# 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:

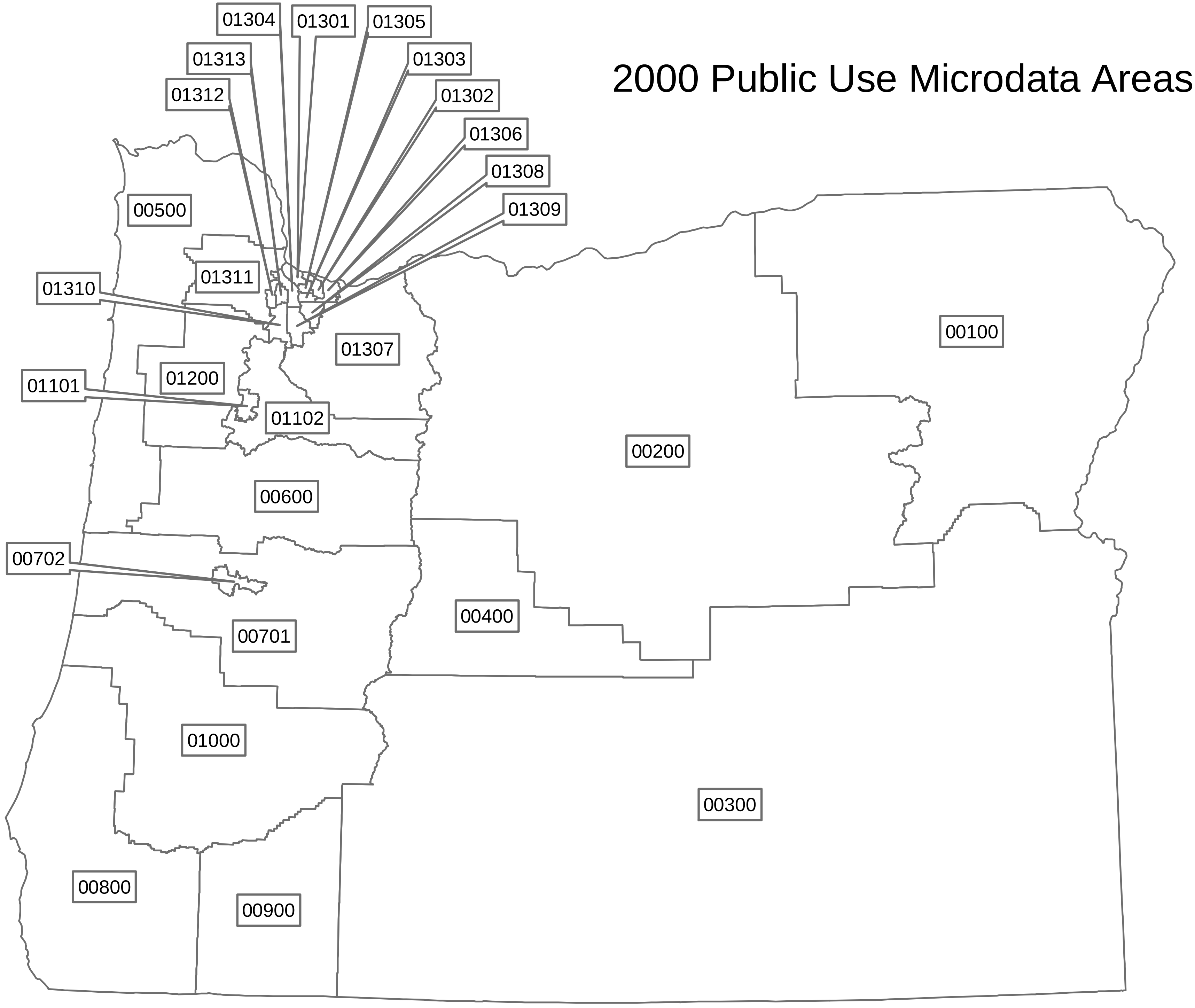
| Variable Class | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| (h) Household Size | 1 | 2 | 3 | 4+ |
| (i) Household Income (Non-MPO) | 0-15k | 15-25k | 25-40k | 40k+ |
| (i) Household Income (MPO) | 0-15k | 15-30k | 30-60k | 60k+ |
| (a) Age of Head of Household | 0-24 | 25-54 | 55-64 | 65+ |

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)

```
$"100"
     i1   i2   i3   i4
h1 5420 2719 1973 1318
h2 2325 2532 3877 6879
h3  859  706 1340 3770
h4  793 1205 2856 6288

$"200"
     i1   i2   i3   i4
h1 4012 1983 1781 1395
h2 1767 2239 3986 7405
h3  632  781 1548 3229
h4  826  967 2407 5885

$"300"
     i1   i2   i3   i4
h1 5133 2450 1703 1067
h2 2187 2808 3638 6626
h3  917 1089 1285 2967
h4  989 1434 2389 5087

$"400"
     i1   i2   i3   i4
h1 3346 2020 2460 2115
h2 1285 2097 3958 9796
h3  513  657 1552 4752
h4  496  750 2008 7137

$"500"
     i1   i2   i3    i4
h1 6885 4100 3058  2588
h2 2412 3485 5548 11918
h3  996  916 1652  5321
h4  807 1103 2074  7888

$"600"
     i1   i2   i3    i4
h1 7090 4099 3395  2902
h2 2910 3549 5314 14015
h3 1225 1027 1738  6472
h4  955 1402 3153 10760

$"701"
     i1   i2   i3   i4
h1 4046 1960 2182 1688
h2 1648 2714 4303 8761
h3  412  689 1347 3618
h4  764  775 1780 6302

$"702"
      i1   i2   i3    i4
h1 10128 5651 5229  4050
h2  5011 4888 7044 14963
h3  1447 1782 2623  7531
h4  1402 1645 2876 11108

$"800"
     i1   i2   i3    i4
h1 9150 3941 3081  2166
h2 3830 5117 6756 11439
h3 1147 1420 2114  4225
h4 1480 1547 2374  6445

$"900"
     i1   i2   i3    i4
h1 7265 4580 3428  2872
h2 2841 3998 6822 13110
```

```
h3 1400 1320 2083  6429
h4  981 1444 3398  9408

$"1000"
     i1   i2   i3   i4
h1 4643 2138 1538 1151
h2 1951 2877 4428 6545
h3  587  863 1655 3248
h4  572 1031 2037 5364

$"1101"
     i1   i2   i3    i4
h1 5565 3826 4407  3596
h2 2394 2659 4661 10902
h3 1053  942 2069  5485
h4 1469 1722 2848  9304

$"1102"
     i1   i2   i3   i4
h1 2798 1613 1605 1424
h2 1252 1214 2888 8465
h3  489  695  881 3944
h4  542  848 2500 7539

$"1200"
     i1   i2   i3    i4
h1 4314 1968 2638  2049
h2 1683 2435 3876 10760
h3  978  946 1422  5459
h4  673 1074 2297 10157

$"1301"
     i1   i2   i3   i4
h1 4005 2443 2775 2129
h2 1216 1467 2749 7738
h3  666  812 1058 3925
h4 1038 1082 1896 4797

$"1302"
     i1   i2   i3   i4
h1 3247 2734 2940 1963
h2 1387 1769 3418 6822
h3  583  803 1320 3491
h4  702  658 1633 5927

$"1303"
     i1   i2   i3   i4
h1 4666 3124 3018 2367
h2 1105 1576 3670 8245
h3  377  601 1354 3949
h4  473  615 1388 5122

$"1304"
     i1   i2   i3    i4
h1 8677 4025 4768  6753
h2 1308 1119 2413 12081
h3  282  554  508  4069
h4  104  107  363  5403

$"1305"
     i1   i2   i3    i4
h1 5433 3527 5045  4085
h2 1140 1836 3488 10578
h3  432  466 1216  4734
h4  457  568  732  5785

$"1306"
```

```
     i1   i2   i3   i4
h1 2739 2482 2381 2487
h2 1016 1330 2955 9017
h3  453  628 1559 4807
h4  787 1095 2072 7444

$"1307"
