Desarrollo de una librería de R para la obtención y análisis de datos desde fuentes open data

Development of an R library for obtaining and analyzing data from open data sources

Andrés Nacimiento García
Dpto. de Matemáticas, Estadística e Investigación Operativa
Escuela Superior de Ingeniería y Tecnología

Trabajo de Fin de Máster

La Laguna, 29 de junio de 2018

D. **Carlos J. Pérez González**, profesor de Universidad adscrito al Departamento de Matemáticas, Estadística e Investigación Operativa de la Universidad de La Laguna

C E R T I F I C A

Que la presente memoria titulada:

"Desarrollo de una librería de R para la obtención y análisis de datos desde fuentes open data."

ha sido realizada bajo su dirección por D. Andrés Nacimiento García.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 29 de junio de 2018.

Agradecimientos

Agradecimiento especial a Jésica Carballo por su apoyo incondicional.

Al director del proyecto por su tiempo y dedicación.

Al Instituto Nacional de Estadística (INE), Instituto de Estadística y Cartografía de Andalucía (IECA) y al Instituto Canario de Estadística (ISTAC) por su interés y apoyo en este proyecto.

A toda esa gente anónima que aporta documentación en internet para que proyectos como éste se puedan llevar a cabo.

Resumen

En colaboración con el ISTAC (Instituto Canario de Estadística, el TFM se centrará en la obtención, tratamiento y análisis de datos desde fuentes de datos abiertos a través de servicios API JSON o XML. Las tecnologías utilizadas serán aplicaciones de software libre y lenguajes de programación como Jupyter, Python y R.

Palabras clave

R, RStudio, Librería, Package, Software libre, Datos abiertos, Análisis, INE, ISTAC.

Abstract

In collaboration with ISTAC (Instituto Canario de Estadística), this TFM will focus on obtaining, processing and analyzing data from open data sources through JSON or XML API services. The technologies used will be free software applications and programming languages such as Jupyter, Python and R.

Keywords

R, RStudio, Library, Package, Open Source, Open Data, Analysis, INE, ISTAC.

Índice General

Índice de figuras

Índice de tablas

Capítulo 1. Introducción

El objetivo del presente trabajo fin de máster es el estudio y análisis de datos desde fuentes de datos abiertos a través de servicios API JSON o XML, así como el desarrollo e implementación de una librería en R que permita el acceso a los mismos. Las tecnologías utilizadas serán aplicaciones de software libre y lenguajes de programación como Jupyter, Python y R.

1.1 Open Data (datos abiertos)

Los datos abiertos, conocidos con el término en inglés Open Data, son una filosofía que persigue que ciertos datos que han estado en dominio de organizaciones (públicas o privadas) sean liberados a todo el mundo, sin restricciones de licencias, copyright y/o patentes. Estos datos están centrados en material no-documental como información geográfica, el genoma, compuestos químicos, fórmulas matemáticas y científicas, datos médicos, biodiversidad, etc. Para que un dato sea abierto, tiene que ser accesible y reutilizable, sin exigir permisos específicos, aunque los tipos de reutilización pueden estar controlados mediante una licencia.

Con esto conseguimos que se fomente la creación de servicios basados en la información pública por parte del sector privado, ya que tanto los ciudadanos como las empresas, u otras instituciones, pueden utilizar la información pública para desarrollar servicios —de pago o no— que complementarán los que les proporciona la Administración.

Entre las iniciativas públicas de datos abiertos podemos citar los portales web de datos del Gobierno de España[1] y del Gobierno de Canarias[2].

1.2 LEY 37/2007 (RISP)

El 16 de noviembre de 2007 se aprobó la Ley 37/2007[3], sobre Reutilización de la Información del Sector Público, que regula la reutilización de los documentos elaborados o custodiados por las Administraciones y Organismos del Sector Público y que surgió como transposición de la Directiva 2003/98/CE del Parlamento europeo y del Consejo[4].

Esta ley no modifica el régimen de acceso a los documentos administrativos ya previsto en el ordenamiento jurídico español, sino que aporta un valor añadido al derecho de acceso, estableciendo un marco de regulación básico para la explotación de la información que obra en poder del sector público.

Se ha elaborado el Real Decreto 1495/2011[5] por el que se desarrolla la Ley 37/2007, de acuerdo con la Estrategia 2011-2015 del Plan Avanza 2.

El objetivo de esta iniciativa es el de detallar para el ámbito del sector público estatal las disposiciones presentes en la citada Ley 37/2007, promoviendo y facilitando al máximo la puesta a disposición de la información del sector público para su reutilización por terceros, con fines comerciales o no, en el marco de unas condiciones claras, transparentes y no discriminatorias.

En cuanto a la forma de proporcionar la información, la iniciativa RISP, siguiendo la corriente internacional promulgada por universidades e importantes empresas y que se sustenta en la propuesta de Tim Berners Lee, inventor de la World Wide Web y Director de la World Wide Web Consortium (W3C)[6] clasifica la forma de ofrecer la información en función de sus formatos de representación en los siguientes grupos:

- **Nivel 1**: Publicación en cualquier formato.
- **Nivel 2**: Publicación en formatos estructurados (por ejemplo, Excel).
- **Nivel 3**: Publicación en formatos no propietarios (por ejemplo, CSV).
- **Niveles 4 y 5**: Publicación mediante formatos con información semántica.

El RD 1495/2011, de 24 de octubre, precisa en su Artículo 7, que serán de aplicación las siguientes condiciones generales para todas las modalidades de puesta a disposición de los documentos reutilizables:

a) No desnaturalizar el sentido de la información.
b) Citar la fuente de los documentos objeto de la reutilización.
c) Mencionar la fecha de la última actualización de los documentos objeto de la reutilización, siempre y cuando estuviera incluida en el documento original.
d) No se podrá indicar, insinuar o sugerir que los órganos administrativos, organismos o entidades del sector público estatal titulares de la información reutilizada participan, patrocinan o apoyan la reutilización que se lleve a cabo con ella.
e) Conservar y no alterar ni suprimir los metadatos sobre la fecha de actualización y las condiciones de reutilización aplicables incluidos, en su caso, en el documento puesto a disposición para su reutilización por la Administración u organismo del sector público.

La Norma Técnica de Interoperabilidad de Reutilización de Recursos de Información[7], recomienda la definición de un esquema de Identificadores de Recursos Uniformes o URI, de modo que se permita disponer de un mecanismo de identificación para los datos que se exponen públicamente.

Posibilitando así la identificación única, fiable y persistente en el tiempo, requisito clave para facilitar la reutilización de estos.

En ese sentido se seguirán las siguientes recomendaciones de la citada Norma Técnica de Interoperabilidad:

- Empleo del protocolo HTTP.
- Empleo de una estructura de composición de URI consistente, extensible y persistente. Siguiendo unos patrones de construcción.
- Seguimiento de una estructura de composición comprensible y significativa.
- Cumplimiento del principio de persistencia.

1.3 API (Application Programming Interface)

La interfaz de programación de aplicaciones, abreviada como API del inglés: Application Programming Interface, es un conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

1.4 Análisis de datos en Estadística Pública

La estadística pública se caracteriza básicamente por estar financiada por recursos públicos y por obtenerse mayoritariamente de las relaciones existentes entre los ciudadanos y la propia administración.

Entre muchos otros, destacamos los siguientes archivos de interés: Padrón Municipal de Habitantes, Estadísticas de movimientos demográficos, Impuesto de actividades económicas, Impuesto de circulación, características catastrales de los edificios, viviendas y del suelo, licencias concedidas para obras de construcción.

La estadística pública debe de generar una atención específica hacia las áreas densamente habitadas, las cuales generan unas dinámicas muy determinadas y contribuyen decisivamente, tanto en cantidad como en calidad al desarrollo económico y social de cualquier país.

En la actualidad, la administración pública dispone de servicios web API para el libre acceso a su información (datos abiertos) que permiten realizar un análisis estadístico de la misma.

Capítulo 2. Estado del arte

En la comunidad existe una escasez de librerías de código abierto que permitan la explotación de datos abiertos (open data), a pesar de la gran utilidad que estas puedan llegar a tener.

A continuación, en la sección 2.1 se estudiarán las principales alternativas software que existen en la actualidad para el análisis de datos estadísticos.

Se presentan en la sección 2.2 algunos ejemplos de APIs y librerías que permiten la extracción de datos abiertos desde fuentes públicas.

2.1 Software para el análisis de datos estadísticos desde fuentes open data

Las librerías y APIs mencionadas en el capítulo anterior, permiten la extracción de datos abiertos de diversas fuentes de información.

En el siguiente apartado 2.1.1 se definen algunas herramientas (software) y lenguajes de programación de software libre, y en el apartado 2.1.2 alternativas de software comercial.

2.1.1 Alternativas libres

Cada día son más los investigadores que se decantan por la utilización de herramientas de software libre para el análisis de datos estadísticos. A continuación, se detallan las principales herramientas y lenguajes de programación utilizados con este fin.

Lenguaje de programación: R

R es lenguaje de programación enfocado principalmente al análisis estadístico y es una implementación de software libre del lenguaje S.

Se trata de uno de los lenguajes más utilizados en investigación, siendo además muy popular en el campo de la minería de datos, investigación biomédica, bioinformática y las matemáticas financieras. A esto contribuye la posibilidad de cargar diferentes bibliotecas o paquetes con funcionalidades de cálculo y gráficas.

En este proyecto se ha utilizado el lenguaje R por su capacidad de extracción, análisis y representación de datos abiertos.

