A Data Restore Model for Reproducibility in Computational Statistics

Daniel Bahls
Leibniz Information Centre for Economics (ZBW)
Neuer Jungfernstieg 21
20354 Hamburg, Germany
d.bahls@zbw.eu

Benjamin Zapilko
GESIS - Leibniz Institute for the Social Sciences
Unter Sachsenhausen 6-8
50667 Cologne, Germany
benjamin.zapilko@gesis.org

Klaus Tochtermann
Leibniz Information Centre for Economics (ZBW)
Düsternbrooker Weg 120
24105 Kiel, Germany
k.tochtermann@zbw.eu

ABSTRACT

Researchers are more and more requested to publish their scientific data sets for purposes of transparency, re-use, and reproducibility. Particularly in economics and the social sciences, researchers often use sensitive statistical data that underlie protection policies which inhibit distribution to third party archives. In addition, a considerable quantity of data sets combines data from one or more external providers, which complicates the setting for curation-related activities. These circumstances give us reason to pursue a *data restore model* on the basis of fine-grained referencing that allows to trace data provenance to the original archive in charge of curation. One goal is to enable data publication in difficult cases, and another one is to show how the gaps between data citation and code integration can be closed in order to eliminate all manual efforts of arranging code and data files for reproduction attempts. On this basis we develop the requirements for a data restore model and elaborate a generic design in view of an overall data management infrastructure. We further explore an experimental implementation which we validate by taking the example of a real-world publication in economics. Eventually we close with the vision of a *data and code ontology* that carries statistical models from paper to a re-usable semantic level.

Categories and Subject Descriptors

H.3.5 [**Online Information Services**]: Data sharing; H.3.7 [**Digital Libraries**]: Miscellaneous; I.2.4 [**Knowledge Representation Formalisms and Methods**]: Semantic networks

General Terms

Design, Economics, Algorithms, Verification

i-Know '13, September 04 - 06 2013, Graz, Austria

Keywords

Research Data Management, Semantic Digital Data Library, Linked Data, Statistics

1. INTRODUCTION

Repeatability of research results is a fundamental criterion in science. Experiments must lead to the same findings when conducted by other persons or in other environments, under the premise of keeping relevant and considered parameters constant. In computational research, this remains unverified in many cases due to various reasons [25]. One of them is the plain fact that the supporting data is often unavailable to others which gives rise for a more organized and open data management practice [32]. While a scientific publication is commonly understood as synonymous with a peer-reviewed article alone, program code and data take the role of nice-to-have additional material only that rarely find their way to the public.

Transparency and verification of results not only are gestures of good scientific practice but essential if the statements published in the name of science are to reflect profound and solid knowledge. The problem might be less significant in technical domains where relevant theories and best practices would be validated eventually as soon as they are applied. Decision makers in public sector and industry, however, often rely on surveys and studies which do not necessarily have to stand proof when made use of. The case of a research study on cultural integration of immigrants in the UK shows that inconsistencies occur in the social sciences as well. The data was available to the community [7] which allowed other researchers to investigate and after all heavily criticize their publication as its results could not be reproduced [2]. To underscore their critique, they published the source code of their own analyses online and in this way called for clarification. The original authors responded with an errata in which they acknowledged the faults. At the same time, they presented a modified version of the statistical model they had applied and managed to produce the previous findings again while admitting that “the results are now less clear-cut and [they lost] statistical significance” [8].

Data availability was the key for verification in the example above. Yet, when it comes to analyses and the methods applied, textual reports typically lack many of the algorithmic details needed for reproduction or re-use after all. Altogether, the media type itself seems rather unsuitable for this purpose, which is why there are many voices in the scientific community promoting the idea of program code to be included as substantial artefacts along with the article. In this sense, some researchers even view source code as the main contribution. This viewpoint is often referred to as the *Claerbout principle*: "An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures" [1] [11]. In support of this view [20] investigated reproducibility in econometrics and pointed out how software tools can be used to organize data, code, and documentation in this domain. As part of a replication study they have also shown that important details are missing in the two example papers examined and filled in the blanks by publishing source code and data of the eventually successful replication attempts.

Both, data and code resources make a valuable contribution if findings are to be reproducible and re-usable in computational research. And yet, there is more to be considered. A comprehensive review of the *JMCB*[1] *data+code archive* revealed that provenance information for the data used is often incomplete and that submitted code does not reproduce the results either [23]. It is not uncommon that even the original authors have difficulties understanding the details of their own work only few months after publishing already. Apparently, these issues are faced in many other areas of computational research too [18] [16] [31] [10] which can be understood as a problem of insufficient documentation.