     i1   i2   i3   i4
h1 2044 1096 1496 1341
h2  616  967 3005 9341
h3  346  383  862 4604
h4  432  408 1376 8790

$"1308"
     i1   i2   i3   i4
h1 3222 2811 3360 3014
h2  976 1517 3606 9903
h3  543  497 1460 5593
h4  654  377 1674 8542

$"1309"
     i1   i2   i3    i4
h1 2141 1671 2592  2828
h2  542 1120 2294 11209
h3  329  346  658  4917
h4  122  422 1084  9458

$"1310"
     i1   i2   i3    i4
h1 2550 2367 3430  3156
h2  852 1220 2286 10930
h3  200  471  966  5333
h4  152  544 1073  8276

$"1311"
     i1   i2   i3    i4
h1 3040 2099 3004  2890
h2  923 1485 2537 11767
h3  423  455 1144  6123
h4  689 1119 2202 10831

$"1312"
     i1   i2   i3   i4
h1 1554 1509 2814 2774
h2  736  896 2035 9448
h3  293  501  916 5956
h4  328  541  970 9803

$"1313"
     i1   i2   i3    i4
h1 2664 2923 3367  3934
h2  624 1262 2840 10246
h3  372  333  802  4254
h4  393  796 1434  6308

```

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

```
$"100"
, , a1

     i1   i2   i3   i4
h1  543  152   39    0
h2  404  188  167    0
h3  212  263  250   37
h4   30  182  221   85

, , a2

     i1   i2   i3   i4
h1 1431 1316 1233  145
h2  854 1161 2247 1615
h3  508  550 1712 1662
h4  679 1750 4264 3114

, , a3

     i1   i2   i3   i4
h1  813  427  369  107
h2  373  551 1588 1159
h3   81  166  293  378
h4   48   87  197  252

, , a4

     i1   i2   i3   i4
h1 2633 1670  442  110
h2  694 2118 1904  590
h3   58  149  228  128
h4   36   76   76   45


$"200"
, , a1

     i1   i2   i3   i4
h1  154   78   13    0
h2  256  145  398   40
h3  125  160  272   24
h4   39  118  153   59

, , a2

     i1   i2   i3   i4
h1 1021 1176  968  135
h2  520 1136 2877 1592
h3  449  913 1753 1195
h4  708 1599 4017 2495

, , a3

     i1   i2   i3   i4
h1  530  513  318  171
h2  420  574 1378 1173
h3    6  116  206  341
h4   55   39  175  199

, , a4

     i1   i2   i3   i4
h1 2307 1018  584  185
h2  571 1632 2030  655
h3   52  115  281  182
h4   24   54  225  126

```

```

$"300"
, , a1

     i1   i2   i3   i4
h1  320   60   27    6
h2  207  293  180    0
h3  143  134   70   57
h4   58   53  205   34

, , a2

     i1   i2   i3   i4
h1 1347 1375  999  113
h2  864 1150 2328 1411
h3  660 1203 1846 1056
h4  907 2098 3609 2297

, , a3

     i1   i2   i3   i4
h1  866  534  124  105
h2  492  782 1124 1145
h3   90   94  280  233
h4   12   55  157  216

, , a4

     i1   i2   i3   i4
h1 2600 1309  412  156
h2  624 1950 1877  832
h3   24   98  208   62
h4   12   16  111   59


$"400"
, , a1

     i1   i2   i3   i4
h1  330  148   30    0
h2   67  391  300   28
h3   85  147  202   61
h4   28  141  174    0

, , a2

     i1   i2   i3   i4
h1 1179 1210 1505  342
h2  466  928 3076 2836
h3  360  917 2062 2340
h4  391 1039 3689 3991

, , a3

     i1   i2   i3   i4
h1  333  423  451  148
h2  415  525 1236 1589
h3   68   18  300  355
h4   52   63  190  325

, , a4

     i1   i2   i3   i4
h1 1504 1411  728  199
h2  337 1735 1824 1383
h3    0   97  268  194
h4   25   64   44  175
```

```


$"500"
, , a1

     i1   i2   i3   i4
h1  313  256   61   32
