Universidad de Deusto
Facultad de Ingeniería

A Data-Driven Visual Approach to Explore Linked Open Data Environments

Tesis doctoral presentada por Oscar Peña del Rio

dentro del Programa de Doctorado en Ingeniería para la Sociedad de la Información y Desarrollo Sostenible

dirigida por el Dr. Diego López de Ipiña González de Artaza
y el Dr. Unai Aguilera Irazabal

El doctorando | El director | El co-director

To my dearest Oiane, Max, Summer & Taimi,
thanks to whom this dissertation would have been
either finished much earlier
...or never at all

Abstract

Humans have registered their environment since ancient times, either verbally, through writing or by using graphics and images. These records have allowed us to learn from the past, analysing previous scenarios and extracting new knowledge that has transformed our societies, making us capable to address and deal with new challenges.

Since the invention of the World Wide Web by *Sir Tim Berners-Lee*, the ease to publish, update, discover and access new data has grown exponentially, and it is estimated that every two years we generate as much data as in the whole history before. *Tim Berners-Lee* envisaged that machines could help humans in data processing and understanding tasks, giving birth to the *Semantic Web* field, an scenario in which data is provided together with semantic annotations, allowing its comprehension by algorithms.

Years later, in 2006, the Linked Data principles were proposed as a method of publishing structured facts, so that they could be connected (linked) to other resources through the World Wide Web. It relies on standard Web technologies, and is intended to be consumed by computers.

Despite the benefits brought by Linked Data, the adoption of its related developments has normalised after the initial years, and little attempts are performed outside the research community. To make Internet users aware of Linked Data's advantages, we propose an approach to explore its datasets using visual means, relying

on our ability to discover patterns and insights through graphic imageries and depictions.

In order to deal with the diversity of structured data published as Linked Data, our proposal takes a data-driven approach, that is, we base our whole analysis on the data itself, avoiding preconceptions that might lead to wrong conclusions. The main objective is to ease semantic data exploration through suitable visualizations, making any user able to interact with novel datasets with no prior knowledge nor skills required.

In this dissertation, we explain the visualization pipeline that allows to take raw semantic data as input, and produces visual representations as output, together with the involved modules and the contributions we have designed and implemented to push forward the State of the Art on Linked Data Visualization.

Resumen

Los humanos han tomado registros de su entorno desde la antigüedad, tanto de forma oral, escrita como usando elementos gráficos e imágenes. Estas anotaciones nos han permitido estudiar nuestro pasado, analizando situaciones previas y extrayendo nuevo conocimiento de las mismas, lo que ha hecho que nuestras sociedades evolucionen, siendo capaces de abordar y superar nuevos desafíos.

Desde la invención de la *World Wide Web* por *Sir Tim Berners-Lee*, la facilidad para publicar, actualizar, descubrir y acceder a nuevos datos ha crecido de forma exponencial, y se estima que cada dos años generamos tanta información como en toda la historia de la humanidad hasta esa fecha. *Tim Berners-Lee* previó que las máquinas podrían ayudar a los humanos en las tareas de procesado e interpretación de los datos, dando lugar al área de la *Web Semántica*, un supuesto en el que los datos son proporcionados junto a ciertas anotaciones semánticas, que les habilitan para ser comprendidos por distintos algoritmos.

Años más tarde, en 2006, enunció los principios de los *Datos Enlazados*, una serie de técnicas orientadas a publicar datos de forma estructurada, de manera que estos puedan ser conectados (enlazados) con otros recursos a través de la *World Wide Web*. Estos principios se cimentan en los estándares web, y están diseñados para ser consumidos por computadoras.

A pesar de los beneficios de los *Datos Enlazados*, la adopción de estas técnicas se ha estabilizado tras el auge inicial, y son pocos los

intentos de mejora que se realizan fuera de la comunidad académica. Para hacer que los usuarios de Internet sean conscientes de las ventajas de los *Datos Enlazados*, proponemos una estrategia de exploración de conjuntos de datos a través de los medios visuales, aprovechando la capacidad humana para detectar patrones y aumentar nuestro conocimiento mediante el uso de representaciones visuales.

A fin de gestionar la diversidad de datos estructurados publicados como *Datos Enlazados*, nuestra propuesta toma un enfoque basado en los propios datos, eliminando las ideas preconcebidas que puedan llevar a conclusiones erróneas. El objetivo principal de esta tesis es facilitar la exploración de datos semantizados a través de visualizaciones apropiadas, de forma que cualquier usuario sea capaz de interactuar con conjuntos de datos originales sin necesidad de conocimientos técnicos avanzados ni habilidades especiales.

A lo largo de esta disertación, explicaremos el proceso de visualización que permite tomar datos semantizados como entrada, y generar representaciones visuales como salida, junto a los módulos involucrados y las contribuciones que se han diseñado e implementado a fin de mejorar el estado de la cuestión en la visualización de Datos Enlazados.

Acknowledgements

In order to acknowledge all the people and entities that have made this doctoral experience possible, I would like to present one of *Fraser Raeburn*'s metaphors about what a PhD might resemble to, entitled: *The PhD as a meal in a Michelin-starred restaurant*[1]:

> A PhD drastically reduces your surplus income and often involves going into debt, unless someone likes you enough to pick up the tab. You keep telling yourself that the experience is totally worth it, although you have nagging doubts that you don't want to express because everyone around you looks really natural and at home in their surroundings. Everything you need to read seems to be in a foreign language. You have an irresistible urge to chronicle everything on social media. Nothing that you end up with on your plate looks anything like what you expected it to when you started.

First of all, thanks to the Basque Government's *Hezkuntza, Hizkuntza Politika eta Kultura Saila* (Department for Education, Linguistics Policy and Culture) and to DeustoTech-INTERNET (University of Deusto) for "picking up the tab" on my PhD and its related activities. It is not easy to find someone who decides to pay for your dinner, specially when you seem to be starving.

[1] http://www.blogs.hss.ed.ac.uk/pubs-and-publications/2016/07/11/why-your-phd-is-a-metaphor

To all my colleagues in the MORElab team, both present and past, for "looking really natural" whilst eating at the restaurant, always telling me that I would eventually enjoy the meal. Even though there were moments when all I wanted was to leave the dinning room (that time when the kitchen took fire, when the health inspector visited the cold chamber, or those situations when I was either eating completely alone in the dinning room or when it was full to bursting), at the end I can not but agree that in perspective, eating in such a restaurant is a unique adventure. Your constant advice, professionalism and expertise have helped me overcome most hurdles, even when the menu was written in a "foreign language". It has been a real honour to met all of you.

To my advisers, true chefs themselves, for their guidance during the menu's selection and encouragement to try new flavours, specially when some of the resulting dishes were not of our taste at all. You accepted me as an inexperienced *foodie* three years ago, and have seen me evolve into a *gourmet*.

Moreover, my whole academic life has been possible thanks to the efforts made by my parents, and I will never be able to thank them enough for it.

Finally, I would like to express my deepest gratitude to my dinner guest, *Oiane*, who has accompanied me to the restaurant's door, bear with all the twists and turns and is expectantly awaiting for the dessert. Also to our three (up to now) furry puppies (*Max, Summer and Taimi*), who have silently cheered me up from the children's table.

Milesker

Oscar Peña del Rio

Table of Contents

List of Figures

List of Tables

List of Listings

Acronyms

CSV Comma Separated Values

CWA Closed World Assumption

D3 Data-Driven Documents

DOM Document Object Model

EDA Exploratory Data Analysis

HTML HyperText Markup Language

HTTP Hypertext Transfer Protocol

IRI Internationalised Resource Identifier

JSON JavaScript Object Notation

LD Linked Data

LOD Linked Open Data

OWA Open World Assumption

OWL Web Ontology Language

PDF Portable Document Format

RDF Resource Description Framework

RDFS Resource Description Framework Schema

SPARQL SPARQL Protocol and RDF Query Language

SQL Structured Query Language

SW Semantic Web

URI Uniform Resource Identifier

W3C World Wide Web Consortium

XML Extensible Markup Language

XSD XML Schema Datatype

The Web as I envisaged it, we have not seen it yet.
The future is still so much bigger than the past.

18th WWW Conf., Tim Berners-Lee (2009)

CHAPTER

1

Introduction

Back in May 2001, Tim Berners-Lee, James Hendler and Ora Lassila presented their seminal article: "The Semantic Web" (Berners-Lee et al., 2001), where they proposed the next stage of the Internet; a data space whose contents can be manipulated by computer programs. The key concept behind the *Semantic Web* (SW) is to extend the Web as we know it by structuring and providing semantic meaning to the contents shared over the Internet, so that *software agents* are able to understand and process them to extract knowledge from raw records. Data in the Semantic Web are mainly represented using any Resource Description Framework (RDF) notation (Lassila and Swick, 1999). RDF is a conceptual model that allows to encode facts as a set of triples, each of them being rather the subject, property and object of a traditional statement. To query and manipulate RDF statements, SPARQL (SPARQL Protocol and RDF Query Language) was standardised (Prud'Hommeaux et al., 2008) using a similar syntax to the one defined by SQL (Structured Query Language).

Both resources and the relations among them are formally described using **ontologies**. Thomas R. Gruber defined the term as (Gruber, 1995):

> An ontology is an explicit specification of a conceptualization. [...] we can describe the ontology of a program by defining a set of representational terms. In such an ontology, definitions associate the names of entities in the universe of discourse (e.g., classes, relations, functions, or other objects) with human-readable text describing what the names mean, and formal axioms that constrain the interpretation and well-formed use of these terms.

As collected in the book "Ontological Engineering" (Gómez-Pérez et al., 2004), Uschold and Jasper (Uschold and Jasper, 1999) refined the definition in the form of:

> An ontology may take a variety of forms, but it will necessarily include a vocabulary of terms and some specification of their meaning. This includes definitions and an indication of how concepts are inter-related which collectively impose a structure on the domain and constrain the possible interpretations of terms.

Thus, ontologies offer a closed taxonomy to describe concepts and relations, together with the possibility to include restrictions about their behaviour. Using ontologies to describe any resources allows data publishers and consumers to share structured information upon previously agreed terms, avoiding misunderstandings and ambiguous declarations. This environment further helps in data integration, and may lead to the discovery of new relationships between different knowledge fields.

Later, on July 2006, Tim Berners-Lee coined the term Linked Data (LD) in a Semantic Web design note[1], a set of principles to connect related resources through the Web. In essence, Linked Data is a method to publish structured data on the web, specially formatted to be used by machines thanks to the semantic statements used to describe each fact. Its principles are informally summarised as:

[1] https://w3.org/DesignIssues/LinkedData

- Use URIs (Uniform Resource Identifiers) as names for things.
- Use HTTP (Hypertext Transfer Protocol) URIs so that people can look up those names.
- When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL).
- Include links to other URIs, so that they can discover more things.

On 2010, Tim Berners-Lee extended the definition of Linked Data to embrace all those datasets that include *Open*-licensed contents, under the *Linked Open Data* (LOD) nomenclature. With the purpose of encouraging data generators to publish their datasets following LOD standards, a 5-stars rating system was developed. The more stars a dataset obtained, the more powerful and easier for people to use it would be:

- **1 star**: Available on the web (whatever format) but with an open license, to be Open Data.
- **2 stars**: Available as machine-readable structured data (e.g. Excel instead of image scan of a table).
- **3 stars**: All the above plus non-proprietary format (e.g. CSV - Comma Separated Values - instead of Excel).
- **4 stars**: All the above plus, use open standards from the World Wide Web Consortium (W3C) (such as RDF and SPARQL) to identify things, so that people can point at your stuff.
- **5 stars**: All the above plus, link your data to other people's data to provide context.

In an effort towards promoting the access and discoverability of all the data sources published as LOD, they are collected and grouped together in what is known as the LOD-Cloud diagram, a node-link graph representation

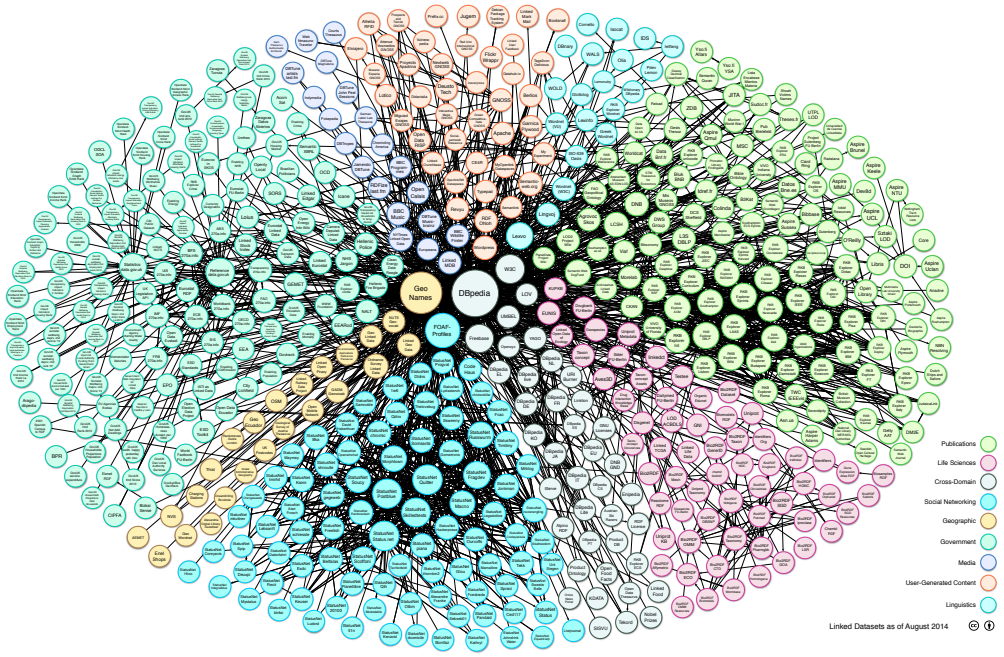


Figure 1.1: LOD-Cloud diagram as of August 2014

of all the datasets registered in the Datahub[1] (a platform to manage datasets promoted by the *Open Knowledge Foundation*[2]) with a tag related to LOD and the connections between them. The last version of the LOD-Cloud diagram was released on August 2014[3] and is depicted in Figure 1.1.

1.1 Motivation

Many technologies have been developed to build the foundations of the Semantic Web, each of them aiming to realise Tim Berners-Lee's vision: a Web of Data understandable by machines, able to provide more complex functionalities than those they exhibit at present. As more data is being published as Linked Data under Open licenses (Schmachtenberg et al., 2014), the need to understand and work with novel facts increases accordingly. Due to the topic diversity covered by these datasets, tools unfitted to operate with records

[1] https://datahub.io
[2] https://okfn.org
[3] http://lod-cloud.net

from different fields or ontologies rapidly make evident their limitations, as they can not adapt to changes in how data is described without the need to partially or entirely rewriting their code.

According to biological research, human vision is the sense with the largest bandwidth for sending information to our brains, as retinae transmit data at roughly the rate of an Ethernet connection, e.g., 10 million bits per second (Koch et al., 2006). Hence, it seems reasonable to rely on data visualization as a compelling approach for fast knowledge acquisition.

After conducting a thorough State of the Art review regarding visualization methodologies and technologies in the Semantic Web field (Chapter 2), we have identified some opportunities to contribute for *smart*, *generic* and *automatic* visualizations for LOD, meaning:

- **Smart**: One of the main contributions of LD is the annotation of concepts using shared schemas, allowing different actors to agree on the definitions used. These descriptions are understandable by machines, so algorithms with a semantic input should provide more fine grained outputs (visualizations) than those created from raw data.

- **Coherent**: The generated visual representations should be suitable for the described datum. If geographical features are identified within the dataset in the format of latitude|longitude pairs of values, a map should be the first option to plot the resources. Other tools that visualise structured data (e.g., Google Spreadsheet's *Explore* feature or the chart generation tool of most spreadsheet software, along with others) detect that these properties are typed as numbers, and render them on either a scatter-plot or a column chart, missing the opportunity to provide real value to end users.

- **Automatic**: Our aim is to lower the knowledge barrier required to explore semantic datasets, despite the users' background. Whereas more automation means less customization of the resulting visualizations, it also reduces the resources and skills needed to play with the data, easing the path towards knowledge discovery using LOD.

1.2 Hypothesis and goals

Based on the current state of Linked Open Data visualization, the hypothesis of this dissertation is stated as:

Hypothesis

The usage of **semantic annotations in publicly available datasets** enables their exploration by visualization tools designed to leverage the benefits of semantically enriched data, offering a better understanding of the described features in opposition to those scenarios in which raw, plain data is used as input. The potential visualizations that can be produced by relying upon semantics would allow to **explore datasets with no prior experience with their contents or semantic technologies comprehension**. By taking into consideration the inner structures, types, features and so of the target datasets, **visual representations** that are **suitable** for the data under focus should be generated.

In order to validate the formulated hypothesis, the main goal of this dissertation is established as follows:

Main goal

Design and implement a **visualization pipeline** capable of taking raw Linked Open Data as input, and generate generic visual representations which provide a quick overview on any dataset's contents, easing the data exploration and discovery stages.

The main goal can be achieved by addressing the following more specific and measurable objectives:

- Design and implement a visualization tool that best represents the contents of each data dimension within a semantic dataset, following the best practices from the information visualization field.
- Support the data exploration task with basic analysis techniques, easing the discovery of relationships, patterns and trends in the data that could lead to improved knowledge acquisition.

- Evaluation of the prototyped system with real users, to assess the validity of the proposal for potential users of the pipeline.
- Validation of the designed approach and its related modules, assessing the value of the contributions made, and comparing the results to more widespread and popular proposals.

1.3 Scope, context and restrictions

To achieve the goals stated in this dissertation and check our proposal's validity, the following assumptions about the environment are established, which delimit the context in which this research will be performed.

- We presume that the analysed datasets are publicly available through a SPARQL endpoint, allowing queries to be performed over the published facts. Our approach works also with downloaded dataset dumps, but we encourage and support the publication of LOD on the web, accessible for anyone that might be interested in it. By providing a unique point of access (the SPARQL endpoint) that might be updated at any time, we could loose experiment reproducibility, but make sure that analyses performed in the same time span work with the exact, same data.
- SPARQL endpoints should be robust enough, being able to always produce an answer to a simple query in reasonable time, even if the answer is an empty set. Server time-outs or errors are not considered. In real environments, the inability to produce an answer without errors is a common scenario (as remarked in the dissertation's final conclusions). To avoid dealing with non-responsive services, we have only selected reliable endpoints which are able to answer simple queries.
- Throughout this dissertation, we use the terms Linked Data and Linked Open Data in a detached fashion, using both terms to refer to datasets with semantic annotations. The exhibited approach is applicable to either definition, the only difference being the public access to LOD resources, whereas some LD datasets might have restricted access policies.

Together with the previously exposed surmises, we limit our approach within the subsequent boundaries, describing the scenarios that this dissertation is conceived to target.

- The target is set on 4 and 5-star Linked Open Data, this is, our solution only works with data described as RDF, regardless of the notation used. Any other structured format is not supported. Up to 3-stars, data is not semantically annotated, so our approach is not able to extract any information from it. Some works address this issue by trying to parse data from PDFs (Portable Document Format) and scanned tables with limited success, and further developments are required to make low-level LOD feasible to be suitable for automatic analyses.

- One of the key goals of this work is to generate automatic visualizations that aid users during an exploratory analysis. Our solution does not intend to produce optimal results when compared to those generated by experts in the field or in data analysis/visualization, but to guide the exploration when a new data source is dealt with, despite the technical background and skills of the user. We acknowledge that a skilled analyst, expert in the domain being analysed and with a strong background in visualization, would get finer grained insights on its own that by using our pipeline.

- The proposed solutions and prototypes are web based, allowing universal access as was originally intended by the Internet. Native (desktop or mobile) based solutions are not taken into account. Developing a tool which is accessed through an Internet browser ensures that all users access the same version of the application, without the need of a special software suite to explore LOD datasets.

- We do not judge ontology visualization (hierarchical representation of the classes, properties and their relationships within a vocabulary) to be classified under the LOD visualization scope. Those tools are supplementary to our proposal.

1.4 Methodology

To carry out all the steps involved in the process of pursuing the research envisaged for this thesis, the research strategy depicted in Figure 1.2 has been adopted. All the comprised stages are categorised into three main blocks, identified by different colours.

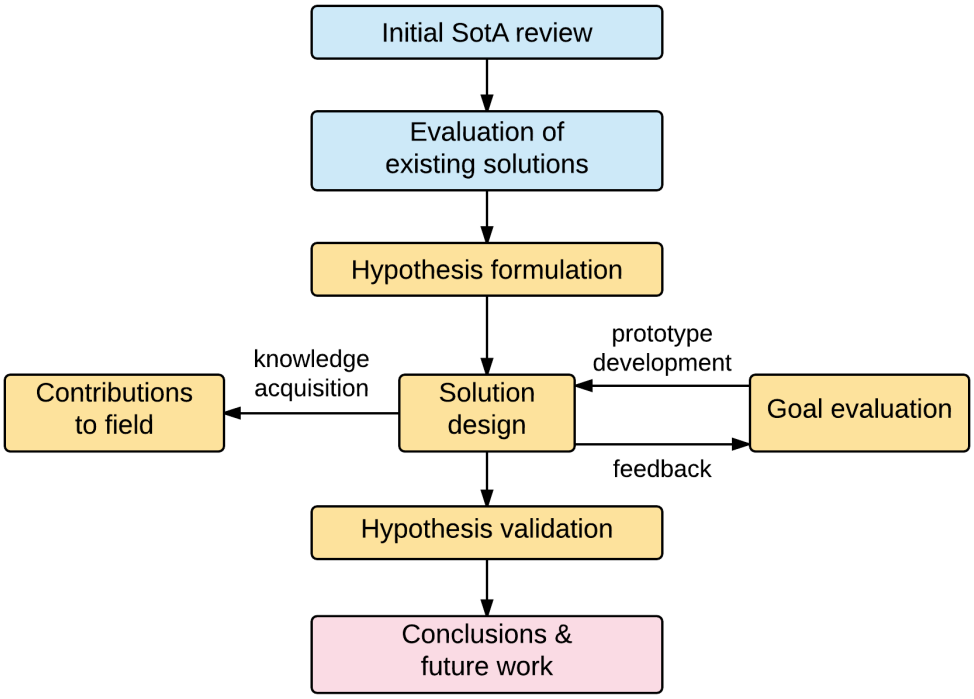


Figure 1.2: Followed research methodology.

- **SotA review (blue)**: Literature review on Linked Open Data visualization approaches and applications, focusing on those dealing with heterogeneous data. Our understanding of the field has been enriched by attending international events specialised in the field, and the collected references from the most significant works that form the State of the Art in the matter (consult Chapter 2). Having identified the spots that could benefit from further research, guided our approach towards the potential contribution areas.

- **Design and implementation of our approach (orange)**: After analysing the existing approaches towards semantically annotated data visualization, the research hypothesis was formulated (as exposed in Section 1.2). Together with its formalization, a series of goals were proposed and its limitations and scope exposed. In order to validate the proposed hypothesis, a LOD visualization pipeline was designed, and its modules implemented, allowing to transform raw input data into interactive web visualizations in an automated manner. The modules' functionalities are featured in Chapter 3 and Chapter 4. We followed an iterative process, in which prototypes were designed, implemented and improved using the received feedback about each of them (refer to Chapter 5 to read about how the whole prototype was evaluated). The knowledge acquired was communicated publishing academic articles in different conferences and journals, receiving valuable feedback from field experts and colleagues in the scientific community. The ideas taken served to improve and redefine some of the modules. Finally, all the contributions were tested and the validity of the hypothesis checked. The achieved goals were also reviewed, comparing them to the initially established objectives.

- **Dissertation writing and future goals (pink)**: Once all the dissertation's contents were written, we have presented the conclusions extracted after conducting this research (see Chapter 6). This section also addressed the limitations of the approach, and proposed some future lines of work that might inspire further research on the area.

1.5 Dissertation outline

The dissertation is structured as follows:

Chapter 1 briefly presents the Linked Open Data field and explains the motivation behind our research proposal, a visualization pipeline that allows exploratory analyses on LOD datasets. To that end, we formulate the hy-

pothesis and list the research goals, together with its scope and restrictions. Finally, the methodology followed through the thesis is described.

In Chapter 2, a quick overview on visualization history is given, making the reader aware of the evolution of this field since the first drawings to its latest developments. Later, we review the current state of the art on LOD visualization, outlining the contributions made by each work to our specific area of interest. This chapter compares the existing approaches and summarises their scopes and limitations, identifying the features that could benefit from further improvement.

Chapter 3 deals with how to infer what are the contents of each dataset by focusing on the data itself. To understand which topics the data covers and how we can manipulate it, its structure and some extra metadata metrics are extracted, storing the results for further reuse. Evaluations of the automatic data profiling are also provided.

Chapter 4 takes the knowledge that allowed to profile each dataset in the previous chapter, and devises an heuristic that allows to determine the perfect visual representation match for the task at hand. By taking into account what the analysis objectives are, the visualization prototype proposes the best suited graphics to depict the data.

Chapter 5 describes our prototype implementation of a LOD visualization pipeline: John Snow. This prototype brings together all the concepts presented during the dissertation, and its iterative development provided valuable feedback that determined some of the approaches taken in this research work. A user evaluation is also presented to assess its validity.

Finally, Chapter 6 draws the conclusions we come up with after developing this research. We exhibit a critical analysis of the performed work, and sketch out some feasible future lines of work, to be addressed in further proposals.

Bernard of Chartres used to say that we [the Moderns] are like dwarves perched on the shoulders of giants [the Ancients], and thus we are able to see more and farther than the latter. And this is not at all because of the acuteness of our sight or the stature of our body, but because we are carried aloft and elevated by the magnitude of the giants.

John of Salisbury (1159)

CHAPTER 2

State of the Art

As centuries passed by, human beings have dealt with bigger information quantities. In its early stages, mankind transmitted facts using verbal communication, what is formally known as *oral tradition*, broadcasting knowledge from one generation to the next one. This communication process had a limited reach (family, friends and reduced audiences), so the information's integrity could not be guaranteed. A common practice was to embed the messages in songs, chants, storytales and so on in order to ease recall and prevent deterioration because of the transmission process.

Thanks to *writing*, people could register, store and communicate facts in a more trustworthy fashion. With a lower loss rate than oral tradition, manuscripts could be copied and reach a wider public, as was addressed by the Latin proverb "*Verba volant, scripta manent*" (spoken words fly away, written words remain) attributed to Roman senator *Caius Titus*. With the invention of the printing press around 1440 by the German *Johannes Gutenberg*, the generation of printed scripts at large scale became economically viable, therefore allowing to communicate information with much less effort.

In the last decades, thanks to many technological advances, we live in what is called "*The Information Age*": a knowledge-based society able to

share, access and publish information easily using Information Technologies. It is impossible to calculate the amount of data that is stored in the world presently. Actual estimations guess that every two years, we generate the same amount of information as in all previous human history together, and as many of these data are available on the Internet, different studies have been carried out in order to figure out the number of websites, content generated and the like (van den Bosch et al., 2016), providing an overall impression of the amount of information an interested mind can have access to.

2.1 Data visualization

Concurrently with all these knowledge transmission techniques, mankind has used images to depict the reality they are surrounded by. When thinking about the first exhibitions of visual representations, the reader might have brought into memory the pictograms that were painted on ancient caves, or the hieroglyphs which recorded the whereabouts of past Egyptian cultures. But the use of imagery has far more uses than the mere depiction of what we see or the illustration of texts.

2.1.1 Visualization through history

Statistics and data analysis usually acquainted their results in pure numerical format or, when multiple results were presented from the same experiment, in tabular form. A few pioneers started using visual representations to communicate the results of their quantitative analyses. The early developments of data visualization allowed to identify patterns, trends and outliers, and these first experiments were born in order to reason about a particular set of evidence and reach a conclusion.

2.1.1.1 19^{th} century

Most works refer to **William Playfair** (1759-1823), a Scottish engineer and social scientist, as the first promoter of graphical methods applied to statistics. He was not the first one to come up with the idea of using visual

means to encode information. Joseph Priestley (1733-1804) already used basic timeline charts to depict the lifespan of historic characters in his book *A Chart of Biography* (1765). Playfair took inspiration from Priestley's works, and included the developments of the French philosopher and mathematician René Descartes (1596-1650) in the Cartesian space to invent the bar chart (Figure 2.1). Playfair is also credited to have devised line graphs as a way to represent changes through time, and designing pie charts to display value distributions (Figure 2.2).

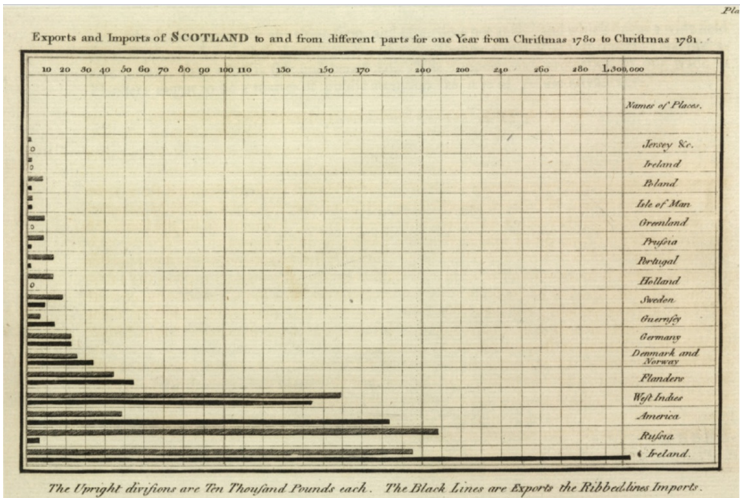


Figure 2.1: Bar chart displaying the exportations and importations from Scotland, as appeared in "*The commercial and political atlas*" (1786).

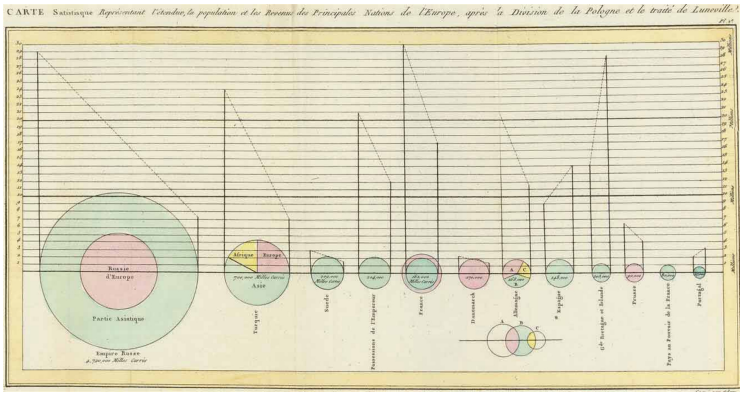


Figure 2.2: Extent, population and revenue of the principal nations in Europe using pie charts, "*The statistical breviary*" (1801).

Charles Joseph Minard (1781-1870) was a French civil engineer who has also been commonly acknowledged for his contributions to the data visualization field. His most significant work is the depiction of *Napoleon*'s 1812 Russian campaign, addressing the losses suffered by his troops (Figure 2.3). The chart is famous for the inclusion of many different variables in two dimensions: the chart shows the number of troops initially deployed for the campaign, and how they were redistributed and decimated. In the lower section of the graphic the temperatures at each point of the march are displayed using a line chart. The below zero weather conditions contributed to the death of thousands of soldiers due to freeze. The analysis has also a strong geographical component, as it flows from the departure point to Moscow, and all the way back. The thin black, irregular lines make reference to the rivers the troops crossed, and it can be clearly seen how many lives were lost at them. The chart has been one of the most reproduced ones since its publication, and has been recognised as "*one of the best statistical drawings ever created*" (Tufte, 1986).