With the increasing use of statistics software [14] these problems emerge in empirical research in general where more and more statisticians promote the use of literate programming techniques [22] [29] which combine report text and coding in single executable scripts in a way that clarifies the link between research code, the figures produced, and the findings naturally. In both of the examples above, the authors made use of the literate programming tool Sweave [21] so that their reports can be generated again, from data processing to generating the report document, under the premise that the data is given and the system environment allows for correct execution.

Empirical research is often based on sensitive data that cannot be shared easily. One solution to this problem is to transform such data into anonymized public-use files that can be distributed without infringements of data protection. But doing so requires careful verification in terms of cell sizes and de-anonymization and requires additional efforts, often on the researchers' side. Moreover, researchers hesitate to publish their own data if incentives are not clear, especially when they had a lot of work producing it. In addition, data is often obtained from several sources and the mere citation of them does not give any details on the file formats used or way of composition as used as input for the computer program.

In this paper we aim to overcome these challenges for the domain of statistical data and elaborate on a holistic solution to enable the exact reconstruction of individual research data sets as used originally by a researcher. We pursue the idea of a *data restore model* that is based on fine-grained data referencing as described in earlier works [3] where further background and motivation is explained more thoroughly. Since this referencing technique can be viewed as means for precise citation, the linkup to provenance and further documentation on the particular data version is given which again contributes to clarity. The idea is to equip researchers with means that make full transparency possible even in difficult settings of protected and distributed data sources, so that reconstruction remains a question of access permission only. The restore model is designed to bridge the gaps between data citation and program code and to seamlessly integrate with preferred research tools, so that no further manual work is required in setting up data and code resources for replication attempts. Eventually, the goal is to provide researchers with the ability to close all missing links from data sources and processing to figure plotting and report generation as to create trust in their findings which from another perspective could be an incentive to do so. On the basis of this data restore model, we elaborate the requirements it carries on the entire information infrastructure between researchers, archives, and libraries.

The remainder of the paper is structured as follows. In Section 2 we discuss related work relevant to our approach. The requirements for a data restore model are elaborated in Section 3 by which we develop a generic solution. An experimental implementation is presented in Section 4 where we explore its application in context of a case study from economics. Finally, we conclude in Section 5 with summary and outlook on future work and present a vision that illustrates the potential of this approach .

2. RELATED WORK

The problem of reproducibility in computational research has been addressed by many. If re-execution of experiments is the goal, it is necessary to capture information on the entire system environment used in order to make sure respective processes can be performed under the same parameters [27] [28] . This alone is a very hard problem already, because the chain of dependencies in terms of hardware and software has to be captured and restored to the very detail. In this regard, [30] illustrates how the problem of software dependencies for computational research could be solved with the help of build scripts such as make files. Nevertheless, the achievements in this area seem promising, and the opportunities for application such as [24] are great. As the goal of our research is to find means for precise reconstruction of data sets and direct integration with program code, these approaches make a good complement.

Particularly for the domain of empirical research, we found the literate programming approach very appealing as it seems light-weight and suitable for the many, less complex scenarios. It interweaves source code and text for the report, so that all processing steps are already quite well-documented. Tools like Sweave [21] or Knitr[2] are already accepted as helpful means to reproduce analyses and document its steps [22] [29] and therefore suitable for the reproduction of research results.

Source code for computational research does not need to be executable in order to make a valuable contribution to transparency as it is a detailed documentation of the processing and a valuable resource as such, which is part of the argumentation in [13]. The Verifiable Computational Research community aims to collect data, code and intermedi-

[1] Journal of Money, Credit, and Banking

[2] `http://yihui.name/knitr/`

ate results of research processes as a trace to make findings verifiable. Their automatic capturing process is integrated with the researchers' workflow, and hence successful as well as unsuccessful attempts can be reviewed eventually. We share their view in a pragmatic sense. The code does not need to be executable, if it is, the better. Either way, source code should be picked up as part of the publication material whenever possible [5][3].

One of the main problems in empirical research, however, is that a large amount of the data used is sensitive and hence protected, or underlies certain usage rights due to commercial use. Therefore the problem of data availability cannot be solved as the legal situation simply prohibits the further distribution to other parties. But recently, as a result of the Open Data movement a large degree of data sets has been made publicly available by governments, statistics offices and research organizations. This data can be used for research, but is not always citable or referencible in a way it is required for scientific publications like using persistent identifiers. The publictaion of such data as Linked Open Data [4] allows a precise identification and recerence of single entities such as data sets or particular indicators. Furthermore, by using the Linked Science Ontology [19] abstract research processes can be modelled and represented. But, this ontology does not address the issues of data access and availability.

3. THE DATA RESTORE MODEL

We begin this section by developing the requirements for a data restore model. Subsequently, we elaborate a generic design and discuss its implication on an overall data mangement infrastructure and the workflows involved.

