Exploring the U.S. Census

Your Guide to America's Data

Online Supplementary Materials

Frank Donnelly

SAGE

Los Angeles | London | New Delhi
Singapore | Washington DC | Melbourne

SAGE

FOR INFORMATION:

SAGE Publications, Inc.
2455 Teller Road
Thousand Oaks, California 91320
E-mail: order@sagepub.com

SAGE Publications Ltd.
1 Oliver's Yard
55 City Road
London, EC1Y 1SP
United Kingdom

SAGE Publications India Pvt. Ltd.
B 1/I 1 Mohan Cooperative Industrial Area
Mathura Road, New Delhi 110 044
India

SAGE Publications Asia-Pacific Pte. Ltd.
18 Cross Street #10-10/11/12
China Square Central
Singapore 048423

Acquisitions Editor: Leah Fargotstein
Editorial Assistant: Claire Laminen
Content Development Editor: Chelsea Neve
Production Editor: Rebecca Lee
Copy Editor: QuADS Prepress Pvt. Ltd.
Typesetter: Integra
Proofreader: Ellen Brink

CONTENTS

4

SUBJECT CHARACTERISTICS

4.7 EXERCISES

Conditionals in Spreadsheets

In the first exercise in Chapter 4, we aggregated age data into generational categories by manually assigning the age ranges to each category. Wouldn't it be nice if we could do this automatically?

Spreadsheets have special functions called conditional statements, which specify that something should be done if certain criteria are met. There is a general statement called IF: =IF(B2>10, B2+C2, B2). IF B2 is greater than 10 (the condition), then sum B2 and C2 in this cell, otherwise just show B2 in this cell. There are also conditionals that count values (COUNTIF), sum values (SUMIF), and average values (AVERAGEIF).

In our example, we could use SUMIF and SUMIFS to categorize the age data. First, in the sheet that has the single-year age data, we would add a column in E to hold just the age number. You would type 0 in the cell beside the total for Under 1 Year and then paste the value all the way down: Calc automatically increments the values by 1. We would just have to go to the bottom of the sheet and modify top age categories to 105 and 110 years, respectively.

Second, back in the generational sheet for the Greatest generation, we would use this statement: =SUMIF(Sheet2.E2:E104, ">"&D2, Sheet2.D2:D104). The first argument is the range that we are evaluating the argument against, the second is the criterion that must be met, and the third is the range that contains values that should be summed if the criterion is met. So look at the range of ages that are in column E in the age data, and if any values are greater or equal to the oldest person of that generation, sum the values (the number of people in each bracket) for those ages. In Calc, for criteria that are applied against a cell, you place the operator in quotes and attach it to the cell using the ampersand "&." In Excel, you would simply quote the operator and cell without the ampersand.

The other generations are a bit different as they require multiple criteria: People must be greater than one age but less than another. In these cases, we use the SUMIFS function:

=SUMIFS(Sheet2.D$2:D$104, Sheet2.E$2:E$104, ">="&D3, Sheet2.E$2:E$104, "<="&E3).

The first argument is the range of values that should be summed (population), followed by the first range of values that we apply the criterion against (range of ages) and the first criterion (for Silent generation, greater than or equal to the youngest age), then another range for criterion (same as before) and the second criterion (less than or equal to oldest age). We lock the value and criteria ranges with a "$" sign to keep them fixed, but we do *not* lock the criterion. This allows us to copy and paste the formula all the way down, so we can easily apply it to the other generations.

This is a quicker approach, once you learn how to use the functions. Remember, you can always go to Insert—Function and use the wizard for step-by-step assistance.

5

THE DECENNIAL CENSUS

5.5 EXERCISES

Accessing Data in Bulk: The FTP Site

While data.census.gov is the destination for most users who want to look up census statistics or download a few tables, it will not be your destination if you need to download data in bulk. There are a few alternatives that you can use, and the census FTP (File Transfer Protocol) site is the place to go if you really want all data for all geographies for a particular state for a specific dataset. The site is organized similarly to a file system on a computer or local network: There are folders for census programs with subfolders for specific datasets. You navigate through the folders until you find what you're looking for, and you download what you need in a large zip file. Given the size of these files and the way they are constructed, using a spreadsheet is out of the question. In this exercise, we'll examine how Summary File 1 (SF1) is constructed and how you can load this data into SQLite.

You can access the FTP site directly at `https://www2.census.gov/`, but it's probably easier to navigate via individual program and dataset pages, as these will lead you directly to the area you want. The program pages also contain the technical documentation that's essential for understanding how the files are constructed. For example, visit the page for SF1 for the 2010 census, and you'll get information that's specific to that dataset with a direct link to the FTP location: `https://www.census.gov/data/datasets/2010/dec/summary-file-1.html`.

SF1 (as well as the other decennial and American Community Survey [ACS] summary files) is organized, so there's one extract for each state that contains all geography that nests within the state, as well as a few geographies that are split in part between states. There is a national-level summary file, but this does *not* contain all the census data for the entire nation; it only includes geographies that do not nest within states. So it has ZIP Code Tabulation Areas, regions and divisions, metropolitan areas, and urban/rural divisions, but it doesn't have counties, tracts, or block groups. These geographies are only available state by state in the state files. If you need data for all tracts or block groups for the nation, there are better ways of getting it (we'll demonstrate the Dexter tool in the next chapter). The FTP site is really intended for getting everything there is for one state at a time, or for the truly die-hard who need absolutely everything and plan to download and stitch everything together.

The summary files for each state are a collection of text files that contain every variable for all geographies. The 2010 SF1 files include 47 comma-delimited text files that contain all the variables, plus one fixed-width geographic header file that contains the details about each piece of geography. Each data file contains a range of variables that is published in a specific table, such as P1, P2, H1, and so on. Using the Census Bureau's SF1 technical documentation, you can look up a table number and identify which file it falls in. Each record in each file is uniquely identified with an ID number called LOGRECNO, which allows you to relate the table to other tables and to relate it to the geographic header file, so you can select topic records for specific geography. The data files do not contain the standard GEOIDs or FIPS codes; to get the geography, you link the data files back to the geographic header file via LOGRECNO, and the header file contains the constituent parts of GEOIDs.

The structure of each series of state files and the national file are identical; the number of files and columns and the positions of the columns in the files are the same. The summary files for the 2000 census used a similar structure as 2010, and it is likely that 2020 will follow suit. (Data for the ACS is packaged on the FTP site in a similar way).

One of the problems with the decennial summary files is that they do not include header rows; there isn't a row that contains the ID names of the columns, so you have no idea what the column refers to unless you look it up in the technical documentation. The Census Bureau provides table shells for Microsoft Access databases, so you can create the blank table structures with the header rows and import the files into them. The import process is cumbersome given the limitations of Microsoft Access. If you wanted to use another database or a statistical program, you would need to use the documentation to manually create the blank tables.

Fortunately, we can rely on the kindness and diligence of others who have already done this work! A number of researchers have created scripts for importing census summary files into different statistical packages like R and SAS and have written SQL CREATE TABLE scripts for generating table shells for many databases. You can find these by searching the internet and, in particular, by looking on GitHub, which is a large open source repository and version control system for writing scripts and software. A scripter/programmer posted SQL scripts for creating tables for SF1 in PostgreSQL, which is a popular open source enterprise-level database. I have forked this repository and created a new one with scripts for SQLite: `https://github.com/frankpd/create_census_tables_sql`.

What do the data files look like, and why do we need a script to load them? Here is the structure of the first SF1 data file for Hawaii, hi000012010.sf, showing the first 10 records:

```
SF1ST,HI,000,01,0000001,1360301
SF1ST,HI,000,01,0000002,0
SF1ST,HI,000,01,0000003,0
SF1ST,HI,000,01,0000004,0
SF1ST,HI,000,01,0000005,0
SF1ST,HI,000,01,0000006,0
SF1ST,HI,000,01,0000007,0
SF1ST,HI,000,01,0000008,30858
SF1ST,HI,000,01,0000009,1360211
SF1ST,HI,000,01,0000010,432868
```

Notice there is no header row; we have to consult the documentation to see what the columns represent in each file. The first two fields indicate the summary file (SF1) and state (Hawaii). The third and fourth fields are ID numbers for specific characteristics, which are largely not used in state-level SF1. The fifth field is the LOGRECNO unique ID, which is associated with a specific geography *within* this summary file. So LOGRECNO 0000001 in the Hawaii file refers to the State of Hawaii, while in a summary file for another state 0000001 would refer to that state. These first five fields are the same in every one of the 47 tables. The last field in this data file is p0010001, which refers to the first population variable in the dataset: total population as published in Table P1. Some of the files (like this one) are relatively small and contain variables from one census table, but in most cases, variables from several tables will appear in one file. For example, P1 appears in File 1 and P2 is in File 2, but File 3 contains variables from Tables P3 through P9.

The format for the geographic header file is more complex, as it is not a delimited text file with values separated by a comma or tab but is a fixed-width file. In a fixed-width file, values occupy specific positions across the row, and you must delineate them using a code book. For example, Positions 1 through 6 hold the fileid, Positions 7 and 8 hold the state abbreviation, and so on. You can parse fixed-width files manually in Calc or other spreadsheets, but it is a tedious process for large files. It's better to use either SQL or a scripting language to parse the files. Fortunately, others have written scripts for doing this, and the scripts I have modified and created for SQLite include a parsing script.

A final note before we begin: One of the tables that SF1 does *not* include is the demographic profile table DP01, which as we have seen contains a cross section of the most essential census variables in one table. The demographic profile is packaged separately in its own series of state-based and national summary files. It is a much simpler dataset, as it contains just one data file and the geographic header file. If SF1 is overkill for your needs, the DP summary file is a quick alternative. It is released before SF1 with a smaller subset of geographies, and then is updated following the release of SF1 to include all the geographies that were published in SF1. `https://www.census.gov/data/datasets/2010/dec/demographic-profile-with-geos.html`.