Algunos packages (librerías) de R para el análisis de datos a destacar son Rcommander[1] y ROctave[2].

Entorno de desarrollo integrado (IDE): RStudio

RStudio es un entorno de desarrollo integrado (IDE) para el lenguaje de programación R, dedicado a la computación estadística y gráficos.

Este entorno de desarrollo incluye una consola, editor de sintaxis que apoya la ejecución de código, así como herramientas para el trazado, la depuración y la gestión del espacio de trabajo.

RStudio es un software que está disponible para Windows, Mac y Linux o para navegadores conectados a RStudio Server. RStudio tiene la misión de proporcionar el entorno informático estadístico R y permite un análisis y desarrollo para que cualquiera pueda analizar los datos con R.

---

[1] http://www.rcommander.com/

[2] http://www.omegahat.net/ROctave/

Lenguaje de programación: Python

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.

Python tiene su origen en 1991, y fue creado por Guido Van Rossum[8] con el objetivo de hacer un lenguaje de programación ágil y sencillo, con una curva de aprendizaje muy corta.

Esto es una gran ventaja para el crecimiento en el uso de la sintaxis a nivel internacional. Desde sus comienzos está destinada a los profesionales procedentes del mundo de la estadística y ciencia de datos, pero sus características han ampliado mucho el campo de uso de Python.

Una de las principales ventajas de este lenguaje de programación es que es uno de los lenguajes de programación con mayor crecimiento en la actualidad.

En junio de 2017 fue el primer mes en que Python fue la etiqueta más visitada en Stack Overflow[3] junto a Java y JavaScript.

Esto es especialmente impresionante porque en 2012, fue menos visitado que cualquiera de los otros 5 lenguajes, y ha crecido 2.5 veces en ese tiempo.

[3] https://stackoverflow.com/

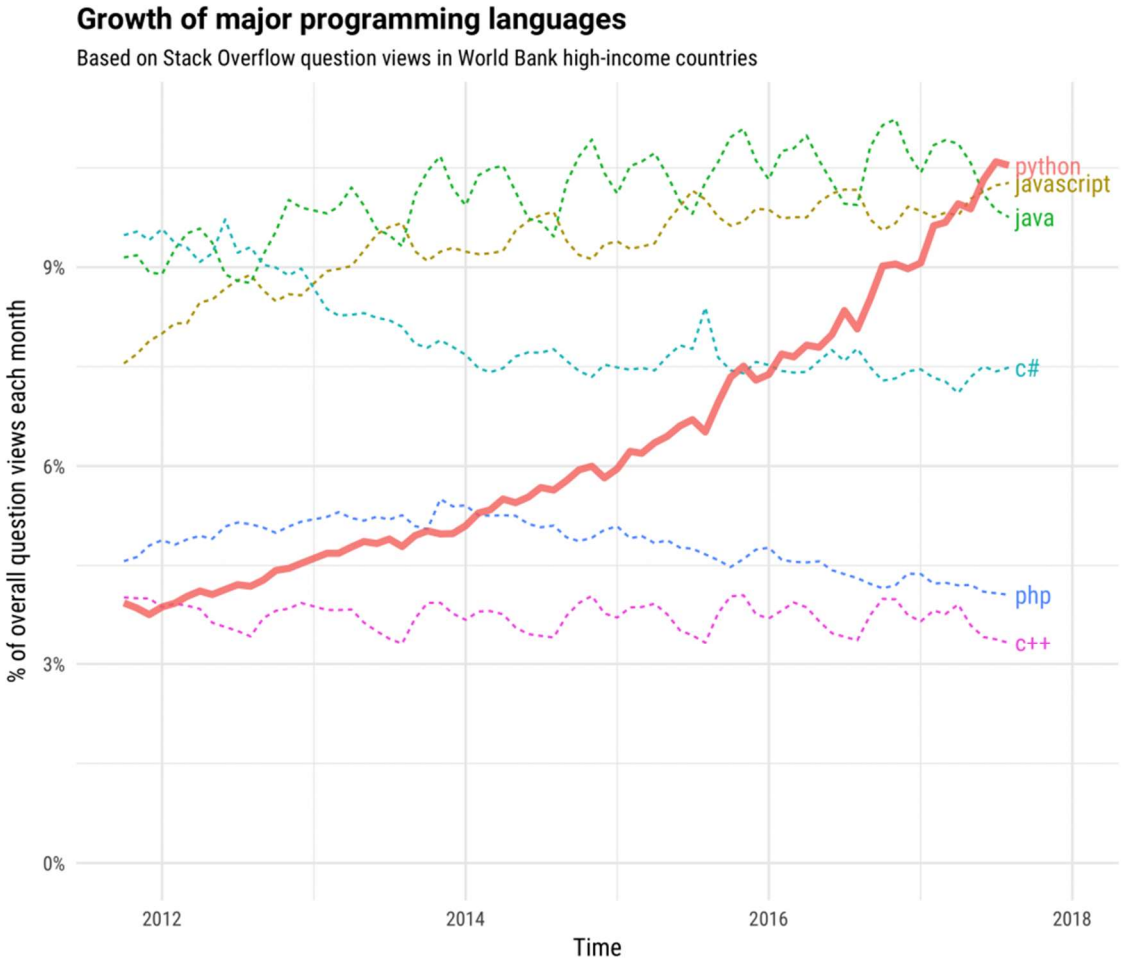

Figura 2.1 Gráfico que muestra el crecimiento del lenguaje Python en los últimos años[4]

[4] https://stackoverflow.blog/2017/09/06/incredible-growth-python/

Weka

Weka (Waikato Environment for Knowledge Analysis) es un software para el aprendizaje automático y la minería de datos escrito en Java y desarrollado en la Universidad de Waikato[5], es software libre y está distribuido bajo la licencia GNU-GPL[9].

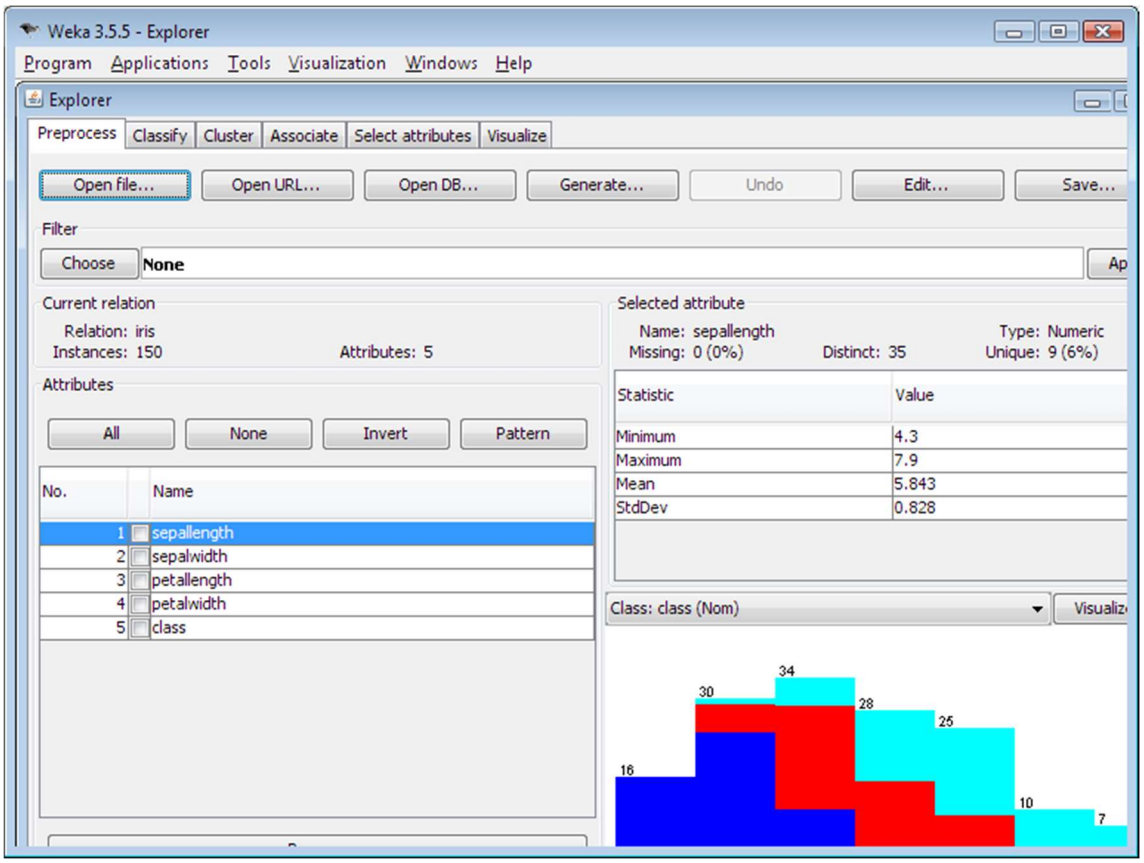

Figura 2.2 Interfaz de usuario del software Weka

[5] https://www.waikato.ac.nz/

Otras herramientas que incluyen algunos aspectos de análisis de datos son:

SCaVis

SCaVis es un entorno de computación científica, análisis de datos y visualización de datos para científicos, ingenieros y estudiantes.

El programa es multiplataforma e integrado con Java y varios lenguajes de scripts: Jython (Python)[6], Groovy[7], JRuby[8], BeanShell[9].

