Journal of Geographical Systems 22 (2020): 217 - 239

DOI: https://doi.org/10.1007/s10109-019-00309-y

Strategies to access web-enabled urban spatial data for socioeconomic research using R functions

Andrés Vallone, [1,2,3]

Coro Chasco, [1]✉,[2]

Phone +34 4974266
Email coro.chasco@uam.es

Beatriz Sánchez, [4]

[1] Department of Applied Economics, Universidad Autónoma de Madrid, C/ Francisco Tomás y Valiente 5, 28049 Madrid, Spain

[2] Nebrija University, C/ Sta. Cruz de Marcenado, 27, 28015 Madrid, Spain

[3] Escuela de Ciencias Empresariales, Universidad Católica del Norte, Larrondo 1281, Coquimbo, Chile

[4] Department of Economics and Business, Catholic University of Ávila, Calle de los Canteros, s/n, Ávila, Spain

Received: 8 September 2018 / Accepted: 13 August 2019

Abstract

Since the introduction of the World Wide Web in the 1990s, available information for research purposes has increased exponentially, leading to a significant proliferation of research based on web-enabled data. Nowadays the use of internet-enabled databases, obtained by either primary data online surveys or secondary official and non-official registers, is common. However, information disposal varies depending on data category and country and specifically, the collection of microdata at low geographical level for urban analysis can be a challenge. The most common difficulties when working with secondary web-enabled data can be grouped into two categories: accessibility and availability problems. Accessibility problems are present when the data publication in the servers blocks or delays the download process, which becomes a tedious reiterative task that can produce errors in the construction of big databases. Availability problems usually arise when official agencies restrict access to the information for statistical confidentiality reasons. In order to overcome some of these problems, this paper presents different strategies based on URL parsing, PDF text extraction, and web scraping. A set of functions, which are available under a GPL-2 license,

were built in an R package to specifically extract and organize databases at the municipality level (NUTS 5) in Spain for population, unemployment, vehicle fleet, and firm characteristics.

AQ1

Keywords

Web scraping
URL parsing
Spatial microdata
Spain

JEL Classification

C81
C88
R58

1. Introduction

In recent years, there has been an impressive increase in research that is facilitated through the Internet. Web-enabled platforms, systems, technologies, and tools have been introduced to assist in transmitting knowledge, skills, and services (Beretta et al. 2018; Glavas et al. 2018; Paskaleva and Cooper 2018).[1] Nowadays, it is common to use databases available via the Internet—many times under open data standards—which allow users to access and visualize a wide variety of information (William Xu and Liu 2003; Roy et al. 2010; Hansen et al. 2014). These databases are obtained by either primary data online surveys (Wright 2005; Chang et al. 2006; Howard et al. 2015; Siewert and Udani 2016) or secondary official and non-official registers. Secondary data are "ready-made" statistical variables originally collected by persons other than the researchers, for a different purpose than the ones of corresponding investigation (Atkinson and Brandolini 2001; Westling et al. 2009; Zagayevskiy and Deutsch 2016).

The Internet has also transformed the way researchers interact with secondary data, reducing the cost of collecting, updating and storing datasets from government agencies. It has also increased the availability of non-structured information in non-official web pages used for research (Hooley et al. 2011; Edelman 2012). However, information disposal varies depending on data category and country (Graham et al. 2014). On many occasions, the collection of microdata at low geographical level for urban analysis could become a challenge. In effect, certain data collected for statistical purposes by government agencies from households, individuals, and business establishments through census and surveys are never made available due to a pledge of confidentiality restrictions (National Research Council 2005). Instead, data is provided either in the form of restricted-access data files or as anonymized data products, in which geocoded information is only available at a regional—aggregated—spatial level.

The most common difficulties when working with secondary Internet-enabled data can be grouped into two categories: accessibility and availability problems. Accessibility problems are present when the way that data is published on servers blocks or delays the download process. Then, data collection becomes a tedious and reiterative task that can produce errors in the construction process of big databases. Availability problems usually arise when official agencies restrict access to the information for statistical confidentiality reasons or when data is simply non-existent.

Two elements can resolve these problems. First, the increasing use of open-source software, like Python or R, that facilitates data collection, manipulation, and publication processes into a single software environment (Munzert et al. 2015). Second, the increasing use of API technologies and new data collection techniques like web scraping (Mehlführer 2009; Penman et al. 2009; Glez-Peña et al. 2014; Nolan and Temple Lang 2014; Salamone et al. 2014; Munzert et al. 2015; Braaksma and Zeelenberg 2015).

In order to overcome these challenges, this paper presents different strategies based on URL parsing, PDF text extraction, and web scraping. These approaches have been used to extract and organize several databases on population, unemployment, vehicle fleet and firm characteristics in Spain at the municipality level (LAU),[2] for which accessibility to information is limited and problematic. Each strategy consists of a set of functions built in the R package, "DataSpa",[3] which is available under a GPL-2 license (Vallone et al. 2017). They allow the collection of higher-quality information by avoiding potential human errors due to different impediments and restrictions imposed by official and non-official web portals to microdata extraction. This package, which constitutes the main contribution of this paper, was built to elaborate the 2017 Socioeconomic Atlas of Extremadura, which serves as the most important official database of municipality variables in this region. "DataSpa" is very useful for not only the researchers interested in the

analysis of urban systems in Spain but also, by a convenient adjustment of the package functions, any other analysts who face similar problems in other countries.

Our contribution is in line with many packages developed in R and Python to extract data published on the Internet, particularly with those built to extract information from official web portals. This is the case of, for example, "tidycensus" (Walker et al. 2019) and "cenpy" (Wolf 2019), for the US decennial censuses, "Web-Scrapping-and-EDA", for Indian demographics (Katre 2019), "tcmbapessoal", for public local accounts in Brazil (Santiago 2019) and "volnormativo", for the Uruguay's Presidency website (Xavier 2019). Other packages aim to fetch data from sub-national official bodies like "MTA-extraction", for the New York MTA subway turnstile data files (Lagacé 2019) and "sp-subway-scraper", for the official subway company page of São Paulo (Navarro 2019). Finally, many others allow scraping other type of data like "sentiment-analysis-goodreads-reviews" for sentiment analysis of book ratings in the Goodreads website (Sellers 2019) or "ratingpy", to get daily TV ratings in Turkey (Deniz 2019). This paper is also connected with contributions made in the development of spatial software tools in the social sciences (Rey and Anselin 2006; LeSage 2015), particularly for the extraction of big geo data, like social media (Fernández et al. 2018; Chen et al. 2019; Papapesios et al. 2019) or the treatment of ontologies and semantic integration (Chaabane and Jaziri 2018).

This paper is organized as follows: after the Introduction, Sect. 2 presents URL parsing as a suitable strategy to download and encode population and unemployment databases, for which a sophisticated publication platform creates serious accessibility problems. In Sect. 3, we illustrate the performance of the PDF extraction strategy with the case of the vehicle fleet database, in which the absence of an API and some blocking systems lead us to download the PDF files with the municipality reports to extract the available information. Section 4 presents the use of web scraping to download a database of firms published by a private company, which helps to solve the absence of this information in Spain at the urban and individual level. In Sects. 4 and 5, we present the conclusions and references, respectively.

