

Transcript: American Community Survey Resources, Shortcuts, and Tools Webinar (June 26, 2024)

Mark Mather, PRB: Okay, well, I think we should go ahead and get started. Hi, everyone. Thanks for joining today's webinar on ACS resources, shortcuts, and tools. I'm Mark Mather, and for those who don't know me, I help manage the ACS Online Community website and other activities in partnership with the U.S. Census Bureau.

I am very excited to introduce the three speakers in today's webinar. Lillian Kilduff is a Research Analyst at PRB and will provide a brief overview of the ACS Data Users Group and Online Community. Following Lillian, we'll have Mary McKay, who's a survey statistician in the American Community Survey Office. Mary is going to show you how to access the ACS through the Census Bureau's API and microdata extraction tool, also known as the MDAT. And then we have Kevin Kane, who's a program manager with the Southern California Association of Governments. Kevin is going to describe how he uses R and the tidycensus package to access and output ACS data.

A few housekeeping notes. We're going to save the Q&A until the end. We do have a large number of participants. We encourage you to use the raise hand feature in Zoom, and then we'll try to unmute you to ask your question, but you can also feel free to use the question box at the bottom of your panel there, and you can type in your questions at any time during the webinar.

Closed captioning is also available as an option at the bottom of your screen. And in addition to our three panelists, we also have several other Census Bureau staff members on standby to answer your questions today. And finally, this webinar is being recorded, and we will send you a link to the recording after the event. And with that, I'm going to turn it over to Lillian.

Lillian Kilduff, PRB: Thanks, Mark. I'm going to be talking about the ACS Online Community today and also showing the new upgrades. If you haven't already seen to the look and feel of the website, I'm going to go ahead and share my screen real quick. Right here. Okay. Sorry about that. Okay. Um, so I'm going to provide the brief introduction to the ACS Online Community.

So here's an overview of the presentation today. First we're going to do a quick recap of the American Community Survey itself. Then we're going to talk about the ACS Data Users Group and Online Community. Then we can go over the tabs of the ACS Online Community, and that includes the discussion forum, the ACS resources, webinars, and conferences tabs.

After that, we're going to talk about the ACS Online Community itself. So behind the scenes, how many members do we have, threads and replies, page views, response rates, and then also talk about the discussion forum topics that often get viewed. We'll go over the site upgrade if you haven't already seen the changes and talk about how to join the ACS Online Community.

So just to review, if you're new to the American Community Survey, people use the American Community Survey to get, uh, data on the demographic characteristics. So that would include social characteristics, economic, housing, and demographic. And you can see some of the examples in those parentheses there. The data products include one-year estimates, one-year supplemental estimates, five-year estimates, and you can access those through many tools including tables, the summary file, and PUMS.

Here is a quick hierarchy of the geographies available. So we have from the nation down to block groups.

When it comes to the ACS Online Community, this is a partnership between us at PRB and the American Community Survey Office at the Census Bureau. The ACS Online Community's purpose is so that ACS data users can share tips and tricks, questions, materials, and then also we post announcements about things like today, the webinar. Membership is free and open to all ACS data users and new ACS data users. The group is led by our steering committee, and we try to pick a steering committee that represents all different data users, local governments' data users, geography, geography data users. And we just had a new steering committee this this year.

So I'm going to show you the home page. Okay. This is fine. Here is the home screen of the new ACS Online Community. Here is just what I talked about, the purpose of it. Here's some quick facts about it. And we also have the most frequently asked questions. That's based on questions from data user surveys and also from the most viewed and interacted discussion forum post. You can view more FAQs on the FAQs page from there. We also have latest discussions, people who are posting in the ACS Online Community. We have a link to the Census Bureau website.

The discussion forum is the main part of the ACS Online Community. Here is an example of a discussion forum post. So a data user is asking a question, and then we get a reply from another data user. You can upvote replies, and if you become a member of the ACS Online Community, you can do things like uploading, replying, and adding to the discussion forum. Here you can see the views, replies. You can also add tags to new discussion forum posts. And then we have more information over here.

Next is the ACS resources page. Here you can see a lot of different links to ACS resources under these helpful headings. If you aren't already familiar, the ACS handbooks are a great place to start, and we also have handbooks that are catered to certain data users.

Here is our webinar page. So this is the webinar we're having today. And then we also have links to past webinars with recordings, information, and even the slide decks.

We hold a biannual conference every year. The latest conference was the 2023 ACS Data Users Conference. We have the agenda from that. That includes the recordings of the presentations and also the slides as well. We have the previous conferences, and those include that information as well.

I'm going to go back to my PowerPoint now.

Again, this is the discussion forum. Here is an example of a notification of a Federal Register Notice. That's one of the examples of a discussion forum post that's helpful to ACS data users.

Okay, so behind the scenes we can talk about the membership. We have over 5,600 members as of the end of May. Here you can see the fiscal year 2022 and 2023, and membership can vary over that time. And usually when we, when we have events like conferences or a new series of events called ACS on the road—we just went to the Texas Demographic Conference—we can see an increase in membership.

There's a lot of discussion forum posts, and they get a lot of replies. Here again we have, uh, the total number of the threads and replies across the two last, last fiscal years.

And here are the number of page views that the ACS Online Community gets. If you've ever googled a question about the American Community Survey and its data, a lot of times the first Google result is the ACS Online Community itself. And you can see that overall, uh, with the last fiscal year, the page views and the ACS Online Community have increased.

The great thing about the ACS Online Community is that we do have a great response rate. So you can see that just within one day, if you post a question or an announcement or a comment, you get a pretty good, uh, you know, response rate.

And here are the top 10 discussion topics. We get a lot of questions about calculating margin of error, especially, uh, zip code-level geographic questions.

So onto our site upgrade. This is how the ACS Online Community used to look. You may remember it this way, but now live on the site, we have this new upgrade that's, uh, more intuitive and more modern in the look and feel. This is how the discussion forum used to look. And now here is the upgraded website.

So finally, how do you get involved with this site? You don't need to be a member of the ACS Online Community to view the posts, but you do need to be a member to post in the Online Community, comment, uh, and also upvote. You can tailor the email notifications that you get, so, uh, to new threads and comments. And these are all optional. So if you're hesitant about joining the ACS Online Community because you're worried about a lot of email notifications, you can cater those. You can also bookmark discussion forum threads so you can reference those whenever you have questions about a certain topic.

And again membership is free and signing up is very simple. First, you click on the sign up button in the top right and then just answer a few questions. We use this information so that we can better cater to different data user groups.

Finally, there is a picture of one of the ACS data users conferences, and please give us your feedback or suggestions.

Thank you so much. Here is my contact information if you ever have any questions. And I can either answer the questions or direct you to someone who will know your, the answer to it.

Mark Mather: Great. Thank you, Lillian. Next up we have Mary.

Mary Ana McKay, American Community Survey Office: Hello? Hello. Okay. All right, I can share my screen once Lillian is done sharing hers.

Lillian Kilduff: Yep. Um. Stop there.

Mary Ana McKay: Perfect. Knock on wood. Awesome.

Okay, so hello, everyone. My name is Mary Ana McKay. I'm a survey statistician with the Census Bureau's American Community Survey Office. I'm here to highlight two data products and tools that you may be familiar with or you've never heard of before. And just a little bit of housekeeping, I'm

going to apologize in advance if I speak quickly. I just have a ton of information that I want to share with you all, and I'm very excited to be here. I'm excited that you're all here.

So without further ado, let me get started. I want to give a broad roadmap of what I'm going to be presenting during this workshop. So to start, I'm going to dive into the ACS Public Use Microdata Sample, or PUMS. This portion is going to cover basics. Then I'm going to run through the Census Bureau's tool to access these data, and then I'll wrap up that section with some resources for you as you dive in on your own.