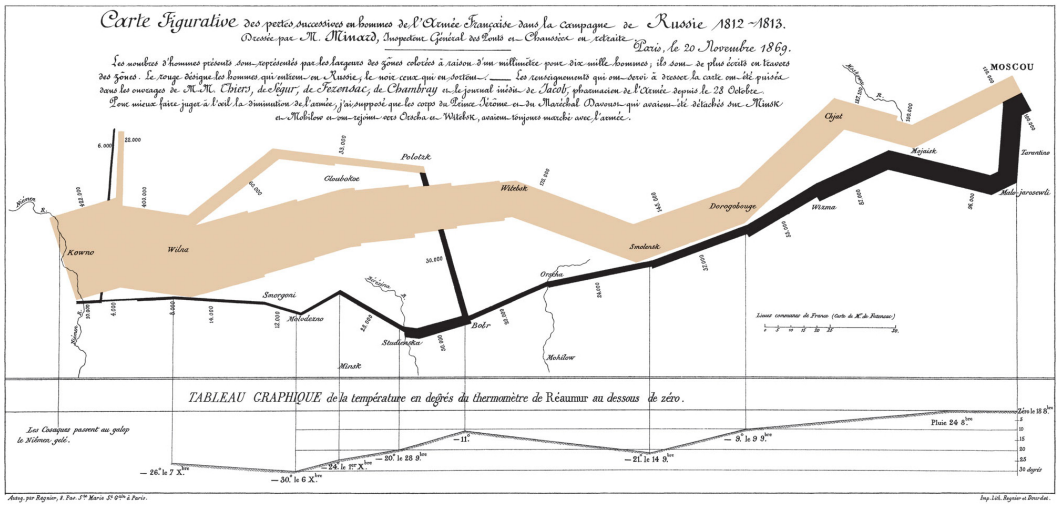


Figure 2.3: Charles Minard's original chart, named (in french): "*Carte figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie 1812-1813*" (1869).

Years later, **John Snow** (1813-1858), a British physician famous for his knowledge on anaesthesia and its application to surgical processes, gained relevance for his visually-supported analyses. John Snow is widely known for his implication tracing the cholera outbreak which struck London between 1853 and 1854. Thinking that the cholera was neither transmitted through air, nor by direct contact between patients (as it was commonly believed at the time), he conducted different experiments to validate his hypothesis: infected water was responsible for the epidemic. In order to illustrate his intuition, he elaborated a dot map in which he draw a point in each location a deceased was found, and marked with *X* the spots of each public water pump (see Figure 2.4). The high density of deaths spread around a particular water pump, the one located in *Broad Street*, confirmed his hunch. He contacted the local authorities to disable the pump, and the number of cases rapidly diminished. John Snow is considered the father of modern epidemiology, as well as one of the early exhibitors of visual reasoning.

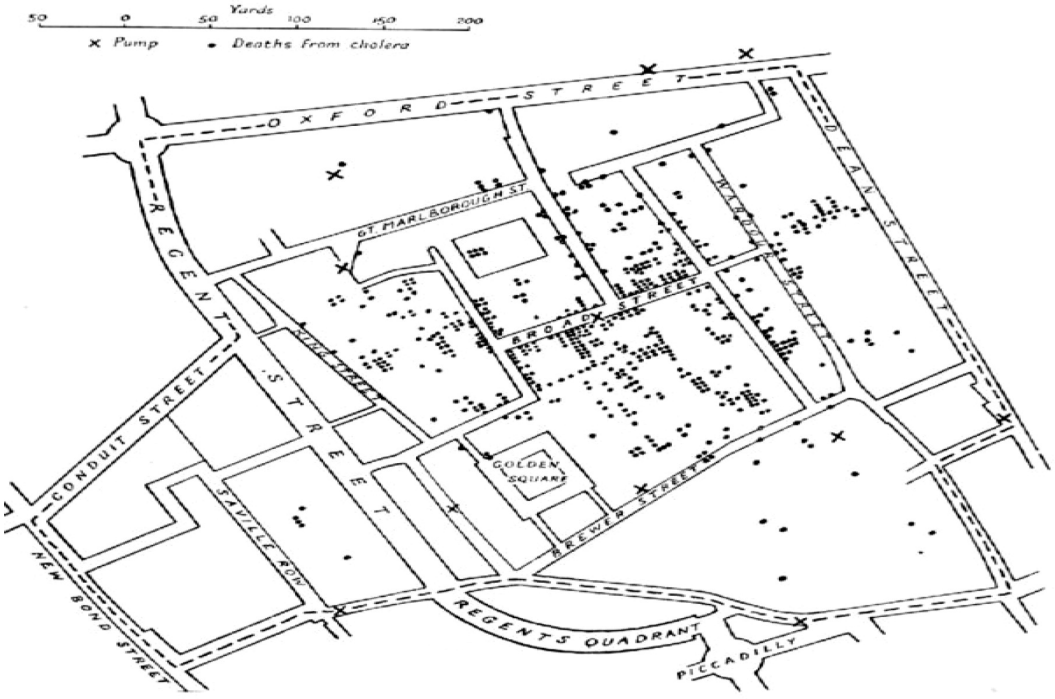


Figure 2.4: John Snow's cholera map, the source water pump can be found near the *D* letter in *Broad Street*, at the centre of the map (1854).

Contemporary to *John Snow*, **Florence Nightingale** (1820-1910), a member of the elite British society of the XIX century, raised as an exponent of visual reasoning in the sanitary field. Because of her privileged background, she was educated in different areas by her father, developing a special interest in mathematics. She was also active in philanthropy from a very young age, what would encourage her years later to pursue a nursing career in opposition to her parents, who thought a girl of her status should become a wife and mother. The outbreak of the Crimean War (1853-1856) where both the British and Russian Empires engaged in a war for the control of the Ottoman Empire, sent her and other 38 volunteer nurses to Scutari (currently Üsküdar, a district of Istanbul, Turkey), under the authorisation of the Secretary of War. The awful condition of the facilities in which the wounded soldiers were attended, among other factors, resulted in a high mortality rate. She identified that the poor hygiene was a key element in infection expansion, and directed her efforts to guarantee that some minimal conditions were met. The best way she transmitted the results of her work was the elaboration of a chart displaying the causes of mortality (Figure 2.5), a polar area chart (a pie chart variation also known as rose diagram at the time) that highlighted the huge mortality descent after the application of hygienic measures. The diagram shows that a high percentage of fatalities were not due to direct injuries, but a result of complications related to infection spread caused by the hospital's condition. She is considered the founder of modern nursing, acknowledged to have improved the work status of women worldwide and a clear reference for the use of graphics to communicate statistical analyses.

We can summarise that most of the works created in the 19^{th} century were oriented towards **communicating** the results of experiments or statistical analyses using visual means. The previously presented characters are considered to be amongst the first ones to create visualizations to support their observations, establishing the foundations of new knowledge fields through their paths. It is noteworthy to state that many of the chart types that we consider *basic* nowadays (e.g., bar and column charts, pie charts, polar diagrams and line charts) were conceived at most two centuries ago.

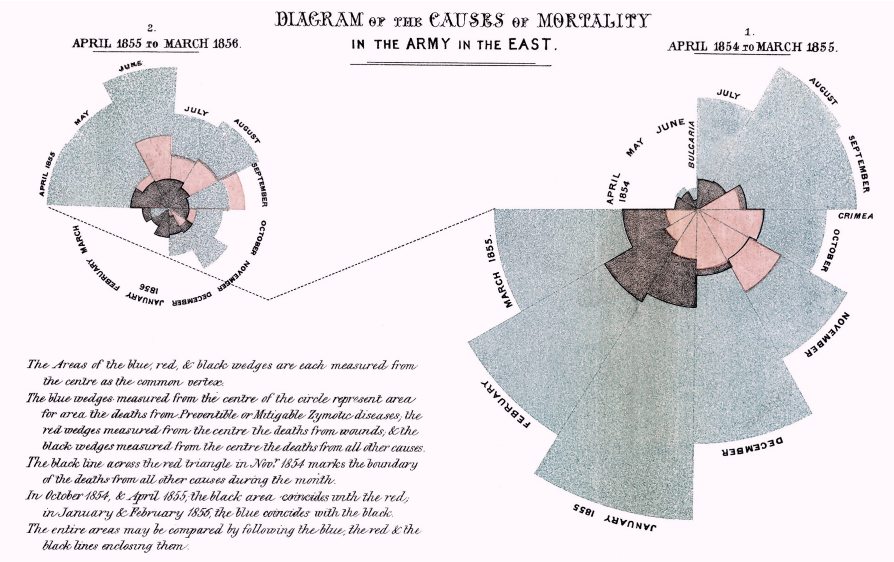


Figure 2.5: Florence Nightingale's diagram of the causes of mortality during the Crimean War, as published in "*Contribution to the Sanitary History of the British Army*" (1859).

2.1.1.2 **20th century**

Viewed in perspective, the following century laid the foundations for the visualization field as a knowledge area by its own rights, and not only as a supporting activity to other sciences and knowledge fields. The definition of many key concepts, the elements that intervene in the generation of a chart and the user's perception of the resulting images built the core of visualization.

The first author to establish the theoretical background of information visualization was **Jacques Bertin** (1918-2010), a French cartographer and geographer who published back in 1967 his book "*Sémiologie graphique*" (translated in English as *Semiology of Graphics*). This volume set the groundwork to classify raw information. For this purpose, *J. Bertin* described 5 "*marks*" that could represent information. These *marks* are:

- **Points**: Dimensionless locations on a flat surface.
- **Lines**: Entities only providing its length.
- **Areas**: Bi-dimensional spaces, with both length and width.

- **Surfaces**: Areas in a three-dimensional space, but with no thickness defined.

- **Volumes**: Pure three-dimensional entities (width, length and thickness).

Within these *marks*, we can encode and modify information using the original *visual variables* identified by *Bertin*: position, size, shape, value (variations from light to dark, as in a grayscale), colour, orientation and texture. For his works, he is acknowledged to be one of the forefathers of *Visual Analytics* (the field of reasoning using visual means).

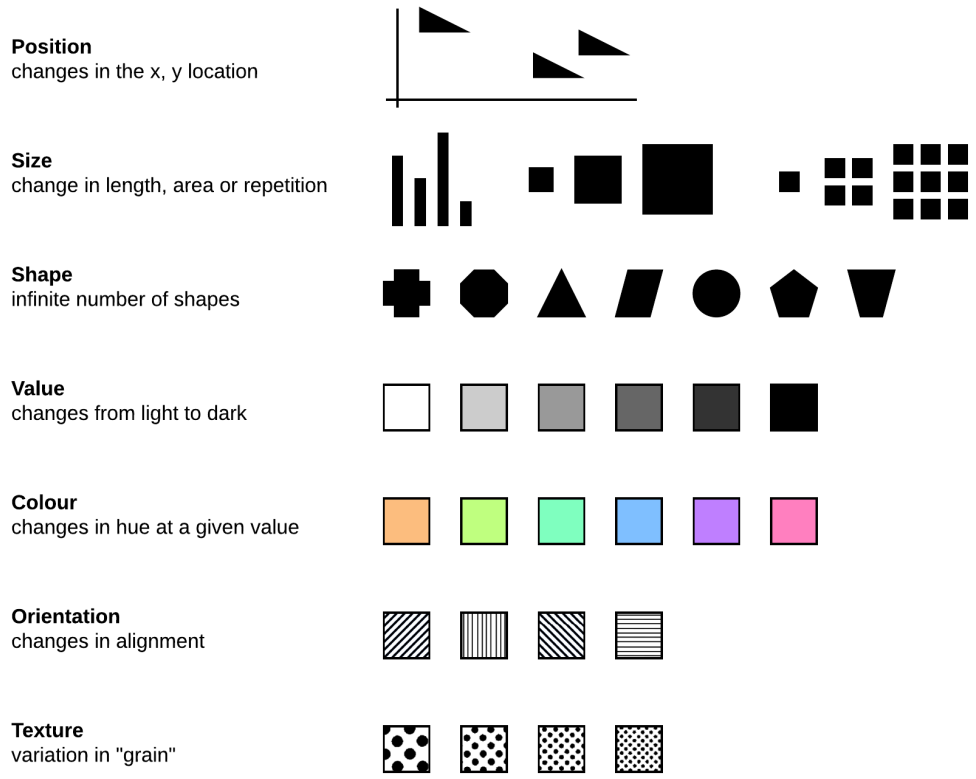


Figure 2.6: Definition and examples of the seven visual variables originally described by *Jacques Bertin*. He envisaged that combining any of these variables, any quantitative or qualitative analysis could be visually represented.

Ten years after *Jacques Bertin*'s seminal book, the American mathematician **John Wilder Tukey** (1915-2000) presented his book "*Exploratory Data Analysis*", an innovative approach to analyse data sets which heavily relies on statistical graphics. Exploratory Data Analysis (or EDA, by its acronym) lets any analyst condense the features of a new data source, exploring the information without rigid models to fit the data in, and allowing to formulate hypothesis which could unveil hidden insights. Some of the new graphics and charts proposed in *Tukey*'s book are: the box-and-whiskers plot (also known as boxplot), the stem and leaf diagrams to display distributions and rootograms. Through this doctoral dissertation, we exhibit a strong influence of Tukey's work in some approaches to deal with LOD visualization. Besides his remarkable contributions to the statistical visualization field, *John W. Tukey* is one of the best statisticians of his time, as some of his most notable works record: the development of the Fast Fourier Transformation algorithm, invention of different statistical tests (Tukey's range test, Tukey lambda distribution, Tukey's test of additivity, among others), coined the term "*bit*" (binary digit) and possibly was the first one to use the word *software*.

2.1.1.3 **21st century**

Currently there are a lot of interesting developments in the visualization field, specially when is used to support other activities, contributing to make the messages more attractive and helping towards reaching a wider audience. On this section we present some of the most relevant people and works that are still active in the visualization field (even if their biggest contributions were made in the past century).

In 1983, **Edward R. Tufte** (born 1942) printed his groundbreaking and influential book on information design and visual literacy: "*The Visual Display of Quantitative Information*". His commitment to improve the way information was transmitted using visualizations, thinking about how to generate correct representations, omitting irrelevant elements and avoiding biased presentations of facts, have praised *Edward Tufte* as an expert in data visualization. Tufte's many courses, invited talks and years of teaching have instructed new generations of data visualisers, from business managers to data

journalists, democratising knowledge representation and encouraging its adoption by practitioners.

Ben Shneiderman (1947) is an American computer scientist whose contributions to the *Human-Computer Interaction* field (HCI) have improved the way we work with User Interfaces. Regarding Information Visualization, he was the first one to come up with the concept of *treemaps* in order to visualise hierarchical data. Shneiderman is also the author of the *Visual Information Seeking mantra*, usually summarised as: **Overview first, zoom and filter, then details on demand**, a visualization approach that can be found in many present tools and which inspires our proposal, developed through this dissertation.

In most recent years, **Mike Bostock** can be considered one of the major game changers in the visualization field. Mike is the creator of *d3.js*[1] (or D3, standing for Data-Driven Documents), a JavaScript library to create interactive visualizations for web browsers, taking the advantages of different web standards and technologies. He also worked as the graphics editor for *The New York Times*, but is currently devoted to the D3 community. The opportunity to manipulate data directly on the browser has encouraged many visualizers to publish their representations and works, expounding them to public scrutiny and bringing new ideas to practitioners. *D3* is a successor to *Protovis*, a pioneering visualization tool for web browsers. One of the key items for *D3*'s success relies on its ability to generate any visual representation and animate its transitions, as it only allows to describe the behaviour of primitive visual elements, in opposition to the pre-defined templates of other libraries (such as those provided by the well known *Google Charts*[2]).

Finally, we would like to briefly acknowledge some important names in the application and promotion of visualization in different areas: *Stephen Few* (writer and expert in information design and communication), *Alberto Cairo* (data journalism), *Ola Rosling* (visual storytelling), *Fernanda Viegas* and *Martin M. Wattenberg* (promoters of Google's *Knowledge Graph*, an innovative work in visual information architecture, and also known for their

[1] `https://d3js.org`
[2] `https://developers.google.com/chart`

works in the artistical dimension of data visualization), *Nathan Yau* (blogger in statistical visualization), *Katy Börner* (science and academic mapping) and many others.

2.2 LOD visualization tools

RDF's data model is based upon the concept of publishing statements about different resources, in the fashion of *subject-predicate-object* expressions (triples). Thus, the most basic approaches towards visualizing LOD datasets present the information in tabular format, such as:

- **Pubby**: Pubby (Cyganiak and Bizer, 2008) provides a LD interface for SPARQL endpoints, taking care of *HTTP 303 redirects*[1] and content negotiation. This project gained massive attention within the LD community as the *de-facto* fashion to initially access and navigate a SPARQL endpoint, rendering the properties and values of a particular resource in tabular format with a default recognisable green-ish interface.

- **brwsr**: This lightweight Linked Data browser was developed within the *Data2Semantics* project[2], and is heavily inspired by *Pubby*, with a cleaner interface and more configuration options.

Even though tables and lists can be considered text-based representation techniques, constituting a familiar rendering for spreadsheets and relational database dumps, for the rest of this dissertation we will use the term *visualization* to make reference to graphical depictions of the data, such as charts, diagrams, plots and others.

After a few years from the Semantic Web's definition, a stage during which many tools were designed and developed to provide the infrastructure to realise Tim Berners-Lee's vision, a seminal survey (Dadzie and Rowe, 2011) collected all the visualization tools available at the moment, presenting the most complete State of the Art on LOD exploration. They focused on LD browsers,

[1] https://tools.ietf.org/html/rfc7231#section-6.4.4
[2] http://data2semantics.github.io

distinguishing between those only providing text rendering, and those able to generate graphics. This section updates the analysis of existing tools, gathering the implementations since *Dadzie* and *Rowe*'s survey, and omitting the works which have been already discontinued (despite the popularity some of them reached during the consolidation stage of the LD community).

During this study, we will categorise LOD visualization tools between those tailored to fit a particular purpose or domain, and the ones which are able to adapt to new environments and ontologies.

2.2.1 Tools customised for a specific scenario

This section groups tools which have been designed and implemented to generate graphical representation for a particular ontology or knowledge field. The main advantage of works taking this approach is that by knowing beforehand the scenario data belongs to, visual solutions can be greatly refined, offering the best navigation and knowledge extraction experiences. However, our research goals are focused on providing a generic way to approach LOD datasets, regardless of the ontologies used, or the corresponding knowledge areas.

On the following subsections, some examples and prototypes implementing a customised approach are analysed.

2.2.1.1 Map4RDF

Map4RDF (de Leon et al., 2012) is a faceted browser to explore and interact with datasets in the geospatial domain. The navigation interface allows to filter the rendered resources by facets, so that end users can control what type of entities are being pin-pointed on a map (see Figure 2.7). Map4RDF is highly configurable, allowing to choose the preferred base layer templates (Google Maps, Open Street Maps and so on), and rendering diverse geographical features over them, e.g., points, lines and polygons.

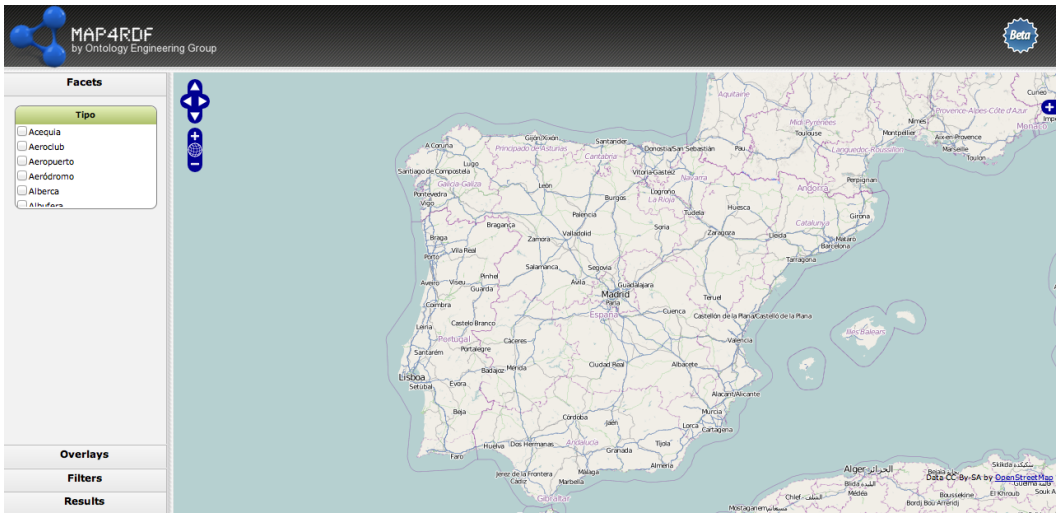


Figure 2.7: Initial display of Map4RDF, with the facet filtering widget on the left side of the display. Adding new layers, each with its filters, allows to create complex map representations.

2.2.1.2 **SWJ's scientometrics portal**

The Semantic Web Journal by IOS Press[1] is an indexed publication to address the efforts on information access and sharing within the Semantic Web field. It characterises for its open and transparent review process, making accessible online all the data about each submitted manuscript. All these facts were annotated semantically and exposed through a SPARQL endpoint, which was later used to develop a LD based scientometrics portal (Hu et al., 2013) that allows to explore all the data using visualizations. The scientometrics portal displays extended information about each author, publications each one has contributed to, how they are related, topics and collaborations at country level, and so on (Figure 2.8).

A very similar approach is followed by the *DEKDIV* visualization tool (Hu et al., 2014), applied to the LAK (Learning Analytics and Knowledge) Challenge's dataset.

[1] http://semantic-web-journal.net

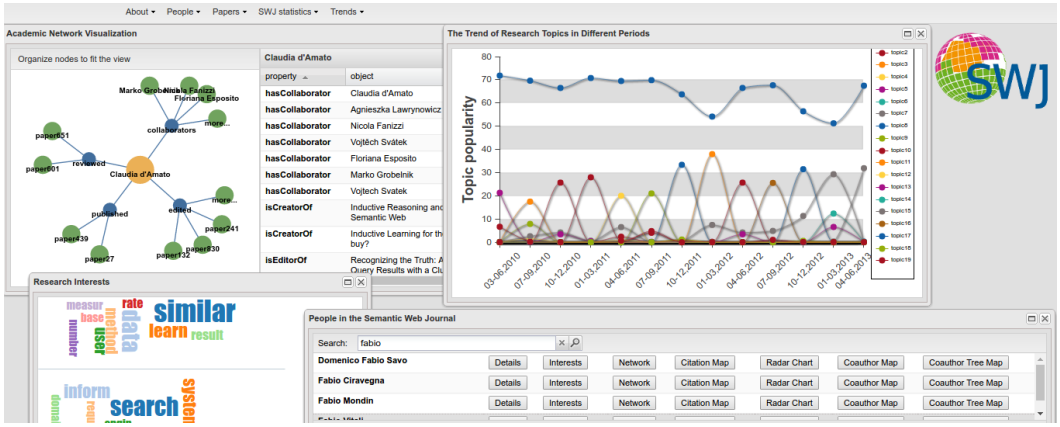


Figure 2.8: Some of the widgets available on the Semantic Web Journal's scientometrics portal. Users can consult data about the most used topics, the collaboration networks between researchers and so on.

2.2.1.3 LOD/VizSuite

Researchers from the iMinds Digital Research Centre[1] designed, implemented and evaluated an interactive visual workflow to explore LOD. Their approach narrows the dataset from high level group overviews towards more detailed views. Providing a general overview the dataset reveals its underlying structure, and internal relationships are exhibited as the users go in depth on the contents. This tool is named LOD Visualization Suite (De Vocht et al., 2014).

After the data narrowing stage, a coordinated view is presented in order to allow exploratory searches. This set of actions is focused to lead to other datasets through links should they prove relevant enough. Broadening (the discovery of related datasets through links in the dataset) is provided by the ResXplorer tool (De Vocht et al., 2013).

In order to demonstrate the workflow's implementation, an online application was made available[2] to explore the *Research Information Linked Open Data* (RILOD) dataset, a repository consisting of heterogeneous resources related with the research landscape within the region of Flanders. Figure 2.9 shows the connections between Flemish researchers, providing a sidebar with the topics covered by the selected authors in the graph.

[1] http://iminds.be
[2] http://ewi.mmlab.be/academic

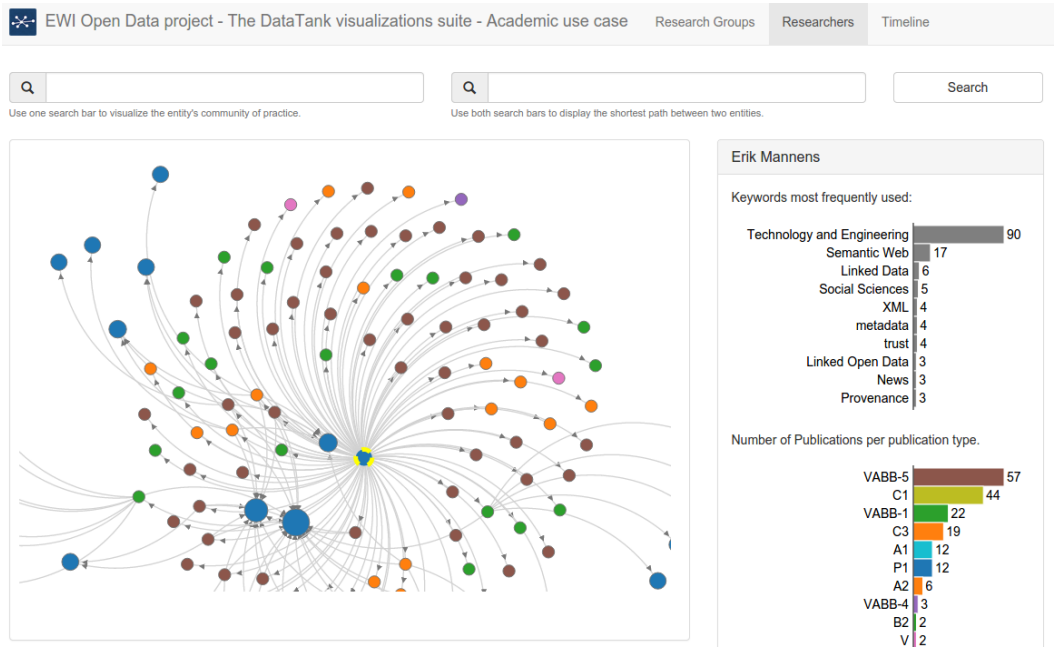


Figure 2.9: LOD/VizSuite's interface displaying the co-authorship network between Flemish researchers through a node-link graph representation.

2.2.1.4 CODE

Within the EU-funded CODE research project[1], its *Visualization Wizard* tool (Mutlu et al., 2013, 2014) provides a platform to visualise academic publications data extracted by other tools in the CODE project. These extracted facts are published using the RDF Data Cube Vocabulary (DCV), a W3C standard to represent statistical data in RDF. Depending on the nature of the data to be depicted, those visual representations which do not fit will not be selectable. This visualization recommender can be improved by implementing new *generators*, well-defined interfaces with the ability to map data when plugged to the Visualization Wizard. Users can interact with the framework by changing how data dimensions are mapped, changing the visual components they refer to and updating the visualizations accordingly.

[1] http://code-research.eu

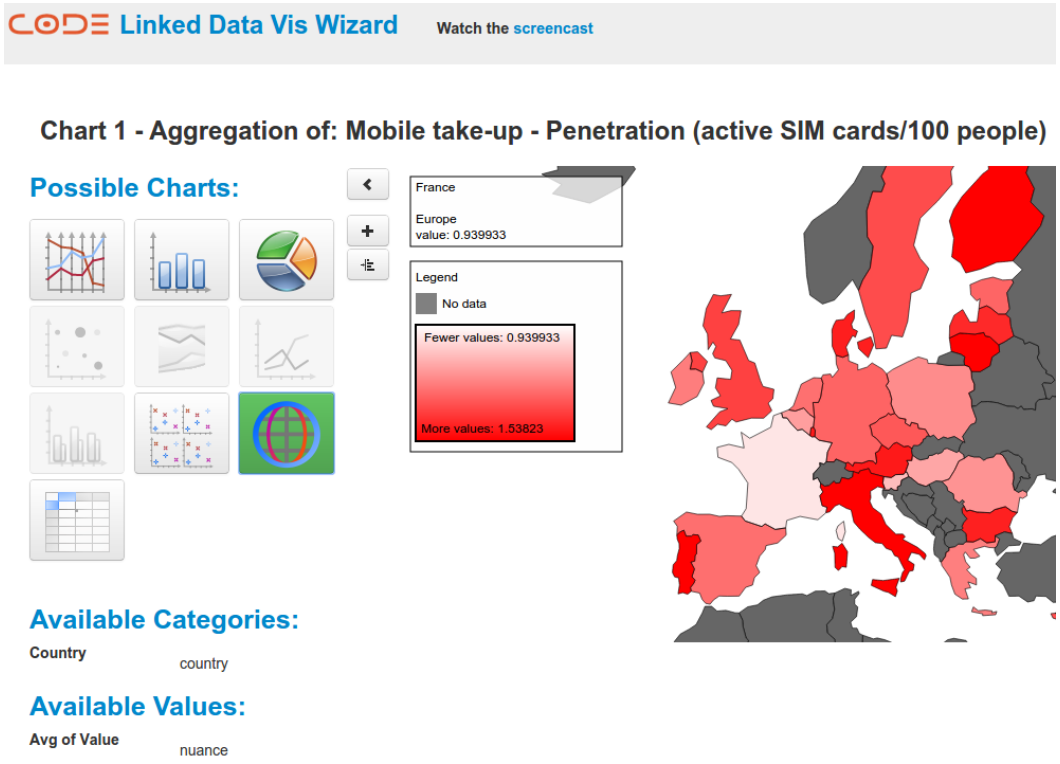


Figure 2.10: CODE Visualization Wizard, plotting the selected features over a map, offering other representations that might fit the data.

2.2.2 Generic and domain agnostic approaches

The aforementioned visualization tools provide a great starting point to distance from more traditional ways of presenting LOD, such as plain tabular formats (Cyganiak and Bizer, 2008), or node-link graphs (Deligiannidis et al., 2007; Hoefler et al., 2013). Nevertheless, in order to promote the consumption of LOD, whatever its topic might be, we support the development of generic visualization tools able to deal with different datasets, from a wide variety of knowledge areas. The more independent the tools are, the better they are able to adapt to datasets that have not been published yet, and whose data structure is not defined.

Next, we present some of the tools taking a domain agnostic approach.

2.2.2.1 **SemLens**

SemLens (Heim et al., 2011) is a visual tool to analyse semantic data retrieved through SPARQL queries. The user can later arrange the data in a scatter plot, selecting which properties are going to be mapped for each axis. After the chart is generated, they are allowed to draw Semantic Lenses over the points. This "magic-lenses" (Bier et al., 1993) are defined by a semantic filter, which will discard the properties' values which do not fit the specification. The semantic lenses can be of any desired size within the chart, and will differentiate the scatter plot's points between those which meet the condition within the lense's area, and those that do not. The main limitation of SemLens is that it only provides one graphical representation (scatter plots) with the extended feature of lenses filtering. The user is held responsible to select which properties are used for each axis, and no restrictions are applied, thus trying to generate a scatter plot even when the scenario makes no sense at all.

2.2.2.2 **LDVizWiz**

With the goal of providing general purpose visualizations of any SPARQL endpoint, LDVizWiz (Atemezing and Troncy, 2014) inspects the features of the dataset to understand the underlying data and inner structure. To that end, it constructs specific SPARQL queries to detect the presence or absence of predefined types of information. In particular, LDVizWiz looks for data belonging to any of these categories: Geography, Temporal, Event, Agent/Person, Organization, Statistics and Knowledge. To provide this classification, it queries for particular classes and properties explicitly, if a *Person/Agent* instance is not defined by the *FOAF* (Friend-of-a-Friend) ontology, it would not be categorised accordingly. As LOD takes an Open-World vision (more on Section 3.1), data publishers are allowed to use and describe resources using any vocabulary of their preference, which could cause LDVizWiz failing to recognise the category.

