# **1.0. Introduction to the Census API**

The United States Census Bureau provides an official API for accessing its data. This API, while very powerful, has a learning curve. Please keep in mind that no matter your task, you’re going to need to learn this API if you want results. This section will teach you the basics of it, and will show you some of the code snippets we have made to assist your efforts.

This example API call of the decennial census shows some of the fundamental parts of the API.

[https://api.census.gov/data/2010/dec/sf1?get=P001001,NAME&for=us:\*](https://api.census.gov/data/2010/dec/sf1?get=P001001,NAME&for=us:*)

The section written in green is the base API endpoint. This part never changes, it’s the stuff afterwards that changes to get your specific results. The part highlighted in orange is the year of the dataset. Since the decennial census happens every decade, we need to make sure we get the right year for our question. The part written in yellow is the dataset. dec/sf1 means the decennial census/summary file 1. Finding the right dataset to answer your question can be challenging, but we have summarized them all in our Census Dataset Summary document.

The portion written in purple is the start of the query. If you have worked with REST APIs before, then this should be familiar to you. Otherwise, the ? mark indicates that the following text is the query, while the get= portion essentially means to “get” the values for the following variables. The portion written in red are the variables to grab. This is one of the least intuitive parts of the API, so buckle in. NAME essentially means the name of the location you’re querying. In the case of our query, we’re not looking for anywhere in particular; we’re getting the aggregated data from the entire United States. The [P001001](https://api.census.gov/data/2010/dec/sf1?get=P001001,NAME&for=us:*) variable is the weird part. This is essentially an internal code name for a specific figure, in this case it’s the total population. If you’re wondering where I found these variables and the descriptions, note that every dataset has them listed, and indeed you can go to find them using a similar URL structure to the api call.

For example, to see the list of variables for the dataset we’re calling above, just go to:

[https://api.census.gov/data/2010/dec/sf1](https://api.census.gov/data/2010/dec/sf1?get=P001001,NAME&for=us:*)/variables.html

You will notice instead of beginning an API query with the ?get= section, we instead just enter in variables.html. You can do that for any Census dataset that you know of. Again, we would recommend you take a look at the Census Dataset Summary document and the instructions there to find the dataset you want.

The last part of the API query, written in blue, also has some parts that may be familiar to those who have used a REST API before. The “&” is the start of the predicate, which is how you’ll filter things down. In this case, we’re doing [&for=us:\*](https://api.census.gov/data/2010/dec/sf1?get=P001001,NAME&for=us:*). The “for” here can mean “get the data for…”, while the “us:” part specifically means we’re getting the data from the entire U.S.. Lastly, the “\*” is the wildcard; this means it gets all of the data for a location. Since we’re getting the total population of the entire United States, we’ll only have one result.

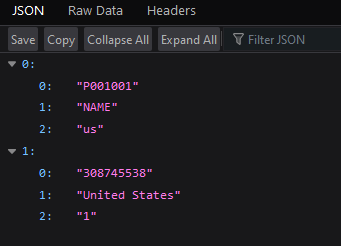


Image 1: The results from the API call above.

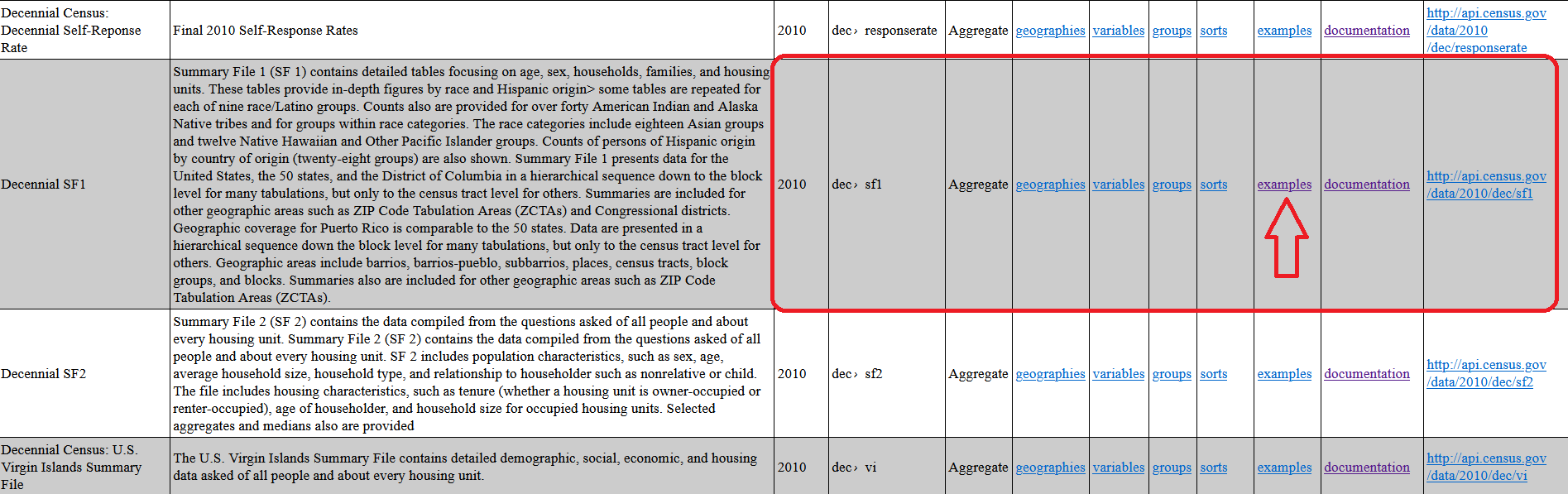
The data you see when you call the API is structured like an API call. Here, we can see the first section, which has "P001001","NAME","us" as the variables we searched. The section afterwards is the corresponding data to each variable. So, P001001, which we know is the total population, corresponds to the value 308,745,538. The NAME is the place we’re getting the data from, which is the United States (rather than any particular state or county), and the “us” part basically just clarifies that this is the first (and in this case, only) result from this query.

If this is all very overwhelming, don’t worry. You will likely not need to have a particularly fancy query to get the data you want. In fact, the U.S. Census API comes with example queries!

Just like above with the list of variables, we can look up some examples:

[https://api.census.gov/data/2010/dec/sf1](https://api.census.gov/data/2010/dec/sf1?get=P001001,NAME&for=us:*)/examples.html

You can also find the examples in the [Census API Data list](https://api.census.gov/data.html), as well as some of the variables and other valuable information. Now that you know how to customize your queries, we recommend starting off with one of the examples and customizing it to your liking.



**1.1. API Keys**

You may have noticed in the link to the query examples above that every query has this at the end:

&key=YOUR\_KEY\_GOES\_HERE

Many API services will require you to get a key if you’re using it a certain amount. In the U.S. Census’s case, they only require you to use an API key if you’re making more than fifty queries a day. You likely are not doing that, so it is unnecessary. However, if you would like to request a key, [you can do so here](https://api.census.gov/data/key_signup.html).

So, if you’re going to copy an example query, remember to take off the &key part. For example, if the example query is:

https://api.census.gov/data/2010/dec/sf1?get=P001001,NAME&for=region:\*&key=YOUR\_KEY\_GOES\_HERE

Then simply make it:

https://api.census.gov/data/2010/dec/sf1?get=P001001,NAME&for=region:\*

Try it out, it’ll run just fine without the key section!

**1.2. TIGER/Line Shapefiles**

The Census also offers Shapefiles, which contain geographical information. This information can be used in conjunction with Census data to help display census data on a map. While this may be useful in some niche cases, you likely will not need to download it, and indeed we would encourage you to seek easier methods before you employ these. Rather than use the [TIGERweb API](https://www.census.gov/data/developers/data-sets/TIGERweb-map-service.html), we recommend you simply use the [Census Geographies webpage](https://www.census.gov/geographies/mapping-files.html) to download the Shape files you might need. Again, we will reiterate, even if you are mapping data with tools like ArcGIS or Tableau, you likely won’t need these kinds of files.

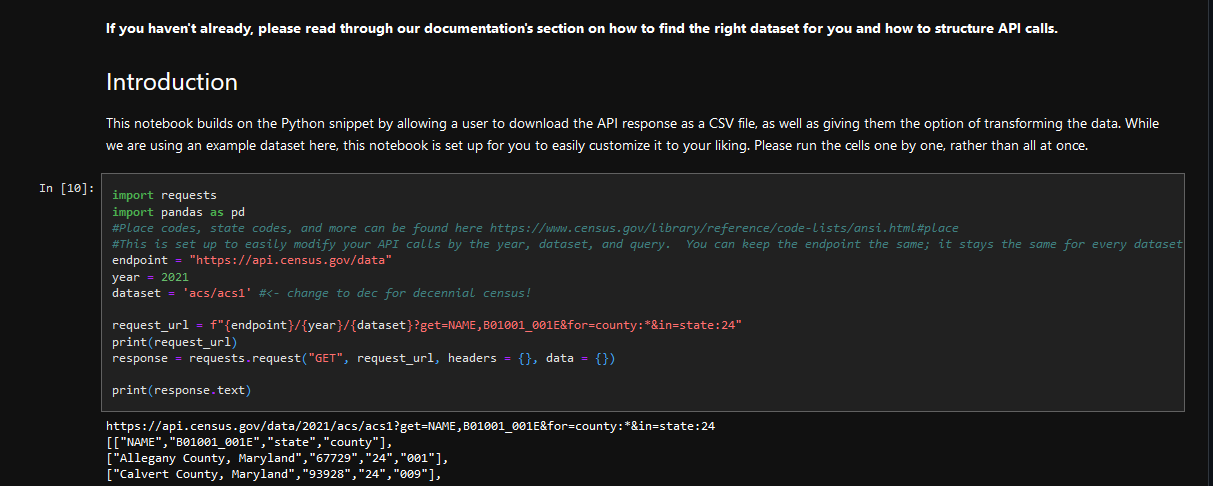
**1.3. Further Reading on the Census REST API**

* The [Census API User Guide](https://www.census.gov/content/dam/Census/data/developers/api-user-guide/api-guide.pdf) is a fantastic resource for learning about the Census API, and is where we would recommend you go first if you have any further questions about how to work the API.
* There are also [official tutorials for utilizing Census data](https://www.census.gov/data/what-is-data-census-gov/guidance-for-data-users/video-tutorials.html) via the API that you may be interested in. You can find information regarding finding, extracting, analyzing, and mapping Census data by experts in these datasets.