2. URL parsing for databases with accessibility problems

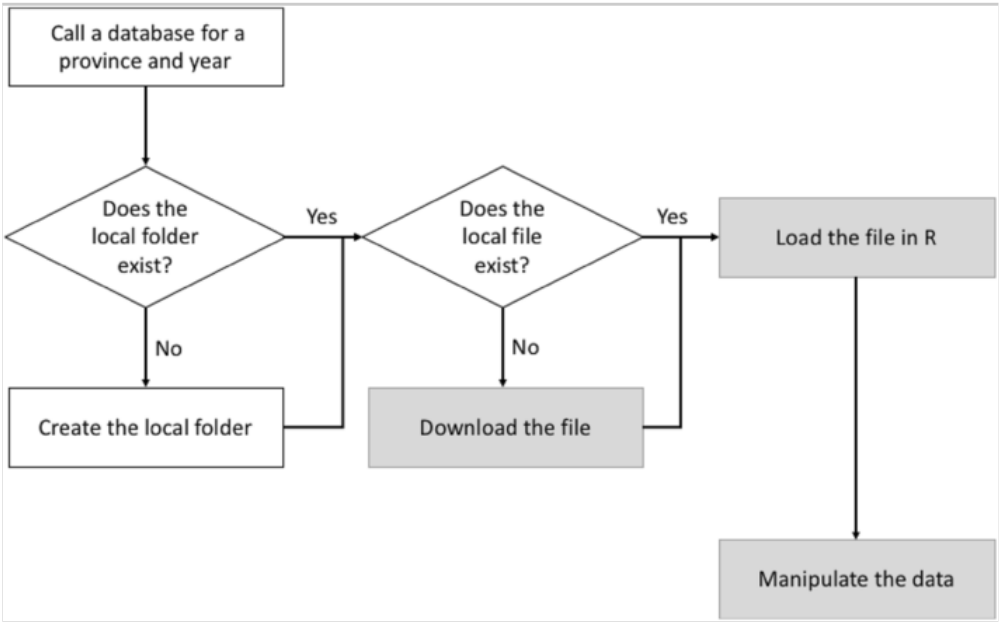
In many cases, information is available on statistical websites, but the sophistication of their organizational structure, the absence of adequate APIs, and/or the lack of codification of the data makes the downloading process tiresome and very exposed to human failures. This is the case of two main variables for socioeconomic analysis like population or workforce in Spain, which are provided by two official agencies via their web portals, the National Statistics Office (INE)[4] and the Spanish Public Employment Service (SEPE),[5] respectively, but with certain access barriers. By way of illustration, downloading data population for the municipalities of only one of the 52 Spanish provinces requires clicking at least five different URL links to reach the final data in a webpage. The construction of a whole panel data on population, which is available on the INE website since 1996 for more than eight thousand municipalities and different categories (population by sex, age group and nationality), is a long-term task. In the case of unemployment, the SEPE web server contains municipality data on registered working contracts and the unemployed population. The downloading conditions for these variables are even worse than in the previous case: to the difficulty of clicking at least five times to reach the final database for each province, we have to add that municipality data is only provided as a monthly series and, before 2012, without the official municipality identification code. Hence, the operating time for data extraction is even higher in the SEPE web portal, which presents the extra obstacle of having to code the municipalities for cross-matching tasks. It must be said that either INE or SEPE admits custom demands of large volumes of data, but they are not always satisfied or free, depending on the requesting institution, and they usually involve a certain amount of discouraging paperwork.

In order to overcome these problems, we created a code to download and manipulate the population and workforce municipality data, which are freely available in the respective official agencies but rather limited, as shown previously. The construction strategy is based on the "Uniform Resource Locator" (URL) parsing method, which consists of taking a URL in order to break it out in its standard components: scheme, domain, port, path, query string and fragment identifier to extract information about the abstract or physical resource associated with it (Berners-Lee and Masinter 2015). This computational science method has been used to avoid financial phishing attacks (Liu and Zhang 2012; Zuhair et al. 2016), evaluate security problems (Bhargavan et al. 2013), analyze the user's cost-benefits of ignoring the security adviser (Herley 2009), and retrieve images using Web mining (Chen et al. 2001), among others. In a similar process than the used by this later paper, we apply text analysis over the URL path. The path of a URL is made of text segments that represent a structured hierarchy, similar to a directory structure, where each segment is separated by the "/" character. The common path segments can be detected from the analysis of the text to identify different category segments corresponding to characteristics of the database (sex, year, province, etc.). With this information, it is possible to reproduce the specific URL of the file containing a desired dataset, which will be available to subsequent downloading and manipulation.

Figure 1 presents the general workflow of this URL parsing functionality, designed for the databases on population and unemployment in Spain. Based on the province and year of the user's interest, a set of R functions are implemented. At the beginning of the execution, the code checks the existence of a folder to host the download file. If this element does not exist, a folder named "data_poblacion" or "data_paro" is created depending on the required data. Then, it looks for the existence of the file in the local host. If this file does not exist there, it will be downloaded from the server and hosted with a determined name, depending on the function. The creation of a local data store facilitates its accessibility because it reduces Internet and URL dependencies. In effect, a local data store allows using the database without an Internet connection and prevents from changes in URLs. Additionally, it improves the code performance in the last part of the process, reducing the user's elapsed time to import the data into R.

Fig. 1

Workflow of the URL parsing functions to download databases with accessibility problems. *Source*: self-elaboration



We built a set of nested functions to increase the code flexibility. Each gray box in Fig. 1 corresponds to one of the following function subsets: download functions, loading functions and data manipulation functions.

2.1. URL parsing download functions

The main purpose of the download functions is connecting to the agency server to download a requested file. These functions have a simple two-step process. First, depending on some given arguments, they check the existence of a local folder, which will be created—when not available—to store the file. Second, they create the URL address connected to the file and download the file. Next, we present a brief description of the download functions used to extract the municipality data on population and workforce from the INE and SEPE web portals:

1. *getbase.pob(year,"provincia",extr = FALSE,anual = FALSE)*: This function downloads a file with the municipality population data by sex and five-year age for a Spanish province for a specified year. It calls the file "pob_q_year_provincia.xls" and saves it in the folder "data_poblacion". This function has four arguments. "*year*" is a numerical value for the reference year of the dataset. "*provincia*" corresponds to each of the 52 Spanish provinces. "*extr*" ("*is foreign population?"*) is a logical variable with FALSE as the default value, for which "extr = TRUE" downloads and saves the foreign population dataset by sex and major group-year age. "*anual*" ('is data required by age?') is a logical variable with FALSE as the default value, for which "anual = TRUE" downloads and saves the population by sex and one-year age. Since there is no data for foreign population by one or five-year age, the combination of extr = TRUE and anual = TRUE will generate an error message ("No data for these cases"). For example, the command *getbase.pob(2016,"Badajoz")* downloads municipality population data of the province of Badajoz corresponding to the year 2016 in the file "pob_q_2016_BADAJOZ.xlsx".

2. *getbase.fen(year,"provincia")*: This function downloads a file with other municipality demographic data (live births, fetal deaths, marriages, etc.) for a Spanish province for a specified year. It calls the file "fen_year_provincia.xls" and saves it in the folder "data_poblacion". This function has two arguments: "year", which is a numerical value for the reference year of the dataset and "provincia", which corresponds to each of the

52 Spanish provinces. For example, the command *getbase.fen(2016,"Badajoz")* downloads municipality demographic data of the Badajoz corresponding to the year 2016 in the file "fen_2016_BADAJOZ.xlsx".

3. *getbase.paro(year,"mes","provincia")*: This function downloads a file with the municipality unemployment data by sex of a Spanish province for a specified a period of time. It calls the file "paro_MUNI_provincia_mmyy.xls" and saves it in the folder "data_paro". The function has three arguments: "year", which is a numerical value for the reference year of the dataset, "mes", which is the value for the reference month of the dataset and "provincia", which corresponds to each of the 52 Spanish provinces. For example, the command *getbase.paro(2016,"julio","Badajoz")* downloads municipality unemployment data of Badajoz corresponding to the month of July of the year 2016 in the file "paro_MUNI_BADAJOZ_0716.xls". As an example of URL parsing download function, we illustrate this routine in Algorithm 1.[6]

Algorithm 1. URL parsing download function *getbase.paro()*

```
# Example of download function used to extract the Spanish municipality data on workforce
# Output: The Ms. Excel file "paro_MUNI_provincia_mmyy.xls", which is saved in the folder "data_paro".
getbase.paro<-function(year,mes,provincia){
        year<-as.character(year)
        if(dir.exists(file.path(getwd(),"data_paro"))==FALSE){
                dir.create(file.path(getwd(),"data_paro"))
                }
        provincia<-toupper(provincia)
        provincia<-a.letter(provincia)
        mes<-tolower(mes)
        nn.mes<-seq(1,12,1)
        names(nn.mes)<-
c("enero","febrero","marzo","abril","mayo","junio","julio","agosto","septiembre","octubre","noviembre","diciembre")
        cod<-paste("0",nn.mes[mes],substr(year,3,4),sep="")
        name<-paste(paste("MUNI",provincia,cod,sep="_"),".xls",sep="")
        url<-
paste("http://www.sepe.es/contenidos/que_es_el_sepe/estadisticas/datos_estadisticos/municipios/",year,"/",paste(mes,year,sep="_"),"/",na
me,sep="")
        dir<-paste(getwd(),"/data_paro/",sep="")
        file<-paste(dir,"paro_",name,sep="")
        download.file(url,file, mode='wb')
}
```

2.2. URL parsing loading functions

The mean purpose of the loading functions is importing into R the already downloaded and stored databases. These functions have a simple two-step process. First, depending on the given arguments, they check the existence of a required file in the local folder. Second, if the file does not exit, they call the corresponding download function to create it, using the "xlsx" R package to import the file. These functions are the following:

1. *paro(year,"mes","provincia")*: This function has the same arguments of the already shown "*getbase.paro()*" function. The output of this function is a data frame containing the following variables: official municipality identification code ("cod"), municipality name ("nombre"), number of unemployed people in the municipality ("paro total"), number of unemployed males ("hombres") and number of unemployed females ("mujeres"). For example, the command *paro(2016,"julio","Badajoz")* generates a data frame containing the unemployment database corresponding to the municipalities of Badajoz in July 2016.

