

Course data & APIs

Hitchhiker's Guide to Reproducible Research

Julia Wrobel and David Benkeser

 Course Website

CDC open data

The CDC has [1331](#) open-source datasets available at data.cdc.gov.

- Topic areas include injury & violence, vaccination, smoking, pregnancy, chronic disease, and disease surveillance
- Great source of Covid surveillance data

FYI, under the OPEN Government Data Act (2018), government data is required to be made available in open, machine-readable formats, while continuing to ensure privacy and security.

- data.gov hosts additional 250K+ datasets

Covid19 wastewater data

Throughout this course, we will work with the National Wastewater Surveillance System (NWSS) Public SARS-CoV-2 Concentration in Wastewater Data.

- SARS-CoV-2 concentration at different sampling locations
- Updated daily

Longitudinal data

- Provides concentrations over time
- 4 columns

Cross-sectional data

- Current concentrations and other summaries
- 16 columns, including state, county

Covid19 wastewater data

We will merge these datasets and analyze concentration over time in different counties.

- Data can be downloaded from [Socrata](#)
 - Socrata is a cloud-based platform used by many local/state/federal governments for data-sharing
- Goal will be to produce end-to-end reproducible workflows with this data

Covid19 wastewater data


NWSS Public SARS-CoV-2 Concentration in Wastewater Data

👍 *Good to go!* You're already using the latest version of this dataset API.

This dataset provides a complete time history of SARS-CoV-2 concentrations in wastewater for each sampling location. All dates - even when no wastewater sample was collected at a given sampling location - are provided for each sampling location since wastewater sampling started at that location.

Getting Started

All communication with the API is done through HTTPS, and errors are communicated through HTTP response codes. Available response types include `JSON` (including `GeoJSON`), `XML`, and `CSV`, which are selectable by the "extension" (`.json`, etc.) on the API endpoint or through content-negotiation with HTTP `Accepts` headers.


This documentation also includes inline, runnable examples. Click on any link that contains a  `gear` symbol next to it to run that example live against the `NWSS Public SARS-CoV-2 Concentration in Wastewater Data` API. If you just want to grab the API endpoint and go, you'll find it below.

```
▶ try it | docs | copy | <> json ▼  
🔗 https://data.cdc.gov/resource/g653-rqe2.json
```

Learn more about:

- [Getting started with the SODA Consumer API](#)
- [Output formats and content negotiation](#)
- [Response codes & error messages](#)
- [How to stay up to date on API changes](#)

About This Dataset

Dataset Identifier: `g653-rqe2`
Total Rows: 201983
Source Domain: data.cdc.gov
Created: 3/30/2022, 3:20:50 PM
Last Updated: 7/14/2024, 9:11:27 PM
Category: Public Health Surveillance
Attribution: National Wastewater Surveillance System
Owner: [John Person](#)
Endpoint Version: 2.1
Embed These Docs:  [copy](#)

[View dataset »](#)

Download & Export

Just want to grab this dataset in bulk to analyze offline? You can use the [SoQL paging parameters](#) to iterate through the dataset, or you can export the entire dataset as a static, downloadable CSV file.

[Export dataset as CSV »](#)

APIs

Luckily, this data can be accessed reproducibly using an **API**.

API: Application Programming Interface

- Essentially, a set of rules and tools that allows different software programs to communicate with each other. Think of it as a waiter in a restaurant: you tell the waiter what you want (make a request), and the waiter brings you what you asked for (the response). In the same way, an API lets one program request data or services from another program, and then delivers that data back. This makes it easier for different software systems to work together.
- A way to extract and share data within and across organizations. You are using an API every time you use a rideshare app, send a mobile payment, or use Google Maps.
- APIs have protocols that are standardized across apps and websites

APIs

What does this mean for us?

- We can download CDC data using a script rather than point-and-click
- R interface for Socrata (`RSocrata` package) means this can be done directly in RStudio
- Leads to more reproducible workflows

Let's take a look...