# **2.0. Code Snippets**

In order to help you get your project off the ground as soon as possible, we have compiled a few code snippets for you to use on our GitHub. You can download these files via GitHub Desktop, or download the repository as a ZIP file. Instructions for how to do that [can be found here](https://www.gitkraken.com/learn/git/github-download). The R and Python snippets are easy templates and starting points for your projects, and they show how you can call the API and manipulate its data.

The snippet entitled “download\_file\_python\_snippet.ipynb” is more complicated, but it is a very helpful script for certain platforms. This script lets you call the API, load it into a pandas Dataframe, and save it as a .csv file. You can also use it to manipulate the values to create new columns, and to rename old ones. This can be very useful when using tools like Tableau, and to clarify your data. While the datasets we use in these snippets are probably not the ones you need, you can easily change these snippets to suit your needs, and we have even left comments to guide you through that.

The first two cells of the download\_file\_python\_snippet. Note the commentary and instructions.

**2.1. Required Libraries**

The Python scripts and notebooks require the requests and pandas libraries to function. You will need to download these libraries onto your system. You can do that using Python’s pip feature.

* If you want to download libraries on Windows, [use this tutorial.](https://blog.openthreatresearch.com/installing_python_library_vscode)
* If you want to download libraries on Mac, [use this tutorial.](https://www.geeksforgeeks.org/how-to-install-pip-in-macos/)

Once you have pip working, simply paste in the following two commands:

pip install requests

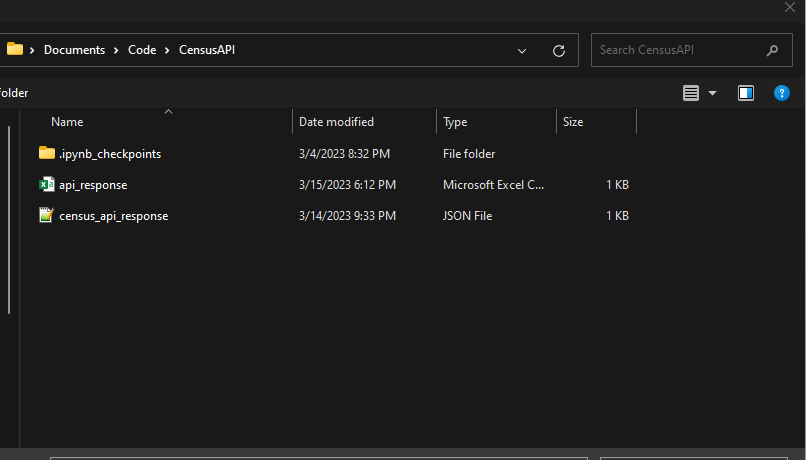
pip install pandas

For R, our snippets use the

# **3.0. Tableau**

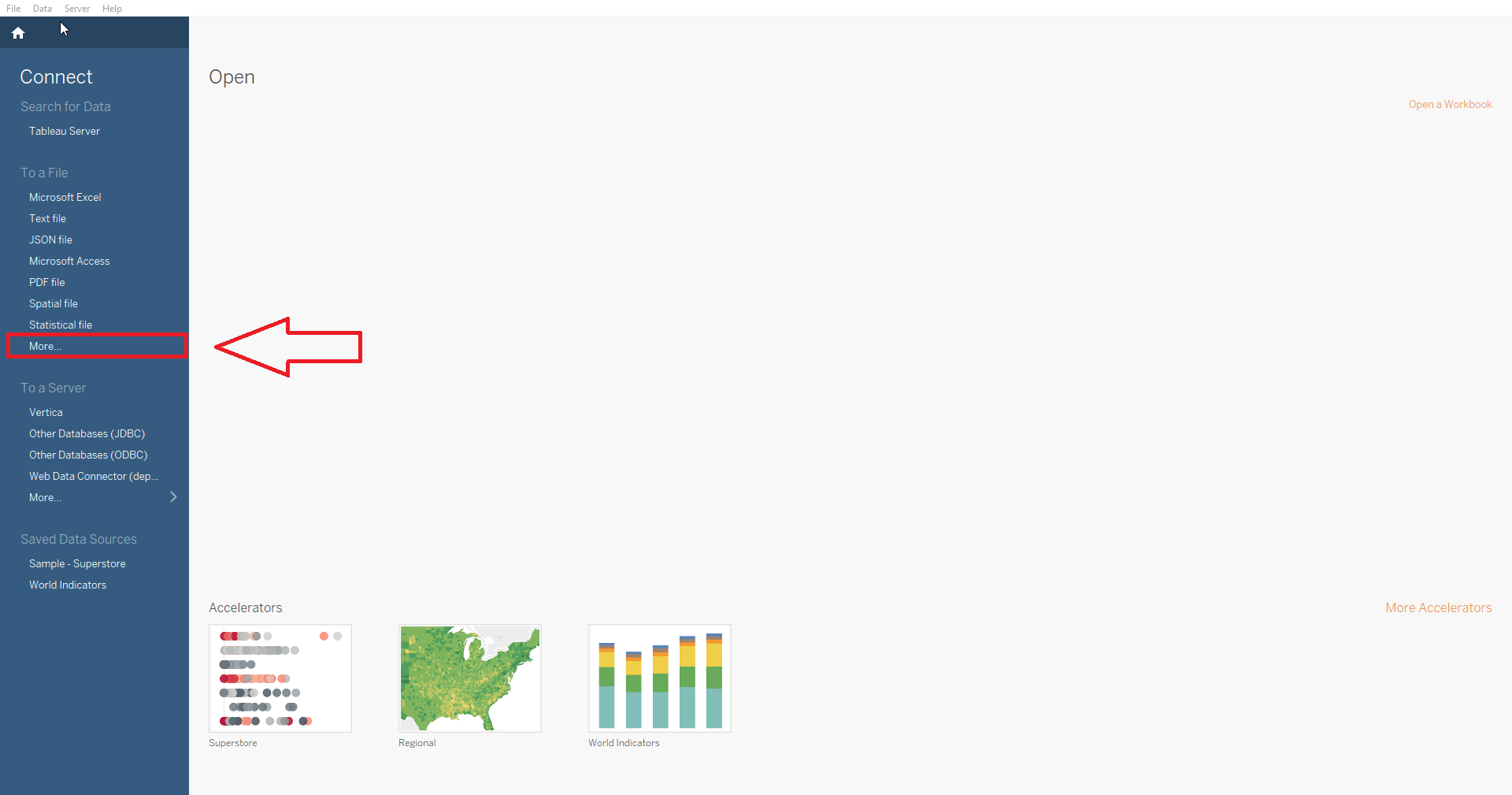
**If you plan on using Tableau, please look at the code snippets and download the download\_file\_python\_snippet.ipynb script**. The following are the necessary steps to start visualizing Census data via Tableau:

1. Download the download\_file\_python\_snippet notebook on GitHub. Read through it.
2. Modify the notebook to fit your needs, such as changing which dataset you plan to call and the query.
3. Download the file to your computer and open up Tableau.

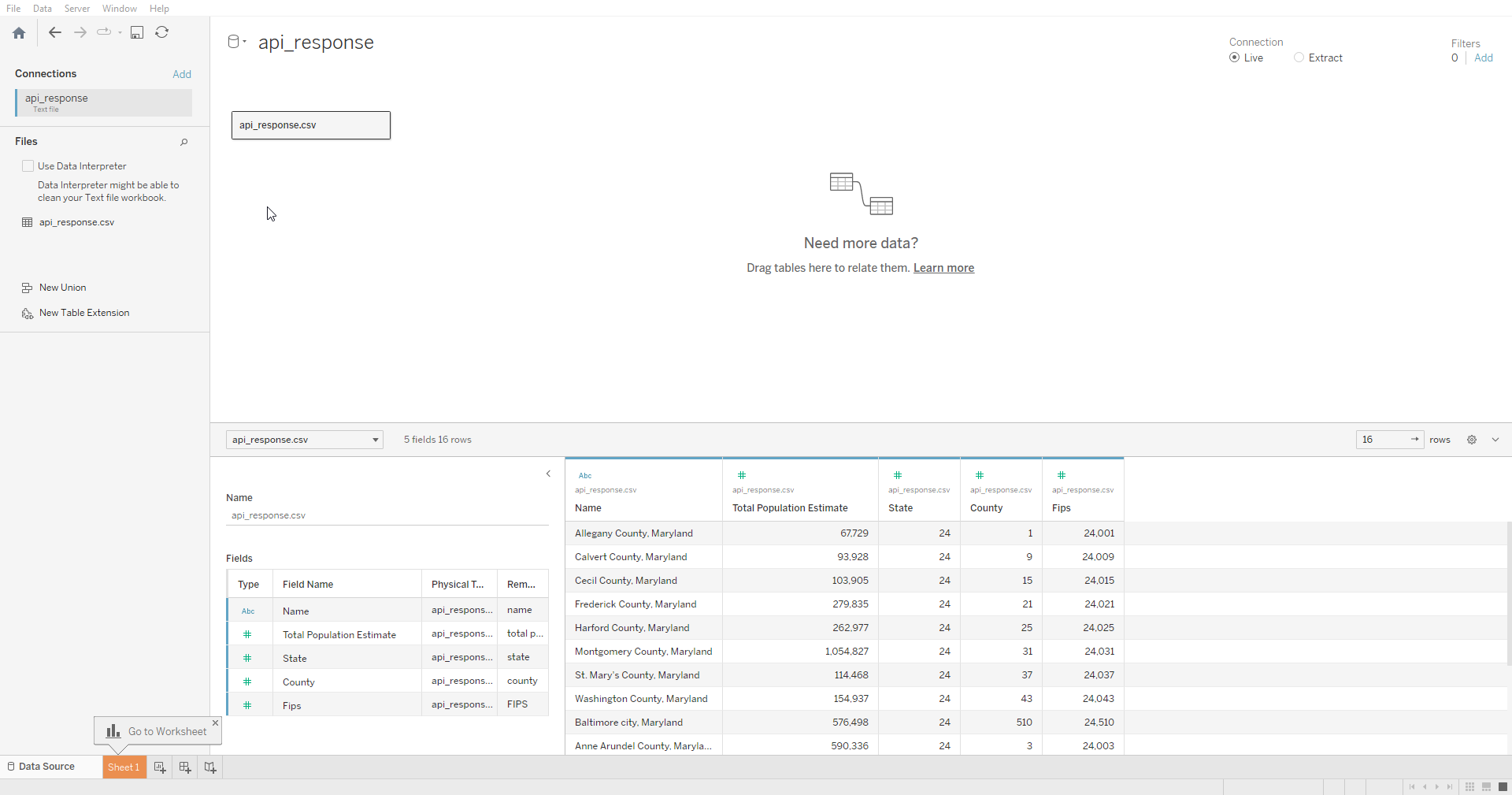


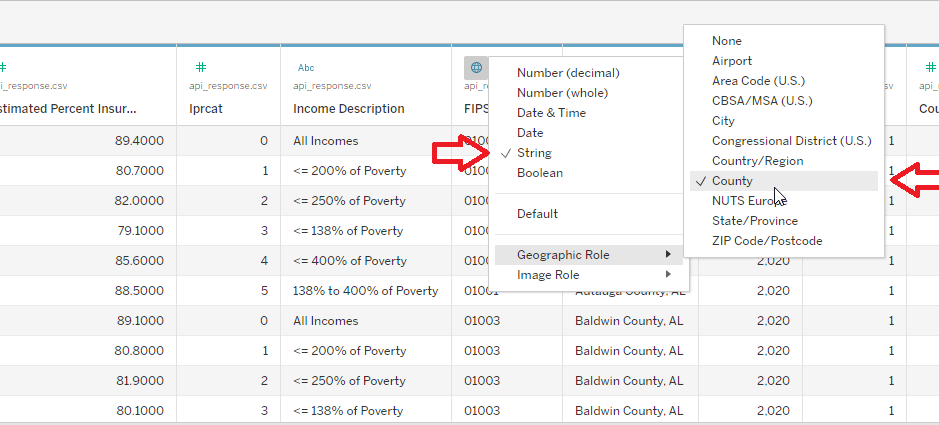
The .csv file will be generated in the same place as where you put the notebook.