h2  227  239  298   42
h3  231  139  317   27
h4   94  116  140   18

, , a2

     i1   i2   i3   i4
h1 2027 2184 1678  409
h2 1096 1592 3168 3175
h3  695  967 2455 2171
h4  704 1436 4387 4020

, , a3

     i1   i2   i3   i4
h1 1116  538  700  138
h2  402 1070 2546 2029
h3   61   91  374  514
h4    9   19  197  387

, , a4

     i1   i2   i3   i4
h1 3429 2494  998  258
h2  687 2484 2768 1540
h3    9  179  316  339
h4    0   83   75  187


$"600"
, , a1

     i1   i2   i3   i4
h1 1259  128   33    0
h2 1106  917  568   24
h3  379  266  469   50
h4  123  270  475   59

, , a2

     i1   i2   i3   i4
h1 1873 2370 2438  796
h2 1021 1613 4623 3914
h3  735 1009 2217 3327
h4  820 1895 5599 5885

, , a3

     i1   i2   i3   i4
h1  781  625  401  203
h2  271  571 1795 2059
h3   86  118  251  789
h4   12  190   87  375

, , a4

     i1   i2   i3   i4
h1 3177 2435  606  361
h2  512 2289 2755 1750
h3   25  175  320  246
```

```
h4    0   33  112  335


$"701"
, , a1

   i1   i2   i3   i4
h1  154   46    0    0
h2   74  243  268   27
h3   49   78  132   12
h4   43    0   99   40

, , a2

   i1   i2   i3   i4
h1 1137  962 1309  198
h2  533  816 2725 2229
h3  330  850 1655 1589
h4  657 1219 3488 3541

, , a3

   i1   i2   i3   i4
h1  633  488  375  165
h2  298  748 1632 1572
h3    0   82  312  285
h4   49   34   36  197

, , a4

   i1   i2   i3   i4
h1 2122 1378  664  245
h2  743 2278 2443  797
h3   33   79  442  138
h4   15    0   82  121


$"702"
, , a1

   i1   i2   i3   i4
h1 2200  521  137    0
h2 2022 1124  736  119
h3  672  849  615   47
h4  270  402  437  164

, , a2

   i1   i2   i3   i4
h1 3871 3424 3551  700
h2 2082 2963 6267 4001
h3  647 1722 3519 3275
h4 1082 1893 5517 6255

, , a3

   i1   i2   i3   i4
h1  981  843  831  572
h2  403  845 1795 1967
h3   31  177  309  607
h4   50   61  372  310

, , a4

   i1   i2   i3   i4
h1 3076 2604 1276  471
h2  504 2304 3075 1699
```

```
h3   97  204  303  309
h4    0   24   86  108


$"800"
, , a1

     i1   i2   i3   i4
h1  370  233   24    0
h2  321  386  124   70
h3  241  101  178   27
h4  172  106  132    0

, , a2

     i1   i2   i3   i4
h1 2128 1636 1535  339
h2 1326 1671 3814 2508
h3  823 1469 2333 1584
h4 1240 2012 4260 2678

, , a3

     i1   i2   i3   i4
h1 1408  827  684  158
h2  839 1109 2287 1764
h3    6  151  261  382
h4   25  194  273  202

, , a4

     i1   i2   i3   i4
h1 5244 2316 1072  364
h2 1344 4038 4108 1433
h3   77  319  713  241
h4   43   91  278  140


$"900"
, , a1

     i1   i2   i3   i4
h1  515  152   85   25
h2  447  581  572   88
h3  184  113  493    0
h4   30  273  225   77

, , a2

     i1   i2   i3   i4
h1 2110 2243 1799  390
h2 1083 2130 3393 3278
h3 1029 1719 2609 2713
h4  871 2113 5335 5020

, , a3

     i1   i2   i3   i4
h1 1125  990  512  232
h2  488 1017 2304 1978
h3  130  113  521  735
h4   80  103  223  344

, , a4

     i1   i2   i3   i4
h1 3515 2666 1297  489
```

```
h2  823 2652 3844 2093
h3   57  108  341  367
h4    0  110  168  259


$"1000"
, , a1

     i1   i2   i3   i4