And then immediately following the PUMS, I'm going to jump over and give a very brief introduction to the application programming interface, the API. We won't go too much into details, but you will learn the basics, so you're hopefully able to build off what we do today as you go off onto your own data journey. And we'll go through an example API call, and then I'll share just a sample of the many, many resources available to you as an API data user before I turn it over to Kevin for the last leg of this workshop.

So before I dive into the PUMS and API, I want to remind everybody about data.census.gov. It's a really powerful tool for you as you grow your ACS data accessing skills. So many of you here today are probably familiar with data.census.gov, which is the primary way to access data from the American Community Survey, 2020 Census, and more. And I'll be sprinkling my use of it throughout my two demonstrations, but it's not the star of the show, so I'm kind of going to run through them a little bit more quickly than I would otherwise. But in an effort to be brief, I will let you know that there are a variety of how-to materials, video tutorials, webinars, and FAQs to help you use data.census.gov.

And I'm going to step aside again and just mention there are links at the bottoms of a lot of my slides. I have a colleague who will be sharing some of them in the chat, but also the PDF version of this presentation is going to have clickable links too.

So the ACS Public Use Microdata Sample can be overwhelming, but we're going to briefly cover basics to start to get you familiar and hopefully comfortable with this powerful data set.

And I want you to think about these questions: What are your main goals when accessing ACS data? Are you primarily accessing pretabulated estimates? Are you finding that the data you need are not published in these estimates? And what about when you're looking at cross tabulated estimates? How do you primarily access ACS data? Are you using data.census.gov or a third-party tool such as Social Explorer? What do the data look like on a daily basis? And finally, with the tool or tools you are using, what limitations do you face accessing ACS data?

So these questions might have different answers depending on the day or the data you need. So in some cases, the tool that we are going to explore will be the best option, but other times another method will work better. It's all about the best way to address your needs. And I always check data.census.gov—I'm going to say this constantly throughout my portion—just to see if there's pretabulated estimates for the data product and the geography of interest. But in cases that I need something a little bit more specific, I'll hop over to PUMS.

So, for example, today I'm curious about poverty among veterans by age, and I know I can find tables in data.census.gov that might get close to what I need but not quite exact. And luckily, PUMS is going to be able to step in and get us the table that I want.

So I want to introduce a few PUMS basics before we work on an example. And finally I will share some resources that you can access on our website.

So again, when I say PUMS, I am referring to the Public Use Microdata Sample. ACS data products are released about one year after the data are collected, and the PUMS is a publicly available subsample of ACS records. The one-year PUMS estimates are a subsample of data collected over a calendar year, 12 months, and they constitute approximately 1% of U.S. households. Whereas the five-year PUMS combines data collected over 60 months, or five years, and they constitute approximately 5% of all U.S. households.

Additional restrictions are added to protect data confidentiality, such as including broader categories of data or grouping together extreme values in the form of top and bottom coding. And you're going to see a couple examples of this top coding in my demonstration.

PUMS files allow data users to calculate their own estimates and margins of errors that may not be available on data.census.gov. Statistical software is recommended when working with PUMS data unless you are working with our microdata access tool on data.census.gov, and this is the tool that I'm going to be demonstrating today.

So here are some examples of why you might want to use the PUMS. These data come in handy when you are looking for cross tabulations that might not be part of the standard table packages released in the ACS. For example, you could be looking for specific poverty thresholds or income levels for veterans at a specific age ranges like I am today. Again, always check data.census.gov and the pretabulated estimates. They may have exactly what you need.

This information is going to be a little bit heavy, but I want to mention it before we continue. So PUMS data provide individual records that data users must aggregate to form estimates. Unlike in data.census.gov, there are no pretabulated data. Weights are included on the PUMS files so that data users may create weighted population estimates. If you are working with housing records, you will use the housing weights. And if you're working with person records, you're going to use person weights.

When working with a merged file that includes both housing and person records, person weights should be used to produce estimates for person characteristics. Housing characteristics cannot be tallied from this merged file without taking extra steps to ensure that each housing weight is only counted once per household. In today's example, I am using all person records.

And then replicate weights, those numbered one through 80 are used for calculating replica estimates needed to calculate standard errors. These standard errors are necessary in order to calculate the associated margins of error or MOEs, and we won't be going this in-depth for this presentation, but there are guided examples that I can direct you to for more.

The five-year PUMS is the equivalent of five one-year files, so again includes about 5% of all U.S. households. So people often ask, and you may be wondering, what is the benefit of the five-year

PUMS? So there's some nice standardization for the five-year PUMS that you can't necessarily get by merging five- to one-year files. For example, there are new weights that are produced for these records so that the weighted population matches the latest population estimate. Dollar amounts have an adjustment factor to standardize them to the latest year, so that no one is comparing varying levels of inflation. Other coding schemes are updated, such as ancestry and occupation, so you don't have to recode those yourself.

I'm going to focus on a limitation data users might experience someone accessing PUMS, and that's geography. To ensure the confidentiality of ACS respondents, the Census Bureau has to balance geographic detail with detail in the data. There are more than 250 variables on a single PUMS person record. This means that we cannot identify as many small geographies in the PUMS as users might hope. We can put the region, division, and state on the file, but the only other geography is something called a Public Use Microdata Area, a PUMA. PUMS is not designed for statistical analysis of small geographic areas, but the PUMAs can still be used for focus analysis in counties and cities of about 100,000 people or more as well as many metro areas.

So I want to spend a little bit more time here on PUMAs. PUMAs are areas with a population of, again, at least 100,000, which is large enough to meet disclosure avoidance requirements. PUMAs are identified by a five-digit code that is unique within each state. These geographies are redefined after each decennial census and are defined by either the state data center or, in some cases, the Census Bureau's regional geography staff. For example, the 2020 PUMA definitions were introduced with the 2022 PUMS files.

As with many geographic concepts, seeing PUMAs on a map may help you understand them better. So as you can see, some PUMAs are small and others are large, because, again, PUMAs are built on population and not geography. The smaller PUMAs here on this map are mainly concentrated in the Buffalo and Rochester regions of this map, and some counties in this region that have smaller populations are combined together as part of a multi-county PUMA.

So I use data.census.gov here to visualize geographies. This is a screenshot that shows, um, the PUMAs that make up Marin County, California. So as you can see, there are two that make up the county. So you can combine data from both to approximate estimates for the county. The primary difficulties occur when we get further away from urban centers to counties with smaller populations, which are then again combined with other counties to make PUMAs. And in these cases it becomes less feasible to infer data about the individual county. Furthermore, while I am showing you an example here of PUMAs that adhere to county boundaries, it is not actually a requirement that PUMAs be designed that way, although it is recommended.

And I want to acknowledge really quickly that some of you might know that data.census.gov now has an address lookup option in the search bar. I just want to let you know that right now, PUMA geographies do not pop up when you use that option. I just tried it before, but hopefully someday you'll be able to put in an address and see what PUMA that falls into.

All right, let's get our hands dirty with PUMS data. And to start, I'm going to heed my own advice and go directly to data.census.gov. I'm going to first see what tables I might find. And again, I'm going to zip through this because I want to focus more on the microdata access tool. I'm going to use the advanced search feature.

And again, today I'm interested in poverty among veterans by age. I'm going to apply two filters: "veterans" and then I'm going to select "poverty" to see what tables come up. I'm going to click the search bar. And I see here there's actually a table age by veteran status by poverty status. And it's a little bit more detailed; it also has disability status. But it does have generally what I'm looking for. So again I said poverty among veterans by age.