1. Load the file into Tableau by clicking the “More…” button on the Connect page and navigating to the .csv.

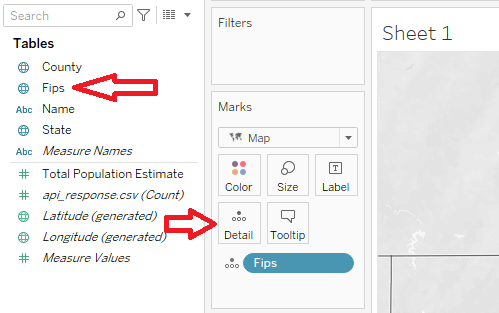


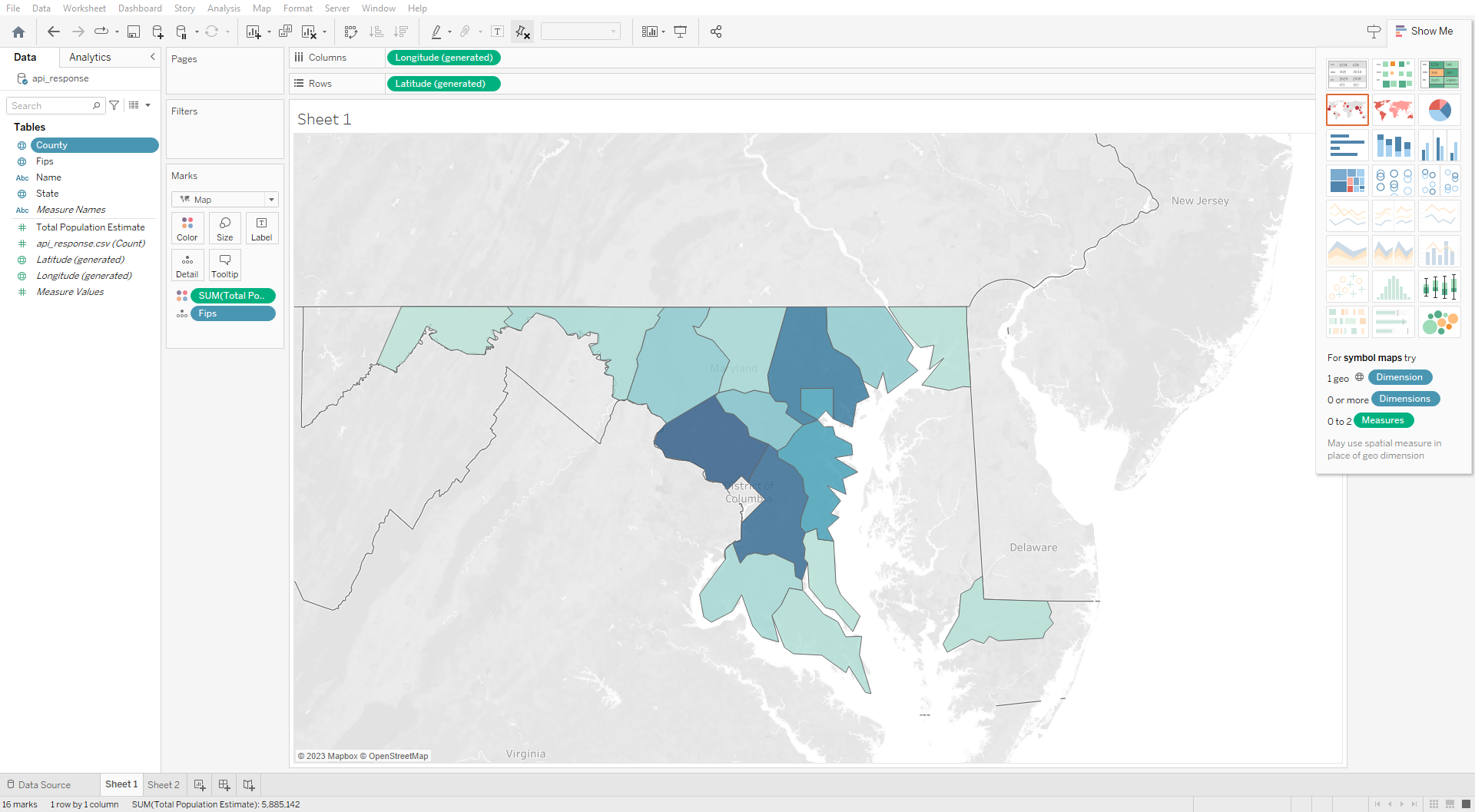
1. Look at your dataset in Tableau. Make sure the values are correct, then change the data types as needed. Remember, for geographic variables like County, they must also be a string data type.





1. Open a new worksheet/dashboard/story and start visualizing!



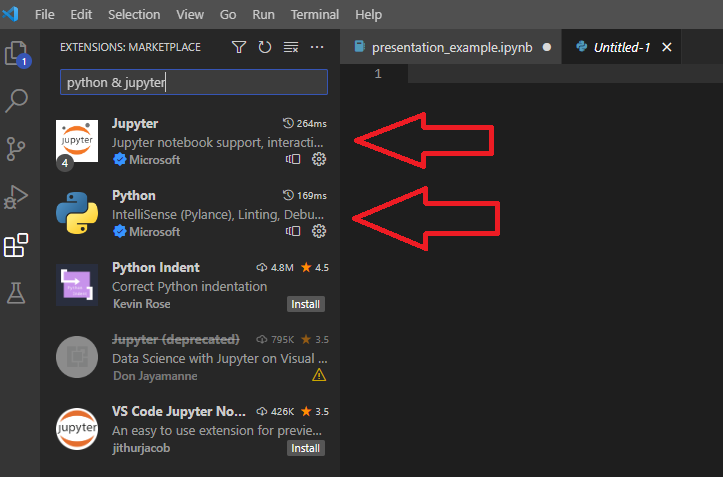


An example visualization of Census data using the FIPS codes for mapping.

# **4.0. Python**

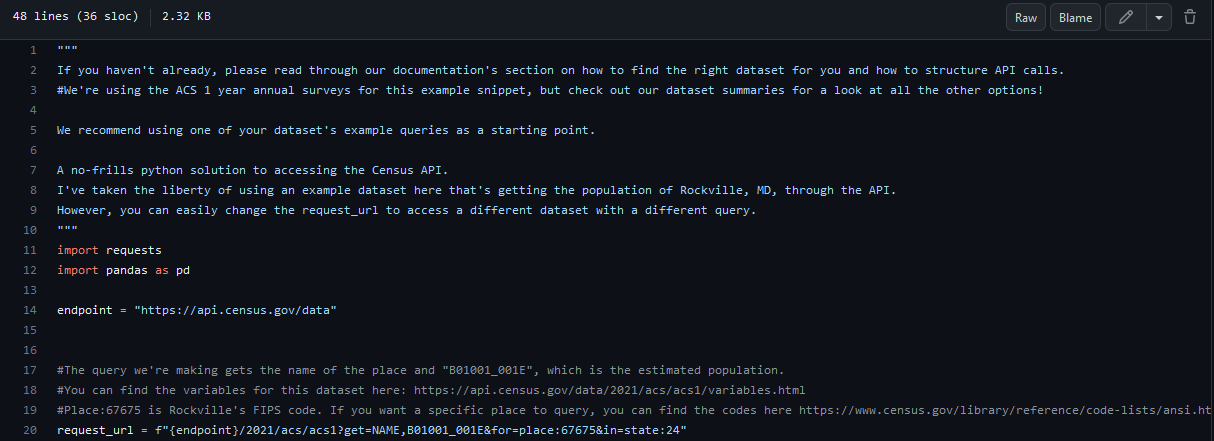
Regardless of your environment, we recommend using VSCode as your integrated development environment.

1. [Download VSCode here.](https://code.visualstudio.com/download)
2. Once you have downloaded it, go to the extensions tab and download the Python extension. You may also optionally install the Jupyter extension for Jupyter notebooks.



The extensions tab with a list of downloadable extensions.

1. Once you have downloaded the Python extension, you should work through the required libraries. See section 2.1. of this document for instructions on how to install the pandas and requests libraries.
2. The script named python\_snippet.py on the GitHub page will give you an example of how to call the API, parse the results, and transform the data using the pandas library. Like our other code snippets, we recommend you read through it, then download it and modify it to suit your needs.



A screenshot of the python\_snippet.py script.

