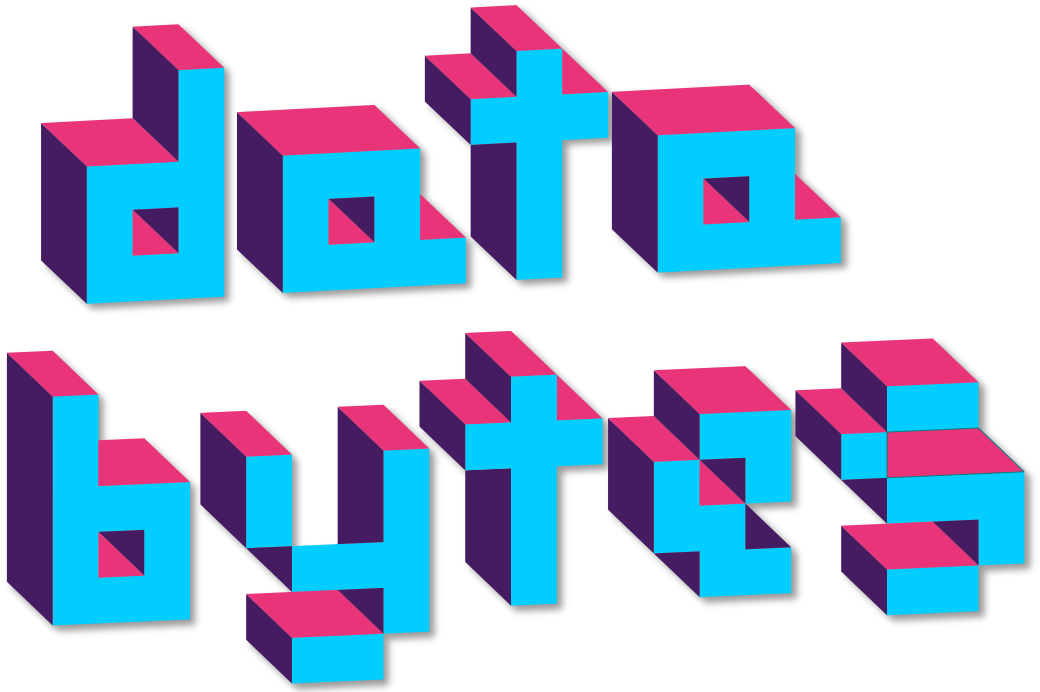
Introduction to APIs In R



Reina Chano Murray

JHU Data Services

github.com/jhu-data-services

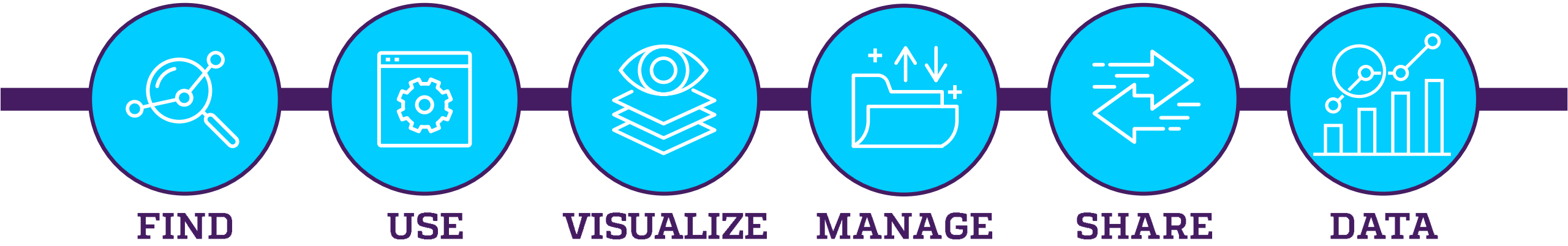
dataservices.library.jhu.edu

dataservices@jhu.edu

These materials are licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License, attributable to Data Services, Johns Hopkins University.