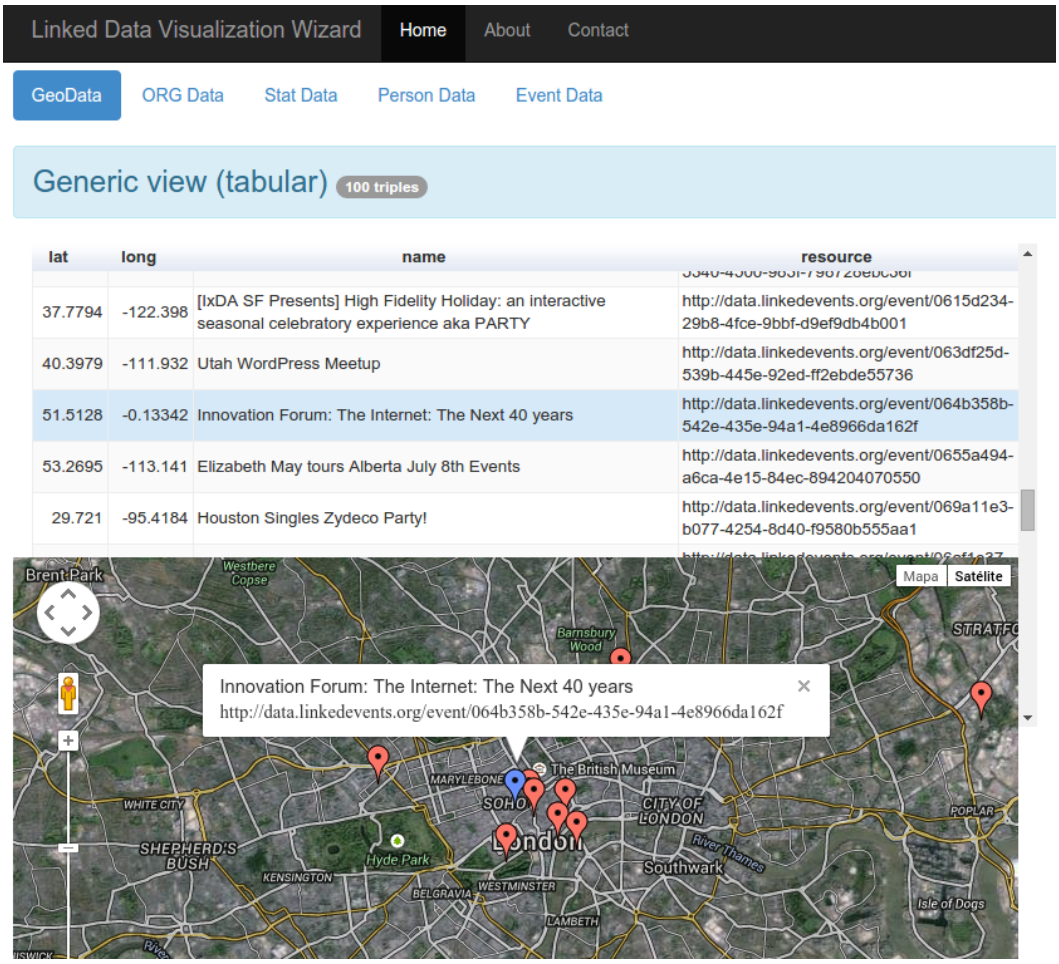


Figure 2.11: LDVizWiz's detection of geographical features within a dataset also generates a map-based representation, taking the data from the table records listed above.

2.2.2.3 LODVisualization

LODVisualization is a demo tool developed to show the implementation of the Linked Data Visualization Model, abridged LDVM (Brunetti et al., 2013), which lets users to relate different data sources in a dynamic fashion using visualizations. LODVisualization relies on the conceptual Data State Reference Model's framework (Chi, 2000) to generate visual representations from raw data in a pipelined process (see Figure 2.12). To implement the first visualization task proposed by Ben Shneiderman (Shneiderman, 1996) (*Overview*,

a stage aimed to get an overall impression of the entire collection), LODVisualization visualizes the hierarchical class and properties within a SPARQL endpoint. Users can also consult the properties shared between two classes, those instances with the highest in/out-degree values (incoming/outgoing links) and the like. LODVisualization can render treemaps, tables and bar charts from the data retrieved from any SPARQL endpoint, without the need of individual configuration schemas.

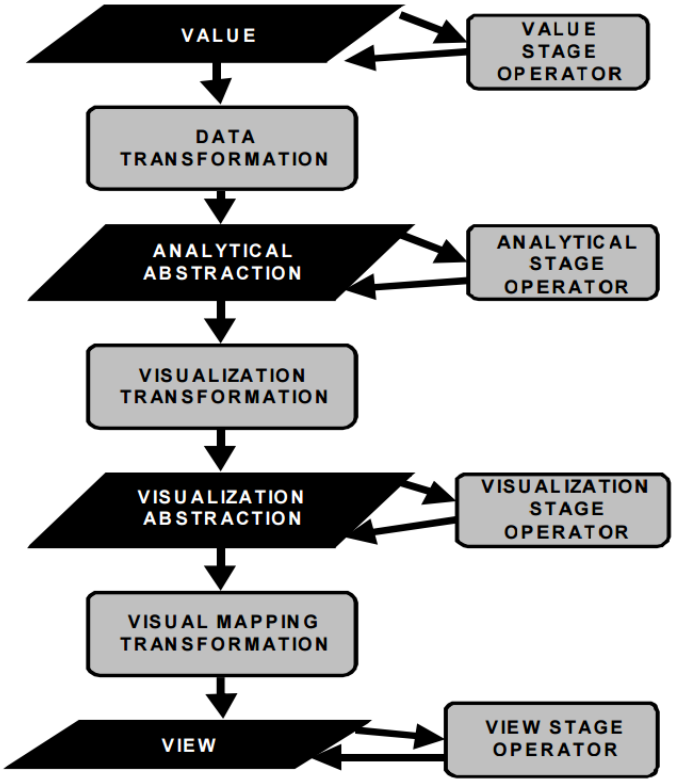


Figure 2.12: Ed Chi's Data State Reference Model, a data visualization pipeline which goes from raw data to visual representations by applying different transformations in each stage.

2.2.2.4 **Payola**

Payola (Klímek et al., 2013) is a web framework to analyse and visualize Linked Open Data, enabling users to build their own instance of LDVM's pipelines. An implementation prototype adapted for the Czech LOD ecosystem was developed in order to explore relevant data for Czechoslovakians

(Klímek et al., 2014), such as public inspections and sanctions as depicted in Figure 2.13. The first step is to select the data to be analysed, either by choosing the desired SPARQL endpoints, or by providing the input RDF files. Then, the user can perform different operations in order to analyse the data. Finally, the visualization abstraction is estimated, and later mapped into final views (visualizations) on a web browser. Technical users can also improve the process by adding plugins, thus obtaining more refined outcomes.

Collaboration between users is encouraged, as visualizations, plugins and operators can be shared within the platform. This does not only let users to re-run experiments, but to connect existing plugins and operators to new datasets and workflows, producing new enriched views of the data.

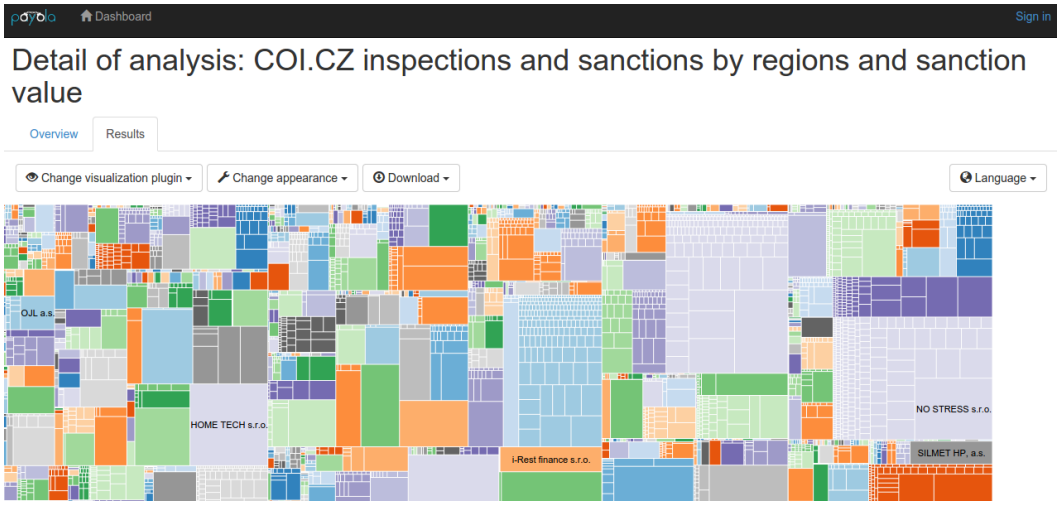


Figure 2.13: TreeMap visualization of Czech's public inspections and sanctions data in Payola.

2.2.2.5 rdf:SynopsViz

The idea behind rdf:SynopsViz's development (Bikakis et al., 2014) is to provide hierarchical charting and exploration of LOD. Its approach relies on metadata visualization, computing aggregation values for the properties within a dataset, e.g., number of triples, variance and mean values and so forth. LOD exploration is mainly based on facet filtering, as the user selects the dimensions to be plotted by rdf:SynopsViz. The facet navigation and fil-

tering is available for the dataset's classes, numerical and date properties. If an area of special interest is zoomed in, more detailed information is provided.

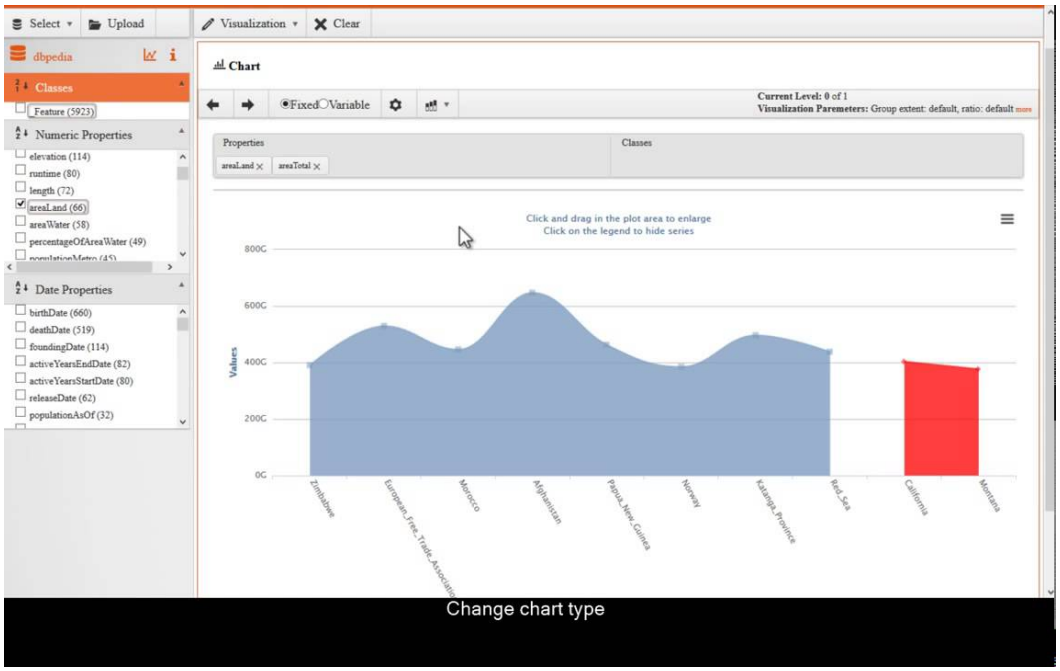


Figure 2.14: Once the data is facet filtered in rdf:SynopsViz it is depicted using pre-defined charts.

2.2.2.6 **Sgvizler**

Sgvizler (Skjæveland, 2012) is a JavaScript wrapper which allows to display in a graphical manner the results of a SPARQL query. To render the charts, Sgvizler makes use of HTML5's *data-* prefixed attributes, in which the user will encode the configuration options of the drawing. As all the programming is made on the client's JavaScript console, the target SPARQL endpoint must be CORS (Cross-Origin Resource Sharing) enabled, as no results would be returned otherwise. The greatest drawback of Sgvizler is that it is intended for technical users, as they need to access the source code of the website to include the extra code within the HTML, and requires a basic understanding of the SPARQL language in order to formulate the queries.

Query: *The total production of oil, gas, condensate, NGL and water per year on Ekofisk.*

The year is set as the first column, becoming the X-axis value (remember to sort), and the remaining result set columns become the columns in the chart.

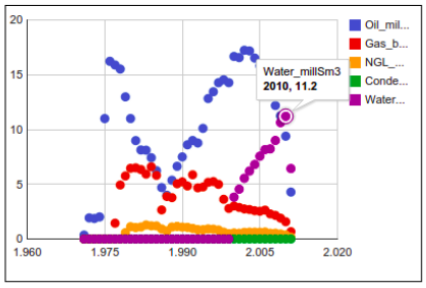
Note that we cast the year to integer (in the dataset it is a string) in order to have the column on the correct format.

HTML

```
<div id="ex"
   data-sgvizler-endpoint="http://sws.ifi.uio.no/sparql/npd"
   data-sgvizler-query="
        SELECT xsd:int(?year) ?Oil_millSm3 ?Gas_billSm3 ?NGL_millSm3 ?Condensate_millSm3 ?Water_millSm3
        { ?period a npdv:FieldProductionPeriod ;
              npdv:hasField &lt;http://sws.ifi.uio.no/npd/field/Ekofisk&gt; ;
              npdv:year ?year ;
              npdv:producedNetOilMillSm3 ?Oil_millSm3 ;
              npdv:producedNetGasBillSm3 ?Gas_billSm3 ;
              npdv:producedNetNGLMillSm3 ?NGL_millSm3 ;
              npdv:producedNetCondensateMillSm3 ?Condensate_millSm3 ;
              npdv:producedWaterMillSm3 ?Water_millSm3 ;
              OPTIONAL{ ?period npdv:month ?month } .
              FILTER (!bound(?month))
         } ORDER BY ?year"
   data-sgvizler-chart="google.visualization.ScatterChart"
   style="width:400px; height:265px; border:1px solid black; display: inline-block;"></div>
```

Result

The result of the HTML above (left), and a google.visualization.Table rendering of the same query (right).



.o	Oil_millSm3	Gas_billSm3	NGL_millSm3	C
1971	0.357	0	0	
1972	1.927	0	0	
1973	1.87	0	0	
1974	2.014	0	0	
1975	10.995	0	0	
1976	16.227	0	0	
1977	15.888	1.441	0	
1978	15.522	4.931	0	
1979	12.978	5.759	0.577	
1980	10.999	6.455	1.118	
1981	8.986	6.497	1.042	

Figure 2.15: Sgvizler's allows to insert SPARQL queries inside a webpage's DOM structure (top) and render its output in both graphical and tabular formats (bottom).

2.2.2.7 VisualBox

In a similar fashion to Sgvizler, Visualbox (Graves, 2013) requires that users have a strong technical background together with basic notions of the Semantic Web. Visualbox provides various features in a single platform: SPARQL syntax highlighting to detect query formulation errors, endpoint connection to retrieve data and customization of the resulting visualizations using pre-defined templates. Visualbox relies on LODSPeaKr[1], a framework to develop LOD-based applications created by the same author. In order to visualise data, Visualbox provides a custom template language (based on the one used by

[1] https://alangrafu.github.io/lodspeakr

Django[1], a popular web framework for the Python language[2]) that is substituted in runtime with a visual encoding of the retrieved values.

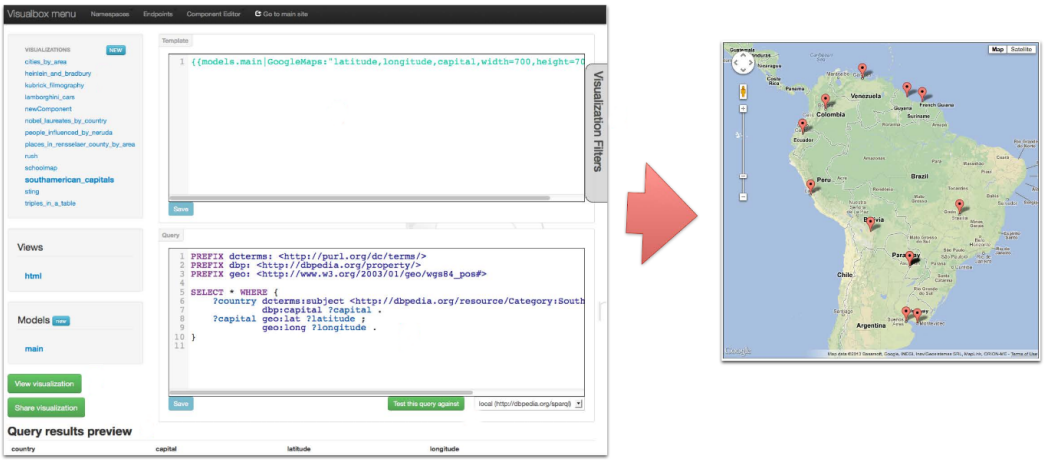


Figure 2.16: Visualbox provides a custom templating language to display the visual output of SPARQL queries.

2.2.2.8 **VizBoard**

Vizboard (Voigt et al., 2013a) is a visualization tool built on top of the CRUISe platform (Pietschmann et al., 2009), acting as a mash-up tool where users are able to combine different dimensions and create insightful visualizations, allowing any user to understand the whole picture of a dataset. The actions performed in any of VizBoard's panels automatically update the rest. Interaction with the platform is heavily based on facet filtering, letting users focus on the most relevant data for their analysis purposes, keeping non-desired data out of scope. VizBoard publishes all the information on visualization rendering using VISO (VISualization Ontology) (Polowinski and Voigt, 2013), a multimodel vocabulary which describes all the concepts and relations within the visualization field.

[1] https://djangoproject.com
[2] https://python.org

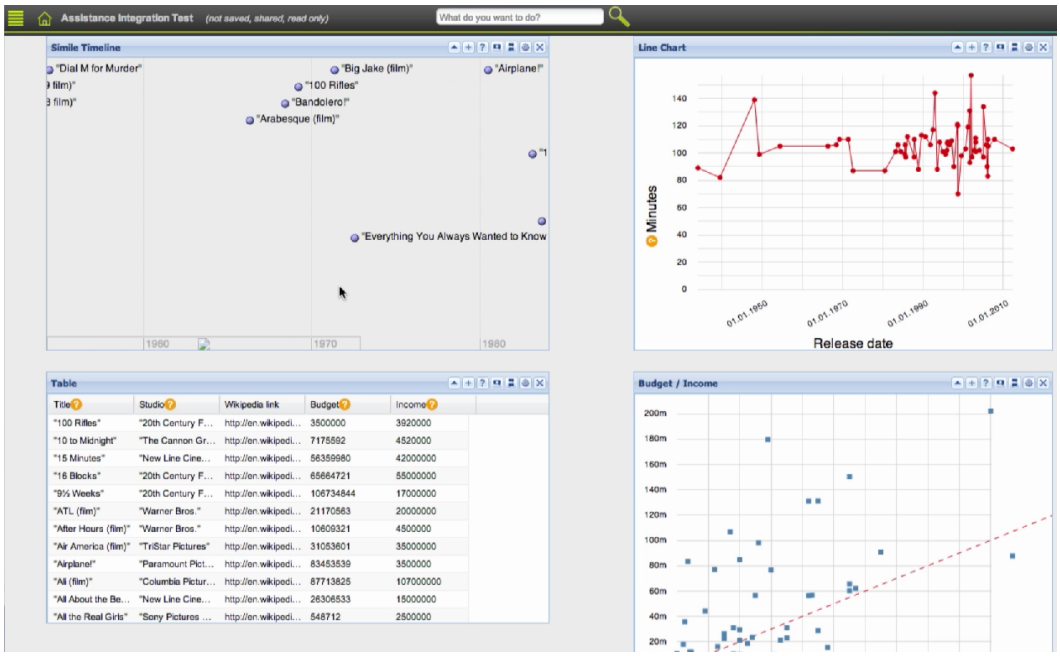


Figure 2.17: VizBoard offers different panels which encode data's features, hoping to provide insights from an analyst's perspective.

2.2.3 Conclusions

The works and prototypes listed before are the best examples of available LOD visualization tools. Some of them might still be under active development at the time of writing this dissertation, but we will only analyse structural features of the tools, which are not likely to experience great modifications in the near future.

The first task to evaluate is the datatype support of all the tools analysed during the State of the Art review. Ben Shneiderman proposed seven datatypes a singular datum could be classified as (Shneiderman, 1996), limiting the usage and operators applicable to each dimension. According to his taxonomy, a datum could be: uni-dimensional, bi-dimensional (or planar), three-dimensional (or volumetric), multi-dimensional (for more than 3 dimensions), temporal, tree-structured (hierarchical) and network-like. Table 2.1 summarises the datatype support of each one of the tools previously analysed.

	1D	2D	3D	Multi-dimensional	Temporal	Tree	Network
Map4RDF		✓					
SWJ's portal	✓				✓		✓
LOD/VizSuite					✓		✓
CODE		✓		✓			✓
SemLens		✓					
LDVizWiz		✓		✓	✓		✓
LODVisualization		✓				✓	
Payola		✓		✓		✓	✓
rdf:SynopsViz				✓	✓	✓	
Sgvizler	✓	✓		✓	✓	✓	✓
VisualBox		✓		✓	✓	✓	✓
VizBoard				✓	✓		✓

Table 2.1: Analysed LOD visualization tools support of Shneiderman's seven basic datatypes.

However, these seven datatypes might be misleading, generating trouble when we need to determine the classification of an uncertain feature: Are maps the only examples of bi-dimensional datatypes? Could tree structures be a sub-type of networks? To avoid issues when assigning categories to a dataset's dimensions, we propose a curated set of datatypes in Section 3.3. To guarantee a good coverage of distinct datatypes should fulfil the set goal of representing each data dimension as best as possible.

We also established the goal to design a usable approach, despite the technical background and skills of the end-users. A.-S. Dadzie and M. Rowe identified three main types of users attending to their skills (Dadzie and Rowe, 2011), which could be categorised as *lay-users*, *tech-users* and *domain experts*. We have adapted the definition of each group to reflect the background of web users nowadays, which could become potential targets of LOD visualization tools.

- **Lay-users**: This category is formed by those who are able to navigate through websites and find information using traditional search engines, therefore constituting the largest user-group of the three. *Lay-users* might have a temporary interest in the data they explore, usually without deep knowledge on the field of the study and lacking analytical skills. Thus their approach is more an exploratory one, with little effort required to satisfy their needs.

- **Tech-users**: This profile is aware of the technologies and concepts related to the Semantic Web, and have worked with data published using any RDF notation. They might lack analytical skills, but they are able to extract and filter data in more advanced ways than using search engines, taking advantage of LOD publishing.

- **Domain experts**: The last category makes reference to those users which might not be aware of the Semantic Web, but have a deep and broad experience in a particular knowledge field. When lacking a domain specific tool tailored for analysing the dataset, expert users might benefit from the generic workflow of our proposal, using the extra information available through semantic annotations to draft the first stages of a suitable exploratory path.

	Lay-users	**Technical users**	**Domain experts**
Map4RDF		✓	✓
SWJ's portal	✓		
LOD/VizSuite	✓		✓
CODE	✓	✓	✓
SemLens	✓		
LDVizWiz	✓		✓
LODVisualization	✓		
Payola	✓	✓	✓
rdf:SynopsViz			✓
Sgvizler		✓	✓
VisualBox		✓	
VizBoard	✓		

Table 2.2: Target users by the analysed LOD visualization tools

The LOD visualization tools reviewed in this section provide different compliance levels with these three profiles, as exposed in Table 2.2. Only two of the selected tools (CODE and Payola) are intended to satisfy all three user categories. However, according to Table 2.1, both tools are not capable to represent some datatypes, being of special importance the lack of support of temporal data. This might be a direct consequence of the origins of each tool: for example, *CODE* is primarily designed to work with academic publications data, whereas *Payola* is more oriented to public administrations.

Therefore, through this dissertation we will develop the required modules to explore LOD datasets with no prior knowledge required (both about their domain or the specificities of the Semantic Web), adopting a data-driven approach which adapts to new datasets and descriptions without the need of external involvement. The envisaged modules will form a visualization pipeline: from accessing SPARQL endpoints to retrieve input data, to the generation of the most suitable visualizations as output. The implemented modules will be connected in a LOD visualization prototype, in order to verify its purpose with real users in exploratory data analysis scenarios.

Our primary goal is set on providing a valid solution for each of the following matters, that any LOD visualization pipeline should be able to answer:

- What are the facts and features within a dataset intended to model?
- How could end users get the most of semantically annotated data using visualizations?

Together with the design of an approach to address the previous issues, during our research we will make minor contributions to other LOD visualization related tasks, such as concept relevance identification, LOD metadata extraction and visualization knowledge reusability.

2.3 Summary

In this chapter we have introduced the Data Visualization field by presenting an historical overview, highlighting the most relevant developments in visualization science and listing some of the best representatives of the field. Despite the fact that humans have been depicting the world since ancient times, the theoretical background of information visualization was set nearly two centuries ago, and many charts and graphics did not become broadly used until recently. Nowadays, visualization is perceived as a supplementary task to other fields, providing support to analyse and communicate facts in areas such as journalism, governance, statistics, scientific fields and so on.

After the background description, we present the current State of the Art in LOD visualization, listing and analysing the tools developed to ease the access to the Semantic Web using visual means. Most research on LOD visualization has resulted from the need to visually represent features in specific domains, implementing visualization prototypes that are tightly coupled to be used in very particular scenarios. However, some of these tools were conceived and designed to work with different data sources, being able to take a generic approach. These are the ones we have most focused on, as our research goals have established the design of a visualization prototype able to work with a variety of data topics.

In an extreme view, the world can be seen as only connections, nothing else. We think of a dictionary as the repository of meaning, but it defines words only in terms of other words. I liked the idea that a piece of information is really defined only by what it's related to, and how it's related. There really is little else to meaning. The structure is everything.

"Weaving the Web", Tim Berners-Lee (1999)

CHAPTER 3

Metadata and structure of LOD datasets

The Linked Data principles are all about publishing structured machine-readable data so it can be connected (linked) with related resources over the Internet. Whilst working with standard Web technologies, each resource is assigned a unique identifier within a single, global information space, favouring the discoverability of assets.

Linked Data allows the integration of heterogeneous data sources on an unprecedented scale, providing access to a wide range of sources (Bizer et al., 2009a). By describing all resources using shared schemas, machines are able to *understand* what each instance is about, and how it is related with others.

The more datasets are published as LOD and the greater detailed they are, the harder it gets to focus on the specifics and more relevant aspects of the data. This concept is known as *Information Overload* (Eppler and Mengis, 2004), and deals with the challenges to understand and make decisions when many facts are presented at once. The inaccuracies and contradictions which might appear in the data, or the impossibility to compare and summarise

datasets do nothing but aggravate this sensation. In the following sections, we propose the combination of different approaches to deal with LOD.

First, we will analyse which role LOD plays in the global data space, and the foundations of the Semantic Web. Later, our proposal will try to infer the structural layout of the data, extracting metadata metrics which could lead to a standardised characterisation of datasets. Categorising and ranking data based on intrinsic features also permits to identify the most important resources and entities of each dataset, offering a useful and simple approach to order and summarise data presentation.

3.1 Open versus Closed World Assumptions

One of the main differences between the Semantic Web's approach towards modelling data and other structured data formats when dealing with knowledge representation is that the former takes the Open World Assumption (OWA), in opposition to the Closed World Assumption (CWA) followed by the latter.

We, humans, are used to think about real world entities as *containers with features*. At the same time, *containers* can be composed of a set of *components*, which could act as containers for other sub-elements and have custom features on their own as well. Finally, we categorise similar entities in groups that share some characteristics, creating hierarchical structures that connect our perspective of the environment. For example, should we be required to explain what a cat is, we first think about an animal (high-level entity) with some specific characteristics, and formed by different constituent parts: eyes which are different from other animals, specially adapted to see in dark scenarios; legs ended in paws, each of them having a limited number of toes and so on. We go through similar processes when picturing a car, a country, an individual academic course, the full procedure to obtain a PhD, and so forth.

This approach for knowledge representation is common practice amongst information architects. *Object-Oriented programming* (OOP) is a programming paradigm to encode real world objects using *classes* (containers) and

attributes (features), which can later be used in computer programs and algorithms. The *Relational model* for databases (Codd, 1970) also uses this technique to store information. The CWA presumes that all the features are well known and controlled by the information system, so new instances just need to fill the entity's template with their own values. In the CWA, datasets are supposed to be complete, so if any value is not known to be true, it must be false.

These simplifications of the world allow to represent the information we collect and store, process, manipulate and reuse it to increase our knowledge. Often, the better the entities are modelled, the more advantageous they become to understand their behaviour. However, some situations exist in which a full description of an entity is not possible:

- Some features are not feasible to be measured in a reliable fashion.
- There might be some unknown facts about the entity, making them ineligible to be represented.
- Any item can become as complex as we want it to be: over-simplifying it will not allow for deep insights about the objects, and an extremely detailed description would become really difficult to manage effectively.

Given these environments, the OWA takes for granted that no single agent has complete knowledge about every entity, therefore being unable to adopt CWA's perspective. This is the exact scenario for the Web: a global system with incomplete information. As stated by the OWA, if any value is not known to be true, it is unknown. The absence of a statement simply means that it has not been made explicit yet, and might never be. This uncertainty does not constitute a drawback compared to the previous approach, instead, Linked Data assumes that data structures could be created, removed, updated to provide more detailed entities, simplified, merged and the like.

To publish, store or query data following the CWA, we must know exactly which is the schema followed by each entity, whereas under the OWA we find lesser restrictions. Any authorised body is free to publish different statements about the same resource, thus contributing towards a global knowledge

information base. Nevertheless, we need to follow some rules when publishing new facts, in order to enable their exploration by third parties. Such rules should define which terms are available to describe resources, and how those terms relate. This can be achieved using simple controlled vocabularies (known as taxonomies), or more complex logical models to fit a certain domain: ontologies.

In summary, CWA is a perfect approach to describe constraints and validate data; whereas OWA lets to represent knowledge that can be later extended or reviewed.

3.2 Data discovery by exploratory analysis

Each time we face a data source we have not previously worked with, or which covers topics outside our expertise fields, we need to answer a simple, naive question: **What is this data about?**

Due to the nature of Linked Open Data, we can not consult a unique, centralised knowledge repository where the exact structure of each dataset is detailed. Different schemas (ontologies) are available to be used in data descriptions, and few data providers pay attention to the quality of their information, which makes some datasets difficult to work with. Moreover, more information can be modelled at any time, extending the descriptions originally included within the dataset. Taking all these facts into account, there are only two features left that will always be present and unaltered in LOD: the data itself, and the semantic descriptions of such data.

In the Semantic Web, resources are described using statements that are encoded as RDF triples, each of them composed by a **subject**, a **predicate** and an **object**. Reading a singular triple, we can make a very basic sentence about the target resource, for example: *Tim Berners-Lee is the creator of the World Wide Web*. Adding new triples with the same *subject* delves into its description, providing a more detailed view. A collection of RDF triples (like the one listed in Code 3.1) is known as a RDF graph, and can be illustrated as a directed arc diagram. This simple example is depicted in Figure 3.1, where *resources* are drawn as ovals, literal *objects* are represented using rectangles,

and the connections between them are depicted using arrows (the arrow head displaying the direction of the relationship).

```
@base <http://dbpedia.org/resource/> .

@prefix dbo:    <http://dbpedia.org/ontology/> .
@prefix dbp:    <http://dbpedia.org/property/> .
@prefix foaf:   <http://xmlns.com/foaf/0.1/> .
@prefix rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

<Tim_Berners-Lee>   rdf:type         dbo:Scientist ;
                    dbp:birthName    "Timothy John Berners-Lee" ;
                    dbo:birthDate    "1955-06-08"^^xsd:date ;
                    dbo:birthPlace   <England>, <London>.

<England>   dbp:capital     <London> .

<London>    dbo:isPartOf             <England> ;
            dbp:subdivisionName      <England> .
```

Code 3.1: RDF example triples using turtle notation

The triples listed in Code 3.1 are a brief summary about the resource *Tim Berners-Lee* as collected by the English chapter of DBpedia[1], a semantic dump of Wikipedia's infoboxes which acts as the *de-facto* reference dataset within the LOD Cloud (Auer et al., 2007; Bizer et al., 2009b; Lehmann et al., 2015). Due to the size and topic coverage of the Wikipedia, its semantically annotated counterpart has become the most referenced dataset when talking and writing about LOD. In its latest version (as of July 2016), descriptions for more than 38.3 million things are stored in DBpedia. Going back to the simplified excerpt about *Tim Berners-Lee*, it can be understood that:

- The individual instance describing *Tim Berners-Lee* acts as the **subject** of most triples. This resource is referenced by a unique HTTP IRI (Internationalised Resource Identifier), so that any other instance can create links to and from it. In order to abbreviate IRIs, common

[1] http://dbpedia.org

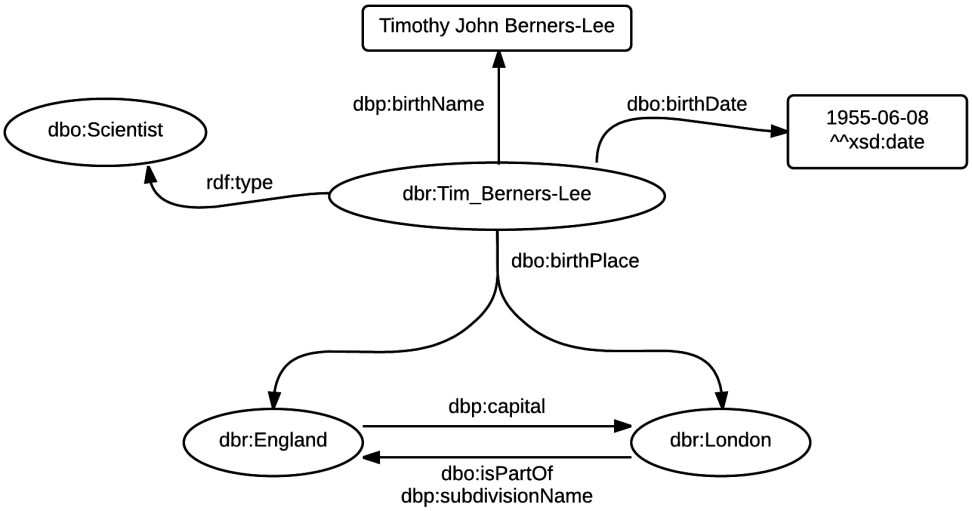


Figure 3.1: Graph exhibiting the entities and connections represented using the Resource Description Framework in Code 3.1.

prefixes are extracted and shortened, creating namespaces that improve readability, e.g., the *http://dbpedia.org/ontology/* IRI is used for all resources which are managed by DBpedia. In order to ease readability and reusability, the *dbo* prefix is used, establishing a namespace which can be used during the querying stage.