Accessing data through RSocrata

```
library(tidyverse)
```

```
## — Attaching core tidyverse packages ————— tidyverse 2.0.0
## ✓ dplyr      1.1.4      ✓ readr      2.1.5
## ✓ forcats   1.0.0      ✓ stringr    1.5.1
## ✓ ggplot2   3.5.1      ✓ tibble     3.2.1
## ✓ lubridate 1.9.3      ✓ tidyr      1.3.1
## ✓ purrr     1.0.2
## — Conflicts ————— tidyverse_conflicts
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
```

```
library("RSocrata")
```

```
df = read.socrata(url = "https://data.cdc.gov/resource/g653-rqe2.json")
```

```
tibble::as_tibble(df)
```

```
## # A tibble: 206,129 × 4
##   key_plot_id                                date pcr_conc_lin normaliz
##   <chr>                                           <chr> <chr>         <chr>
## 1 CDC_VERILY_al_2629_Treatment plant_post gri... 2024... 53341336.07... flow-pop
```


Accessing data through RSocrata

```
?read.socrata
```

Usage

```
read.socrata(  
  url,  
  app_token = NULL,  
  email = NULL,  
  password = NULL,  
  stringsAsFactors = FALSE  
)
```

Terminology

App Token

- A unique passcode of letters and numbers that grants access to an API
- Each user signs up for their own token [here](#)
- "A limited number of requests can be made without an app token, but they are subject to much lower throttling limits than request that do include one."

Fields

- Correspond to columns in the dataset (`key_plot_id` is 1st field in our Covid dataset)
- Can use SoQL **queries** for each field to filter what is returned by each API request

App token

```
df <- read.socrata(  
  "https://data.cdc.gov/resource/g653-rqe2.json",  
  app_token = "YOURAPPTOKENHERE",  
  email     = "user@example.com",  
  password  = "fakepassword"  
)
```

Example query

```
df = read.socrata(  
  "https://data.cdc.gov/resource/g653-rqe2.json?key_plot_id=NWSS_wv_  
  )  
  
tibble::as_tibble(df)
```

```
## # A tibble: 1,006 × 4  
##   key_plot_id                date    pcr_conc_lin normaliz  
##   <chr>                      <chr>  <chr>         <chr>  
## 1 NWSS_wv_2558_Treatment plant_raw wastewater 2024-... 100976425.3... flow-pop  
## 2 NWSS_wv_2558_Treatment plant_raw wastewater 2024-... 36693803.39... flow-pop  
## 3 NWSS_wv_2558_Treatment plant_raw wastewater 2024-... 72412878.28... flow-pop  
## 4 NWSS_wv_2558_Treatment plant_raw wastewater 2024-... 58113474.08... flow-pop  
## 5 NWSS_wv_2558_Treatment plant_raw wastewater 2024-... 46608112.37... flow-pop  
## 6 NWSS_wv_2558_Treatment plant_raw wastewater 2024-... 65282077.04... flow-pop  
## 7 NWSS_wv_2558_Treatment plant_raw wastewater 2024-... 83258938.74... flow-pop  
## 8 NWSS_wv_2558_Treatment plant_raw wastewater 2024-... 29597111.69... flow-pop  
## 9 NWSS_wv_2558_Treatment plant_raw wastewater 2024-... 49155438.03... flow-pop  
## 10 NWSS_wv_2558_Treatment plant_raw wastewater 2024-... 961584.6527... flow-pop  
## # i 996 more rows
```

Another example query

```
df = read.socrata(  
  "https://data.cdc.gov/resource/g653-rqe2.json?$where=date>'2023-12-3  
  )  
  
as_tibble(df) %>%  
  arrange(date)
```

```
## # A tibble: 67,360 × 4  
##   key_plot_id                                date pcr_conc_lin normaliz  
##   <chr>                                       <chr> <chr>         <chr>  
## 1 CDC_VERILY_ks_2670_Treatment plant_raw wast... 2024... 73375611.46... flow-popu  
## 2 CDC_VERILY_tx_1140_Treatment plant_post gri... 2024... 75936785.57... flow-popu  
## 3 NWSS_az_1271_Treatment plant_raw wastewater 2024... 5286836.011... flow-popu  
## 4 NWSS_ca_211_Treatment plant_raw wastewater 2024... 100115253.8... flow-popu  
## 5 NWSS_ca_2330_Treatment plant_raw wastewater 2024... 1155987303.... flow-popu  
## 6 NWSS_co_122_Treatment plant_raw wastewater 2024... 48564123.71... flow-popu  
## 7 NWSS_de_1187_Treatment plant_raw wastewater 2024... 194741518.0... flow-popu  
## 8 NWSS_de_1194_Treatment plant_raw wastewater 2024... 145042261.4... flow-popu  
## 9 NWSS_ga_2697_Treatment plant_raw wastewater 2024... 156565896.2... flow-popu  
## 10 NWSS_in_1619_Treatment plant_raw wastewater 2024... 123362442.2... flow-popu  
## # i 67,350 more rows
```