JHU Data Services

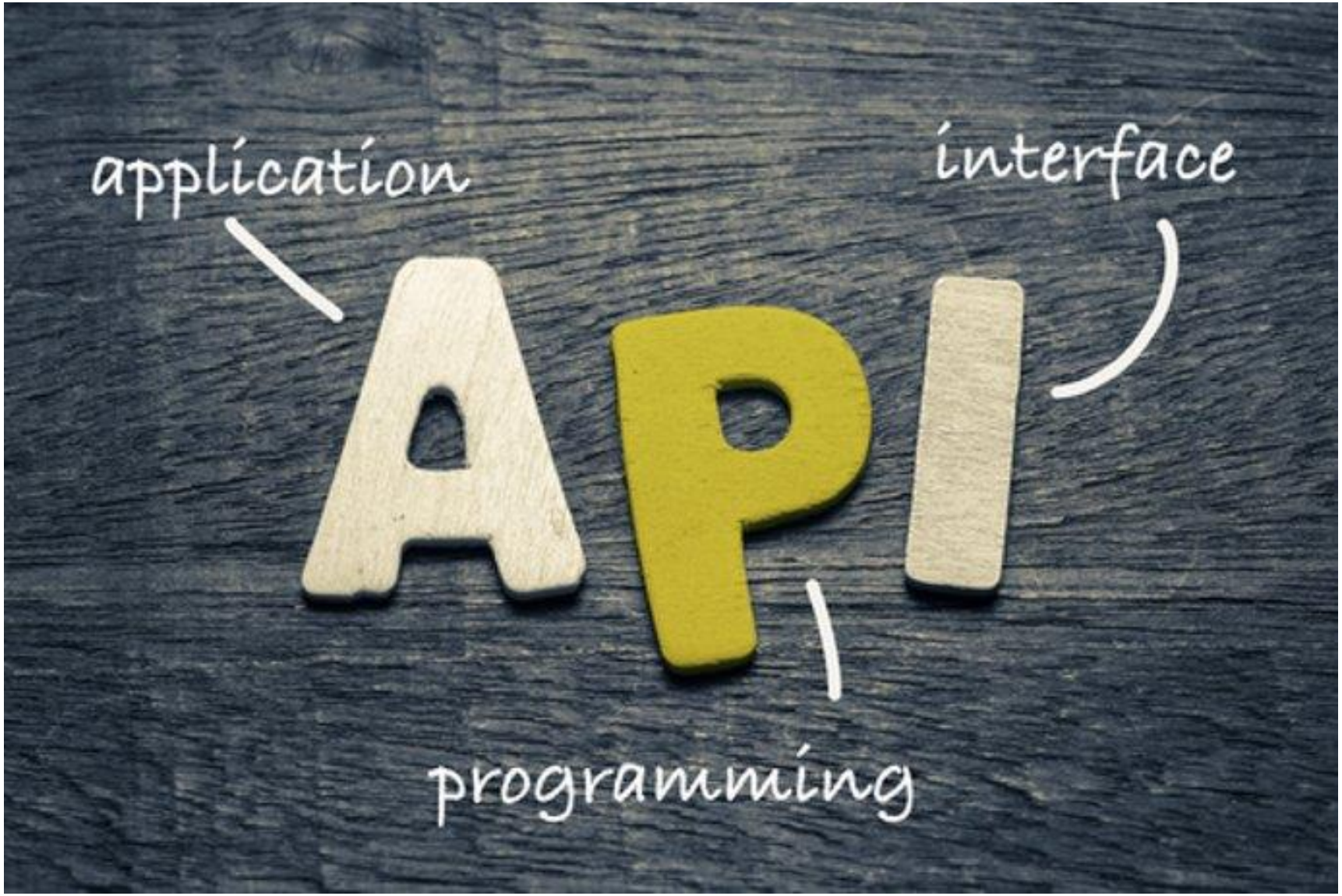
WE HELP FACULTY, RESEARCHERS AND STUDENTS:

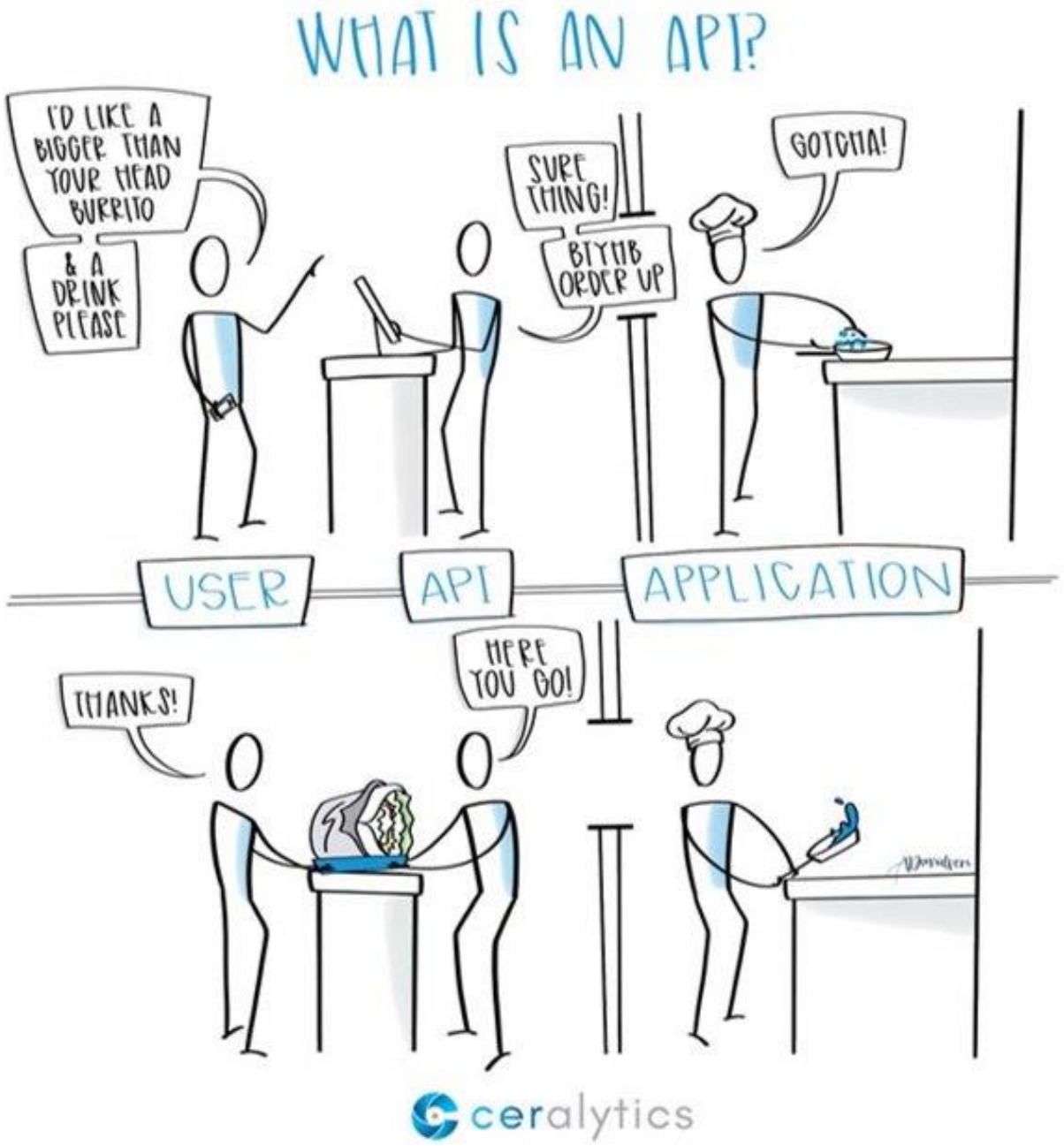


Agenda

- Introduction to APIs
  - What are they?
  - Why use them?
- Getting started with APIs
  - What you need
- Demonstrations with U.S. Census Bureau APIs