# **5.0. ArcGIS**

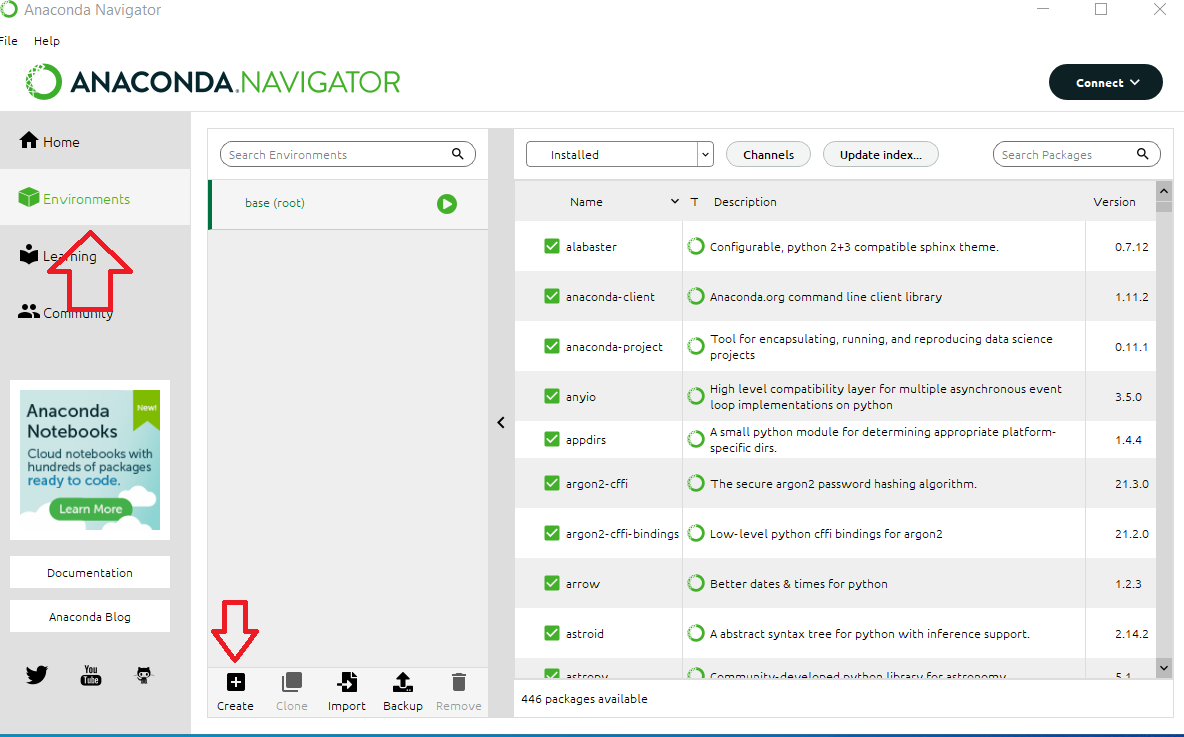
Before using ArcGIS, you should first do the following:

* Check if your data exists publicly on ArcGIS’s Living Atlas. [You can search for it here](https://livingatlas.arcgis.com/en/browse/#d=2). If it doesn’t, you should use a different platform like Tableau.
* Set up an ArcGIS Python environment. [ArcGIS recommends using Anaconda for this](https://developers.arcgis.com/python/guide/install-older-versions/#get-conda-with-anaconda-for-python-distribution). You can use Anaconda on both Windows and Mac.

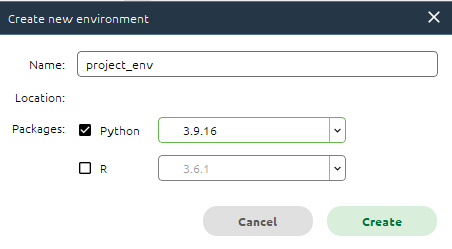
**5.1. Installation Instructions for ArcGIS with Anaconda Package Manager**

ArcGIS is most easily installed with Anaconda, which handles the dependencies and packages you need.

1. Download and run the [Anaconda installer, found here](https://www.anaconda.com/download).
2. Once it has installed, search for Anaconda Navigator on you computer and open it. Then, navigate to the “Environments” section, like so:



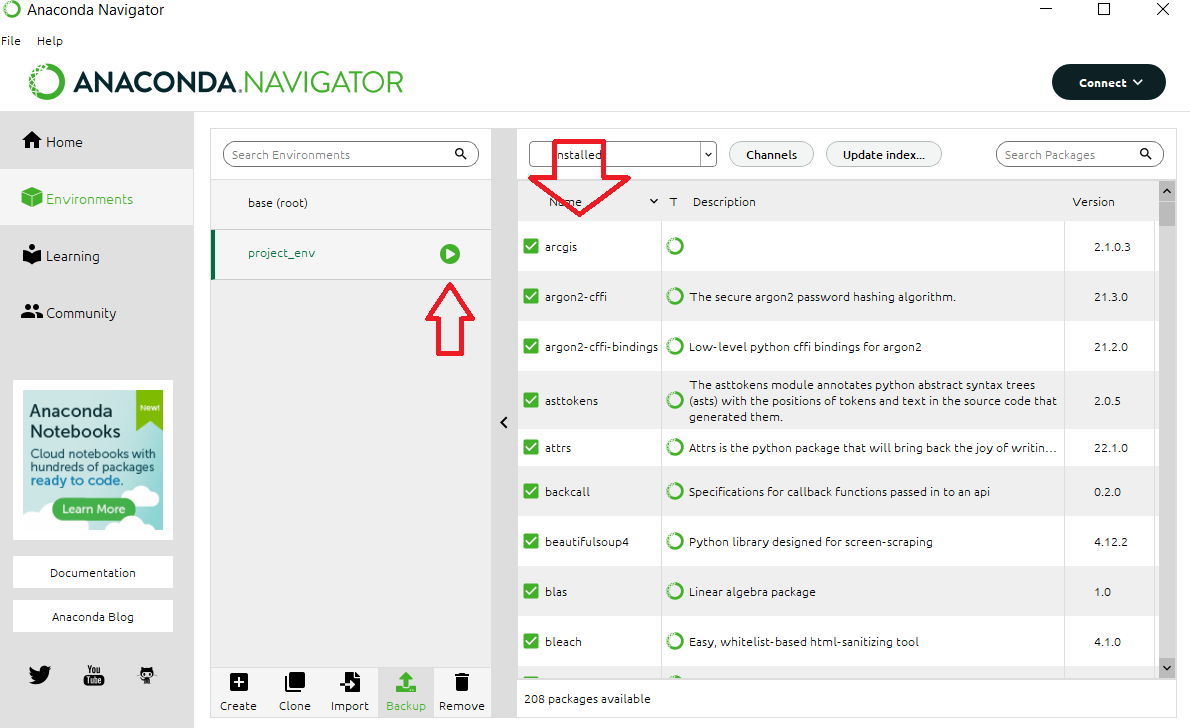
1. At the bottom, you will see a button labeled “Create.” Click that to create a new environment for your project. We recommend you use Python version 3.9 for this.



1. Once it’s been created, open up the program “Anaconda Prompt,” which was also installed. It should take you to a terminal interface.
2. From here, enter the following command to install ArcGIS to your environment, where “project\_env” is the name of the environment you made:

conda install -c esri arcgis--name project\_env

1. Once it has finished installing, you should be able to see the ArcGIS package in the list of installed packages on your environment in Anaconda Navigator. Once you have verified that, click the green play button to launch the environment as a Jupyter Notebook. We recommend choosing a browser to open it.



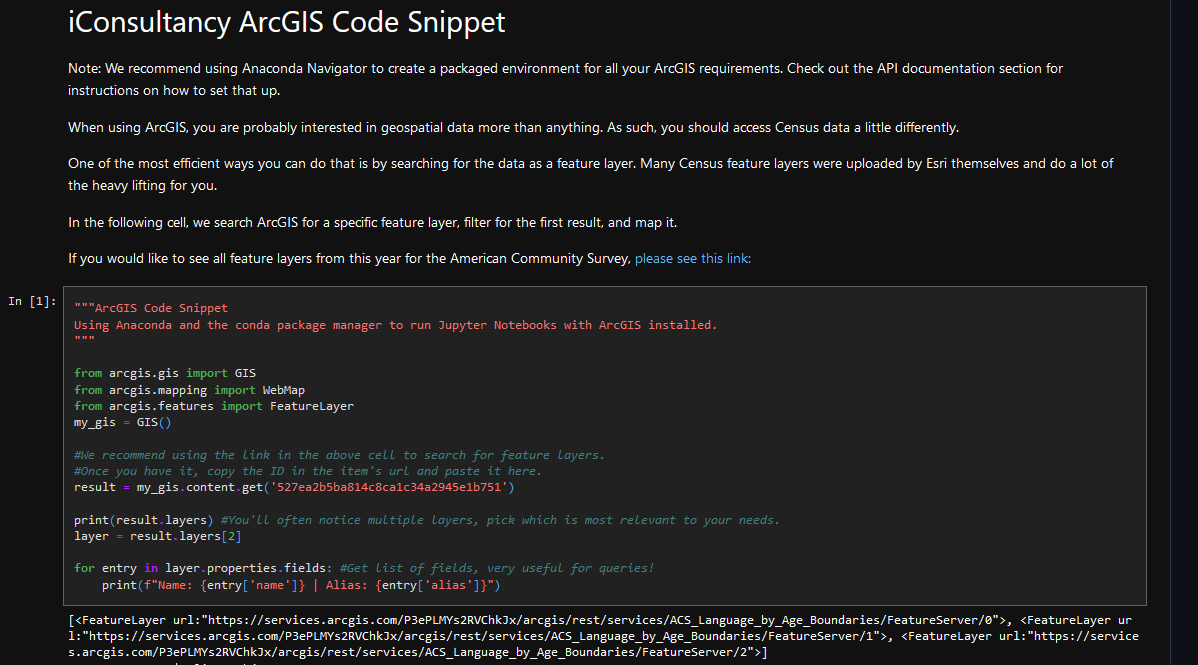
1. You should come to a screen like this:



1. Once there, simply navigate to your arcgis code (such as the arcgis\_snippet.ipynb notebook) and you’re ready to begin using ArcGIS!

**5.2. Quick-Start Code Snippet for ArcGIS with Python**

We have a notebook called **arcgis\_snippet.ipynb** in the GitHub repository that can be used as a tutorial/framework for your projects. It provides detailed instructions on how to use the ArcGIS API to query and visualize the Living Atlas Census datasets.