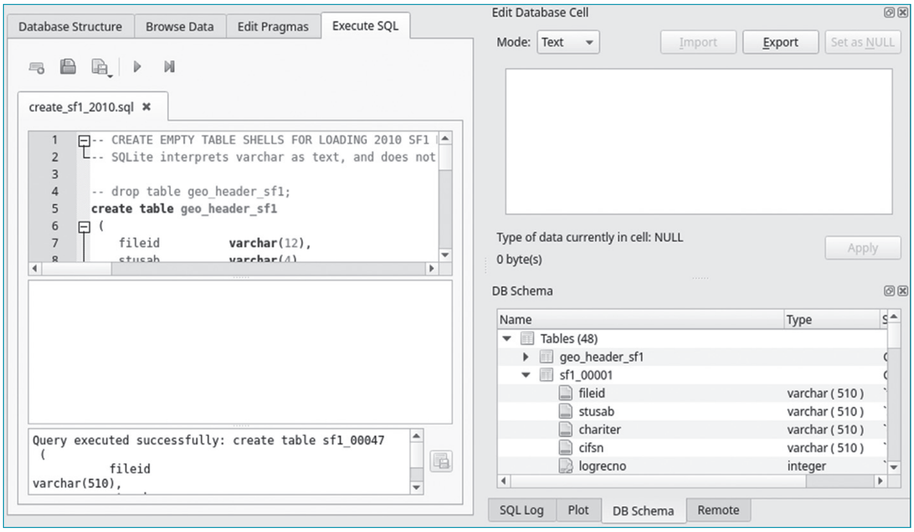
Chapter 5 Supplemental Exercise folder contains the SF1 data for Hawaii, the SQL scripts we need for loading and parsing data in SQLite, and the technical documentation from the Census Bureau in PDF format. Alternatively, you could access the data from the following source:

hi2010sf1 : 2010 Decennial Census, all SF1 data, Hawaii (state)—`https://www2.census.gov/census_2010/04-Summary_File_1/`

1. **Create a new database**. Launch the DB Browser for SQLite. Create a new database called exercise3.sqlite. Save it in Chapter 5 Supplemental Exercise folder. When prompted to create a new, blank table, hit the Cancel button.

2. **Run the SQL file to create the empty table shells**. Select the Execute SQL tab. Hit the second button above the SQL window that looks like a folder—this is the Open SQL file button. Browse to Chapter 5 Supplemental Exercise folder, and select the file create_SF1_2010.sql. Once you do, the file will be loaded in the SQL window, and you can see the long list of CREATE TABLE statements. In the last exercise, we executed statements one by one, but it is also possible to execute statements in a series. Each statement ends with a semicolon that indicates where one statement ends and the next one begins. Hit the blue arrow button to execute the script. This will create all the blank tables, which you can see and explore if you select the DB schema tab in the lower right-hand corner (Figure 5.1). Save your work by hitting the Write Changes button.

FIGURE 5.1 ⬢ LOAD AND RUN SCRIPT TO CREATE SF1 TABLES



```
-- CREATE EMPTY TABLE SHELLS FOR LOADING 2010 SF1
-- SQLite interprets varchar as text, and does not

-- drop table geo_header_sf1;
create table geo_header_sf1
 (
    fileid          varchar(12),
    stusab          varchar(4)
```

```
Query executed successfully: create table sf1_00047
 (
            fileid
varchar(510),
```

Type of data currently in cell: NULL

0 byte(s)

Name	Type
Tables (48)	
geo_header_sf1	
sf1_00001	
fileid	varchar (510)
stusab	varchar (510)
chariter	varchar (510)
cifsn	varchar (510)
logrecno	integer

3. **Load the first population file**. Go to File—Import—Table From CSV File. Browse to Chapter 5 Supplemental Exercise hi2010sf1 folder. At the bottom of the screen, change the option from Text files to All files (even though the data files are in a CSV format, their file extension is .sf1, so they don't appear when looking for files with the .csv extension). Select the first sf1 file hi000012010.sf1, and hit Open. Under table name, change the name to sf1_00001. The field separator should be a comma. Uncheck the box that says Column names in first line, but check the box to Trim fields (Figure 5.2). Hit OK. You'll get prompted with a message that says "There is already a table with that name. Do you want to import data into it?" Click Yes. The data will import. When it's finished, hit the Browse tab, and in the table drop down, select table sf1_00001. You should see all the data has been imported into the empty table. Save your work.

4. **Load the next two population files**. Repeat the previous step and load in the next two population files, file hi000022010.sf1 into table sf1_00002 and file hi000032010.sf1 into table sf1_00003.

5. **Create staging table for geographic headers**. Getting the geographic header file into the database is a two-step process. First, we'll load the fixed-width file into a temporary table where we store everything in a single field. Hit the SQL tab, and hit the first button above the window to add a new SQL window.

FIGURE 5.2 ⬢ IMPORT FIRST SF1 TABLE

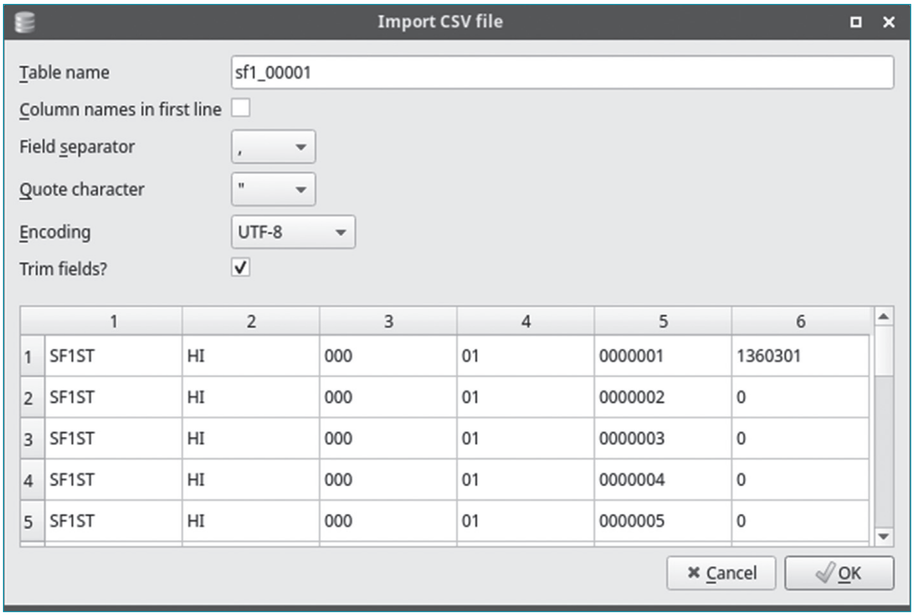


Table name: sf1_00001

Column names in first line: □

Field separator: ,

Quote character: "

Encoding: UTF-8

Trim fields?: ☑

	1	2	3	4	5	6
1	SF1ST	HI	000	01	0000001	1360301
2	SF1ST	HI	000	01	0000002	0
3	SF1ST	HI	000	01	0000003	0
4	SF1ST	HI	000	01	0000004	0
5	SF1ST	HI	000	01	0000005	0

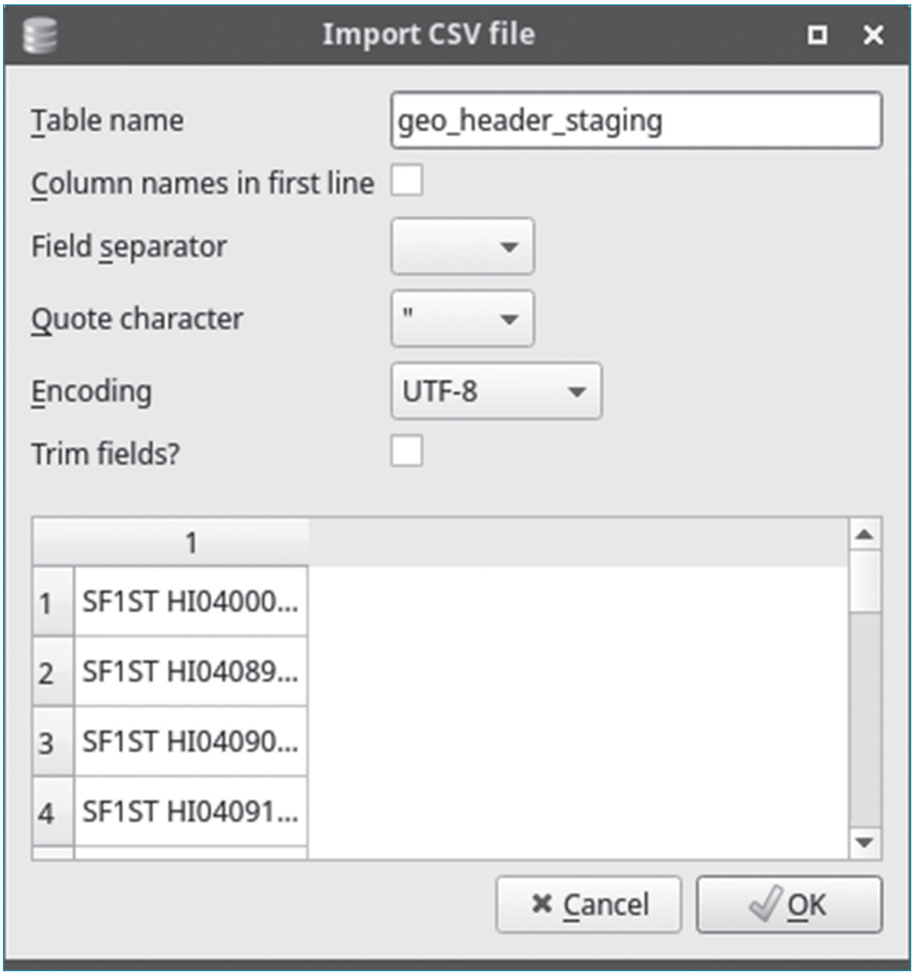
Then close the window that had our creation script by hitting the X. Write this statement in the empty window:

```
CREATE TABLE geo_header_staging (
  data TEXT);
```

Then import the geographic header file higeo2010.sf1 (last file in the file list) into the geo_header_staging table. In the import window, change the table name to geo_header_staging, change the field separator option to Other, and leave it blank. Also uncheck the trim fields box (Figure 5.3). When finished, browse the table to make sure the data was imported.