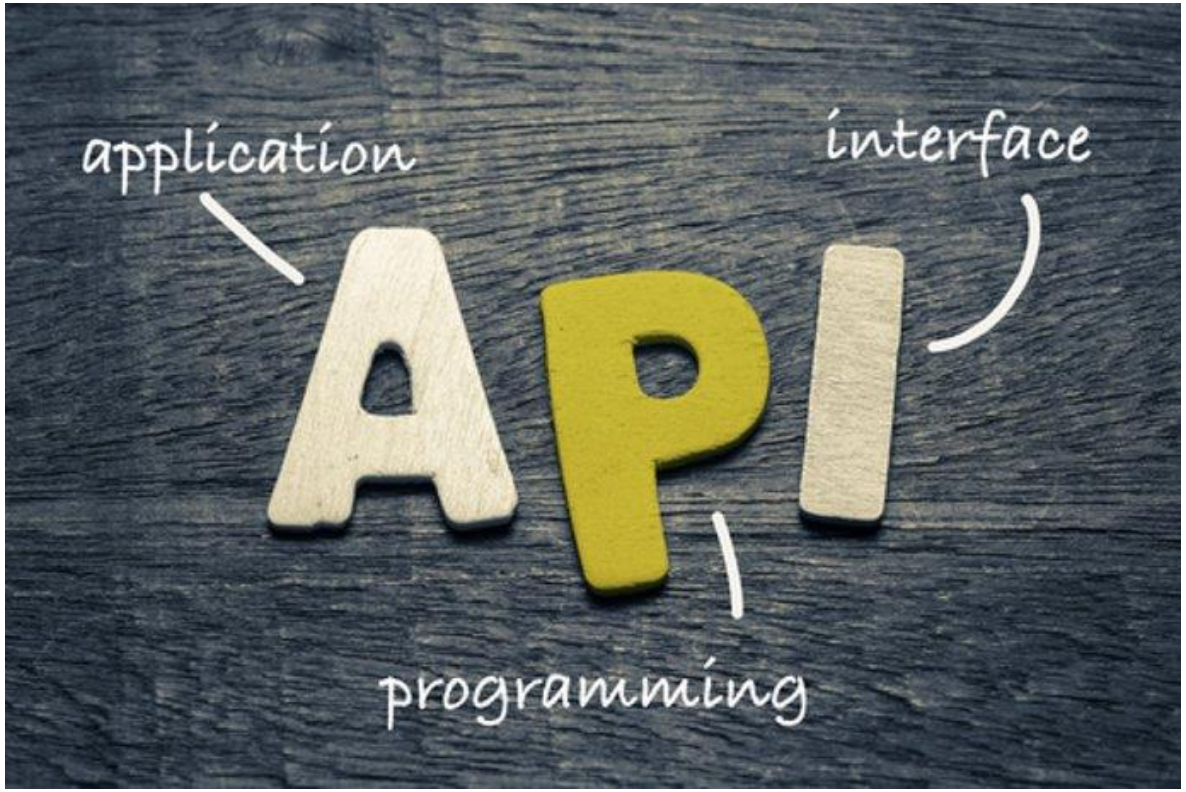
Introduction to APIs


application
interface
API
programming



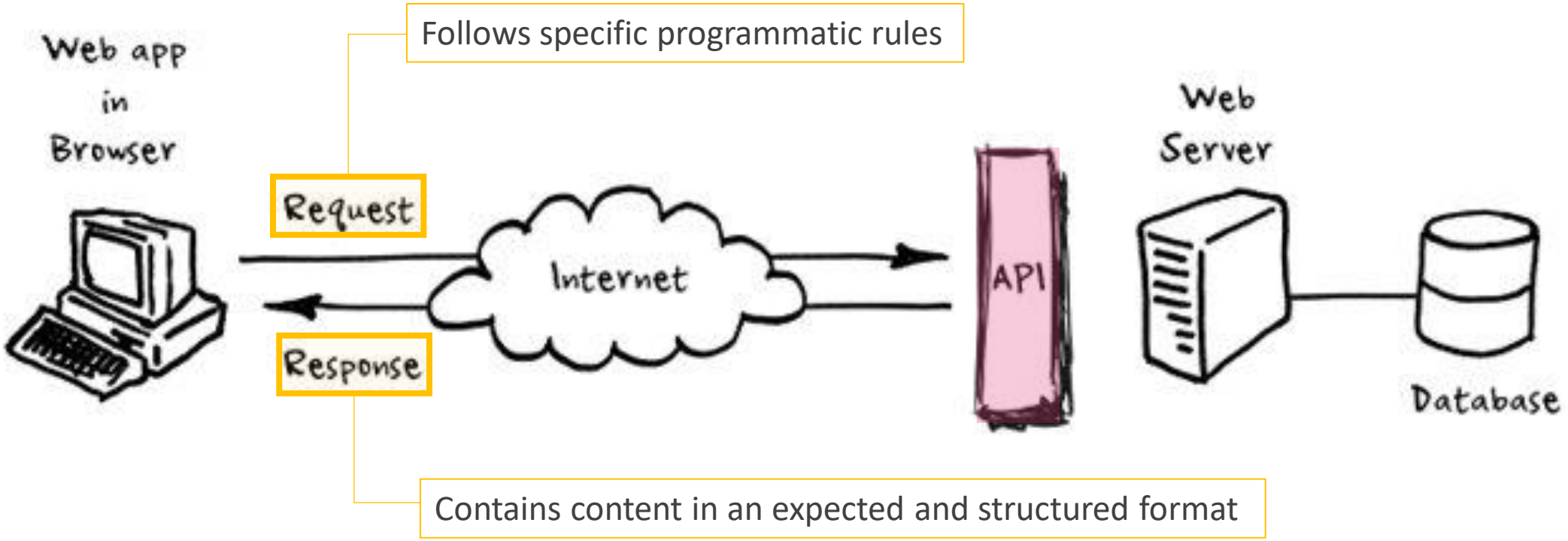
Ordering a Meal

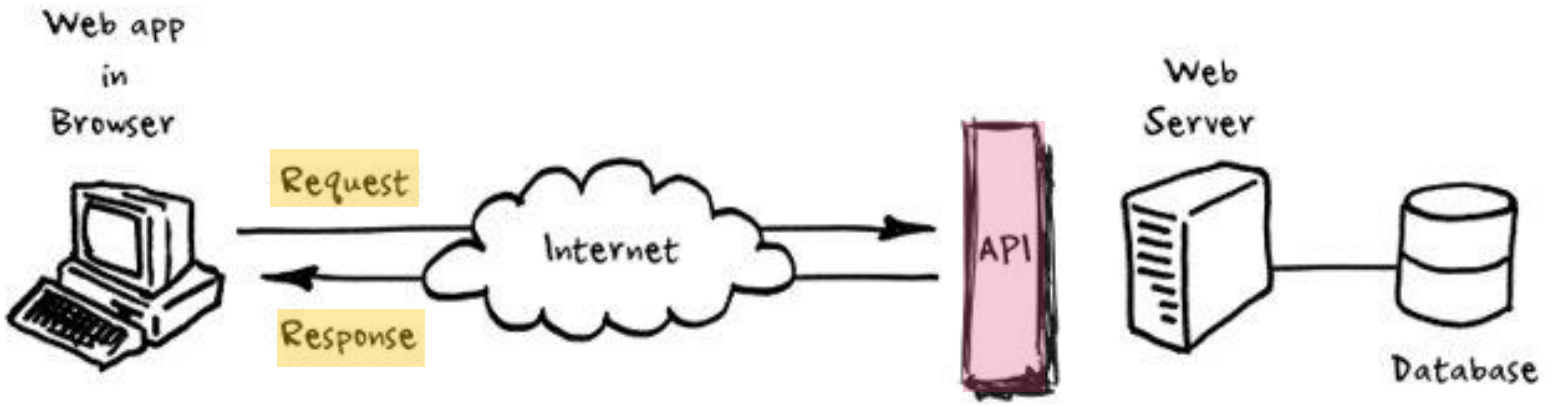
- Customer makes an order
- Server takes down order, processes it, and passes it to cook
- Cook acknowledges order
- Cook makes the order and gives it to the server