3.1 Requirements

In the following, we want to develop the ideas of a referencing technique and take into account the recommendations fromulated by [23] which we refer to using parentheses *(A-J)*. Whichever data management solution is applied eventually should fit to the researchers' workflows and practices in order to be effective and thus accepted [12]. In the ideal case, researchers can use their preferred tools with no further hassle preparing anything at the time of data submission. On the side of reuse, there should not be any efforts in composing and organizing these resources either, meaning that no puzzling and manual tinkering in organizing data and code is required to make it run *(F partly)*. Location and structure of data files on the local file system should be inherently specified, so that it integrates perfectly with the code. [5]

There are plenty of prevailing file formats, ranging from CSV, XML, Excel, entire RDBMS dumps to specific software-dependent and even unknown individual "raw" types of formats. McCullough et al. recommend the use of ASCII encoding *(D)* for the benefit of re-usability, we therefore want to exclude binary formats from our discussion. As to be less restrictive and support more contemporary encodings such as UTF-8, we want to pursue the idea of self-describing documents with information on its own encoding contained.[6] A format-independent solution should be found whereas most importantly, any restored data set must equal the original as to perfectly integrate with the code in the end *(E)*. Thus, we note that data citation by persistent identifier (as is the current practice in data publishing) is not sufficient for this purpose, since these identifiers point to data sets on a more abstract level only and do not distinguish or specify a particular format. [7]

Further, the data restore model should be applicable for protected data as well *(supports G)*. This might sound surprising to the reader as sensitive data cannot be shared per se and a restore concept appears redundant. Nevertheless, we want to make the point that the spectrum from open to closed data is gradual, since some data is access-restricted only and can be obtained under certain policies and regulations. Whether or not a data set can be handed out should be a question on legal level only, the restore technique itself, however, should cater for these cases as to give means for repeatability in every scenario.

Either way, pointers to the provenance of data should be available that give documentation and further background such as contact information for any version being used. While new persistent identifiers are commonly given for major data updates and releases only, those versions with minor bugfixes cannot be referred to with precision so far.[8] The use of versioning systems could help identify individual releases at detailed level [20] which should integrate seamlessly with the data restore model. Eventually, references to service points providing this information should be given along with the submission material. This could further be realized with Web Services that accept requests for information or data content on a technically fully supported level, so that up-to-date user assistance can be provided directly within the tools used in a research environment *(B)*. Depending on regulations, the actual data content identified by a given model instance could be transferred on request, possibly using authentication mechanisms, so that the data set can be restored locally using the original data sources. [9]

In contrast to the copy-based approach, the referencing technique allows data and documentation to be maintained at one responsible archive only, with the result that no duplicate efforts have to be made in terms of data curation which is particularly relevant in cases where (several) external sources were used. This improves quality of service as all enquiries about data can be answered with the highest possible level of expertise available and gives a clear view on responsibilities in addition.

As a result of this discussion we formulate the following requirements for the data restore model:

[3] `http://sciencecodemanifesto.org/`

[4] http://linkeddata.org/

[5] In this regard, software and system specifications are as important. This kind of documentation can be organized independent of the issues addressed in this paper, we therefore do not further address it here.

[6] Otherwise, data would need to be converted back and forth which causes trouble whenever a character has no representation in the ASCII standard

[7] Most data agencies offer download in various formats while the content is filed under the same persistent identifier.

[8] Surely, an author can give a note on the time of data download or acquisition in the paper. This is rarely practiced, and data reconstruction in this case again involves vagueness and human efforts.

[9] Authentication can be realized using SSL for example. If transfer via internet is considered too insecure, the API can be used to collect data requests as such and to initiate the procedure according to the respective policies.

1. **Equality:** A data file restored by the model must equal its original on character level[10], so that it integrates perfectly with the software and program code used by the researcher.

2. **Protection:** The model must be applicable for protected data (closed data) as well, in strict accordance with secured protection.

3. **Curation:** Using the model, data maintenance and curation needs to be done for the original copy only, right at the responsible archive, whereas documentation can be distributed efficiently on the basis of harvesting techniques.

4. **Usability:** The method should integrate seamlessly with the researchers' workflows and comply with their preferred tools.

5. **Formats:** The method should be sustainable and applicable to common, alternative, unknown and future data formats.

6. **Traceability:** The technique itself must be traceable, so that references to data content can be followed efficiently, by humans and machines.

7. **API:** The data restore model should include pointers to Web Services that provide documentation, contact information or, if applicable, the actual data content upon request.

