

Bridging the Censuses: An Object-Oriented Metadata-Driven Approach

By Curtis Mack, Krupski Consulting.

The VISTA group needed population estimates that were not supplied by either the Census Bureau or by the Washington State Office of Financial Management (OFM). What they needed were small area population estimates with a complete matrix of race, Hispanic ethnicity, detailed age, and sex. They needed these as a continuous series for every year 1990 to present so that they could be used for trend analysis. These estimates needed to agree with published OFM numbers at the county level and with 1990 and 2000 Census counts.

They came to us because we had done similar things in the 90's. The actual deliverable was the following:

Census Block Level population estimates for 1991 – 2003 by MARS OMB 77 race by Hispanic Ethnicity by Single Year of Age by Gender

Census Block Level population estimates for 2001 – 2003) by MARS OMB 97 race by Hispanic Ethnicity by Single Year of Age by Gender

They also needed all of those estimates by Census block group, tract, and county, as well as by school district, and historical Zip Codes 1990 – 2003.

However, there were two major obstacles to the creation of these estimates that did not exist in the 1990 census. One was that the federal Office of Management and Budget (OMB) had made major changes to the race definitions that the Census used in 2000. The other is that the Census chose in 2000 not to use a “hot decking” process to protect confidentiality at low geographic levels, and rather chose to use suppression of values in small geographic areas. What follows is how we address these problems methodologically and programmatically.

Working with Suppressed Data

In 2000 the Census Bureau did not provide a full demographic breakdown at any detailed level of geography. They didn't do this in 1990 either, but they got a lot closer, and they did some special runs for Claritas with a full breakdown that Washington State purchased. This was partially due to them going from 5 race groups to 64 but there will be more on that later.

In 2000, suppression was applied at different geographic levels depending on the population size of the geography. A complete multi-race breakdown was provided in Summary File 1 (SF1) at the block level, but it had no age or sex detail. The Non-Hispanic population is there with a full race breakdown, but not the Hispanic population. Using some basic subtraction, the best demography available at the block level is a full

multi-race breakout, by Hispanic ethnicity, by adult / child. We needed to go through several steps to get that age breakdown finer.

To start, for the single race only race groups, there is a twenty-three age group by sex break down at the block level. That breakdown is not available for the individual multi-race groups but it is for the combined “Two or More Races” category. We used these to distribute the adult / child groups from the earlier tables to these more detailed age groups. We used the distribution of the “Two or More Races” category for all of the multi-race categories. There is also, at the census tract level, a single year of age by sex table for the same simple race categories. We used that to further distribute the block level age groups to finally get an estimate at the block level with the full age, race, and sex detail we needed.

To further refine this estimate we chose to use some data from Summary File 2 (SF2). Before I go any further, for those of you that don’t spend hours digging into census data, this file has a little trick. If you look at the definition of this file’s tables in the documentation you will see a complete single year of age by sex break down, but nothing by race. The trick is that this file comes in several “characteristic iterations”. This means there are several different versions of the SF2 file, each of which only counts one race, ethnicity, or tribal group. This includes all of the detailed multi-race crossings as well as breakdowns for Hispanics and for the “Two or More” and “Three or More Race” groupings. It doesn’t however supply the Hispanic crossing for the individual multi-race categories, but only crosses Hispanic with the “Alone” races, and “Two or More” and “Three or More Race” groups. The real downside of this file however is its geography. The file does have census tract level values. Unfortunately it only gives them if the size of the population of that characteristic iteration, in that tract, meets the threshold of 100 people. If not, that characteristic iteration is not supplied for that tract, and one must go to the county level for that value. That is of course unless the county doesn’t have 100 people of that iteration either. Then they will suppress the county level. They even suppress the state level data for many of the iterations. We wanted to control our data so that it would agree with these published detail where they were given. Honestly, what we really wanted was to avoid somebody saying, “Why don’t your totals agree with this SF2 table? You did use the Census data right?” To avoid this irritation, we created a process that pulled all of this data and found the lowest level of geography at which each iteration was published for each tract. The process also subtracted all of the known lower level values from the higher levels, resulting in a balance to which we could control all the unpublished tracts.