2. A set of nine functions, which share the same arguments, but produce different outputs. Each function has two arguments: "year", which is a numerical value indicating the year of the requested data and "provincia", which is one of the 52 Spanish provinces. While the functions have a different output, there are two common variables listed by default: the official municipality identification code and the municipality name. Next, we present a brief description of these outputs by function. Functions *pob.a()* and *pob.q()* produce three data frames, all of them containing total population by sex and age (one-year and five-year age groups, respectively). *pob.e()* creates a list of three data frames, all of them containing total population and population by age (major groups) and nationality (nationals and foreigners), for both sexes, males and females. *pob.tot()*, *pob.h.tot()* and *pob.m.tot()* each generate a data frame containing municipality data for total, male and female population, respectively. *pob.n.tot()* and *pob.e.tot()* each generate a data frame containing municipality data for total national and foreign populations, respectively. *pob.fen()* generates a data frame containing municipality data for the number of births and deaths. As an example of URL parsing loading function, we illustrate this last routine in Algorithm 2.

Algorithm 2. URL parsing loading function *pob.fen()*

```
# Example of loading function used to generate a data frame of Spanish municipality data on births and deaths
# Input: a URL parsing download function getbase.fen(year,provincia) of the "DataSpa" package.
# Output: an R data frame.
pob.fen<-function(year,provincia){
                year<-as.character(year)
                provincia<-toupper(provincia)
                prov<-provincia
                provincia<-a.letter(provincia)
                dirc<-paste(getwd(),"/data_poblacion/",sep="")
                file<-paste(paste("fen",year,provincia,sep="_"),".xlsx",sep="")
                if(sum(dir(dirc)==file)==0){
                        getbase.fen(year,provincia)
                }
                abre<-paste(dirc,file,sep="")
                datos<-xlsx::read.xlsx(abre,1,colIndex=c(1,2,5))
                datos<-datos[which(complete.cases(datos)==TRUE),]
                datos<-datos[-1,]
                d<-dim(datos)
                nombres<-as.character(datos[,1])
                codigo<-rep("AA",d[1])
                municipio<-rep("AA",d[1])
                for (i in 1:d[1]){
                        nn<-unlist(strsplit(nombres[i]," "))
                        codigo[i]<-str_trim(nn[1])
                        if(length(nn)>2){
                                nom<-paste0(nn[2:length(nn)],collapse=" ")
                                municipio[i]<-nom
                        } else {
                        municipio[i]<-str_trim(nn[2])
                        }
                }
                cifras<-as.data.frame(datos[,2:3])
                cifras<-apply(cifras,2,as.numeric.factor)
                ids<-as.data.frame(cbind(codigo,municipio))
                salida<-cbind(ids,cifras)
                colnames(salida)<-c("Cod","Municipio","Nacidos","Fallecidos")
                salida
}
```

2.3. URL parsing manipulation functions

These functions have the purpose of manipulating data to build space–time panels of municipality variables for different periods or compute demographic indicators. These new variables will be stored as R data frames and/or Ms. Excel output files. All functions employ either a download or a loading function. Next, we present the manipulation functions included in the "DataSpa" package.

1. A set of six "*.ev()*" functions for the construction of panels of population variables at the municipality level for a time period.[7] All these functions share the same three arguments: "inicio", which is the starting year of the panel, "fin", which is the last year of the panel and "provincia", which is one of the 52 Spanish provinces. Although these functions also have a different output, there are four common variables listed by default: the official municipality identification code, the municipality name, and the columns corresponding to the initial and final years of the population panel. *pob.ev()*, *pob.h.ev()* and *pob.m.ev()*, *pob.n.ev()*, *pob.e.ev()* each generate a data frame containing a panel of municipality variables for total population, males, females, nationals and foreigners, respectively, for a given time period. *pob.fen.ev()* produces two data frames, all of them containing the output elements (municipality codes, names and panel variables), for births and deaths. For example, pob.ev(2000,2016, "Badajoz") generates a data frame containing the total population corresponding to the municipalities of Badajoz for the period 2000–2016. As an example URL parsing manipulating function, we illustrate the *pob.ev()* routine in Algorithm 3.

2. *pob.ind(year,"provincia",print = FALSE)* computes a data frame with a set of demographic indexes at the municipality level. It has the following arguments: "year", which is a numerical value indicating the year of the requested data, "provincia", which is one of the 52 Spanish provinces and "print", which is a logical variable with FALSE as the default value, for which "print = TRUE" saves the dataset as a Ms. Excel file. The output of this function is a data frame containing a set of ten demographic indexes: childhood, youth, third age, dependence, unemployment rates (both sexes, males and females) and municipality average age (both sexes, males, females). There is also a similar function *pob.ind.p()*, which computes finer indices using one-year age groups (instead of five) from 2011.

3. *ind.ev(inicio, fin,"provincia",print = FALSE)*. It has four arguments: "inicio", which is the starting year of the panel, "fin", which is the last year of the panel, "provincia", which is one of the 52 Spanish provinces and "print", which is a logical variable, with FALSE as the default value, for which "print = TRUE" saves the dataset as a Ms. Excel file called "pob_ev_index_provincia_inicio-fin.xlsx". The output creates ten data frames, all of them

containing the municipality code and name and the requested time series, for each of the demographic indices obtained with the *pob.ind()* function.

Algorithm 3. URL parsing manipulation function pob.ev()

```
# Example of manipulation function used to generate a data frame of Spanish municipality data on population for a given time period
# Input: a URL parsing download function  getbase.fen(year,provincia) of the "DataSpa" package.
# Output: an R data frame.
pob.ev<-function(inicio,fin,provincia,print=FALSE){
        if(fin<inicio) stop("La fecha de inicio debe ser mayor que la fecha de fin")
        n<-seq(inicio,fin,1)
        year<-as.character(sort(n,decreasing=TRUE))
        base<-pob.tot(year[1],provincia)
        for (i in 2:length(year)){
                aux<-rep(NA,dim(base)[1])
                pob<-pob.tot(year[i],provincia)
                v<-intersect(base[,1],pob[,1])
                for(j in 1:length(v)){
                        aux[which(base[,1]==v[j])]<-pob[which(pob[,1]==v[j]),3]
                }
                base<-cbind(base,aux)
                }
        colnames(base)<-c("Cod","Municipio",year)
        orden<-c(1,2,seq(dim(base)[2],3))
        base<-base[,orden]
                if (print==TRUE){
                        if(dir.exists(file.path(getwd(),"Outputs"))==FALSE){
                        dir.create(file.path(getwd(),"Outputs"))
                        }
                file<-paste(getwd(),"/Outputs/pob_total_ev_",provincia,"_",paste(inicio,fin,sep="-"),".xlsx",sep="")
                xlsx::write.xlsx(base,file)
                }
        base
}
```

3. PDF text extracting for databases with accessibility problems

The Portable Document Format (PDF) is a widely used digital document file format. It is designed to allow users to view, print, and exchange electronic documents, preserving their look across platforms with different operating systems and hardware environments (Marinai 2009). Though there are many tools for generating PDF files from text documents, there is no standard tool for converting PDF files into texts with 100% accuracy (Thaiprayoon and Haruechaiyasak 2016). That is, the PDF format does not easily allow extracting the file information (text, tables, images, databases etc.) in a straightforward way because it is hard to handle (Castillo-Fernández 2015). A PDF file describes the appearance of a page but does not mark up the logical content. Any transformation of the content of a PDF file into text format will imply a reconstruction of words and sentences from the raw positions of the letters included in the PDF.