But as I'm looking through this table, the age ranges are not quite what I'm looking for, and I'm actually interested at below, at, and above poverty. So this just has two thresholds; I want to add a third. So in any other day but today this table might actually serve the exact purpose I'm looking for, but now I'm going to use the PUMS data to get what I really want.

I'm going to click on the logo to go back to data.census.gov home page, and on the top right, you probably can't see it, there's a little button that says apps. I'm going to click on that. And it's this first option here that says microdata. So this is what you're going to see. The default data set is the ACS one-year PUMS. And the select vintage is 2022. And perfect, that is exactly what I want. I'll click next so I can select my variables.

So before I select my variables, I want to search for what they might be called. I know I want poverty, I want veteran status, and I want age. So I like to use the label option here—and I'm going to zoom in, I might have to zoom in and out—I like to use the label here to use keywords to see what pops up. And we also have PUMS documentation with data dictionaries, so you can do the same thing before you get into this tool.

So for the first one I'm going to type in "poverty," and I see this income-to-poverty ratio recode; I selected this for, uh, today's demonstration because this is the poverty variable in PUMS, so I want to show people how to use it. It does give me a little bit of a warning here that the variable is continuous, but we're going to make a custom group with this variable to be able to put on our table, so we don't have to worry about that quite yet.

And so for my veteran's variable I'm going to type in "veterans" or "veteran." And I'm going to open the detail of the three variables that show up. And this isn't quite what I'm looking for. This veteran period of service is a little bit more detailed. I just want to know if a person has ever served in the military or not.

So now I'm going to try typing another keyword. So I'll do "military." And luckily for me I have this military service. Let's cross our fingers. And yes, okay, this is exactly what we want. We have a value 2 that says "On active duty in the past, but not now." So that's how I'm operationalizing veterans. I'm going to select this variable. So now I have two. And my final one is age. So it's right here at the top. It's going to give me that same warning that the variable is continuous, but that's totally fine.

So from here we have our three in the data cart. We're going to click on View Table and see what we have to start with. So for most situations simply selecting the variables is not going to be the last step for you, for your table, unless by some chance it's laid out exactly how you want it and the categories are exactly what you want.

So at first glance, there is a lot going on, and I'm going to rename the table just to keep myself organized up here. You can go in and change that title as much as you want, but I'm just going to do

“Poverty x Age for Veterans,” so that’s just going to keep it organized in my head as to what we’re doing.

So we see that the default table has military, that military variable on the columns. We have nothing on rows. And then we have two variables here in the values in table cells. Then in this drop-down this is the first thing I’m going to change. I’m going to click on this and select Count. So this is going to give us a value for how many fall in each category.

So I’m going to organize to make variables, and then I’m going to put them so we have our universe limited to just veterans. And then I’m going to create grouped categories for age. And then income-to-poverty ratios on the columns will be three thresholds. Or I’ll make a threshold of three.

So to put in simple terms, our universe is going to be just veterans. My columns are going to be the recode of that income-to-poverty ratio. And then finally the rows are going to be simplified categories of age. And what’s great about this tool is you can organize and flip-flop your rows and columns super easily, so if you don’t like what we have planned, we can change it when we’re done.

So we’re going to start first with making our universe what we want, which is just veterans. So I’m clicking on the variable. I’m going to deselect everything that says Include in Universe. And I’m only interested in Value 2: “On active duty in the past, but not now.” I’m going to select that option, and I like to click into View Table just to see kind of what we’re working with with every change that I make. So now I see my universe is only limited to my definition of veterans.

So now let’s move on and make the age category. So I’m going to click on the Age variable. I’m going to click on Create Custom Group. From here we’re going to use the Auto Group feature. I’m going to change the start age to 17 because that’s generally the cutoff date to join the military. And then for this, this is an example of a top-coded variable, we have 99. So anybody who’s 99 years or older is going to be in this category. And then I want groups of 10 years. It’s not going to be perfect with the values that I have, but for what I need, this is going to be fine. And I’m going to click Auto Group, and you see that it makes those groups for you.

The last thing I’m going to do is there’s a Not Elsewhere Classified category. I’m going to click on Edit Group. These are all the values that aren’t in the groups that I just designated. I’m going to toggle to show off the table. So I’m going to toggle that on, and you have to click Save Group. So now this isn’t going to show in my table. Let’s view the table and see what we have. It doesn’t show up, but we’re just going to click and drag, and to keep myself organized, we have the rows is what we’re going to have for age. So I just clicked it and dragged it over to On Rows. And we’ll see. Now we have account for the people who are veterans in these different age groups.

And the last thing we have is to make the poverty variable. So again I’m clicking on the POVPIT variable. And just to look at this, it is continuous. And I want to explain a little bit more about what the numbers mean before I go in and make my custom group. So for this variable, less than 1 or 100%, because this is a percentage, is below poverty; 1 or 100% is at poverty; and above 1 or 100% is above poverty.

So these are the actually the three categories I’m going to create. But this is an instance where you can really go where your research question or your need takes you. For example, I know that 200%

poverty is a threshold a lot of data users need, and there are limited options on data.census.gov. So using PUMS here is, you're going to be able to get that.

So the calculation for this specific variable is simply to divide income by poverty thresholds, which are determined by number of children, size of family, and inflation. So for this I'm going to click on Create Custom Group. I am not going to use the Auto Group feature. I'm going to dig in right here where it says Group Label. I'm going to start with Below Poverty. And again you can go in and change these group labels. Um, as you're going through, if you want to relabel it, you're able to do that.

So I'm going to click on below 501%. The bottom value I want is zero. And then the top value I want for this one is 99. I'll click Save Group. So it makes that for me I'm going to click back into Not Elsewhere Classified. Let's do at poverty. And this is going to be a single value. You can do that. So just when we're looking at estimates, note that this only has one single value in it. So we have 100 to 100, Save Group.

And then finally we're going to have above poverty. We're going to select the remaining of the between 101 and 500. And then since this is another top-coded variable, I want this 500% or more because that's above poverty. I'll click Save Group. The last step similar to that Auto Group you're going to click into, Not Elsewhere Classified. I don't want this on my table so I'm going to toggle it off, Save Group. And now we'll view table.

So again right now POVPIT doesn't show, that Recode doesn't show. But I'm going to click hold and drag on to columns. I can actually take the military variable off the table because it is my universe. I don't need to have it on there. It's included. And here is the example of the table. So now I have the poverty thresholds for different age ranges among veterans.

So I didn't dive into this. But I want to mention that you can click Change Geography up here at the top. And you see that we have the geographies that we talked about. And the default is going to be the United States. And since PUMAs, the Public Use Microdata Areas, have populations of 100,000 or more, all of them and all of these geographies are going to be included in both the one-year and the five-year PUMS. So from here you can click, download, and share what you've made. And remember that you can calculate the error with resources available on the ACS website.

So now I want to go briefly and share some few links with valuable resources for you. So I do my best learning when I am practicing. So if you're like me, I like to follow along with webinars that have some activities to check, and I put together a list of videos to see step-by-step directions for various aspects of the MDAT tool. So the data gems are going to be shorter, more brief videos, whereas the webinars go into a little bit more detail.

And I'm going to make a plug for the PUMS documentation page. I did mention it, but we didn't go into it. It has all the resources you're going to need for every data release. You can explore user guides, data dictionaries, and more. And this is also where you're going to find directions for calculating variances.

And finally, I think a really great resource that we spent a lot of time perfecting, and Lillian talked about it briefly, are the data users handbooks. We do have one for PUMS users, and I also don't want to spoil the next part of my presentation too much, but you can find the PUMS on the API.