3.2 Designing the model

The requirements can be realized in several ways. However, giving freedom to researchers to use or reuse data in their preferred environments and associated formats needs careful consideration. Data could be offered in a variety of formats, each version provided with means for persistent identification, so that data citation as well as seamless code integration is ensured. But this approach requires program code to function on complete data sets, which impairs usability considering that researchers typically select slices of indicators and observations[11] for input to their programs. Further, the approach would lead to multiple persistent identifiers for content-wise identical data and complicate bibliometric measures. These conditions suggest the use of a primary data representation from which all other formats can be derived whereas one referencing method only is needed to serve all scenarios. As a consequence, such model must give means for identification of any possible data slice from global scope. Pointers to respective repositories in combination with structured queries (such as SQL or SPARQL) could be a solution. A challenge in this approach, however, is that such queries produce result sets in the shape of tables which yet need to be transformed into this unspecified format as used by a researcher. In the following, we elaborate a model that aims to overcome these challenges.

The model has to make sure that individual data slices and values can be identified without ambiguity in the long term. As we could not find an established data model for statistical data that is not subject to change and incorporates versioning at the same time, we discarded model-based queries and decided to use identifiers for every single value of a data set. The number of IDs needed is certainly high and questions of scalability raise immediately. On the other side, the IDs we are going to introduce can as well be interpreted as single-result queries or as the variables of a regular query. However, we would like to have a discussion on this naïve approach and leave the investigation of scalability for now. We will introduce the concept of a *data template* which is based on the idea to replace every value within a data file with its associated ID, so that the original file can be restored again simply by replacing them back with its associated values. This could also help solve some of the problems in using linked data for scientific purposes [6][12], which will be the basis of the model.

With the ongoing linked open data movement, awareness and knowledge about semantic technologies and the RDF data model has found its way to many application domains, among which the domain of statistical data[17]. These technologies break down the separation between black box data bundles and attached metadata as they interweave data with its documentation. The data content is more accessible and interpretable to machines which sets the stage for a variety of new applications, e.g. for data retrieval [3].

3.3 Workflows

If implemented in big scale as a holistic information infrastructure for research data management in statistics, the presented approach has further implications on workflows and gives ideas for a big picture. Data archives could offer the download of data templates along with the actual data, so that researchers have the precise references right from the beginning of their research. Alternatively, researchers could pull the data on the basis of these data templates alone from within their research tool, provided that a respective plugin is available. The advantage in the latter case is that this procedure could be embedded within their program code and no further modifications on data and code need to be done at the time of publishing. The researcher simply submits code and data templates, and other researchers can rerun the same analysis without any manual arrangements or collection of resources, simply by executing the code.

In case the researcher used own data, the data would have to be submitted to an affiliated archive[13] in beforehand where the respective data template and IDs involved are generated first. At this point, metadata on provenance and other documentation should be collected which can be supported through scientific workflow systems[9]. The archive returns the template to the researcher who can then publish it in the same way as described above. This procedure must be supported with tools for mapping and ID generation. Furthermore, these tools should provide for further semantic annotation of the content (using the RDF Data Cube Vocabulary for example) so that documentation about the data is available as well. Figure 1 outlines the overall process.

[10] Depending on character encoding and operating system, the files may differ on bitstream level while the textual content is equal.

[11] e.g. selection of certain indicators and reference periods from large time series data

[12] The problem of precise data references has been recognized by the Semantic Statistics community as well `http://datalift.org/en/event/semstats2013`.

[13] Researchers have to archive their data, either locally or at another archive. By ”affiliated archive” we denote this repository.

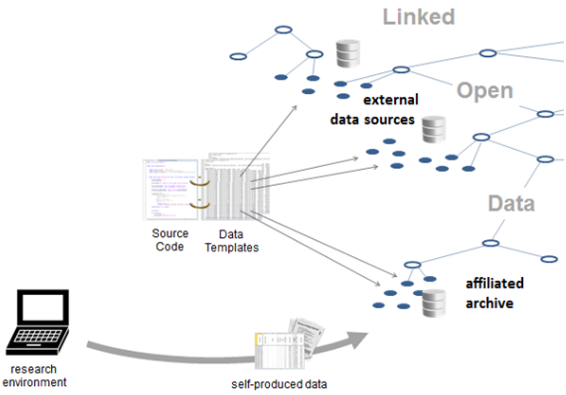


Figure 1: Using the data restore model

4. EXPERIMENTAL IMPLEMENTATION

In this section, we present a prototype implementation of the data restore model and a plugin called *ddocks*[14] for the computing environment R. Since literate programming already is a prominent approach to replicability in the computational statistics community [22] (particularly supports recommendation *(A)*) we want to demonstrate how the plugin can further strengthen this concept.

In the following, we explain the concept of data templates by taking the example of two data sets from the sources of [20]. A description of the *ddocks* plugin implementation follows thereafter, and a basic validation is given at the end of this section which we have incorporated as part of the Sweave sources for the article the reader is currently looking at. The source code is available on GitHub, and respective links are given at the bottom of each subsection.

4.1 Generating the Data Templates

The data template contains a header section in which we can specify encoding and tokens that are used to identify IDs within the template. In addition, we also declare Web Service addresses for every namespace used (which is only one for our example) under the pragmatic assumption that every namespace can be associated with one archive providing one such Web Service for data enquiries.