Statistical web portals and digital platforms usually offer their databases in different formats (HTML, spreadsheets, etc.) in order to reduce data accessibility problems, such as the absence of an API, sever instabilities, or limitations in the downloadable records. The problem arises when data is only accessible from non-editable formats, like PDF files. This is—partially—the case of the municipality database on vehicle fleet in Spain. Vehicle fleet is a relevant variable in urban studies, which is used to study, for example, residential location (Eluru et al. 2010), urban air pollution (Mage et al. 1996; Kahn and Schwartz 2008; Wang et al. 2009), and effect on urban structure and commutation choice (Bento et al. 2005). The Spanish National Department of Traffic (DGT)[8] collects and distributes information about vehicle fleet at the municipality level, but before 2014, there was no API to access this server. One single download of municipality information always exceeds the maximum allowed data volume. In addition, a CAPTCHA field must be filled to avoid robot access and adds more time to the data collection process. Hence, the only way of downloading the whole vehicle fleet database at once is extracting the information provided by the DGT in PDF format.

Despite the aforementioned difficulties, there are a set of tools which transform and extract information from a PDF file into readable format (Hadjar et al. 2004). Generally, a PDF data extraction process involves at least two steps: first, it transforms the PDF to a readable file and second, it extracts the demanded information. These tools have been used by administrative services to automatically extract documents in digital libraries (Marinai 2009) or metadata from scientific articles (Aumueller 2009; Beel et al. 2013). They have also been used in more sophisticated contexts to generate text input for text mining software in situ in the Mouse Genome Informatics (MGI) system (Dowell et al. 2009), or to extract and classify vectorized diagrams (Futrelle et al. 2003).

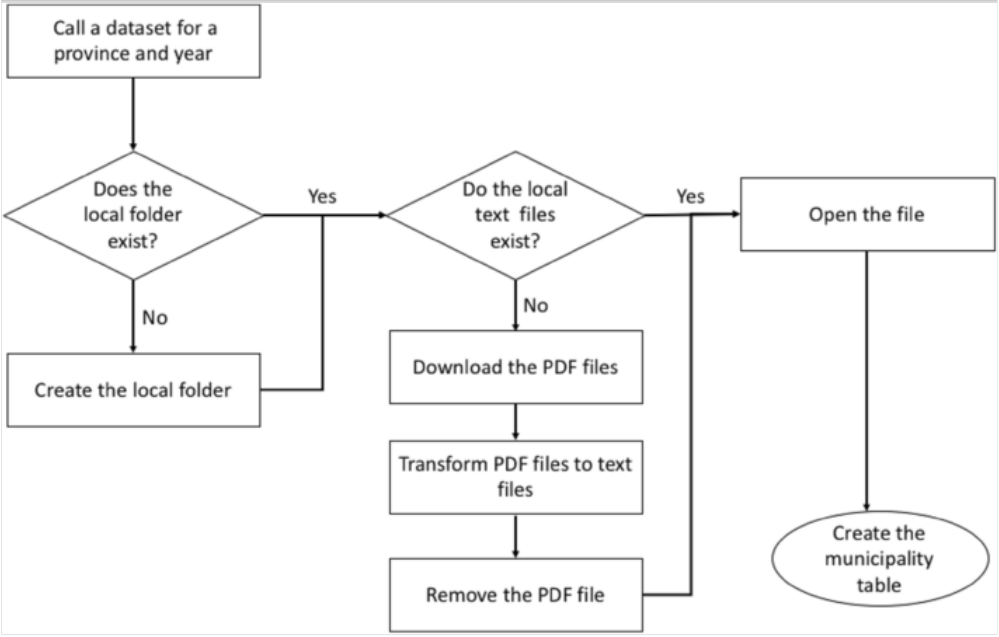
For the municipality database on vehicles, we created an R function which downloads the municipality report in PDF format to extract all the available information on it. We combine URL parsing and PDF extraction methods to create the

function. URL parsing is useful to detect and construct the URL that leads to the PDF information. To this aim, we use the "rvest" R package (Wickham 2016). We construct a web crawler to extract a list of URLs with the download addresses corresponding to each municipality PDF file. The DGT web portal offers its databases in a treelike structure: each NUTS2 region ("comunidad autónoma") has as many URLs as NUTS 3 provinces, annual periods and NUTS5 municipalities.[9] Second, a PDF extraction method is used to extract the information from the PDF file. In this case, we use the free software "PDF2TXT"[10] to transform the PDF file into an editable text format to reconstruct the database using text mining in R.

Figure 2 summarizes the function *parque.aut()* workflow. To reduce the Internet dependencies and the elapsed time, the function creates a local folder to host the files. The first step of the process consists in checking the existence of the local folder: if it does not exist, the function creates it. Each province and year has its own folder into the local host and the files are stored as "year_name.pdf", where "year" is the requested annual period and "name" is the municipality name. In a second step, this function looks for the existence—in the local host—of the municipality reports in text format: if they do not exist, the function creates the list of URLs to the PDF reports to download the files into a specific local folder. These PDF files are transformed into plain text and they are stored in text files. Since the PDF files are no longer necessary, they are deleted. When the text files are available, the function imports the files into R and creates a data frame with the municipality information for the requested province.

Fig. 2

The function *parque.aut()* workflow. *Source*: self-elaboration



The function *parque.aut(year,"*ca*","provincia")* has three arguments: "year", which is a numerical variable indicating the year of the requested data, "ca", which is one of the 17 NUTS2 regions ("autonomous communities") and "provincia", which is one of the 52 Spanish provinces. The output of this function is a data frame containing the following variables: the municipality name, number, and average age of vehicle fleet; type of vehicles (cars, vans, trucks, motorcycles, buses, etc.) and some other variables related to the register of drivers; accidents; and vehicle taxes.

4. Web scraping for databases with availability problems

Availability problems arise when certain information is not available at all or it is only obtainable partially, which is quite common in micro-territorial databases due to confidentiality, privacy, or any other reasons. In these occasions, the use of alternative information sources—when existing—is a convenient alternative. In Spain, this is the case of firm data, which is not publicly available at the micro-level. Firm data is a key information in urban studies (location decision, agglomeration economies, entrepreneurship), particularly at firm level or lower spatial aggregations, like census tracts, districts or municipalities (Arauzo Carod 2005; Arauzo-Carod and Viladecans-Marsal 2009)(Jofre-Monseny et al. 2011).

The INE's Central Company Directory (DIRCE)[11] provides annual information on the number and distribution of companies and establishments, but only at meso (regional NUTS 2 and 3) and macro-level (national NUTS 1), in order to preserve statistical confidentiality. The Industrial Establishment Register (REII),[12] published by the Ministry of