h1  185  105   27    0
h2  167  114  225   30
h3  128  188  213   45
h4    6   88  165   27

, , a2

     i1   i2   i3   i4
h1 1283  894  793   90
h2  674 1050 2311 1453
h3  359  869 1674 1300
h4  506 1271 3249 2625

, , a3

     i1   i2   i3   i4
h1  574  504  395  113
h2  417 1001 1581 1022
h3   42  172  282  316
h4   60  126  268  202

, , a4

     i1   i2   i3   i4
h1 2601 1252  427  227
h2  693 2170 2269  624
h3   58  154  357  196
h4    0   78  162  171


$"1101"
, , a1

     i1   i2   i3   i4
h1  580  217   27    0
h2  850  476  653   83
h3  366  243  432   95
h4  165  436  367  129

, , a2

     i1   i2   i3   i4
h1 1797 2616 2530  601
h2  960 1646 3534 3208
h3  638 1075 2918 2397
h4 1200 2219 5227 4789

, , a3

     i1   i2   i3   i4
h1  704  813  911  218
h2  215  438 1408 1543
h3   24  153  123  557
h4   85   48  183  317

, , a4

     i1   i2   i3   i4
```

```
h1 2484 2390 1168  338
h2  369 1768 2340 1125
h3   25  167  102  234
h4   19    0   88   71


$"1102"
, , a1

     i1   i2   i3   i4
h1  189  177    0    0
h2  143   88  175   80
h3   51  284   71   69
h4   61  180  127   36

, , a2

     i1   i2   i3   i4
h1  388  702 1005  228
h2  336  518 1782 2393
h3  376  545 1584 1886
h4  394 1474 4007 4360

, , a3

     i1   i2   i3   i4
h1  349  370  350  142
h2  199  457 1223 1332
h3   37   43  167  371
h4   81   28   94  250

, , a4

     i1   i2   i3   i4
h1 1872 1072  471  125
h2  574 1062 2232 1225
h3   25   73  298  129
h4    6    6  176  149


$"1200"
, , a1

     i1   i2   i3   i4
h1  403   68   37    0
h2  485  461  458   19
h3  317  256  261   33
h4   60   91  220   44

, , a2

     i1   i2   i3   i4
h1 1044 1131 1644  454
h2  675 1139 3225 3116
h3  627  575 2571 2427
h4  580 1697 4834 5509

, , a3

     i1   i2   i3   i4
h1  529  421  574  103
h2  139  255 1201 1882
h3   28   92  372  450
h4    0  114  221  393

, , a4

```

```
   i1   i2   i3   i4
h1 2338 1318  736  169
h2  384 1943 2366 1006
h3    6  302  382  106
h4   33   54  193  158


$"1301"
, , a1

   i1   i2   i3   i4
h1  226  128   19    0
h2  211  238  134   61
h3  112  229  259   15
h4  103   83  165   52

, , a2

   i1   i2   i3   i4
h1 1610 1660 2388  453
h2  603 1188 3508 2694
h3  539  524 1872 1650
h4  850 1386 3045 2316

, , a3

   i1   i2   i3   i4
h1  590  556  490  150
h2  167  359  738  734
h3   15  119  190  243
h4   24  172  162   96

, , a4

   i1   i2   i3   i4
h1 1579 1027  369  107
h2  235  681 1069  550
h3    0  218  238  238
h4   61   74  111  113


$"1302"
, , a1

   i1   i2   i3   i4
h1  312  149   65    0
h2  162  276  223   75
h3  149  289  158   99
h4   46   82  138   46

, , a2

   i1   i2   i3   i4
h1 1057 1612 1842  410
h2  759 1161 2302 1967
h3  358  771 1722 1455
h4  589 1104 3528 2674

, , a3

   i1   i2   i3   i4
h1  472  506  446   36
h2  148  223 1009  523
h3    0   64  277  187
h4   43   39   79  259

, , a4
```

```

     i1   i2   i3   i4
h1 1406 1807  581  183
h2  318 1025 2109 1116
h3   76  154  265  173
h4   24   30  158   81


$"1303"
, , a1