- Server gives the customer their order



APIs

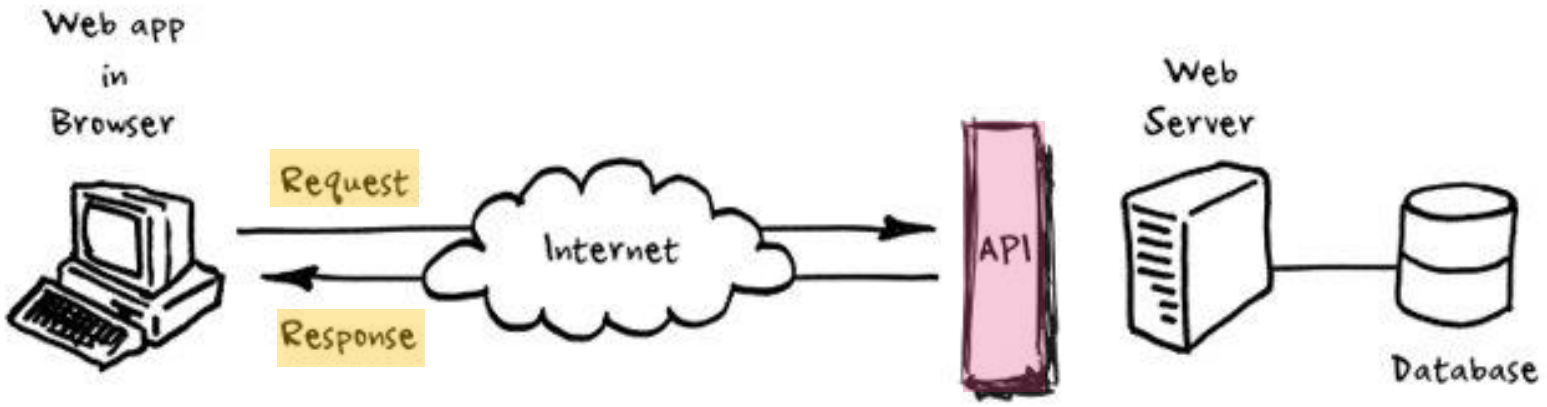
- User (client) makes a **request**
- API takes down request, processes it, and passes it to the application
- Application acknowledges request
- Application processes request and signals API
- API returns a **response** to user