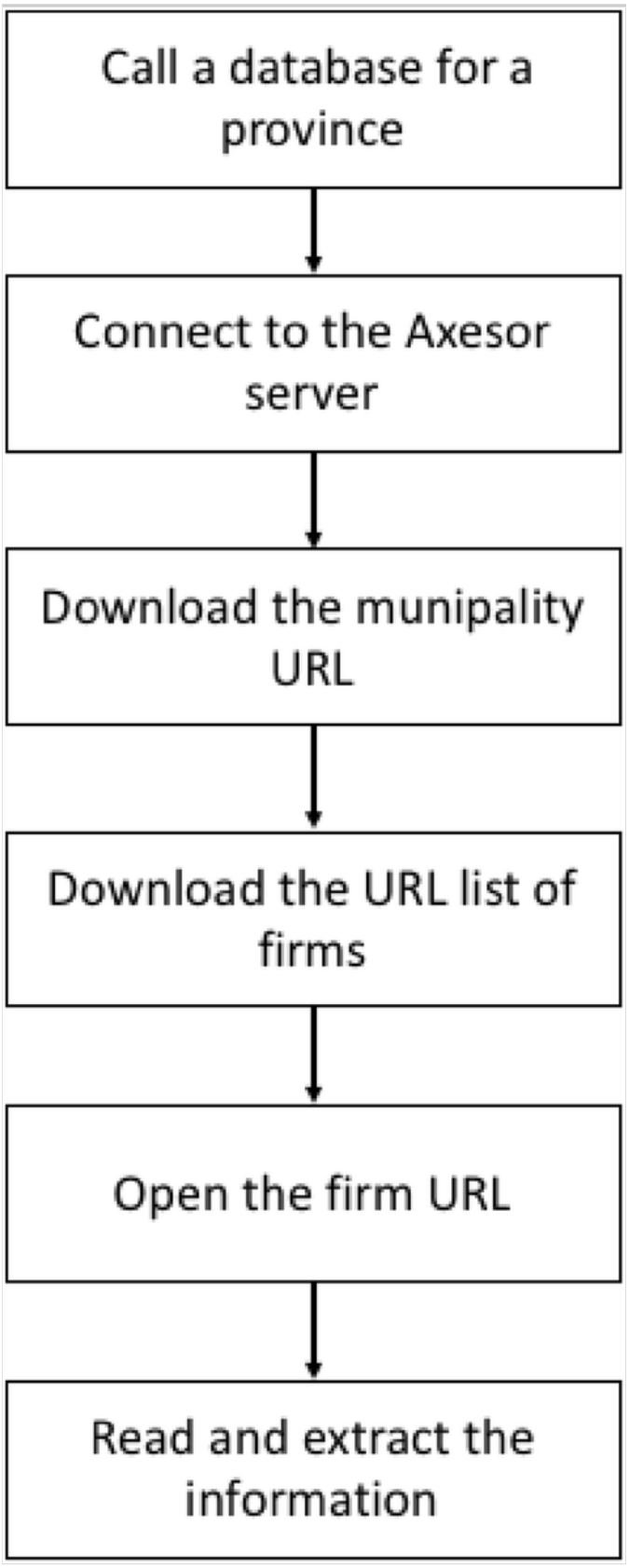
Energy, Tourism and Digital Agenda, contains a census of industrial establishments located in Spain at the municipality level (NUTS 5) and above, but it is not complete for some regions and industrial sectors. It only contains companies operating in the industrial sector and it also has some downloading restrictions. Hence, the only way of obtaining firm data for all the economic sectors at the individual level or, at least, aggregations for census tracts, districts or municipalities, is buying them from specialized companies, such as Camerdata,[13] the Iberian Balance Sheets Analysis System (SABI),[14] the Global Entrepreneurship Monitor (GEM),[15] or Axesor.[16] All in all, it is not always possible to download the complete database once and for all. For example, SABI has a restricted access to only a set of 50.000 weekly records of Spanish firms.

One of these private consulting firms, Axesor, offers part of its huge database on firms and freelances freely online. Hence by web mining this information, it is possible to create a database for Spain at the firm level and above. Web scraping is a software technique which extracts information from websites, usually simulating human exploration of the World Wide Web (Kumar 2015). Human behavior can be simulated by a web crawler, which is a bot that systematically browses the World Wide Web. It starts with a seed list of URLs to visit in order to identify all the hyperlinks in these pages to add them to a new list of URLs called the crawl frontier. Then, the URLs of the frontier are recursively visited according to a set of policies (Kumar 2015). Web scraping is focused to transform unstructured or semi-structured data on the web, typically in HTML format, into structured information (Mehlführer 2009; Kumar 2015). Typically, this extraction is made by a text mining process, searching for key words and extracting the information associated with them. This technique has multiple uses in economic research (Edelman 2012); for example, to compute Consumer Price indexes (Griffioen et al. 2014; Nygaard 2015; Polidoro et al. 2015) and enterprise innovation activities (Gök et al. 2015). The Axesor database is grouped by province, having each of them its own website with a municipality list. Every municipality, in turn, has a website with a list of firms, which likewise leads to another website containing extensive information about location, corporate information, and several business and financial indicators for different annual periods. In January 2017, there were about 3,500,000 firms in Spain according to Axesor.

Figure 3 shows the workflow of the function *data.firm* for firms (and *data.firm.a* for the freelancers). First, for a given province, the function connects to the Axesor server, downloads the municipality URL links, and enters in each of them to download their corresponding firm URLs. Second, every firm URL is opened so the functions can read and extract the data and construct a data frame. We have followed a design strategy based on two R packages. First, after exploring the Axesor web page HTML code, using the “rvest” package (Wickham 2016) we built a web crawler to obtain the firm URLs. Second, once the firm URLs were accessible, the “stringr” package (Wickham 2017) builds a function to analyze and extract the text information from the web page.

Fig. 3

The function *data.firm()* workflow. *Source*: self-elaboration



The use of the *data.firm()* and *data.firm.a()* functions is simple because they only depend on one argument: *data.firm(provincia)* and *data.firm.a(provincia)*, where "provincia" is a character variable indicating one of the 52 Spanish provinces. The main difference between both functions is the output: while *data.firm()* brings a data frame containing 21 variables, *data.firm.a()* creates a data frame of 12 variables, because the freelanders or self-employed dataset contains less information. The data frame output of the *data.firm()* function contains the following variables for each company: location (province, municipality, address, geographic coordinates), company characteristics (name, birth, legal form, social object), main figures (number of employees, social capital, sales), economic activity codes and firm URL. The *data.firm.a()* function output data frame does not contain the variables of geographical coordinates and main figures.

Since the data collection process is time-consuming, it is possible to divide the whole procedure into a set of functions, allowing an advanced R user to parallelize the process, though it is not recommendable to avoid server crashes. These functions are available and documented in the "DataSpa" package.

Axesor constitutes an interesting alternative to the well-known SABI dataset, though it should be said that some of these variables (mainly the main figures) present incomplete information.

5. Case example: the 2017 Socioeconomic Atlas of Extremadura

The "DataSpa" package and its routines were built in order to prepare the 2017 Socioeconomic Atlas of Extremadura (Junta de Extremadura 2017). This online publication constitutes an extensive compendium of valuable statistical information relevant to the region or autonomous community of Extremadura (Spain), which is presented in tables and thematic maps for different spatial scales. Most of the tables are maps containing data of each of the 388 municipalities (LAUs) of Extremadura. There are some indicators and maps for the districts of the seven main municipalities (those with more than 25,000 inhabitants): Almendralejo, Badajoz, Cáceres, Don Benito, Mérida, Plasencia, and Villanueva de la Serena. Some tables also present aggregated data for the two NUTS 3 or provinces of Extremadura and 28 commonwealths of municipalities or "mancomunidades" (formerly NUTS 4 or LAU 1).

The Atlas is comprised of nine chapters and four annexes including, among others, many indicators of economic activity, demographic phenomena, entrepreneurship, and social welfare. Many of these variables have been downloaded and treated using "DataSpa", as shown in Table 1. URL parsing download, loading, and manipulating functions have been crucial to generate tables of socio-demographic indicators. This is the case of the following variables:

Table 1

"DataSpa" functions used in the 2017 Socioeconomic Atlas of Extremadura

Atlas chapters	Statistical information	Package functions				
		Download	Loading	Manipulation	PDF extraction	Web scrapping
I: Economic indicators	Population	*getbase.pob()*	*pob.tot()*			
	Unemployment by sex	*getbase.paro()*	*paro()*	*pob.ind()*		
	Vehicle fleet by type				*parque.aut()*	
II: Demographic indicators	Population by sex and age groups	*getbase.pob()*	*pob.tot()* *pob.h.tot()* *pob.m.tot()* *pob.a()* *porb.q()*	*pob.ind()* *pob.ind.p()*		
	National and foreign population by sex	*getbase.pob()*	*pob.n()* *pob.e()*			
	Demographic phenomena	*getbase.fen()*	*pob.fen()*	*pob.ind()*		
IV: Trade areas	Population	*getbase.pob()*	*pob.tot()*			
V: Entrepreneurship						*data.firm()* *data.firm.a()*
VI: Evolution of indicators	Population by sex panels	*getbase.pob()*	*pob.tot()* *pob.h()* *pob.m()*	*pob.ev() pob.h.ev()* *pob.m.ev()*		
	National and foreign population panels	*getbase.pob()*	*pob.e()*	*pob.n.ev()* *pob.e.ev()*		
	Demographic panels	*getbase.fen()*	*pob.fen()*	*ind.ev()*		
	Unemployment panels	*getbase.paro()*	*paro()*	*ind.ev()*		
	Vehicle fleet panels				*parque.aut()*	
VIII: Mancomunidades	Population	*getbase.pob()*	*pob.a()*			
	Demographic phenomena	*getbase.fen()*	*pob.fen()*			
	Unemployment	*getbase.paro()*	*paro()*			
IX: Municipality maps	Population by age groups	*getbase.pob()*	*pob.q()* *pob.e()*	*pob.ind()* *pob.ind.p()*		

Atlas chapters	Statistical information	Package functions				
		Download	Loading	Manipulation	PDF extraction	Web scrapping
	Foreign population	*getbase.pob()*	*pob.e()*			
	Demographic phenomena	*getbase.fen()*	*pob.fen()*	*pob.ind()*		
	Population panels	*getbase.pob()*	*pob.tot()*	*pob.ev()*		

(a) Total population.

