pRycolletion: Diverse datasets from Paraguay

Andreas Schneider

Abstract The main goal of pRycollection is to provide datasets about Paraguay for research and teaching that are not easily found or accessible. Data and metadata often live separate lives, making data analysis difficult. The data package was build from the beginning with FAIR principles in mind. FAIR stands for Findable, Accessible, Interoperable, and Reusable. These principles are critical to maximizing the impact and value of data in research and practice.

Palabras clave: Paraguay - data - datasets - FAIR

Introduction

Data for Paraguay is generally not always public or easily available, for example, online or via an API. Motivated by the global reappraisals of the impact of the COVID-19 pandemic on various areas of society, it became clear that obtaining data, and in particular more detailed data, was difficult. This makes research in general, and in particular for local researchers, difficult and is probably the main reason way Paraguay as a country is often not included in international research. The datasets are a valuable resource since it is difficult to get hands on these data. Furthermore, metadata can provide much more context than they usually do, because often data and their respective meta data live in separate places. This can make data analysis more difficult. The main goal of pRycollection is to provide an additional resource of datasets about Paraguay for research and teaching that are not easily accessible. pRycollection is a combination of the 3-letter code for Paraguay - PRY - and the word collection. The 3-letter code was chosen to avoid possible confusion with the programming language Python (py). This datapackage is a living project, i.e. data will be added as soon as they become available.

Methods

Most datasets were obtained from public administrative sources through a freedom of information request. After cleaning the data all datasets were enriched with meta data using the frictionless package (Desmet et al., 2025) and hosted on Zenodo. All individual datasets were then packed in pRycollection with additional meta data using the datasets package (Antal, 2024, 2025).

Instalation

You can install the development version of pRycollection from GitHub with:

```r
# install.packages("pak")
pak::pak("schneiderpy/pRycollection")

# load pRycollection
library(pRycollection)
```

You can find the raw data on Zenodo.

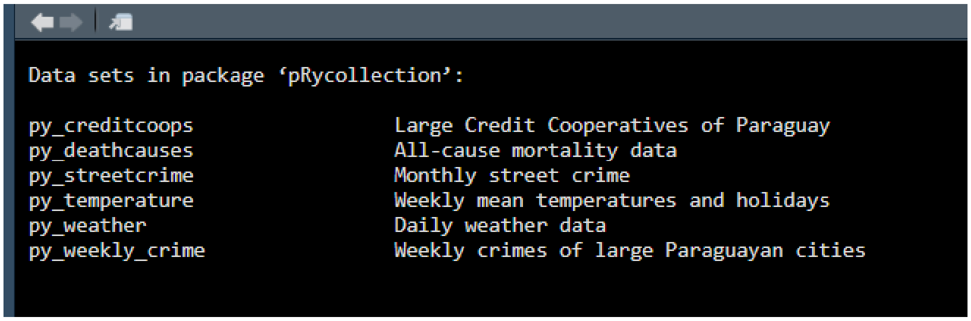
About the data

The pRycollection data package was build from the beginning with **FAIR** principles in mind. **FAIR** stands for **F**indable, **A**ccessible, **I**nteroperable and **R**eusable. These principles are critical to maximizing the impact and value of data in research and practice.

To see what datasets are included in the package load the pRycollection data package and the dataset package (to access metadata). Then type the following code line:

```r
data(package = "pRycollection")
```

This will open a new tab in your source pane listing all available datasets.



```
Data sets in package ‘pRycollection’:

py_creditcoops              Large Credit Cooperatives of Paraguay
py_deathcauses              All-cause mortality data
py_streetcrime              Monthly street crime
py_temperature              Weekly mean temperatures and holidays
py_weather                  Daily weather data
py_weekly_crime             Weekly crimes of large Paraguayan cities
```

Figure 1: All available datasets.

Examples

This is a basic example which shows you how to use pRycollection. Let's use the py_temperature dataset and print a summary of chosen dataset.

```
summary(py_temperature)
#> Schneider (2025): Summary of Weekly mean temperature data [dataset], https://doi.org/10.5281/zenodo.16729963
#>
#> Country name
#> Country ISO code
#> Mean temperature (degrees Celsius)
#> Holiday indicator
#>     rowid             country              ISO                 city
#>  Length:1565        Length:1565        Length:1565        Min.   :1
#>  Class :character   Class :character   Class :character   1st Qu.:2
#>  Mode  :character   Mode  :character   Mode  :character   Median :3
#>                                                           Mean   :3
#>                                                           3rd Qu.:4
#>                                                           Max.   :5
#>       week                avg_temp          holiday
#>  Min.   :2016-01-04   Min.   : 9.329   Min.   :0.0000
#>  1st Qu.:2017-07-03   1st Qu.:20.043   1st Qu.:0.0000
#>  Median :2018-12-31   Median :24.214   Median :0.0000
#>  Mean   :2018-12-31   Mean   :23.280   Mean   :0.1885
#>  3rd Qu.:2020-06-29   3rd Qu.:26.529   3rd Qu.:0.0000
#>  Max.   :2021-12-27   Max.   :32.000   Max.   :1.0000
```