SCaVis se puede usar para trazar funciones y datos en 2D y 3D, realizar pruebas estadísticas, extraer datos, cálculos numéricos, minimizar funciones, álgebra lineal, resolver sistemas de ecuaciones lineales y diferenciales.

Regresión lineal, no lineal y simbólica también están disponibles. Se admiten elementos de cálculos simbólicos mediante el uso de scripts de Octave y Matlab.

Este proyecto ha sido migrado a DataMelt, un software que detallaremos a continuación.

DataMelt

DataMelt es un software matemático gratuito para científicos, ingenieros y estudiantes. Se puede usar para cálculos numéricos, estadísticas, cálculos simbólicos, análisis de datos y visualización de datos.

---

[6] http://www.jython.org/
[7] http://groovy-lang.org/
[8] http://jruby.org/
[9] http://www.beanshell.org/

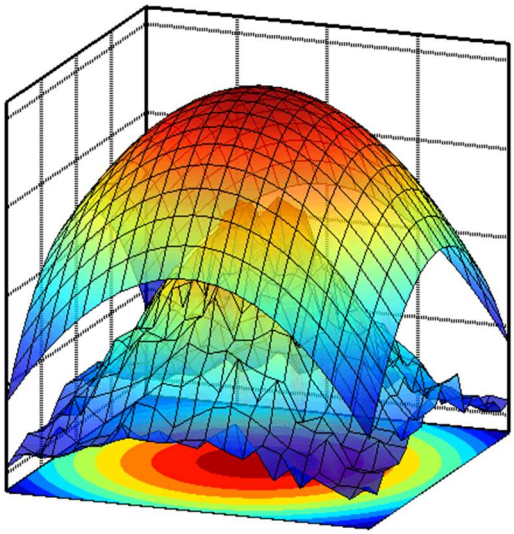

Figura 2.3 Representación gráfica de datos con el software DataMelt

RapidMiner

RapidMiner es un programa informático para el análisis y minería de datos. Permite el desarrollo de procesos de análisis de datos mediante el encadenamiento de operadores a través de un entorno gráfico y está considerada la plataforma de software libre n. 1 para análisis predictivos.

Permite a las empresas fusionar fácilmente datos, crear modelos predictivos y operacionalizar el análisis predictivo dentro de cualquier proceso comercial.

Sus principales características son:

- Desarrollado en Java.
- Multiplataforma.
- Representación interna de los procesos de análisis de datos en ficheros XML.
- Permite el desarrollo de programas a través de un lenguaje de script.

Puede usarse de diversas maneras:

- A través de un GUI.
- En línea de comandos.
- En batch (lotes).
- Desde otros programas a través de llamadas a sus bibliotecas.
- Extensible.
- Incluye gráficos y herramientas de visualización de datos.
- Dispone de un módulo de integración con R.

TANAGRA

TANAGRA es un software gratuito de minería de datos para fines académicos y de investigación. Este software propone varios métodos de minería de datos para análisis exploratorio de datos, aprendizaje estadístico, aprendizaje automático y área de bases de datos.

Este proyecto es el sucesor de SIPINA que implementa varios algoritmos de aprendizaje supervisado, especialmente una construcción interactiva y visual de árboles de decisión. TANAGRA es más potente, contiene algunos aprendizajes supervisados, pero también otros paradigmas como agrupamiento, análisis factorial, estadísticas paramétricas y no paramétricas, reglas de asociación, selección de características y algoritmos de construcción.

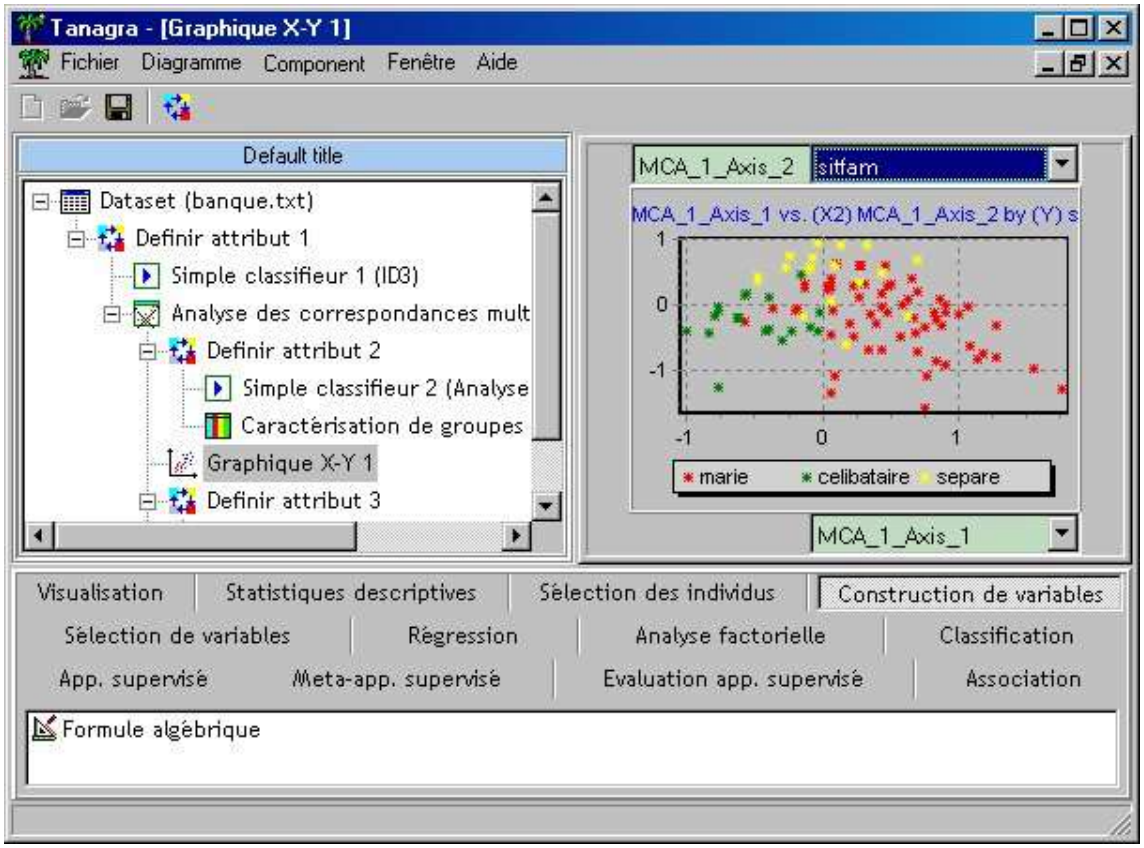

Figura 2.4 Interfaz de usuario (GUI) de la herramienta TANGARA

2.1.2 Alternativas comerciales

Además de las alternativas libres comentadas en la sección anterior, existe también gran variedad de herramientas y lenguajes de programación comerciales utilizados para el análisis y extracción de datos desde fuentes abiertas. En esta sección se detallarán alguna de las principales.

Lenguaje de programación: SAS

SAS es un lenguaje de programación desarrollado por SAS Institute[10] a finales de los años sesenta. Existen dos intérpretes de dicho lenguaje: uno desarrollado por SAS Institute y otro por la empresa World Programming[11].

El lenguaje SAS opera principalmente sobre tablas de datos: puede leerlas, transformarlas, combinarlas, resumirlas, crear informes a partir de ellas, etc.

El núcleo del lenguaje (conocido habitualmente como SAS Base) incluye:

- Pasos data que permiten realizar operaciones sobre las filas de un conjunto de datos.
- Procedimientos de manipulación de datos que permiten ordenar tablas, enlazarlas, etc.
- Un intérprete de SQL.
- Un súper lenguaje de macros.

MATLAB

MATLAB es una herramienta de software matemático que ofrece un entorno de desarrollo integrado con un lenguaje de programación propio y está disponible para las plataformas Unix, Windows, Mac OS X y GNU/Linux.

MATLAB combina un entorno de escritorio perfeccionado para el análisis iterativo y los procesos de diseño con un lenguaje de programación que expresa las matemáticas de matrices y arrays directamente.

Este software es principalmente utilizado para analítica de datos: big data, aprendizaje automático (machine learning) y análisis de producción.

[10] https://www.sas.com/es_es/home.html
[11] https://www.worldprogramming.com/es/home

Minitab

Minitab es un programa de computadora diseñado para ejecutar funciones estadísticas básicas y avanzadas. Combina lo amigable del uso de Microsoft Excel con la capacidad de ejecución de análisis estadísticos.

La ventaja de este software es que no tienes que ser un experto en estadística para obtener la información que necesitas de tus datos, Minitab cuenta con un asistente que te orienta a través de cada paso e incluso te ayuda a interpretar tus resultados.

SPSS

SPSS es un programa estadístico informático muy usado en las ciencias sociales y aplicadas, además de las empresas de investigación de mercado y ha sido desarrollado por IBM para un análisis completo.

SPSS es un software popular entre los usuarios de Windows, es utilizado para realizar la captura y análisis de datos para crear tablas y gráficas con datos complejos.

Este software permite resolver una gran variedad de problemas de negocio e investigación. Proporciona distintas técnicas, incluyendo el análisis ad-hoc, pruebas de hipótesis e informes, para facilitar la gestión de datos, seleccionar y realizar análisis y compartir los resultados.