(b) Population by sex: males and females.

(c) Population by age groups: childhood index, youth index, old-age indexes, average age of the population.

(d) Population by nationality: nationals and foreigners.

(e) Natural population movement: birth, death, fertility, and maternity rates.

(f) Unemployment: number of unemployed people and unemployment rates by sex.

(g) Panels of time series for many of the previous municipality databases from 2000 to 2016.

A PDF extraction function was employed to download and build tables for vehicle fleet by type (automobiles, motorcycles, vans, etc.) and their corresponding time-series panels from 2000 to 2015. Finally, two web scrapping functions allowed us to download and build the following variables for entrepreneurship in companies and the self-employed: number of entrepreneurial activities, entrepreneurial activity index, sectoral participation rates and local sectorial specialization rates. All these variables are offered by activity sectors:

(a) Primary sector: agriculture, farming, forestry and fishing activities.

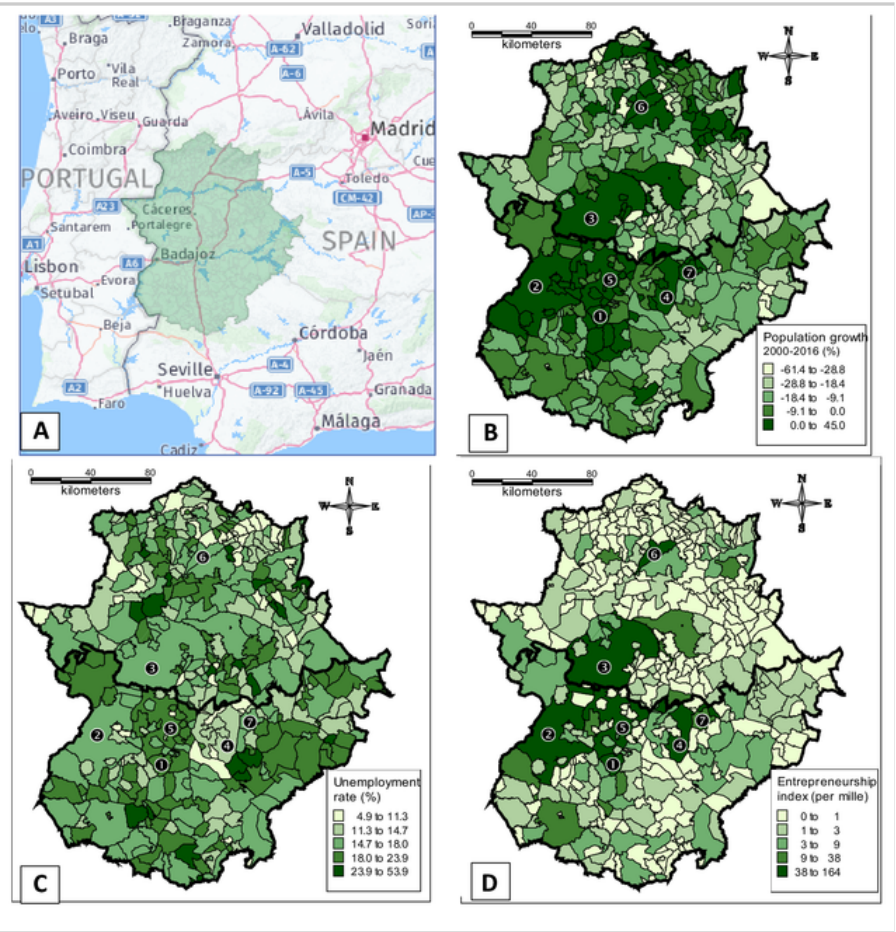
(b) Secondary sector: industry and construction.

(c) Tertiary sector: wholesale, retailing (food, non-food and department stores), hotels and restaurants, transport and communications, financial and real estate, education, health and social services, and professional, artistic and leisure activities.

Figure 4a represents the zoning map of Extremadura, which is an inland autonomous community of southwestern Spain whose capital city is Mérida. It is a large region, compared to Spain as a whole, with more than one million inhabitants and very low population density (26 km^2), which is divided into two provinces (NUTS 3), Badajoz and Cáceres. Located halfway between Madrid and Lisbon, it is a great hub to access the Spanish and Portuguese markets through its good communications with the most important Atlantic seaports of the Iberian Peninsula. However, Extremadura has traditionally been a rural impoverished region of Spain whose difficult conditions pushed many of its young people to seek their livelihood elsewhere and even overseas. For this reason, it is the only Spanish region receiving structural funds from the European Union. In spite of this secular backwardness, Extremadura is currently a region where incipient network of RTD + i centers supporting entrepreneurs, wildlife, ancestral customs, and historic cultural heritage come together in harmony.

Fig. 4

Zoning map of the region of Extremadura (Spain) and thematic maps of some indicators treated by the “DataSpa” package for the 2017 Socioeconomic Atlas of Extremadura. The main municipalities, with more than 25,000 inhabitants, are 1 Almendralejo, 2 Badajoz, 3 Cáceres, 4 Don Benito, 5 Mérida, 6 Plasencia and 7 Villanueva de la Serena



In order to have a better knowledge of such a diverse autonomous community, the regional government of Extremadura publishes the Socioeconomic Atlas biannually, containing almost 200,000 data and more than 400 variables. In Fig. 4b, c, we represent two indicators extracted and built—from INE and SEPE, respectively—with the "DataSpa" URL parsing functions. As shown in (B), during the last 15 years, population growth was negative at the level of municipalities, except in the bigger towns and their surroundings. This evolution is part of the "population desertification" process existing in the inlands in Spain, which particularly affects the peripheral Extremaduran municipalities. Unemployment (C) is still a big issue in this region especially affecting, among others, the commonwealths of municipalities ("mancomunidades") located at two main natural reservoirs of the Tagus and Guadiana rivers. Finally, we illustrate the distribution of the entrepreneurship index (D), which is the share (in per thousands) of the local firms and self-employed —web scrapped from the Axesor database with "DataSpa"—over the regional aggregates. Entrepreneurial activity is concentrated in the main cities, though there are also some intermediate centers arising from this cores in the towns of Alburquerque and Jerez de los Caballeros (west), Zarza de Granadilla, Navalmoral de la Mata and Villanueva de la Vera (north) and Llerena (south).

6. Conclusions

The internet undoubtedly increases the information availability and the way that researchers interact with data. Nowadays the use of internet-enabled databanks increases the chance of access to a large amount of` primary and secondary information. However, information disposal varies depending on data category and country. Major difficulties arise with geographical downscaling. In fact, the collection of microdata at low geographical level could become a challenge for urban and intra-urban analysis.

Particularly at these lower geographical scales, researchers may deal with data availability and accessibility problems. Accessibility problems are caused when the way that data is published on servers blocks or delays the download process, often producing errors in the construction process of big databases. Availability problems usually arise when the official agencies restrict access to the information, producing empty data records and incomplete databases. To overcome these problems, it is necessary to use new data extraction strategies and explore new information sources. In this paper, we present a set of functions which explore different methods and sources to generate databases for Spain at the municipality level (NUTS 5).