Um, and with that, that's the worst segue I've ever had, so again, I apologize, but now we're going to jump immediately into talking about the Census Bureau's application programming interface. So let's take a deep breath and move on to the next part of the workshop.

So I want you to think again about the same questions we had when we were exploring PUMS data. So what are your main goals when accessing ACS data? Are you primarily accessing pretabulated estimates? Are there a few variables within a single table that you find yourself going to more and more? And what about variables across different geographies or across years? How do you primarily access ACS data? Are you using data.census.gov or third-party tools such as Social Explorer? And what do the data look like on a daily basis? With the tool or tools you are using, what limitations do you face accessing ACS data? Being able to answer these questions can determine if the API is a good option for your needs.

Now on to the basics. When you use the API, imagine that you are in a strawberry field since it is summer. The strawberries are data points you seek, and in order to go get them, you are going to be running calls or going around the field and picking the ripe strawberries. [Data.census.gov](https://data.census.gov) itself is a fellow strawberry picker. What we are doing today is just a smaller example of what data.census.gov does through its website. We are trying to directly access the data in a very simple way.

So some of you may be creating dashboards on your websites that users will access to get different data to display, given certain criteria. Others might be trying to make data visualizations, and there may be some of you who are using R to run analysis. It's also okay if you are none of these types of users. The API can still be a very simple process to get the estimates that you want.

As I was just describing what uses the Census API might be for, here are some more specific examples. What if you simply need just one variable, let's say percent below poverty level for individuals under 18 and nothing else within the table? What if you wanted to grab all the census tracts within a county in Delaware? How about an estimate for an individual below poverty level at the census tract, county, state, and national level? It could just be that you have a data point that you're trying to easily access year after year. I'm going to show you some ways to simplify that process for you using the API. And I will say this, and I've said it several times before using the API, consider checking out data.census.gov.

So with that let's run through an API call. These are the ACS data tables that you can find on the API. In data.census.gov, the second column here is what the table ID starts with. For our example today we're going to be using subject tables from the five-year estimates. So we're going to be using this here. So after you put the beginning of the call, you're going to put in the variables the tables and the geographies you want, but we're going to get there in a second.

We're going to start with data.census.gov, like I've said a million times already. And just for the purposes of time, I have screenshots here. So I typed in "poverty" because that's what I'm interested in for this example. I found Table S1701. And then I limited my geography to Wyoming County, New York. That's my hometown is there. And this is a smaller county, so it's going to be the five-year estimates. It has a population of fewer than 65,000 residents, so we're going to be using the ACS five-year estimates.

Now on this table I see, and I'm sorry if it's hard to read, we have below poverty levels. So we have the estimate and the margin of error. That's what I'm interested in. Just those two pieces of the entire table. This table also has percent below poverty level, which is a measure I would prefer, especially if I'm going to be comparing with other counties of varying sizes, but for this example, I'm just going to stick with the estimate and its margin of error.

I'll mention one cool thing about data.census.gov, there are many, but if you look along the top of your table, there's actually an API button now that you can click and it'll create the call for the table that you're looking at. So this can be really helpful if you're using the filter options to select geographies, and you might just want that entire table you're looking at. You can also use it as a starting point to build off. If you want a little bit more detail with your call. And I highly recommend always working off an example when you're working on calls; it makes it a lot easier than building from the ground up.

So we only want two variables: the estimate and then the margin of error. And what I'm showing here is the entire call. But we're going to dissect it before running and seeing what happens. I use the slide a few back to figure out what table type I had. And then I did a few additional steps, using some web pages to figure out (1) the variables that I need and (2) the geography.

So to start to break it down, this is the base for all Census API queries. This second set pulls out the data product year, 2022; the program, ACS; the date, the data set, ACS five-year, so this is the 2018–2022 ACS five-year; and then, finally, the table type, which is subject. And again you can refer back a few slides to see the base of all the table types. That slide will get you the portions up until this point. So once we get to this after ?get, that's where the customization gets started.

So this pulls out, this is where I'm picking the variables. And how did I get here? We're going to hop over to the website, and just for transparency, I'm using Google Chrome because that's what I prefer to use when I'm doing API. So I'm going to census.gov/api, the main website, and I'm going to scroll down to latest available available APIs and view all available APIs. From here you see what's available. I'm going to click on American Community Survey, in theory. And we divide it by the different data products, um, which I find they're all pretty similar for all of them. So it's easy once you know how to use one, you can jump around and use the other ones.

So we're selecting the five-year data. We release this for every data release. So we're here in 2022. I'm going to scroll down, and I find Subject Tables. So this is again the same for all table types, what I'm doing; you just have to make sure that you're following along with your table type.

So the first thing I'm going to start with is the second bullet down: the 2022 ACS Subject Table Variables. I'm going to click on the HTML. So for API, Ctrl+F is going to be your best friend, if it's not already. So I'm going to click Ctrl+F on my keyboard. And we're going to type in "poverty" because I want to overwhelm you briefly with what shows up.

So as it's loading, in theory, we're going to have thousands of options. So it's loading, um, there's so many of it that now it doesn't want to do it. So there's actually over 3,700 results on this page for poverty. And that's a lot to go through. So I'm going to show you a little bit of an insider secret, or at least that's what I like to call it.

Um, I'm back on S1701. I've magically loaded it for us here, and I'm going to talk about the different columns. So this is a column set 1. We have the total. And then for this table, there's a column set 2. Now what does that mean? We're going to go back to this table, the variable lists. And if I start to scroll down, you hopefully can see that there's a table ID, then there's an underscore, and a CO1 that corresponds with column 1. So I can use this as my base to Ctrl+F again. And since I'm looking at S1701, I'm going to type that in. It's going to jump me to the first time that that shows up. When I do the underscore, it's going to jump me to the section for this table.

And I know I'm looking for the second set of columns, so I'm just going to write in CO2. And luckily for me it's this first estimate in column set 2. So we have below poverty level population for whom poverty status is determined. Then the one that ends in E is going to be my estimate, and I want that margin of error, and you should too. That's going to be the one that just ends in N.

So let's hop back over to the slides to see what I did here. So I have the two variables that I found and I put them in here. I also put Name here. So to make sure that I get the geography names when I run the call. But this is not a necessary component of your call. I tend to use it just to confirm that I have the right geography, so I can run it with that, confirm I have the right geography, and then you can run it again without if you don't need it for the larger purposes of your call.

One thing I will note, you separate the variable names with just a comma. If you add a space or an additional character, you are going to get an error when you run your call. So working backwards, if you get an error, double-check your call and make sure that there's no spaces in between the commas. You can pull up to 50 variables with this method, and if you want more than 50, it's likely that you just have to pull the entire table and then work from there.

I also want to mention one more thing. You can pull variables from different tables of the same type. Say, for instance, you want to pull all of the same variable in a table series for different race iterations. So we have detailed tables for the different race and ethnicity iterations that end in A through I. You can pull the same variable from those different tables.

I also want to jump back to this name variable and give you a little bit of a warning. So it does cause a shift in Excel, especially if it's a geography within a geography. And you're going to see this when we open the file from our example here. And I'm not sure if this happens with every table type, but just keep that in mind that I know for a fact that we do not recommend using it for group calls, particularly with data profiles. So just keep that in mind that it can get a little bit messy. But again, I like to have it as a little check for me.

So before I move on, what happens if you want all variables in the table? What if you want the entire S1701? You can use a group call. So I have that down here. Um, you can also use data.census.gov if you have the geographies you selected already. That API button is going to do exactly what this is going to do for us.

So now we have the last part, which is the geography. And in many instances you will want to limit to a specific geography. And in this example I want one county. And you may be wondering how I got these numbers. And I did not, in fact, memorize every county code for every state to figure this out. I'm going to share another secret, and I think this one's a little bit more exciting, but who knows? You'll have to tell me.