We used a simple script that scans any given input text file for contained values using a regular expression which was specially designed for the data of our example. For every identified value a static counter variable increments by one which is used to generate the value ID. We could have used the plain counter value as such, but we decided to hash it only to highlight that these IDs need no meaningful structure at all. In combination with the examplary archive namespace *http://demoarchive.demo/data/* we obtain full URIs for all values, making them referenceable worldwide from any context. To obtain the *body* of a data template, all values are replaced with their abbreviated URIs and surrounded with angle brackets for parsing purposes. A *header* section is prepended to this part which we start with an XML-like notation that declares character encoding and *ddocks* version as to give basic information about how to interpret the contents of this file. The *ddocks-header* tag opens the actual header section in which the namespaces used are listed with their associated URI prefix and repository API for which the prototype expects the address of a SPARQL endpoint. It further declares the parse tokens used to identify IDs in the template body, so that clashes with format-specific syntax can be avoided. Eventually, we append the body section right after the newline character behind the close tag of the header section to complete the data template. Figure 2 shows an excerpt from one of the two data templates we have created.

The script also produces an RDF document that contains the mapping from the URIs to their associated values. The simple ontology behind consists of one class only (*DataReference*) and all its instances (value IDs) map to a literal (value) using the *rdf:value* relation. Finally, we added these triples to an experimental triple store at GESIS that is presently open to the public. Please note that no further semantics were modelled as we wanted to demonstrate the minimal efforts necessary to follow the presented approach.

`https://github.com/dbahls/ddocks-utils.git`

4.2 The R-Plugin

The signature of *ddocks_restore*, the only function offered by the plugin, expects two parameters. The first parameter is a connection object from which the data template will be read. The second parameter specifies the target file the restored content will be saved to.

First, *ddocks* interprets the information given in the header. Using the declared open and close tokens, it collects all IDs occurring in the template body and organizes them by namespace which is the key to identify the respective data repository. Iterating over these namespaces, the values are queried from the associated SPARQL endpoint[15] and replace the IDs in the template body which is then written to the target file.

We run this procedure for the two data templates we have generated earlier

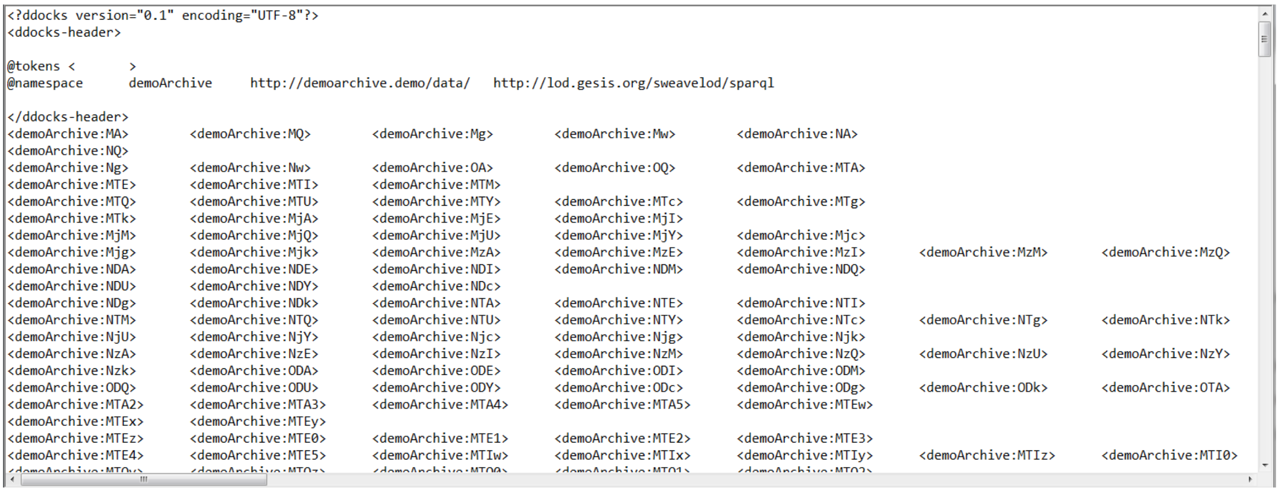
```
R> ddocks_restore(
+    file("./data-templates/data.dj.ddocks"),
+    targetfile="data.dj")

R> ddocks_restore(
+    file("./data-templates/rk.raw.ddocks"),
+    targetfile="rk.raw")
```

and reproduce the data files on the local file system. *ddocks* takes these files as a cache and skips when the restore command is run again, so that the data does not have to be reconstructed every time the code is executed. Researchers could embed these calls within their source code to load the data by template at the beginning of their analyses without being bothered again, and no further modifications are needed at the time of publishing.