     i1   i2   i3   i4
h1  327  172    0    0
h2  238  438  433   40
h3  108   83  217   66
h4  134   69  209   52

, , a2

     i1   i2   i3   i4
h1 1601 2146 2609  633
h2  426 1450 4045 3138
h3  269  746 2190 1519
h4  300  836 2552 2665

, , a3

     i1   i2   i3   i4
h1  543  467  503  117
h2  136  243  525  614
h3    0   67  264  227
h4   24    0  279  268

, , a4

     i1   i2   i3   i4
h1 2195 1323  481   58
h2  305  777 1283  505
h3    0  127  176  222
h4   15    0   68  127


$"1304"
, , a1

     i1   i2   i3   i4
h1 1993  679  181    0
h2  522  615  382   74
h3   36  154  135  115
h4   22   21   81  102

, , a2

     i1   i2   i3   i4
h1 3919 3698 3670 2377
h2  467  868 3035 5098
h3  209  409  960 2489
h4   82  171  836 4352

, , a3

     i1   i2   i3   i4
h1  881  501  799  488
h2   56  278  377 2165
h3    0   81   86  398
h4    0    0   65  170

```

```
, , a4

     i1   i2   i3   i4
h1 1884 1090 1342  721
h2  263  250  894 1577
h3   37    0   34  270
h4    0    0   21   54


$"1305"
, , a1

     i1   i2   i3   i4
h1  506  278   39    0
h2  278  381  558   91
h3   90  168  223  140
h4   49   80  231   54

, , a2

     i1   i2   i3   i4
h1 2359 2987 4091 1255
h2  478 1397 3929 4655
h3  309  528 1883 2412
h4  377  607 1818 3706

, , a3

     i1   i2   i3   i4
h1  713  522  506  177
h2  162  226  555 1138
h3   33   48  191  383
h4   31  118  119  148

, , a4

     i1   i2   i3   i4
h1 1855 1655  845  302
h2  222  802 1458  712
h3    0   67  136  237
h4    0   30   64  110


$"1306"
, , a1

     i1   i2   i3   i4
h1  225  251   18    0
h2  165  365  537   45
h3  113  156  384   70
h4   59  250  274   36

, , a2

     i1   i2   i3   i4
h1  846 1488 2487  500
h2  592  868 3012 3366
h3  340  788 2244 2347
h4  695 1423 3047 4898

, , a3

     i1   i2   i3   i4
h1  241  524  364  147
h2  129  204  915  936
h3    0   94  201  252
h4   18   24  131  264
```

```

, , a4

   i1   i2   i3   i4
h1 1427 1096  354  121
h2  130  705 1652  697
h3    0  105  212  141
h4   15   64  110   90


$"1307"
, , a1

   i1   i2   i3   i4
h1   96   93   66    0
h2   49  116  175   58
h3   54   63   42   79
h4    0   27   71   31

, , a2

   i1   i2   i3   i4
h1  667  497 1128  392
h2  175  549 2200 3265
h3  258  404 1706 2405
h4  325  684 3369 5249

, , a3

   i1   i2   i3   i4
h1  310  274  198  105
h2  111  480 1528 1879
h3    0   43  290  516
h4  107   68  180  400

, , a4

   i1   i2   i3   i4
h1  971  735  403   42
h2  281  924 1475  664
h3   34  108   74  119
h4    0   66  276  153


$"1308"
, , a1

   i1   i2   i3   i4
h1  169  305   56   28
h2  147  338  540  146
h3  103   46  184   75
h4  119   53  186    0

, , a2

   i1   i2   i3   i4
h1  896 1609 2238  745
h2  420  763 2747 3478
h3  337  664 2063 3041
h4  516  922 3517 4992

, , a3

   i1   i2   i3   i4
h1  417  631  496   72
h2  219  403 1260 1345
h3   18   33  546  359
```

```
h4   19    0  237  358

, , a4

   i1   i2   i3   i4
h1 1740 1621 1048  336
h2  190 1303 1841  862
h3   85   33  287  219
h4    0   15  161  152


