Making Social Science More Reproducible by Encapsulating Access to Linked Data

Albert Meroño-Peñuela[1], Richard Zijdeman[2], Ashkan Ashkpour[2], and Rinke Hoekstra[3]

[1] Department of Computer Science, Vrije Universiteit Amsterdam, NL
{albert.merono,rinke.hoekstra}@vu.nl
[2] International Institute of Social History, Amsterdam, NL
richard.zijdeman@iisg.nl
[3] Elsevier, Amsterdam, NL r.hoekstra@elsevier.com

Abstract. Reproducibility of studies in social science history is problematic. Published papers do not contain enough information for a complete replication of the study: a primary principle of the scientific method. Typically, the first problem is the absence of a *link to the actual data* on which the study is based. While most times this link is included in the form of a dataset citation, a link pointing directly to the exact same data used originally is required for reliable replication. Imprecise data citation is especially problematic when data is volatile, or when the dataset comprises multiple sources, an increasingly common phenomenon in social science and history. Moreover, referring to "just the data" is not enough for reproducibility: it is necessary to point to the *sources* and *queries* (research questions) over these sources that generated the data. Not including this information means that standard methods such as regressions, correlations, and visualizations are no longer reproducible.
In this paper, we describe `grlc`, a method that enables the curation, versioning, publishing, sharing and replication of *queries* over collections of research data. `grlc` makes the results that answer historical questions *actionable* via a single, unique web address (URL). Furthermore, **the independence between queries and datasets allows the same research question to be asked to different data sources**, which also empowers generality of methods and pattern discovery. We argue that sharing these queries, along with their provenance and the meta-data needed for their execution, enables their universal reuse, thus facilitating the reproducibility of studies. To illustrate our approach, we describe a use case of `grlc` in social history studies.

Keywords: Replication, actionable historical data, Linked Data

1 Introduction

Reproducibility of research is the ability to get the same research results using the raw data and computer programs provided by researchers. But despite this being a main principle of the scientific method (as a means to achieve replicability), most studies in natural and social sciences, including social history, are

hardly reproducible. More than 60% of the time spent in these studies is devoted to *data munging*, preparation and manipulation that are not included in published papers [4]. At best, papers include a citation to the dataset in which its conclusions are based on.

Such a citation is, however, insufficient for a reliable, efficient and automatic reproduction of the paper. This is caused by two different factors. The first is that studies are increasingly based in not just one dataset, but a *combination of multiple datasets*. This is especially the case in social science and history, where datasets from different disciplines (e.g. demography, economics, history of work, etc.) need to be combined in order to be useful to accept or reject hypotheses. The second is that these datasets are rarely considered as a whole in the research; only a very specific *subset of the original data* is used. Finding a universal pointer (e.g. an HTTP link) in these papers solving these two issues (*(i)* references to all combined datasets; and *(ii)* instructions on how to transform and reduce them to the useful set) is extremely rare.

Research in Semantic Web technology has recently enabled the generation of such universal links to research data. "Linked Data" [6][4] is a paradigm of publishing data online that uses URIs and HTTP to connect bits of information in the same way HTML documents are connected in the World Wide Web. Two key technologies of the Semantic Web, the Resource Description Framework (**RDF**) and the SPARQL Protocol And RDF Query Language (**SPARQL**), have shown a great potential at dealing with the two aforementioned problems of combining multiple datasets and selecting their interesting parts. While RDF facilitates the creation of links that connect related pieces of information on the Web, SPARQL allows to execute very specific queries –which often implement the *research question* of the study– on those combined RDF datasets. The effectiveness of using RDF and SPARQL in combining and solving research questions has been observed in datasets from a great number of disciplines [6], including history [12] and social history [13].

However, this potential is often diminished due to two important issues, both regarding the query language SPARQL. The first is that encoding a research question as a SPARQL query is difficult, especially if it needs to be written by a non-Semantic Web expert. The second is that the maintenance of these SPARQL queries is very poor, mostly due to the non-availability of tools to maintain, curate, and successfully execute those queries.

In this paper, we address these two issues by proposing a tool called `grlc`. `grlc` addresses the problem of *writing difficult SPARQL queries* by enabling collaborative query curation workflows on the Web, concretely by using the git-based version control system GitHub[5]. To address the problem of query curation and execution, `grlc` uses the powerful principle of RESTful Web APIs to retrieve the outsourced SPARQL and supply its execution as a URI that both humans and machines can use to interact with data. In this way, `grlc` ensures

[4] See also `https://www.w3.org/DesignIssues/LinkedData.html`

[5] See `https://github.com/`

that research questions that (i) combine multiple datasets, and (ii) reduce and transform those datasets, are reproducible by just clicking one link.