So we're back on the ACS five-year API page, and we're still in the subject table section. I'm going to click on the fourth bullet down that says Examples. So this breaks it up by geographies. And since I'm looking at state and county, I'm going to look at the example API calls that I have here. And fortunately for me I've used this so much that it's already, um, calling itself out.

There's one here that has a wild card, wild card or the asterisks for county and state. So if I click on this, it's going to actually give me, um, and hopefully let's, that we've zoomed in, it's giving me all counties in all states. It does have a random variable. Um, just to call it out again, as an example, you can leave that in there, or you can delete it with the comma and just have name. So now you have the call to get all of the counties in all of the states.

And again, your best friend, at least for now, is Ctrl+F. You're going to start to type in your geography of interest. And luckily for me, the first Wyoming on this list is actually Wyoming County. So I can use context clues here and see 36 for all of the New York counties that shows up. So I know that's my state code. And then the second three digit code, 121, is going to be my county code.

So now we have all the pieces we need. I'm going to jump back. And we have the &for county 121 and &in state is 36. So the nice thing here is that you don't have to remember the little syntax components, the codes. If you follow an example, you're going to be able to always have access to what you want, and then you can customize from there.

So much like getting the full table, that group call, you can get full geographies. So what if you wanted all counties within a state? You can use that wildcard in your calls like we just did. For some geographies, as we just did to get our geocodes, you can do the wild cards for both components. It's really trial and error.

So let's take this call. I'm going to copy it from my document, and I'm going to run it in the browser. So I copy paste it, and now I'm going to run, and this is what we have. So we are getting the number the estimate of those in Wyoming County in New York who live below poverty with the corresponding margin of error. And we see we have the name here, we have the estimate, the margin of error, and then the state and the county codes.

So I want to just show you back on this slide that your output might look different than what you see here. Sometimes it's the browser you're using or the settings. But it's okay, because when you download it, it's all the same.

So jumping back over to the browser, if all you needed was the estimate, you can stop here, but you can also download it. And what you're going to do is you're going to right click. You're going to click Save As. You're going to name your file. And this is important, you're going to type in the file name .csv. The last step for the Save As type you're going to select All Files. So you're going to click Save. And it's going to download that CSV. And I'll open it up just to show you what we have.

So like I mentioned briefly, or maybe not briefly, I think briefly, name does cause a shift, especially when you have a geography within a geography. So I had a county within a state. So here it shifted my variables, and all I'm going to do is highlight these. I'm going to cut and paste to move them over. Um, so that is just what you're going to look like, what it's going to look like when you download your file.

So hopefully that was not too overwhelming. Um, and that was just a little bit of a breakdown of what the API is. So really quick, I now want to share some resources as you go on your own. But don't worry, I do have contact information so you can always be connected with our team if you get stuck.

So when you're on your own, start with checking the example calls to get yourself started. I sound like a broken record when I say that. That's what we did today. So I want to emphasize how useful they can be and how much time you can save. You can always edit them to fit your needs, but having the base like we walked through can be really helpful.

Unfortunately, some variable names change with every data release, so variables are added and subtracted from tables, so it's important that you check the variable names if you're looking at data year after year, to make sure you are extracting the same data variable. It's super easy when you use that variable list, so I always just open that HTML as soon as I get started, as you saw in the walkthrough. And then the other one is that Examples page. So these are the two that I use when I'm customizing the components of my call.

And one thing I want to mention is keys. Um, some of you may be wondering what or why, and a key is essentially just that: a way to open the door to more calls. Without a key, you are maxed at 500 calls a day, and if it's just you and your organization running calls here, there, a key isn't necessary. But if you are creating a dashboard that's going to get a lot of traffic, you might consider a key. It's completely free, and it takes mere moments.

Um, and I will mention that if you're going to use the R package tidycensus, you need a key. And Kevin's probably going to repeat that as well. Can't do it without a key.

Um, this is just a start regarding the resources. Again, this PDF is going to be able to be clickable if you can't get access, um, in the chat to the links. So there's a lot on here. And if you're lost I can always connect you. The last two in this webinar list, um, are going to be a good run-through of an example similar to what we did today with a little bit more detail. And I also included some resources for using open-source data and programs, which is really helpful if you're using the API.

One really unique and valuable tool we have to offer is the Slack channel. There are Census staff that engage on their every day to help with data user questions, especially if you're accessing data through different ways such as R or Python. And finally, as I mentioned, tidycensus, it's a great R package to use with the Census API. It is not maintained by us, but it has great resources to guide you.

Um, and I finally want to mention a few final things before turning it over to Kevin to wow us with his expertise with tidycensus. There is a team at Census that has live workshops to go over that MDAT tool and the Census API. I highly recommend you sign up if you're curious to learn more about either. These are great for both beginners and advanced users. Please consider joining the ACS Data Users Group that Lillian highlighted at the beginning of this workshop if you aren't members already.

And I know these were very quick demonstrations of the PUMS and API, but you can email our team at acso.users.support@census.gov if you have any questions in the future. Thank you so much. And Kevin, the floor is yours.

Kevin Kane, Southern California Association of Governments: Well, goodness, Mary, thanks for such a thorough and comprehensive, uh, you know, overview of both PUMS and, uh, Census API calls. Hopefully I can build on it. Um, doing these is kind of your job, for the most part, I just kind of, uh, do this as somewhat of a service to a degree.

I'm Kevin Kane. I'm the Program Manager for demographics and growth visioning here at the Southern California Association of Governments. Uh, why do I do this, uh, type of, this type of a webinar? Just, you know, um, I find it extremely useful to kind of have effective workflows, certainly in my field, which is regional planning and demographics. But, uh, I also teach this material to a course at the University of Southern California.

So, um, you know, bottom line, uh, I find, Mary's API call workflow, uh, to be really useful, but you are a little bit limited in terms of the replicability of it, um, by putting calls into a URL. And, uh, she gives me a hard time every time I follow her after a webinar, um, because of what I've titled this, uh, "R tidycensus: Your graceful exit from data.census.gov." And what I'll share with you here is basically the workflow that I kind of developed once data.census.gov, um, started a few years back in order to just kind of help, uh, you know, be a little bit more replicable.

Uh, Southern California Association of Governments has 191 cities under its purview across six counties in Southern California. So we're working with a lot of county-, place-, uh, and tract-level data longitudinally, uh, and kind of that's buried within either PUMS or other detailed tables. I'm sure that's the workflow for a lot of folks here.

So, um, I'll be very brief in terms of, uh, slides here, but, uh, really, what I want to mostly show to you is a demonstration. Um, because frankly, it's not possible in 20 or so minutes to actually get into R or RStudio or a coding environment. But basically what I'm going to pick up where Mary left off, uh, and wrap that within an R, or a code-based workflow.

So R is an open-source, uh, programming language. RStudio is a freeware wrapper of it that just makes it a little bit easier to use. Um, I've included here some very easy installation instructions, uh, for you, uh, like teaching in this because, uh, it's not a commercial product. You can take it to wherever you work, uh, and not have to worry about a license.

The second thing that I'll say is I've posted a lot of training materials here on this GitHub, uh, website here. I'm not sure who, uh, you know, the level of folks are GitHub users or not. I frankly just use this for file transfer. I am going to have to confess, I'm more of an intermediate-level user of this and frankly of some of the R packages. But like all of us, you know, hey, we're, we're doing this to do our jobs better.