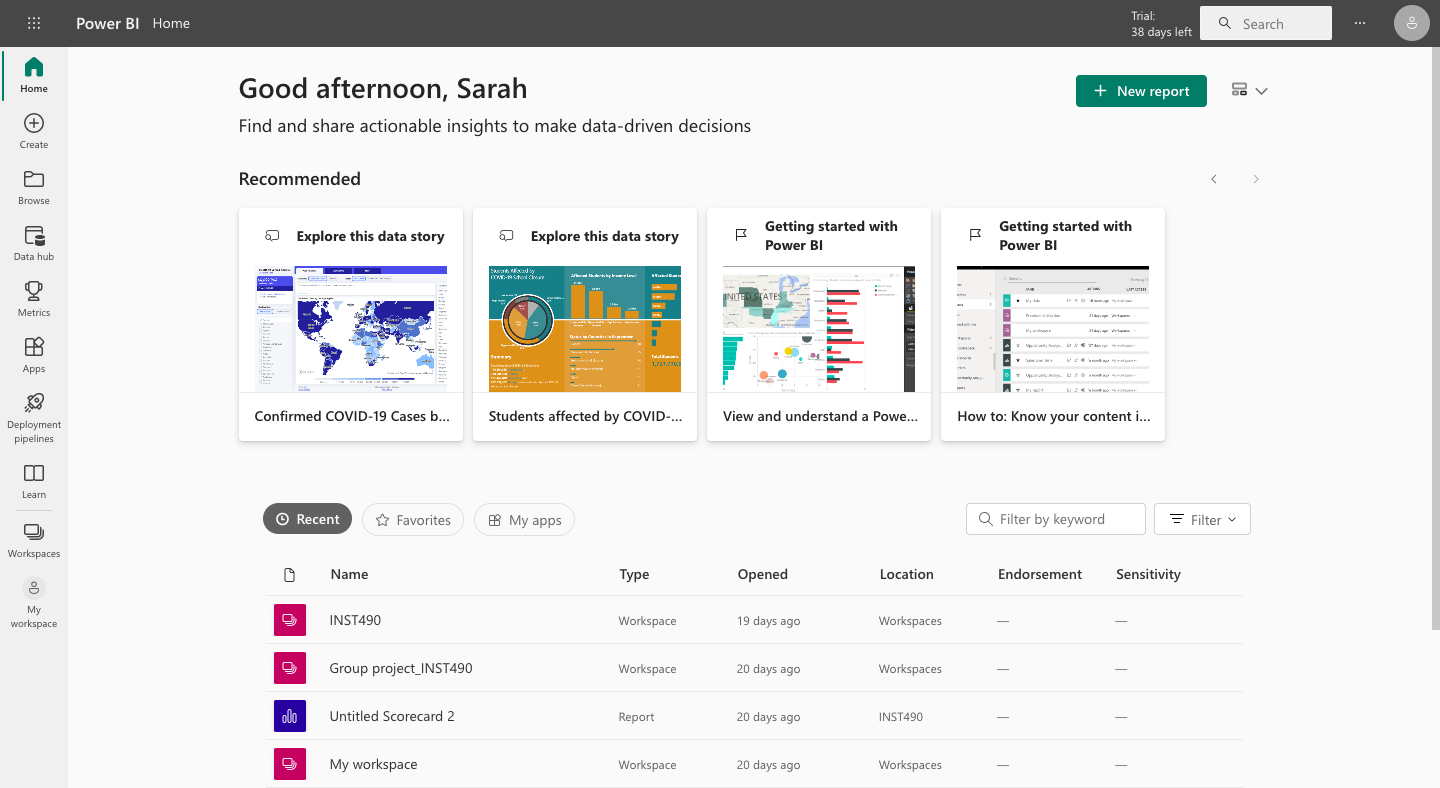
The arcgis\_snippet.ipynb Jupyter Notebook has detailed instructions on how one can search for, query, and visualize Census data uploaded to ArcGIS’s Living Atlas

# **6.0. PowerBI**

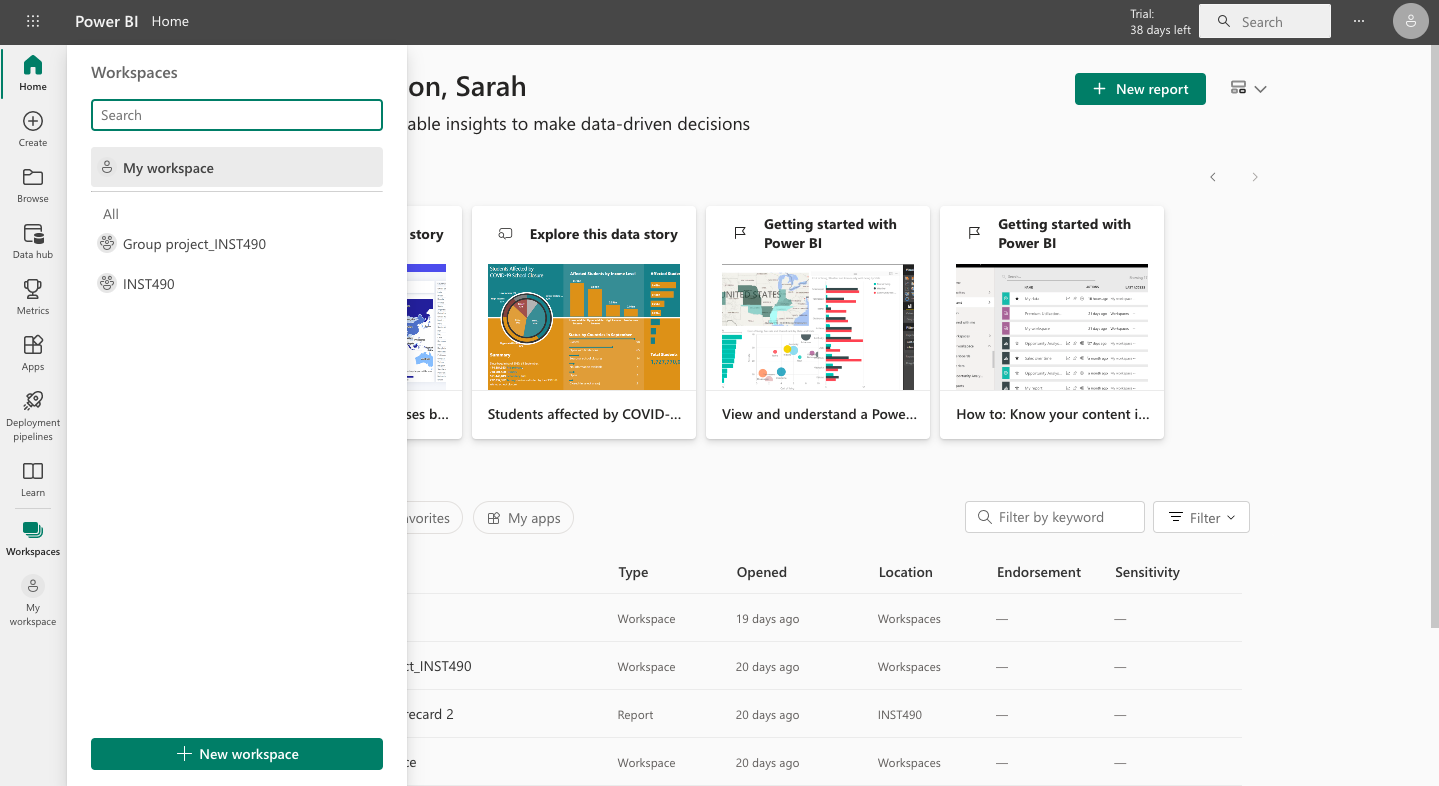
PowerBI is a great tool that contains several software services, apps and connectors that work together to turn unrelated datasets into coherent data. It also offers visuals and interactive insights. If you would like to learn more about what PowerBI offers and does, visit the following website. If you are unfamiliar with PowerBI, this site is a great tool since it gives an overview of what the software is and breaks down in detail all its features.

<https://learn.microsoft.com/en-us/power-bi/fundamentals/power-bi-overview>

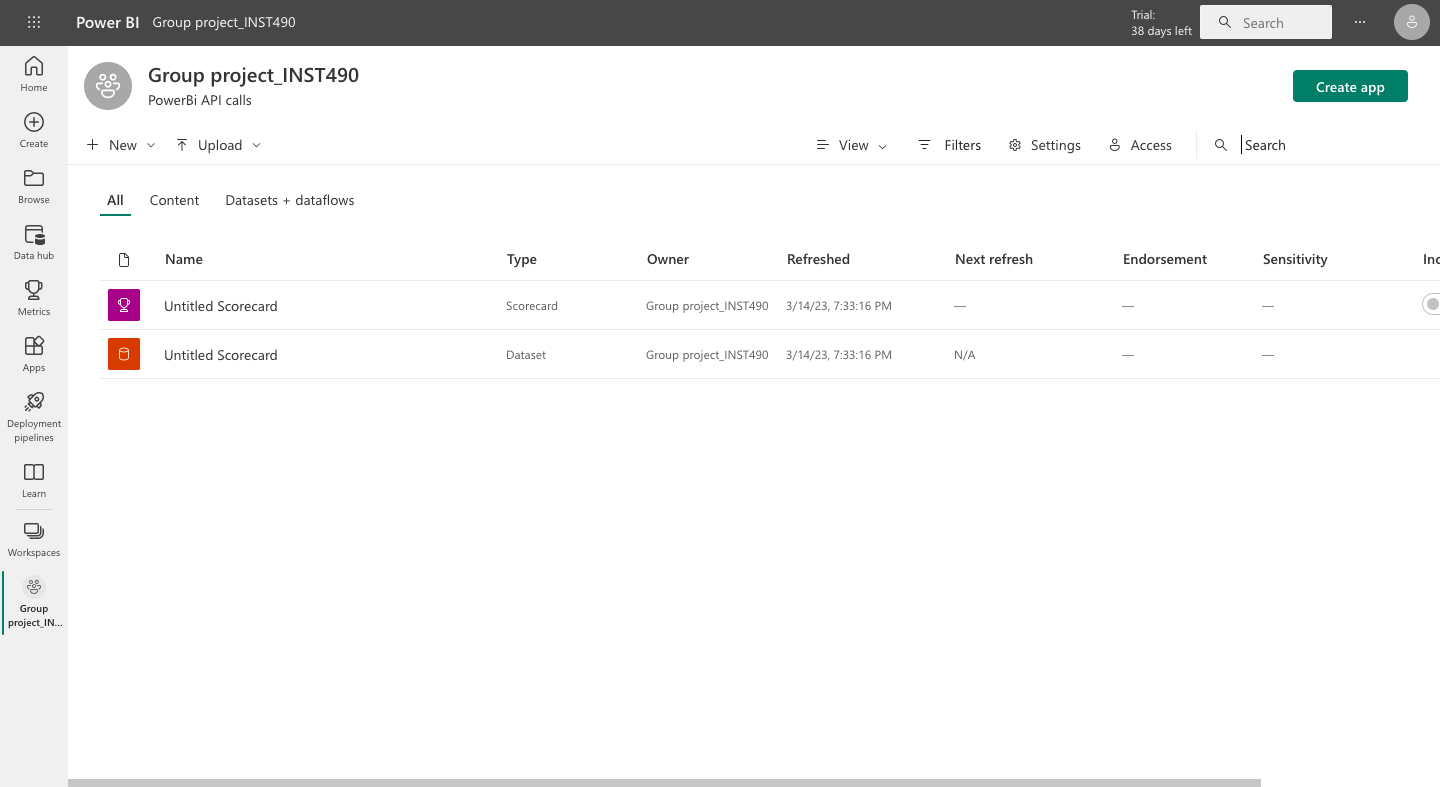
1. To get started go to: <https://powerbi.microsoft.com/en-us/>
2. Create an account with your personal or school email. I recommend using your school email because you might be able to get a free trial for a certain number of days if you are a student affiliated with University of Maryland, College Park.
3. One you sign in, the home page will look like the following

