The Journal of Open Source Software

# Crunching COVID-19 Vaccination Data for Brazilian Municipalities

**Emanuele Guidotti**[1]

**1** University of Neuchâtel, Switzerland

**DOI:**

**Software**

- Review
- Repository
- Archive

**Submitted:**
**Published:**

## Summary

The Ministry of Health of Brazil, through the Information System of the National Immunization Program (SI-PNI), has made available data relating to the National Vaccination Campaign against COVID-19 for analysis by interested institutions and the public.[1]

The dataset is provided in a tidy data format (Wickham, 2014), where each row represents an administered COVID-19 dose. As of October 2021, there are more than 250 million records and 34 variables (columns) associated with each record, including information about the patient, the municipality of residence, the vaccination center, and the type of dose administered. The data are provided as a Character Separated Value (CSV) file, with a size over 130GB that makes it hardly accessible.

Although it is possible to crunch this amount of data using proprietary software and High Performance Computing (HPC), no easy way of extracting relevant information exists when we rely exclusively on freely available resources to enhance transparency and trust.

This paper shows how to extract typical vaccination metrics (Mathieu et al., 2021) for Brazilian municipalities from the large dataset, using only open-source software and freely available resources. The data extracted are made available in the form of lightweight, ready-to-use CSV files that are updated daily and encourage re-use.

This work joins the ongoing global efforts around the topic of COVID-19 data collection (Dong, Du, & Gardner, 2020; Guidotti & Ardia, 2020; Hale, Webster, Petherick, Phillips, & Kira, 2020; Hasell et al., 2020; Mathieu et al., 2021; Wang et al., 2020).

## Workflow

Working with large files poses several issues. First, a high-speed internet connection is needed to download the data file. Second, the disk size must be sufficienly large to store the data. Third, the data must be efficiently loaded in memory to avoid memory leaks. These issues are especially relevant when we rely exclusively on freely available resources.

### Working with large files and limited resources

To download large amounts of data at speed, one option is to rely on GitHub-hosted cloud runners. GitHub offers Linux runners with the following hardware resources:[2]

- 2-core CPU
- 7GB of RAM memory
- 14GB of SSD disk space

[1] https://opendatasus.saude.gov.br/dataset/covid-19-vacinacao

[2] https://docs.github.com/en/actions/using-github-hosted-runners/about-github-hosted-runners

The Ministry of Health of Brazil is providing the dataset in two ways: (1) a unique CSV file with the complete dataset, and (2) one CSV file for each Brazilian state (26 states plus the federal district). As the complete 130GB dataset is too large to be stored on disk, we need to work with the dataset split by state. However, also the individual CSV files can be larger than the disk size available (e.g., the data file for São Paulo is 33GB).

Although the disk space available is only 14GB, public Github-hosted runners are using Azure DS2_v2 virtual machines, featuring a 84GB OS disk at the time of writing. In principle, it should be possible to remove unwanted preinstalled software and make additional space available for our data acquisition pipeline. To maximize the available disk space, we can rely on a GitHub action that increases the disk space up to 50GB.[3] It is now possible to download the CSV data by state.

To read the large files efficiently in R (R Core Team, 2020), a typical approach is to use the function `fread` from the package `data.table` (Dowle & Srinivasan, 2021). One issue is that `fread` always has to map the entire file to memory[4], which is too limited (7GB) with respect to the size of the data. To reduce the size of the data on disk, the `awk` Linux utility is used to extract only three columns from the original CSV file: date of vaccination, municipality of residence, and type of vaccination dose. This reduces the size of the data file by a factor of 10, which is enough to finally load the data into memory.

| Requirements | Adopted Solution |
|---|---|
| High-speed internet connection | Run workflows on GitHub-hosted cloud runners. |
| Large disk size | Maximize available disk space for build tasks with Github actions. |
| Reading large CSV files efficiently | Use the `data.table` R package for fast read/write. |
| Avoid memory leak | Use the `awk` utility (Unix/Linux) to subset the data file before loading into memory. |

## Data processing

For each state, an independent workflow maximizes the disk space available, downloads the data, and loads the data into memory. Then, the data are aggregated in R to compute the number of vaccine doses administered for each date, municipality, and type of dose. The results are stored in compressed format and pushed to the GitHub repository. The size of the data is now less than 5MB for each file.

