Integrating data and analysis: On bridging data publishers and computational environments

Markus Stocker[1,2], Uwe Schindler[2], Robert Huber[2]

[1]*TIB Leibniz Information Centre for Science and Technology, Welfengarten 1 B, 30167 Hannover, Germany*;
[2]*PANGAEA Data Publisher for Earth & Environmental Science, MARUM Center for Marine Environmental Sciences, University of Bremen, Leobener Str. 8, 28359 Bremen, Germany*

Corresponding author(s) e-mail: markus.stocker@tib.eu

ABSTRACT:

Prior to analysing data, researchers today need to perform the 'janitorial' step of the data life cycle. This step involves cleaning, harmonizing, or integrating data and typically relies on loading data from one or multiple sources into a computational environment and one of its native data structures. Loading data consumes only a small fraction of the estimated 80% of time consumed by the 'janitorial' step overall in data analysis. Yet, it is baffling how much effort it can take to load data into a native data structure of a computational environment.

What could arguably be as straightforward as providing a DOI to a specialized function that returns the corresponding data (and metadata) represented in a data structure native to the computational environment in reality generally amounts to resolving the DOI using a browser, navigating a landing page to identify data and metadata, download data to a file, and ultimately load the data from the file using one of several specialized functions that read data in one of many file formats. The matter is further complicated by Web APIs that - while easing access and download - generally require prior knowledge for how to retrieve data. Such knowledge needs to be encoded in programming code using the computational environment of choice. Surely the required pieces of technology exist to directly access data given a DOI and negotiate content between data provider and consumer so that the computational environment can automatically load data into a native data structure. Yet we still have some way to go before the subtask of loading data into a computational environment is truly easy.

Using PANGAEA as a data publisher and a couple of other data sources, and Jupyter as a computational environment, in this talk we highlight the problem and delineate a solution. Specifically, we will demonstrate how, given a DOI name, PANGAEA data can be automatically loaded into a Python Data Analysis Library (pandas) DataFrame with a mere call of a specialized function. We will also discuss some of the challenges and implications of performing such operation on Linked Data. While the prototype does not do justice to the complexity of generalizing the implementation over heterogeneous data sources and data types, we argue the talk contributes to improving how a minor but necessary subtask of the data life cycle may be executed in computational environments, and thus contribute to seamless integration of data and analysis.

KEYWORDS: Computational Environments, Data Publishers, Data Analysis, Integration, Linked Data