Using the URL parsing strategy, we built a set of functions to download, load, and manipulate population and unemployment databases, thereby solving accessibility problems presented on two official agency web portals. We resolved accessibility problems in the vehicle fleet database using a combination of URL parsing and PDF extraction strategies. We built the *parquet.aut()* function, which employs a URL parsing strategy to download the PDF files with the municipality reports from the DGT web portal, in order to extract statistical data with a PDF extraction strategy. We also dealt with availability problems in the construction of the firm database, for which we applied a web scraping strategy with the functions *data.firm()* and *data.firm.a()* to download the information of firms and freelancers freely published by a private company. The creation of a firm database is very helpful to facilitate knowledge about the distribution of economic activities in Spain at urban and individual levels.

All these functions comprise the "DataSpa" R package, which is freely accessible under a GPL-2 license. This package is useful as a case example for countries and regions with similar statistical problems to Spain. It allows for researchers, entrepreneurs, and policy makers to have a better, more specific knowledge of Spanish cities and regions by linking complex statistical information systems. This is the case of the 2017 Socioeconomic Atlas of Extremadura, for which "DataSpa" was built, which constitutes the most important official database of municipality variables in this region.

Acknowledgements

This work was supported by Spanish Ministry of Economics and Competitiveness (ECO2015-65758-P) and the Regional Government of Extremadura (Spain). The usual disclaimers apply.

References

Arauzo Carod JM (2005) Determinants of industrial location: an application for Catalan municipalities*. Pap Reg Sci 84:105–120. https://doi.org/10.1111/j.1435-5957.2005.00006.x

Arauzo-Carod J-M, Viladecans-Marsal E (2009) Industrial location at the intra-metropolitan level: the role of agglomeration economies. Reg Stud 43:545–558. https://doi.org/10.1080/00343400701874172

Atkinson AB, Brandolini A (2001) Promise and pitfalls in the use of "secondary" data-sets: income inequality in OECD countries as a case study. J Econ Lit 39:771–799. https://doi.org/10.1257/jel.39.3.771

Aumueller D (2009) Retrieving metadata for your local scholarly papers. BTW

AQ2

Beel J, Langer S, Genzmehr M, Müller C (2013) Docear's PDF inspector: title extraction from PDF files. In: Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries. ACM, pp 443–444

Bento AM, Cropper ML, Mobarak AM, Vinha K (2005) The effects of urban spatial structure on travel demand in the United States. Rev Econ Stat 87:466–478. https://doi.org/10.1162/0034653054638292

Beretta M, Bjork J, Magnusson M (2018) Moderating ideation in web-enabled ideation systems. J Prod Innov Manag 35:389–409. https://doi.org/10.1111/jpim.12413

Berners-Lee RFT, Masinter L (2015) Uniform Resource Identifier (URI): generic syntax, request for comments: 3986, January 2005

Bhargavan K, Delignat-Lavaud A, Maffeis S (2013) Language-based defenses against untrusted browser origins. In: USENIX security symposium, pp 653–670

Braaksma B, Zeelenberg K (2015) "Re-make/Re-model": should big data change the modelling paradigm in official statistics? Stat J IAOS 31:193–202. https://doi.org/10.3233/sji-150892

Castillo-Fernández O (2015) Web scraping: applications and tools. European Public Sector Information Platform

Chaabane S, Jaziri W (2018) A novel algorithm for fully automated mapping of geospatial ontologies. J Geogr Syst 20:85–105. https://doi.org/10.1007/s10109-017-0263-0

Chang C-H, Kayed M, Girgis MR, Shaalan KF (2006) A survey of web information extraction systems. IEEE Trans Knowl Data Eng 18:1411–1428. https://doi.org/10.1109/TKDE.2006.152

Chen Z, Wenyin L, Zhang F et al (2001) Web mining for web image retrieval. J Am Soc Inform Sci Technol 52:831–839. https://doi.org/10.1002/asi.1132

Chen M, Arribas-Bel D, Singleton A (2019) Understanding the dynamics of urban areas of interest through volunteered geographic information. J Geogr Syst 21:89–109. https://doi.org/10.1007/s10109-018-0284-3

Denissen JJA, Neumann L, van Zalk M (2010) How the internet is changing the implementation of traditional research methods, people's daily lives, and the way in which developmental scientists conduct research. Int J Behav Dev 34:564–575. https://doi.org/10.1177/0165025410383746

Deniz C (2019) A command line program to get daily tv ratings in Turkey: https://github.com/coskundeniz/ratingpy

AQ3

Dowell KG, McAndrews-Hill MS, Hill DP et al (2009) Integrating text mining into the MGI biocuration workflow. Database (Oxford). https://doi.org/10.1093/database/bap019

Edelman B (2012) Using internet data for economic research. J Econ Perspect 26:189–206. https://doi.org/10.1257/jep.26.2.189

Eluru N, Bhat CR, Pendyala RM, Konduri KC (2010) A joint flexible econometric model system of household residential location and vehicle fleet composition/usage choices. Transportation 37:603–626. https://doi.org/10.1007/s11116-010-9271-3

Fernández P, Suárez JP, Trujillo A et al (2018) 3D-monitoring big geo data on a seaport infrastructure based on FIWARE. J Geogr Syst 20:139–157. https://doi.org/10.1007/s10109-018-0269-2

Futrelle RP, Shao M, Cieslik C, Grimes AE (2003) Extraction, layout analysis and classification of diagrams in PDF documents. In: Proceedings. seventh international conference on Document analysis and recognition, 2003. IEEE, pp 1007–1013

Glavas C, Mathews S, Russell-Bennett R (2018) Knowledge acquisition via internet-enabled platforms: examining incrementally and non-incrementally internationalizing SMEs. Int Mark Rev 36:74–107. https://doi.org/10.1108/IMR-02-2017-0041

González-Peña D, Lourenço A, López-Fernández H et al (2014) Web scraping technologies in an API world. Brief Bioinform 15:788–797

Gök A, Waterworth A, Shapira P (2015) Use of web mining in studying innovation. Scientometrics 102:653–671. https://doi.org/10.1007/s11192-014-1434-0

Graham M, Hogan B, Straumann RK, Medhat A (2014) Uneven geographies of user-generated information: patterns of increasing informational poverty. Ann Assoc Am Geogr 104:746–764. https://doi.org/10.1080/00045608.2014.910087

Griffioen R, de Haan J, Willenborg L (2014) Collecting clothing data from the Internet. In: Proceedings of meeting of the group of experts on consumer price indexes, pp 26–28

Hadjar K, Rigamonti M, Lalanne D, Ingold R (2004) Xed: a new tool for extracting hidden structures from electronic documents. In: Proceedings of the first international workshop on document image analysis for libraries, 2004, pp 212–224

Hansen MC, Egorov A, Potapov PV et al (2014) Monitoring conterminous United States (CONUS) land cover change with Web-Enabled Landsat Data (WELD). Remote Sens Environ 140:466–484.

https://doi.org/10.1016/j.rse.2013.08.014

Herley C (2009) So long, and no thanks for the externalities: the rational rejection of security advice by users. In: Proceedings of the 2009 workshop on new security paradigms workshop. ACM, pp 133–144

Hooley T, Wellens J, Marriott J (2011) What is online research? Using the Internet for social science research. A&C Black

Howard P, Pulcini C, Levy Hara G et al (2015) An international cross-sectional survey of antimicrobial stewardship programmes in hospitals. J Antimicrob Chemother 70:1245. https://doi.org/10.1093/jac/dku497

Jofre-Monseny J, Marín-López R, Viladecans-Marsal E (2011) The mechanisms of agglomeration: evidence from the effect of inter-industry relations on the location of new firms. J Urban Econ 70:61–74. https://doi.org/10.1016/j.jue.2011.05.002

Kahn ME, Schwartz J (2008) Urban air pollution progress despite sprawl: the “greening” of the vehicle fleet. J Urban Econ 63:775–787. https://doi.org/10.1016/j.jue.2007.06.004

Katre P (2019) Web scrapping and exploratory data analysis using beautiful soup and plotly on Indian demographics. katreparitosh/Web-Scrapping-and-EDA

Kumar SN (2015) World towards advance web mining: a review. Am J Syst Softw 3:44–61