Next came the “Raking”. “Raking” (I am going to stop using the quotes now) is slang for Iterative Proportional Fitting. Proportional Fitting is the process of taking your detailed estimates, and figuring out by how much they differ from a known (or more authoritative) set of published values at a higher level of aggregation. The slang for such a higher-level table is a “marginal”. You then create a ratio of your values to the marginal’s values for each of the categories provided by the marginal. You then apply that proportion to your data, and that potentially embarrassing disagreement goes away. Now comes the iterative part. Say there are more than one of those pesky marginal tables

out there. What is done then is you proportionally fit your data to each marginal in turn. Of course the second fitting will undo some the first fitting so you keep looping through them and hopefully your estimates will approach a happy compromise. This is not always the case, especially if you are unfortunate enough to have marginals that explicitly disagree. In that case you will need to choose which one to use. This occurred a couple of time in this process

We raked by several marginal tables in addition to the SF2 data described above. In total we raked to 9 different marginal totals many of which were themselves derived from more than one table. The marginals were:

- The full block level multi-race breakout, by Hispanic ethnicity, by adult / child mentioned earlier, that we were able to derive from SF1 using subtraction.
- The SF2 characteristic iterations data, also mentioned earlier, where we calculated to the lowest level of geography available for the detailed race, age, and sex breakdown at the tract level or higher.
- The same SF2 characteristic iterations data only for the “Two or More Race” group only.
- The same SF2 characteristic iterations data only for the “Three or More Race” group only.
- The same SF2 characteristic iterations data only for the Hispanic, “Two or More Race” group only.
- The SF1 Block level P012 table which is the Total Population by Single Year Age by Sex.
- The SF1 Block level P012 “H” table which is the Hispanic Population by 23 Age Groups by Sex.
- The SF1 Block level P012 “G” table which is the “Two or More Race” Population by 23 Age Groups by Sex.
- The SF1 Tract level PCT012 “H” table which is the Hispanic Population by Single Year Age by Sex.

The end result is that desired complete demographic breakdown that matches the published 2000 census as closely as we could get it.

Race Bridging

Once we had these detailed race values from the 2000 census, we then needed to convert them to the race codes used by the 1990 census and to those used in the marginal tables to which we needed to control. There are many different race standards being used today and it has become quite a problem. I was able to easily think of 10 different ones in common use. The five that were involved in this process were the following.

OMB 1977 (MARS) – Used by the 1990 Census MARS files.

White, Black, American Indian / Native Alaskan, Asian / Pacific Islander

Census OMB 1977 – The format used by the 1990 Census Summary Files.

White, Black, American Indian / Native Alaskan, Asian / Pacific Islander, Other Race

OMB 1997 (MARS) – Used by the 2000 Census MARS Files.

Any combination of

White, Black, American Indian / Native Alaskan, Asian, Native Hawaiian / Other Pacific Islander

Census OMB 1997 – Used by the 2000 Census Summary Files.

Any combination of

White, Black, American Indian / Native Alaskan, Asian, Native Hawaiian / Other Pacific Islander,
Some Other Race

OFM 2000 – Used by OFM in their 2000 and later population estimates.

White, Black, American Indian / Native Alaskan, Asian / Pacific Islander, Multi-Race

Some of these race definitions can be directly derived from one another. For example, one can easily get the OFM 2000 definition from the OMB 1997 definition by summing the people with more than one race selected into a single “multi-race” category, and by combining the “Asian” and “Native Hawaiian / Other Pacific Islander” people into one “Asian / Pacific Islander” category. However, converting from the Census OMB 1997 to the MARS OMB 1997 requires a few assumptions be made to allocate the people who selected “Other Race” to some other race group. The Census did this for the MARS files at the county level using data that is not available to the public so we chose the following technique. First, some people who selected “Other Race” also selected another race. For those people “Other Race” was removed from their list of selected races. For those who selected only “Other Race”, one of two things was done. If the person was Hispanic, they were put in the “White” category. If they were not Hispanic, they were randomly assigned to one of the race codes. There are some things more that could have been tried, but they would have required complex modeling.

Conversion from the multi-race OMB 1997 style races to the single race OMB 1977 style races was a much more complex process. Fortunately for us the National Center for Health Statistics (NCHS) put together one of those complex models and developed a method for doing this nationally at the county level. Better yet, they published the results, the methodology, and the coefficients they used. This NCHS Race bridging methodology uses data from the National Health Interview Survey, which collects race using both a single and a multi-race question. From it they derived a probability model used to generate the percent of each multi race group that would have chosen each of the single races if given only one choice. Take for example the Black/Asian group. It will calculate what percentage of the population if allowed to select only one race, would have selected Black and what percent would have selected Asian. It does this for every possible combination of race by age and sex, for each geographic area. The model is based on the age, sex, race, and Hispanic ethnicity, of the population as well as the region, and urbanization level of the county. One of the authors, Deborah Ingram, was very helpful in supplying us with some of the data not included in the report and in answering our non-statistician questions on how the model should be implemented. If you are interested in the details of their methods, they can be found at <http://www.cdc.gov/nchs/about/major/dvs/popbridge/popbridge.htm>

Inter Censual Estimation

We applied the NCHS model to our 2000 block level estimates while retaining the county level coefficients. In this way we got that complete demographic breakdown we needed; only now the race codes matched those used in the 1990 Census. Now the only difference left between the structures of the 1990 and 2000 estimates was the Census blocks. The Census made no effort to maintain a consistent definition for the blocks. The lowest level at which they do that is the tract. We therefore needed to estimate the 1990 population by 2000 census blocks. This was done using an areal overlay of the two sets of block boundaries using a Geographic Information System or GIS. The result is a row for each 1990 block for each 2000 block that it overlaps. Each row contains the area of that overlap. Using this we could proportionally divide the population of each 1990 block into each of the 2000 blocks it overlaps. When these were resumed by 2000 block, the results were 1990 estimates by 2000 blocks.

