSEHOLD records
37 #-----
38 # Read in PUMS data
39 pums5.41 <- scan("Pums5_41.txt", what=character(), sep="\n")
40
41 # Grab only household records
42 hPums <- pums5.41[grep("H",pums5.41)]
43
44 #Define the columns of interest starts and ends of the fixed width format strings
45 record
46 #See pums.pdf for more information
47 #"PUMA" = Public Use Microdata Area
48 #"SERIALNO" = Serial number (to tie to Person records)
49 #"HWEIGHT" = Housing Weight
50 #"PERSONS" = Number of Persons in Household
51 #"VEHICL" = Number of Vehicles Available
52 #"HINC" = Total Household Income in 1999
53 #"WIF" = Number of Workers in Family
54 starts <- c(2,14,102,106,134,251,247)
55 ends <- c(8,18,105,107,134,258,247)
56
57 #Define a dummy matrix to hold the results
58 housingDataMtx <- matrix(0,length(hPums), length(starts))
59 colnames(housingDataMtx) <-
60 c("SERIALNO","PUMA","HWEIGHT","PERSONS","VEHICL","HINC","WIF")
61
62 #Loop through all the records and split the strings at the specified starts and ends
63 #Then assign the strings to the output matrix
64 for (i in 1:length(hPums)) { housingDataMtx[i,] <-
65   as.numeric(substring(hPums[i],starts,ends)) }
66 rm(hPums)
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

```



```

65 #Replace 0 household weights with 1 in order to avoid dropping the records that
66 #have a 0 for their household weight when creating the joint distribution tables
67 housingDataDf <- data.frame(housingDataMtx)
68 rm(housingDataMtx)
69 housingDataDf$HWEIGHT[housingDataDf$HWEIGHT==0] <- 1
70
71 #Classify the household records according to our HH size (1,2,3,4+) and assign the
  factor level codes to new fields
72 housingDataDf$HHSIZE <-
  cut(housingDataDf$PERSONS,c(0.5,1.5,2.5,3.5,max(housingDataDf$PERSONS)+1), labels=F)
73
74 #USE EITHER MPO OR NON-MPO HOUSEHOLD INCOME CLASSES
75 #Classify the household record according to nonMPO HH income groups (0-15k,
  15-25k, 25-40k, 40k+)
76 housingDataDf$HHINC <- cut(housingDataDf$HINC,
  c(-1,15001,25001,40001,max(housingDataDf$HINC)+1), labels=F)
77
78 #Classify the household record according to MPO HH income groups (0-15k,
  15-30k, 30-60k, 60k+)
79 housingDataDf$HHINC <- cut(housingDataDf$HINC,
  c(-1,15001,30001,60001,max(housingDataDf$HINC)+1), labels=F)
80
81 #-----
82 # PUMS PERSONS DATA
83 # Create a data frame of the select fields from the PERSONS records
84 #-----
85
86 # Grab only household records
87 pPums <- pums5.41[grep("P",pums5.41)]
88
89 #Define the columns of interest starts and ends of the fixed width format strings
  record
90 #See pums.pdf for more information
91 #"SERIALNO" = Serial number for HH record (and PERSON records tied to that HH).
92 #"PNUM" = Person sequence number (in HH)
93 #"PWEIGHT" = Person weight
94 #"RELATE" = Relationship of person to person 1 (householder)
95 #"AGE" = Age of person
96 starts <- c(2,9,13,17,25)
97 ends <- c(8,10,16,18,26)
98
99 #Define a dummy matrix to hold the results
100 outputDf <- matrix(0,length(pPums),length(starts))
101 colnames(outputDf) <- c("SERIALNO","PNUM","PWEIGHT","RELATE","AGE")
102
103 #Loop through all the records and split the strings at the specified starts and ends
104 #Then assign the strings to the output matrix
105 for (i in 1:length(pPums)) { outputDf[i,] <-
  as.numeric(substring(pPums[i],starts,ends)) }
106 rm(pPums)
107
108 #Grab only those records with PNUM equal to one (householder)
109 outputDf <- data.frame(outputDf)
110 outputDf <- outputDf[outputDf$PNUM==1,]
111
112 #Replace person weights of 0 with 1 so they don't drop out
113 outputDf$PWEIGHT[outputDf$PWEIGHT==0] <- 1
114
115 #Classify the PERSON records according to our age definition
116 #(15-24, 25-54, 55-64, 65+) and assign the factor level codes to new fields
117 outputDf$AGE <- cut(outputDf$AGE,c(14.5,24.5,54.5,64.5,max(outputDf$AGE)+1),
  labels=F)
118
119 #Attach Age of Householder to Housing Records
120 housingIndex <- match(housingDataDf$SERIALNO, outputDf$SERIALNO)
121 housingDataDf$AGE <- outputDf$AGE[housingIndex]
122

```

```
123 #-----
124 # CREATE CROSS CLASSIFICATIONS
125 # Create HH size by income by age of head
126 #-----
127
128 #Split all Oregon records into a list of data frames split by PUMA
129 pumsLst <- split(housingDataDf, housingDataDf$PUMA)
130
131 #Function to create 3way Cross Class
132 crossClass <- function(xDf) {
133     size <- rep(xDf$HHSIZE, xDf$HWEIGHT)
134     income <- rep(xDf$HHINC, xDf$HWEIGHT)
135     age <- rep(xDf$AGE, xDf$HWEIGHT)
136     siaAry <- table(size, income, age)
137     dimnames(siaAry) <-
138     list(c("h1", "h2", "h3", "h4"), c("i1", "i2", "i3", "i4"), c("a1", "a2", "a3", "a4"))
139     siaAry
140 }
141 #Apply to PUMA list
142 pumsLst <- lapply(pumsLst, crossClass)
143
144 #Output the HH size, income and age joint distribution tables to a text file
145 sink("puma00_hhsizeincage.txt")
146 print(pumsLst)
147 sink()
148
149 #Collapse 3D to 2D keeping household size and household income (for non-MPO models)
150 #Note - make sure to use non-MPO income classes above
151 pums2dLst <- lapply(pumsLst, function(x) apply(x, c(1,2), sum))
152
153 #Output the HH size and income joint distribution tables to a text file
154 sink("puma00_hhsizeinc.txt")
155 print(pums2dLst)
156 sink()
```