The final workflow takes care of loading the state-by-state data into memory and computes, for each municipality, the time-series of the total number of doses administered, total number of people who received at least one vaccine dose, and total number of people who received all doses prescribed by the vaccination protocol. These metrics allow journalists, researchers, policymakers, and the public to understand the evolution of the COVID-19 vaccination rollout (Mathieu et al., 2021).

## Code style

The code style is dictated by the principle of making the code easy to understand and inspect in order to promote bug reports, transparency, and trust. For this reason, the code is provided as a set of well-documented R scripts rather than a standard R package that would add an additional layer of complexity. In the same way, the data processing

[3] https://github.com/easimon/maximize-build-space

[4] https://github.com/Rdatatable/data.table/issues/3526

relies mainly on the package dplyr (Wickham, François, Henry, & Müller, 2020) rather than data.table. Although less efficient, dplyr provides an excellent interface that is intuitive to understand even for users with little to no knowledge in R programming.

# Results

The data extracted are made available in the form of lightweight, ready-to-use CSV files with the following structure:

| Field | Description |
|---|---|
| IBGE | 7 digits IBGE code to identify Brazilian municipalities |
| Municipio | The name of the municipality |
| Population | The total population (2021) |
| Date | Date in the format YYYY-MM-DD |
| TotalVaccinations | Total number of COVID-19 vaccination doses administered |
| PeopleVaccinated | Total number of people with at least one vaccine dose |
| PeopleFullyVaccinated | Total number of people that completed the vaccination cycle |

Interactive visualization of the latest data is available here.

# Conclusion

The software extracts, for each municipality, the time-series of administrated COVID-19 vaccine doses from the 130GB dataset published by the Ministry of Health of Brazil. All the code is open-source and relies exclusively on freely available resources to enhance transparency and trust by exposing, and never breaking, the direct link with the original data provider.

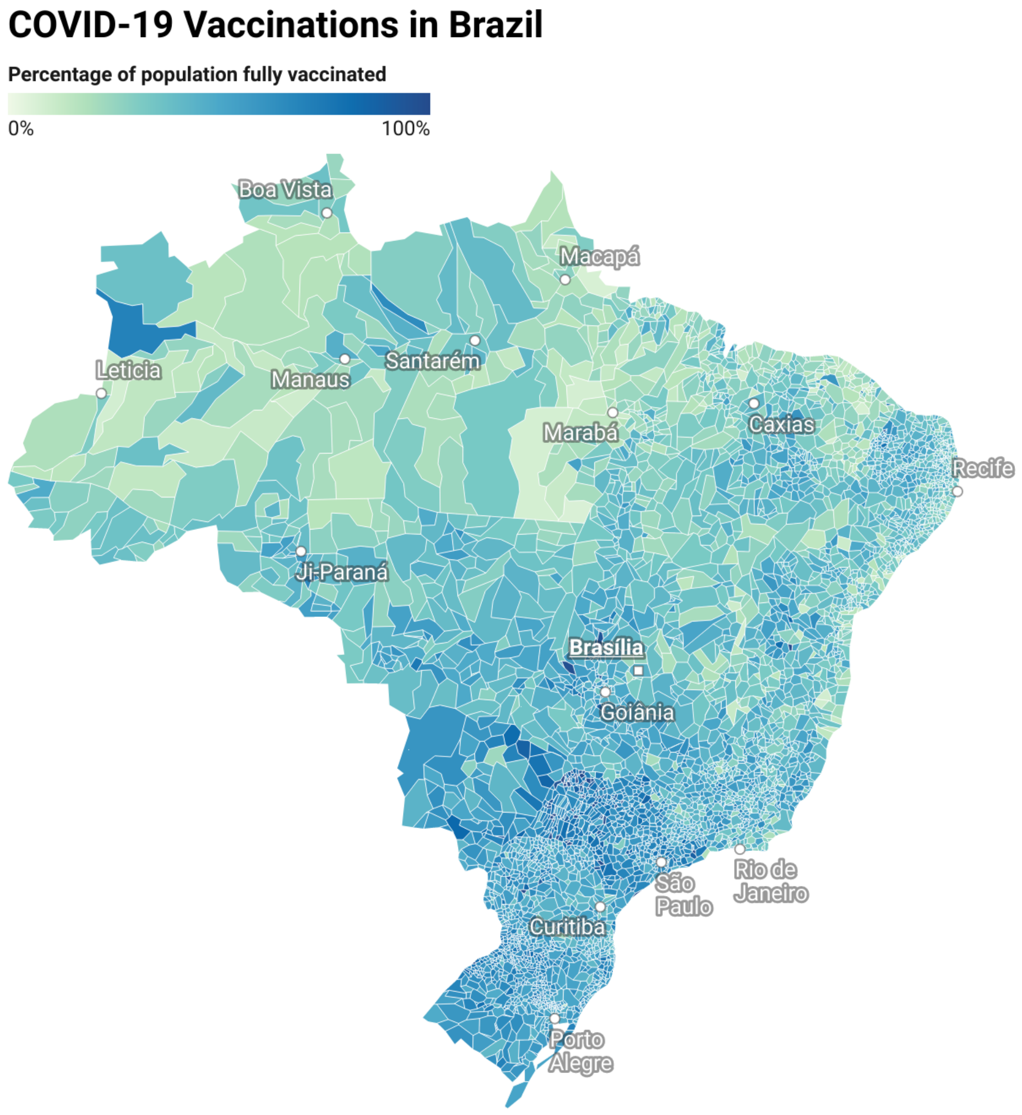
The data extracted are made available in the form of lightweight, ready-to-use CSV files that are updated daily and encourage re-use.[5]