Producing the 1991 to 1999 estimates was then a relatively simple matter of assuming constant population change between the censuses for each block for each demographic sub-group. Once those were done, we raked the results to match the published OFM County by age and sex estimates for the corresponding years and got the final tables that VISTA needed for those years.

Post Census Estimation

The post census estimation process was not as rigorous as it could have been. OFM publishes county by age group and sex estimates for every year. In addition to that they published in 2002 county age and sex estimates that included a five race group breakdown. Those races are described in the earlier table, and they lump all multi-race people into a single category. The 2002 estimate was derived by re-raking the 2000 estimates to match those from the OFM 2002 county, race, age, sex, estimates. To get 2001, we split the difference between the 2000 and 2002 estimates, and rake the results to OFM's 2001 county, age, and sex estimate. For 2003, we just re-raked the 2002 estimate to OFM's 2003 county, age, and sex estimate.

And that was the methodology we used to create these estimates. The next section describes the coding technique that we employed to create them.

The innovative approach used in coding this solution.

Krupski Consulting, which took on the project, has adopted an innovative approach to all of its coding. This group has worked with analysts and statisticians for years and has seen that many of the same tasks have come up repeatedly. Often much of the work in their projects was in applying very similar tasks to the different structures found in each customer's data. In response to this and other aspects of their work, they have developed an innovative Object Oriented / Metadata driven approach to solving most of their coding tasks in an effort to improve re-use of code, and therefore improve the over-all quality of their methods.

Some of the original inspiration for this approach came from the books “Table-Driven Strategies for Rapid SAS® Application Development” and “Programming Techniques for Object-Based Statistical Analysis with SAS® Software” by Tanya Kolosova and Samuel Berestizhevsky both published by the SAS Institute. These are very interesting books that gave us the seed that started our thinking differently about coding. While we do recommend these books, none of the code or table designs from these texts fit into our eventual approach.

The key to the approach is all tables and columns are recorded in a set of metadata tables before they are created. Tables are then generated from the information in the metadata. This includes all column definitions and integrity constraints. The metadata tables were designed to emulate the object model as much as possible when working inside SAS. Columns are defined as classes. Tables are also defined as classes; only they are composite classes that contain a collection of component (column) classes. The rows in the resulting tables are considered the objects. SAS macros are written to work with certain classes whether they materialize as tables or columns. These macros are also recorded in the metadata and are associated with those specific classes. They become the “methods” of the system. The classes describe the nature of the data elements or structures, and the methods are associated with these particular data elements and structures. The methods are also written to respond differently based on the component classes of a collection class, or in other word to different types of columns within a table.

For the example, part of the class structure for dates columns looks like this:

```
...
Fact
  CatagoricalFact
    CatagoricalNumericFactOrdinal - Numeric
      TimePeriod
        UnAssignedDate - Date Length 8
        DateTimePointInTime - Time Length 8
          DateTimeStartOfSpanOfTime
          DateTimeEndOfSpanOfTime
          DateTimeCreated
          DateTimeOfBirth
        DayPeriodOfTime - Date Length 4
          DateCreated
          DateUpdated
          DateOfBirth
          CensusDate
          DateOnlyStartOfSpanOfTime
          DateOnlyEndOfSpanOfTime
        MonthPeriodOfTime - Numeric Length 3
        YearPeriodOfTime - Numeric Length 3
```

In this example you can see that “CensusDate”, which we will discuss more latter, inherits from “DayPeriodOfTime” from which it inherits its type and length. Going

farther up the tree it eventually inherits from “CatagoricalFact” which will be important to the methods we used in the population estimates. Other interesting examples in this hierarchy are the “DateCreated” and “DateUpdated” classes. The Krupski system is written so almost all basic data activities are handled through methods. The “LoadDbmsTable” method for example is used to load data into a table and to update values if they match on the key values. That method does many other things such as verifying integrity constraints. It is also sensitive to any column class that inherits from one of those two date classes. If such a column exists in a destination table, it is updated with the correct values automatically. The programmer need only add a column of one of those classes to a table, and they will get a column that always contains the date the row was added or the date it was last updated.

For this population estimation application several existing classes were used and a few new classes were defined. Here are some of the key classes.