6. **Parse and load the geo header data into the permanent table**. Go to the Execute SQL tab, and erase the previous statement. Hit the open SQL file button (second button), and select the geoheader_sqlite_sf1_2010.sql file to load the script into the SQL window. Hit the execute button (Figure 5.4). This script takes each row in the staging table and splits it into individual values to fit into the geo_header_sf1 table. The values are split using the substring function that we saw earlier at designated positions based on the Census Bureau's documentation. We also use a function called trim to remove blank spaces before or after

FIGURE 5.3 ⬢ IMPORT GEOGRAPHIC HEADER FILE



Import CSV file

Table name: geo_header_staging

Column names in first line □

Field separator:

Quote character: "

Encoding: UTF-8

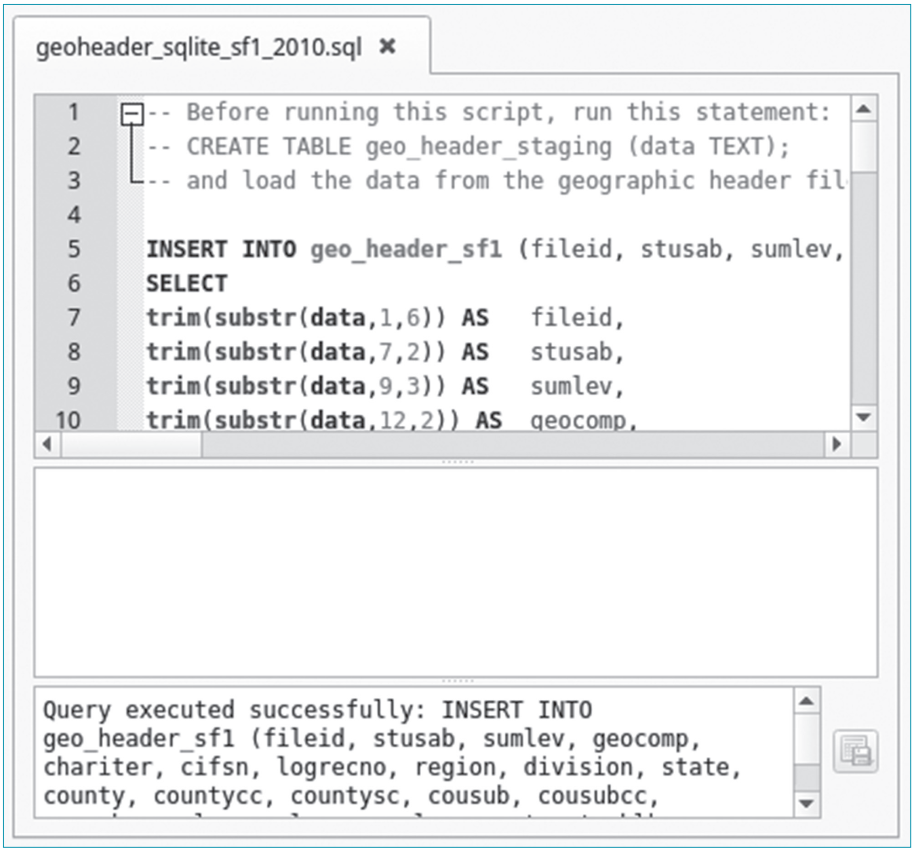
Trim fields? □

	1
1	SF1ST HI04000...
2	SF1ST HI04089...
3	SF1ST HI04090...
4	SF1ST HI04091...

each value and, in some cases, a function called cast to convert values from one data type to another (in this case text to integers).

7. **Browse the geo_header_sf1 table.** Browse the data for geo_header_sf1. Click on the sumlev column to sort the data by summary level. This table contains records for every piece of geography in Hawaii that nests fully (and in some cases partially) within the state. Remember the summary level codes from Chapter 3? Code 040 is for the state level. The first record with sumlev 040 and geocomp 00 is for the state as a whole. The geocomp field indicates whether the record pertains to a specific geographic component of the area, for example, sumlev 040 geocomp 95 indicates this record is for the portion of

FIGURE 5.4 ● LOAD AND RUN SCRIPT TO PARSE AND INSERT GEOGRAPHIC HEADER DATA



```
-- Before running this script, run this statement:
-- CREATE TABLE geo_header_staging (data TEXT);
-- and load the data from the geographic header fil

INSERT INTO geo_header_sf1 (fileid, stusab, sumlev,
SELECT
trim(substr(data,1,6)) AS   fileid,
trim(substr(data,7,2)) AS   stusab,
trim(substr(data,9,3)) AS   sumlev,
trim(substr(data,12,2)) AS  geocomp,
```

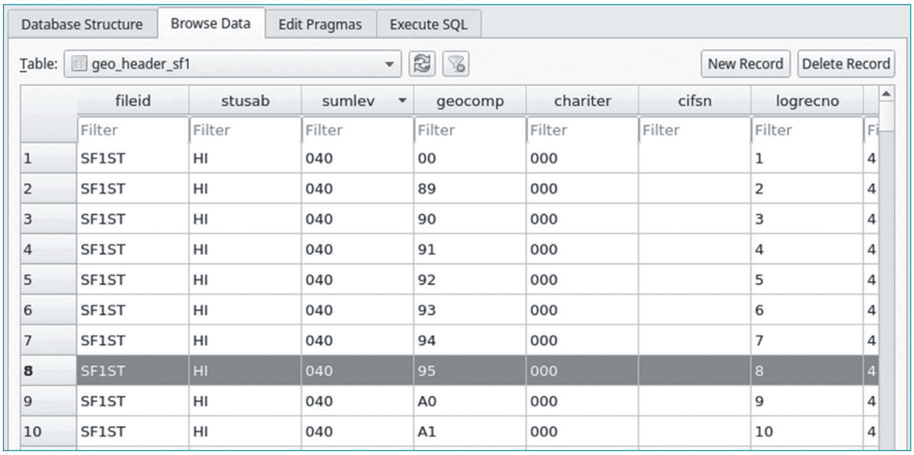
```
Query executed successfully: INSERT INTO
geo_header_sf1 (fileid, stusab, sumlev, geocomp,
chariter, cifsn, logrecno, region, division, state,
county, countycc, countysc, cousub, cousubcc,
```

the state that is Hawaiian Home Land, while sumlev 040 geocomp A1 is for the portion of the state that is in a metropolitan or micropolitan area (Figure 5.5). These are provided as convenient summaries (see the technical documentation for a full list of codes). In the state-level summary files, only states have component codes.

8. **Identify all summary levels**. Go to the Execute SQL tab, and hit the first button to open a new SQL tab. The DISTINCT qualifier returns all unique instances of a value or combination of values. Type this query to identify the available summary levels in this dataset:

```
SELECT DISTINCT sumlev
FROM geo_header_sf1
ORDER BY sumlev;
```

FIGURE 5.5 SUMMARY LEVEL AND GEOGRAPHIC COMPOSITION CODES



	fileid	stusab	sumlev	geocomp	chariter	cifsn	logrecno	
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Fi
1	SF1ST	HI	040	00	000		1	4
2	SF1ST	HI	040	89	000		2	4
3	SF1ST	HI	040	90	000		3	4
4	SF1ST	HI	040	91	000		4	4
5	SF1ST	HI	040	92	000		5	4
6	SF1ST	HI	040	93	000		6	4
7	SF1ST	HI	040	94	000		7	4
8	SF1ST	HI	040	95	000		8	4
9	SF1ST	HI	040	A0	000		9	4
10	SF1ST	HI	040	A1	000		10	4

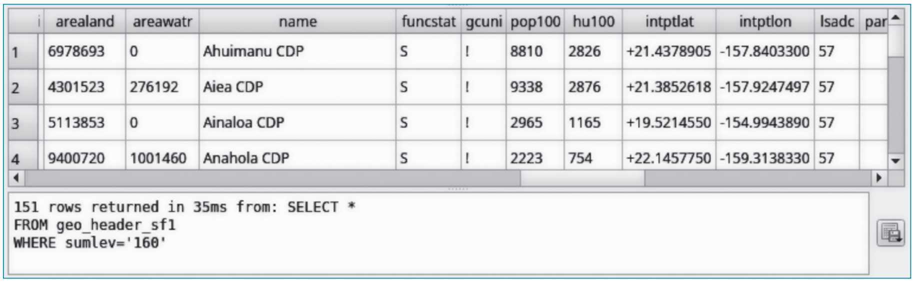
9. **Select all the place records in Hawaii**. Let's select the records for all places in Hawaii. Erase the previous query and enter this one:

```
SELECT *
FROM geo_header_sf1
WHERE sumlev='160';
```

Once you get the results for the 151 places, scroll across the table and explore a little. You will see that in addition to the place codes, you also have fields for codes that are above that summary level. Since places nest within states at summary level 160, we don't have codes for counties or county subdivisions as places can cross these boundaries. Keep scrolling right, and we will see the name of the areas under "name" and the area in land and water, which is in square meters. The total population and housing units are included for convenience in fields pop100 and hu100 (Figure 5.6).