In the rest of the paper, we describe the way in which we use GitHub to maintain and curate research question as SPARQL queries (Section 2), and we show how `grlc` uses these queries to generate actionable data links (Section 3). We illustrate how the same query containing a research question can be reused in a different dataset in Section 4. To show its current use, we describe a use case where the system is used in social history research (Section 5), before we draw some conclusions (Section 6).

2 SPARQL in GitHub: Collaborative Research Questions

Linked Data is a data publishing paradigm that has emerged from the Semantic Web community since 2008 [6]. It contains the two same basic elements as HTML in the traditional Web of documents: (1) a markup language to describe things; and (2) a mechanism to establish links between documents. In the traditional Web of documents, HTML is the markup language, and the <a> tag establishes the actual links. Linked Data follows the same paradigm, but with the following important differences: (a) HTML documents, which are more suitable for humans to read, are replaced by RDF triples, which are best suited for machines to process; and (b) these triples connect to other triples to establish the links. A triple is just a statement of the form `<subject> <predicate> <object>`, where each of these elements are represented by a global identifier (URI). For example, the triple

```
<https://www.w3.org/People/Berners-Lee/>
<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
<http://xmlns.com/foaf/0.1/Person>
```

represents on the Web the abstract statement that *Tim Berners-Lee is a person*. Other triples can connect to the FOAF class *Person* (the object in the above triple statement), and this class to others, building a Web-graph of interconnected statements.

Databases containing RDF triples are usually accessible via a URI link, called an *endpoint*, and there users can query them using the SPARQL language. The example query shown in Listing 1.1 encodes the research question *How many people lived in the Netherlands in the municipality of A, with sex B, with residence status C in the year D* (where *C* is one of *temporary absent* and *temorary present*, and *D* is one of 1859, 1869, 1879, 1889, 1899, and 1920).

Since writing such queries is difficult for non-Semantic Web experts, and systems that assist users in writing them have not achieved enough maturity, we propose a different approach based on *collaborative platforms*. Concretely, we propose to store these queries in GitHub, a git-based service to manage code in a collaborative fashion. There, users can create repositories and files, just as they would in their local file systems, and start writing their SPARQL queries. But contrarily to the locality of their file systems, other users can find, clone, fork, and

```
SELECT (SUM(?pop) AS ?tot) FROM <urn:graph:cedar-mini:release> WHERE {
        ?obs a qb:Observation.
        ?obs sdmx-dimension:refArea ?_location_iri.
        ?obs cedarterms:Kom ?_kom_iri.
        ?obs cedarterms:population ?pop.
        ?slice a qb:Slice.
        ?slice qb:observation ?obs.
        ?slice sdmx-dimension:refPeriod ?_year_integer.
        ?obs sdmx-dimension:sex ?_sex_iri.
        ?obs cedarterms:residenceStatus ?_residenceStatus_iri.
        FILTER (NOT EXISTS {?obs cedarterms:isTotal ?total }) }
```

Listing 1.1. Example of a research question written as a SPARQL query.

edit those queries[6]; and more importantly, *contribute* to those queries by sending *pull requests* to the original authors. Pull requests are basically improvements over the original code (SPARQL in this case) that original authors are free to ask about, reject, or accept. In the latter case, the external contributions get merged with the code base, fixing errors and getting closer to the desired behavior.

In order to enrich SPARQL queries with useful metadata (like the link to the endpoint they will be sent to), we add the following YAML notation in the first lines of files containing the SPARQL code:

```
#+ summary: A brief summary of what the query does
#+ method: GET
#+ endpoint: http://dbpedia.org/sparql
#+ tags:
#+     - I am a tag
#+     - Awesomeness
```

These lines help us to identify SPARQL queries maintained in GitHub for reproducibility purposes, and add descriptive metadata about their *endpoint* (the public HTTP database to interrogate with the query), a *summary* (describing what the query does), a *method* (whether the query gets or sends data), a number of *tags* (to meaningfully group queries together if they are related), etc.[7]

3 grlc: Actionable Data Links

`grlc`[8] is a server that leverages publicly available SPARQL queries in GitHub (see Section 2) to create *shareable links that execute them*. A group of such

[6] More details on the usual git and GitHub workflows can be found at `https://guides.github.com/`

[7] A full description of the supported decorators is available at `http://grlc.io/` and in [14].