By simplifying the access to the data, this resource aids policymakers and researchers in assessing the impact of the National Vaccination Campaign against Covid-19 in Brazil and in understanding its interactions with non-vaccination policy responses.

# License

The open data are released by the Ministry of Health of Brazil under the Creative Commons Attribution License. The software is released under the GPL-3.0 License.

[5] https://github.com/eguidotti/covid19br



Map: Emanuele Guidotti • Source: Ministério da Saúde • Created with Datawrapper

# References

Dong, E., Du, H., & Gardner, L. (2020). An interactive web-based dashboard to track covid-19 in real time. *The Lancet Infectious Diseases*, *20*(5), 533–534. doi:10.1016/S1473-3099(20)30120-1

Dowle, M., & Srinivasan, A. (2021). *Data.table: Extension of 'data.frame'*. Retrieved from https://CRAN.R-project.org/package=data.table

Guidotti, E., & Ardia, D. (2020). COVID-19 data hub. *Journal of Open Source Software*, *5*(51), 2376.

Hale, T., Webster, S., Petherick, A., Phillips, T., & Kira, B. (2020). Oxford covid-19 government response tracker. *Blavatnik School of Government*.

Hasell, J., Mathieu, E., Beltekian, D., Macdonald, B., Giattino, C., Ortiz-Ospina, E., Roser, M., et al. (2020). A cross-country database of covid-19 testing. *Scientific Data*, *7*(1), 1–7.

Mathieu, E., Ritchie, H., Ortiz-Ospina, E., Roser, M., Hasell, J., Appel, C., Giattino, C., et al. (2021). A global database of covid-19 vaccinations. *Nature Human Behaviour*, 1–7.

R Core Team. (2020). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

Wang, L. L., Lo, K., Chandrasekhar, Y., Reas, R., Yang, J., Eide, D., Funk, K., et al. (2020). CORD-19: The covid-19 open research dataset. *arXiv preprint arXiv:2004.10706*.

Wickham, H. (2014). Tidy data. *Journal of Statistical Software*, *59*(1), 1–23.

Wickham, H., François, R., Henry, L., & Müller, K. (2020). *Dplyr: A grammar of data manipulation*. Retrieved from https://CRAN.R-project.org/package=dplyr