****

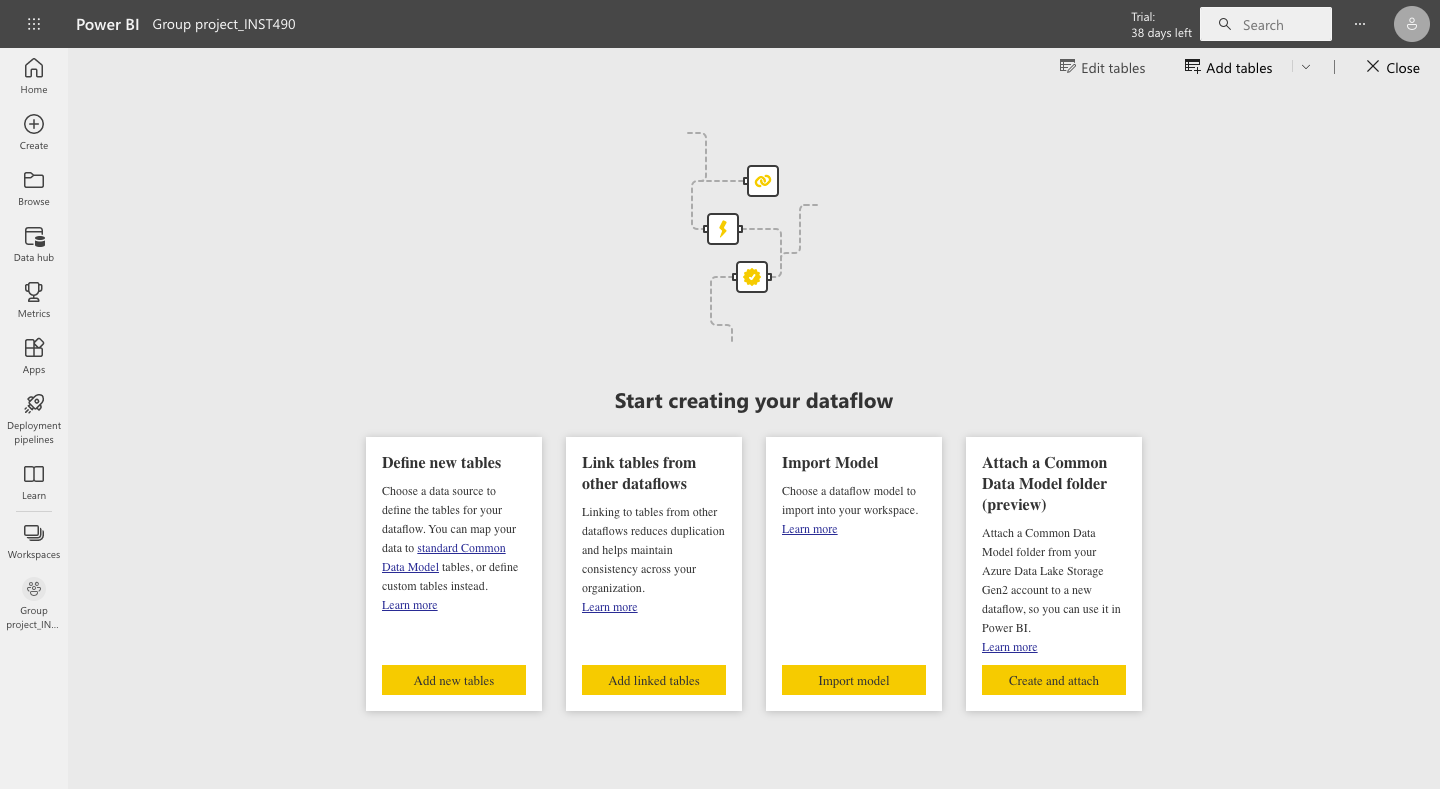
1. The main menu is on the left hand side, create a new workspace by clicking on workspaces.



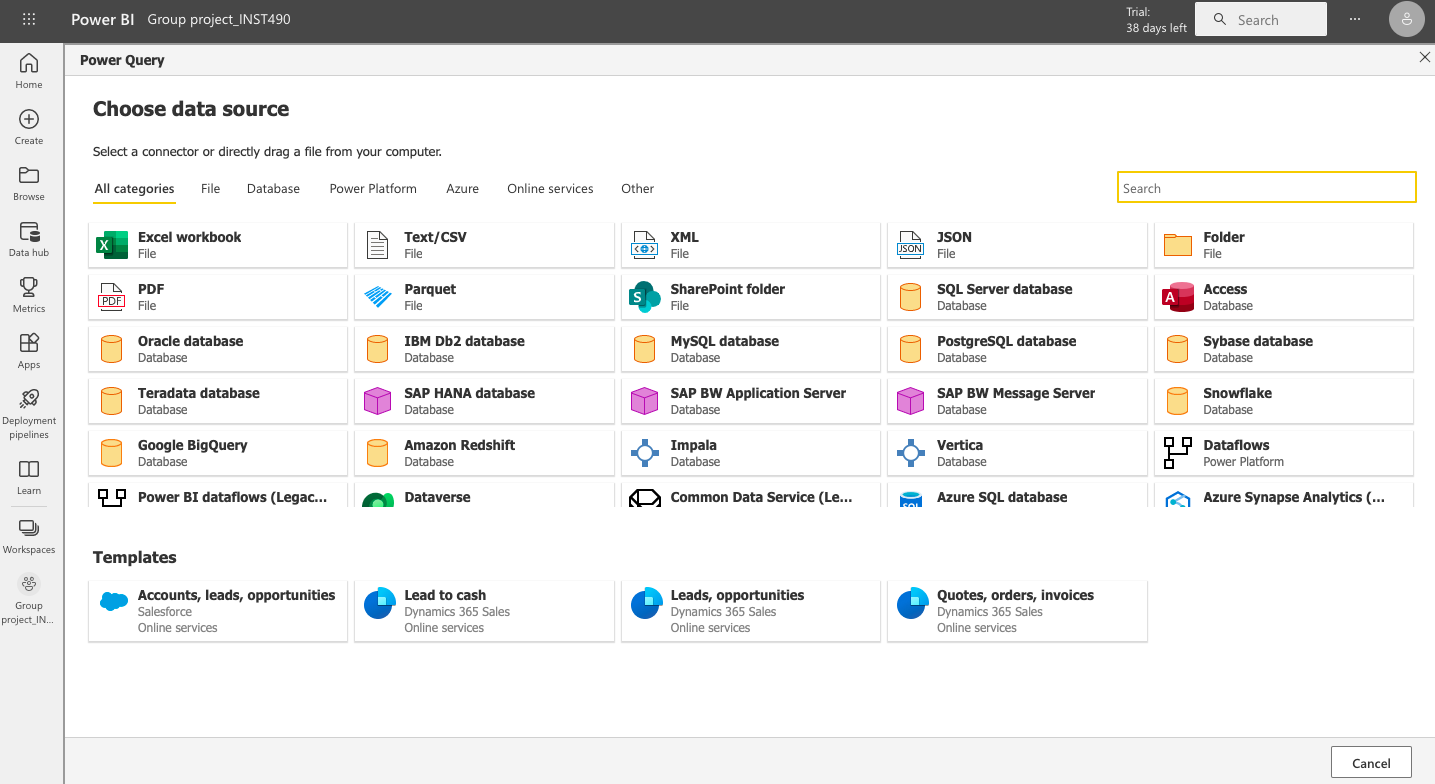
1. Once you’ve created a workspace, give it a name. For example ‘’ INST490\_ group project.
2. Once the workspace is created you can access it where you can see more features



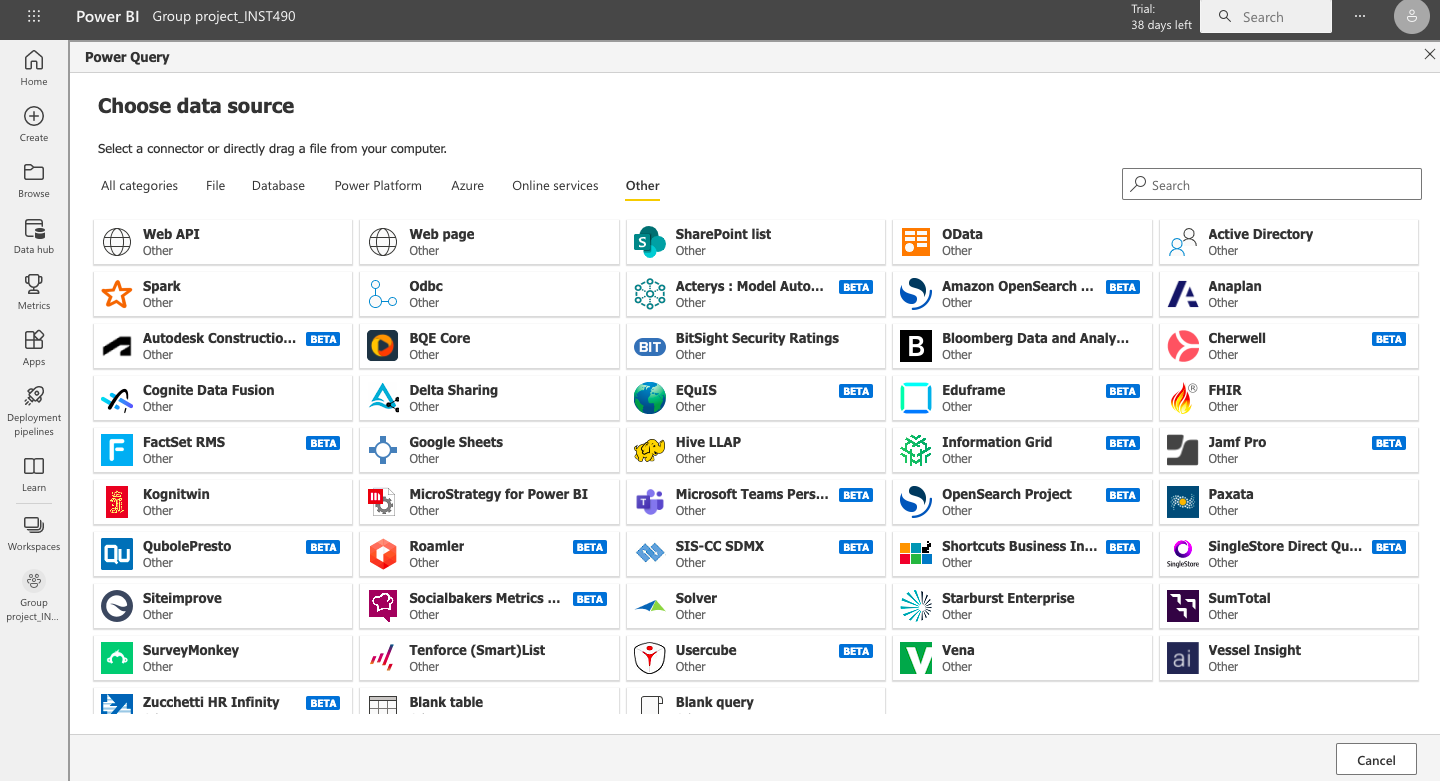
1. Under the workspace’s name there is a plus sign labeled as new, click on new, a drop down menu will pop out. Select Data flow.