You can already see that the summary() function produces a lot more additional metadata, such as author, year, title of the dataset, and variable labels and unit of measure along with the standard summary output. Let us look at the first six rows of the dataset.

```
head(py_temperature)
#> Schneider (2025): Weekly mean temperature data [dataset], https://doi.org/10.5281/zenodo.16729963
#>   rowid     country   ISO       city         week       avg_temp  holiday
#>   <defined> <defined> <defined> <defined>    <dttm_dfn> <defined> <defined>
#> 1 obs:1     Paraguay  PY        1 [Asuncion] 2016-01-04 27.8      0
#> 2 obs:2     Paraguay  PY        1 [Asuncion] 2016-01-11 30.3      0
#> 3 obs:3     Paraguay  PY        1 [Asuncion] 2016-01-18 29.9      0
#> 4 obs:4     Paraguay  PY        1 [Asuncion] 2016-01-25 27.3      1
#> 5 obs:5     Paraguay  PY        1 [Asuncion] 2016-02-01 26.6      0
#> 6 obs:6     Paraguay  PY        1 [Asuncion] 2016-02-08 30.1      0
```

You see that all variables have a defined class instead of the usual chr int or dbl. This means that they have additional meta data.

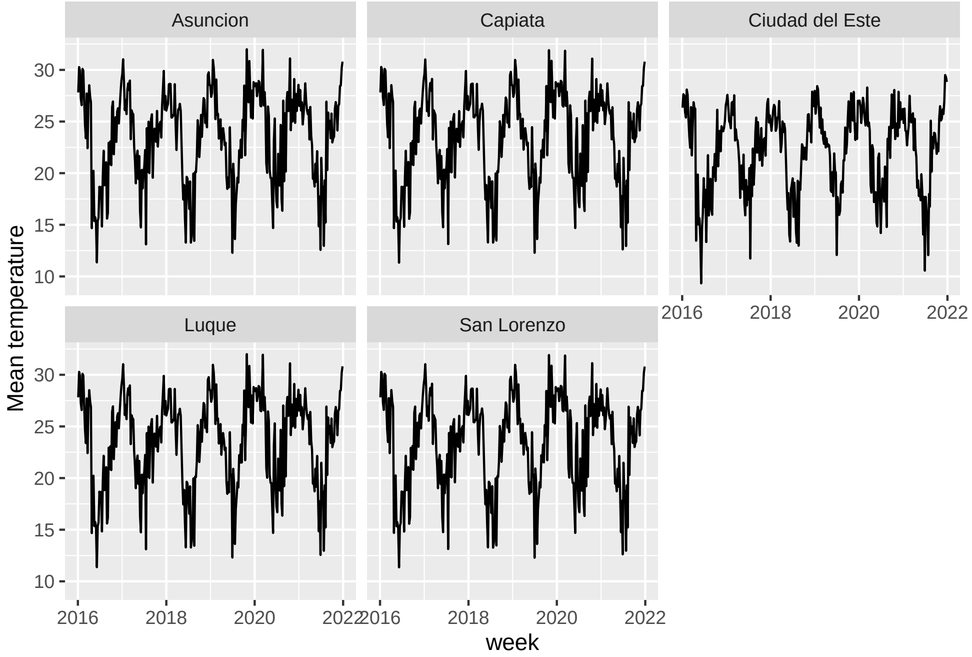
Let us see this in more detail:

```
var_label(py_temperature$country)
#> [1] "Country name"
```

```
var_label(py_temperature$avg_temp)
#> [1] "Mean temperature"
```

```
var_unit(py_temperature$avg_temp)
#> [1] "degrees Celsius"
```

Let us explore some more examples: We stay with the `py_temperature` dataset, which has 7 variables and 1565 weeks of temperature data for the cities of Asuncion, Capiata, Ciudad del Este, Luque, San Lorenzo. We could make, for example, a facet plot of the average weekly temperature (y-axes) for each city (x-axes) in the dataset.



For more information or examples you can visit the dedicated website

References

- Antal D, (2025). dataset: Create Data Frames that are Easier to Exchange and Reuse. R package version 0.3.99. https://dataset.dataobservatory.eu/
- Antal D, (2024). The dataset R Package: Create Data Frames that are Easier to Exchange and Reuse. R{R package version 0.3.4. https://doi.org/10.32614/CRAN.package.dataset
- Desmet P, Oldoni D, Huybrechts P, Govaert S (2025). frictionless: Read and Write Frictionless Data Packages. R package version 1.2.1.9000, https://github.com/frictionlessdata/frictionless-r.