`https://github.com/dbahls/ddocks.git`

[14] *ddocks* is short for *data docks* which is meant to reflect the “transshipment” character of the interface between data retrieval and processing.

[15] This is done in bundles so as not to query every single value separately. Then again, as to avoid oversized queries, these bundles are limited to a number of 30 IDs per query.



```
<?ddocks version="0.1" encoding="UTF-8"?>
<ddocks-header>

@tokens <       >
@namespace      demoArchive     http://demoarchive.demo/data/   http://lod.gesis.org/sweavelod/sparql

</ddocks-header>
<demoArchive:MA>        <demoArchive:MQ>        <demoArchive:Mg>        <demoArchive:Mw>        <demoArchive:NA>
<demoArchive:NQ>
<demoArchive:Ng>        <demoArchive:Nw>        <demoArchive:OA>        <demoArchive:OQ>        <demoArchive:MTA>
<demoArchive:MTE>       <demoArchive:MTI>       <demoArchive:MTM>
<demoArchive:MTQ>       <demoArchive:MTU>       <demoArchive:MTY>       <demoArchive:MTc>       <demoArchive:MTg>
<demoArchive:MTk>       <demoArchive:MjA>       <demoArchive:MjE>       <demoArchive:MjI>
<demoArchive:MjM>       <demoArchive:MjQ>       <demoArchive:MjU>       <demoArchive:MjY>       <demoArchive:Mjc>
<demoArchive:Mjg>       <demoArchive:Mjk>       <demoArchive:MzA>       <demoArchive:MzE>       <demoArchive:MzI>       <demoArchive:MzM>       <demoArchive:MzQ>
<demoArchive:NDA>       <demoArchive:NDE>       <demoArchive:NDI>       <demoArchive:NDM>       <demoArchive:NDQ>
<demoArchive:NDU>       <demoArchive:NDY>       <demoArchive:NDc>
<demoArchive:NDg>       <demoArchive:NDk>       <demoArchive:NTA>       <demoArchive:NTE>       <demoArchive:NTI>
<demoArchive:NTM>       <demoArchive:NTQ>       <demoArchive:NTU>       <demoArchive:NTY>       <demoArchive:NTc>       <demoArchive:NTg>       <demoArchive:NTk>
<demoArchive:NjU>       <demoArchive:NjY>       <demoArchive:Njc>       <demoArchive:Njg>       <demoArchive:Njk>
<demoArchive:NzA>       <demoArchive:NzE>       <demoArchive:NzI>       <demoArchive:NzM>       <demoArchive:NzQ>       <demoArchive:NzU>       <demoArchive:NzY>
<demoArchive:Nzk>       <demoArchive:ODA>       <demoArchive:ODE>       <demoArchive:ODI>       <demoArchive:ODM>
<demoArchive:ODQ>       <demoArchive:ODU>       <demoArchive:ODY>       <demoArchive:ODc>       <demoArchive:ODg>       <demoArchive:ODk>       <demoArchive:OTA>
<demoArchive:MTA2>      <demoArchive:MTA3>      <demoArchive:MTA4>      <demoArchive:MTA5>      <demoArchive:MTEw>
<demoArchive:MTEx>      <demoArchive:MTEy>
<demoArchive:MTEz>      <demoArchive:MTE0>      <demoArchive:MTE1>      <demoArchive:MTE2>      <demoArchive:MTE3>
<demoArchive:MTE4>      <demoArchive:MTE5>      <demoArchive:MTIw>      <demoArchive:MTIx>      <demoArchive:MTIy>      <demoArchive:MTIz>      <demoArchive:MTI0>
```

Figure 2: Data template for one of the two data sets (*rk.raw*)

4.3 Validation

We validate this experimental implementation with respect to the requirements developed in Section 3.1. As this approach is based on URIs, namespaces can be used to identify archives, and in combination with locally organized IDs every single value of any data set can be referenced from any context (*Traceability*). The requirement item *Curation* is addressed as well, because data can be referenced with precision and maintenance of every data can be done in one place. Since the use of RDF allows us to add an additional link between ID and value, we can separate the two and disclose all information about the data in very detail in machine-understandable form, and yet protect the values (*Protection*). Furthermore, when every value can be referred to from anywhere, a simple technique can be applied to enable format-independent reconstruction of data files for which we introduced the concept of a *data template*. Particularly the document-specific declaration of tokens enables a format-independent solution (*Formats*) as this flexibility allows us to avoid syntactical clashes with the particular data format used. We have used SPARQL endpoints as a generic implementation of the *API* which only served for the purpose of data retrieval. For actual application, a more elaborated API should be developed that could be based on Web Services.