Um, so what I've done here is included a package which I call the kind of a half-day R introduction. There's also a video where I did the full webinar for this, uh, if you like the workflow. Um, I would say it probably would take you about a half a day, roughly, to get through it and to actually learn R to a point where you can use the Census API usefully. Um, if you hit this code here, you can download a ZIP file containing all of this. The key file is one that has a dot R at the end of it. And that's what we're going to be kind of going through mostly today.

Um, switching back here to kind of all the information you'll need. Um, Mary already gave you a lot of the Census API information, so I won't repeat that. Um, uh, there's a full recording, uh, of, of the

webinar that takes you through how to actually get up and running in our studio so that you can get to the point where we'll start here today. Um, and also the details on Kyle Walker is amazing, tidycensus package, uh, which, although not maintained by the Census Bureau, uh, clearly is good enough to make a make a guest appearance in a Census Bureau closing slide. So, uh, certainly has kind of revolutionized how I interact with American Community Survey material.

So, um, how to get up and running here. Basically, uh, I'm going to open up this particular dot R file for you in our studio. If you've gone to the GitHub page that I shared with you before, uh, and I'm sure perhaps, uh, if you do want to follow along, maybe I could task Lillian, who has this slide deck to toss it into the chat for folks. Um, but if you're, uh, I'll just go through a couple of ways to, uh, to kind of access and use code here.

But, um, at the, at the bottom bullet here, uh, is what's in this, Rbootcamp file. I basically have 10 modules here. Module sections 1 through 6 are just basic data usage skills and visualization skills using R. I'm not going to go over those today. I'm going to skip them and start with section 7, which is how to use the Census API. Um, and then I'm going to provide you with section 8, which is basically a replicable code block for doing those API calls. Uh, once you're, uh, kind of up and running in R, you can use that to basically declare whatever variables you want, geographies, etc., um, and get them in, in a nice tabular format, in Excel format, even a shapefile format, if you like to do that.

Um, and, uh, new since last time we've done this, I've added a little bit of a code block to get longitudinal ACS data if you want the full series from 2005 or 2009, uh, when when ACS one and five years started respectively until now on the same thing. Uh, and then a new little section here at the end on, um, doing a tract-level map of something in your census place or in your city, uh, as, uh, Lillian shared in one of her earlier slides. I'm going to nab it here, um, you know, a lot of kind of how you interact with this, the API is, as Mary also said, uh, it follows the Census Bureau's geographic hierarchy. Um, you know, and there's, there's a difference whether you're on kind of this main vertical or if you're off the main vertical.

Um, you know what I tend to focus on, uh, are counties or, you know, as kind of a reflection of the overall trend or census tracts to kind of be reflective of neighborhood-type dynamics. ACS oftentimes does go down to block group as well, but you tend to get those high margin of errors, which, uh, you know, well, I'll leave it to you to decide the level of importance of the margin of error for your for your, uh, for your workflow.

But one of the challenges is that, um, cognitively, uh, and electorally and everything like that, places are pretty important census places, uh, which are basically cities, towns, CDPs, etc., are really, uh, you know, how people interact with information. So if you're looking to get an understanding of how a phenomenon, uh, is dispersed across the neighborhoods of a city, you really need this tract-to-place relationship. So I'll go into that a little bit, um, as I do the demo.

Um, apologies. I'm not really able to see the chat right now, but, uh, please, please holler if any issues. And thanks, Lillian, uh, for putting those, uh, those links up there.

So I'm going to, uh, go over now to RStudio, where I've just opened up Rbootcamp, uh, 2024.R. So really basic two ways, two main ways to enter code. On the righthand side here I've got a script file,

which is, um, I prepared this, this one for you here. It's about 500 lines or so and goes through those 10 modules. You can update it, change it, change things, um, and using this nice hashtag here could kind of comment something out. So, for example, line 22 here, um, the command is `Print`; I'm gonna print something, and then I made myself a little note behind the hashtag here.

On the left side is actually where you're executing code. It's got this little triangle called a chevron and a blinking cursor. So if I want to use the `Print` command to say "hello world," which is sometimes what folks do when they start a new programming language, it's going to return to me a line that says "hello world" back, because that's what I asked it to do. Um, certainly when we get a little bit more sophisticated with our calls and things that we want to put into the console here, uh, typing it is not going to be efficient. So that's why we have the script file up on the righthand side here.

So long as your cursor is on a line or has highlighted a portion of code, there are a lot of easier ways to run that code. The first one is to go up here to the top right and hit run. It's going to do the same thing. Or if your cursor is just on it and you hit `Ctrl+R` if on a PC, `Command-R` on a Mac, or in some instances it's `Ctrl+Enter`. I'm not sure exactly why people's computers all have slightly different setups. That's going to be the other way that you can run this line of code.

The second thing that I'll mention about kind of RStudio in general, um, in terms of this workflow, is to just be really careful what you're working directory is. What that means is a file path on your computer somewhere where you're saving data, where you're saving images, where you're saving your output, or sometimes reading in data as well.

Um, there are a few ways to do this. Um, you can, uh, if I type "`getwd`," it's going to get my working directory. Goodness. The default is, uh, what appears to be somewhat something of a My Documents on a C drive. Um, I can go up here to Session, Set Working Directory, and choose, uh, where I want to pull information from. Or I can declare it in the code here. I've already written it down here is "`setwd`." So if I "`setwd`," um, something I'd like to do kind of early in the workflow, uh, I'm going to be working with this folder. Um, and you can see it's in Dropbox, Rbootcamp as, as the folder.

So, um, that's just the absolute basics again. Um, if you want more information, you know, certainly I would suggest downloading, uh, the package from GitHub, including this dot R file following along yourself or following it along in the video link there. And right now I'm going to scroll down to the fun step to actually the, uh, using Census API here in R, which is, uh, what I have as section 7 here.

So, um, the way that, uh, R is, is useful is that it kind of has a lot of base functionality kind of built into it. And then it's very customizable. Folks have built, um, tons of different packages in it. And the one that's really helpful is called `tidycensus`. I'm also going to be using a few other packages here to be able to work with spatial data and to do some other data manipulation.

Um, when you install it, you have to do two things to use a package in R, first you have to install it, and you just have to do that once. But then every time you open R or RStudio, you do have to kind of invoke the package or activate the package. So you install it with this line here, `Install Packages`. And I'm not going to run that because it's installed already on my machine. But I am going to highlight all of these and activate these four packages here by running this line of code. So this is

basically telling our studio, hey, add this new functionality to this instance of the program that you're working on.

Mary already mentioned getting a Census API key, which you will need. Uh, it takes, she said, mere moments to sign up. I think it takes probably like 2.5 seconds perhaps. Uh, and that, that is an alphanumeric code that's a little bit ugly here, but, um, it allows you to actually use this because you are going to be iterating and pulling a lot of things. Um, it's nice not to overwhelm the, uh, the, you know, our, our federal government's, uh, servers, uh, the Census Bureau.

So they're, uh, the first thing that you'll have to do is to enter your Census API key here. And tidycensus has a command called, you know, what do you know, Census API key. So you put it in here like this and hit Run. Here's my Census API key and boom, you're done. Um, it gives you a new flashing chevron. Uh, so that means it's taking the line of code, uh, effectively.

So, um, really we're just working with a couple of key commands here. Um, as Mary had mentioned earlier, there are a lot of things available through the Census API, the Economic Census, the decennial, various other programs that the Bureau has, and ACS being the key one.

In tidycensus, you've got decennial and you've got ACS. So "get_decennial" is the command here for how to how to get something from the decennial census. And this gets decennial census command takes a few different arguments. And you can see what I've set up here in line 405 is, well let's see, I want state-level geography. So I want state-level data. I want this variable. We'll get to how you search for variables in a little bit. You know a little bit already.