[8] The public instance of `grlc` can be found at `http://grlc.io/`

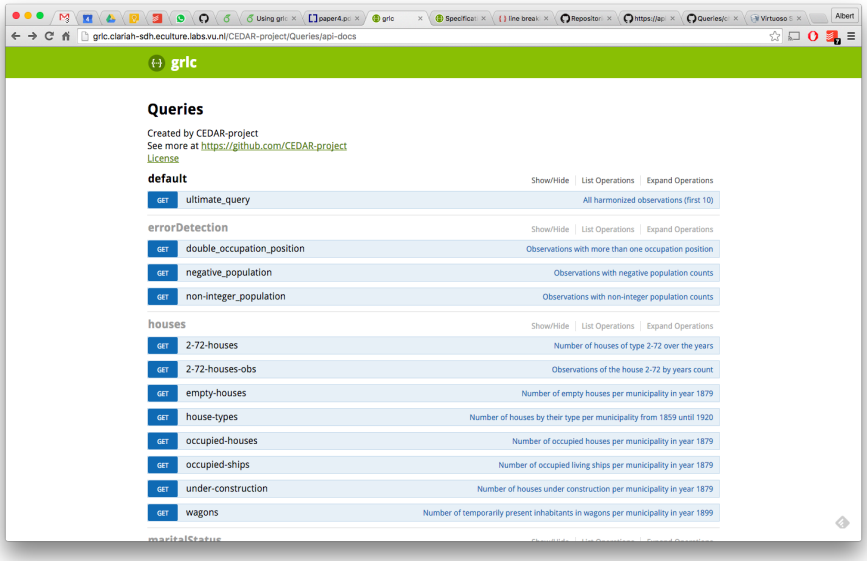

Fig. 1. Screenshot of an API generated with `grlc`.

shareable links that hold some relationship together is known as an **API**. More concretely, `grlc` generates Linked Data APIs following the OpenAPI specification[9], a standard for building RESTful APIs that both humans and machines can use for data retrieval. Where `grlc` sets difference with respect to similar tools is that it generates actionable links automatically for shared SPARQL queries that *combine and reduce* multiple datasets into a meaningful set on which a specific research question is answered. Such actionable links can then be included in research papers as unique entry points to the data and methods used to draw scientific conclusions.

An example API generated with `grlc` holding a collection of actionable links is shown in Figure 1. Each blue line represents a collaboratively curated research question as a SPARQL query in GitHub, and hence a unique, actionable link that executes that query and retrieves its answer. `grlc` uses the Swagger-UI user interface to help humans perform some basic data exploration tasks, such as suggestions of what parameters can be used (see Figure 2).

The actionable data links that `grlc` generates follow an established pattern. If the GitHub repository containing queries is located at `https://github.com/:owner/:repo`, then the `grlc` API provides the following routes:

- `http://grlc.io/api/:owner/:repo/spec`: JSON OpenAPI-compliant specification for the queries of `:owner` at `:repo`.
- `http://grlc.io/api/:owner/:repo/api-docs`: Swagger-UI, rendered using the previous JSON spec, as shown in Figure 1.

[9] See `https://www.openapis.org/`

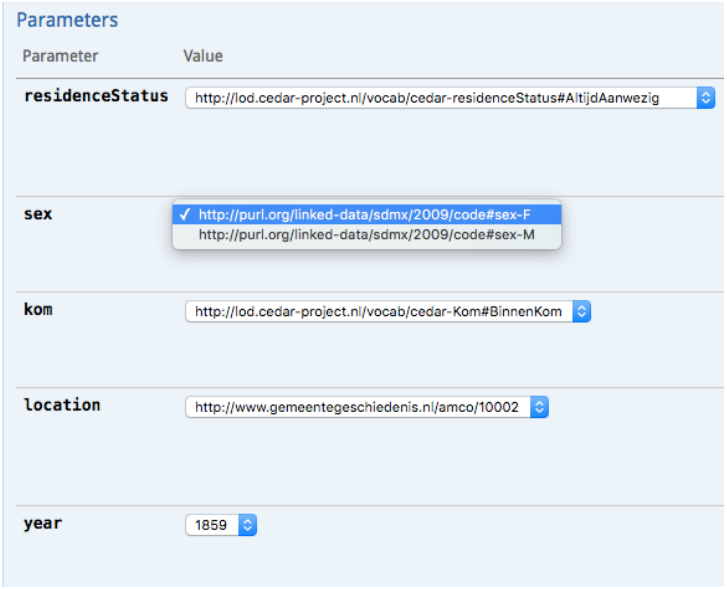

Fig. 2. Screenshot of the Swagger user interface rendering parameter enumerations generated by `grlc`.