SAS Enterprise Guide (software)

SAS Enterprise Guide es una aplicación de cliente de Windows fácil de usar que ofrece estas características:

- Acceso a mucha de la funcionalidad de SAS
- Una interfaz intuitiva, visual y personalizable
- Acceso transparente a los datos
- Tareas listas para usar para análisis e informes
- Formas fáciles de exportar datos y resultados a otras aplicaciones
- Scripting y automatización
- Una instalación de edición de código

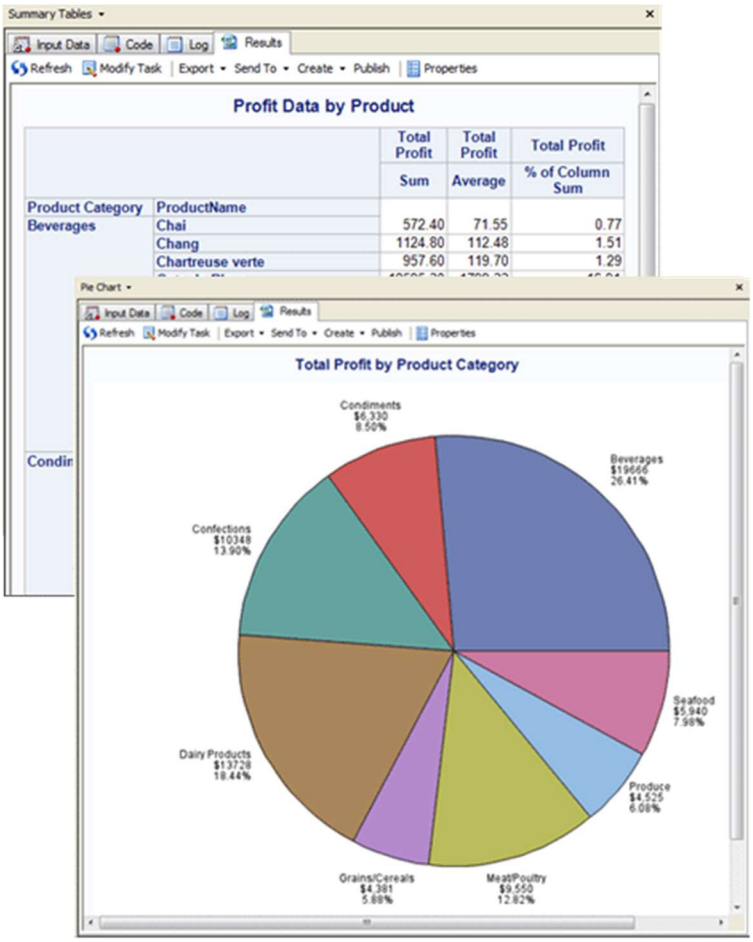

Figura 2.5 Algunas pantallas de la interfaz gráfica del software SAS Enterprise Guide

Excel

Microsoft Excel es una aplicación de hojas de cálculo que forma parte de la suite de oficina Microsoft Office. Es una aplicación utilizada en tareas financieras y contables, con fórmulas, gráficos y un lenguaje de programación.

En las nuevas versiones, Excel aprende tus patrones y organiza tus datos para ahorrarte tiempo. Crea hojas de cálculo con facilidad a partir de plantillas o desde cero, y realiza cálculos con fórmulas modernas.

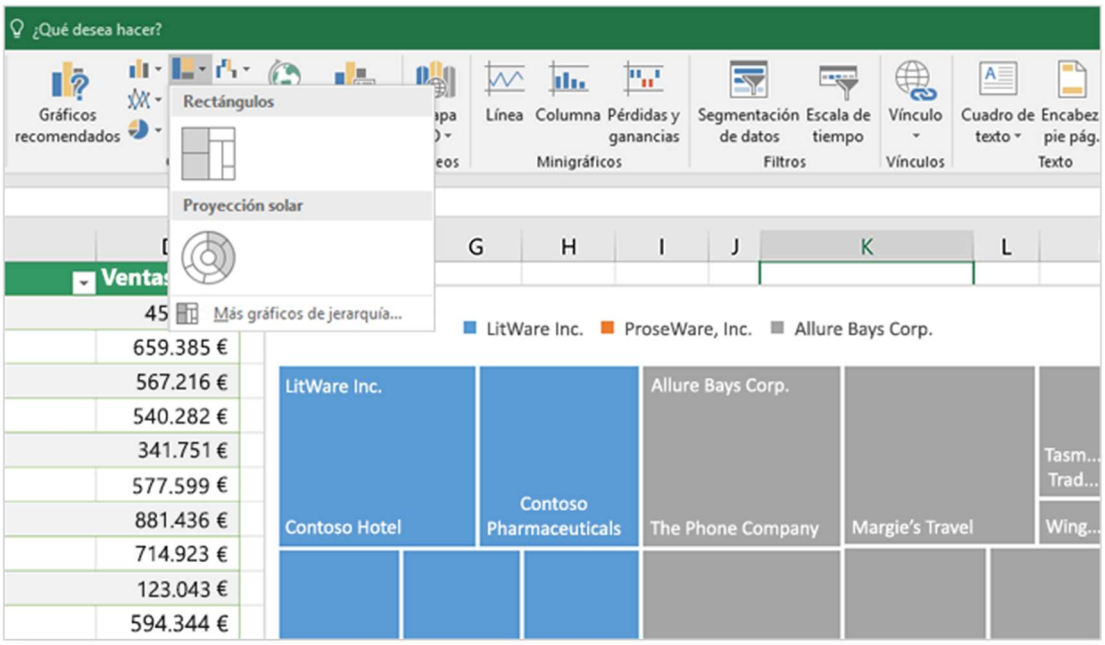

Figura 2.6 Microsoft Excel

Tableau

Tableau es un software que combina un enfoque totalmente centrado en cómo las personas ven y comprenden los datos con una plataforma robusta y escalable, válida incluso para las organizaciones más grandes del mundo. Descubra cómo la plataforma de Tableau lo ayuda a convertir sus datos en información útil, a la vez que hace feliz al equipo de TI.

Esta herramienta está desarrollada por Tableau Software, una empresa de software con su sede principal en Seattle, Estados Unidos, la cual desarrolla productos de visualización de datos interactivos que se enfocan en inteligencia empresarial.

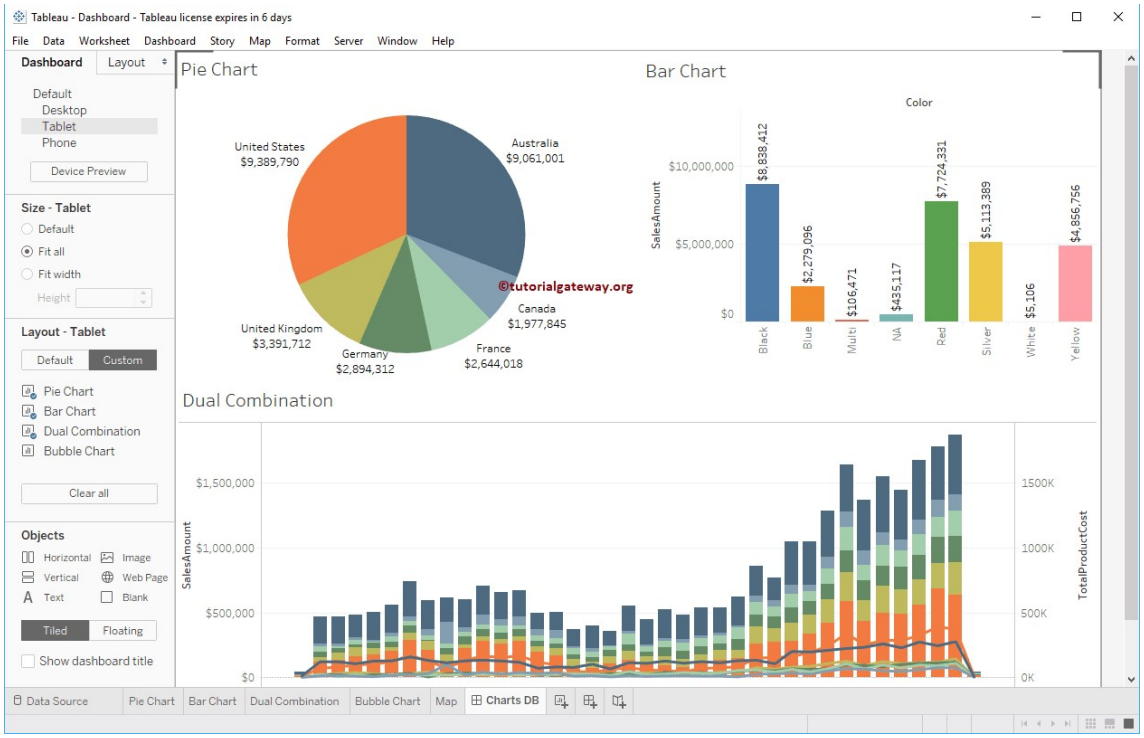

Figura 2.7 Cuadro de mandos de la herramienta Tabbleau para el análisis de datos.

2.2 Ejemplos de fuentes de datos en APIs públicas

El problema de extraer datos de una API es que se necesita un software (programa) o alguna herramienta para la captura de información. Por ejemplo, mediante librerías que funciones bajo algunos programas o lenguajes.

A continuación, se presentan algunos ejemplos de packages (librerías) en R para acceder a algunas APIs de entidades públicas:

Acceso a series del U.S. Bureau of Labor Statistics (API BLS)

Esta librería permite a los usuarios solicitar datos para una o varias series de empleo en USA a través de la API BLS (API del U.S. Bureau of Labor Statistics, en español, Oficina de Estadísticas Laborales de EE. UU.). Esta API[12] se muestra en la Figura 2.1.