10. **Tie data together**. Now, what if we wanted to select some variables for specific places? Let's say we want to see the total population and the total population who are Hawaiian or Pacific Islanders alone for all places. To do this, we need to consult the summary file technical documentation. First, we'd locate the summary-level code for places under the summary-level sequence chart in Chapter 4. Then we need to locate the tables that contain data on race by looking at the Subject Locator in Chapter 3 or by browsing through the list of tables in Chapter 5. Based on what we've already covered in this book, we

FIGURE 5.6 ⬢ PLACES IN HAWAII IN THE GEOGRAPHIC HEADER FILE



i	arealand	areawatr	name	funcstat	gcuni	pop100	hu100	intptlat	intptlon	lsadc	par
1	6978693	0	Ahuimanu CDP	S	!	8810	2826	+21.4378905	-157.8403300	57	
2	4301523	276192	Aiea CDP	S	!	9338	2876	+21.3852618	-157.9247497	57	
3	5113853	0	Ainaloa CDP	S	!	2965	1165	+19.5214550	-154.9943890	57	
4	9400720	1001460	Anahola CDP	S	!	2223	754	+22.1457750	-159.3138330	57	

```
151 rows returned in 35ms from: SELECT *
FROM geo_header_sf1
WHERE sumlev='160'
```

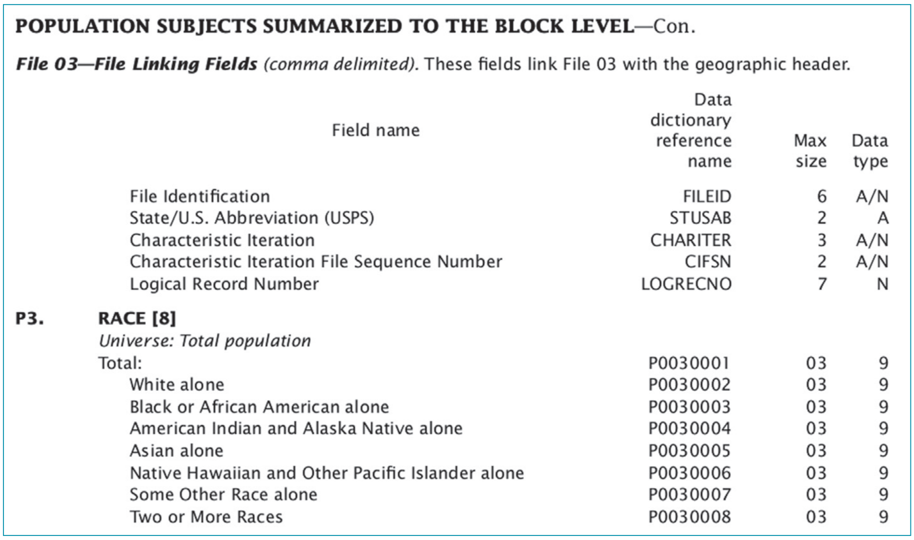
know that Table P3 is the primary table for Race Alone. We can go into the Table Matrix section in Chapter 6 and get a readout for Table P3 that tells us that it is located in file 03 and consists of eight variables: The first one P0030001 is for total population and P0030006 is for Hawaiian and Other Pacific Islander Alone (Figure 5.7). Knowing this, we can write the following query:

```
SELECT g.place, g.name, s.P0030001, s.P0030006
FROM geo_header_sf1 g
INNER JOIN sf1_00003 s ON (g.logrecno=s.logrecno)
WHERE g.sumlev=’160’
ORDER BY g.place;
```

11. **Final tweaks**. To clearly identify what the output of our query represents, we can add aliases to replace the variable ID codes. We can also calculate some percent totals and add the proper FIPS code. The summary files do not provide full GEOIDs; if we needed them for some reason (maybe for joining this data to another dataset), we can construct them out of the identifiers. Recall from Chapter 3 that FIPS codes for places that nest within states consist of two parts: (1) the two-digit state code and (2) the five-digit place code. Revise your previous statement with this one (Figure 5.8):

```
SELECT (g.state || g.place) AS fips, g.name,
  s.p0030001 AS totpop, s.p0030006 AS islanderpop,
  ROUND(((CAST(p0030006 AS REAL) / CAST(p0030001 AS REAL))
  *100),1) AS pct
FROM geo_header_sf1 g
INNER JOIN sf1_00003 s ON (g.logrecno=s.logrecno)
WHERE g.sumlev=’160’
ORDER BY g.place;
```

FIGURE 5.7 ⬢ TECHNICAL DOCUMENTATION SHOWING FILE CONTENTS

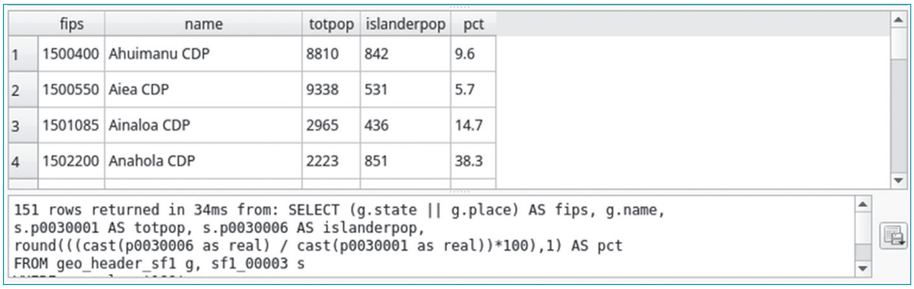


POPULATION SUBJECTS SUMMARIZED TO THE BLOCK LEVEL—Con.

File 03—File Linking Fields *(comma delimited)*. These fields link File 03 with the geographic header.

	Field name	Data dictionary reference name	Max size	Data type
	File Identification	FILEID	6	A/N
	State/U.S. Abbreviation (USPS)	STUSAB	2	A
	Characteristic Iteration	CHARITER	3	A/N
	Characteristic Iteration File Sequence Number	CIFSN	2	A/N
	Logical Record Number	LOGRECNO	7	N
P3.	**RACE [8]**			
	Universe: Total population			
	Total:	P0030001	03	9
	White alone	P0030002	03	9
	Black or African American alone	P0030003	03	9
	American Indian and Alaska Native alone	P0030004	03	9
	Asian alone	P0030005	03	9
	Native Hawaiian and Other Pacific Islander alone	P0030006	03	9
	Some Other Race alone	P0030007	03	9
	Two or More Races	P0030008	03	9

FIGURE 5.8 ⬢ QUERY RESULT WITH ALIASES AND CALCULATED FIELDS



	fips	name	totpop	islanderpop	pct
1	1500400	Ahuimanu CDP	8810	842	9.6
2	1500550	Aiea CDP	9338	531	5.7
3	1501085	Ainaloa CDP	2965	436	14.7
4	1502200	Anahola CDP	2223	851	38.3

```
151 rows returned in 34ms from: SELECT (g.state || g.place) AS fips, g.name,
s.p0030001 AS totpop, s.p0030006 AS islanderpop,
round(((cast(p0030006 as real) / cast(p0030001 as real))*100),1) AS pct
FROM geo_header_sf1 g, sf1_00003 s
```

To build the FIPS code, we use the double bars "||" to take text in one column and combine it with another (in a spreadsheet this is known as a CONCATENATE formula). We use AS to provide an alias for this calculated field, as well as to simply rename the variable ID columns. Why does the percent total calculation look so intense? In a database, when you divide one integer by another integer, the resulting value is an integer. When calculating a percent total, the value will always be rounded off to 0 or 1, which is not useful. To avoid this, we use the CAST function to convert the integer to a real (decimal) number. The result is going to be many decimal places long, so we use the round function to get one decimal place. The easiest way to read nested statements like this is from the inside out: Cast the first value as real, then the second value, then divide them, then multiply by 100, then round off everything to one decimal place. The parentheses are key to ensuring that the order of operations is followed and that the functions are given the correct arguments. In this case, round takes two arguments: the result of all the math that happens and then one that rounds the result to one decimal place.

12. **Save as a view**. In the statement in the previous step, before the SELECT clause, type CREATE VIEW hiplacepop AS, and hit the execute button to save the statement as a view. This allows you to quickly pull the view up in the Browse tab.

This exercise has demonstrated the power of databases for storing and retrieving data and illustrated how the summary files on the FTP site are constructed so you can grab data in bulk. Unfortunately, it takes time to manually import all the SF1 tables one by one. You could save time by just importing files that you know you'll need rather than importing everything. Consult the technical documentation to identify the relevant files.

If you were working with the Demographic Profile or with PL 94-171, this process would be much simpler, as these are smaller datasets with one or two data files. In Chapter 5 Supplemental Exercise folder, there is an extra subfolder that contains the summary file for the 2010 census Demographic Profile for Hawaii, SQL scripts for creating the data table (there's only one table) and the geographic header file, and technical documentation. Based on what we did in this exercise, for aditional practice, try creating a new SQLite database for the Demographic Profile. Once the data is loaded, write a SQL statement that ties a specific type of geography from the header file to some variables of your choosing in the data file, and save it as a view.

Like the decennial census, the ACS is packaged into summary files (one for each period estimate) that you can access via the FTP site. ACS variables are spread across many individual files sorted by their table numbers; rows are uniquely identified with a LOGRECNO number, and this number can be linked back to specific geography in a geographic header file. The ACS summary files are more numerous than the decennial census; the state-level extract for 2016 has 246 files. The Census Bureau provides templates (table shells) for each data table and the geographic header file in Excel format, and all the data files are in a CSV format *including* the geographic header file. The data is packaged in collections of files for each state and for the nation, and for the 5-year estimates, these collections are broken into two extracts: one that contains block groups and tracts and the other for all other geographies. Within each set, the data for the estimates and margins of error are packaged in separate files, identified with a prefix "e" or "m."