Follows specific programmatic rules
Web app in Browser
Request
Response
Internet
API
Web Server
Database
Contains content in an expected and structured format



APIs

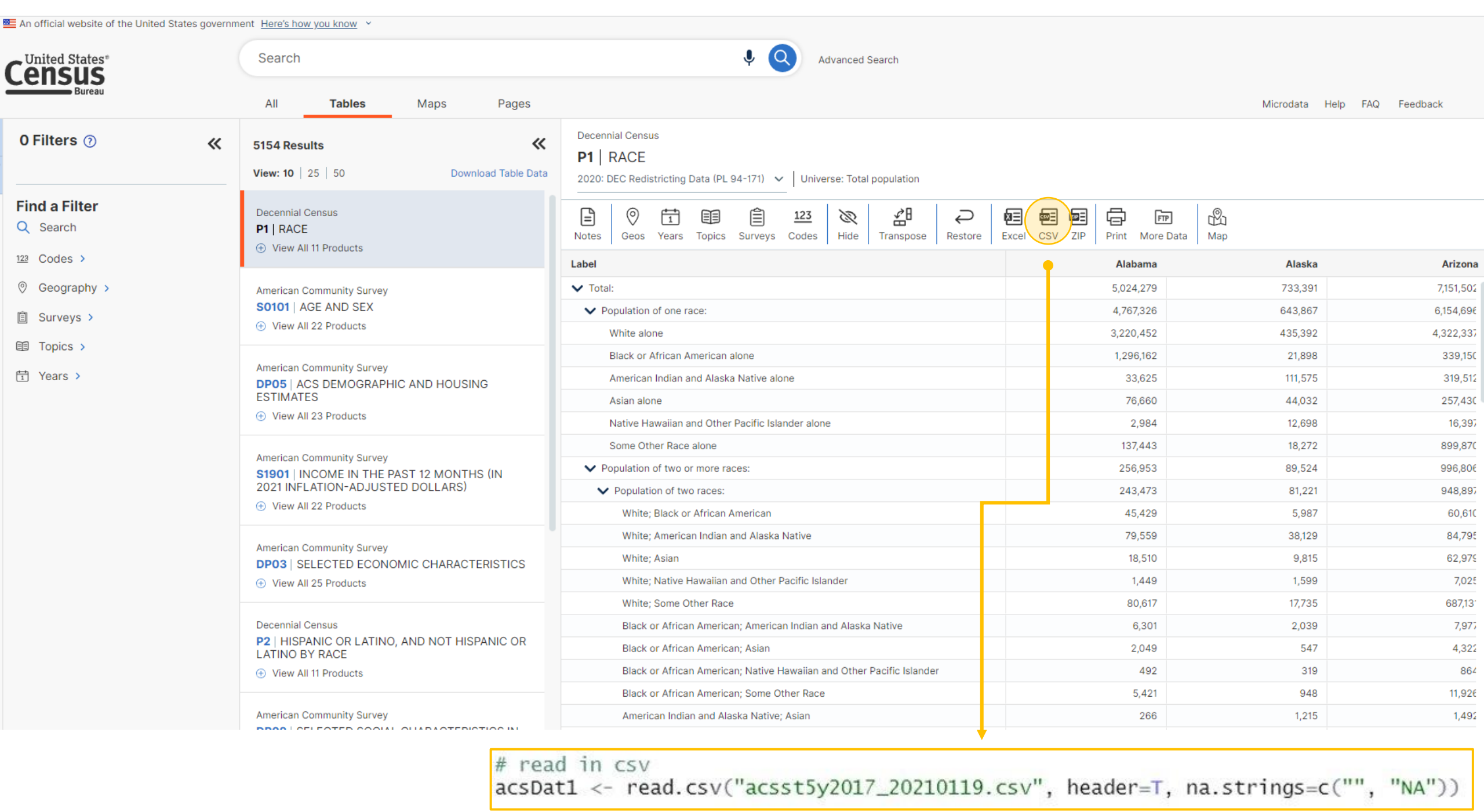
- Is **NOT** the database or the server – it is the code that governs the access point(s) for the server
- APIs cover a broad category that includes all interfaces that facilitate communication between computer applications
- We're talking about **Web APIs** – APIs that expose an application's data and functionality over the internet, allowing two computers (the client and the server) to interact with each other to request and provide data.