[12] https://www.bls.gov/developers/api_r.htm

Los usuarios proporcionan los parámetros especificados en su portal web y la función devuelve una cadena JSON.

Figura 2.8 Web de U.S. Bureau of Labor Statistics

Acceso a USDA National Agricultural Statistics Service.

Esta librería permite descargar los datos de cosechas del Servicio Nacional de Estadísticas Agrícolas (NASS) del USDA para un estado específico. La documentación de esta librería se puede consultar en CRAN[10].

Las utilidades para fips, abreviatura y conversión de nombre también se proporcionan. La funcionalidad completa requiere una conexión a Internet, pero los conjuntos de datos se pueden almacenar en caché para su uso posterior fuera de línea.

Acceso a datos del archivo UK National River Flow Archive

Conjunto de funciones para recuperar datos del archivo National River Flow Archive del Reino Unido.

Este package en R, contiene envoltorios R para la API temporal de datos NRFA del RU. Hay funciones para recuperar estaciones que caen en un cuadro delimitador, para generar un mapa y extraer series temporales e información general.

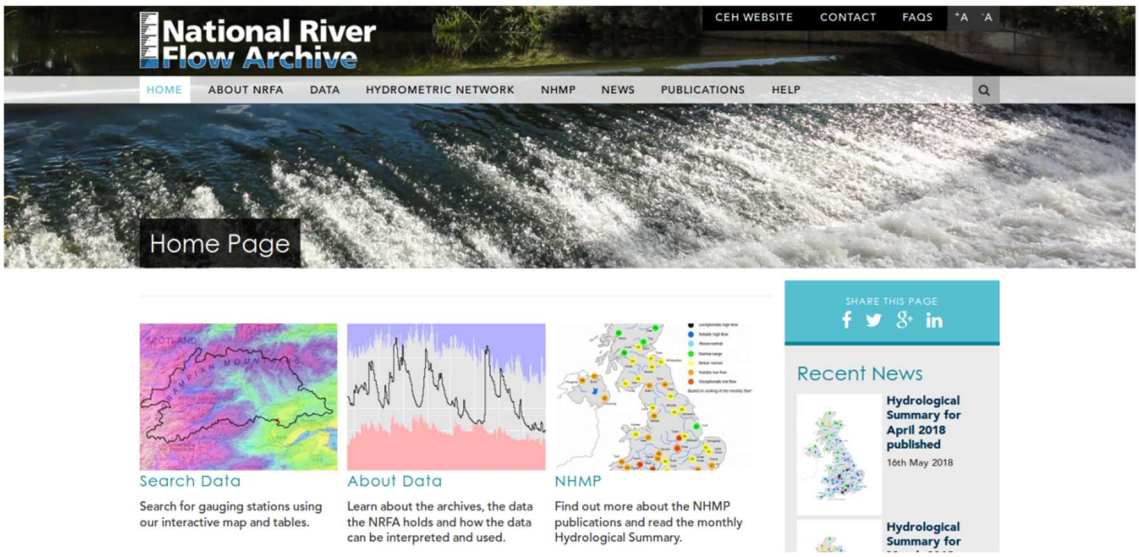

Figura 2.9 Web del UK National River Flow Archive[13]

Acceso a datos hidrológicos diarios de los servicios web de USGS (U.S. Geological Survey)

Esta librería permite importar datos hidrológicos del Servicio Geológico de EE. UU. (USGS) de los servicios web de USGS, traza los datos, resuelve algunos problemas de datos comunes y calcula y traza anomalías. La documentación de esta librería está disponible en CRAN[11].

[13] https://cran.rstudio.com/web/packages/rnrfa/index.html

Acceso a conjuntos de datos financieros y económicos (API Quandl)

Librería con funciones para interactuar directamente con la API de Quandl, para ofrecer datos económicos (por ejemplo, de London Stock Exchange Prices) en varios formatos utilizables en R, descargando un archivo zip con todos los datos de una base de datos Quandl y la capacidad de búsqueda. Este paquete R (Quandl [12]) usa la API Quandl[14].

A continuación, destacamos los siguientes ejemplos de APIs de organismos públicos o entidades gubernamentales:

Acceso a datos abiertos meteorológicos publicados por la Agencia Estatal de Meteorología (AEMET)

La Agencia Estatal de Meteorología es una Agencia Estatal de España, cuyo objetivo básico es la prestación de servicios meteorológicos, que sean competencia del Estado.

AEMET OpenData[15] es una API REST desarrollada por AEMET que permite la difusión y la reutilización de la información meteorológica y climatológica de la Agencia.

14 https://www.quandl.com/docs/api
15 https://opendata.aemet.es/

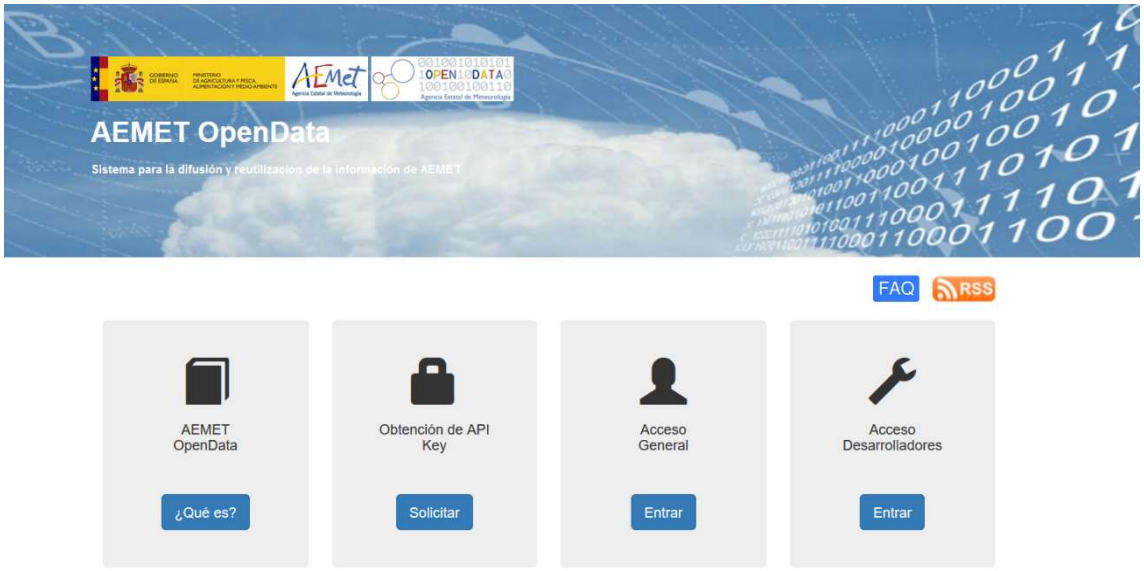

Figura 2.10 Portal de datos abiertos publicados por la AEMET

Acceso a datos abiertos estadísticos publicados por el Instituto Canario de Estadística (ISTAC)

El Instituto Canario de Estadística (ISTAC), es el órgano central del sistema estadístico autonómico y centro oficial de investigación del Gobierno de Canarias de Estadística de la Comunidad Autónoma de Canarias (España).

Es un organismo autónomo dependiente de la Consejería de Economía y Hacienda del Gobierno de Canarias y es de carácter administrativo.

El ISTAC tiene a disposición del ciudadano de forma abierta y gratuita un catálogo de APIs[16] donde se publican:

- Operaciones estadísticas
- Recursos estructurales
- Metadatos comunes
- Recursos estadísticos
- Indicadores

[16] http://www.gobiernodecanarias.org/istac/api/

Acceso a datos abiertos estadísticos publicados por el Instituto Nacional de Estadística (INE)

El Instituto Nacional de Estadística (INE) es un organismo autónomo de España encargado de la coordinación general de los servicios estadísticos de la Administración General del Estado y la vigilancia, control y supervisión de los procedimientos técnicos de los mismos.

El Instituto Nacional de Estadística tiene a disposición del ciudadano una API con datos abiertos[17].

Se detallan algunos ejemplos de librerías de R para acceder a algunas APIs de entidades comerciales o privadas:

Acceso a series de tiempo o marcos de datos en DataMarket.com

Esta librería permite obtener datos de DataMarket.com, ya sea como series temporales en forma de zoológico (dmseries) o como marcos de datos de formato largo (dmlist).

Los metadatos, incluida la estructura de dimensiones, se obtienen con dminfo o solo las dimensiones con dmdims. Esta librería está disponible en CRAN[13].

Acceso a API web para información química.

Esta librería permite obtener información química de toda la web. Este paquete interactúa con un conjunto de API web para información química. Esta librería está disponible en CRAN[14].

[17] http://www.ine.es/dyngs/DataLab/manual.html?cid=45

Por último, se detallan algunos ejemplos de librerías en Python que facilitan la extracción de datos abiertos:

Acceso a datos abiertos publicados en data.world

Esta librería en Python (data.world-py[18]) facilita a los usuarios de data.world[19] extraer y trabajar con datos almacenados en data.world.

Además, la biblioteca proporciona envoltorios convenientes para las API de data.world, lo que permite a los usuarios crear y actualizar conjuntos de datos, agregar y modificar archivos, etc., y posiblemente implementar aplicaciones completas sobre data.world.