While the Census Bureau provides easy-to-follow instructions for importing an individual data file into one of the Excel templates, using a spreadsheet to work with this data is tough. For a particular ACS table, you would need to import at least three files—(1) the geographic header, (2) the estimate table, and (3) the margin of error table—and would use VLOOKUP formulas to tie them together. This is fine if you need a couple of tables for all geographies, but this is not a feasible approach for working with a lot of data, or even a few variables scattered across many files. It still would be better to import the files into a database or a statistical package. Besides Excel, the Census Bureau provides table structures in a CSV and SAS format (SAS programmers can use preassembled macros). As before, you could also search github for scripts for generating tables in SQL or another stat package. The best launching point for accessing the ACS summary files is from the ACS technical documentation page: `https://www.census.gov/programs-surveys/acs/technical-documentation/summary-file-documentation.html`.

6

THE AMERICAN COMMUNITY SURVEY

6.4 EXERCISES

Ranking ACS Data and Testing for Statistical Difference

Census data is commonly used to categorize and rank places. As we have discussed, the census is used to distribute hundreds of billions of dollars in federal aid. In their case study of three federal programs, Nesse and Rahe (2015) found than none of the programs incorporated the margin of error (MOE) for American Community Survey (ACS) estimates into their calculations. In this exercise, we will explore the impact the MOE has on rankings and will get some more practice with aggregating values.

We will use one of the programs from this study in our example. The U.S. Department of Agriculture (USDA) runs the Supplemental Nutrition Assistance Program (SNAP), which provides food stamps to individuals in need. Funding is distributed to each of the states based on a variety of criteria and formulas. One aspect of the funding mechanism is the award of performance bonuses to states that do the best job at administering the program. Four different evaluation categories are used to administer bonuses, and one of them relies on census data. The Program Access Index (PAI) is used to award $12 million to eight states: the four states that have the highest index value and the four states that showed the most improvement in their index compared with the previous year. Nine states receive awards in the event of a tie, and the money is divided in the same manner as it would be for eight states. If a state is among the top four in both the best category and best improved category, that state receives only one award and the fifth best state is granted an award to bring the number up to eight.

The index is calculated for each state by dividing the average number of SNAP recipients in a month by the total population living below 125% of the poverty line in a calendar year. The data on poverty is from the 1-year ACS, while the data on SNAP recipients comes from the USDA's administrative records (the ACS includes data on SNAP recipients, but it's published at the household level, while the USDA's data is at the person level). The USDA makes a number of adjustments prior to calculating the index. The number of SNAP recipients is adjusted by deducting people who received benefits as part of disaster assistance and participants in the Food Distribution Program on Indian Reservations (FDPIR). The number of people in poverty is adjusted by deducting the number of average monthly participants in FDPIR and the total number of low-income Supplemental Security Income (SSI) recipients in California (U.S. Department of Agriculture, 2018).

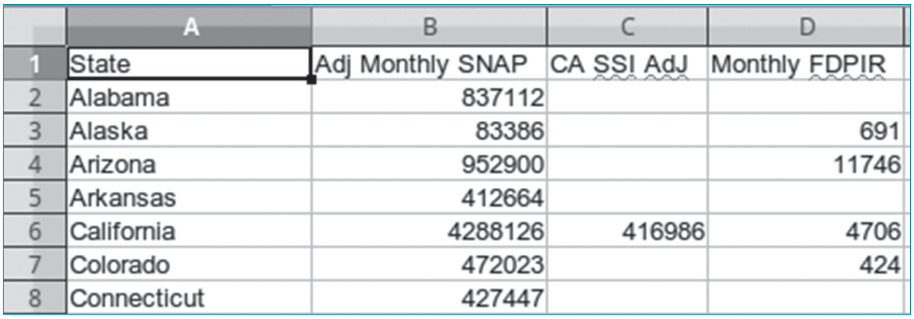
In our example, we will account for the MOE in the calculations to create adjusted poverty estimates and the index. We will look at the rankings and identify the top four states in the best program category and will run the statistical difference test to see if the values between the fourth- and fifth-best state are truly different. The USDA publishes a thorough summary of the data used and statistics generated in an annual report USDA (2018). I created a spreadsheet that contains the necessary data for making the poverty adjustments and calculating the index by copying the relevant columns out of the report and pasting them into the workbook. We will import the poverty data from the ACS so that we get the MOEs (as they are not included in the report). The data is available in the Chapter 6 Supplementary Exercise folder, or you can obtain it from the following source:

B17002 Ratio of Income to Poverty Level in the Past 12 Months: 2016 ACS, All states in the United States (state)—`https://data.census.gov/`

Calculating the SNAP Program Access Index: A Step-By-Step Guide: UDSA, 2016 report—`https://www.fns.usda.gov/snap/calculating-supplemental-nutrition-assistance-program-snap-program-access-index-step-step-guide`

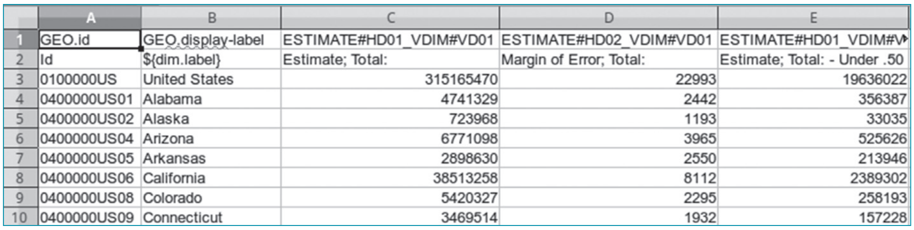
1. **Open the SNAP spreadsheet.** Go to Chapter 6 Supplementary Exercise folder and open the spreadsheet snap_pai2016.xlsx in Calc. You'll see four columns: (1) the name of the state, (2) the average adjusted monthly participants in SNAP, (3) the number of participants in California's SSI program, and (4) the average monthly participants in FDPIR (Figure 6.1). We will use these last two

FIGURE 6.1 ⬢ 2016 USDA SNAP PROGRAM ACCESS INDEX COMPONENTS



	A	B	C	D
1	State	Adj Monthly SNAP	CA SSI AdJ	Monthly FDPIR
2	Alabama	837112		
3	Alaska	83386		691
4	Arizona	952900		11746
5	Arkansas	412664		
6	California	4288126	416986	4706
7	Colorado	472023		424
8	Connecticut	427447		

FIGURE 6.2 ⬢ ACS TABLE B17002 RATIO OF INCOME TO POVERTY LEVEL



	A	B	C	D	E
1	GEO.id	GEO.display-label	ESTIMATE#HD01_VDIM#VD01	ESTIMATE#HD02_VDIM#VD01	ESTIMATE#HD01_VDIM#V
2	Id	${dim.label}	Estimate; Total:	Margin of Error; Total:	Estimate; Total: - Under .50
3	0100000US	United States	315165470	22993	19636022
4	0400000US01	Alabama	4741329	2442	356387
5	0400000US02	Alaska	723968	1193	33035
6	0400000US04	Arizona	6771098	3965	525626
7	0400000US05	Arkansas	2898630	2550	213946
8	0400000US06	California	38513258	8112	2389302
9	0400000US08	Colorado	5420327	2295	258193
10	0400000US09	Connecticut	3469514	1932	157228

columns to adjust our poverty numbers. Note that the UDSA excluded New Mexico from the rankings in 2016 due to some data quality issue.

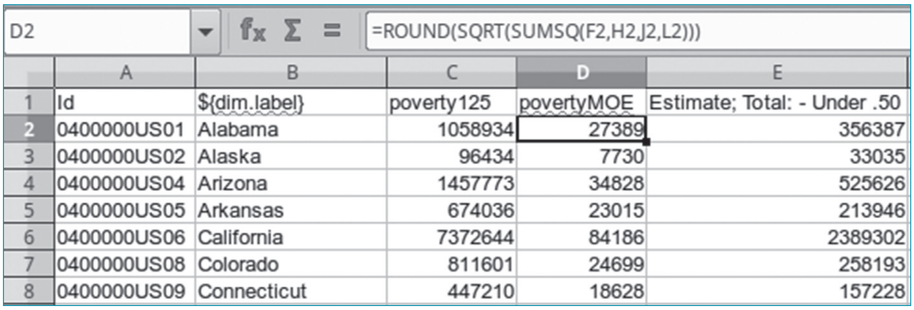
2. **Import the poverty data**. Import the ACSDT1Y2016_B17002_with_ann.csv file from Chapter 6 Supplementary Exercise folder. Click OK to get through the import screens. This is state-level data from Table B17002, Ratio of Income to Poverty Level in the Past 12 Months from the 2016 1-year ACS (Figure 6.2). The total estimate is for all persons for whom poverty status could be determined. The individual estimates tell us the number of people who fall in a certain range above or below the official poverty line. For example, in column E, the estimate is for the total people who live below 50% of the poverty line, while column K is the number of people who live at the poverty line (100%) to 125% above the poverty line. These ratios are used to capture a broader range of people who are living with limited means but who are above the official poverty level and to identify the absolute poorest Americans who are living below poverty line. The SNAP PAI incorporates everyone who is below 125% of poverty line, so we will need to aggregate several columns and exclude others.

3. **Remove unnecessary columns and rows**. Delete columns M through AB, as these fall outside the range we need. Then delete columns C and D as we don't need the totals. Delete the first row that contains the IDs, then delete the summary row for the United States. Scroll down to the bottom of the sheet and delete the row for Puerto Rico (since it's not a state, it doesn't participate in SNAP). Save your work by doing Save As, and save the file in a new spreadsheet called exercise2.