Eventually, we have to show that the restored files equal their originals on character level (*Equality*). If the two expressions

```
R> all.equal(
+    readLines("data.dj"),
+    readLines("koenker-zeileis-09/data.dj"))

[1] TRUE

R> all.equal(
+    readLines("rk.raw"),
+    readLines("koenker-zeileis-09/rk.raw"))

[1] TRUE
```

evaluate to TRUE we have successfully validated the approach in terms of functionality.

We also inserted the two restore calls above in the source code of our example paper ([20]) right before the load-data calls respectively, and running Sweave on it produced a report identical to the published article.[16]

`https://github.com/dbahls/paper-restore-model.git`

5. SUMMARY, VISION AND OUTLOOK

The ideas presented in this work can be used to offer data and code behind research publications, which increases transparency, re-use, and reproducibility. We have formulated requirements on the basis of the recommendations in [23] and further followed the idea of fine-grained data referencing in our data restore model. The introduced concept of data templates uses precise pointers to the original data sources and can be disseminated freely as they do not disclose sensitive content. Since data files can be restored in the format used by the researcher, the template technique closes the gaps between data citation and program code integration. The *ddocks* prototype illustrates how the restore model can be applied in a way that enables a "one-click" solution for re-running analyses on other systems provided that users have permission to acquire the data and system dependencies are resolved. Especially if linked with research articles, the resources can be used in tutorials to teach computational research in practice, serve as re-usable research resources, or can be used within review-processes. With the precise identification of values, data provenance can be clarified comprehensively whereas responsibilities for maintenance and documentation of a data set are left to one archive respectively, so that harvesting techniques can be used efficiently to carry forward documentation, notes and updates about the data.

Assigning URIs to single data values sets the stage for integration with RDF which has potential for valuable applications. A semantic model could offer researchers to run same analyses on updated data with no efforts on data retrieval and composition. While most statistical models today exist on paper only, the current work on reproducibility allows to rerun it again on the original or updated data, and the vision is to make them re-usable for other data as well. A re-

[16] using R x64 2.15.3 on Windows

search publication could remain more relevant for extended periods of time if its predictive model on time series data for example can still be used on new records while the specific results of a published article may no longer be relevant to the community. Linked with thesauri, researchers could find these models by topic and re-use them in their own context while valuable user assistance can be given automatically if the semantic annotations specify clearly the kind of data it can process meaningfully.

Moreover, these models could serve as an input for the artificial intelligence community as they often describe measures for abstract concepts like *living standards* or *justice* which are typically hard to model. Although these models are and should always be debatable, the semantic linkage to statistical indicators gives information on aspects that are relevant for understanding these concepts. Further, predictive models could be combined in ensembles trained with machine learning techniques to further increase accuracy, which might again serve the computational statistics community in their research.

Considering the coupling of data or data references with code and the report text of a research publication eventually, the idea of a *data and code ontology* comes up that models the interconnection between data, code and research results in detail. Moreover, it could also be used to model algorithmic transitions from original to intermediate and result data sets, which would help clarify data provenance and allow for reproduction from open intermediate data sets while original data might be closed an unaccessible. We want to explore this scenario on the basis of [2] for which Sweave sources are available while the data set is access-restricted. Yet, scalability remains to be tested in regard to curation efforts and technical performance. We will investigate the design and development of a data and code ontology in future work for which we take into account recommendation *(C)* in [23] and other [20] [4] [15] [26] for the purpose of replicability in computational research.

6. REFERENCES

[1] Wavelets and Statistics. chapter Wavelab and reproducible research. Springer-Verlag, Berlin, New York, 1995.

[2] M. Arai, J. Karlsson, and M. Lundholm. On fragile grounds: A replication of ”are muslim immigrants different in terms of cultural integration?”. *Journal of the European Economic Association*, 9(5):1002–1011, 2011.

[3] D. Bahls and K. Tochtermann. Addressing the long tail in empirical research data management. In *Proceedings of the 12th International Conference on Knowledge Management and Knowledge Technologies*, i-KNOW '12, pages 19:1–19:8, New York, NY, USA, 2012. ACM.

[4] G. Baiocchi. Reproducible research in computational economics: guidelines, integrated approaches, and open source software. *Computational Economics*, 30(1):19–40, 2007.

[5] N. Barnes. The science code manifesto, 2012.

[6] S. Bechhofer, J. Ainsworth, J. Bhagat, I. Buchan, P. Couch, D. Cruickshank, D. D. Roure, M. Delderfield, I. Dunlop, M. Gamble, et al. Why linked data is not enough for scientists. In *e-Science (e-Science), 2010 IEEE Sixth International Conference on*, pages 300–307. IEEE, 2010.

[7] R. e. a. Berthoud. Fourth national survey of ethnic minorities, 1993-1994. 1997. DOI: 10.5255/UKDA-SN-3685-1.