- **Properties** list the attributes of those subjects, that is, what can be said about each individual instance. They are always expressed by a HTTP IRI, and can be looked up in order to obtain extended information about their usage.

- Finally, **objects** constitute the last component of the triples. They make reference to the value assigned to a given instance for a particular feature. It can adopt a *literal* value, or lead to another resource's IRI, characteristic that sets the roots of the **Linked** nature of LOD.

OWL[1] ontologies allow to declare ranges and domains for each property. *Domains* point out all the classes that properties are intended for, that is, the **subjects** they describe should belong to the expressed ontological types; whereas *ranges* restrict the set of datatypes **objects** can adopt. Depending on these datatypes, properties fall within one of the following categories:

- *Object properties*: Relate two resource individuals, each of them belonging to a specific ontological class. This feature enables the connection between entities, fostering data discovery. For example, in Code 3.1, the resource *Tim Berners-Lee* is described to be born in both *England* and *London*. Both locations are identified by unique IRIs, and might be subject of further triples, as can be seen in the same example (lines 13 to 16).

- *Datatype properties*: Provide a value defined by a singular XML Schema Datatype (XSD)[2]. In the previous example on Code 3.1, the resource *Tim Berners-Lee* is formally named "Timothy John Berners-Lee" and was born on "1955-06-08". Whereas both values are expressed as literal strings, the latter is declared to be typed as a *xsd:date*, so, a machine able to work with semantic data would interpret it as the date corresponding to June 8$^{\text{th}}$, 1955.

The last example, in which the machine knows the value literal makes reference to a date, hints the potential of Linked Data for machine processing. Unveiling the nature of the encoded value, in this case a date datatype, algorithms can use partial components of the whole value in their analytical processes, adapting the format should the analysis require it (e.g., extract just the month of the date, calculate the difference between two date instances, present the date using a variety of abbreviation formulae, etc).

Moreover, ontologies might include the definition of asserted facts or axioms, which provide the ability to make logical inferences from any dataset's descriptions. Semantic reasoning is a wide research field on its own, where

[1] http://w3.org/TR/owl2-primer
[2] http://w3.org/TR/xmlschema11-2

rules and techniques from the Machine Learning (ML) and Artificial Intelligence (AI) areas are combined in order to extract new knowledge from the already available data. Reasoning on semantic datasets lets analysts detect errors in the published data (e.g., a date object with a "2016-02-31" value would be reasoned to be invalid due to the number of days of February), or new triples from the existing ones.

The last scenario can be exemplified as in Figure 3.2: given a set of triples stating that Jason is Aaron's son and Peter is Aaron's brother, a semantic reasoner can infer that Peter must be Aaron's uncle, and therefore, Jason Peter's nephew (many properties have inverses, this is, the opposite term for the *ont:uncle* relationship is one of the *ont:nephew*/*ont:niece* set). Moreover, it can be inferred that the relationship between Aaron and Peter is uncle/nephew and not aunt/niece because of the *foaf:gender* → *male* statements.

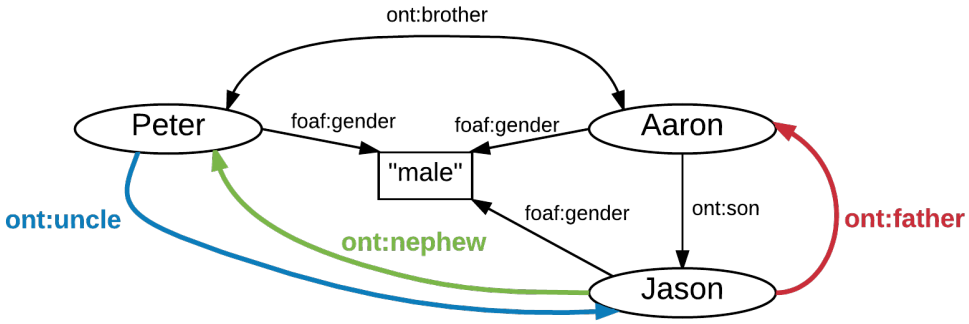


Figure 3.2: The original triple set (black) and the inferred properties by a semantic reasoner (coloured).

Due to the *schemaless* structure of the Semantic Web, our approach towards LOD visualization is designed to solely rely on the present data, as is the one factor that is always available to query in any dataset. This data-driven strategy is also adopted by most generic visualization tools from Section 2.2. To design a data-driven approach, through which data is discovered as it is being explored, we searched for a data analysis framework that could fit our proposal. Mathematician John Wilder Tukey explored these data-based approaches more than four decades ago, introducing EDA: Exploratory Data

Analysis (Tukey, 1977) as a set of basic statistical techniques supported by graphical representations to deal with new data sources. EDA does not demand data to fit a specific model to start the analysis stage. Instead, it takes the data *as-is*, creating a data-driven model on the fly, lacking the rigidity of more formal methodologies.

A well-known case used to justify and encourage the usage of visual representations, is what is known as **Anscombe's quartet**, a demonstration proposed in 1973 by the English statistician *Francis Anscombe* to exhibit the limitations of summary statistics. The example presents the four datasets displayed in Table 3.1, each of them formed by eleven sets of (x, y) pairs.

$dataset_1$		$dataset_2$		$dataset_3$		$dataset_4$	
x_1	y_1	x_2	y_2	x_3	y_3	x_4	y_4
10.00	8.04	10.00	9.14	10.00	7.46	8.00	6.58
8.00	6.95	8.00	8.14	8.00	6.77	8.00	5.76
13.00	7.58	13.00	8.74	13.00	12.74	8.00	7.71
9.00	8.81	9.00	8.77	9.00	7.11	8.00	8.84
11.00	8.33	11.00	9.26	11.00	7.81	8.00	8.47
14.00	9.96	14.00	8.10	14.00	8.84	8.00	7.04
6.00	7.24	6.00	6.13	6.00	6.08	8.00	5.25
4.00	4.26	4.00	3.10	4.00	5.39	19.00	12.50
12.00	10.84	12.00	9.13	12.00	8.15	8.00	5.56
7.00	4.82	7.00	7.26	7.00	6.42	8.00	7.91
5.00	5.68	5.00	4.74	5.00	5.73	8.00	6.89

Table 3.1: Anscombe's quartet: x and y values for each dataset.

Even though the table can be considered *small* from an analyst's point of view, the first idea that comes to mind for many seasoned examiners is to summarise the table. In order to do that, some typical statistical indicators are calculated with the following results for each of the four datasets:

	$dataset_1$	$dataset_2$	$dataset_3$	$dataset_4$
$\overline{x}$	9	9	9	9
$\overline{y}$	7.50	7.50	7.50	7.50
Var[x]	11	11	11	11
Var[y]	4.12	4.12	4.12	4.12
$\rho(x,y)$	0.816	0.816	0.816	0.816

Table 3.2: Some summary statistics of Anscombe's quartet: mean, variance and correlation of their x and y variables.

Should we stop at this point, we could summarise that the datasets are the same, misjudging the differences between the four. For all datasets, the *mean*, *variance* and correlation of its variables remains the same. Furthermore, the equation to fit a regression line for each dataset follows the formula:

$$y = 0.5x + 3$$

What both *Anscombe* and *Tukey* proposed, was to add a quick and simple visualization stage to most analytical processes, as it could uncover hidden insights of the data. If we happen to represent these four datasets using a basic *scatter plot*, the visualizations would be the ones depicted in Figure 3.3, providing a quick overview of the appearance of each dataset.

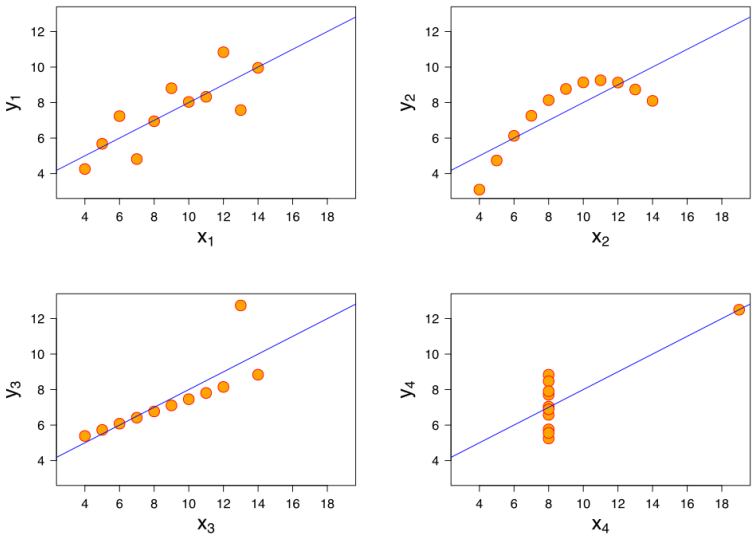


Figure 3.3: Scatter plot of each of the four datasets that are part of Anscombe's quartet. All four datasets are indistinguishable when statistically summarised, but its differences can be quickly observed once visualized.

Looking at the data itself, exploratory approaches follow more natural and suggestive data discovery paths, avoiding the biases produced by assumptions usually made from the analysts' perspective. The "*follow-your-nose principle*" (Yu, 2011) which is sometimes referenced within the Semantic Web community, encourages a similar discovery plan, acquiring new knowledge whilst navigating through related resources in linked datasets.

3.3 Primitive datatypes inference

In order to apply any preprocessing, analytical or visualization techniques, we need to figure out **which** is the nature of the data, an information that would allow to determine how each property can be visually mapped and represented. The first strategy could be to rely on each property's *rdfs:range* to detect the values' types. Nonetheless, this might not be the best approach. A careless, unsupervised publication of datasets under the Linked Data guidelines might cause major faults on correct datatype assignment, leaving most datatype-properties values typed as plain, literal strings (Zembowicz et al., 2010).

On July 2016, we conducted a brief experiment to study the usage of *rdfs:range* in a real environment. The target dataset was the English chapter of DBpedia, the nucleus for the Web of Data (Auer et al., 2007), describing in its latest version more than 4.58 million items[1]. Through querying its SPARQL endpoint[2], the number of properties that defined a valid *rdfs:range* value was retrieved and compared to the total number of distinct properties defined within DBpedia. Only **2,420** out of **63,152** properties in DBpedia (**3.83%**) provide information about their range. It might happen that despite the small presence of properties with a range defined, they are used in many statements within the DBpedia dataset. Retrieving the number of triples whose predicate's property declared a range value, we counted **68,652,893** axioms out of the total number of triples in DBpedia: **438,038,746** (**15.67%**). The same queries retrieved back in November 2015 produced a similar result: **4.23%** and **15.25%** respectively (Peña et al., 2016).

[1] http://wiki.dbpedia.org/about/facts-figures
[2] http://dbpedia.org/sparql

Given this scenario, exclusively trust on defined property ranges is not feasible to produce satisfactory results. For our research purposes, we propose a mixed solution using both the defined *rdfs:range* values (if any), and the results from a custom datatype inference algorithm. The inference step will try to classify each value to the datatype categories where they best fit, profiling the way in which each feature should be interpreted, how is implemented, encoded and stored within the data repository, together with the set of available operations, its meaning and the value ranges allowed for each observation (Parnas et al., 1976).

The selection of datatypes to classify data values has been a challenge for different researchers working on automatic information categorisation. Before implementing the algorithms that will assign the most suitable datatype, these categories must be specified and described in detail. The most extended datatypes are the ones described by Shneiderman back in 1996 (Shneiderman, 1996), which are used in *LDVM*'s pipeline (Brunetti et al., 2013) to categorise data values. These datatypes are: 1-dimensional, 2-dimensional (planar), 3-dimensional (volumetric), multi-dimensional, tree, temporal and network. Due to the relevance of this classification scheme, the visualization tools reviewed for the State of the Art have been scored according to these datatypes (consult Table 2.1). However, in our opinion this classification might produce some doubts in later inference stages, as the features of each category can not be easily implemented. For example, what is the structure of a *network* value? Are all triples in a LOD dataset *network*-like due to their encoding as RDF triples? And how are overlaps between categories solved? A similar strategy was proposed by Melanie Tory and Torsten Möller (Tory and Moller, 2004), adding sub-types for numerical data, such as: discrete, continuous, nominal and ordinal. Whilst these dimensions are extremely useful for more complex analysis, it suffers from the same disadvantages exposed before.

LDVizWiz (Atemezing and Troncy, 2014) uses a different approach, identifying "content spaces" that can be directly mapped to different sets of vocabularies. These spaces are: Geography, Geometry, Time, Event and Government. This approach is very simple to implement, querying the dataset using the *ASK* clause of the SPARQL language to detect if the property belongs

to the list of valid ontological properties selected for each space. The limitations of this approach are the need of re-implementation in order to include new properties, and that given so few categories, some values will be difficult to detect: Where does a *Person* instance belong to? Are all organizations classified under the *Government* space?

Finally, Donato Pirozzi and Vittorio Scarano (Pirozzi and Scarano, 2016) have recently taken a more practical approach to programmatically infer the datatypes by syntactically parsing the values of the dataset. Their classification is depicted in Figure 3.4. The inference is performed using *JSDataChecker*[1], an open-sourced library implementing the regular expressions used for detecting the selected categories.

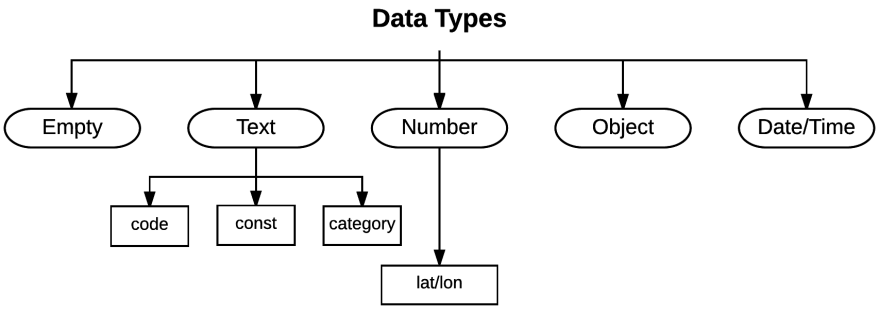


Figure 3.4: Data types and sub-types identified by D. Pirozzi and V. Scarano. Types are enclosed within rounded boxes, and its related sub-types are depicted inside rectangles. Sub-types are intended for a further inference on the data.

Even though some of the previous categories are similar (many works recommend a temporal dimension, or a plain text one), we have found inconsistencies when categorising values from real datasets. Properties should be evaluated in a quick manner to speed up the exploratory analysis stage, and should be sufficiently generic in order to avoid biases and limitations due to datatypes restriction.

For our research, we propose the following **primitive datatypes** within queried properties can be classified into. The reader should take into consid-

[1] https://github.com/donpir/JSDataChecker

eration that these categories are intended to be conceptual and programming-language agnostic.

- **Integer**: Composed by the finite computer representable subset of whole numbers, such as the height of a person in centimetres, or the number of wheels in a given vehicle. Negative values are permitted.

- **Float**: The representation of any real number, as the height of a person in metres. Negative values are permitted.

- **Boolean**: A value meaning a logical truth, such as "*true/false*", "*0/1*", "*yes/no*" pairs of values.

- **IRI**: Internationalised Resource Identifiers are a standard defined upon the URI scheme, formed by any Unicode character sequence which uniquely identifies any resource over the Internet. IRIs are especially relevant in the SW, as they constitute one of its core components, making resources to be linkable and discoverable between them. IRI values are the only representatives of *object-datatypes*.

- **String**: Defined as any sequence of characters, they are understood as a superset covering the rest. In fact, many values in LD are typed as plain strings (*xsd:string*, *rdfs:Literal* or any of its variants), without any more concrete "^^*xsd:datatype*" defined for the object's values or the property's *rdfs:range*.

- **Datetime component**: A part of either a date, a time or both, expressed in any standardised format (preferably following ISO 8601's directives).

- **Geographical component**: Any geographical dimension which could help locating a resource in space, e.g., a pair of *latitude-longitude* coordinates and its projection system, a geographical feature or point and the like.

- **Categorical data**: Marking a property as *categorical* means that the range of used values is limited or within a certain range, which enables new visualizations to represent the property, e.g., histograms which display the instance count per occurrence, value distribution or usage.

Our inference algorithm will try to map each ontological property to the primitive datatypes which best represent the values declared for each property. The automatic publication of datasets under the LOD premises may produce errors which are difficult to detect by data publishers. Within large datasets it is fair to assume that a certain percentage of the values will be incorrectly declared. Even though a small percentage of errors in a large dataset might seem insignificant from an experienced analyst's perspective, it may present challenges for our inference algorithms, which take a data driven approach and thus, are affected by its quality. The assignment of the most suitable primitive datatypes will take into account the different scenarios the data values analysis might cope with.

- Missing values do not imply a great issue for our approach, as triples are not generated when an object value is not present (consult OWA in section 3.1 for further details). The only exception would be when empty values do generate triples, for example, empty strings (''). In our study, we will not try to estimate the values of missing information, as (also taking the assumptions made in OWA environments) we can not state that there are any missing values at all.

- As declared in this section's introduction, the *rdfs:range* values for each property are retrieved should they happen to be available within the dataset or encoded in the corresponding ontology's description file. In those favourable cases in which ranges are stated for the property, they can lack specificity (e.g., the property is described to belong to the real numbers domain when most values can be enclosed in the integers subset, or the most common case of properties being typed as plain, literal strings), or not be applicable by some values (e.g., a generation error that includes a text string value when many other values are typed using digits).

To deal with these contexts, a threshold-limited procedure has been established to infer the primitive datatype categories. Besides, the retrieval of all the values of each property in order to infer the datatypes should be avoided whenever possible, as querying large amounts of data in SPARQL usually results in low performance, and may ultimately lead to time-out errors in the response. Also, as this operation is performed for each property, and may require periodical updates, it is recommended to keep it reasonably efficient whereas accuracy does not drop. For our research, we have envisaged the querying strategy depicted in Figure 3.5.

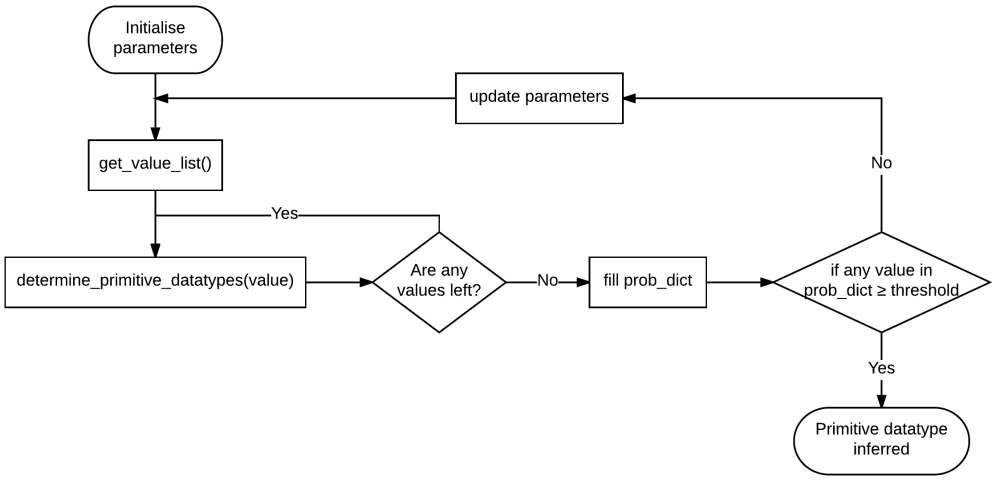


Figure 3.5: Primitive datatype inference iterative algorithm.

The main idea is to retrieve the minimum number of values to detect the essence of the property, looking for a homogeneous datatype that represents the whole set of values. Thus, the first task is to initialise the parameters that will be used by the inference process. These variables are:

- **threshold**: This constant determines the minimum agreement percentage that needs to be reached in order to assign data categories to the analysed property with a proper trust.
- **sample_size**: The number of sample values to query from the SPARQL endpoint, similar to the *LIMIT* clause in *SQL* statements.

- **offset**: The starting value from which to start querying values. The purpose of this parameter is to avoid always retrieving the first results. From our experience querying SPARQL endpoints, the first results may have some generation errors which would interfere with our classification. To avoid exceptions from exceeding the number of results, the offset is set to a *random* integer from 0 to the difference between the total number of values for the property and the defined *sample_size*. As the offset is randomised for each iteration, we ensure the retrieval of different subsets in each inference process.

- **multiplier**: Designates the factor by which the *sample_size* would be increased in the next iteration. If no primitive datatype category exceeds the established threshold, more sample values will be retrieved to offer a new detection opportunity.

Each time a sample value list is retrieved, an individual inference algorithm tries to map each of its values to the primitive datatypes explained in this section: Integer, Float, Boolean, IRI, String, Datetime component and Geographical component. Once all the values have been scrutinised, each value contributes in the estimation of the property's probability of belonging to each datatype category, by adding its individual weight to the equation. If there is a consensus of category belonging above the established threshold, the datatypes are assigned and remembered for further analyses. Otherwise, a bigger value subset is queried and runs through the same procedure.

3.3.1 Evaluation of datatype inference

In order to validate the primitive datatype inference stage presented in this section, which tries to classify each property based on the features of their related values, we evaluated the performance of the inference algorithm with real datasets. We selected 5 different datasets available through their respective SPARQL endpoints, each of them covering a different topic. The datasets are briefly summarised next.

SPARQL endpoint 1: Air quality

`http://zaragoza.es/ciudad/risp/detalle_Risp?id=131`

Provides information about the air quality of the Spanish city of Zaragoza, together with the daily measurements for most known pollutant particle levels.

SPARQL endpoint 2: Restaurants

`http://zaragoza.es/ciudad/risp/detalle_Risp?id=285`

Contains information about the different eating options in the Spanish city of Zaragoza.

SPARQL endpoint 3: Historical sites

`http://zaragoza.es/ciudad/risp/detalle_Risp?id=86`

A dataset featuring the interesting monuments and historical sites of the Spanish city of Zaragoza, with additional information that might be interesting for tourists.

SPARQL endpoint 4: MORElab

`http://apps.morelab.deusto.es/labman/sparql`

This set of data contains information about our research unit, providing information about innovation projects, publications, researchers profiles, along with others.

SPARQL endpoint 5: Teseo

- Currently not available -

It contained all the metadata about the PhD dissertations that have been publicly defended in Spain since 1976. Due to a formal request from the Spanish Ministry's of Education, the dataset was retired from public access short after carrying out these experiments.

These datasets contained more than 15 million statements about nearly 3 million unique resources, described using 194 different ontological properties, 149 of which were unique. An overview of them can be seen on Table 3.3.

Dataset	Triples	Resources	Classes	Properties
Air quality	$>11M$	$>2.2M$	6	28
Restaurants	$\approx 32.5K$	$\approx 4.1K$	4	36
Historical sites	$\approx 60K$	$>6.8K$	6	30
MORElab	$\approx 27K$	$>3K$	30	72
Teseo	$\approx 4M$	$>650K$	6	28

Table 3.3: Figures for the selected datasets (number of triples, resources, classes and properties), as of August 2016.

We asked 6 experts in computer science with knowledge in semantics and LOD to classify and tag each property, using the datatype categories defined in Section 3.3. Every time they were presented a new property, a set of 20 related values were randomly retrieved from the dataset, so they could figure out what data dimensions each property was related to under their judgement. To assign a datatype to an evaluated property, we established a minimum threshold of agreement of 80%. Having 6 people involved in the classification task, at least 5 of them should assign the same category to have the property classified with it. Once this tagging was finished, it served as our ground truth, the reference our classification algorithms would be validated against.

Our first approach to categorise properties (*determine_primitive_datatype(value)* stage from Figure 3.5) is based on a simple pattern matching algorithm. As all values are retrieved in string format from any SPARQL query, we implemented a series of basic rules that use regular expressions and type introspection techniques to categorise each value. The algorithm starts by trying the most restrictive datatypes, and if a pattern match is not detected, tries softer ones. For example, given a new singular value, it will try to detect integers (only digits, no decimal numbers allowed). If the value does not fit the regular expression, the algorithm will try to detect the layout of a decimal (float) number.

The results of this approach are exhibited in Table 3.4, which renders the number of detections made by the classification algorithm compared to the answers provided by the experts. For each property, we checked the algorithm's output against the tagged dataset, comparing one by one each of the primitive datatypes. The classification is said to be *Correct* when both the algorithm's output and the experts agreed on the property belonging or not to a particular category. In this case, the *Correct* column can be referred to as *accuracy*. Both *Type I* (False Positives, FP) and *Type II* (False Negatives, FN) errors count towards the *Incorrect* ratio.

Dataset	**TP**	**TN**	**FP**	**FN**	**Categories**	**Correct**	**Incorrect**
Air quality	17	160	2	10	5	93.65%	6.35%
Restaurants	17	201	3	17	5	91.60%	8.40%
Historical sites	14	165	4	13	3	91.33%	8.67%
MORElab	56	399	15	13	12	94.20%	5.80%
Teseo	22	162	4	1	3	97.35%	2.65%

Table 3.4: Sensitivity and specificity of the primitive datatype inference task, when compared to the datasets tagged by experts. The table presents the TP (True Positives), TN (True Negatives), FP (False Positives) and FN (False Negatives) numbers collected after the evaluation, together with the number of identified categorical properties as reference.

In order to test the sensitivity and specificity of our inference stage proposal, we have tried different values for the *threshold* and *sample_size* variables that intervene in the primitive datatype inference algorithm depicted in Figure 3.5. To reach a trade-off between the algorithm's efficiency and resource consumption (number of queries performed against the SPARQL endpoint, time to reach an agreement above the threshold and so on), we executed the datatype inference feature with different values, as exhibited in Table 3.5. The table shows the *[threshold - sample_size]* value combinations, together with the time required by the algorithm to obtain the results (taking an average value from 3 consecutive runs), and the accuracy of the datatype inference task for the *Zaragoza's historical sites* dataset.

threshold	sample_size	t_1	t_2	t_3	$\bar{t}$	accuracy
0.65	10	8.84482	8.66309	8.93391	8.813	91.33%
0.65	50	11.45653	11.48960	11.52691	11.491	91.33%
0.65	100	13.31747	13.42749	13.38761	13.377	91.33%
0.75	10	8.89752	8.88305	8.84635	8.875	91.33%
0.75	50	11.56456	11.5569	11.46349	11.528	91.33%
0.75	100	13.34920	13.26753	13.36739	13.328	91.33%
0.85	10	8.78624	8.85464	10.10007	9.246	90.84%
0.85	50	12.26136	12.17050	12.14460	12.192	90.84%
0.85	100	13.64277	13.90993	13.76596	13.772	90.84%

Table 3.5: Experiment to test the performance of the primitive datatypes inference algorithm with different default values.

As it can be observed, the more instances are retrieved (*sample_size*), the longer it takes to finish the execution. To retrieve 10 samples the average time was 9 seconds for each *threshold* value, whereas to retrieve 100 samples the time required was around 13.5 seconds. As accuracy did not experiment a great decrease, with only a minimal reduction when the agreement threshold was raised to 85% compared to the experts' answers and due to *Type I* errors, we recommend to use the following value range for each of the variables:

- **threshold**: ≥ 0.8
- **sample_size**: $25 \leq sample\_size \leq 50$

Values for the analysis in Table 3.4 were obtained with a *threshold* value of 0.8, and a *sample_size* of 30.

3.3.2 Conclusions

To understand the datatypes and categories each property could be classified into allows to know what dimensions values are representing, and which operations and transformations they allow. For this research, 8 categories have been selected. Even though they can be further extended and refined, we consider that they cover the broad set of data features present in most datasets. The better the classification task is performed, the more reliable

the results from analyses would become. The data-driven approach removes the biases that might derive from some of the property's features: its label, incorrect *rdfs:range* definition and so on. However, performance should not be sacrificed at the expense of accuracy: with increasing sizes of datasets, and the popularity of streaming data, datatype inference should have as little computational impact as possible on the whole visualization pipeline.

3.4 Relevance assessment

"Metadata provides the means to discover datasets, access them and understand them". Through this statement (Carvalho et al., 2014), we can sense that relevant metadata metrics might help in our task at hand: the generation of insightful summary visualizations of LOD.

Different initiatives such as RDFStats (Langegger and Woss, 2009), LODStats (Demter et al., 2012), VoID (Alexander et al., 2009; Mäkelä, 2014), LOD Laundromat (Beek et al., 2014) and so on propose different metadata metrics to describe datasets. Although each project has its own set of properties and focuses on particular statistics, they agree on some common metrics, such as:

- Total number of classes and properties, and instances of each category.
- Number of axioms (triples) in the dataset.
- Count of unique instances, this is, different described resources.
- Licensing (if any), listing the permissions of the data and what limitations it might have.
- In/Out-degrees (number of incoming and outgoing links to external datasets).

These indicators are usually listed in tabular format, leaving their interpretation to analysts. For example, LODVisualization (Section 2.2.2.3) generates visual representations of these metrics, aggregating class and property hierarchies to later plot them as treemaps, allowing users to explore the components of each parent entity in a naive fashion.

With the intention to complement previous works, we propose the extraction of further metrics that help picturing the whole dataset's composition. The goal of these statistics is to rank entities according to different parameters, exhibiting the most appropriate ones depending on the analysis context without information loss.

3.4.1 Property usage

Given the *schemaless* character of LOD entities, each resource can be described using whatever property defined in any ontology. Vocabularies provide definitions for those features they support, modelling how they should be used and the manner they relate to other attributes. Nevertheless, there are no limitations to bring properties from other ontologies to further characterise the resources on any dataset.

Every individual instance is free to provide values for any property, even if they are the only instance to do so. This implies that when querying an ontological class within a SPARQL graph, the number of available properties might become rather lengthy. When instances are characterised in great detail, the analyst seeking a data overview could find herself lost in the amount of provided information. To deal with this scenario, we propose to analyse the usage of each property in any graph.

To avoid performance issues, the operations are intended to be as atomic as possible, and executable against both SPARQL 1.0 and 1.1 versions. The first step is to ask for the number of unique instances (individuals) which are typed with the class under study (the SPARQL query is listed in Code 3.2).

```
1 SELECT
2     COUNT(DISTINCT ?subject) AS ?class_instances
3 WHERE {
4     ?subject a <CLASS> .
5 }
```

Code 3.2: SPARQL query to retrieve the number of unique instances of a target class.