4. **Calculate total for 125% poverty and MOE**. Select column B, right-click, and Insert column to the right. Then do this a second time, so we have two blank columns in C and D. In cell C1, type the label poverty125, and in cell D1, type the label povertyMOE. In cell C2, type the formula =SUM(E2,G2,I2,K2). Copy and paste this formula down column C. In cell D2, type the formula =ROUND(SQRT(SUMSQ(F2,H2,J2,L2))). Copy and paste this formula down column D (Figure 6.3). This is the same formula we used in the last exercise for calculating aggregates, but in this case, we have to provide individual values instead of a range because the columns alternate between the estimates and the MOEs. Save your work.

5. **Copy the poverty columns into the SNAP worksheet**. We don't have matching ID codes in both our worksheets; we could do VLOOKUP on the names of the states, but since there are only 51 and both sheets are in alphabetical order, we'll do a simple copy and paste. Select columns B, C, and D in the poverty worksheet, and copy them. Flip over to the PAI2016 worksheet, click in cell E1, and do Paste Special—Values (do *not* transpose), so that we're pasting the values and not the formulas. To do a quick check that everything matches, in cell H2 type this formula: =IF(A2=E2,"ok","PROBLEM!"). Copy and paste the result down column H (Figure 6.4). The formula checks the state name in column A from our SNAP data against the one in column E from our poverty data. If they match, we know that our data lines up correctly and "ok"

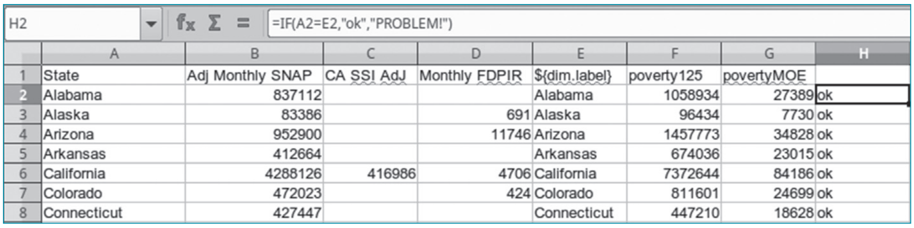
FIGURE 6.3 ⬢ CALCULATE MARGIN OF ERROR FOR 125% POVERTY



D2 =ROUND(SQRT(SUMSQ(F2,H2,J2,L2)))

	A	B	C	D	E
1	Id	${dim.label}	poverty125	povertyMOE	Estimate; Total: - Under .50
2	0400000US01	Alabama	1058934	27389	356387
3	0400000US02	Alaska	96434	7730	33035
4	0400000US04	Arizona	1457773	34828	525626
5	0400000US05	Arkansas	674036	23015	213946
6	0400000US06	California	7372644	84186	2389302
7	0400000US08	Colorado	811601	24699	258193
8	0400000US09	Connecticut	447210	18628	157228

FIGURE 6.4 ⬢ PASTE POVERTY DATA INTO SNAP WORKSHEET, CHECK FOR MATCHING RECORDS



H2 =IF(A2=E2,"ok","PROBLEM!")

	A	B	C	D	E	F	G	H
1	State	Adj Monthly SNAP	CA SSI AdJ	Monthly FDPIR	${dim.label}	poverty125	povertyMOE	
2	Alabama	837112			Alabama	1058934	27389	ok
3	Alaska	83386		691	Alaska	96434	7730	ok
4	Arizona	952900		11746	Arizona	1457773	34828	ok
5	Arkansas	412664			Arkansas	674036	23015	ok
6	California	4288126	416986	4706	California	7372644	84186	ok
7	Colorado	472023		424	Colorado	811601	24699	ok
8	Connecticut	427447			Connecticut	447210	18628	ok

is printed. Otherwise, "problem" gets printed, and we know that we have a mismatch. If everything is "ok," delete column H, then delete column E (the extra state name). Save your work.

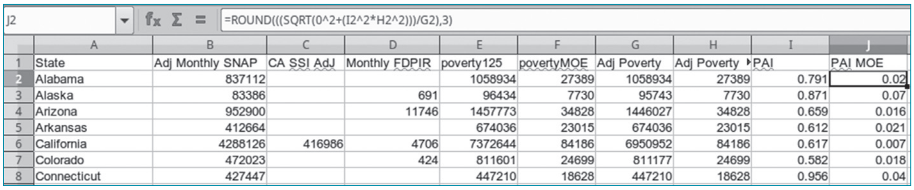
6. **Create the adjusted poverty value**. In cell G1, type the label Adj Poverty. In cell H1, type the label Adj Poverty MOE. In cell G2, type the formula =E2-C2-D2. Copy and paste the formula down column G. In cell H2, type =F2. Copy and paste the result all the way down. Why did we do this? The formula for calculating the MOE for the difference between two values is the *same* formula for calculating a sum. The CA SSI and FDPIR values are not estimates but actual counts taken from administrative data. We would assume their MOE is zero. If we took the square root of the sum of the squares, the values for the admin data would be zero and these zeros would be added to the sum of the square for the poverty MOE. Taking the square root of the square of the poverty, MOE simply returns the same poverty MOE. So we can simply copy and paste the same poverty MOE. Even though the MOE is the same, the poverty estimate is now smaller, so the coefficient of variation for this value will be larger. Save your work.

7. **Create the PAI index**. In cell I1, type the label PAI. In cell J1, type the label PAI MOE. In cell I1, type =ROUND((B2/G2),3). Copy and paste the formula down column I. The PAI is a ratio, so we would use the formula for calculating the MOE for a ratio:

$$\text{MOE} = \frac{\sqrt{\text{MOEsubset}^2 + (\text{Ratio}^2 * \text{MOEtotal}^2)}}{\text{Total}}$$

Type this formula in cell J2: =ROUND(((SQRT(0^2+(I2^2*H2^2)))/G2),3). Copy and paste the result down column J (Figure 6.5). In this case, the MOE for the subset population is zero: The numerator is the number of SNAP

FIGURE 6.5 ⬢ CREATE THE PAI AND CALCULATE ITS MARGIN OF ERROR



=ROUND(((SQRT(0^2+(I2^2*H2^2)))/G2),3)

	A	B	C	D	E	F	G	H	I	J
1	State	Adj Monthly SNAP	CA SSI AdJ	Monthly FDPIR	poverty125	povertyMOE	Adj Poverty	Adj Poverty	PAI	PAI MOE
2	Alabama	837112			1058934	27389	1058934	27389	0.791	0.02
3	Alaska	83386		691	96434	7730	95743	7730	0.871	0.07
4	Arizona	952900		11746	1457773	34828	1446027	34828	0.659	0.016
5	Arkansas	412664			674036	23015	674036	23015	0.612	0.021
6	California	4288126	416986	4706	7372644	84186	6950952	84186	0.617	0.007
7	Colorado	472023		424	811601	24699	811177	24699	0.582	0.018
8	Connecticut	427447			447210	18628	447210	18628	0.956	0.04

participants, which is not an estimate but an actual total created from administrative data. It's assumed that its MOE would be zero, so we just hard code this value into the formula. Check your work by visiting the Cornell ACS calculator and type the values for Alabama in the second calculator: 837,111 and 0 as the first value and 1,058,934 and 27,389 as the second value. Hit the ratio button. Your values for the PAI and PAI MOE should match, approximately 0.791 ± 0.02 (the calculator uses four decimal places, and we rounded it to three to be consistent with the USDA's format). Save your work.

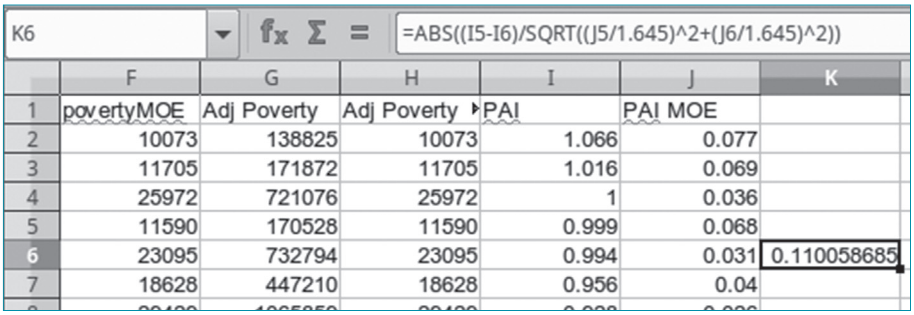
8. **Identify the top-ranked states**. Go to Data—Sort. Under the Options, verify that the first row contains column names (range contains column labels). In the Sort menu, choose PAI as the sort key and sort the data in descending order. The top four states are Delaware, Hawaii, Oregon, and Rhode Island. If we check the USDA's report, the rankings we generated and our PAI values should match the report's values. Based on the program's guidelines, these four states would receive a bonus. But notice that the cutoff between the fourth-ranked Rhode Island and the fifth-ranked Maryland is pretty small: 0.999 for Rhode Island and 0.994 for Maryland. Given the size of the MOEs, it's possible that the true values overlap. Are these estimates truly different, or could any difference be the result of random chance? Let's find out.