- `http://grlc.io/api/:owner/:repo/`: Same as previous.
- `http://grlc.io/api/:owner/:repo/:operation?p_1=v_1...p_n=v_n`: `http GET` request to `:operation` with parameters $p_1, ..., p_n$ taking values $v_1, ..., v_n$.

Additional detail about the architecture of the system and its features can be found at [14].

4 Repeating Questions to Different Datasets

One of the most important features enabled by `grlc` is that of parametrized endpoints. So far, we have seen that research questions can be written as SPARQL queries; and these queries get unique identifiers (URLs) allowing their execution and sharing. However, the fact that their execution is self-contained in these URLs still makes it difficult to separate the questions (queries) from the actual data they are supposed to interrogate (datasets). In this section we clarify how this is done in `grlc`.

The standard way of specifying the link between a query in an API and its corresponding dataset is by indicating the address (URL) of the later in the GitHub repository.[10] This indicates that all queries contained in the repository should be asked against that dataset. While this separates the queries from the data to some extent, it does not allow to specify different datasets for different queries.

[10] An example can be found at `https://github.com/CLARIAH/wp4-queries-hisco`

To this end, we facilitate a second endpoint specification method: `grlc` also allows to define the endpoint address at the *query* level, in addition to the repository level. This indicates that the endpoint indicated at the query has preference over the endpoint indicated at the repository. While this allows specifying different datasets for different queries, it still binds the query and the dataset together during execution.

In order to address this, we facilitate a third endpoint specification method, which has priority over the previous two. This method allows to define the endpoint address also at the query level, but as a *parameter to be supplied on execution*, rather than hard-typed in the query. This is done by simply adding `?endpoint=http://example.com/sparql` at the end of a `grlc` API operation (or by using any other HTTP GET parameter supply method). In this way, data sources and queries (research questions) are completely separated, and only specified at query execution time.

The latter method allows not only higher reproducibility of results, but also increases studies of generality and discovery of patterns, since the same query can be scripted for execution against a list of endpoints or data suppliers.

5 Use Case

From the start of its operation in July 2016, the public instance of `grlc` has attracted 1,551 unique visitors, 58.97% of return rate, and generating 3,251 sessions. `grlc` has also attracted the attention of external developers, who have sent 31 pull requests. A list of community maintained queries and matching APIs is available at `http://grlc.io`.

In this section we evaluate the requirements satisfied by `grlc` in a use case in social history. The International Institute for Social History partners in the CLARIAH[11] project for digital humanities, specifically by contributing the dataLegend infrastrucure[12]. Typical social history research requires querying across combined, structured humanities data, and performing statistical analysis in e.g. `R` [8]. Given that there are potentially infinitely many such research queries, building a one-size-fits all API is not feasible. The `R` SPARQL package [5] allows one to use SPARQL queries directly from `R`. However, this results in hard-coded, non reusable, and difficult to maintain queries. As shown in Figure 3, with `grlc` the `R` code becomes *clearer* due to the decoupling with SPARQL; and *shorter*, since a `curl` one-liner calling a `grlc` enabled API operation suffices to retrieve the data. Furthermore, the exact query feeding the research results can be stored, and shared with fellow scholars and in papers.

Economic and social history takes questions and methods from the social sciences to the historical record. An important line of research in social and economic history focuses on the determinants of historical inequality. One hypothesis here is that prenatal [2] and early-life conditions [7] have a strong impact on socioeconomic and health outcomes later in life. A recent study on the

[11] `http://clariah.nl/`

[12] See `http://www.datalegend.net/`

United States found that people born in the years and states hit hardest during the Great Depression of the 1930s had lower incomes and higher work disability rates in 1970 and 1980 [15]. This study inspired this use case.