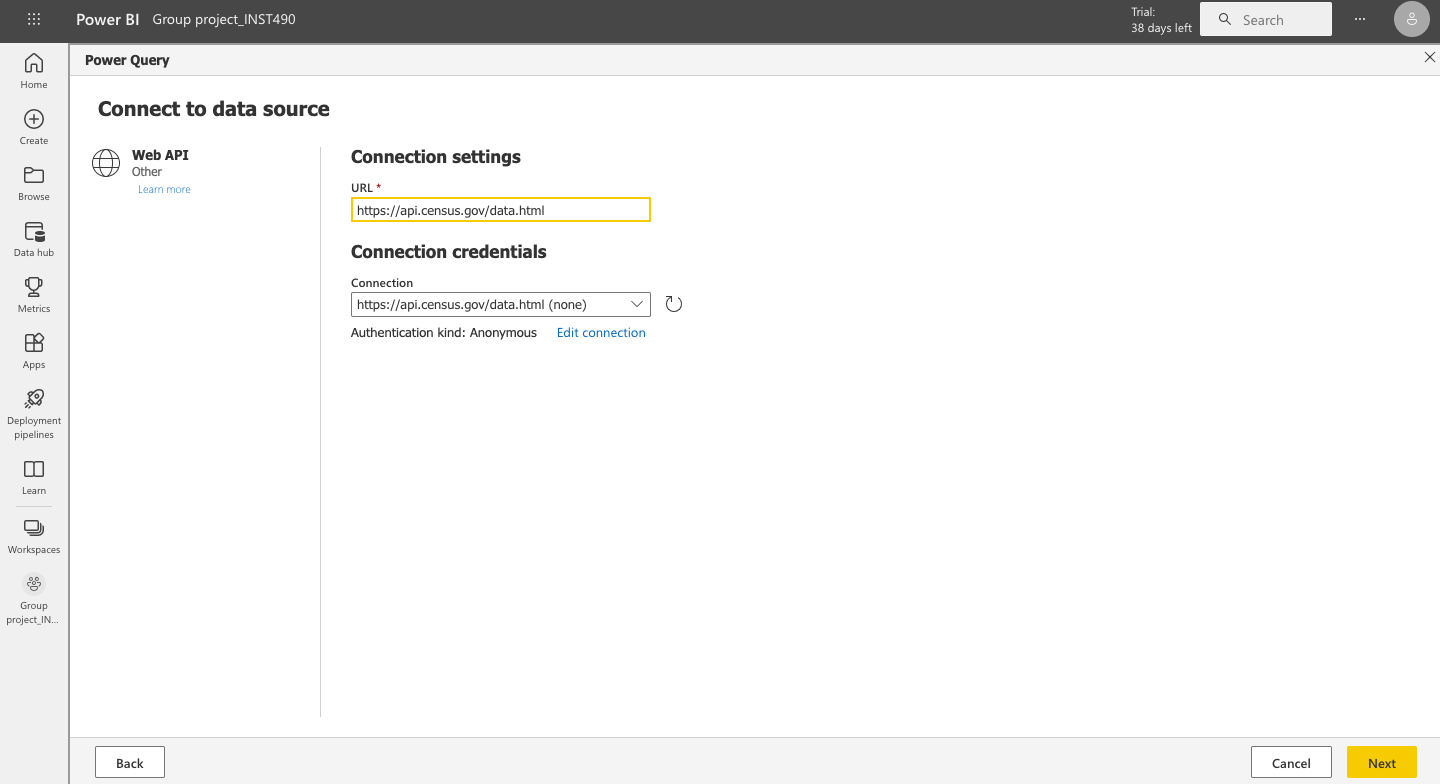
1. Once you have accessed the data flow you will have 4 options as you can see above. Select the first options ‘’ Define new tables’’
2. Once you select Define new tables, it will take you to a section where you can select the data source in order to access the census data API. Under choose data source it asks you to select a connector or drag a file from a computer. In this case you can choose ‘’other’’ from the menu below ‘’ Choose data source’’

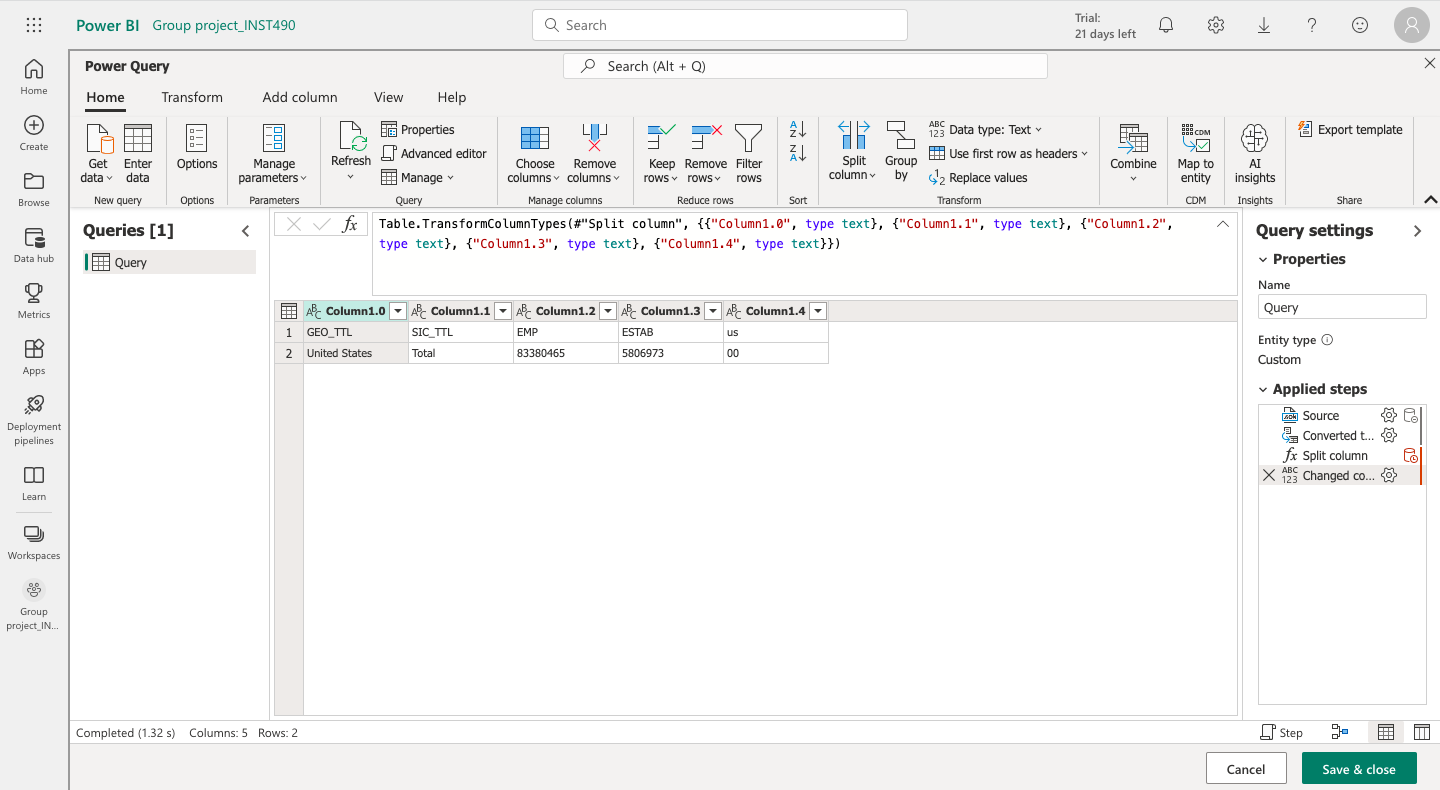


1. Once you click on Other, you will have the option to select Web API.



1. Once you click on Web API, it will ask you to put in the URL to connect to the census data. After that you can click “next” and your data will be ready!





Once you put in the dataset API link it should connect you to the main PowerBI platform.

# **7.0. R**

**The full R code snippet is available in the Github repository**

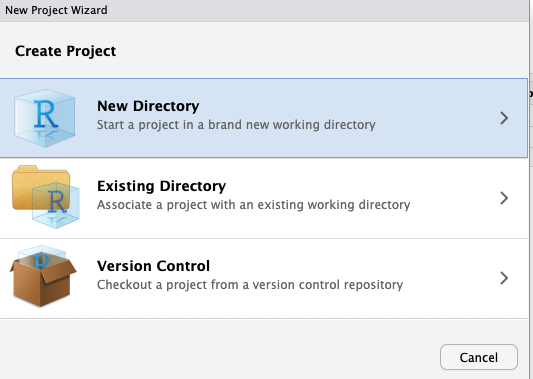
To get started with R, you should download RStudio, a software application that makes R easier to use. To download R and RStudio applications, please [click here](https://cran.r-project.org/).

**For Mac users**, click the "Download R for Mac" button. Then click on the R-3.0.3 package link. An installer will download and guide you through the installation process. If your computer requires a password before installing new applications, you should be able to input your password.

**For Windows users**, click the "Download R for Windows" button. Then click the "base" link. Next, click the first link at the top of the new page. This link will lead you to the page "Download R 3.0.3 for Windows" which will replace not the latest version of R. Follow the steps of the installation wizard to install.