$"1309"
, , a1

   i1   i2   i3   i4
h1   67  155   65    0
h2  157  183  304  105
h3   21  130   85   36
h4   19   81  126   13

, , a2

   i1   i2   i3   i4
h1  675  932 2086  693
h2  198  625 2150 4874
h3  219  443  912 3189
h4  103  536 2395 7192

, , a3

   i1   i2   i3   i4
h1  287  211  305  200
h2  109  174 1211 1812
h3   37    0  123  724
h4    0   61   57  284

, , a4

   i1   i2   i3   i4
h1 1112 1445  762  237
h2   78  793 1137 1255
h3   52    0  113  166
h4    0   16  147   56


$"1310"
, , a1

   i1   i2   i3   i4
h1  212  227   42   31
h2  193  269  580  110
h3   52  103  178   27
h4   27  125   59    0

, , a2

   i1   i2   i3   i4
h1  698 1547 2176  780
h2  439  652 2493 4322
h3  148  558 1496 3467
h4  125  762 2420 5848

, , a3

   i1   i2   i3   i4
h1  295  400  540  264
h2   52  232  684 2011
```

```
h3    0    0   81  511
h4    0   28   30  509

, , a4

     i1   i2   i3   i4
h1 1345 1822  792  332
h2  168  852 1148 1083
h3    0   22  199  128
h4    0    0   65   47


$"1311"
, , a1

     i1   i2   i3   i4
h1  303  439   80   37
h2   99  264  265  166
h3  116  140  337   41
h4   81  163  218  118

, , a2

     i1   i2   i3   i4
h1 1009 1579 2388  983
h2  376 1020 2371 4961
h3  289  461 2248 3213
h4  568 1315 4961 6497

, , a3

     i1   i2   i3   i4
h1  332  205  539  114
h2  231  372  963 2078
h3    0   33  266  411
h4   15   33  156  408

, , a4

     i1   i2   i3   i4
h1 1396 1050  432  147
h2  217  856 1442 1031
h3   18   78  299  195
h4   25   21   67  195


$"1312"
, , a1

     i1   i2   i3   i4
h1  189  317   97   34
h2  143  346  530  116
h3   72  206  115   36
h4    0  103  133   99

, , a2

     i1   i2   i3   i4
h1  580 1176 2412  824
h2  393  924 2060 4004
h3  182  550 1858 3710
h4  292  736 2560 7220

, , a3

     i1   i2   i3   i4
h1  239  288  476  124
```

```
h2   30  100  707 1740
h3    0   63   84  469
h4   36    0   59  302

, , a4

    i1   i2   i3   i4
h1  546  780  454  115
h2  170  328  877  647
h3   39   62  106  114
h4    0   28   24   50


$"1313"
, , a1

    i1   i2   i3   i4
h1  324  195   78   49
h2  207  402  636   51
h3   40  123   45  154
h4    0   63  155   18

, , a2

    i1   i2   i3   i4
h1 1110 2174 2766  920
h2  311 1084 2726 3989
h3  291  494 1142 2457
h4  393 1213 1904 4696

, , a3

    i1   i2   i3   i4
h1  328  719  651  216
h2   15  158  719 1298
h3   25   24  195  321
h4    0   45  103  147

, , a4

    i1   i2   i3   i4
h1  902 1222  886  348
h2   91  696 1117 1472
h3   16   25  209  200
h4    0    0  131   63
```

# R Code

```r
# pums5pct.R
# This script creates three way classifications of PUMS 2000 Data (5% small areas)
# for household size by household income by age of head of household.
# Ben Stabler benjamin.stabler@odot.state.or.us 9/29/03


# Set working directory to PUMS folder
setwd("F:\\_ben\\census\\pums2000")

#------------------------------------------------------------------------------------
# PUBLIC USE MICRODATA AREA
# Output list of PUM Areas by County Name
#------------------------------------------------------------------------------------