9. **Calculate statistical difference**. The formula for calculating statistical difference is:

$$SD = \left| \frac{\text{EST1} - \text{EST2}}{\sqrt{\text{SE1}^2 + \text{SE2}^2}} \right|$$

We subtract the second estimate from the first estimate and divide it by the square root of the sum of squares for the standard errors of each estimate and take the absolute value of the result. The standard error is a measure of the variability of the sample mean. We can calculate it by dividing the MOE by the Z value for the 90% confidence level, 1.645. In cell K6, type =ABS((I5-I6)/SQRT((J5/1.645)^2+(J6/1.645)^2)). If the test value (the result of this formula) is greater than the Z value of 1.645, then the differences between

FIGURE 6.6 TESTING SIGNIFICANT DIFFERENCE BETWEEN FOURTH- AND FIFTH-RANKED PAI



K6 =ABS((I5-I6)/SQRT((J5/1.645)^2+(J6/1.645)^2))

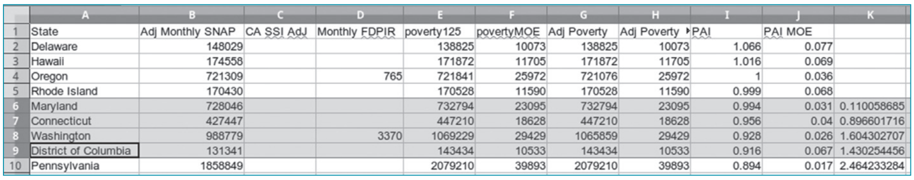
	F	G	H	I	J	K
1	povertyMOE	Adj Poverty	Adj Poverty	PAI	PAI MOE	
2	10073	138825	10073	1.066	0.077	
3	11705	171872	11705	1.016	0.069	
4	25972	721076	25972	1	0.036	
5	11590	170528	11590	0.999	0.068	
6	23095	732794	23095	0.994	0.031	0.110058685
7	18628	447210	18628	0.956	0.04	

the values are significant. Based on the result of our formula, the test value is only 0.110, which is less than the *Z* value (Figure 6.6). This means the PAI for the two states is not statistically different from each other. Based on this outcome, you could argue that Maryland is as equally deserving as Rhode Island to receive a bonus.

10. **Calculate statistical difference for other values**. Let's see if any of the other PAIs are not statistically different. Since the fourth-place cutoff is what matters and the order of ranks below this doesn't, we need to test each state against fourth-place Rhode Island. Modify the formula in cell K6 to lock the values for Rhode Island: =ABS((I5-I6)/SQRT((J5/1.645)^2+(J6/1.645)^2)). Copy and paste the formula down the column. It turns out that there are actually four states whose PAI values are not significantly different from Rhode Island's, as their test value falls below 1.645: Maryland, Connecticut, Washington, and the District of Columbia (Figure 6.7). Ninth-ranked Pennsylvania is the cutoff: Since it's test value is higher than the *Z* value, its PAI is significantly different from Rhode Island's. To make sure your formulas are correct, check a few of the values against Cornell's calculator. The calculator won't return the test score but simply indicates whether the difference is significant or not. Save your work.

This exercise reveals the challenge of working with sample-based estimates, which have a degree of ambiguity that doesn't exist for counts derived from total enumerations or administrative records. Statistically, the PAI for the states ranked fifth through eighth is not different from the PAI for the fourth-ranked state, but the real outcome between being ranked fourth and fifth is sharing in a $12 million bonus versus receiving nothing. You could argue that these states are in fact tied, and the money should be split among all of them (bearing in mind that this example only considers the first aspect of the award mechanism).

FIGURE 6.7 ⬢ PAI VALUES THAT ARE NOT SIGNIFICANTLY DIFFERENT FROM THE FOURTH-RANKED VALUE



	A	B	C	D	E	F	G	H	I	J	K
1	State	Adj Monthly SNAP	CA SSI AdJ	Monthly FDPIR	poverty125	povertyMOE	Adj Poverty	Adj Poverty	PAI	PAI MOE	
2	Delaware	148029			138825	10073	138825	10073	1.066	0.077	
3	Hawaii	174558			171872	11705	171872	11705	1.016	0.069	
4	Oregon	721309		765	721841	25972	721076	25972	1	0.036	
5	Rhode Island	170430			170528	11590	170528	11590	0.999	0.068	
6	Maryland	728046			732794	23095	732794	23095	0.994	0.031	0.110058685
7	Connecticut	427447			447210	18628	447210	18628	0.956	0.04	0.896601716
8	Washington	988779		3370	1069229	29429	1065859	29429	0.928	0.026	1.604302707
9	District of Columbia	131341			143434	10533	143434	10533	0.916	0.067	1.430254456
10	Pennsylvania	1858849			2079210	39893	2079210	39893	0.894	0.017	2.464233284

Under federal regulations, the USDA has flexibility in deciding what criteria to use, how to generate the index, and how to award the bonuses based on the outcome. We are not saying that they are violating the regulations. For example, the regulations specify that they should use 130% of poverty as a ratio, but they use 125% as the former value is not published by the Census Bureau. The USDA previously used poverty estimates from the Current Population Survey but switched to the ACS as the estimates were more precise. One could argue that the rank and thus the award is based on the median value of the interval, regardless of what the MOE is. In public policy, there is always a trade-off between being statistically precise and using a simpler method that more people can grasp and thus agree with.

One final note: You should only create derived values for geographies (like the neighborhood in our first exercise in Chapter 6) or categories (like the PAI in this exercise) that do not already exist. There's no need to aggregate values for census tracts to counties because the Census Bureau has already created these estimates. In this exercise, the value that we calculated for the number of people below 125% poverty was actually already published in another table—summary table S1701 Poverty Status in the last 12 months (this table has a lot more variables in it relative to the table we used). The MOEs for published census estimates are often more precise than the ones we calculate ourselves. That's because the Census Bureau generates its estimates directly from the individual sample records, while our estimates are derived from total estimates. For example, the number of people below 125% poverty in Alabama was 1,058,934 in the 1-year 2016 ACS. Our derived MOE for this estimate was 27,389, but in the Census Bureau's published table, it was 26,546. In circumstances like this exercise, where a small difference could translate to getting millions of dollars or not, we'd want to use the most accurate estimates possible (but in this case, it was better that you got additional practice aggregating the data yourself!).

11

CENSUS DATA DERIVATIVES

11.4 MEANS AND MEDIANS FOR AGGREGATES

Calculating a new median for aggregates is more complex than calculating a new mean, but we will walk through each step since the process is not widely documented. Remember that the median value represents the case that falls in the center of a distribution. To calculate a new median, we would need to combine the records of individual respondents from these two census tracts, sort them, and select the middle value. This isn't possible as we only have access to summarized data. The solution is to use a method called statistical interpolation, where we can derive the median based on ranges of data. The Census Bureau publishes interval data for a number of variables, such as Table B19001 Household Income, where the number of households are counted by income brackets.

The California State Data Center (2016) published a brief tutorial demonstrating how to calculate a derived median and its associated margin of error (MOE). The following steps are based on their example, using our two New Orleans census tracts. The published 2012–2016 medians for the tracts are $68,780 (±5,160) and $64,333 (±12,774), respectively. Table 11.1 is the original household income table for these two tracts. We would aggregate this data for the two tracts using the techniques learned in Chapter 6. Figure 11.1 displays what the final layout of the spreadsheet should look like, and we will refer to it throughout the example. The top of the sheet in this figure displays how the aggregated household data should be reformatted (in the first step), while the bottom of the sheet shows the results of the formulas for each of the steps that we'll take below. The cells that are highlighted in the figure represent values that are used in the calculations. A copy of this spreadsheet as well as the

TABLE 11.1 ⬢ HOUSEHOLD INCOME FOR TWO CENSUS TRACTS IN ORLEANS PARISH, LOUSIANA

Income	Tract 38		Tract 135	
	Households	MOE	Households	MOE
Less than $10,000	66	48	53	34
$10,000 to $14,999	16	17	29	26
$15,000 to $19,999	28	27	53	60
$20,000 to $24,999	70	37	113	92
$25,000 to $29,999	18	21	84	49
$30,000 to $34,999	22	23	43	37
$35,000 to $39,999	25	23	20	21
$40,000 to $44,999	35	28	86	67
$45,000 to $49,999	23	37	16	23
$50,000 to $59,999	27	25	99	49
$60,000 to $74,999	145	64	160	73
$75,000 to $99,999	101	56	193	90
$100,000 to $124,999	91	51	88	54
$125,000 to $149,999	45	35	49	38
$150,000 to $199,999	27	24	88	48
$200,000 or more	96	53	150	77
Total	835	126	1,324	157

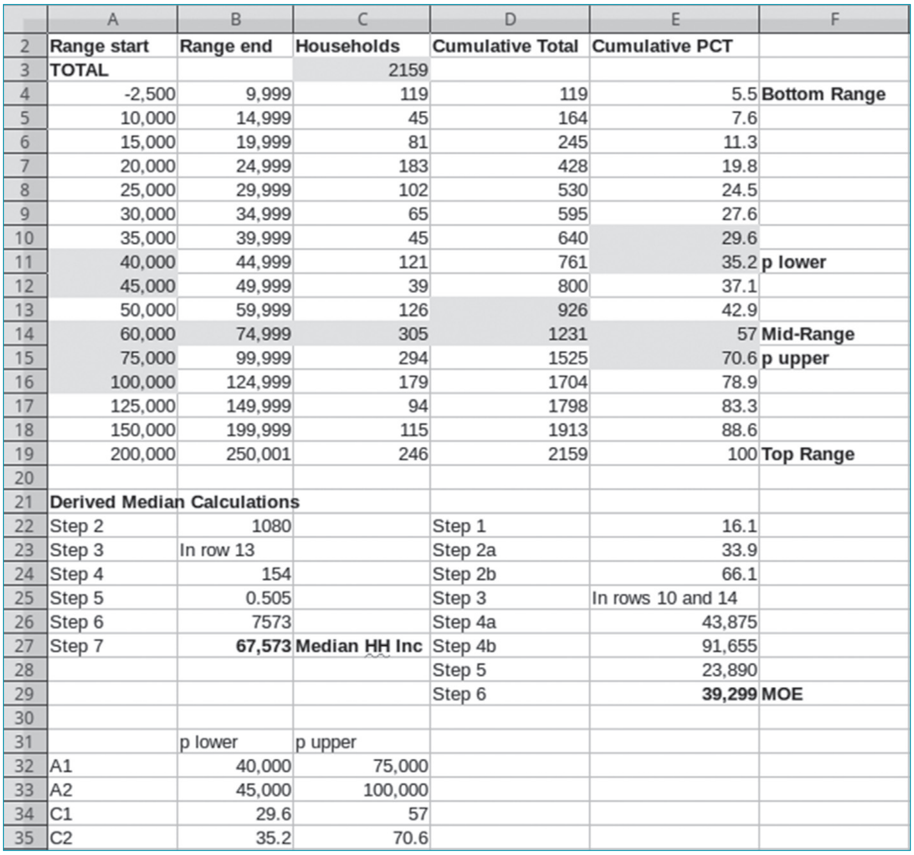
Source: 2012–2016 ACS Table B19001.

original data for these two tracts are stored in Chapter 11 Supplementary Example folder.