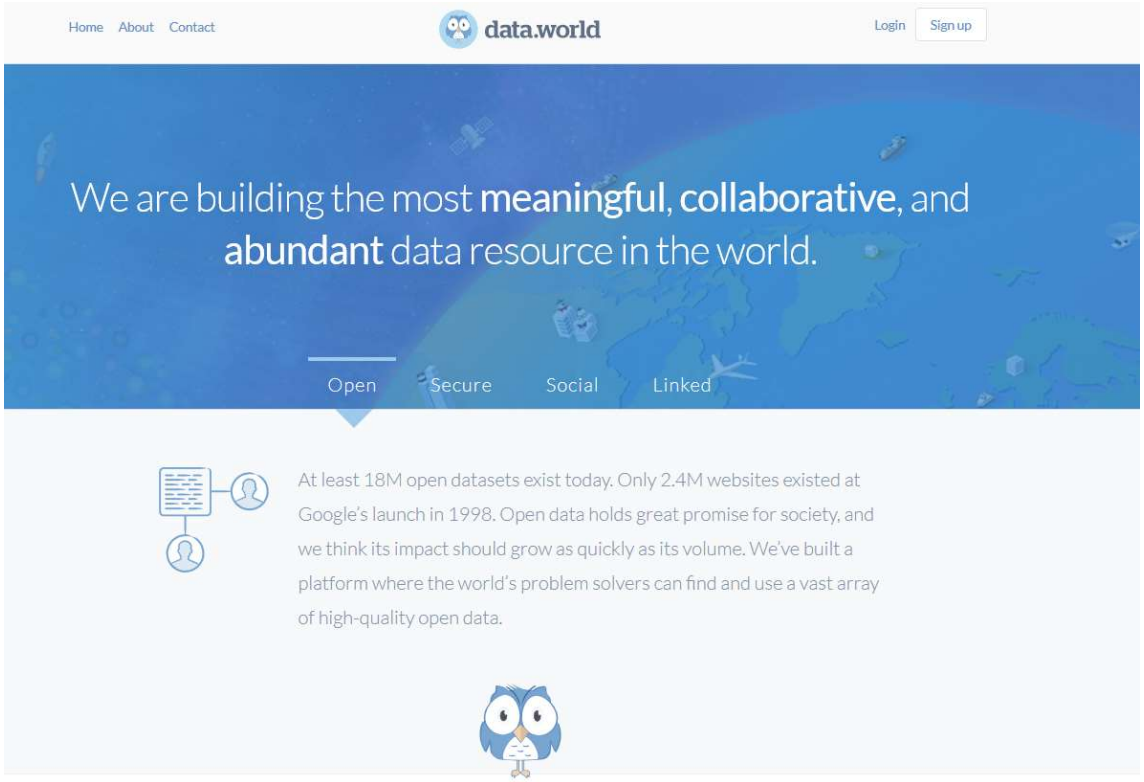

Figura 2.11 Web oficial de data.world

[18] https://data.world/integrations/python
[19] https://data.world/

Acceso a datos abiertos publicados en Socrata

La librería para la extracción desde la API de datos abiertos de Socrata, permite acceder mediante programación a una gran cantidad de recursos de datos abiertos de gobiernos, organizaciones sin fines de lucro y ONG de todo el mundo. Haga clic en el siguiente enlace y pruebe un ejemplo en vivo ahora mismo. Esta librería puede ser descargada desde su repositorio oficial[20].

[20] https://github.com/xmunoz/sodapy

Capítulo 3. Desarrollo de un paquete para R para la explotación de open data.

Debido a la ausencia de una librería en R que analice una fuente de datos estadísticos tan importante como el Instituto Nacional de Estadística (INE), en este proyecto se ha desarrollado una librería en R suficientemente potente para extraer datos de forma estructurada en un tiempo razonable, se ha planteado que esta librería sea bastante sencilla para el usuario de esta.

Nuestra propuesta es una librería que se denomina INEbaseR[15], y cuenta con la ventaja de que en la actualidad no existe algún proyecto similar para la extracción de datos del Instituto Nacional de Estadística (INE).

Una librería similar, es la librería en R (package) denominada istacr[16], que permite la extracción de datos abiertos desde las diferentes APIs del Instituto Canario de Estadística (ISTAC).

Sin embargo, el package INEbaseR, también implementa algunas funciones que permiten algunos análisis de interés mediante la generación de gráficas interactivas, utilizando la librería highcharter[21].

Por otro lado, este es el proceso seguido en el desarrollo de INEbaseR, un proceso recomendado para el desarrollo de packages en R y también sugerido, de forma simular, en el blog Writing an R package from scratch[17]:

1. Librerías externas necesarias.
2. Nombre del package.
3. Estructura de directorios.
4. Principales funciones.
5. Documentación con roxygen2[18, p. 2].
6. Clean Code[19].
7. Valores añadidos.
8. Corrección de errores y pruebas.
9. Instalación.

[21] http://jkunst.com/highcharter/

A continuación, se detallan los distintos pasos sugeridos en el proceso del desarrollo de un package en R para la extracción de datos abiertos (open data) desde una API.

3.1 Librerías externas necesarias

El desarrollo de un package se basa en utilizar un mínimo de dependencias de librerías externas para su mejor mantenimiento.

Estas dependencias pueden ser:

- **jsonlite**: para la extracción de datos en formato JSON.
- **highcharter**: para la representación de gráficos interactivos.

Package: jsonlite

jsonlite es un analizador rápido de datos en formato JSON y un generador optimizado para datos estadísticos y web.

Este proyecto comenzó como un fork del package RJSONIO, pero ha sido completamente reescrito en versiones recientes.

El package ofrece herramientas flexibles, robustas y de alto rendimiento para trabajar con JSON en R, y es particularmente potente para construir tuberías e interactuar con una API web.

Además de convertir datos JSON de/a objetos R, 'jsonlite' contiene funciones para transmitir, validar y embellecer (dar un formato más legible) datos JSON.

Para la extracción de datos en formato JSON existen otros paquetes similares en R, como RJSONIO o rjson muy similares a jsonlite. Curiosamente, los tres paquetes contienen las dos mismas funciones, fromJSON y toJSON, con pequeñas diferencias de implementación entre ellos.

La mayor diferencia entre jsonlite y el resto es que su función fromJSON proporciona la opción de simplificar la lista que produce en algunos casos para construir

una tabla u otras estructuras de datos cuando detecta que es posible. Esto puede ser ventajoso determinadas circunstancias.

Package: highcharter

highcharter es un wrapper de la librería Highcharts que incluye funciones de acceso directo para trazar objetos R.

La librería original Highcharts, es una librería de gráficos implementada en el lenguaje de programación JavaScript en 2009 por la empresa Highsoft[22], y ofrece numerosos tipos de gráficos con una sintaxis de configuración simple.

3.2 Nombre del package

A la hora de elegir un nombre a un package en R hay que tener en cuenta una serie de cuestiones:

- El nombre solo puede constar de letras.
- No se puede usar guiones ni guiones bajos, es decir, "-" o "_", en el nombre del package.

Estas son algunas estrategias, que se recomiendan a la hora de elegir un nombre:

- Elige un nombre único, que puedas fácilmente buscar en Google. Esto facilita que los usuarios potenciales encuentren nuestro package.

- Verifica si ya se utiliza un nombre igual o similar en CRAN entrando en:

```
http://cran.r-project.org/web/packages/[PACKAGE_NAME]
```

- Evita el uso de letras mayúsculas y minúsculas (mezcladas): hacerlo hace que el nombre del paquete sea difícil de escribir e incluso más difícil de recordar. Por ejemplo, es difícil de recordar si es Rgtk2 o RGTK2 o RGtk2.

[22] https://www.highcharts.com/about/

Encuentra una palabra que identifique el problema y modifícalo para que sea único:

- plyr: es una generalización de la familia de aplicaciones y evoca pliers (alicates).
- lubridate: hace que las fechas y los tiempos sean más fáciles.
- testdat: prueba que los datos tienen el formato correcto.

Utiliza abreviaciones:

- Rcpp = R + C++ (C plus plus)
- lvplot = diagramas de valores de letras.

Añade una R adicional:

- stringr: proporciona herramientas de cadena.
- Tourr: implementa grandes giras (un método de visualización).

3.3 Estructura de directorios

Principalmente hay dos formas de crear la estructura de directorios y ficheros base de un package en R:

- Con package.skeleton()
- Con RStudio.

3.3.1 Creación de package con package.skeleton()

Package.skeleton es una función de R que automatiza parte de la creación de un nuevo package.

Crea directorios, guarda funciones, datos y archivos de código R en lugares apropiados, crea esqueletos de archivos de ayuda y crea un archivo "léeme y bórrame" que describe más pasos en el empaquetado.

Un ejemplo de utilización de esta función para la creación de un package en R podría ser:

```
package.skeleton(
      name = "anRpackage",
      list,
      environment = .GlobalEnv,
      path = ".",
      force = FALSE,
      code_files = character(),
      encoding = "UTF-8"
)
```

Esta función tiene los siguientes argumentos que pueden ser utilizados por el usuario a la hora de crear el package:

- **name**: nombre del package.
- **list**: vector (o lista) de objetos de R para incluir en el package (principalmente funciones).
- **environment**: entorno dónde buscar los objetos que se quieren incluir (por ejemplo: .GlobalEnv)
- **path**: ruta donde crear el package.
- **force**: si su valor es TRUE (verdadero), sobrescribe el directorio existente (si lo hubiera).
- **code_files**: vector de caracteres con las rutas a los archivos de código R para construir el package.
- **encoding**: codificación (típicamente se utiliza: "latin1", "latin2" o "UTF-8").