# Read in Public Use Microdata Area Definitions
pumasDf <- read.fwf("PUMEQ5-OR.txt", skip=42, widths=c(4,3,6,6,4,6,6,2,7,5,5,9,62))

# Grab County Names Only
pumasCountyDf <- pumasDf[pumasDf[,1]==781,]
pumasCountyDf <- pumasCountyDf[!is.na(pumasCountyDf[,1]),]

# Grab PUMA, Population, and Name
pumasCountyDf <- pumasCountyDf[,c(4,12,13)]
colnames(pumasCountyDf) <- c("PUMA","Population","Name")
pumasCountyDf[,3] <- gsub(" +$","",as.character(pumasCountyDf[,3]))

# Create Tables by PUMA
pumaTables <- by(pumasCountyDf, pumasCountyDf$PUMA, function(x) x)

# Output Table to File
sink("pumas.txt")
pumaTables
sink()

#------------------------------------------------------------------------------------
# PUMS HOUSEHOLD DATA
# Create a data frame of the select fields from the HOUSEHOLD records
#------------------------------------------------------------------------------------
# Read in PUMS data
pums5.41 <- scan("Pums5_41.txt", what=character(), sep="\n")

# Grab only household records
hPums <- pums5.41[grep("H",pums5.41)]

#Define the columns of interest starts and ends of the fixed width format strings record
#See pums.pdf for more information
#"PUMA" = Public Use Microdata Area
#"SERIALNO" = Serial number (to tie to Person records)
#"HWEIGHT" = Housing Weight
#"PERSONS" = Number of Persons in Household
#"VEHICL" = Number of Vehicles Available
#"HINC" = Total Household Income in 1999
#"WIF" = Number of Workers in Family
starts <- c(2,14,102,106,134,251,247)
ends <- c(8,18,105,107,134,258,247)

#Define a dummy matrix to hold the results
housingDataMtx <- matrix(0,length(hPums), length(starts))
colnames(housingDataMtx) <- c("SERIALNO","PUMA","HWEIGHT","PERSONS","VEHICL","HINC","WIF")

#Loop through all the records and split the strings at the specified starts and ends
#Then assign the strings to the output matrix
for (i in 1:length(hPums)) {  housingDataMtx[i,] <- as.numeric(substring(hPums[i],starts,ends))  }
rm(hPums)

```

```r
#Replace 0 household weights with 1 in order to avoid dropping the records that
#have a 0 for their household weight when creating the joint distribution tables
housingDataDf <- data.frame(housingDataMtx)
rm(housingDataMtx)
housingDataDf$HWEIGHT[housingDataDf$HWEIGHT==0] <- 1

#Classify the household records according to our HH size (1,2,3,4+) and assign the factor level codes to new fields
housingDataDf$HHSIZE <- cut(housingDataDf$PERSONS,c(0.5,1.5,2.5,3.5,max(housingDataDf$PERSONS)+1), labels=F)

#USE EITHER MPO OR NON-MPO HOUSEHOLD INCOME CLASSES
        #Classify the household record according to nonMPO HH income groups (0-15k, 15-25k, 25-40k, 40k+)
        housingDataDf$HHINC <- cut(housingDataDf$HINC, c(-1,15001,25001,40001,max(housingDataDf$HINC)+1), labels=F)

        #Classify the household record according to MPO HH income groups (0-15k, 15-30k, 30-60k, 60k+)
        housingDataDf$HHINC <- cut(housingDataDf$HINC, c(-1,15001,30001,60001,max(housingDataDf$HINC)+1), labels=F)