Then, we list all the properties used to describe the resources belonging to the target class, and for each of them we query the number of unique instances that generate valid value triples (Code 3.3).

```
SELECT
    ?property,
    COUNT(DISTINCT ?subject) AS ?instances_with_property
WHERE {
    ?subject a <CLASS> .
    ?subject ?property ?object .
}
GROUP BY ?property
```

Code 3.3: SPARQL query to retrieve, for each property describing a target class, the number of unique instances using it.

Thereafter, each aggregated value is divided between the total number of unique class instances. We refer to the resulting value as the **property usage**. To illustrate this scenario, we provide the following example:

Let's consider a dataset with 300 instances of the *foaf:Person* ontological class. All of them have at least one *dc:label* value assigned, 225 of them provide information about for their *foaf:gender* and 60 fill in the *foaf:depiction* field. The computed *property usage* values will be:

- *dc:label* → pu = 1 ($\frac{300}{300}$)
- *foaf:gender* → pu = 0.75 ($\frac{225}{300}$)
- *foaf:depiction* → pu = 0.2 ($\frac{60}{300}$)

The property usage ratio allows to rank properties defined within a class using the relative usage as the ordering factor. In a naive manner, should we wanted to retrieve the n most relevant properties according to this metric, we could query the dataset using Code 3.4.

Property usage values are spanned in the $(0,1]$ range. The closer a value is to 0, the less the property has been used to describe the entity. Those properties which have never been used ($prop\_usage = 0$) are not listed, avoiding visual cluttering that big datasets might produce (e.g., DBpedia). The closer the *prop_usage* is to 1, the more representative it becomes to model the class.

```
SELECT
    ?prop,
    xsd:float(COUNT(DISTINCT ?subj2))/xsd:float(COUNT(DISTINCT ?subj1)) AS ?prop_usage
WHERE {
    ?subj1 a <CLASS> .
    ?subj2 a <CLASS> .
    ?subj2 ?prop ?obj .
}
GROUP BY ?prop
ORDER BY DESC(?prop_usage)
LIMIT n
```

Code 3.4: SPARQL query to retrieve the properties related to a particular class within a graph, ordered by their *property usage* value.

The reason behind only retrieving unique entity instances is to prevent the misinterpretation of metrics produced by retrieving all the instances that use a specific property. If the property promotes the generation of various triples for each unique subject, not filtering by unique instances might publicise the property as highly relevant even if it is not the case. For example, lets take a dataset modelling countries. An expected property would have something to do with the "*shares borders with*" feature, which would relate two countries. As countries are usually adjacent to more than one country, a value count for the property which does not take into account instance uniqueness might offer a false impression of its relative significance.

In order to evaluate the validity of this metric to rank properties based on their usage within a class, we have compared our algorithm's performance to one proposal from EURECOM[1] and INRIA[2] (Assaf et al., 2014). In this work, they took 9 classes from DBpedia (detailed in Table 3.6), and extracted their most relevant properties using two different approaches: the information displayed for each entity in Google's Knowledge Panels (GKPs, a similar view to that of Wikipedia's and its infoboxes, where the most relevant information is listed), and a user survey in which 152 different participants where asked about how they would represent each entity.

[1] http://eurecom.fr
[2] http://inria.fr

Name	IRI	# instances	# properties
Book	http://dbpedia.org/ontology/Book	34615	927
City	http://dbpedia.org/ontology/City	20943	2378
Company	http://dbpedia.org/ontology/Company	83220	3414
Country	http://dbpedia.org/ontology/Country	3294	1451
Film	http://dbpedia.org/ontology/Film	101906	1811
Museum	http://dbpedia.org/ontology/Museum	4946	646
Politician	http://dbpedia.org/ontology/Politician	39722	1585
SoccerClub	http://dbpedia.org/ontology/SoccerClub	20877	1445
TennisPlayer	http://dbpedia.org/ontology/TennisPlayer	4725	401

Table 3.6: Number of instances and property figures for the DBpedia classes selected by Assaf et al. in their study.

For each class, we present in Table 3.7 the *property usage* (pu) value extracted for every property (both the ones extracted from GKPs, and those the users considered most relevant in the survey), and added the highest ranked properties attending to their *property usage* value. For the sake of clarity, the latter have been filtered, omitting those that are related specifically to DBpedia (such as links to Wikipedia or IDs) and common properties within LOD resources (labels, comments, types and so on). The names of individual resources are not displayed, being supposed to be present in every instance.

After analysing the results from Table 3.7, it can be seen that for each class queried by *Ahmad Assaf*, around half of the properties are part of the top-5 according to the *property usage* ranking. However, the other properties listed at GKPs received a very low value. An explanation for this fact might be the merge of different sources to generate each class' template in Google's search results pages, which do not extract data solely from *Wikipedia*.

We can also conclude that only a small number of properties are used in more than 80% of any dataset's instances (around 4-5 properties, approximately 10 if we take into account labels, resource names, comments and other common properties). These *long-tail* distributions of *property usage* percentages focus the attention on that only a small number of features are widely used to describe resources in LOD environments, and highlight their importance for tasks like entity summarisation, thus promoting the *property usage* metric as a valid one to quickly evaluate a feature's usefulness within a dataset.

Class	GKP	*pu*	User survey	*pu*	Higher *pu*	*pu*
Book	Published	58.59%	Authors	96.39%	Authors	96.39%
	Authors	96.39%	Literary genres	63.78%	Language	88.37%
	Genres	73.99%	Original language	0%	Publisher	82.70%
	Followed by	31.84%	Publication date	58.59%	Country	82.51%
	Preceded by	28.94%	Language	88.37%	Genres	73.99%
City	Local time	88.95%	Population	94.21%	Population	94.21%
	Area	63.19%	Points of interest	0%	Geolocation	93.11%
	Population	94.21%	Weather	0%	Timezone	88.95%
	Weather	0%	Local time	88.95%	Country	87.98%
	University	0%	Geolocation	93.11%	Population density	80.17%
Company	Founded	57.36%	List of products	34.50%	Homepage	66.72%
	Founders	20.83%	CEO	39.70%	Founding year	57.36%
	CEO	39.70%	Logo	44.13%	Industry	55.10%
	Headquarters	35.50%	Location country	23.77%	Logo	44.13%
	Stock price	0%	Number of employees	22.48%	Key people	39.70%
Country	Founded	81.88%	Official languages	59.65%	Flag	91.23%
	Government	81.63%	Population	12.60%	Long name	90.77%
	Capital	86.79%	Capital	86.79%	Capital	86.79%
	Continent	80.24%	Currencies	40.01%	Dissolution year	82.88%
	Currencies	40.01%	Type of government	81.63%	Founding year	81.88%
Film	Director	96.48%	Release date	39.01%	Director	96.48%
	Running time	74.11%	Film duration	74.11%	Language	95.99%
	Initial release	39.01%	Director	96.48%	Country	91.86%
	Screenplay	63.45%	Awards	0.17%	Starring	85.49%
	Release date	39.01%	Characters	85.49%	Producer	74.42%
Museum	Address	95.73%	Opening hours	0.06%	Location	95.73%
	Hours	0.12%	Website	86.21%	Homepage	86.21%
	Phone	0.12%	Address	95.73%	Geolocation	85.42%
	Opened	77.54%	Master pieces	6.71%	Depiction	78.02%
	Founders	2.26%	Closing hours	0%	Established	77.54%
	Artwork	6.71%	Geolocation	85.42%	Type	64.80%
Politician	Born	82.55%	Political party	66.34%	Description	89.71%
	Died	49.32%	Nationality	21.36%	Birth date	82.55%
	Education	0.52%	Office	44.09%	Term start	67.22%
	Party	66.34%	Birth place	63.21%	Party	66.34%
SoccerClub	Founded	67.85%	League	78.89%	Club name	91.98%
	Arena/Stadium	83.97%	Location	1.66%	Full name	89.93%
	Manager	56.27%	Titles	3.27%	Ground	83.97%
	Location	1.66%	Official club name	89.93%	League	78.89%
	Training ground	0%	Website	48.13%	Socks	75.50%
	League	78.89%	Stadium	83.97%	Shorts	75.44%
TennisPlayer	Born	96.72%	Titles won	73.52%	Birth date	96.72%
	Height	52.11%	Nationality	81.86%	Birth place	89.02%
	Weight	15.51%	Birth date	96.72%	Country	81.86%
	Turned pro	53.71%	Preferred hand	75.05%	Highest singles ranking	81.69%
	Siblings	0%	Career price money	70.14%	Preferred hand	75.05%

Table 3.7: Property usage values for the most relevant properties listed in Assaf et al.'s work, both by means of Google's Knowledge Panels and a user survey asking for user preferences.

3.4.2 Completeness ratio

The *property usage* explained before exhibits the publisher's preferred properties to describe the resources belonging to a certain ontological class, but does not reflect the intended purpose when assigning values to each property.

In the last example, we could imagine that a singular *Country* instance might share borders with more than one different instances, but an individual *Person* generating different triples for its *blood type* looks odd.

For this reason we have envisaged an extra metric: the **completeness ratio**, which addresses how value counts for each property are distributed. The computation takes the total number of values which conform a RDF triple with the property under study and the subject being an instance of a particular class (Code 3.5), and divides it by the number of unique instances that provide a value for the property (query from Code 3.3).

```
SELECT
    COUNT(?object)
WHERE {
    ?subject a <CLASS> .
    ?subject <PROPERTY> ?object .
}
```

Code 3.5: SPARQL query to retrieve the number of values (objects) which are part of a triple with a particular class and property.

The computed value for the *completeness ratio* (cr) can reflect either of these scenarios:

- $cr = 1$: For those individuals using the property, each unique instance does only provide a singular value. Taking some of the examples suggested before, only 50 instances from our 300 *foaf:Person*'s dataset provide information about their *blood type* ($prop\_usage = 0.1\overline{6}$), but each of them assigns one (and only one) value to define it ($cr = 1$).
- $cr > 1$: The instances which use the property to depict the entity, use more than one values to do so. As an example, if our people dataset profiles an attribute featuring the knowledge areas each person works at, we could expect more than one value for every instance of the dataset.

When $cr > 1$ and the number of values is large, the *completeness ratio* concept can better be understood using a visual representation of it. In this particular case, a **histogram** can greatly improve the explanation of the metric. In order to generate the chart, the first step is to retrieve all the triples with the subject and the property fields belonging to the target ontological class and property, respectively. Then, we count the number of triples retrieved for each unique subject. Finally, the count items are aggregated, grouping together those with the same amount of triples. The resulting histogram shows how many values do the instances of the dataset describe.

3.4.3 Conclusions

Research has dealt with resource importance within LOD in the past, focusing on three elements and assessing their relevance: datasets, classes and properties. Both for datasets and classes, the studies versed on topic inference (Peroni et al., 2008; Voigt et al., 2013b; Atemezing and Troncy, 2014; Meusel et al., 2015), trying to profile data according to their contents.

Concerning properties, the most relevant work until now is the one presented in Section 3.4.1 (Assaf et al., 2014). The metadata metrics proposed in this section are intended to improve the manner LOD entities are summarised in a simple and fast manner, focusing the users' attention around those features that contain the essence and most relevant information about the entities described through the datasets.

3.5 Summary

In this chapter we have focused on the main characteristics of the LOD publishing model, which allows to say *everything* about *any* resource using controlled vocabularies. The wide range of possibilities, only limited by the restrictions imposed at ontology definition level, makes difficult to understand **what** the contents of a dataset are, specially when the dataset is large or unknown for the analyst.

To be able to understand the data in such scenarios, we propose a data-driven approach, one that takes the data itself and analyses it in order to profile the contents of any dataset. Some concepts from the *Exploratory Data Analysis* (EDA) field are brought into this approach, promoting data visualization as a perfect candidate to quickly explore the dimensions of a dataset.

Our first goal was to infer the datatypes of each dataset's properties. Datatypes describe what can be done with the data, how the property is defined by its values and opens a path for incorrect usage detection. We have selected a set of datatype categories any value could be classified as, and designed an inference algorithm to detect the datatypes of each property by retrieving a random sample of values from its dataset's SPARQL endpoint. The inference stage is evaluated against the agreement of 6 experts, which have manually tagged almost 200 different properties. An accuracy over 90% in the experiments that were carried out indicates the suitability of this approach for our classification purposes.

Once we know how to make the most out of each property, ranking them is recommended to avoid information overload and guide the users' attention. As anyone can publish anything about resources on the web, a need for filtering and focusing on the most relevant features arises. Semantic datasets might end with hundreds of properties used to describe each entity, with only a small percentage of them containing the majority of valuable data. Therefore, we propose the extraction of different metadata metrics (which allow to calculate the *property usage* and *completeness ratio* values) which are intended to help in ranking tasks, making the most relevant features to stand out whilst adjusting the results to the dataset's reality.

The proposed approach makes feasible to get a quick overview of any dataset's contents, only by retrieving a small subset of random values from it and with little computational impact in the whole visualization pipeline. Each task is independent from the rest, so even a small improvement in a particular job should provide an overall better exploratory experience.

The drawing shows me at one glance what might be spread over ten pages in a book.

"Fathers and Sons", Ivan S. Turgenev (1862)

CHAPTER

4

Visualising LOD

METADATA ANALYSIS has allowed us to profile LOD datasets in order to understand how data is structured and what operations we can perform with it. Knowing which features and values does our target datasets contain, we could now go a step further and represent them using visual means.

However, there is an extensive variety of different visualizations we could map our datasets to. To produce the most coherent illustrations with the data at hand, we have centred our attention in different variables that would mould the final output, trying to encode each value in a manner that keeps most meaning.

These depictions let analysts identify diverse patterns in the data, getting a quick and overall picture of the data they are working with. Whilst exploring data, as stated by EDA's techniques, the idea is to generate *draft* graphics that will be quick and easy to produce and reject, but which will not be refined. These *dirty* charts, as they are sometimes referred to in the literature, would try to represent the dataset in the best possible way, encoding and mapping each feature in the different visualization's dimensions, but with little care about aesthetics. In order to produce each visualization, a series of heuristics will be used taking into account the data's characteristics.

Through this chapter, we will analyse how appropriate each visual representation is for each scenario, how to take into consideration the analysis' objective, and how to encode the heuristics that ultimately lead to the visualization's generation.

4.1 Chart suitability

According to a survey authored by the creators of D3js (Heer et al., 2010): "*Creating a visualization requires a number of nuanced judgements. One must determine which questions to ask, identify the appropriate data, and select effective visual encodings to map data values to graphical features such as position, size, shape, and color*".

Through the combination of these *visual encodings*, which make a direct reference to the *visual variables* originally described by Jacques Bertin (consult Figure 2.6 for further information), the set of visual representations that can be produced is extremely large and diverse. Thus, we need to map each of the data's dimensions to the visual variable that could best ease its comprehension whilst keeping its meaning, and search for the graphic or chart that best utilises those features.

This area of study is named **graphical perception**, and was thoroughly explored by William Cleveland and Robert McGill (Cleveland and McGill, 1984). They conducted a series of experiments that aimed to test how each visual variable affected human perception. The validation of the experiments sets the theoretical background of the field. As addressed by the authors in the article's introduction, after 200 years of visualization developments by its pioneers and some best-practices compilations by renowned visualisers, the scientific foundations of the field were still lacking, and more demonstrable approaches were needed in order to assess that a particular variable was preferred to another one for a specific task.

Both Cleveland and McGill came up with a ranking of the encodings that clearly had an impact on the accuracy of the judgements in their evaluation. The ordered list ended as follows (from most to least relevant variable):

1. Position (along a common scale)
2. Position (along non-aligned scales)
3. Length, Direction and Angle
4. Area
5. Volume and Curvature
6. Shading / Color saturation

This ranking has been frequently reviewed since its publication, and other studies have contributed by rising concerns about how other aspects of the visualization could affect the visual output. Jock Mackinlay expressed that Cleveland's ranking was suitable for quantitative information, but did not address the scenarios in which the nature of data required additional perceptual tasks and rankings (Mackinlay, 1986). His proposed ranking is depicted in Figure 4.1.

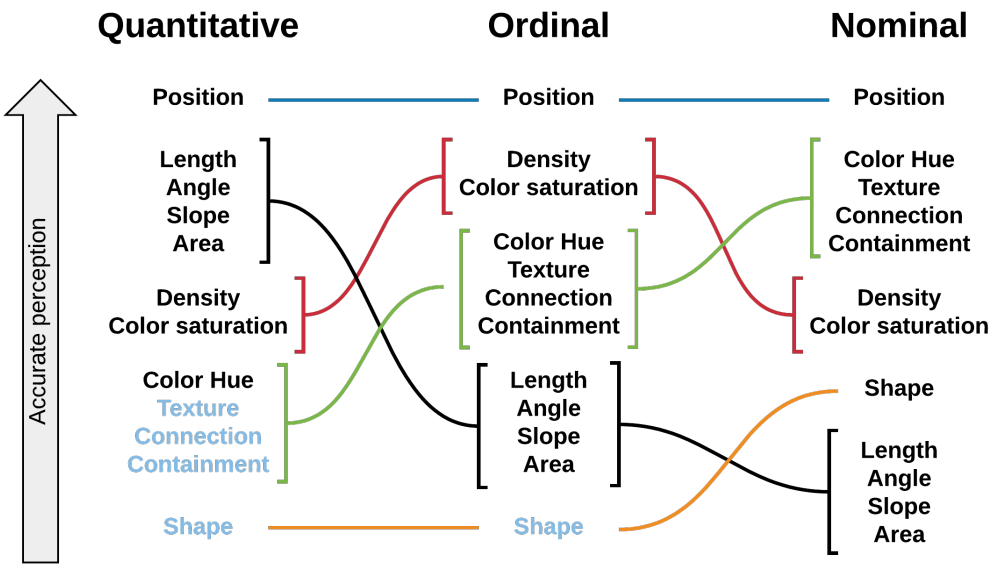


Figure 4.1: Jock Mackinlay's ranking of perceptual tasks, grouped and ordered by type of data (the tasks in light-blue are not relevant to that type of data).

Despite the discussions that try to adapt the ranking to particular scenarios, its ordering is broadly accepted as a guide when selecting visual variables. Therefore, humans are more reactive to changes in position when visualising a chart, whereas colour hue's differences are less noticeable. Area is preceded by length, which is a recurrently hoisted argument when defending the usage of bar/column-charts against pie/donut-charts: as humans are more sensitive to length variations on a singular scale (either width or height), whenever possible charts that encode differences in this dimension should be favoured, in opposition to those using bi-dimensional encodings (areas).

To be used in the rest of the chapter, we have selected some of the most known and used visualizations, in order to evaluate their compliance level to the items featured in each section. Even though the selection might seem limited and with only simple visualizations, they have traditionally been the most used ones to communicate results, being supported by every visualization tool that a user could be familiar with and easily comprehensible. The resulting selection of visualization types is portrayed in Figure 4.2.

Figure 4.2: Selected visualization types for the chart suitability analysis. The graphics are *a)* line charts, *b)* bar/column charts, *c)* radar charts, *d)* heatmaps, *e)* box-plots, *f)* scatter-plots, *g)* pie charts, *h)* tree maps, *i)* map projections and *j)* node-link diagrams.

(a) **Line charts**: Bi-dimensional chart in which data points are connected through line segments (either straight or curved). The datum are usually ordered by the x-axis values, which makes these charts really useful to work with data that changes through time.

(b) **Bar/Column charts**: These graphs depict information using rectangles, encoding the values through one of its sides: either the height or the width (depending if its a vertical column or a horizontal bar chart). The other side stays the same for all data instances.

(c) **Radar charts**: Also known as *star*, *spider* or *polar* chart, it allows to display multivariate data that is semantically connected. Modern visualizations also turn to radar charts to visualise periodical data.

(d) **Heatmaps**: A heat map is usually presented as a matrix that encodes each cell's value using colour differences. That way, users can quickly identify the *hot* or *cold* spots in the data.

(e) **Box plots**: Sometimes referred to as *box-and-whiskers diagrams*, they allow to represent grouped numerical data by their quartiles. Common statistical indicators such as *mean*, *minimum* and *maximum* values are easily included in these charts, and outliers are easy to spot as they appear as unconnected dots.

(f) **Scatter plots**: These graphs use Cartesian Coordinates to depict the values from two variables within a dataset. It is one of the most used chart types as it allows to add new features easily: opacity levels, category colours, size of the dots (being known as *bubble charts*) and so on.

(g) **Pie charts**: One of the best known graphs in the list, is used to depict values that form part of a whole. The corresponding values to each category or group are assigned a proportional section of the circle, thus resembling a sliced pie.

(h) **Tree maps**: They are an area-based type of visualizations, which use nested rectangles to chart data that has some type of hierarchical organisation. The values are encoded within the surface of each rectangle. New layouts have been designed to substitute the original rectangle, resulting in indented trees, icicle layouts, sunbursts, nested circles and so on.

(i) **Map projections**: Plotting data on a map is the *de-facto* manner to visualise datasets with a geographical dimension. Map projections allow to represent data as individual markers on its surface, enclosing areas limited by its given boundaries and overlap layers with extended information.

(j) **Node-link diagrams**: Sometimes referred to as *Network graphs*, it is a structure formed by nodes (vertices) and links (edges) that connect them. They allow to draw relationships between entities using an easily understandable visual representation, and thanks to graph theory they are able to be analysed using a mathematical foundation. This is why they have been widely used to study social connections between individuals, chemical components, molecular structures and others. Graphs can either be directed (the direction of the connections matter) or non-directed (the connections are reciprocal between nodes), and their layouts might deeply alter what insights can be extracted from them (i.e., in Social Network Analysis the force-directed-layout is usually used, as by simulating a gravitational system it brings together nodes that are somehow more related than others).

These visualization types allow to encode nearly any imaginable value, even if it is not in the most effective manner, reason why so many different visual representations exist, and new are being developed every year. However, they establish a perfect foundational visualization pool that enables further enrichment.

In Table 4.1 we have matched the compliance level of each visualization type to the *visual variables* featured at the beginning of this section, according

to the criteria established in Cleveland's work. Visualizations can either have a **high** (H) compatibility with the corresponding visual variable, meaning that any variation rendered in this fashion will be clearly perceived by the user, a **low** (L) compatibility level when differences are noticeable but not undoubtedly; or no compatibility at all. The selected visualizations constitute a complete showcase regarding visual encodings usage, each of them fitted to exhibit value variations in different ways.

	Position	**Length & Direction**	**Area**	**Volume**	**Colour & Shading**
Line charts	H	L	L		
Bar/Column charts		H	L	L	
Radar charts	H		L	L	L
Heatmaps					H
Box plots		H	H	L	
Scatter plots	H	L			L
Pie charts			H	L	
Tree maps	L		H	L	L
Map projections	H		H		L
Node-link diagrams	L	L			L

Table 4.1: Visualization types compliance to Jacques Bertin's visual variables.

Likewise, and thanks to the work performed through Chapter 3 (specially in Section 3.3), we could use the properties' datatypes essence and use them to help in the chart selection task. Most charts are produced by encoding numerical values within the available visual variables, but having extra datatypes such as text strings, boolean values, composed datatypes and so on, would allow to include these dimensions in the charts. Table 4.2 explores the compatibility of each visualization type with the proposed primitive datatypes.

	Integer	**Float**	**Boolean**	**IRI**	**String**	**Datetime**	**Geo.**	**Categorical**
Line charts	✓	✓				✓		
Bar/Column charts	✓	✓	✓					✓
Radar charts	✓	✓				✓		✓
Heatmaps	✓	✓	✓					
Box plots	✓	✓						
Scatter plots	✓	✓			✓			✓
Pie charts	✓	✓						✓
Tree maps	✓	✓			✓			✓
Map projections							✓	
Node-link diagrams				✓			✓	✓

Table 4.2: Visualization types compliance to the primitive datatypes proposed in this research.

4.2 Quantitative messages

Jacques Bertin's *visual variables*, ranked by Cleveland and McGill, provide an intuitive filter when searching for a graphical representation for our dataset. However, due to the extensive variety of visualizations, many different charts might match our design perspectives, and further guides to select an efficient canvas are required.

In order to adapt the visual representations in the best possible way to the user's expectations, a generic LOD visualization tool should be able to take into account the analysis intention when generating the visual outputs, supporting users for a successful data analysis (Grammel et al., 2010). When dealing with a dataset, users usually have preconceived ideas about the results they want to convey, or at least, have a general impression on how to obtain insights from the collection. In this section, we have analysed different approaches that have dealt with message communication and how to select the most appropriate visualization type.

One of the most referenced works to deal with chart selection taking into account the nature of the analysis is the *Chart Chooser* proposed by Dr. Andrew Abela in Figure 4.3 (a full-page version of the diagram can be found online in his website[1]). The chart is very easy to use by visualization neophytes, who can find a suitable representation for their data just answering a couple of questions: what goal the analysis pursues and how many variables are going to be depicted in the chart. Many of the used visualization types are already considered by our graphics selection. For those analysts accustomed to work with *Microsoft Excel* to extract data from spreadsheets, the online *Chart Chooser* tool provided by *Juice Analytics*[2] lets any visitor filter by quantitative message, and download the generic template to generate the visualization easily. The filtering is also based on Dr. Abela's recommendation chart, but has been kept up to date including new templates and providing version support.

[1] http://extremepresentation.typepad.com/files/choosing-a-good-chart-09.pdf

[2] http://labs.juiceanalytics.com/chartchooser/index.html

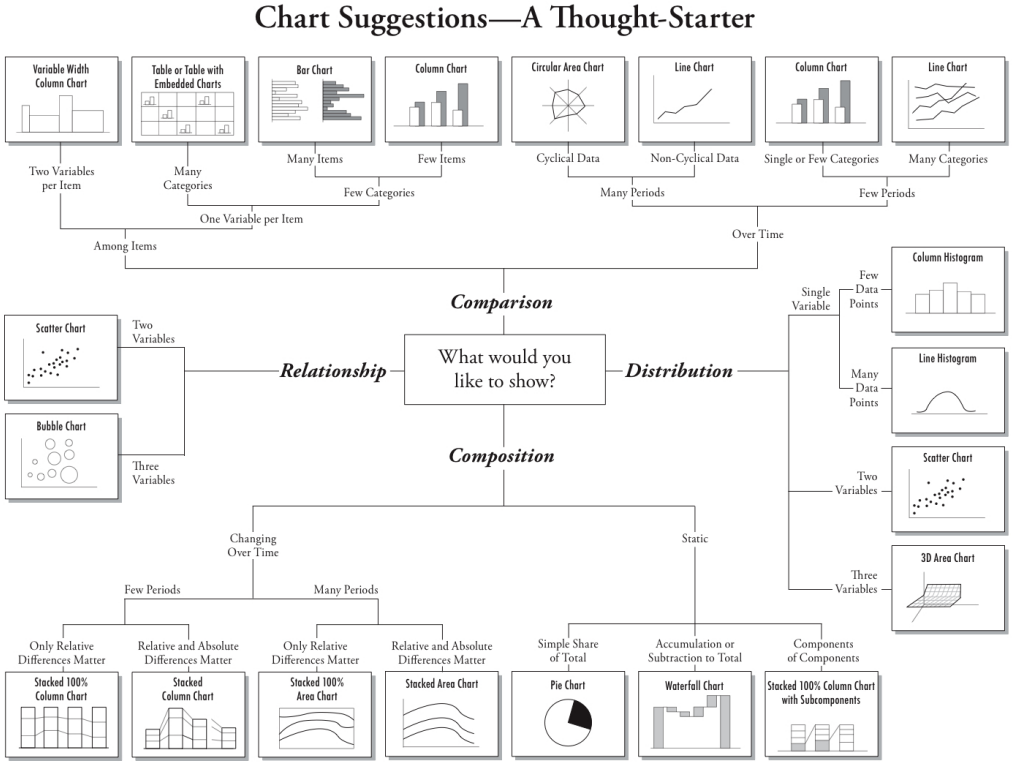


Figure 4.3: Dr. Andrew Abela's chart chooser diagram based on the quantitative message that the analysts wants to communicate.

Despite the usefulness of *Dr. Abela*'s chart chooser for visualization novices, experienced visualisers have expressed some doubts about its convenience for more advanced users[1], who usually seek for more refined representations of the data. In order to provide a solution for more seasoned users, Stephen Few identified up to eight different quantitative messages that analysts can use to present their findings in a visual fashion (Few, 2009). These categories were also used by *Katy Börner* and her colleagues (with a slightly different nomenclature) whilst designing their *Science Maps*[2], a series of visualizations and studies that analyse how research fields have evolved in the North American academic landscape (Börner, 2015).

[1] https://perceptualedge.com/blog/?p=2080
[2] http://scimaps.org

Due to their broad message coverage, and the acceptance of the categories in the literature (both in data visualization and statistical analysis), we add these eight quantitative messages in our LOD visualization approach, as many of them have a direct mapping to the data we have previously inferred for each property. The selected messages are described as follows:

- **Categorisation**: This task consists on putting together data items with similar meaning or features, in what are called *clusters*, *groups*, *classes* or *categories*. These categories can either be manually defined beforehand or automatically computed using clustering techniques.

- **Comparison**: One of the most referred expectations when working with a dataset is to contrast equivalent data items to easily identify similarities and divergences. Visual aids to compare multiple data items often rely on rendering them side to side. Sometimes, this comparison is performed in order to study the deviation from a reference set of values.

- **Composition**: When data items can be organised to form part of a whole, we talk about data composition. This presentation provides a full description of the entity under analysis focus and its components in a hierarchical layout.

- **Distribution**: Looking for data dispersion within any dataset, distribution analyses let users expose underlying patterns, discover frequencies, detect outliers, identify missing gaps and limit the range of values. Some distributions are well known to statisticians, so when a certain feature exhibits a similar behaviour, data forecasting seems more feasible.

- **Geospatial location**: If resources are assigned a geographical position, plotting them over a map can help understand how data entries are spread out on a physical space. Based on the dataset's topic, this analysis is able to unearth invaluable insights.

- **Ordering**: Also known as *sorting* or *ranking*, it makes reference to putting the objects in a particular sequence, due to a certain feature

selected to act as the arrangement dimension. The main reason behind this message is to provide meaning from a human perspective.

- **Relationship**: When data individuals are connected between them, how quantitative variables affect one another can uncover correlations, clusters and other complex associations. In a similar fashion to distribution analysis, studying relationships is a common tool on any statistician's backpack, requiring to be performed carefully to avoid reaching wrong conclusions.

- **Temporal trends**: Observations taken at different times are specially relevant in the Business Intelligence (BI), scientific and data journalism fields. Being able to study how feature values evolve through time allows to understand the past, observe the present and predict the future, therefore its strategic value to any interested user. Time series allow to detect, among others: trends, variations or cycles.

As with visual variables and primitive datatypes, Table 4.3 exhibits the compliance level between the selected visualization types and the message goal categories, giving an overall impression about those that best fulfil the analysis expectations the users' might have.

	Cat.	**Compa.**	**Compo.**	**Dist.**	**Geo.**	**Ord.**	**Rel.**	**Temp.**
Line charts		✓		✓			✓	✓
Bar/Column charts	✓	✓	✓	✓		✓		✓
Radar charts		✓						✓
Heatmaps		✓			✓			
Box plots		✓		✓				✓
Scatter plots		✓		✓			✓	✓
Pie charts	✓		✓					
Tree maps			✓			✓		
Map projections					✓			
Node-link diagrams	✓				✓		✓	

Table 4.3: Quantitative message analysis compliance by most used charts.

4.3 Recommendation heuristics

Once we have discovered how the knowledge extracted through Chapter 3 benefits the representation selection, and how to take into consideration visual techniques in order to produce coherent and meaningful visualizations, we can define our heuristic approach towards visual recommendation.

Exploratory Data Analysis proposes a set of techniques and procedures that look for the best ways to summarise datasets by analysing their most promising features. In order to achieve those goals, EDA often appeals to visual methods, which are able produce a quick overlook of a dataset's contents and provide hints to guide the analysts' approaches.

However, EDA relies on visualization as a tool to ease data exploration, not as an end in itself. The visualizations generated while exploring a dataset are not intended as a resulting output, and they are expected to be drafted, re-done and rejected through the analysis. Moreover, these visualizations are usually simple, as they are quicker to generate and do not need to pay any attention to aesthetics or details' refinement.