Most studies on the impact of early life conditions are case studies of single countries. Therefore, the extent to which results can be generalized – their external validity – is difficult to establish (e.g., differing impact of early life conditions in rich and poor countries). Moreover, historical data is often idiosyncratic. This means that dataset-specific characteristics such as sampling and variable coding schemes might influence the results.

In this use case, we explore the relation between economic conditions in individuals' birth year and occupational status in the historical census records of Canada and Sweden in 1891. In many cases it would be necessary to link the two census datasets so that they can be queried in the same way. Here, however, we use two harmonized datasets from the North Atlantic Population Project (NAPP, Canada 1891 [9] and Sweden 1890 [1]). We emphasize here that we use this data internally and for experimental purposes as this data is not meant for redistribution. The data is therefore only available to the researchers of this use case. Economic conditions are measured using historical GDP per capita figures from the Clio-Infra repository [3]. Because our outcome is occupational status, we have to enrich the occupations in the census with occupational codes and a status scheme. Because the NAPP-project uses an occupational classification that provides no internationally comparable occupational status scores, we have to map their occupational codes to the HISCO system, so that we can use the HISCAM cross-nationally comparable occupational status scheme [11,10].[13]

In general terms, the data requirements are typical of recent trends in large database usage in economic and social history:

1. the primary unit of analysis is the individual (microdata);
2. a large number of observations is analyzed;
3. multiple micro-datasets are analyzed;
4. microlevel observations are linked to macro-level data through the dimensions time and geographical area;
5. qualitative data is encoded to extract more information from it.

Current Workflow The traditional workflow to do this would include the following steps. First, the researcher has to find and download the datasets from multiple repositories. The datasets, which come in various formats, then have to be opened, and, if necessary, the variables have to be renamed, cleaned, and re-encoded to be able to join them with other datasets. We can rely on previous cleaning and harmonization efforts of the NAPP project, but in many other situations the researcher would have to do this manually. Finally, the joined data has to be saved in a format that can be used by a statistical program.

[13] `https://github.com/rlzijdeman/o-clack` and `http://www.camsis.stir.ac.uk/hiscam/`

```
46  ## using grlc API call
47  library(RCurl)
48  canada <- getURL("http://grlc.clariah-sdh.eculture.labs.vu.nl/clariah/wp4-
49  canada <- read.csv(textConnection(canada))
50  sweden <- getURL("http://grlc.clariah-sdh.eculture.labs.vu.nl/clariah/wp4-
51  sweden <- read.csv(textConnection(sweden))
52
53  fit_canada_base <- lm(log(hiscam) ~ log(gdppc), data=canada)
54  fit_canada <- lm(log(hiscam) ~ log(gdppc) + I(age^2) + age, data=canada)
55  fit_sweden_base <- lm(log(hiscam) ~ log(gdppc), data=sweden)
56  fit_sweden <- lm(log(hiscam) ~ log(gdppc) + I(age^2) + age, data=sweden)
```

Fig. 3. The use of `grlc` makes Linked Data accessible from any `http` compatible application.

New Workflow Using dataLegend's infrastructure, the workflow is as follows. Linked-data tools are used to discover data on the platform. In our case, we used the Inspector, a linked data browser[14] and exploratory SPARQL queries. The Inspector provides a simple overview of all datasets in dataLegend.[15] Note that to discover datasets and especially linked datasets, it is necessary that someone uploaded the datasets and created the links in the first place, for example by linking datasets to a common vocabulary. While it is unavoidable that someone has to do this at some point, the idea behind dataLegend is that if it is done once, the results can be re-used by other researchers.

The next step is to specify queries against the data, and store them on GitHub. The result sets that these queries produce against dataLegend are then used to create the dataset that is to be analyzed. The web interface of `grlc` can be used to explore the parameters one can use for each query: grlc populates pull-down menus with potential bindings for each variable. The straightforward HTTP interface, combined with a CSV return format, allows for direct integration in statistical environments such as R.

6 Conclusions

In this paper, we stress the importance of two critical aspects of reproducibility in social science and history: the combination of multiple datasets, and the selection of a subset of those in order to answer research questions. We argue that the ability to encapsulate these two steps in *one single shareable data link* is crucial to enable reproducibility of studies. To this end, we present `grlc`, a Semantic Web tool that achieves both goals by *(i)* maintaining research questions as SPARQL queries in the collaborative coding social network GitHub; and *(ii)* generating actionable data links that can be executed and shared on the Web, and included in papers, for a more effective retrieval of data and enabling a more automated reproducibility.