3.3.2 Creación de package con RStudio

Lo más recomendado es utilizar el IDE RStudio para crear la estructura de directorios y ficheros del package.

Esto se puede hacer siguiendo estos pasos:

Crear nuevo proyecto: File > New project...

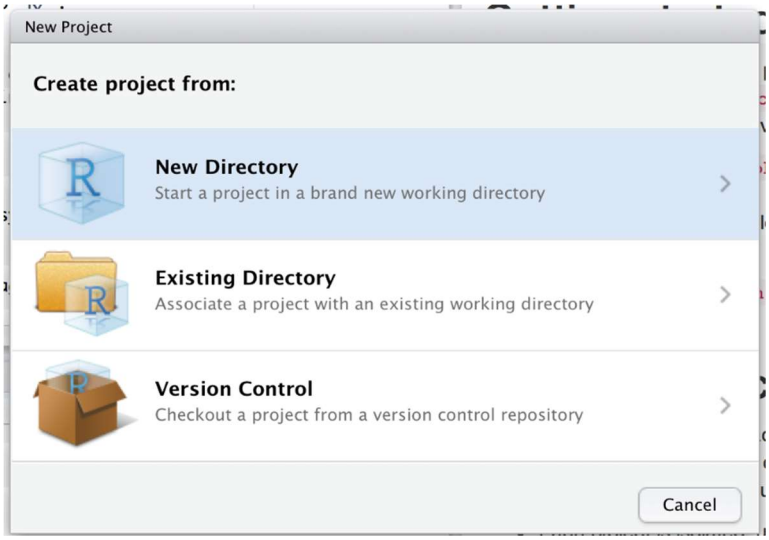

Figura 3.1 Crear nuevo package de R con RStudio (paso 1)

Crear nuevo package en R: New project > R package

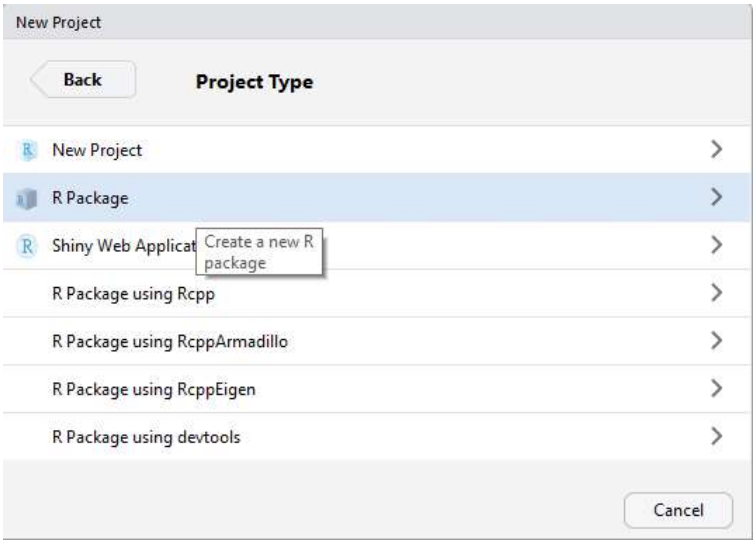

Figura 3.2 Crear nuevo package de R con RStudio (paso 2)

Introducir datos sobre nuestro package:

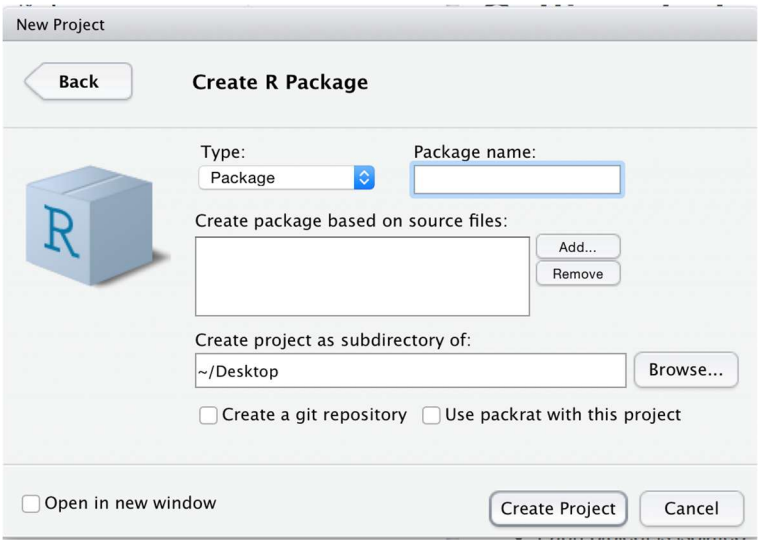

Figura 3.3 Crear nuevo package de R con RStudio (paso 3)

Una vez presionado el botón “Create project”, se creará en el espacio de trabajo del usuario los directorios y archivos básicos para el desarrollo de un package.

3.4 Principales funciones

Durante el desarrollo de un package se desarrollan los diversos métodos, procedimientos y funciones para la extracción de datos siguiendo una metodología de Clean Code (que se detallan en la Sección 3.6)

Para el caso de la extracción de datos abiertos desde una API, las principales funciones que el package debe contener, deben de permitir al menos las mismas funcionalidades que permite la propia API.

Hay que tener en cuenta también, que las funciones del package deben contemplar que estas puedan recibir parámetros obligatorios y opcionales igual que permita la API.

A partir de este punto, se pueden añadir nuevas funciones que permitan crear valores añadidos, tal y como se detallarán en la sección 3.7.

3.5 Documentación con roxygen2

La documentación es uno de los aspectos más importantes del buen código. Sin ella, los usuarios no podrán saber cómo usar nuestro package, y es poco probable que lo hagan.

La documentación también es útil para el desarrollador en el futuro (para recordar qué y cómo has programado tu código) y para otros desarrolladores que trabajan en tu package.

El objetivo de roxygen2 es hacer que documentar su código sea lo más fácil posible. R proporciona una forma estándar de documentación de paquetes y escribe archivos .Rd en el directorio man /. Estos archivos usan una sintaxis personalizada, basada en LaTeX. Roxygen2 proporciona una serie de ventajas sobre la escritura de archivos .Rd a mano:

- El código y la documentación son adyacentes, por lo que cuando modifica su código, es fácil recordar que necesita actualizar la documentación.

- Roxygen2 inspecciona dinámicamente los objetos que está documentando, por lo que puede agregar automáticamente datos que de otro modo tendría que escribir a mano.

- Resume las diferencias en la documentación de los métodos S3 y S4, los genéricos y las clases, por lo que debe aprender menos detalles.

Además de generar archivos .Rd, roxygen2 también crea un NAMESPACE único, y define los principales campos del fichero DESCRIPTION.

3.6 Clean Code

Para un proyecto de código abierto, cuyo código podrá ser consultado y/o mejorado por otros desarrolladores, se recomienda seguir unas buenas prácticas de código limpio (Clean Code).

El término Clean Code hace referencia a un código limpio, legible, claro (que se entienda), que sea fácil de leer y de interpretar. Clean Code es un código que es fácil de extender o adaptarse a cambios. Cumple generalmente características como clases y métodos pequeños, manejables, etc.

Según varios autores, las principales características del código limpio son:

- Fácil de leer, expresivo y sencillo.
- Probado mediante test automáticos.
- Hace una única cosa.
- No existe duplicidad en él.
- Utiliza el menor número de elementos posibles.
- Realiza abstracciones de elementos similares.

Estas son algunas de las características que se han de tener en cuenta a la hora de desarrollar el package:

- Un buen nombre da mucha más información que cualquier otra cosa. Para conseguir buenos nombres hay que usar nombres descriptivos y claros. Deben ser legibles y evitar codificaciones complejas.

- Una buena función es aquella de la que se puede inferir su comportamiento de un solo vistazo. Para ello deben ser cortas, hacer una única cosa y mantenerse dentro del mismo nivel de abstracción.

- Los comentarios no pueden maquillar el mal código. La necesidad de comentarios para aclarar algo es síntoma de que hay código mal escrito que debería ser rediseñado. Es preferible expresarse mediante el propio código.

- El formateo afecta directamente a la legibilidad del código. El código se lee de arriba abajo y de izquierda a derecha. Los espacios verticales y horizontales permiten separar ideas y conceptos distintos.

- Hay que definir de forma clara la frontera entre el código y el exterior para poder acomodar de forma sencilla los futuros cambios, minimizando las partes de nuestro código que dependan de elementos externos.

Siguiendo estas prácticas el autor del libro "Clean Code: A Handbook of Agile Software Craftsmanship"[19] de Robert C. Martin, nos promete conseguir un código con el que será más fácil trabajar y hará mucho menos frustrante nuestro día a día.

3.7 Valores añadidos

Continuando con lo que se comentaba en el apartado 3.4 Principales funciones de esta sección, una vez se hayan desarrollado las principales funciones del package, es recomendado desarrollar nuevas funcionalidades que aporten un valor añadido a este package.

Entre los valores añadidos que puedes realizar, podrían destacarse algunos, como los que se detallan a continuación:

- Mejora de carencias que se encuentren en la API, como, por ejemplo, facilitar consultas al usuario con lenguaje más cercano.
- Optimización de tiempos de carga de datos mediante la utilización de datos almacenados en ficheros locales .RData (caché).