Let's imagine a simple data analysis scenario, as the one depicted in Figure 4.4. An analyst has collected a small dataset with all the height measurements of the students from a particular classroom (**A**). She wants to know how these heights are distributed through her little population, and extract some conclusions from the experiment. Following an Exploratory Data Analysis approach, the first task would consist on homogenising the values, preprocessing the input data in order to be able to work with it (**B**). As she is the one who has collected the data, we are going to assume that all measures have been taken in centimetres (following the standardised units proposed by the International System of Units) and all the data is already stored in a structured format, denoting the name, age, gender and height of each individual.

Now, she should start plotting charts with the objective to understand the dataset. First she loads all the data in her favourite analysis tool, and taking the recommendations made by EDA, she decides to create some sample charts to see if she can get any insight by producing the following visualizations (**C**):

- Two box-plots (one for each gender) with all the measurements taken, depicting the distribution and average values for each group and its outliers, if any (**D**).
- A simple bar/column-chart with the average height for each gender (**E**).
- A scatter plot with all the height values graphed by age, and with their gender encoded as each point's colour (**F**).
- Two line-charts, one for each gender, with all the measurements. The x-axis could be ordered by the student's ID number, and the lines would highlight differences between adjacent people (**G**).

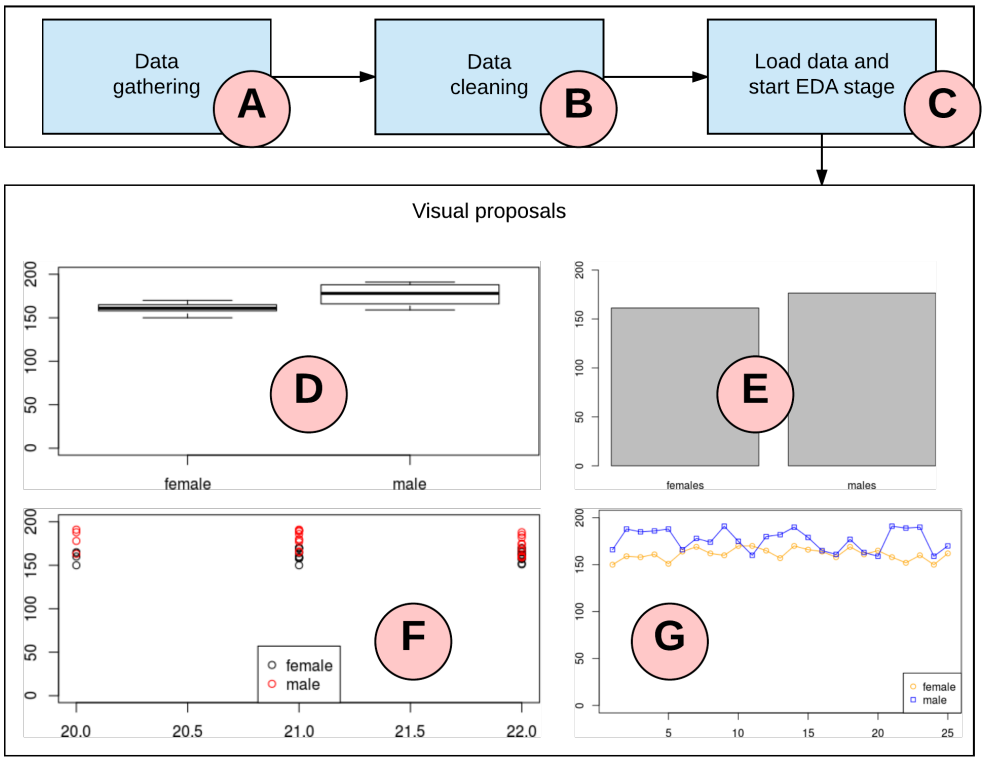


Figure 4.4: Simple workflow of an EDA approach to deal with a dataset. EDA does not impose the usage of any particular procedure, so users are free to design and play with any techniques they feel comfortable with.

Now, it would be up to the analyst to choose the visualization she is more comfortable with, and which best fits the dataset. She could select any of them, but the compliance analysis we have performed in the previous sections might guide the filtering and selection. Thus, the chart which will lay the foundations for further study will be the first one (**D**), as it shows the distributions of two different groups in an effective manner, being easy to spot by a non-expert eye the contents of the dataset. However, the selected chart might still benefit from some refinement, such as: using a colour palette fitted for colour-blind people, use patterns instead of solid colours, having a non-0-based y-axis to have a clearer perspective on the differences and so on. Figure 4.5 exhibits a more polished version of the boxplot in (**D**).

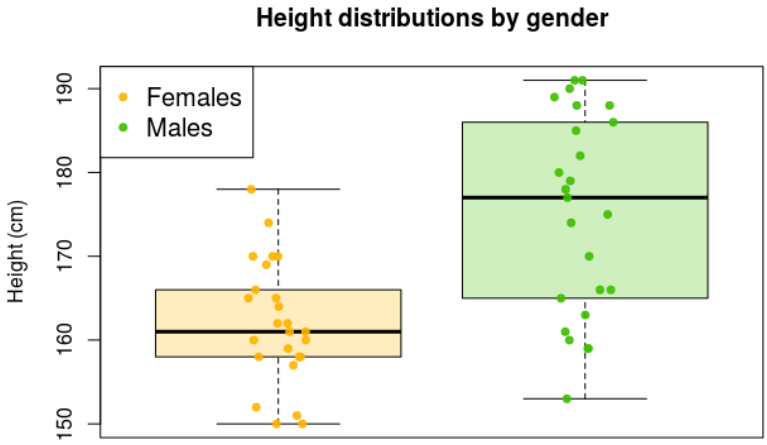


Figure 4.5: Refined final version of the chart that will be presented after the analysis carried out.

Before starting our chart recommendation, we must warn that very domain-specific visualizations will not be produced. As our approach is intended to be agnostic, non-dependant on the dataset that is taken as input, the produced graphs and charts will try to be the best possible ones, even if they should require further refinement. There are some visualizations which have been broadly used in commercial applications and the literature for really specific (usually of a scientific nature) field, such as large DNA samples visualization, molecules representation and so on. The **Scientific Visualization** (SciVis)

field deals with these types of depictions due to the required knowledge, and has traditionally followed a divergent path from **Information Visualization** (InfoVis) (Rhyne et al., 2003; Weiskopf et al., 2006).

To produce the visualization candidates, we propose a selection algorithm that takes the computed data up to this stage of the visualization pipeline, and searches the representations that best fit those features. To be able to produce an instance of a visualization type, it needs to be described to the system. We achieve this goal by defining a pragmatic visualization template for each resulting chart.

Every chart is formed by a set of *foundational components*, the visual elements that allow to encode the values used to produce the whole graphic representation. Through the definition of the variables involved in a *foundational component*'s description, we will use a bar/column chart as an example:

- **Visual variable**: Visual element as described by *Jacques Bertin*, which will depict a singular data dimension. In a column chart, data is rendered through rectangles, whose lengths and colours are directly connected to the data being mapped.

- **Supported primitive datatypes**: Set of primitive datatypes supported by the *foundational component*. The rectangles in a column chart support both integer and float values to define their height, and colours can be used to differentiate grouped data.

- **Data encoding**: Each *foundational component* needs to state how data is going to be mapped in the resulting visualization. Integer or float values will be mapped as numerical values, whereas colours might be coded as RGB strings or HSV triples.

Additionally, as many visualizations make use of Cartesian Coordinates (or any similar reference) to spatially set the visual elements, a generic visualization template should allow the description of its axes, with some distinctive features from the *foundational components*:

- **Axis**: In bi-dimensional charts this variable will either adopt a X or a Y value, but a different axis might also be provided if required (Z axis in 3D graphics, for example).

- **Ordering**: Data instances might be ordered by a particular criteria along an axis. For example, time-series are usually visualised using line charts that use datetime values for one of its axes. Value ordering is a topic that has been closely studied by different researchers and information architecture practitioners. Psychologist *S. S. Stevens* proposed a four-levels classification that has been widely used in statistics (Stevens, 1946):

  - Nominal: Mutually exclusive discrete data that does not have any kind of order, being understood as pure labels.
  - Ordinal: Data values that have a clear order, but difference between values is hard to notice (e.g., Likert scales).
  - Interval: Data values are ordered, and the differences between values are meaningful and measurable (e.g., temperature in Fahrenheit or Celsius degrees).
  - Ratio: Data has an inherent order, differences between values are meaningful, and a clear definition of *absolute zero* exists (e.g., weight or temperature in Kelvin degrees).

  Even though these levels are often found in reports and articles, several authors have suggested alternative taxonomies to rank quantitative values (Velleman and Wilkinson, 1993). The proposal by *Frederick Mosteller* and *John Wilder Tukey* (Mosteller and Tukey, 1977) comprises the following items:

  - Names: Equivalent to the *Nominal* level of measurement described by *S. S. Stevens*.
  - Grades: Labels with some kind of logical ordering (e.g., Freshman, Sophomore, Junior and Senior).

  - Ranks: Integers in which the order is preserved.
  - Counted fractions: Real values bounded between the [0, 1] range, allowing to represent percentages.
  - Counts: Non-negative integers.
  - Amounts: Non-negative real numbers.
  - Balances: Unbounded positive or negative values.

  For our description purposes, we will use the list proposed by *Mosteller* and *Tukey*, providing as value the set of items the axis supports. The scale of the axis will need to take into account these levels, and adapt its range accordingly.

- **Data encoding**: As in *foundational components*, how data values will be inserted into the chart must be declared.

Finally, the visualization representation itself will be described. In its description we include the message goals the template complies with, listing the message types each representation is best suited to shape. These descriptions can be easily formatted within a JSON structured schema, so that all visualization types our system is able to produce could be assigned its corresponding filled template. Code 4.6 shows how the visualization template to produce *column charts* is described.

The last task of the visualization pipeline should be the generation of the visualization. Once the dataset is explored, the resource on which the user wants to focus selected, and its features analysed; the system can select the visualizations that fit the exploratory needs, and apply a visual transformation that takes the input data and produces an output representation in the corresponding view.

In order to perform this transformation, each value should be mapped to its corresponding visual variable, using the visualization library the developer is more familiar with. For our LOD visualization prototype we have worked with both Google Charts and D3js, two JavaScript libraries that are widely used for visualising data on the Web interactively.

```
// VISUALIZATION TEMPLATE: Column chart

{
    "name": "column_chart",
    "message_goals": [
        "categorisation",
        "comparison",
        "composition",
        "distribution",
        "ordering",
        "temporal-trends"
    ],
    "foundational_components": [
        {
            "visual_variable": "length",  // height
            "supported_primitive_datatypes": ["integer", "float"],
            "data_encoding": ["numerical"]
        },
        {
            "visual_variable": "colour",
            "supported_primitive_datatypes": ["categorical", "boolean"],
            "data_encoding": ["numerical", "string"]
        }
    ],
    "axes": [
        {
            "axis": "x",
            "ordering": ["names", "grades", "ranks"],
            "data_encoding": ["numerical", "string"]
        },
        {
            "axis": "y",
            "ordering": ["counted_fractions", "counts", "amounts"],
            "data_encoding": ["numerical"]
        }
    ]
}
```

Code 4.6: The JSON code for the *column_chart* visualization template. The features describe how data can be mapped to produce the graphical output, which will be later transformed using a visualization library to generate the final depiction.

4.4 Summary

Data visualization is much an art as it is a science, and the creativity and previously developed skills could suppose a great difference at the time of producing the end graphic. Each year new visualizations are proposed, and some visualisers design representations that astonish the audience due to their mix of effectiveness and beauty. The artistic dimension of data visualization makes it difficult for an algorithm to create a representation as full of meaning as those produced by a human being. However, there are multiple features we can rely on in order to produce insightful charts.

In this chapter we have proposed a methodology to describe and categorise visualization types, allowing us to recommend the most suitable graphics for a particular analysis task. Exploratory Data Analysis encourages the development of quick and simple visualizations that might highlight interesting facts hidden in an unknown dataset. To filter the most adequate visualizations, we have studied the compliance level of 10 different visualization types to the following features: visual variables, datatype support and analysis goal.

According to a study on how lay users construct visualizations during exploratory data analysis (Grammel et al., 2010), three major activities were identified as part of the visualization process: data attribute selection, visual template selection and visual mapping specification. The first task is addressed in our approach by all the work in Chapter 3, which allows to explore and select any class or property from SPARQL endpoints. The visual template selection deals with choosing the visualization type that data values would be encoded into. In this section we have provided several features and its compliance levels to a selected pool of ten visualization types that helps users find a suitable representation for their analysis.

Finally, the visual mapping specification is in charge of connecting data attributes to visual properties. The last section of this chapter describes how to define visualization templates in order to map values to visual representations, providing the variables that are common amongst visual representations. Profiling each visualization in an structured format, our algorithm is able to select the graphics that are most adequate to visualise the data.

The most serious mistakes are not being made as a result of wrong answers. The truly dangerous thing is asking the wrong question.

Peter Drucker

CHAPTER

5

Design and evaluation of a LOD visualization prototype

All the proposed contributions developed through chapters 3 and 4 are aimed towards improving the ways we can explore, visualise and interact with LOD datasets with which we are not familiar, or those that fall out of our knowledge pond.

Between our goals list for this research (see Section 1.2), we proposed the design and implementation of a generic visualization prototype tool for any dataset published under LOD's principles. To evaluate the suitability of our approach, we have presented the tool to a group of real users.

This chapter presents the design, development and evaluation stages of this visualization prototype, named *John Snow* in honour to one of the pioneers in data visualization, the 19th century physician whose visual approach identified the water pipe involved in the cholera outbreak in London between 1853-1854.

John Snow features the whole visualization pipeline, from raw semantically annotated data to interactive visualizations, using the modules and techniques developed through this dissertation.

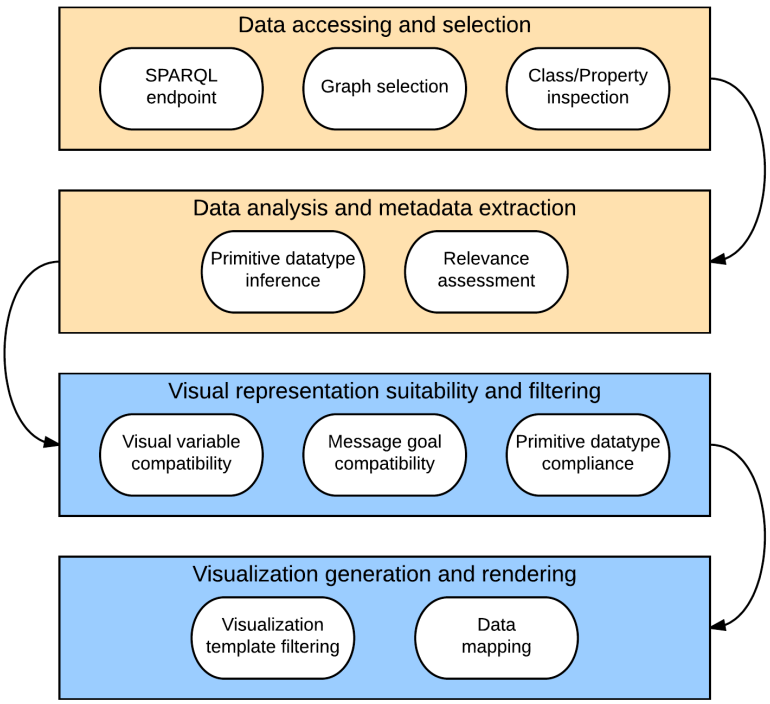


Figure 5.1: Diagram exhibiting the modules of our visualization pipeline prototype: *John Snow*.

Figure 5.1 describes the different modules that constitute the prototype. As stated in the goals of this research, the pipeline takes as input raw semantically annotated data, and produces visualizations as output.

The first two modules (with a pale orange background colour) are explained in detail through Chapter 3, dealing with the data accessing and selection, its structural analysis and the metadata extraction process to evaluate the relevance of the dataset's contents.

After profiling the dataset under focus, the graphic generation stage is launched (as presented through Chapter 4 and depicted with light blue background in the diagram). Taking the information inferred in the previous processes, a filtering algorithm detects the graphical options that are most suitable to depict the information, using visualization templates that contain the whole description of those features. Once a visualization type is selected, the data is mapped to the visual elements that produce the output representation.

5.1 Design and implementation

In order to improve LOD visualization we followed a cyclic approach, in which we designed a basic visualization tool that resembles others in the *State of the Art* and added a new feature in each loop, testing it with real users to get valuable feedback that would lead the design and development of new features.

As stated in our research goals (Section 1.2), our visualization tool was conceived to be browser-based from the beginning, avoiding the incompatibilities that having the tool deployed for different systems could have. Thanks to the implementation of a web application, we could ensure that all people could access the same, exact version of the tool, and minimised any installation or configuration issues. All the algorithms responsible for retrieving the data, analysing LOD resources and calculating the different metrics designed for this research have been encoded in *Python*[1], a general purpose programming language we were already familiar with and which is actively maintained and widely used in academic environments. To develop the web tool we relied on *Django*[2], a Python-based web framework to prototype and implement websites in a quick manner. To deal with all the interactions with SPARQL endpoints we used *RDFLib*'s[3] libraries stack, allowing us to process SPARQL responses in an easier fashion. Finally, we both used a local installation of *Openlink Virtuoso*[4] and *PostgreSQL*[5] to store all the information we needed. The former was primarily used to upload some test datasets, as local accesses were more reliable to perform than accessing the original endpoints, and was a better suited approach to perform all the platform's tests. The latter stored all the extra metrics computed during the dataset's retrieval, to avoid asking for the same information with each new page update.

Within our scope, described in Section 1.3, we will only consider those datasets that are publicly available on the Internet through a SPARQL endpoint, which can be directly accessed and queried to retrieve data in an

[1] https://python.org
[2] https://djangoproject.com
[3] https://github.com/RDFLib
[4] http://virtuoso.openlinksw.com
[5] https://postgresql.org

structured format. Therefore, users must point out the URI of the SPARQL endpoint they want to have analysed by *John Snow*, as it serves as the required minimum input of our visualization prototype.

Once a valid URI is entered into the system, it initially retrieves the list of available graphs. Data publishers usually take one of the following approaches in order to make available their datasets: either they put each dataset under a different URI, or graphs are generated for each dataset under the same URI. Whatever the selected option is, we will always have at least one graph associated to each SPARQL endpoint (when there is only one graph we name it the *default* graph). When executing a SPARQL query, we should always specify the graph it is performed against, as to only retrieve the triples related to that dataset (we can omit the graph name when it is the default one). For each graph the following figures are retrieved: number of triples, unique entities and totals of different classes and properties (see Figure 5.2).

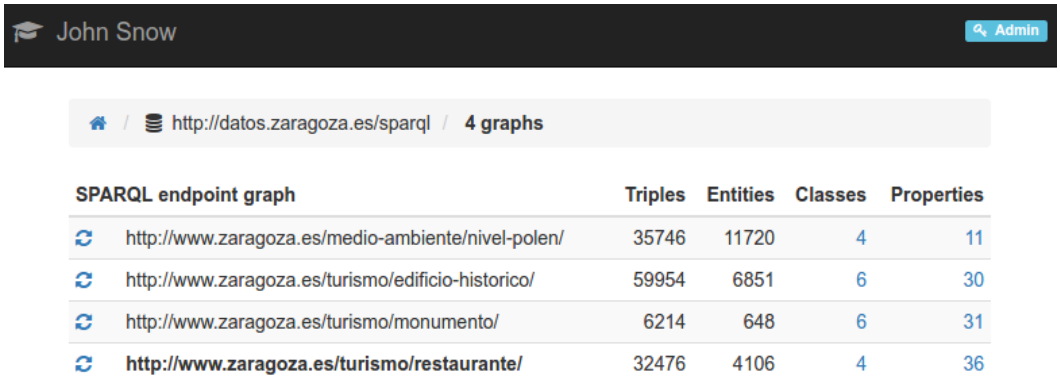


Figure 5.2: Graph list excerpt from the SPARQL endpoint offered by the City Council of Zaragoza (Spain).

Both the number of triples and unique entities allow to easily figure out the size of the dataset, and are quite common within other LOD tools. The amount of different classes and properties are clickable elements, which would allow an interested user to explore the list of elements used to describe the data contained within the SPARQL graph.

5.1.1 Class views

By selecting the number of classes in Figure 5.2, a table listing all the classes used to describe the contents of the dataset is rendered. For each class its name, number of unique individuals belonging to it and the number of related properties (those that are used in triples whose subject belongs to the class under study) are displayed, as shown in Figure 5.3.

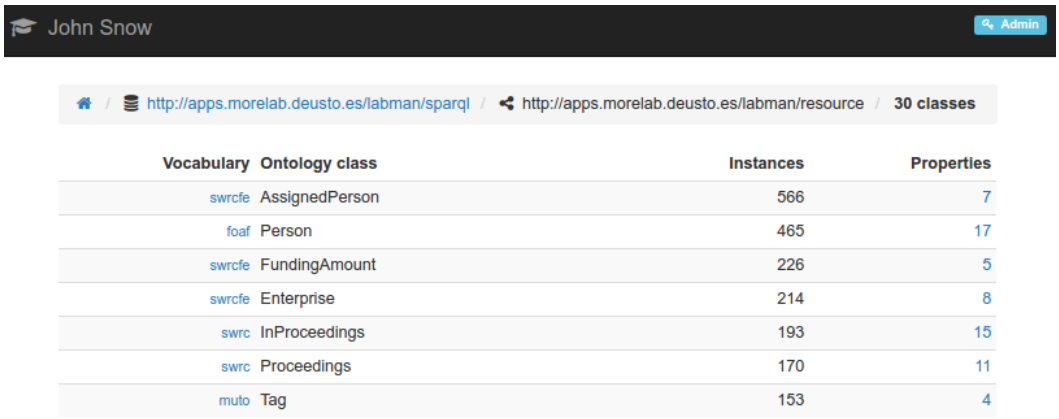


Figure 5.3: Excerpt of the ontological classes listing for the SPARQL graph that contains the information of our Research Unit (MORElab).

The name of each class is represented by its ontology's namespace (a partial URI shared by all resources associated with it), and the class' name itself. To ease readability, namespaces are looked up in *prefix.cc*[1], a service for RDF developers to search for commonly shared prefixes of well-known ontologies. If a prefix is found, it is rendered instead of the full ontology URI, and remembered to substitute long namespaces in further analyses.

We also provide a node-link graph of the listed classes. Each node represents an ontology class and its colour encodes the ontology the class belongs to. The size of the node reflects the number of instances using that class. If two classes share a property when describing their resources, a link (or edge) is established between them. The information for the diagram is retrieved each time a SPARQL graph is queried, so it always represents the latest version of

[1] `http://prefix.cc`

the data. In Figure 5.4, the node-link diagram for the classes describing the contents of our Research Unit (i.e., MORElab[1]) is rendered.

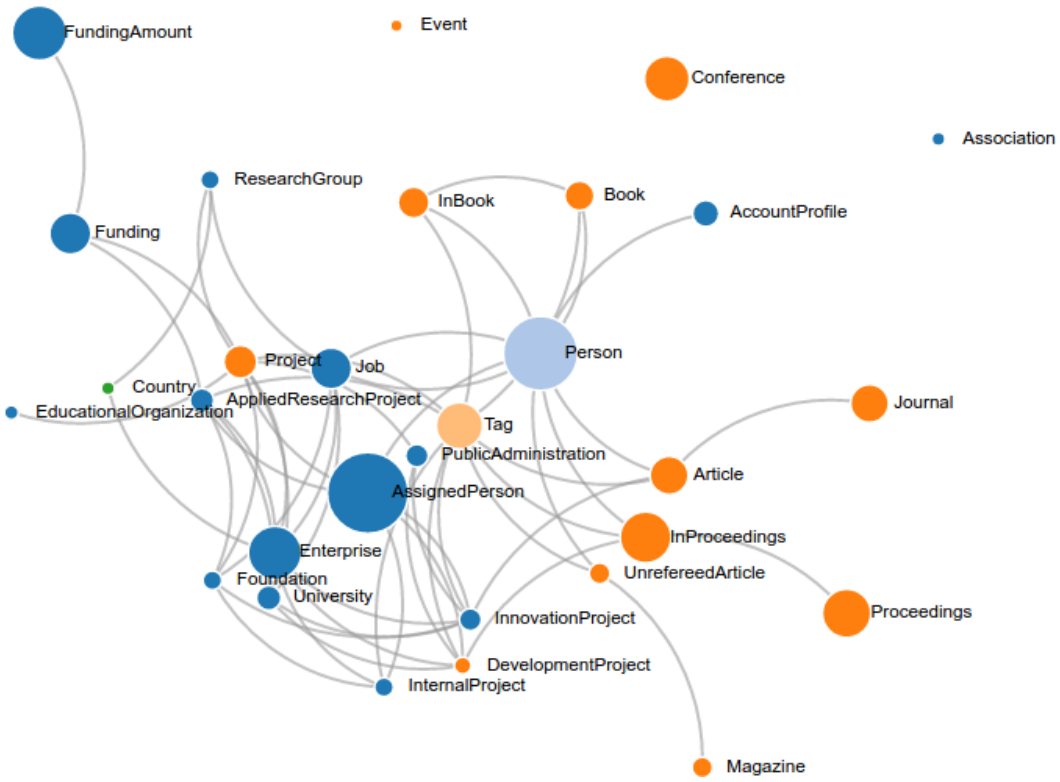


Figure 5.4: Node-link diagram of the classes that are used to describe our Research Unit's data. The classes (nodes) that share properties are connected through links (edges). The colour of each node symbolises the namespace it belongs to, and its size the number of resources described within each class.

5.1.2 Property views

Similar to class views, *John Snow* lists all the properties when selecting the number of different properties within a SPARQL graph. Together with the name of each property (split between namespace and property labels) and number of instances using the property, the visualization tool shows the detected primitive datatypes for each property. Using highlighted icons the interface shows how properties have been classified using the categorisation

[1] http://morelab.deusto.es

explained in Section 3.3, this is, the values that have been detected to belong to the *integer*, *float*, *boolean*, *IRI* or *string* categories, or those that have some kind of *datetime component*, *geographical component* or *categorical* information encoded within the values.

Should a user want to retrieve an example value set for the selected property, the '*plus*' icon queries the SPARQL endpoint to retrieve 20 random values (the length limit of the query can be modified in the tool's configuration). These values might belong to different classes, as the query asks for triples using the target property, but does not filter by results of a particular class.

An example of a *property view* is displayed in Figure 5.5.

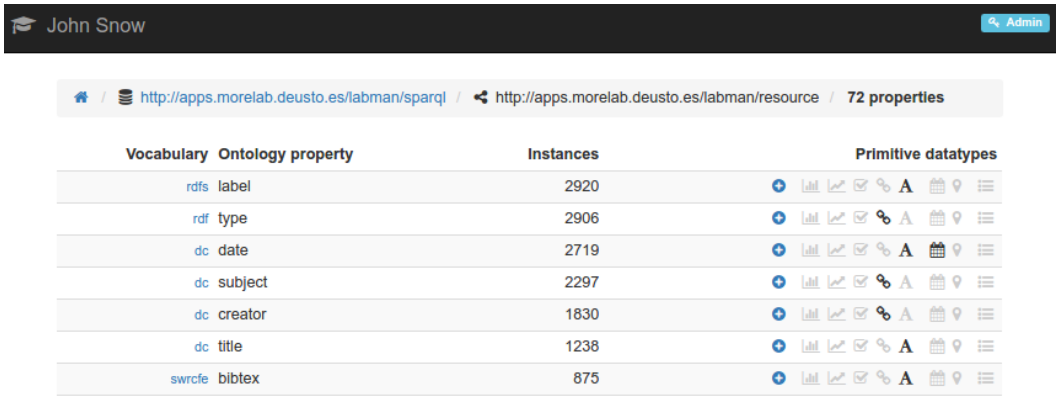


Figure 5.5: Ontological property list of a particular SPARQL graph.

5.1.3 Properties within a class views

Finally, the view which implements most of the contributions explained in Chapter 3 is the one listing all the properties within a selected class. The reason behind implementing most features in these views is that during early stages of the research, when we were looking at how people searched for information in semantic datasets, we realised that it was common practice to first look for the resource category (class) which could contain the information about the instances the analysis was focused on, and later on inspect its features (properties) to extract some information about its behaviour. This process is similar in other information search scenarios as well, and is encoded

in Shneiderman's mantra: "*Overview first, zoom and filter, then details-on-demand*" (Shneiderman, 1996).

Thus, when a user selects the number of properties related to a class, they are all listed in tabular format. On the top of the interface extended metadata can be displayed, to remember some computed metrics about the class under focus. Each property is listed using the previously explained namespace-name duo, as exposed in Figure 5.6.

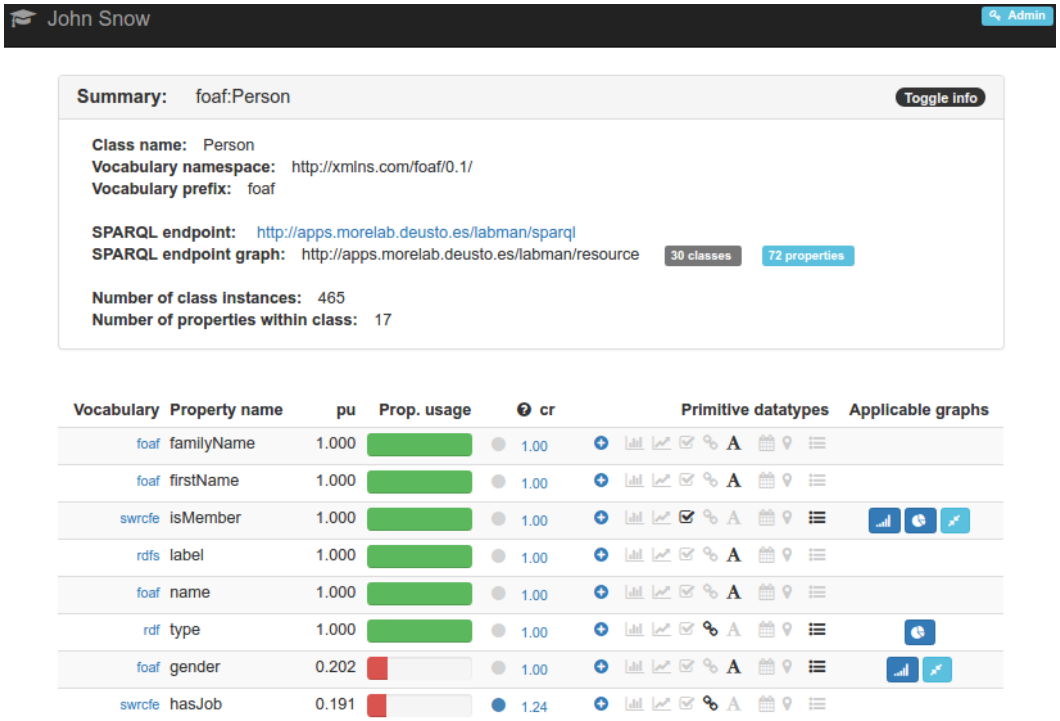


Figure 5.6: Ontological property list within a particular class. These properties are used to describe the resources that belong to the same entity, and are depicted showing some of the contributed metrics developed for this research.

The *property usage* and *completeness ratio* metrics from Section 3.4 are displayed in each row, both in numerical format and through a visual colour coding. For the *property usage* statistic, its value (spanning within the $(0, 1]$ range) is rendered, showing the percentage of class instances using this property. The related colour bar will be painted in green if pu's value above 0.5, orange if it is comprised between 0.25-0.5, and red otherwise, providing a

visual clue of how relevant each property is within the class, and enabling a quick sorting of values.

On the other hand, the *completeness ratio* is represented by its value, which addresses how the property is used, assigning just one or more values for each individual class instance. The colouring in this case just displays a different colour (a blue dot in the portrayed example) when more than one triple is published using that property for a single subject. When instances are described using the property repeatedly, a histogram on its usage distribution can be visualised by selecting the *completeness ratio* value on the interface.

Figure 5.7 exhibits an example of the produced histogram. In this particular case, for the selected property, 72 unique instances only had one triple associated to the property, 14 individuals had exactly two triples using the property, and 7 instances provided 3 different values for the feature.