4 main types of web APIs

1. Open / public APIs – no restrictions to access
2. Partner APIs – requires specific rights and/or licenses
3. Internal APIs – designed for internal use within an organization
4. Composite APIs – combines different data and service APIs

Why Use an API?




```
# read in csv
acsDat1 <- read.csv("acsst5y2017_20210119.csv", header=T, na.strings=c("", "NA"))
```

APIs for Research

- **Get clean and curated data directly from a source**
  - Websites are designed to be easy for humans to read, not for computers. Web scraping often requires additional cleaning and parsing of the data.
  - Data pulled from an API is usually formatted to be immediately usable.
  - Get access to most recent, up-to-date data
- **Integrate and automate your data collection workflow**
  - APIs can let you forgo the process of manually download datasets from a website, managing the data in a file management system, and loading it into the software you are using for analysis.

Working with an API

1. Explore the documentation for the API
2. Make your request
3. Review response

1

Explore the documentation for the API

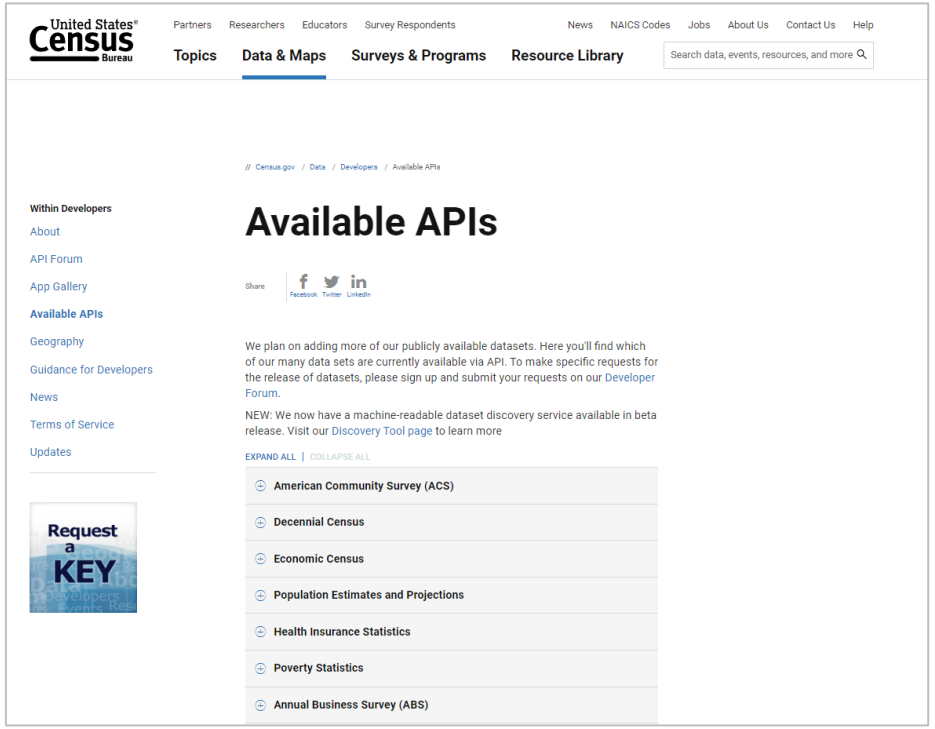
- Do they supply an API that does what you're looking for?
- Review the process for getting access to the API:
  - Do you need to apply / create an account?
  - Applying for an API key – unique identifier to authenticate your particular requests
  - Query limits (if any)

2

Make your request

3

Review response



https://www.census.gov/data/developers/data-sets.html

1

Explore the documentation for the API

- Do they supply an API that does what you're looking for?
- Review the process for getting access to the API:
  - Do you need to apply / create an account?
  - Applying for an API key – unique identifier to authenticate your particular requests
  - Query limits (if any)