Lagacé E (2019) Python script to extract subway turnstile data files from the New York. MTA website: https://github.com/RollingHillsAnalytics/MTA-extraction

LeSage JP (2015) Software for Bayesian cross section and panel spatial model comparison. J Geogr Syst 17:297–310. https://doi.org/10.1007/s10109-015-0217-3

Liu Y, Zhang M (2012) Financial websites oriented heuristic anti-phishing research. In: 2012 IEEE 2nd international conference on cloud computing and intelligence systems, pp 614–618

Mage D, Ozolins G, Peterson P et al (1996) Urban air pollution in megacities of the world. Atmos Environ 30:681–686. https://doi.org/10.1016/1352-2310(95)00219-7

Marinai S (2009) Metadata extraction from PDF papers for digital library Ingest. In: 2009 10th International conference on document analysis and recognition, pp 251–255

Mehlführer A (2009) Web scraping: a tool evaluation. Master's Thesis, Wien University

Munzert S, Rubba C, Meisner P, Nyhuis D (2015) Automated data collection with R: a practical guide to web scraping and text mining. John Wiley & Sons, Ltd. Chichester, West Sussex, UK

National Research Council (2005) Expanding access to research data: reconciling risks and opportunities. Division of Behavioral and Social Sciences and Education, The National Academies Press, Washington, DC

Navarro D (2019) This web scraper builds a dataset for São Paulo subway operation status. https://github.com/douglasnavarro/sp-subway-scraper

Nolan D, Temple Lang D (2014) XML and web technologies for data sciences with R. Springer, New York

Nygaard R (2015) The use of online prices in the Norwegian Consumer Price Index. In: Meeting of the Ottowa Group, Tokyo, Japan

Papapesios N, Ellul C, Shakir A, Hart G (2019) Exploring the use of crowdsourced geographic information in defence: challenges and opportunities. J Geogr Syst 21:133–160. https://doi.org/10.1007/s10109-018-0282-5

Paskaleva K, Cooper I (2018) Open innovation and the evaluation of internet-enabled public services in smart cities. Technovation 78:4–14. https://doi.org/10.1016/j.technovation.2018.07.003

Penman RB, Baldwin T, Martinez D (2009) Web scraping made simple with sitescraper. Citeseer

Polidoro F, Giannini R, Conte RL et al (2015) Web scraping techniques to collect data on consumer electronics and airfares for Italian HICP compilation. Stat J IAOS 31:165–176

Rey SJ, Anselin L (2006) Recent advances in software for spatial analysis in the social sciences. Geogr Anal 38:1–4. https://doi.org/10.1111/j.0016-7363.2005.00670.x

Roy DP, Ju J, Kline K et al (2010) Web-Enabled Landsat Data (WELD): Landsat ETM+ composited mosaics of the conterminous United States. Remote Sens Environ 114:35–49. https://doi.org/10.1016/j.rse.2009.08.011

Salamone S, Scannapieco SM, Scarnò M (2014) Web scraping and web mining: new tools for official statistics. In: Proceedings of the Societa Italiana di Statistica (SIS 2014), Cagliari, Sardegna

Santiago G (2019) Web Scraping para coletar os dados da Folha de Pessoal dos Municípios (BA) no site do TCM-Ba: georgevbsantiago/tcmbapessoal

Sellers J (2019) Document-level sentiment analysis of book reviews scraped from the Goodreads website. Technologies used include TensorFlow, Spark, HDFS, Sqoop, Scrapy, and D3.js.: JohnSell620/sentiment-analysis-g

Siewert W, Udani A (2016) Missouri municipal ethics survey: Do ethics measures work at the municipal level? Public Integr 18:269–289. https://doi.org/10.1080/10999922.2016.1139523

Skitka LJ, Sargis EG (2006) The internet as psychological laboratory. Annu Rev Psychol 57:529–555. https://doi.org/10.1146/annurev.psych.57.102904.190048

Thaiprayoon S, Haruechaiyasak AKC (2016) PDF extraction based on lexical analysis for Thai texts. Int J Appl Comput Technol Inf Syst 5:7-9

AQ4

Vallone A, Chasco C, Sanchez B (2017) DataSpa: functions to collect Spanish data at municipality level: https://github.com/amvallone/DataSpa

Walker K, Eberwein K, Herman M (2019) tidycensus: load US census boundary and attribute data as “tidyverse” and ‘sf’-ready data frames. https://walkerke.github.io/tidycensus/. Accessed 5 Sept. 2018.

AQ5

Wang H, Fu L, Lin X et al (2009) A bottom-up methodology to estimate vehicle emissions for the Beijing urban area. Sci Total Environ 407:1947–1953. https://doi.org/10.1016/j.scitotenv.2008.11.008

Westling EL, Lerner DN, Sharp L (2009) Using secondary data to analyse socio-economic impacts of water management actions. J Environ Manag 91:411–422. https://doi.org/10.1016/j.jenvman.2009.09.011

Wickham H (2016) Package ‘rvest’. https://cran.r-project.org/web/packages/rvest/rvest.pdf. Accessed 5 Sept. 2018.

Wickham H (2017) Package ‘stringr.’ https://cran.r-project.org/web/packages/stringr/stringr.pdf. Accessed 5 Sept. 2018.

William Xu X, Liu T (2003) A web-enabled PDM system in a collaborative design environment. Robot Comput Integr Manuf 19:315–328. https://doi.org/10.1016/S0736-5845(02)00082-0

Wolf LJ (2019) cenpy: explore and download data from census APIs. https://github.com/ljwolf/cenpy. Accessed 5 Sept. 2018.

Wright KB (2005) Researching internet-based populations: advantages and disadvantages of online survey research, online questionnaire authoring software packages, and web survey services. J Comput Mediat Commun.

https://doi.org/10.1111/j.1083-6101.2005.tb00259.x

Xavier R (2019) Web scraping to obtain laws and decrees approved by the Uruguayan government: rxavier/volnormativo

Zagayevskiy Y, Deutsch CV (2016) Multivariate grid-free geostatistical simulation with point or block scale secondary data. Stoch Environ Res Risk Assess 30:1613–1633. https://doi.org/10.1007/s00477-015-1154-x

Zuhair H, Selamat A, Salleh M (2016) New hybrid features for phish website prediction. Int J Adv Soft Comput Its Appl 8

---

[1] We consider 'web-enabled' as different than 'web-based', which is related to methods used in psychology and behavioral studies (Skitka and Sargis 2006; Denissen et al. 2010).

[2] According Eurostat, The LAUs (Local Administrative Units) are subdivisions of the NUTS 3 regions, which consist of municipalities or equivalent units (formerly NUTS 5). The NUTS classification (Nomenclature of territorial units for statistics) is a hierarchical system for dividing the economic territory of the EU. NUTS 1 are major socio-economic regions (e.g. Spain), NUTS 2 are basic regions for the application of regional policies (e.g. autonomous community of Extremadura) and NUTS 3 are small regions for specific diagnoses (e.g. province of Badajoz).

[3] This R package is freely available from the site https://github.com/amvallone/DataSpa. It must be installed in the R console with the command: devtools::install_github("amvallone/DataSpa").

[4] http://www.ine.es.

[5] http://www.sepe.es.

[6] All the R functions are in the aforementioned repository: https://github.com/amvallone/DataSpa.

[7] These functions deal with two important difficulties derived from the construction of panels for municipality data in Spain. First, they control for municipality entries and removals, which take place almost every year, adapting the final data frame to the configuration corresponding to the last period. Second, they produce a list of name equivalences, based on the information provided by the INE, to manage with constant changes in the municipality names, always assigning the one corresponding the last period.

[8] www.dgt.es.

[9] http://www.dgt.es/es/seguridad-vial/estadisticas-e-indicadores/informacion-municipal.

[10] https://www.pdf2txt.com.

[11] http://www.ine.es/dynt3/inebase/es/index.htm?padre=51&dh=1.

[12] http://www.minetad.gob.es/industria/RII/Paginas/Index.aspx.

[13] http://www.camerdata.es/index.php.

[14] https://www.bvdinfo.com/en-gb/our-products/data/national/sabi.

[15] http://www.gem-spain.com.

[16] https://www.axesor.es.