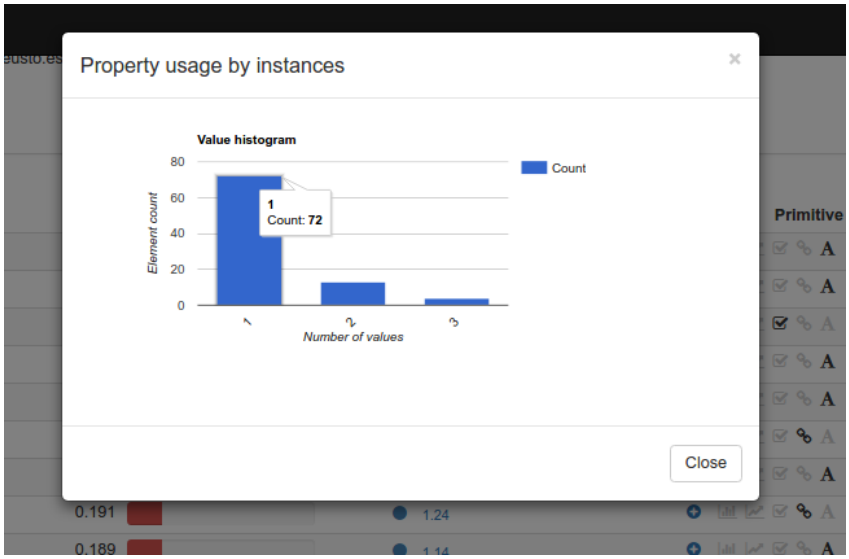


Figure 5.7: An histogram detailing the *completeness ratio* value for a specific property. The chart depicts how resources are distributed when grouped by the number of triples they produce for the same property.

As with property views, any user can retrieve a set of random values for each property, but in this case, limited to those that belong to the scope of the class under study. The primitive datatype inference is also performed with this limitation, inferring only values that are part of the class' description. This

might help improve the datatype inference stage at class level, particularly on those scenarios on which some well-known properties are used with different purposes through the dataset.

Finally, to accomplish the goals envisaged by this visualization pipeline's proposal, the tool must produce some visualizations that represent the contents of the class' properties whenever possible. Therefore, after performing all the data-driven background analyses, if the data visualization heuristics are able to find a representation match for any of the class' properties, an icon depicting the available chart-type would be displayed. Figure 5.8 shows an example *properties-within-a-class* view. All the properties that might be visualised present a selectable button at the end of their row. Once clicked, the user is redirected to the visualization's view.

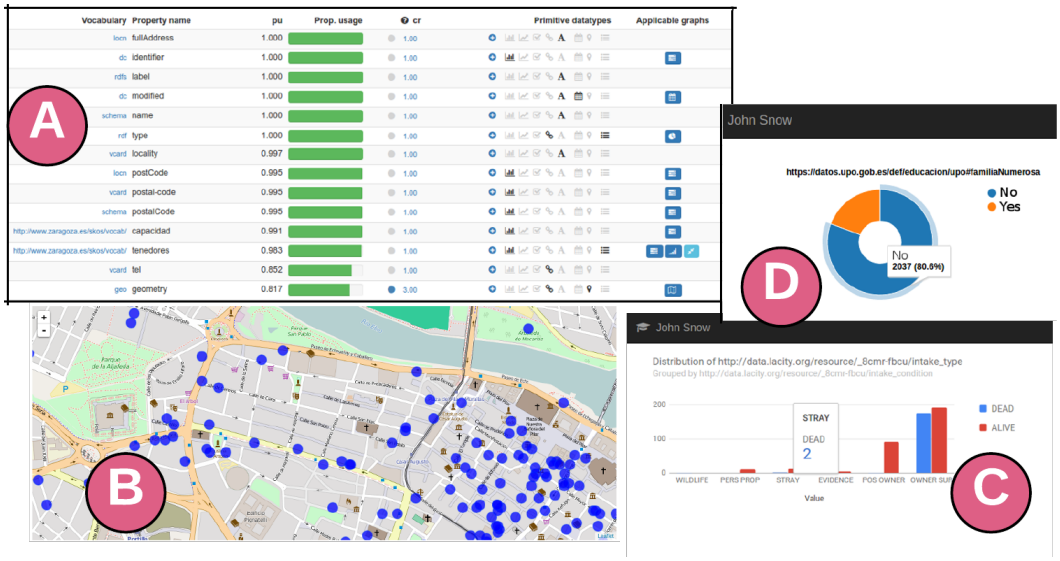


Figure 5.8: By taking into consideration all the work performed in this dissertation, *John Snow* presents a representative icon when it is capable of visually depicting any property's contents (dark blue icons in the right side of screenshot A). Selecting any of the icons will render the visualization using the most appropriate visual abstractions, as can be seen in example views B, C and D.

All chart icons are dark blue, except for the cases when a property can be combined in the same chart with another one from the same class. To make users aware of this possibility, a lighter blue icon is used instead. Once a combinable property is selected, a list of compatible properties are presented.

By choosing any of these properties, a graph will be generated that combines both. The visualization heuristic will look for compatible features, so for example, if two numerical properties are selected, a scatterplot might be easily generated, or if the user selects a numerical and a categorical property, the algorithm would perform extra queries to determine if a pie chart is the more appropriate representation (the feature's values form a whole), or if it should search for other graphical options.

5.2 Participant selection

With the intention to check the validity of the designed approach, we decided to conduct an experiment using *John Snow* with a group of volunteers. User evaluations are a traditional approach to validate research in the visualization field, since humans are ultimately the recipients of the graphic representations due to their ability to easily interpret and understand them.

As user evaluations might be heavily influenced by their biases and previous experiences, solid methodologies and approaches are needed to overtake any partiality that might show up. Recommendations to ensure a valid evaluation are usually focused on two categories: sample population selection and how the evaluations are carried out.

In order to select our user sample, we decided to follow a *convenience sampling* approach, a non-probabilistic sampling method in which participants are selected by their ease of access, availability and willingness to take part voluntarily in the experiment. After making a public request for potential users, a total of 16 people took part in the evaluation.

Even though the number of respondents might seem small, the works that had users involved during the State of the Art evaluation had sample sizes between 3 and 7 participants, making it difficult to extrapolate the findings to a wider population.

Due to the nature of the sampling, we can not guarantee the statistical generalisation of the extracted conclusions. Nevertheless, the results can be accepted if the selected respondents are not supposed to cause a bias in the answers, this is, if the sample is representative of a larger population. In order

to reason about the statistical representativeness of the group, we categorised the respondents according to the user profiles described in Section 2.2.3, whose characteristics are featured in Table 5.1.

	Web browsing	**SW and LOD knowledge**	**Deep domain knowledge**
Lay users	✓		
Technical users	✓	✓	
Expert users	✓		✓

Table 5.1: Skills required to belong to each of the profiles defined in this dissertation for potential LOD visualization users.

In our particular case, all the 16 respondents have a strong background in Information Technologies (IT) (they are all in possession of an engineering degree), guaranteeing that the interaction with a web browser was not an issue (basic requirement of all 3 user profiles). However, only 6 of them declared being familiar with the Semantic Web, and therefore, were categorised as *technical* users (Table 5.2). No *domain experts* took part in the experiments, as we made sure before the experimentation that all the used datasets were unknown both in contents and domain to all potential participants (as eventually confirmed by them). As most modern societies have a high level of IT penetration among their population, with most inhabitants browsing the Web on a daily basis, we can extend the conclusions to a wider audience with a certain level of confidence. All users' opinions are hence split according to their user profile: lay users or technical.

User ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Profile	T	L	T	L	L	L	L	L	L	L	T	T	L	T	T	L

Table 5.2: Respondent belonging to the profiles described above: Lay users (L) and Technical users (T). Our participant group was formed by 10 lay users and 6 technical users.

To better understand the skill set of our volunteers, we asked them to assess their expertise level in the following data analysis and exploration related areas: ability with spreadsheets software and data visualization background.

Each respondent evaluated its skill level on a 1 to 10 Likert scale (Likert, 1932). Through these evaluations, a 10-point discrete analogue scale was favoured in opposition to 3 or 7-points scales, with the aim of capturing the slightest differences of how each participant evaluates its skills (Müller et al., 2014).

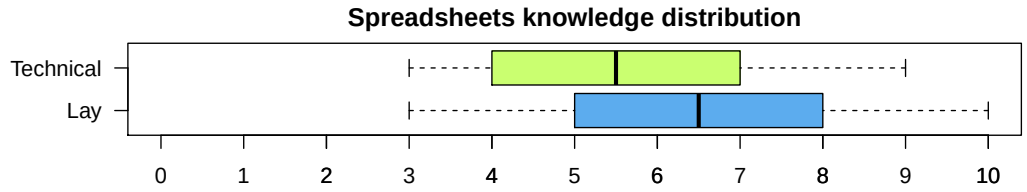


Figure 5.9: Self-assessed level of spreadsheets software usage of the respondents, using a Likert scale from 1 (very low) to 10 (very high).

The participants admitted being quite knowledgeable of spreadsheets software usage (Figure 5.9), with the lay users expressing an overall higher degree of mastery than the technical respondents. Asking about their behaviour when working with this type of software, most of them answered that they usually import data either manually or by loading a CSV file, visualising data when needed using the built-in wizards these software packages often provide. Regarding data visualization skills, some participants relied on the charting libraries implemented for statistical analysis tools such as *R*[1], or language-specific ones like *PyChart*[2] in order to produce visual outputs from datasets. Their expertise level is gathered in Figure 5.10.

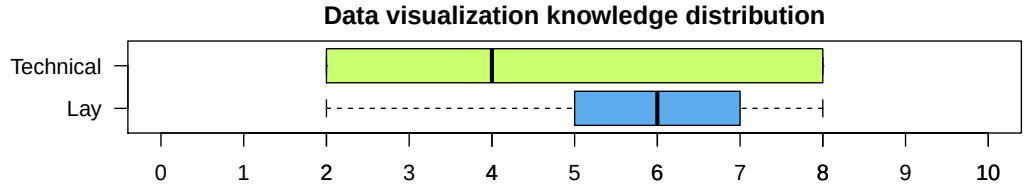


Figure 5.10: Self-assessed level of data visualization of the respondents, using a Likert scale from 1 (very low) to 10 (very high).

[1] https://r-project.org
[2] http://home.gna.org/pychart

5.3 Experiment design

To design the evaluation of *John Snow*, we searched for a solid methodology which allowed us to evaluate our proposal in a proper manner. Due to the variety of set-ups and best practices recommended in the literature, we decided to take a review leaded by Heidi Lam (Lam et al., 2011), who after analysing more than 800 publications in different visualization conferences came up with seven different scenarios to evaluate information visualization:

- Evaluating Environments and Work Practices (EWP)
- Evaluating Visual Data Analysis and Reasoning (VDAR)
- Evaluating Communication through Visualization (CTV)
- Evaluating Collaborative Data Analysis (CDA)
- Evaluating User Performance (UP)
- Evaluating User Experience (UE)
- Automated Evaluation of Visualizations (AEV)

Conforming to the features of our visualization pipeline, the scenario that best supports *John Snow* is the evaluation of Visual Data Analysis and Reasoning (VDAR), as defined by its authors: "*Involving studies that assess how a visualization tool supports visual analysis and reasoning about data, to generate information about the user's domain*". As our primary goal for this research is to bring together both the Semantic Web and the Information Visualization fields, we should set the focus on how charts and graphics can be used to improve the processes that allow to understand and interact with novel datasets that have enriched descriptions, whatever the field and skills of the analysts.

Evaluations accomplished under VDAR directives are focused on how the visualization tool backs the whole analytic process, rather than focusing on isolated stages of it. VDAR also performs well when searching for usability and design problems, perfectly fitting our iterative development environment.

Lam's methodology collects some evaluation questions, proposed in the Notional Model of Analyst Sensemaking (Pirolli and Card, 2005). The evaluation of VDAR environments should refer to how Data Exploration is supported by the tool, how Knowledge Discovery is encouraged, if it helps towards new hypotheses generation and finally, whether it will result in better decision making processes or not. Questions that should confirm if *John Snow* helps towards better knowledge acquisition in LOD environments.

Lam et al. expressed the difficulty in standardising the methods to evaluate VDAR tools due to the variety of approaches followed when analysing and reasoning over data, specially when the topics of the target datasets are greatly context-sensitive and additional variables and previous experiences come into play. Their proposed techniques to evaluate such tools are to use *Case studies* and *Controlled experiments*.

5.3.1 **Experiment set up**

To evaluate our visualization prototype we proposed a goal-directed case study, in which different datasets where presented to the volunteers with the objective to answer some pre-established questions about them, focusing on the path followed to reach a satisfactory answer.

The interviews were carried out in a similar fashion to those performed to discover how people construct visualizations (Zhao et al., 2008; Grammel et al., 2010), gathering the respondents individually in an isolated room with the observer and a laptop, with all the data and tools ready to be used. Each session lasted for about 45-60 minutes, and the audio was recorded to later analyse the participants' answers in a more relaxed environment, without the need of constantly interrupting the users to understand their strategies. To collect all the viewpoints, the respondents were encouraged to use the "*Think aloud protocol*" (Lewis and Rieman, 1993), which asks participants to say out loud anything that comes to their minds whilst completing the set of specific tasks they were required to perform. Verbally registering all the opinions, feelings and intentions provided highly valuable information about the cognitive processes followed by the volunteers.

As *John Snow* was specifically designed and developed to incorporate all the pipeline's modules described in the previous chapters, its interface was unknown to the participants. To explain its workflow, we trained each participant by simulating a simple data analysis task with a dataset already familiar to all volunteers: our research unit's dataset, containing all the information about researchers, publications, projects and so on that contains our website[1]. During this stage, all the views presented in Section 5.1 were explored.

5.3.2 Exploratory analysis tasks

For the evaluation itself, four open-licensed datasets were selected from different Open Data Portals. In order to be eligible, the datasets must be published through an accessible SPARQL endpoint, following the W3C's standards and recommendations. They also needed to provide all the information in various structured formats, including CSV and RDF. Each dataset covered a different topic, unknown beforehand by the participants, avoiding the biases that might have resulted otherwise. The selected datasets were:

Dataset 1: Restaurants

`http://zaragoza.es/ciudad/risp/detalle_Risp?id=285`

Description: The *Restaurants* dataset contains information about the different eating options in the Spanish city of Zaragoza. The resources' descriptions are greatly detailed, listing a total amount of 36 properties, such as capacity, location, cuisine types and so on. The dataset is part of Zaragoza's Open Data portal, one of the most complete and renowned portals among Spanish cities.

Dataset 2: New student enrolments

`https://datos.upo.gob.es/dataset/?id=estudiantes-nuevo-grado`

Description: Detailed information for new enrolled students in the Pablo de Olavide University (Seville, Spain) within the 2012-2013 academic year. For each enrolment, the following information is offered: name, degree, age, gender, nationality, large family status, among others. The statistics which can be extracted from this dataset would allow to analyse the university's performance through the years.

[1] `http://morelab.deusto.es`

Dataset 3: Animal services intake data

`http://catalog.data.gov/dataset/animal-services-intake-data-57f73`

Description: Published by the Animal Intake Service department from the city of Los Angeles, USA, it provides information about the recovered animals and pets in the city's shelters for the 2011-2013 period. The triples contain data about each animal's species, intake's date and status and so on.

Dataset 4: On street crime in Camden

`https://data.gov.uk/dataset/on-street-crime-in-camden`

Description: This dataset is collected from the street crimes information provided by the Police via their public API[a], focusing on Camden's area (London, UK). The data is anonymised, containing geographical and temporal information about different street crimes: shoplifting, bicycle theft, burglary and so on.

[a] `https://data.police.uk/docs`

A set of basic exploratory questions was presented to each participant with the intention to make them figure out the structure and contents of the datasets. The questions were designed to make them interact with different features of the datasets, some of them being quite obvious due to their property names and some others requiring to combine features or look for sample values. The main goal was to use most of the available features in *John Snow*, with simple techniques that could be also simulated in spreadsheets software.

The set of questions was inspired by our background organising and taking part in Open Data-based hackathon events, where we have dealt with the need to understand and analyse previously unknown datasets, usually in a short span of time. When a domain expert is missing or non-available, the analysis workflow needs to start by figuring out how the data is structured and formatted, how the different resources in the dataset are related to each other and estimate how much pre-processing is required by the look of the data. This stage is crucial to identify where to look for the answers that our analysis is intended to provide, and the less resources that are spent, the quicker the analysis itself could start.

- **Task 1** (Dataset 1 - Restaurants)
  - Restaurants are usually categorised according to their overall quality. In the dataset we are focusing on, how are restaurants ranked according to this characteristic?
  - Which is the average number of people these restaurants can take?
- **Task 2** (Dataset 2 - New student enrolments)
  - Students can belong to large families or not. Which is the distribution of students according to their family status?
  - Do more females or males enrolled for the current academic year?
  - Which was the career with the least enrolments (name and number)?
- **Task 3** (Dataset 3 - Animal services intake data)
  - Which shelter picked up more animals (name and location)?
  - Were more cats picked up dead or alive?
- **Task 4** (Dataset 4 - On street crime in Camden)
  - Which was the department/unit that made more arrests (name and number)?
  - What is the geographical location these facts were taken from?

5.4 User evaluation of John Snow

In this section we present the outcomes of the experiments with real users, analysing different aspects of their exploratory data analyses (time to complete each task, viewpoint on *John Snow*'s features and open questions about dataset exploration), discussing the extracted conclusions from each section.

As participants expressed being somehow familiar with Spreadsheets software usage and Data Visualization (consult Figure 5.9 and Figure 5.10 for more information), we proposed each participant to complete all the tasks

using both tools: our visualization prototype (*John Snow*), and a more traditional tool suited to analyse structured data (Google Spreadsheets, LibreOffice Calc and so on). For the latter, CSV dumps containing the exact same data as the RDF datasets were downloaded.

5.4.1 Task completion performance

A common performance metric when evaluating goal-oriented tasks is to measure the time required to reach a valid answer for each assignment (Scholtz, 2006). Readers could ponder about how using both tools to analyse each dataset might affect task performance, as in the second attempt to answer the questions of each task users do already know in which properties they should focus, and therefore, would spent less time to complete the assignment.

To avoid any biases that might be introduced by the selection of the tool to be used first, we let each participant select the order in which to explore the datasets. The randomisation introduced by this mechanism removed any preconceptions and preferences that users could have brought into the analysis. Table 5.3 shows how many participants decided to use each tool for each data analysis task (some records are missing due to audiovisual file corruption of 4 of the evaluation sessions).

	CSV	**John Snow**
Task 1	8	4
Task 2	6	6
Task 3	5	7
Task 4	5	7

Table 5.3: Number of participants that selected a traditional tool to explore the datasets first (CSV) versus the ones that decided to try our prototype (John Snow).

For each participant, we measured the time required to complete each task in the first attempt. The results are displayed in Figure 5.11, in which the time distributions for each task and tool are portrayed. The completion time of every task was comprised between a minute and a half and five and a half minutes (90 to 330 seconds).

Even though our tool is intended as a prototype and no attention was paid to its performance, in overall it took participants less time to answer the questions using *John Snow* than the traditional approach. The only exception was Task 3 (animal services intake dataset), whose second question (e.g., *Were more cats picked up dead or alive?*) exposed a bug in *John Snow* about property compatibility for the first two interviews, that was corrected in following iterations.

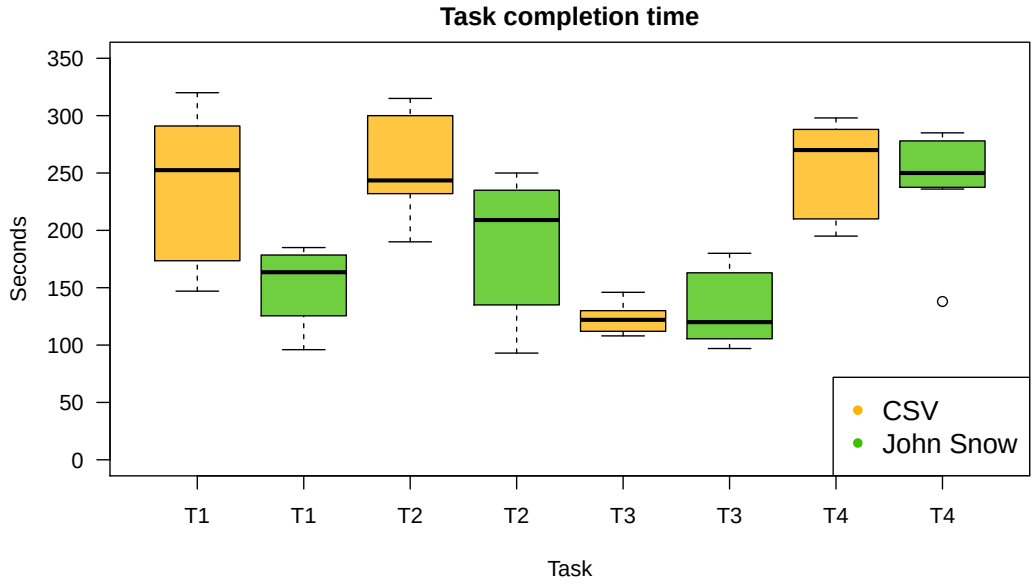


Figure 5.11: Time required to complete each of the exploratory tasks by the participants, taking only the measurements of the first attempt (tool initially used to address each assignment).

Having users analyse the datasets with both tools, allowed us to understand which data exploratory approaches were followed by the volunteers, acquiring a valuable feedback about the paths used to deal with unknown data. The time saved by *John Snow* is primarily due to its **automatic** approach, in which users did not need to apply any operations to the data or work with the encoded values at all. As our pipeline's implementation only produces the most **coherent** visualizations (meaning that only the most appropriate charts were produced after searching for compatible representations, as exposed in Section 4.3), users did not need to filter the graphics at all, pro-

ducing visualizations that would soon be discarded. This made the resulting charts less customisable, but the more intuitive experience was received as an understandable trade-off.

5.4.2 John Snow's features survey

Once all the analysis tasks were completed, we asked every participant to fill out a survey about their sensations when using *John Snow*. The questions were designed using Lam et al.'s guidelines for VDAR evaluations, and aimed to discover if the tool encouraged new knowledge discovery and how it improved exploratory analysis for the proposed assignments.

For each question, the participants' answers are displayed.

Question 1: How would you assess chart availability in order to answer the questions?

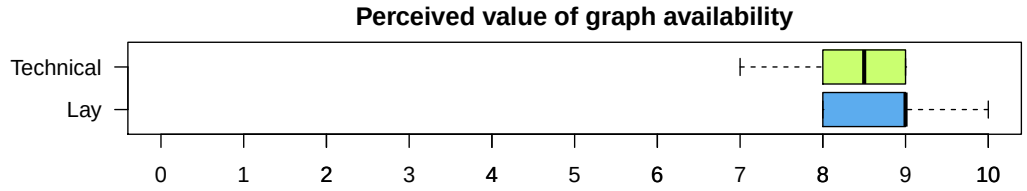


Figure 5.12: Perceived value of the graph availability in John Snow, using a Likert scale from 1 (low availability) to 10 (high availability).

At the time the evaluations were conducted, *John Snow* was able to produce the following charts: line charts, column charts, box-plots, scatter plots, pie charts, timelines, map projections and node-link diagrams. Despite the limited range of options, respondents scored the chart availability in a very positive way (Figure 5.12), as these representations allowed to quickly explore the proposed datasets. *John Snow* proved to adapt to diverse topics and structures, a primary objective of generic tools.

Question 2: How useful did you find the primitive datatype inference section for each property, which highlighted the identified datatypes for each of them?

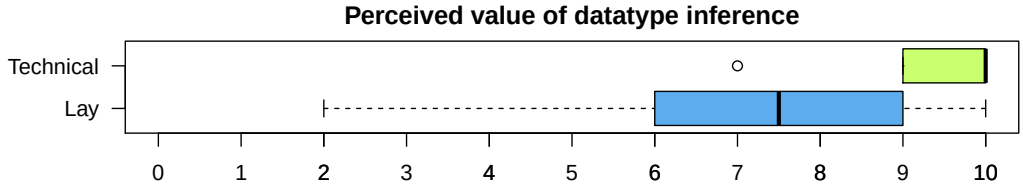


Figure 5.13: Perceived value of the datatype inference rendering in John Snow, using a Likert scale from 1 (not useful) to 10 (very useful).

This point focused on the visual rendering of the primitive datatype inference task, depicted as a series of icons (one for each of the 8 datatypes) that are highlighted when the inference process detects a matching. The feature received the less favourable feedback in the survey (Figure 5.13), as some lay users recognised not paying attention at all to these icons whilst exploring the data. On the other hand, technical users appreciated the characteristic, due to their previous experience with the lack of correct typing within LOD.

Question 3: Some metadata values were rendered visually (see Section 5.1) in *John Snow*'s interface. Did you use them during the exploratory analysis?

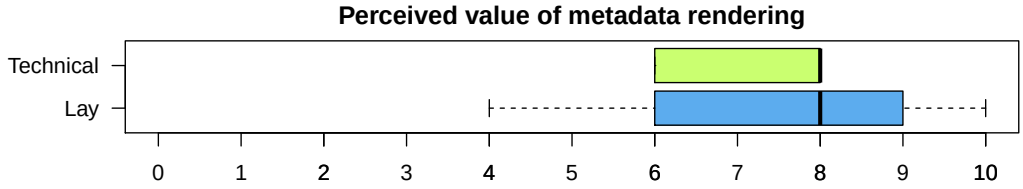


Figure 5.14: Perceived value of the metadata rendering feature in John Snow, using a Likert scale from 1 (not useful) to 10 (very useful).

The metadata values included in *John Snow*'s interface represented the *property usage* and *completeness ratio* values from Section 3.4, together with the common figures displayed by other tools in the State of the Art: number of entities, axioms, properties and so on. By providing these data in different

views, users could have an overall perspective of the whole dataset at any point, fulfilling both the *overview first* and *details-on-demand* tasks proposed by Shneiderman. This feature received disperse evaluations by lay users, in opposition to the good opinions of the technical users, who were already familiar with some of these metrics.

Question 4: For some primitive datatypes, specific extended metrics were displayed. Did these indicators help you reach some answers more quickly, or in a more intuitive way?

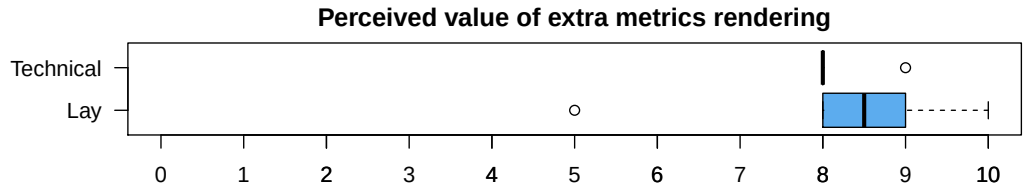


Figure 5.15: Perceived value of the extra metrics rendering in John Snow, using a Likert scale from 1 (not intuitive at all) to 10 (very intuitive).

Some visual representations were accompanied by panels with extra information, such as the summary statistics panel rendered for boxplot visualizations of numerical data. These descriptive metrics can be directly retrieved from SPARQL queries (using MIN, MAX and the AVG clauses of the language, among others), without the need to compute them on the pipeline's side. These metrics ease the interpretation of the visualization, providing figures that directly correspond to the chart being displayed. Both user profiles answered positively to the inclusion of these panels in the visualizations (Figure 5.15).

Question 5: How suitable did you find the datatype categories (i.e., integer, float, boolean, IRI, string, datetime component, geographical component and categorical) in order to classify property values?

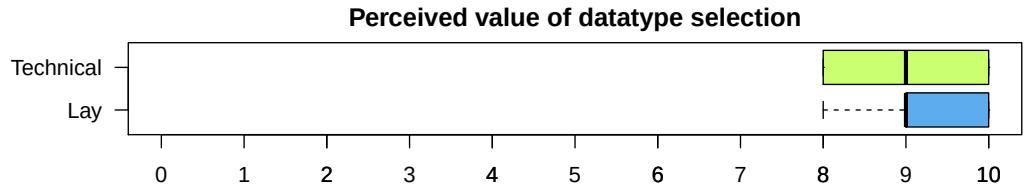


Figure 5.16: Perceived value of the primitive datatype selection in John Snow, using a Likert scale from 1 (not suitable) to 10 (very suitable).

Finally, all respondents kept a great consideration of the 8 selected primitive datatypes to categorise property values (Figure 5.16). When asked specifically about if they missed or would discard any datatype, none of the participants came up with recommendations or changes, declaring that all datatypes seemed to cover any value they could think of.

5.4.3 Thematic analysis of users' impressions

In addition to the closed questions addressed by the survey, all respondents were encouraged to express any impression about dataset exploration, writing down their comments in the blank space provided at the end of the questionnaire.

To analyse the opinions we relied on an *inductive thematic analysis* approach (Braun and Clarke, 2006), a data-driven topic extraction framework used in qualitative evaluations. This method proposes six stages to develop the thematic analysis:

1. *Familiarising yourself with your data*: Read the collected data until being comfortable with its contents. It may require a data transcription stage if the comments were recorded solely in audio or video.

2. *Generating initial codes*: Note down the initial ideas (keywords or expressions) that can be extracted from the comments.

3. *Searching for themes*: Look for patterns that are repeatedly addressed in the previous stage, selecting the topics that allow to gather them.

4. *Reviewing themes*: Produce a thematic map of the analysis, assuring that all codes from stage 2 are taken into account.

5. *Defining and naming themes*: Refine the selected topics, providing a clear definition of each one.

6. *Producing the report*: Generate the analysis' output, in order to be used as conclusions for the research question.

To provide a certain level of inter-rater reliability of the qualitative evaluation, the first 4 stages of the thematic analysis were conducted concurrently by two researchers (Armstrong et al., 1997). Each analyst extracted its own themes and codes from the 53 comments the respondents made, which were later shared carrying out the last two stages together. The produced report is summarised in Table 5.4 and detailed below.

Barriers and troubles	**Viewpoints on John Snow**	**Suggestions for improvement**
Data pre-analysis stage	Intuitive	Filters and facets
Frustration	Ease of use	Descriptive properties
Performance	Background not required	Property combinations
Data publisher's responsibility	Useful charts	More graphic options
Effort	Efficacy and Efficiency	Aesthetic changes
Dataset deficiency	Satisfaction	
	Homogeneous interface	
	Reliable	

Table 5.4: Themes extracted using the inductive thematic analysis approach on the 53 comments gathered from the respondents.

5.4.3.1 **Barriers and troubles**

This theme groups all the comments whose discourse had a negative connotation, usually addressing issues with the datasets' contents. The participants expressed their *frustration* with wrongly parsed data and missing information within the datasets, which made them require extra *effort* to conduct the analyses. They also addressed that data publishers should be *held responsible* and

to blame for the *deficiencies* in the data, as they should be in charge of their maintenance and reusability. Finally, the bad *performance* of SPARQL was also criticised, as data retrieval from endpoints took more time than expected by the volunteers.

5.4.3.2 **Viewpoints on John Snow**

About our visualization prototype, many opinions contained a positive component, appreciating the *ease of use* of the tool. Thanks to the *homogeneous interface* of John Snow, datasets from different topics could be explored in a *reliable* and *intuitive* fashion, with no *background knowledge required* to start an analysis. Visualising data through charts was perceived as a *useful* approach, and having only the most coherent visualizations produced gave an impression of both *efficacy and efficiency*.

5.4.3.3 **Suggestions for improvement**

Finally, some observations provided feedback on the tool's features, noticing a few aspects that could benefit from further development. The inclusion of *facet filtering* to explore data was a recurring theme, as many participants were deeply familiarised with this interaction technique. There are other tools in the literature that focus on this type of data browsing, and John Snow was originally conceived to have a different approach. However, those properties with categorical values do generate visualizations grouped by categories, which sometimes simulates the behaviour of facet filtering. Users also noted that properties should be listed providing a better *description* of them, not just by their names. John Snow looks for *rdfs:label*, *rdf:comment*, *dc:description* and so on values, in an effort to provide a friendly description of each property. However, these triples describing the ontology in a human-readable format are not published in the dataset's endpoint, so in few occasions we are able to provide the information. A thorough analysis of each OWL ontology might produce more satisfactory results.