[8] A. Bisin, E. Patacchini, T. Verdier, and Y. Zenou. Errata corrige:“are muslim immigrants different in terms of cultural integration?”. *Journal of the European Economic Association*, 9(5):1012–1019, 2011.

[9] P. C. Brauer and W. Hasselbring. Pubflow: provenance-aware workflows for research data publication. In *5th USENIX Workshop on the Theory and Practice of Provenance (TaPP '13)*, April 2013.

[10] P. Cassey and T. M. Blackburn. Reproducibility and repeatability in ecology. *BioScience*, 56(12):958–959, 2006.

[11] J. De Leeuw. Reproducible research. the bottom line. 2001.

[12] M. Feijen. What researchers want - a literature study of researchers' requirements with respect to storage and access to research data, February 2011.

[13] M. Gavish and D. Donoho. A universal identifier for computational results. *Procedia Computer Science*, 4:637–647, 2011.

[14] J. E. Gentle, W. K. Härdle, and Y. Mori. How computational statistics became the backbone of modern data science. SFB 649 Discussion Papers SFB649DP2011-020, Humboldt University, Collaborative Research Center 649, 2011.

[15] R. Gentleman and D. T. Lang. Statistical analyses and reproducible research. *Journal of Computational and Graphical Statistics*, 16(1):1–23, 2007.

[16] J. M. González-Barahona and G. Robles. On the reproducibility of empirical software engineering studies based on data retrieved from development repositories. *Empirical Software Engineering*, 17(1-2):75–89, 2012.

[17] W. Halb, Y. Raimond, and M. Hausenblas. Building Linked Data For Both Humans and Machines. In *WWW 2008 Workshop: Linked Data on the Web (LDOW2008)*, Beijing, China, 2008.

[18] T. Hothorn and F. Leisch. Case studies in reproducibility. *Briefings in Bioinformatics*, 12(3):288–300, 2011.

[19] T. Kauppinen, A. Baglatzi, and C. Keßler. Linked Science: Interconnecting Scientific Assets. In T. Critchlow and K. Kleese-Van Dam, editors, *Data Intensive Science*. CRC Press, USA, forthcoming 2012.

[20] R. Koenker and A. Zeileis. On Reproducible Econometric Research. pages 1–13, Oct. 2008.

[21] F. Leisch. Sweave: Dynamic generation of statistical reports using literate data analysis. In W. Härdle and B. Rönz, editors, *Compstat 2002 — Proceedings in Computational Statistics*, pages 575–580. Physica Verlag, Heidelberg, 2002. ISBN 3-7908-1517-9.

[22] F. Leisch, M. Eugster, and T. Hothorn. Executable papers for the r community: The r2 platform for reproducible research. *Procedia Computer Science*, 4:618–626, 2011.

[23] B. D. McCullough. Got replicability? the journal of money, credit and banking archive. *Econ Journal*

Watch, 4(3):326–337, September 2007.

[24] P. Nowakowski, E. Ciepiela, D. Harężlak, J. Kocot, M. Kasztelnik, T. Bartyński, J. Meizner, G. Dyk, and M. Malawski. The collage authoring environment. *Procedia Computer Science*, 4(0):608 – 617, 2011. <ce:title>Proceedings of the International Conference on Computational Science, ICCS 2011</ce:title>.

[25] R. D. Peng. Reproducible research in computational science. *Science*, 334(6060):1226–1227, 2011.

[26] H. Rahmandad and J. D. Sterman. Reporting guidelines for simulation-based research in social sciences. *System Dynamics Review*, 28(4):396–411, 2012.

[27] A. Rauber. Digital preservation in data-driven science: On the importance of process capture, preservation and validation. In *SDA*, pages 7–17, 2012.

[28] K. Rechert, D. von Suchodoletz, and R. Welte. Emulation based services in digital preservation. In *Proceedings of the 10th annual joint conference on Digital libraries*, JCDL '10, pages 365–368, New York, NY, USA, 2010. ACM.

[29] A. Rossini and F. Leisch. Literate statistical practice. 2003.

[30] M. Schwab, M. Karrenbach, and J. Claerbout. Making scientific computations reproducible. *Computing in Science and Engg.*, 2(6):61–67, Nov. 2000.

[31] P. Vandewalle, J. Kovacevic, and M. Vetterli. Reproducible research in signal processing. *Signal Processing Magazine, IEEE*, 26(3):37–47, 2009.

[32] J. Wood, T. Andersson, A. Bachem, C. Best, F. Genova, D. R. Lopez, W. Los, M. Marinucci, L. Romary, H. Van de Sompel, J. Vigen, P. Wittenburg, and D. Giaretta. *Riding the wave: How Europe can gain from the rising tide of scientific data*. European Union, 2010. Final report of the High Level Expert Group on Scientific Data: A submission to the European Commission.