Many challenges remain open for the future. First, we will enlarge our currently supported collaborative environments (GitHub, GitLab, SPARQL, dumps,

[14] `https://github.com/Data2Semantics/brwsr`

[15] Currently at `http://inspector.datalegend.net/overview`

etc.). Secondly, we devise a JSON transformation language for customizing the structure of API results, which is important to application developers in order to enhance data integration among applications. Finally, we intend to investigate the provenance, reusability, exchangeability, and linkability of semantic query catalogs created by users of `grlc`.

Acknowledgements. This work was funded by the CLARIAH project of the Dutch Science Foundation (NWO).

References

1. National Sample of the 1890 Census of Sweden, Version 1.0. The Swedish National Archives and Umeå University, and the Minnesota Population Center, Minneapolis, MN (2011), minnesota Population Center [distributor]
2. Barker, D.J.: The fetal and infant origins of adult disease. BMJ: British Medical Journal 301(6761), 1111 (1990)
3. Bolt, J., Timmer, M., van Zanden, J.L.: GDP per capita since 1820. In: How Was Life? Global well-being since 1820, pp. 57–72. Organisation for Economic Co-operation and Development (Oct 2014)
4. Garijo, D., Alper, P., Belhajjame, K., Corcho, O., Gil, Y., Goble, C.: Common motifs in scientific workflows: An empirical analysis. Future Generation Computer Systems 36(Supplement C), 338 – 351 (2014), `http://www.sciencedirect.com/science/article/pii/S0167739X13001970`, special Section: Intelligent Big Data Processing Special Section: Behavior Data Security Issues in Network Information Propagation Special Section: Energy-efficiency in Large Distributed Computing Architectures Special Section: eScience Infrastructure and Applications
5. van Hage, W.R., with contributions from: Tomi Kauppinen, Graeler, B., Davis, C., Hoeksema, J., Ruttenberg, A., Bahls., D.: SPARQL: SPARQL client (2013), `http://CRAN.R-project.org/package=SPARQL`, R package version 1.15
6. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. Morgan and Claypool, 1st edn. (2011)
7. Heckman, J.J.: Skill Formation and the Economics of Investing in Disadvantaged Children. Science 312(5782), 1900–1902 (Jun 2006), `http://www.sciencemag.org/content/312/5782/1900`
8. Hoekstra, R., Meroño-Peñuela, A., Dentler, K., Rijpma, A., Zijdeman, R., Zandhuis, I.: An Ecosystem for Linked Humanities Data. In: Proceedings of the 1st Workshop on Humanities in the Semantic Web (WHiSe 2016), ESWC 2016 (2016), under review
9. Inwood, K., Jack, C.: National Sample of the 1891 Census of Canada. University of Guelph, Guelph, Canada (2011)
10. Lambert, P.S., Zijdeman, R.L., Van Leeuwen, M.H., Maas, I., Prandy, K.: The construction of HISCAM: A stratification scale based on social interactions for historical comparative research. Historical Methods: A Journal of Quantitative and Interdisciplinary History 46(2), 77–89 (2013)
11. van Leeuwen, M., Maas, I., Miles, A.: HISCO: Historical International Standard Classification of Occupations. Leuven University Press (2002)

12. Meroño-Peñuela, A., Ashkpour, A., van Erp, M., Mandemakers, K., Breure, L., Scharnhorst, A., Schlobach, S., van Harmelen, F.: Semantic Technologies for Historical Research: A Survey. Semantic Web – Interoperability, Usability, Applicability 6(6), 539–564 (2015)
13. Meroño-Peñuela, A., Guéret, C., Ashkpour, A., Schlobach, S.: CEDAR: The Dutch Historical Censuses as Linked Open Data. Semantic Web – Interoperability, Usability, Applicability (2015), in press
14. Meroño-Peñuela, A., Hoekstra, R.: Automatic Query-centric API for Routine Access to Linked Data. In: The Semantic Web – ISWC 2017. 16th International Semantic Web Conference, Proceedings. LNCS, Springer-Verlag, Berlin, Heidelberg (2017)
15. Thomasson, M.A., Fishback, P.V.: Hard times in the land of plenty: The effect on income and disability later in life for people born during the great depression. Explorations in Economic History 54, 64–78 (Oct 2014)