To calculate the new median, you need to do the following:

1. **Reformat the data**. Create a table that resembles the top of the sheet depicted in Figure 11.1. We take the labels from the ranges and split them into separate columns that contain just the values, so we can incorporate them into our calculations. We replace the bottom and top ends of the ranges with values suggested by the Census Bureau: −2,500 for the start and 250,001 for the end. Using spreadsheet formulas, we calculate running totals for both the number

FIGURE 11.1 ⬢ CALCULATING A NEW MEDIAN FOR AGGREGATED ACS DATA IN A SPREADSHEET



	A	B	C	D	E	F
2	Range start	Range end	Households	Cumulative Total	Cumulative PCT	
3	TOTAL		2159			
4	-2,500	9,999	119	119	5.5	Bottom Range
5	10,000	14,999	45	164	7.6	
6	15,000	19,999	81	245	11.3	
7	20,000	24,999	183	428	19.8	
8	25,000	29,999	102	530	24.5	
9	30,000	34,999	65	595	27.6	
10	35,000	39,999	45	640	29.6	
11	40,000	44,999	121	761	35.2	p lower
12	45,000	49,999	39	800	37.1	
13	50,000	59,999	126	926	42.9	
14	60,000	74,999	305	1231	57	Mid-Range
15	75,000	99,999	294	1525	70.6	p upper
16	100,000	124,999	179	1704	78.9	
17	125,000	149,999	94	1798	83.3	
18	150,000	199,999	115	1913	88.6	
19	200,000	250,001	246	2159	100	Top Range
20						
21	Derived Median Calculations					
22	Step 2	1080		Step 1	16.1	
23	Step 3	In row 13		Step 2a	33.9	
24	Step 4	154		Step 2b	66.1	
25	Step 5	0.505		Step 3	In rows 10 and 14	
26	Step 6	7573		Step 4a	43,875	
27	Step 7	**67,573**	**Median HH Inc**	Step 4b	91,655	
28				Step 5	23,890	
29				Step 6	**39,299**	**MOE**
30						
31		p lower	p upper			
32	A1	40,000	75,000			
33	A2	45,000	100,000			
34	C1	29.6	57			
35	C2	35.2	70.6			

of households and the percentage of the total. We don't need to use the MOEs for the households, so it's best to hide or remove this information from the spreadsheet to avoid confusion.

2. **Determine which range holds the midpoint**. Divide the number of households by 2, and if the result is a fraction, round up.

$$2,159 \div 2 = 1,080$$

3. **Identify the range that contains this midpoint**. Using the cumulative total, the 1,080th household falls within the $60,000 to $74,999 income bracket. The previous bracket ends at the 926th household and this bracket ends at the 1,231st, so 1,080 falls within this range.

4. **Calculate how many households in the midrange are needed to reach the midpoint**. We subtract the number of households from the previous range from the midpoint.

$$1,080 - 926 = 154$$

5. **Calculate the proportion of the number of households in the midrange (60,000–74,999) that are needed to get to the midpoint**. Using the result from the previous step and the number of households in the midrange,

$$154 \div 305 = 0.505$$

6. **Apply this proportion to the width of the midrange dollar values**. First, we calculate the width, then we apply the proportion:

$$(74,999 - 60,000) * 0.505 = 7,573$$

7. **Calculate the new median**. Using the beginning of the midrange and the result from the last step:

$$60,000 + 7,573 = \$67,573$$

This result is close to the medians of the individual tracts, so this outcome seems reasonable. Calculating the MOE is more involved and uses formulas that are typically employed when creating derivatives of public use microdata.

1. **Approximate the standard error of a 50% proportion**. The formula for this is printed below, where B represents the total number of households (the base) and DF is the design factor from the Public Use Microdate Samples (PUMS) files. The DF is a table of constants published for different variables at the national level and for individual states. We'll discuss the source for these variables at the end; the DF for income variables for the State of Louisiana in 2012–2016 is 1.5:

$$SE(50\%) = DF * \sqrt{\frac{99}{B} * 50^2}$$

$$1.5 * \sqrt{\frac{99}{2,159} * 50^2} = 16.1$$

2. **Subtract and add the standard error from the last step to 50%**. This will give us the lower and upper bounds.

$$\text{p\_lower} = 50 - 16.1 = 33.9$$
$$\text{p\_upper} = 50 + 16.1 = 66.1$$

3. **Identify the ranges in the distribution that contain the lower and upper bounds**. Using the cumulative percentages, 33.9 (p_lower) falls within the 40,000 to 44,999 range; the previous range stops at 29.6, and this range stops at 35.2, so 33.9 falls within it. The 75,000 to 99,999 range contains 66.1 (p_upper). Note that it is possible that your lower and upper bounds could fall within the same range; we'll discuss this variation at the end.

4. **Approximate the lower and upper bounds for a confidence interval about the median**. To do this, we need to identify these four variables for each of the bounds:

   a. A1: the smallest value in the range (40,000 for p_lower and 75,000 for p_upper)
   b. A2: the smallest value in the next higher range (45,000 for p_lower and 100,000 for p_upper)
   c. C1: the cumulative percentage of households less than the A1 range (29.6 for p_lower and 57.0 for p_upper)
   d. C2: the cumulative percentage of households less than the A2 range (35.2 for p_lower and 70.6 for p_upper)

In your spreadsheet, it's helpful to explicitly label these values as depicted in the bottom of Figure 11.1, as this can help you avoid mistakes. Instead of hard coding the value beside the label, you can provide a reference to the cell that contains it, that is, ="cell value." Once you've identified the values, you can insert them into this formula and calculate the lower and upper values:

$$\text{Bound} = \left(\frac{p - C1}{C2 - C1}\right) * (A2 - A1) + A1$$

$$\left(\frac{33.9 - 29.6}{35.2 - 29.6}\right) * (45,000 - 40,000) + 40,000 = 43,875$$

$$\left(\frac{66.1 - 57}{70.6 - 35.2}\right) * (100,000 - 75,000) + 75,000 = 91,655$$

5. **Approximate the standard error of the median**. Using the upper and lower bounds from the last step:

$$0.5 * (91,655 - 43875) = 23,890$$

6. **Calculate the MOE**. Remember from Chapter 6 that 1.645 is the constant that applies to all ACS estimates, which are published at a 90% confidence interval:

$$1.645 * 23,890 = 39,299$$

The median household income for these two census tracts combined is $67,573 (±39,299). The MOE is quite high, but this isn't unusual for this procedure. These interpolation methods produce more conservative or higher MOEs than you would get if you were creating a median using the original, individual data values. If we used this procedure to calculate the median and MOE for a known published value, such as income for one of the census tracts, the calculated estimate would differ and its MOE would be higher.

One variation that you may encounter is that your bounds for p_lower and p_upper could fall within the same value range. If this occurs, you would only need to create one set of A and C values in Step 4 and apply them to both the lower and upper bounds formula (so the only variable that would differ between the two formulas would be the p_bound variable).

The Census Bureau publishes and annually updates the DF variables used in Step 1 in technical documentation titled "Public Use Microdata Sample (PUMS) Accuracy of the Data" (U.S. Census Bureau, 2017). This documentation is published in PDF format and is stored under the PUMS documentation section of the ACS website. Factors for each variable are published for the nation as a whole and each individual state (including the District of Columbia and Puerto Rico) in the appendix of this document. You will need to look up the appropriate factor for the specific year, state, and variables you're working with. In this example, the DF for household or family income variables for Louisiana for 2012–2016 was 1.5; for the entire nation, the variable is 1.6.

REFERENCES

California State Data Center. (2016, April). *Recalculating medians and their margin of errors for aggregated ACS data* (January 2011 Network News, revised 2016). Sacramento: California State Data Center, Demographic Research Unit. Retrieved from http://www.dof.ca.gov/Forecasting/Demographics/Census_Data_Center_Network/.

Nesse, K., & Rahe, M. L. (2015). Conflicts in the use of the ACS by federal agencies between statutory requirements and survey methodology. *Population Research and Policy Review*, *34*, 461–480.

U.S. Census Bureau. (2017). *Public Use Microdata Sample PUMS accuracy of the data (2012–2016)* (Tech. Rep.). Washington, DC: Author. Retrieved from https://www.census.gov/programs-surveys/acs/technical-documentation/pums/documentation.html.

U.S. Department of Agriculture. (2018, February). *Calculating the Supplemental Nutrition Assistance Program (SNAP) Program Access Index: A step-by-step guide for 2016* (Tech. Rep.). Washington, DC: Food and Nutrition Service. Retrieved from https://www.fns.usda.gov/snap/calculating-supplemental-nutrition-assistance-program-snap-program-access-index-step-step-guide.