Um, I want the census summary file, and I want from the year 2000. So I'm going to run this "get_decennial" command. And then what this equal sign does is it puts it in an object or thing or a, you know, something that you can call back called medrent00. I've just called it medrent00. I could call it whatever I want. So I'm going to run this line here, and what it's doing there, uh, for that a quarter second is it's actually getting the data. And now if I just type that run, oh, um, it's going to show me the median rent across all the 50 states.

Uh, I can make it a little bit easier by using the view command and view medrent00. What that will do is pop it up into something that looks a little bit more like Excel or tabular data and see that, um, goodness, in Alabama in 2000, rent is probably a heck of a lot less than it is today. Um, quite a bit higher in Alaska. So, you know, this passes the smell test. Always a good check. Uh, when you're, when you're doing a new data extraction process.

Um, so that's useful. Um, you know, you can certainly there's, there's commands, right dot CSV commands to save this in Excel. You know, if you really want to, you can just grab and copy or what have you, uh, from here. But R has a lot of really nice visualization capabilities, so it's nice to be able to take advantage of them.

I've left you with a few examples in this code here that you can, you know, certainly, uh, you know, modify the name of the, the variable, the data set you've extracted or the variable or change some of the other parameters. But if I run this line here, it's going to make a nice little bar plot, um, of states by rent. And you can see here. Oh, Hawaii is quite, by quite a bit the highest. And this is alphabetized, um, well, not quite alphabetized by FIPS code, but, you know, thereabouts.

I'm going to close this here, and, and I've made a slightly fancier bar plot here with some bells and whistles by sorting the data, adding some color, adding a label, adding some guidelines. And I can highlight all of this and hit Run or Ctrl+R or what have you. And it gives me a really nice little bar plot here of state median rents in the year 2000. Again, seeing how it varies from a high of Hawaii to a low of North Dakota. Um, did not expect that to be lower than in Puerto Rico even, but goodness.

So, uh, here. So that's, that's just the way to kind of get a little bit of a visualization. And I haven't uploaded any data into my program, which you usually have to do. Um, as long as you have the internet and a Census API key and tidycensus, uh, as a package installed, uh, you're able to just extract it in one clean flow.

Now, in order to find good variables to use, uh, Mary already gave a little bit of a tutorial to that, but, um, you can do that within tidycensus if you want to. So, um, load variables is a, is a command here. And I've just asked it to look at 2022, five-year ACS, um, and put it into an object that I'll call "acsvars" and, um, oh goodness, I have 28,152 entries for, for, uh, you know, explicit ACS variables that are coming in through what I imagine, uh, Mary can correct me if I'm wrong, what I imagine are the detailed tables rather than the summary tables.

Um, in any case, uh, this is a little bit cumbersome, you know, certainly. Um, and Ctrl+F is one of your friends. You can write this to a CSV here as comma-separated values file and, and open it up if you want to. But in the GitHub site I've included, um, my little cheat sheet. Um, if it's useful to you, happy to share. But these are my top one, top most commonly used 125 ACS variables with their code and a somewhat intuitive abbreviation, um, that I've, that I've, uh, renamed it, "totpop," for example, or median age. Um, this includes just some of the, the age structure, basics, race, race, ethnicity, commuting, educational attainment, income, and housing. Just to give kind of a smattering. Um, so if you want to start there, um, that's, that's not a bad way, at least, at least in my view.

So, um. Right. Uh, so, so now that we've found some good census variables to use, and we'll scroll down just a little bit here and try to assemble, um, some tract-level variables for a county. Um, now this is the kind of the main command here. It's get underscore ACS and you pass it a lot of information. I want tract-level data. I want the state of California, Orange County, and this variable here, 25035, which is the median age of the housing stock in each tract. You'll notice that I've also added this argument called geometry equals true. This will also extract the data as spatial data so that you can visualize it right here in R. Or you can export it as a shapefile if you're a GIS user.

So, um, it just takes maybe two or three seconds or so to get all the tracks, uh, in Orange County, California. Um, if I look at what this is, "head" just gives me the first five rows of any given data set. Uh, let's see, I've got a GEOID. This looks like my FIPS code. I've got estimate, which is actually the value I'm looking for. So this tracks median housing home year, built year was 1971, 1959. All right. So this passes the smell test. Certainly these are reasonable values especially in the western United States.

Um, so I can do just a little bit of manipulation, renaming it old age. Um, you know, getting rid of the old one. And if I want to see how many rows there are, take a quick look and see that there are 614 tracts in Orange County, California. So now I have a good understanding of, of the rows and columns, which at the end of the day, that's all data are.

What if you need more than one variable? Um, tidycensus will extract it for you, but it's a little bit trickier because it does it long. Um, to show you what I mean, I'm going to, um, make a list of three variables: population; housing stock age, which we already did; uh, and median household income. And I can extract those in one single call by, uh, declaring this list that I made as the variables that I want. So I'm going to call this one TR underscore plus. Again, it just took a second.

And if I want to see how many rows are in underscore plus, oh goodness, it's 1,842. Well I know there are 614 tracts. So, um, I can take a look at it and see that, hmm, this is not stacked in a terribly intuitive way. I've got three records for each tract, and each one's for a different variable. Kind of a pain in the butt when you want to do math, compare it, put things as a rate, uh, put them on a map, uh, or anything like that. So, um, you know, if you really want to use it, you can use something called the match command, which is described in the earlier sections that I totally glossed over. Uh, and, and do a subset of this lengthy file and then and then bind it to your original text file. So now I have 614 entries here and eight total columns. I've got one for home age, total population, median income. My apologies to the Bureau for omitting the margins of error here, but you can grab those as well, especially for tracts. Mea culpa.

Um, some of the other nice features within R is that you can actually just plot this as a map using one of the, using what's called the SF package. So if I hit line 450 here, sorry, um, uh, goodness, I can get a nice little map already of the tracks in Orange County. And again, uh, let's see, we've got 1940s, 1950s here, kind of in the North Side. This is downtown Santa Ana, the older neighborhoods of city of Anaheim. Those look a little bit older, uh, then used to get to the south, to Irvine, to Laguna, Niguel, Coto de Caza. These are the newish developments up in the hills. You can see that reflected in the more curvilinear boundaries but also in the, the newer home ages there.

So, um, neat little trick there. And if you are a GIS user, you can use this "st_write" command here. Um, whoops. To write an entire shapefile. Um, now this is, uh. Let's see. What did I call it? I called it orange underscore merge. So if I go back to here now, I have four files here I've seen if you're a GIS user, you know, you've got somewhere between three and eight files typically together in a shapefile format. But now I can use this in GIS. I have orange underscore merge. All right.

So, uh, racing along right here. Um, hope folks are getting a little bit out of this at least. But what I've built here in section 8, um, is a way to group a lot of variables together. Uh, like I said, it's a little bit clunky to extract variables one by one because they're stacked long. So you want to make a loop and, uh, loops are, you know, a little bit more advanced coding skill. Uh, but I've built this to hopefully make it so that you can just enter your parameters here, um, and, uh, and then run this big block of code in section 8 and then get a good data set.

So I'm going to ask the audience here for somebody to put in the chat a state and a county, like not a tiny county, at least the medium-sized one. You know, five more seconds before I use Tampa. Okay. Sacramento. Let's do, let's do Sacramento okay. Thank you.

So Sacramento County, California, my state equals CA. My county equals Sacramento. Let's see. Let's I'll run the first chunk of this. And the first chunk. The way I've set this up is it's just grabbing total population B01001 underscore 001. And then what I'm doing is taking this whole big list of 125 variables that I've shared with you earlier in this spreadsheet here. Uh, and then I'm renaming them to something that's a little bit intuitive. Um, not perfect, of course, but, uh, you know, if you follow a

logic, uh, you know, commute, walk, uh, median household income, you know, female aged 5 to 9, etc. Uh, you know, should be logical. Select all of this, even do a little bit of math on the end of it. And it's really only going to take probably a few seconds to extract this for, um, 125 different variables for, um, the tracts in Sacramento County.