#-------------------------------------------------------------------------------------
# PUMS PERSONS DATA
# Create a data frame of the select fields from the PERSONS records
#-------------------------------------------------------------------------------------

# Grab only household records
pPums <- pums5.41[grep("P",pums5.41)]

#Define the columns of interest starts and ends of the fixed width format strings record
#See pums.pdf for more information
#"SERIALNO" = Serial number for HH record (and PERSON records tied to that HH).
#"PNUM" = Person sequence number (in HH)
#"PWEIGHT" = Person weight
#"RELATE" = Relationship of person to person 1 (householder)
#"AGE" = Age of person
starts <- c(2,9,13,17,25)
ends <- c(8,10,16,18,26)

#Define a dummy matrix to hold the results
outputDf <- matrix(0,length(pPums),length(starts))
colnames(outputDf) <- c("SERIALNO","PNUM","PWEIGHT","RELATE","AGE")

#Loop through all the records and split the strings at the specified starts and ends
#Then assign the strings to the output matrix
for (i in 1:length(pPums)) {  outputDf[i,] <- as.numeric(substring(pPums[i],starts,ends))  }
rm(pPums)

#Grab only those records with PNUM equal to one (householder)
outputDf <- data.frame(outputDf)
outputDf <- outputDf[outputDf$PNUM==1,]

#Replace person weights of 0 with 1 so they don’t drop out
outputDf$PWEIGHT[outputDf$PWEIGHT==0] <- 1

#Classify the PERSON records according to our age definition
#(15-24, 25-54, 55-64, 65+) and assign the factor level codes to new fields
outputDf$AGE <- cut(outputDf$AGE,c(14.5,24.5,54.5,64.5,max(outputDf$AGE)+1), labels=F)

#Attach Age of Householder to Housing Records
housingIndex <- match(housingDataDf$SERIALNO, outputDf$SERIALNO)
housingDataDf$AGE <- outputDf$AGE[housingIndex]

```

```
#-------------------------------------------------------------------------------------
# CREATE CROSS CLASSIFICATIONS
# Create HH size by income by age of head
#-------------------------------------------------------------------------------------

#Split all Oregon records into a list of data frames split by PUMA
pumsLst <- split(housingDataDf, housingDataDf$PUMA)

#Function to create 3way Cross Class
crossClass <- function(xDf) {
        size <- rep(xDf$HHSIZE, xDf$HWEIGHT)
        income <- rep(xDf$HHINC, xDf$HWEIGHT)
        age <- rep(xDf$AGE, xDf$HWEIGHT)
        siaAry <- table(size, income, age)
        dimnames(siaAry) <-
list(c("h1","h2","h3","h4"),c("i1","i2","i3","i4"),c("a1","a2","a3","a4"))
        siaAry
}
#Apply to PUMA list
pumsLst <- lapply(pumsLst, crossClass)

#Output the HH size, income and age joint distribution tables to a text file
sink("puma00_hhsizeincage.txt")
print(pumsLst)
sink()


#Collapse 3D to 2D keeping household size and household income (for non-MPO models)
#Note - make sure to use non-MPO income classes above
pums2dLst <- lapply(pumsLst, function(x) apply(x, c(1,2), sum))

#Output the HH size and income joint distribution tables to a text file
sink("puma00_hhsizeinc.txt")
print(pums2dLst)
sink()
```