2

Make your request

- Prepare the URL
- Query the API

3

Review response

- Get the response
- Convert response to a dataframe

US Census Data API

- https://www.census.gov/data/developers/guidance.html
- Available data
  - American Community Survey (ACS), Economic Indicators Time Series, Decennial Census, Economic Census, County Business Patterns and Nonemployer Statistics, Population Estimates and Projections, International Trade, etc...
- Query small quantities of data (up to 50 variables in a single query, and up to 500 queries per IP address per day)
  - To make more than 500 queries per IP address per day, request an API key

Creating Queries in a Browser

https://api.census.gov/data/2018/pep/charagegroups?get=POP,GEONAME,DATE_DESC&DATE_CODE=11&RACE=10&for=county:*&in=state:24

Creating Queries in a Browser

https://api.census.gov/data/2018/pep/charagegroups?get=POP,GEONAME,DATE_DESC&DATE_CODE=11&RACE=10&for=county:*&in=state:24

1. Start with the host name

Creating Queries in a Browser

https://api.census.gov/data/2018/pep/charagegroups?get=POP,GEONAME,DATE_DESC&DATE_CODE=11&RACE=10&for=county:*&in=state:24

1. host name

2. Add dataset name
(list available at https://api.census.gov/data.html)

Creating Queries in a Browser

https://api.census.gov/data/2018/pep/charagegroups?get=POP,GEONAME,DATE_DESC&DATE_CODE=11&RACE=10&for=county:*&in=state:24

1. host name

2. Add dataset name
(list available at https://api.census.gov/data.html)

3. Add variable requests
using `?get=`

Creating Queries in a Browser

https://api.census.gov/data/2018/pep/charagegroups?get=POP,GEONAME,DATE_DESC&DATE_CODE=11&RACE=10&for=county:*&in=state:24

1. host name

2. Add dataset name
(list available at https://api.census.gov/data.html)

3. Add variable requests using `?get=`

4. Filter variables using `&for=`

Creating Queries in a Browser

https://api.census.gov/data/2018/pep/charagegroups?get=POP,GEONAME,DATE_DESC&DATE_CODE=11&RACE=10&for=county:*&in=state:24

1. host name

2. Add dataset name
(list available at https://api.census.gov/data.html)

3. Add variable requests using `?get=`

4. Filter variables using `&for=`

5. If needed, add your key using `&key=`

Click here to see RESULT

Creating Queries in a Browser

https://api.census.gov/data/2018/pep/charagegroups?get=POP,GEONAME,DATE_DESC&DATE_CODE=11&RACE=10&for=county:*&in=state:24

1. host name

2. Add dataset name
(list available at https://api.census.gov/data.html)

3. Add variable requests using `?get=`

4. Filter variables using `&for=`

5. If needed, add your key using `&key=`

- Making queries in your browser is a good way to practice and get comfortable with the variables and filters available in the Census Data API
- Works best on Chrome or Firefox
- In web browser, right-click and select “Save As” to save results as a csv
- Sample queries: https://www.census.gov/data/developers/guidance/api-user-guide.Example_API_Queries.html

Demonstrations

Demo 1

- Install and load RJSONIO – a package that converts data to and from JSON format
- Structure mimics creating API queries in a browser
- Works – but tedious. You need to make sure you understand the structure of your data well, and convert JSON to R objects.

Demo 2

- tidycensus: an integrated package that allows users to access decennial census and ACS data as tidyverse-ready data frames.
- tidyverse: collection of R packages designed for data science. (come to our other R workshops to learn more about this collection)

R packages for data science

The tidyverse is an opinionated **collection of R packages** designed for data science. All packages share an underlying design philosophy, grammar, and data structures.

Install the complete tidyverse with:

```
install.packages("tidyverse")
```

Demo 3

- censusapi: R wrapper for US Census Data API.
- Covers more datasets than tidycensus.
- Returns data frames of census data **and metadata** for Decennial, ACS, Small Area Health Insurance Estimates, Small Area Income and Poverty Estimates, Population Estimates and Projections, etc.

Fall 2022 Data Bytes Schedule

All sessions are on Mondays from 12 to 1 pm.

Finding Maps and Map Data
Sept 12th

Choosing a Python IDE
Sept 19th

Creating Infographics in Business Analyst
Sept 26th

Debugging your Python Code
Oct 3rd

Speeding up your Python Code
Oct 10th

Introduction to Leaflet
Oct 17th

Advanced StoryMaps Tips and Tricks
Oct 24th

Introduction to APIs in R
Oct 31st

GIS and Maps Programming

More info at:
bit.ly/data-bytes

Thank you for attending!

Please complete our survey at:
bit.ly/data-bytes-survey

Questions?