The following two topics suggested a series of features to be implemented in future iterations: more *property combinations* and *visualization types*. The

heuristics to select the properties that can be combined were refined after some evaluations, which made some tasks easier to resolve. Nevertheless, the inclusion of more visualizations in next versions of the prototype should improve its adoption.

Finally, there was a participant that recommended updating the *User Interface*, reasoning that a more *aesthetic* look and feel would allow to attract the users attention.

5.5 Conclusions

The evaluation's outcomes show that *John Snow*, our LOD visualization pipeline prototype, improved how users face new semantic datasets with minimal training required. Dataset exploration was achieved quicker in overall terms using our proposal, mainly due to the time saved at the data and visualization selection stages. *John Snow* only requires the user to select the class or properties she wants to focus on, generating the summaries and visualizations that are considered more relevant for her. On the other hand, when using a more traditional approach, even if the analyst knows exactly which features to explore, she needs to select the set of data instances and map them to a visual representation of her choosing, thus incrementing the time spent in the task.

The primitive datatype inference received some of the lowest scores when exploring the data, as users did not pay much attention to the visual rendering of the inference stage when analysing the datasets. However, this is one of the key contributions of this research, and most participants recognised that they were looking for examples of values when in doubt for a property, and that having a clue in the properties' view could ease the exploratory stage.

With little technical skills required, users can start exploring any LOD dataset thanks to our visualization pipeline, navigating through classes and properties from the very first moment. Errors in the analysis are not penalised, and just browsing to a previous page allows to discover new paths.

He who fights with monsters might take care lest he thereby become a monster. And when you gaze long into an abyss the abyss also gazes into you.

"Beyond Good and Evil", Friedrich Nietzsche (1886)

CHAPTER

6

Conclusions

NOWADAYS, lots of data visualization examples can be found applied to diverse areas, presenting information using graphical means and promoting the field as a perfect option to communicate results to a broad audience in a quick and effective fashion.

In this dissertation, we have focused on applying data visualization techniques to the Semantic Web community, in an effort to promote the publication of new datasets under the Linked Open Data principles and guidelines. It has been 25 years since Sir Tim Berners-Lee invented the World Wide Web, and 15 from the moment when he drafted the path towards improving the Internet describing his vision about a Semantic Web.

Many developments have been made to realise his propositions. Still, both the Semantic Web and Linked Data communities have failed in becoming mainstream. Our LOD visualization pipeline proposal aims to bring closer the benefits of Berners-Lee's new stage for the Internet to users, moving away from its technicalities and implementation details, and focusing on the data discovery and exploration experiences.

During this closing chapter, we will revise the lessons learned through the performed research, summarise the limitations of our approach and sketch some future lines of work.

6.1 Discussion

Through the performed research, we have made some contributions in the LOD visualization field, relying on previous works and proposing some concepts in order to automatise the processes involved. Below we list the tasks and challenges we have addressed, together with the designed approaches:

- In Chapter 1 we introduced the topic, explained our motivation to focus our efforts in this field and proposed the research hypothesis, that would be validated by achieving some goals.
- Next, Chapter 2 presented a brief overview on data visualization, showing how graphics and charts have been used in the past to communicate results effectively (Section 2.1). Then, we reviewed the most promising tools and prototypes to visualise data in the Semantic Web through Section 2.2, analysing and comparing their features to detect what was missing and open to improvements.
- Chapter 3 presents our contributions towards extracting the essence of the data, with the intent to automatically figure out what the data is about within a novel dataset. Thus, we have designed a data-driven approach that allows to infer the structure of any dataset, which can take any imaginable layout as stated by the Open World Assumption. Through the analysis, our approach categorises each property's values according to a set of primitive datatypes (consult Section 3.3), allowing to understand the nature of the data that is being explored. Each previous work has categorised property values in different ways, but we deem that an essential task is to understand how each value can be used, transformed and analysed. After that, we proposed new metrics to rank resources (both classes and properties) according to metadata features in Section 3.4, providing a sortable list depending on the characteristics that are most promising for each analysis purpose. Some of these metrics are used by other prototypes as well, but only to extend the information offered for each entity, not to evaluate their relevance in any scenario.

- Chapter 4 explains how even simple visual representations can encode data in such a way that they are suitable to effectively communicate results to non-skilled users, summarising the facts in order to be understandable with no extra knowledge required. After studying each chart type's suitability (Section 4.1), we have merged this information with the analysis goal pursued by the user (Section 4.2). Taking into consideration what we want our audience to focus on, we should choose a different representation in each scenario (Section 4.3). This is common practice in speeches and public presentations, but we usually pay little attention to the efficacy of our charts. A well designed visualization can unveil interesting insights that were hidden on a big set of data, or can mislead the analysts judgement by presenting facts in a poor way.

- Finally, a prototype of the designed LOD visualization pipeline is presented in Chapter 5: *John Snow*. It implements many of the previously exposed contributions, presenting a web application that enables the interaction with any SPARQL endpoint. Its design and development was tested with some real users, in order to detect design flaws of our approach thanks to their feedback (Section 5.4).

After reviewing the contributions made to improve the actual State of the Art on LOD visualization, the following conclusions have been extracted from our research:

- The primitive datatype inference task from Section 3.3 allows to understand how each property can be interpreted, describing what operations and transformations it can perform. Our selection of primitive datatypes can always be improved by adding new categories, but the actual eight categories chosen (e.g., integer, float, boolean, IRI, string, datetime component, geographical component and categorical) are able to represent any value, are easily encoded in any programming language and are all at the same *hierarchical level*. One of the drawbacks we saw in other studies was that they selected categories belonging to different levels, for example: organizations and numbers. We have demonstrated that there is a missing gap there which our approach might solve.

- Most LOD tools provide some basic metadata metrics to characterise each dataset: number of resources, triples, classes and properties, in/out-degrees and so on. By taking those metrics and applying some basic operations, we are able to transform these *de-facto* metrics in very valuable information, allowing to rank and compare ontological entities between them. In our case, we focused on providing extra metrics for ontological properties, as they are the resources in which most data can be found, but similar ideas might be applicable for ontological classes as well. As most LOD tools already provide the input metrics, just by performing the basic operations we propose, they would be able to offer the computed metrics which were explained in Section 3.4. Together with the datatype inference task, the semantic annotations present in the data allows to perform operations that would otherwise be more costly. As stated in our initial motivation (Section 1.1), the understanding of data by machines should contribute towards providing **smart** LOD summaries and visualizations.

- In Chapter 4 we have described how to adapt the generated visualizations depending on the data's structure, the analysis intention or the data representation capacities of each chart type. All these items have an impact on how the data is finally rendered, and might alter the output to highlight specific components. By taking into account these elements, the resulting visualizations are **coherent** from the users' perspective, and have meaning within the analysis being carried out. To the best of our knowledge, no previous attempts have tried to take into consideration the user's intentions when depicting datasets. This fact greatly affects how visualizations should be constructed, since due to the diverse variety of representations available, some are better designed to highlight our interests than others.

- *John Snow*, the prototype presented in Chapter 5, serves as the implementation of the whole LOD visualization pipeline, adding all the tasks and metrics developed in the two previous chapters whilst proving to be usable for exploratory analyses by people who is not trained in data

visualization and with no prior experience with the domain. The interaction with the tool has been addressed easy enough to be helpful in dataset exploration for both lay and technical user profiles. The loss of visualization tailoring in favour of a more **automatic** approach was received as an understandable trade-off during the users evaluation stage, as they preferred to sacrifice a little of chart customisation to have less responsibility in the graphics generation.

Therefore, we conclude that our **research hypothesis** (please refer to Section 1.2) is **validated**, as we have demonstrated that by having data published together with semantic annotations, a visualization pipeline whose modules are designed to take the advantages of LOD is able to offer visual representations that provide insightful exploratory analyses. Leveraging the power of semantics the tool can be used by users with no further skills required than those of browsing web pages.

Moreover, **the main goal** of designing and implementing a pipeline able to produce visualizations taking raw LOD as input **has been achieved**, as demonstrated by the prototype presented in Chapter 5: *John Snow*.

We believe to have helped towards the democratisation of the Semantic Web, bringing its benefits closer to users through an intuitive and simple exploratory analysis. However, we have also identified some issues that might explain why both the Semantic Web and Linked Data fields are failing to catch wider adoption.

Our feeling is that the LODCloud (i.e., the collection of datasets published with Open licenses in *Datahub*) is dying at a slow pace, as each time we look for datasets to work with we confirm that dataset sustainability is a big issue. According to SPARQLES (SPARQL Endpoint Status)[1], 275 out of 553 endpoints are available as of August, 2016 (less than 50% of the whole LOD Cloud). However, when testing queries with some datasets that are listed as available, we have had issues with non-answering servers (being out of service for several days), services that are unable to produce an answer within a reasonable time (time-out errors after being unable to answer basic SPARQL

[1] http://sparqles.ai.wu.ac.at

queries after 2-3 minutes of processing) and completely out-of-date datasets, that have been forgotten by they publishers a long time ago. This situation is not encouraging for attracting new people into the research community, as it is seen as neglected.

Together with the low maintenance rates, many data publishers have not fully complied to the Linked Data agreement. Many governments promoted Open Data policies years ago, in order to improve the low transparency levels of public administrations. Data portals such as those from the United States of America[1], United Kingdom[2] or our local government's website for open data within the Basque Country region[3] became recognised worldwide as exemplary exercises of public openness. Still, little consideration was put into **how** it was published. Therefore, many datasets were published containing document scans that were hard to digitalise, used proprietary formats such as Excel spreadsheets or DBMS (DataBase Manager System) dumps or simply structured formats such as CSV and text files. This way data was easily outdated, contained lots of errors that were difficult to detect and analysts needed to make all the pre-processing tasks in order to start working with the data. There is still room for improvement and aim for the 4 and 5-star levels in Berners-Lee's LOD scale.

6.2 Limitations

Research can never be thought of as a concluded task, as one can always find innovative, better or more effective manners to achieve her goals, even those she did not thought of at first.

In order to focus our research, we defined a scenario in which our proposal could be validated, and whose problems our approach would try to solve. This scope, described in detail in Section 1.3, takes some assumptions that guided and constrained our approach.

[1] https://data.gov
[2] https://data.gov.uk
[3] http://opendata.euskadi.eus

- We only considered datasets published through a SPARQL endpoint. For some of our tests, he have downloaded semantic dumps of the data, and imported them into a local Openlink Virtuoso instance successfully. However, one of the main goals of the Internet is to be accessible by anyone, anywhere at anytime. By making datasets accessible through a public SPARQL endpoint, data can be accessed in real time ensuring it is updated, with any change being extensible for everybody. However, many RDF datasets in *Datahub* are only provided as compressed files, so developing a tool to ease the publishing process could benefit the input of LOD visualization tools.

- We have encountered several errors when querying SPARQL endpoints. Many of them had to do with time-outs and connections errors, which did not allow to generate a valid answer in a reasonable time. We did not make our prototype error-aware, so whenever a non-answering server was reached, we provided an empty list on *John Snow*'s interface. Regarding the primitive datatype inference and relevance assessment tasks described in Chapter 3, sometimes the data-retrieval processes needed to be performed several times before an answer from the servers could be obtained. That was the reason to store all the intermediate conclusions, in order to have a previously computed version available to answer when real-time querying failed.

- Our approach does not allow to merge data from two or more datasets, only allowing to inspect and navigate through resources in the same SPARQL graph. Due to the *linked* nature of LOD, this might seem to be a big issue. However, our contributions to inspect datasets at the class and property levels can be shared between datasets, and exploratory approaches are still valid. The required changes to support dataset merging should address both resource compatibility and graphic generation heuristics.

- We understand that an analyst who is an expert on the dataset's field, or any visualization *connoisseur* would produce more refined visual rep-

resentations on her own than by using our proposal. Our goal is not to produce excellent results, but good enough graphics that allow data exploration with no prior knowledge required. This approach, as those techniques which usually can be found within EDA processes, are not targeted to produce optimal results, but to speed up initial analyses.

Despite the constraints listed above, we think that our approach, together with the contributions made and the developed LOD visualization prototype have allowed us to better understand how to improve LOD visualization in an automatic fashion, taking a data-driven approach that abstracts itself from any pre-conceived ideas that users might bring to the analysis.

6.3 Future work and open issues

As stated by the English theoretical physicist Paul Dirac: "*The measure of greatness in a scientific idea is the extent to which it stimulates thought and opens up new lines of research*". We must admit that some ideas and lines of work that shown up during the research have not been accomplished, either by lack of time and resources, or by focusing on more related tasks to the study and goals at hand.

Next we suggest some ideas and open issues that might guide future researchers to satiate their knowledge hunger.

- During early stages of the research, seeing that many works in the field categorised data in high level classes such as *Organisation*, *Person* and so on, we thought about a task temporarily named as *Class-based Visualization Templates* (CVT). The main idea was to characterise each ontological class according to their most relevant features, similar to what Wikipedia depicts in their infoboxes, later used to generate the DBpedia dataset. As some datasets are deeply hierarchically organised, we thought of some kind of property specialization. That way, as *dbo:Athlete* is a child of *dbo:Person* (see Figure 6.1), it would inherit its father's most relevant information, and add their own. However, we

launched an experiment for DBpedia's ontology, identifying that leave-classes were highly specialised with many super-classes on top of them. Classes with more than one super-class that did not share properties increased the complexity level as well. Designing an approach to characterise well-known classes would become an interesting area to perform new research.

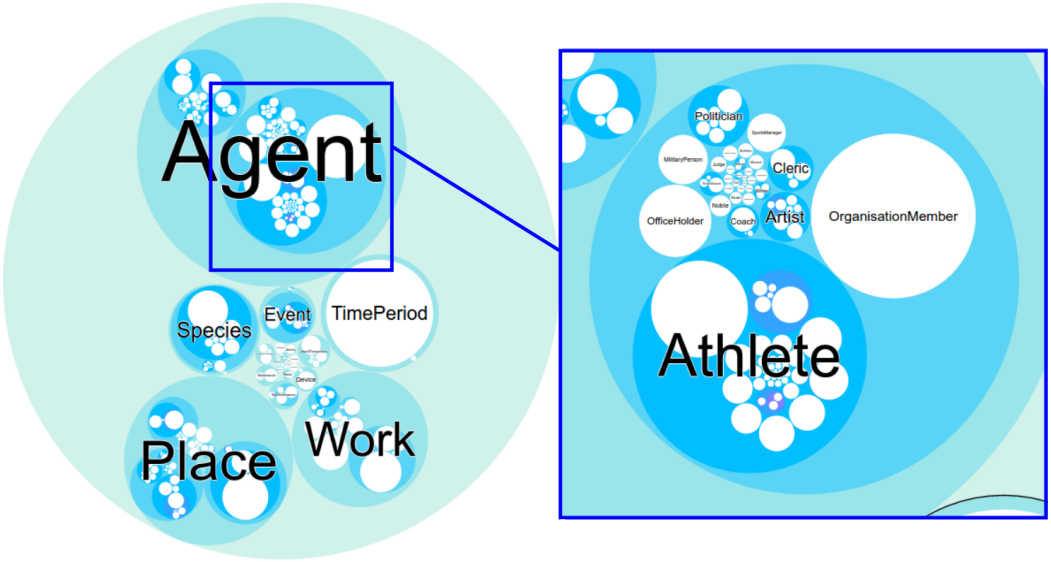


Figure 6.1: Circle pack layout visualization for DBpedia ontology, with *dbo:Person* class zoomed. Each bubble is a child class to its parent circle, with more than 500 classes involved in this graphic.

- One of the future objectives of information visualization is the design and development of collaborative visualization tools (Viegas et al., 2007; Heer et al., 2008; Elmqvist, 2014), in which a group of people works together in the representation generation. By working on the same data, each person can focus on a small portion of the whole dataset, dealing with big amounts of information in a simpler fashion. Works improving collaborative visualization could enhance multiple fields, such as data journalism, having analysts from all around the world inspecting and searching for interesting insights that are worthwhile publishing. An example of this could be the release of the *Panama papers*[1], a joint

[1] https://panamapapers.icij.org

effort of more than 300 journalism professionals who discovered off-shore entities that committed fraud.

- Finally, we consider a good contribution to publish all the information about how each dataset has been profiled, what categories each class and properties have been assigned in previous analyses and the features chosen to generate each visualization; allowing other researchers to reuse the existing information to develop their own tools without repeating these processes. The perfect match for publishing all this information in a LOD environment is to make it available as LOD also. However, agreeing on a common vocabulary that fits all aspects of our analysis seems a difficult task. Some visualization ontologies such as VISO (Voigt et al., 2012; Polowinski and Voigt, 2013) and DVIA[1] are only designed to portray one of the pipeline's components, being complex to merge in a single structure.

6.4 Final remarks

The presented work, developed through my 3 year long pre-doctoral period has aimed to provide significant contributions to a research field that I consider will further help in a broader adoption of Semantic Web and Linked Open Data practices. I wish that the extracted conclusions, together with the drafted future lines of work might inspire other researchers to develop their own ideas, contributing to the subject and improving the manner the Semantic Web community publishes and consumes Linked Open Data.

[1] http://www.eurecom.fr/~atemezin/datalift/visumodel/visu-vocab.rdf

Bibliography

Alexander, K., Cyganiak, R., Hausenblas, M., and Zhao, J. (2009). Describing linked datasets - on the design and usage of void, the 'vocabulary of interlinked datasets'. In *WWW 2009 Workshop: Linked Data on the Web (LDOW2009)*, Madrid, Spain. 62

Armstrong, D., Gosling, A., Weinman, J., and Marteau, T. (1997). The place of inter-rater reliability in qualitative research: An empirical study. *Sociology*, 31(3):597–606. 115

Assaf, A., Atemezing, G. A., Troncy, R., and Cabrio, E. (2014). *What Are the Important Properties of an Entity?*, pages 190–194. Springer International Publishing. 65, 69

Atemezing, G. A. and Troncy, R. (2014). Towards a linked-data based visualization wizard. In *Proceedings of the 5th International Conference on Consuming Linked Data-Volume 1264*, pages 1–12. CEUR-WS. org. 29, 52, 69

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). *Dbpedia: A nucleus for a web of open data*. Springer Berlin Heidelberg, Berlin, Heidelberg. 45, 51

Beek, W., Rietveld, L., Bazoobandi, H. R., Wielemaker, J., and Schlobach, S. (2014). *LOD Laundromat: A Uniform Way of Publishing Other People's Dirty Data*, pages 213–228. Springer International Publishing. 62

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, 284(5):34–43. 1

Bier, E. A., Stone, M. C., Pier, K., Buxton, W., and DeRose, T. D. (1993). Toolglass and magic lenses: the see-through interface. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 73–80. ACM. 29

Bikakis, N., Skourla, M., and Papastefanatos, G. (2014). rdf: Synopsviz–a framework for hierarchical linked data visual exploration and analysis. In *The Semantic Web: ESWC 2014 Satellite Events*, pages 292–297. Springer. 32

Bizer, C., Heath, T., and Berners-Lee, T. (2009a). Linked data-the story so far. *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, pages 205–227. 41

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., and Hellmann, S. (2009b). Dbpedia-a crystallization point for the web of data. *Web Semantics: science, services and agents on the world wide web*, 7(3):154–165. 45

Braun, V. and Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2):77–101. 114

Brunetti, J. M., Auer, S., García, R., Klímek, J., and Nečaskỳ, M. (2013). Formal linked data visualization model. In *Proceedings of International Conference on Information Integration and Web-based Applications & Services*, page 309. ACM. 30, 52

Börner, K. (2015). *Atlas of knowledge: Anyone can map*. MIT Press. 79

Carvalho, P., Hitzelberger, P., Otjacques, B., Bouali, F., and Gilles, V. (2014). Open data integration-visualization as an asset. In *DATA 2014-3rd International Conference on Data Management Technologies and Applications*, pages 41–47. 62

Chi, E. H. (2000). A taxonomy of visualization techniques using the data state reference model. In *Information Visualization, 2000. InfoVis 2000. IEEE Symposium on*, pages 69–75. IEEE. 30

Cleveland, W. S. and McGill, R. (1984). Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American statistical association*, 79(387):531–554. 72

Codd, E. F. (1970). A relational model of data for large shared data banks. *Commun. ACM*, 13(6):377–387. 43

Cyganiak, R. and Bizer, C. (2008). Pubby-a linked data frontend for sparql endpoints. 23, 28

Dadzie, A.-S. and Rowe, M. (2011). Approaches to visualising linked data: A survey. *Semantic Web*, 2(2):89–124. 23, 37

de Leon, A., Wisniewki, F., Villazón-Terrazas, B., and Corcho, O. (2012). Map4rdf - faceted browser for geospatial datasets. In *PMOD workshop*. 24

De Vocht, L., Dimou, A., Breuer, J., Van Compernolle, M., Verborgh, R., Mannens, E., Mechant, P., and Van de Walle, R. (2014). A visual exploration workflow as enabler for the exploitation of linked open data. In *Proceedings of the 3rd International Conference on Intelligent Exploration of Semantic Data-Volume 1279*, pages 30–41. CEUR-WS.org. 26

De Vocht, L., Softic, S., Mannens, E., Van de Walle, R., and Ebner, M. (2013). Resxplorer: Interactive search for relationships in research repositories. In *Semantic Web Challenge, part of the 12th International Semantic Web Conference*. Citeseer. 26

Deligiannidis, L., Kochut, K. J., and Sheth, A. P. (2007). Rdf data exploration and visualization. In *Proceedings of the ACM first workshop on CyberInfrastructure: information management in eScience*, pages 39–46. ACM. 28

Demter, J., Auer, S., Martin, M., and Lehmann, J. (2012). LODStats—An Extensible Framework for High-performance Dataset Analytics. In *Proceedings of the EKAW 2012*, Lecture Notes in Computer Science (LNCS) 7603. Springer. 62

Elmqvist, N. (2014). Visualization reloaded: Redefining the scientific agenda for visualization research. In *Proceedings of HCI Korea*, HCIK '15, pages 132–137, South Korea. Hanbit Media, Inc. 127

Eppler, M. J. and Mengis, J. (2004). The concept of information overload: A review of literature from organization science, accounting, marketing, mis, and related disciplines. *The information society*, 20(5):325–344. 41

Few, S. (2009). *Now you see it: simple visualization techniques for quantitative analysis*. Analytics Press. 79

Grammel, L., Tory, M., and Storey, M.-A. (2010). How information visualization novices construct visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):943–952. 78, 89, 105

Graves, A. (2013). Creation of visualizations based on linked data. In *Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics*, page 41. ACM. 34

Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, 43(5):907–928. 2

Gómez-Pérez, A., Fernández-López, M., and Corcho, O. (2004). *Ontological Engineering*. Advanced Information and Knowledge Processing. Springer-Verlag, London. 2

Heer, J., Bostock, M., and Ogievetsky, V. (2010). A tour through the visualization zoo. *Commun. ACM*, 53(6):59–67. 72

Heer, J., van Ham, F., Carpendale, S., Weaver, C., and Isenberg, P. (2008). *Creation and Collaboration: Engaging New Audiences for Information Visualization*, pages 92–133. Springer Berlin Heidelberg, Berlin, Heidelberg. 127

Heim, P., Lohmann, S., Tsendragchaa, D., and Ertl, T. (2011). Semlens: visual analysis of semantic data with scatter plots and semantic lenses. In *Proceedings of the 7th International Conference on Semantic Systems*, pages 175–178. ACM. 29

Hoefler, P., Granitzer, M., Sabol, V., and Lindstaedt, S. (2013). Linked data query wizard: A tabular interface for the semantic web. In *The Semantic Web: ESWC 2013 Satellite Events*, pages 173–177. Springer. 28

Hu, Y., Janowicz, K., McKenzie, G., Sengupta, K., and Hitzler, P. (2013). A linked-data-driven and semantically-enabled journal portal for scientometrics. In *The Semantic Web–ISWC 2013*, pages 114–129. Springer. 25

Hu, Y., McKenzie, G., Yang, J.-A., Gao, S., Abdalla, A., and Janowicz, K. (2014). A linked-data-driven web portal for learning analytics: Data enrichment, interactive visualization, and knowledge discovery. *LAK Workshops*. 25

Klímek, J., Helmich, J., and Nečaskỳ, M. (2013). Payola: Collaborative linked data analysis and visualization framework. In *The Semantic Web: ESWC 2013 Satellite Events*, pages 147–151. Springer. 31

Klímek, J., Helmich, J., and Necaskỳ, M. (2014). Application of the linked data visualization model on real world data from the czech lod cloud. In *LDOW*. 32

Koch, K., McLean, J., Segev, R., Freed, M. A., Berry, M. J., Balasubramanian, V., and Sterling, P. (2006). How much the eye tells the brain. *Current Biology*, 16(14):1428–1434. 5

Lam, H., Bertini, E., Isenberg, P., Plaisant, C., and Carpendale, S. (2011). Seven guiding scenarios for information visualization evaluation. Technical Report 2011-992-04, University of Calgary, Calgary. 104

Langegger, A. and Woss, W. (2009). RDFStats - An Extensible RDF Statistics Generator and Library. In *20th International Workshop on Database and Expert Systems Application, 2009. DEXA '09*, pages 79–83. 62

Lassila, O. and Swick, R. R. (1999). Resource description framework (rdf) model and syntax specification. 1

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., et al. (2015). Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195. 45

Lewis, C. and Rieman, J. (1993). Task-centered user interface design. *A Practical Introduction*. 105

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of psychology*. 103

Mackinlay, J. (1986). Automating the design of graphical presentations of relational information. *ACM Trans. Graph.*, 5(2):110–141. 73

Meusel, R., Spahiu, B., Bizer, C., and Paulheim, H. (2015). Towards automatic topical classification of lod datasets. In *Proceedings of the 24th International Conference on World Wide Web, LDOW Workshop*, pages 18–22. 69

Mosteller, F. and Tukey, J. W. (1977). Data analysis and regression: a second course in statistics. *Addison-Wesley Series in Behavioral Science: Quantitative Methods*. 86

Mutlu, B., Hoefler, P., Sabol, V., Tschinkel, G., and Granitzer, M. (2013). Automated visualization support for linked research data. *I-SEMANTICS (Posters & Demos)*, 1026:40–44. 27

Mutlu, B., Hoefler, P., Tschinkel, G., Veas, E., Sabol, V., Stegmaier, F., and Granitzer, M. (2014). Suggesting visualisations for published data. In *Information Visualization Theory and Applications (IVAPP), 2014 International Conference on*, pages 267–275. IEEE. 27

Mäkelä, E. (2014). Aether – Generating and Viewing Extended VoID Statistical Descriptions of RDF Datasets. In *The Semantic Web: ESWC 2014 Satellite Events*, number 8798 in Lecture Notes in Computer Science, pages 429–433. Springer International Publishing. DOI: 10.1007/978-3-319-11955-7_61. 62

Müller, H., Sedley, A., and Ferrall-Nunge, E. (2014). *Survey Research in HCI*, pages 229–266. Springer New York, New York, NY. 103

Parnas, D. L., Shore, J. E., and Weiss, D. (1976). Abstract types defined as classes of variables. *ACM SIGPLAN Notices*, 11(SI):149–154. 52

Peroni, S., Motta, E., and D'Aquin, M. (2008). Identifying key concepts in an ontology, through the integration of cognitive principles with statistical and topological measures. In *Proceedings of the 3rd Asian Semantic Web Conference on The Semantic Web*, ASWC '08, pages 242–256, Berlin, Heidelberg. Springer-Verlag. 69

Peña, O., Aguilera, U., and López-de Ipiña, D. (2016). Exploring lod through metadata extraction and data-driven visualizations. *Program*, 50(3):270–287. 51

Pietschmann, S., Voigt, M., Rümpel, A., and Meißner, K. (2009). *Cruise: Composition of rich user interface services*. Springer. 35

Pirolli, P. and Card, S. (2005). The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of international conference on intelligence analysis*, volume 5, pages 2–4. 105

Pirozzi, D. and Scarano, V. (2016). Support Citizens in Visualising Open Data. In *Proceedings of the 20th International Conference in Information Visualization 2016*, pages 271–276, Lisbon, Portugal. IEEE. 53

Polowinski, J. and Voigt, M. (2013). Viso: a shared, formal knowledge base as a foundation for semi-automatic infovis systems. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, pages 1791–1796. ACM. 35, 128

Prud'Hommeaux, E., Seaborne, A., et al. (2008). Sparql query language for rdf. *W3C recommendation*, 15. 1

Rhyne, T.-M., Tory, M., Munzner, T., Ward, M. O., Johnson, C., and Laidlaw, D. H. (2003). Information and scientific visualization: Separate but equal or happy together at last. In *IEEE Visualization*, volume 3, pages 611–614. Seattle. 85

Schmachtenberg, M., Bizer, C., and Paulheim, H. (2014). Adoption of the linked data best practices in different topical domains. In *The semantic web–ISWC 2014*, pages 245–260. Springer. 4

Scholtz, J. (2006). Beyond usability: Evaluation aspects of visual analytic environments. In *2006 IEEE Symposium On Visual Analytics Science And Technology*, pages 145–150. 109

Shneiderman, B. (1996). The eyes have it: a task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343. 30, 36, 52, 98

Skjæveland, M. G. (2012). Sgvizler: A javascript wrapper for easy visualization of sparql result sets. In *The Semantic Web: ESWC 2012 Satellite Events*, pages 361–365. Springer. 33

Stevens, S. S. (1946). On the theory of scales of measurement. *Science*, 103(2684):677–680. 86

Tory, M. and Moller, T. (2004). Rethinking visualization: A high-level taxonomy. In *Proceedings of the IEEE Symposium on Information Visualization*, INFOVIS '04, pages 151–158, Washington, DC, USA. IEEE Computer Society. 52

Tufte, E. R. (1986). *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, USA. 16

Tukey, J. W. (1977). *Exploratory Data Analysis*. Pearson, Reading, Mass, 1 edition edition. 49

Uschold, M. and Jasper, R. (1999). A framework for understanding and classifying ontology applications. In *Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods (KRR5), Stockholm, Sweden*. 2

van den Bosch, A., Bogers, T., and de Kunder, M. (2016). Estimating search engine index size variability: a 9-year longitudinal study. *Scientometrics*, 107(2):839–856. 14

Velleman, P. F. and Wilkinson, L. (1993). Nominal, ordinal, interval, and ratio typologies are misleading. *The American Statistician*, 47(1):65–72. 86

Viegas, F. B., Wattenberg, M., Van Ham, F., Kriss, J., and McKeon, M. (2007). Manyeyes: a site for visualization at internet scale. *IEEE transactions on visualization and computer graphics*, 13(6):1121–1128. 127

Voigt, M., Pietschmann, S., Grammel, L., and Meißner, K. (2012). Context-aware recommendation of visualization components. In *Proceedings of the 4th International Conference on Information, Process, and Knowledge Management*, pages 101–109. 128

Voigt, M., Pietschmann, S., and Meißner, K. (2013a). A semantics-based, end-user-centered information visualization process for semantic web data. In *Semantic Models for Adaptive Interactive Systems*, pages 83–107. Springer. 35

Voigt, M., Tietz, V., Piccolotto, N., and Meißner, K. (2013b). Attract me!: How could end-users identify interesting resources? In *Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics*, WIMS '13, pages 36:1–36:12, New York, NY, USA. ACM. 69

Weiskopf, D., Ma, K.-L., van Wijk, J. J., Kosara, R., and Hauser, H. (2006). Scivis, infovis-bridging the community divide. In *Proceedings of the IEEE Visualization Conference. IEEE*. Citeseer. 85

Yu, L. (2011). Follow your nose: a basic semantic web agent. In *A Developer's Guide to the Semantic Web*, pages 533–557. Springer. 51

Zembowicz, F., Opolon, D., and Miles, S. (2010). Openchart: Charting quantitative properties in lod. In *LDOW*. Citeseer. 51

Zhao, H., Plaisant, C., Shneiderman, B., and Lazar, J. (2008). Data sonification for users with visual impairment: A case study with georeferenced data. *ACM Transactions on Computer-Human Interaction*, 15(1):1–28. 105

Declaration

I, Oscar Peña del Rio, herewith declare that this dissertation is my own original work, carried out as a doctoral student at the University of Deusto. All assistance received and notions from other sources have been identified as such, acknowledging their correspondent contributions and citing them properly.

This work contains no material which has been presented in identical or similar form to any examination board, except where due acknowledgement is made in the dissertation.

This dissertation was finished writing on November 29$^{\text{th}}$, 2016.