“SuperNormalizedEntity” – This is an existing fundamental class of the system. It is a table type class and a table of this class must have one and only one component that inherits from the “OrdinalFact” class. All other columns must inherit from the “CategoricalFact” class. One of the more interesting methods associated with it is the “Raking” method which is very important to the estimation process.

“DemographicTable” – This is a new class defined for this process. It inherits from “SuperNormalizedEntity” and therefore inherits from it all of its methods including “Raking”. This class however has more restrictions. Tables of this class must have one component that inherits from each of the following classes:

- “CensusPopEst” – This is a descendent of the “OrdinalFact” class and meets that requirement of a “SuperNormalizedEntity”.
- “CensusDate” – Which is listed in the tree above and inherits from “CategoricalFact”.
- “CensusGeographyTable” – This is actually a table type class. By making it a component of another table type class, the primary key of the “CensusGeographyTable” table is included as a column in the “DemographicTable” table. The primary key of a “CensusGeographyTable” in turn inherits from “CategoricalFact” meeting the requirements.

Optionally a “DemographicTable” can also have components that inherit from

- “RaceDefinitionTable”
- ”AgeGroupDefinitionTable”
- “Hispanic”
- “Gender”.

Most of the new methods developed for the population estimates were associated with the DemographicTable class. The main methods were:

- **CensusLoadSfTable** – This method is used to load the data from a Census SF file into the system, and record its attributes in metadata. It is actually just representative of several methods that were written to read in all of the different data sources. These methods use hand created tables that describe the demographic and geographic classes of all of the incoming fields, and all of that information is loaded into the metadata. From then on, all of the methods know what type of information each column contains, and can act accordingly.
- **MergeDualValueIntoUniverse** – This method is called on one demographic table and passed a parameter of another. This is the subtraction method I mentioned was used on the block level SF1 data. It assumes that the table on which it is called has a universe, and that the parameter table contains a subset of that universe. For example, it can be called on a total population by block table with a parameter table of the Non-Hispanic population by block. The code will recognize that Hispanic is a two value attribute (hence the “dual value” in the method name) and assume that subtracting will result in the Hispanic population. The result will be a new table with a Hispanic / Non-Hispanic breakdown.
- **DisaggregateOnDistribution** – This method is called on a table that has detail in one category, and is passed parameter of a table that has detail in a different category. It then distributes the values in the first table by the distribution found in the second. For example the table on which it is called could be a block level estimate with ten-year age groups. The parameter table could be a tract level table with single year age groups. This method will match the two tables on all of their other shared attribute such as sex and/or race, and create distribution ratios to estimate single year age groups at the block level.
- **BridgeToRaceMars77** – This method is called on a table that has the OMB MARS 97 race definition and applies the NCHS bridging method, resulting in a table by OMB MARS 77 race. It also requires a CensusCounty2000 and AgeSingleYear column so that the model coefficients can be calculated.
- **ConvertGeogUsingOverlay** – This is called on an estimate table and passed a parameter of an areal overlay table. The overlay table must have one attribute which is a geographic code that matches one of the geographic attributes of the first table. The overlay table must also have one other geographic code attribute, and finally an attribute that inherits from the “Area” class. This will contain the area of the overlap of the two geographies represented by the columns. The method does the calculation to convert the first table from one geographic code to the other.
- **InterpolatePopBetweenYears** – This is called on a table that is the start of a year time span and has a parameter of a table that is the end of that time span. It interpolated the years between them using a passed population growth table.

The beauty of these methods is that you do not pass them many parameters. They are intelligent enough to find the correct columns for the type of functions they perform. They also recognize different situations react accordingly. For this demographic work, the most impressive thing is that they can recognize the like columns between to tables. So for example, the Raking method is only given the list of marginal tables. The program figures out which columns are the same in these tables and the table being raked and

performs the proportional fitting accordingly. Many of the programs recognize nesting age groups and nested geographic levels and will automatically convert them so that the columns can be matched. These are just a sample of the method created, in total there were about 50 methods created specifically for this project. Many more were re-used from previous non-demographic work. Of those fifty, thirty-three were for loading data into the system and were likely project specific. The other eleven did things like convert age groups, convert race groups, and find the lowest level geography for the SF2 data. Most of these can be re-used in future projects. Since the work is put in up-front to create intelligent methods. Changing the methodology is achieved by simply re-arranging the calls on different tables.

Summary

While there is admittedly more effort involved up-front to create these methods with a generic approach, the benefit comes from the relative simplicity it offers when actually implementing the methodology decisions. With this system, changing the marginal tables, or adding or removing steps becomes a matter of changing a few lines of code in the main programs. It should also reduce the effort needed to create the next population estimate. This will improve their timeliness. Quality of future estimates should also be improved because the tools already exist, thereby reducing the tendency to do something “quick and dirty”. Was the effort worth it? Only Time will tell!