All right. There we go. I can view this. I just called it D to keep it a little bit easier. Uh, and now you can see all the tracts, uh, total households, median age of 29. Goodness, that's quite a bit under median. So that must be a young area. Um, race ethnicity, variables, etc. Um, how I put them up, I've got 135 total columns in this, uh, in this data frame right now. I can write it to a CSV right here. Whoops. I called it Hillsboro, Florida. Sacramento. Don't get confused now. My Hillsborough file is messed up, but, uh, that was, uh, that was from somebody else. So Sacramento tracts and ACS can just open it up in Excel in a comma-separated values format, um, and, um, manipulate that however you like.

So there you go. You've got, um, uh. Uh, you can also write it to a shapefile here. Um, Again, make sure you name it the right thing so you don't forget that that's Cook County, Illinois. Um, and, uh, you know, you can do some plotting. Um, here is median median home value in Sacramento County. I'm not super familiar with the urban geography of Sacramento, but I'm assuming this is a little bit more kind of an inner ring neighborhoods in the downtown core and then the fringe, you see some higher income as well. This is all in the SF package. So there are a lot of parameters that you can do here.

Uh, the nice thing is, well, by doing this workflow is that you can just do math right here. So what if I want to know if the the share of commuters who work from home. Uh, a question that we get asked all the time. Uh, so I could just do the, uh, number who work from home divided by the total population of commuters. Do a little math here and then plot that variable. So, okay, the work from home share in Sacramento County is way high out here, fairly high in somewhat of the downtown core, and a little bit mixed. Again, you can do quite a bit of a different analysis here if you'd like.

Um. And then if you want to plot the variable a little bit more neatly, um, I've got median home value pulled up here with the Jenks optimization so that it gets some nice natural breaks. You can do a reasonable looking plot just right here in R without having to open up GIS or anything else.

Two more quick tricks here before the getting in under the gun at 12:30, uh, Pacific time. that is, um, is a task we often need is to get longitudinal ACS data. Um, I find this a little bit tricky, um, because you do have to iterate quite a bit. Um, can somebody, let's see. I'm going to pull, um, Milwaukee County, Wisconsin, from, from the chat here for this example. But basically what I'm doing here, um, is I'm making a sequence of all the ACS years that are available. Um, sending, I use a lot of one-year data because I tend to work in big counties. Um, so it's a little bit tricky because it didn't exist for 2020. So you have to make sure to make a list that has that gap in there. Um, five year, that's not an issue.

But in any case, um, what I've kind of given here is a not quite as sleek of a, of a loop as, as earlier, but a mechanism to, uh, go through and enter whatever I'd like to here, Milwaukee County. So, um, this is going to take a couple of seconds, in fact, to run this because it's pulling, um, well, that's what I've come across. Oh, I don't think that. I think there we go. So now you can see as this runs here in the red text, it's 2008, 2009. It's just looping through, uh, all of the available ACS years to get

me, um, two variables here. I put them in. I kind of snuck it in here. One is, it's what I just showed you earlier, the number of people who work from home versus the total commuters. So, um, and what this can give you right here is total commuters in 2005, the number who work from home, and then a really nice time series of how work from home has evolved since the ACS started collecting data on it, uh, in 2005.

So again, you can write that, you can use it later. Um, I can plot it here, make a little plot and see. Goodness, that's what happened here during COVID. Uh, and then in the most recent year, a little bit of a dip. I can make a better line graph that I've put a few bells and whistles into. Uh, whoops, I forgot to change this to Milwaukee County, Wisconsin. I'll do that in just a second here. Um, and I can even make a, um, a comparative graph. Change that to Milwaukee, just so I don't get confused. So an example of how to do a little bit of these edits here.

And what I'm also going to do is I'm going to, I'm going to make a comparative graph here. I'm going to also extract Sangamon County, Illinois, which is Springfield, which is kind of a smallish mid-sized city. Uh, and then, um, and then run through this again. And once I run through this again, I'll be able to have a graph that compares two different places in their work from home trajectories, which is kind of interesting. And this is probably the slowest part of the Census API, at least the way I built this here.

All right. So now I can see Milwaukee County work from home. Goodness, shot up during COVID and went down, but a much smaller, um, you know, uh, metro area, uh, had not only a lower level overall but didn't see a kind of a drop in 2022 as a return to office happened. So again, just an example of some of the analysis you might be able to do with this.

I'll share one final tip and trick in the last couple of minutes that I have with you here. Um, and it's something that we just, just figured out. Um, my colleague Echo Xiang, who's also on the call, and I, um, is to do a tract-level map of something in a single city. So while I'm doing this, if somebody can, um, tell me a city and the county it's in, uh, to make a tract-level map, something that actually has not just a few tracts, something that's a little bit at least medium size.

And this is going to, um, uh, this is going to require a few new packages: terra, readr, and mapview. All right. Let's do, uh, let's do, um, Oklahoma City. Actually, it has city twice. I'm not 100% sure it's going to work. Um, how about, um, Tempe, Arizona, Maricopa County. Tempe, Arizona.

Um, I'm just going to work with median household income right now to show you this. And, um, you know, also one, one thing that I've given you here, um, in the GitHub is a file, that's a relationship file developed from geo core that I use to relate tracts to census places, because, again, it's not on that main spine of the census geographic hierarchy. Um, so, you know, it gives you the percentage, you know, tracts don't necessarily nest within cities or places. And this, this tells you, for example, uh, Autauga County, Alabama, which we always see when we're doing census work nationwide. Uh, nine, 98.42% is in this tract, and 1.5% is apparently outside of Prattville, Alabama. Um, so just, that's all to say that you can define a threshold, um, to kind of get rid of some of the superfluous stuff that's, you know, 99% outside of the city.

So I'm going to declare a place, a variable household income. I'm going to make sure that I asked for Maricopa County, Arizona. Uh, solicit this tract data here. Make sure everything works. Okay. Looks

like it all works. And then I'm going to use this neat mapview feature here, see. All right. Some, some issues here. So I'm going to go back to, uh, press the old Riverside County, California. What is this for? Apologies for the work. If you troubleshoot, I'm sure.

All right, here we go. A dynamic map of Riverside County, California, by median household income. Uh, mapview even allows you to hover and see what the household incomes are. You can do a pretty yeoman's job of exporting the image. And, uh, there you go. There is your analysis.

So anyways, check the GitHub. Um, hope this was a helpful demonstration. A little bit sloppy, albeit, but, um, uh, enjoy. And thanks for participating. I think I'll turn it back to, uh, Lillian and/or Mark for the kind of the closing.

Mark Mather: Great. Thanks so much, Kevin. Um, we are we, it's 3:20, it's 3:29 East Coast time. So I know we're almost at the end of the time for the webinar, but, um, and this was an incredible amount of information. So just as a reminder, we will be sending out a recording and the slides that have all of the relevant links. I think that, um, flew by in many of these, in many of these presentations.

Um, I think because of the time we are going to officially close the webinar, but the panelists have agreed to, I think that you all agreed to stay for a few more minutes. If anybody wants to stay behind, uh, more informally and ask them some questions, we can, um, unmute you and, um, you know, five or 10 more minutes, I think, and we can, uh, turn off the recording so we can just speak more informally. But, uh, with that, I do want to officially close the webinar. I'll stop recording. And thank you all for joining.