1. **Click "File" and click New "Project".**



1. **Click "New Directory"**

# 

# 

# 

# 

# 

# **Click "New Project".**

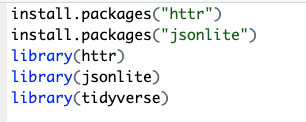
# **Input Directory name of yours and make sure you save in the appropriate folder. Then click "Create Project".**

# Step 4 will lead you to this console. Click the green plus button at the top left corner and click R script.

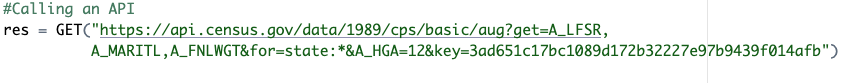
# 

# 

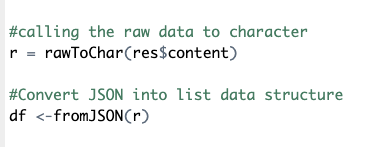
# After you open up the R Script, you should install required packages "httr" and "jsonlite" then call the library of each package. "tidyverse" library is needed to change data types. The image below demonstrates how to install and call packages.



# R uses different types of requests for different action; GET, POST, PUT. To create a GET request, the GET() function requires a URL of the dataset. To check the GET() output, call the output like line 7. The below image demonstrates how to use the GET() function.



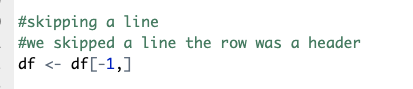
1. Census datasets are usually in JSON format. JSON is formatted as a series of key-value pairs meaning "key" is associated with a particular value. To handle JSON format, rawToChar() function is required. This will convert to JSON structure in character format.We can convert into list data structure using … from JSON() function from jsonlite library. The below image demonstrates how to use rawToChar() function.



1. Create a data frame of the JSON dataset for data cleaning/modification

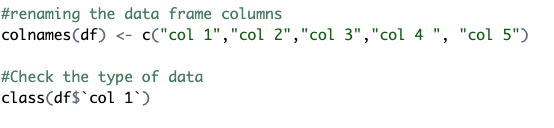


1. If you need to remove the first row, you can simply skip a line using the code below.

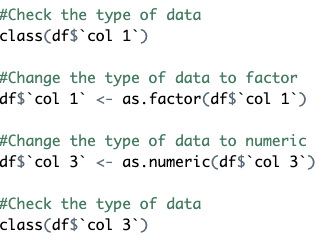


1. To rename columns of the dataset, you can simply rename the columns using the code

Below. In the order of "col1", "col2", … "col5" indicates the order of columns.



1. To change the type of data to factor or numeric, you can simply change the data type using the code below.



If you need more information about R API calls, please use the [link](https://cran.r-project.org/web/packages/jsonlite/jsonlite.pdf).

# **8.0. Excel**

**For Excel, we recommend using the “ download\_file\_python\_snippet ” script, which lets you call the API, load it into a pandas Dataframe, and save it as a .csv file using Python. You may then import the csv file as you normally would into Excel.**

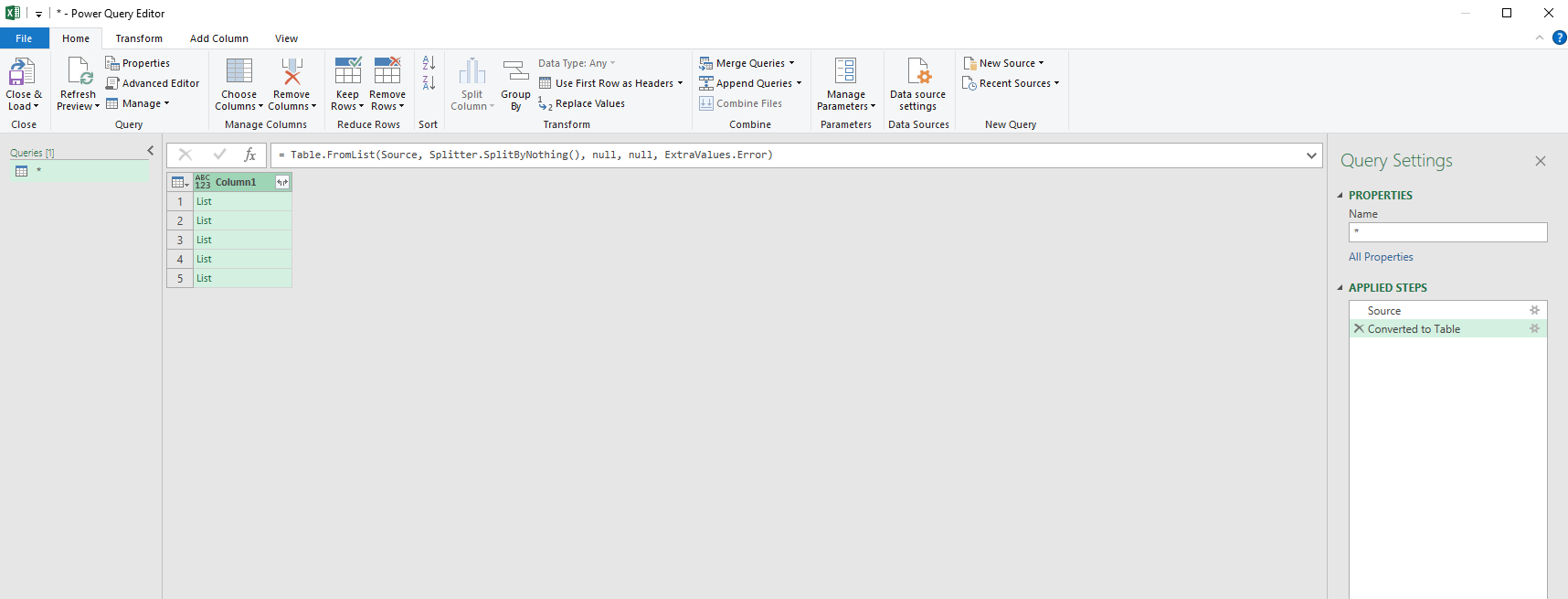
If you are opposed to using the python script, there are several ways to make an API call using Excel. The easiest way to make an api call from excel without having to learn a new programming language is the PowerQuery tool.

* **Power query**. PowerQuery is a plugin for microsoft products such as Excel and PowerBI, and has a user-friendly interface.

(*Note*: You can't access external api’s from excel on Mac products. PowerQuery for Excel on Mac is limited and can not currently connect to external web applications to import data. Mac’s are essentially “sandboxed”. One way to get around this is to create a virtual machine with windows installed on it.

**Steps for making an api call using Power query.**

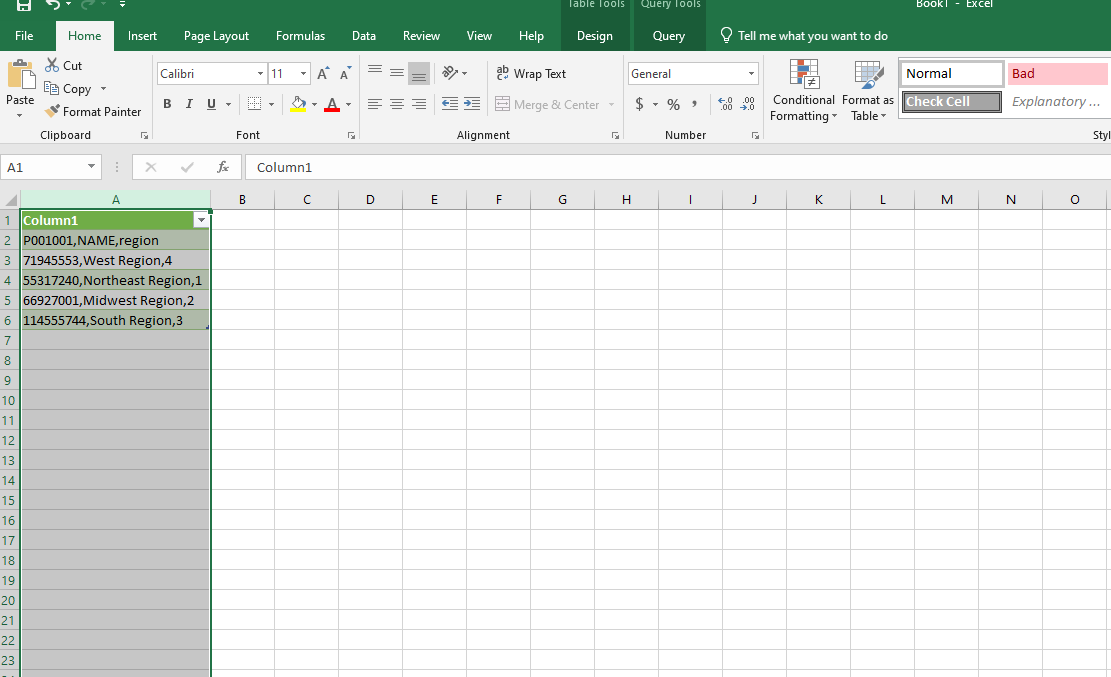
* Click data tab
* Click from web
* Paste the web link with parameters already included & click ok
* The power query editor will now pop up and the data will now be imported into Excel as a list of lists.

PowerQuery wasn't able to recognize the existing data format the census had their data in, in this case a list of lists, and as you can see from the picture below, it simply displayed that there were several lists within a list without displaying the actual values in said lists. When trying to extract the values from the lists, the delimiters present in the data, in this case commas, got deleted. 

Because of this, it will require some minor “finessing” in order to get the data into our desired format. To demonstrate how to manipulate the imported data into our desired format, well be using the following api call which was also shown at the top of the document,  
https://api.census.gov/data/2010/dec/sf1?get=P001001,NAME&for=region:\*

The following steps are how we manipulated the data into the desired format.

* Under transform, select “to table”
* Under the “select or enter delimiter” option select “none” & for “How to handle extra columns” select the “show as errors” option & click ok.
* At the top right of the column, there is a button, that if you click will display two options, expand to new rows and extract values. We want to click extract values.
* A pop up will display that says “Select a delimiter to use for concatenating list values” We want to click comma, then click ok.
* We now want to click close and load in the top right which can be found in the top left under the home tab of the PowerQuery editor.
* As we can see below, all the data is now imported into an excel spreadsheet under one column. We originally had 3 columns which consisted of P001001, NAME, and region, so we will need to create new columns to separate our data.



* We must now click on the column as displayed above
* Click on the data tab and click “Text to Columns”
* A pop up will display and you will have two options: delimited and fixed width. You must click delimited then click Next.
* We must now make sure that the Comma box is checked under delimiters. Then click Next.
* Click Finish, and your data will now be separated into three columns as it was on the census api.