- Nuevas funciones que filtren o hagan una unión de datos que sirvan para realizar comparativas.
- Funciones que permitan la representación de datos en plots para su mejor análisis.
- En general, cualquier funcionalidad que mejore el uso de los datos extraídos de la API al usuario.

3.8 Corrección de errores y pruebas

Es importante realizar una serie de pruebas de funcionalidades o test de código para que tu package en R tenga el menor número de bugs (errores) posibles.

Para la corrección de sintaxis de código, el IDE RStudio incorpora opciones que permiten al desarrollador detectar estos errores de forma muy sencilla.

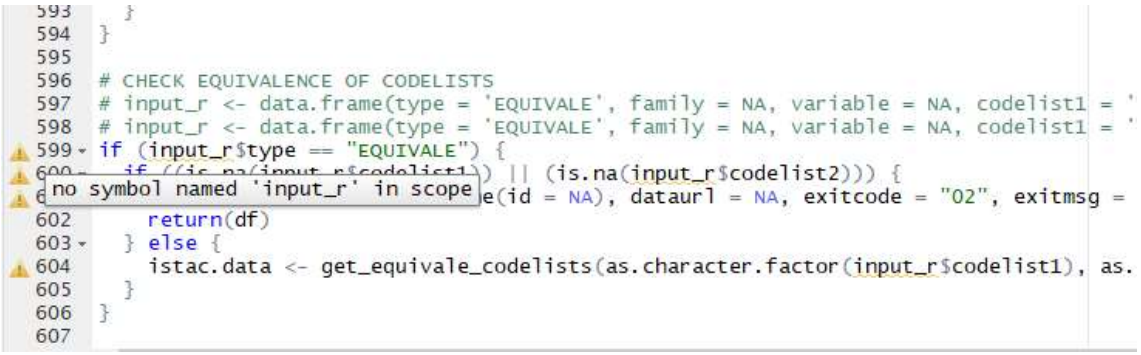

Figura 3.4 Corrección de sintaxis de R con RStudio

3.9 Instalación

Finalmente, el usuario podrá descargar e instalar el package en R de diferentes formas:

- Desde el repositorio CRAN

Si se trata de un package en R localizado en el repositorio CRAN (The Comprehensive R Archive Network), se puede instalar mediante el comando:

```
install.package("nombre_del_package")
```

- Desde el repositorio GitHub[23]

Si por el contrario se trata de un package en R localizado en un repositorio de GitHub, se puede instalar mediante el comando:

```
devtools::install_github("nombre_usuario/nombre_repositorio")
```

[23] https://github.com/

Capítulo 4. Análisis de la API del Instituto Nacional de Estadística (INE)

Es necesario comprender cómo funciona la API del Instituto Nacional de Estadística (INE) para saber qué métodos y funciones necesitamos desarrollar en nuestro package.

La API proporciona información que el INE produce, a destacar las estadísticas sobre la demografía, economía, y sociedades españolas. A través de la página web oficial se pueden seguir todas las actualizaciones de los distintos trabajos y estudios.

El sistema de información del INE se presenta en 2 bloques:

- INEbase (que se accede vía web[24])
- API JSON (que es el sistema para acceder a la base de datos de difusión del Instituto Nacional de Estadística: Tempus3[25]).

Esta API tiene distintos apartados dirigidos a extraer datos y metadatos de series (entre otras) que estudiaremos en el siguiente apartado.

4.1 INEbase

INEbase es un portal web temático, con la información estructurada por apartados, con enlaces a los distintos bloques o archivos de datos.

¿Qué es INEbase?

INEbase es el sistema que utiliza el Instituto Nacional de Estadística (INE) para el almacenamiento de la información estadística en internet. Contiene toda la información que el INE produce en formatos electrónicos.

La organización primaria de la información sigue la clasificación temática del Inventario de Operaciones Estadísticas de la Administración General del Estado (IOE).

[24] http://www.ine.es/dyngs/INEbase/listaoperaciones.htm
[25] http://www.ine.es/dyngs/DataLab/manual.html?cid=45

Aunque la unidad básica de INEbase es la operación estadística tal y como se recoge en el IOE, en algunos casos se han agrupado diferentes operaciones en una única entrada para facilitar el acceso a la información.

Organización de la información

A las operaciones estadísticas se puede acceder directamente a través de la lista completa de operaciones de INEbase o a través de los menús temáticos. Estos menús permiten conocer toda la información disponible de cada tema.

Para las operaciones que son competencia del INE, tanto las que se realizan de forma periódica como que no tienen una periodicidad fija o se han dejado de elaborar, se presenta junto con el acceso a los resultados una pequeña descripción de los objetivos, variables estudiadas, periodicidad y ámbito geográfico, así como la referencia del último dato publicado.

Para las operaciones realizadas por otros organismos del Sistema estadístico nacional, se facilita tanto el enlace a la ficha correspondiente en el IOE como el acceso a la web del organismo donde pueden consultarse los resultados.

Para cada operación estadística en INEbase existe una página que da acceso a toda la información relativa a la misma: los últimos datos, los resultados detallados, la última nota de prensa publicada, el calendario de disponibilidad de datos y toda la información metodológica o descriptiva que ayuda a la mejor comprensión e interpretación de los datos (metodologías, cuestionarios, clasificaciones, notas explicativas, etc.).

En INEbase los datos sólo se obtienen en ficheros físicos en formatos conocidos: Excel, CSV y Pc-Axis.

4.2 El sistema API JSON (tempus 3) del INE

El servicio API JSON (Javascript Object Notation) INE permite acceder mediante peticiones URL a toda la información disponible en la base de datos Tempus3 de difusión de resultados del INE. La estructura de dichas peticiones y la simplicidad del formato JSON hacen que este tipo de servicios sean ampliamente utilizadas para

ofrecer datos y metadatos que permiten la explotación automática de la información estadística.

Conceptos previos: operaciones y series

Antes de comenzar a definir cómo funciona la API JSON del INE, es necesario introducir los siguientes conceptos estadísticos:

- **Operaciones**: son un conjunto de actividades que comprenden el diseño, recolección, organización, procesamiento, análisis, presentación y divulgación de los resultados estadísticos sobre una determinada área o tema de la realidad nacional.

- **Series**: es un conjunto de observaciones medidas en determinados momentos del tiempo, ordenados cronológicamente y normalmente espaciados entre sí de manera uniforme.

Arquitectura del sistema de API-JSON

La información en el sistema se estructura en torno a la serie temporal, como elemento principal de Tempus3, y se organiza de la siguiente forma:

Figura 4.1 Elementos principales de Tempus3.

4.2.1 Variables

Una variable es una característica que puede fluctuar y cuya variación es susceptible de adoptar diferentes valores.

Las variables o listas de valores contenidas en Tempus3 y utilizadas en la difusión, son comunes a todas las operaciones estadísticas, es decir, no están duplicadas en el sistema, su identificador en Tempus3 (Id) y sus descriptores son únicos.

Uno de estos descriptores es el campo código que contiene, en el caso de existir, el identificador estándar de la variable. Es conveniente que las definiciones y codificaciones de ciertas variables estadísticas sean siempre las mismas.

Son ejemplos de variables las listas: Grupos ECOICOP, Sexo, CCAA, Provincias, etc.

Figura 4.2 Estructura de una variable en Tempus3.

Se pueden obtener las variables de una operación siguiendo el siguiente proceso:

1. **Obtener las operaciones disponibles.**

La base de datos Tempus3 contiene la información de todas las operaciones estadísticas coyunturales del INE, aquellas cuya periodicidad de difusión de resultados es inferior al año, además de algunas operaciones estadísticas estructurales. La relación de operaciones en Tempus3 cambia a medida que se van integrando en la base de datos.

Se pueden consultar todas las operaciones disponibles mediante la siguiente URL:

```
http://servicios.ine.es/wstempus/js/ES/OPERACIONES_DISPONIBLES
```

2. **Obtener las variables de una operación.**

Se pueden obtener las variables de una operación, por ejemplo, de la operación IPC, mediante la siguiente URL:

```
http://servicios.ine.es/wstempus/js/ES/VARIABLES_OPERACION/IPC
```


```
0:
   Id:      3
   Nombre:  "Tipo de dato"
   Codigo:  ""
1:
   Id:      70
   Nombre:  "Comunidades y Ciudades Autónomas"
   Codigo:  "CCAA"
2:
   Id:      115
   Nombre:  "Provincias"
   Codigo:  "PROV"
3:
   Id:      269
   Nombre:  "Grupos especiales 2001"
   Codigo:  ""
4:
   Id:      270
   Nombre:  "Rúbricas 2001"
   Codigo:  ""
```


Figura 4.3 Variables de la operación IPC en la API JSON del INE.

4.2.2 Valores

Los valores son los estados que puede presentar una variable determinada. Por ejemplo, la variable Provincias contiene los valores: Áraba/Álava, Albacete, Alicante/Alacant, etc.

Los valores contenidos en Tempus3 y utilizados en la difusión, son comunes a todas las operaciones estadísticas, es decir, no están duplicadas en el sistema, su identificador en Tempus3 (Id) y sus descriptores son únicos.

Figura 4.4 Estructura del esquema variable-valor en Tempus3.

Los valores de una variable se pueden obtener siguiendo el proceso de la sección anterior, y luego obteniendo los valores de la variable elegida:

Operaciones > Variables de una operación > Valores de una variable.

En el caso que queramos obtener los valores de la variable "Provincia" disponible en la operación "IPC", podemos consultar la siguiente URL de la API:

`http://servicios.ine.es/wstempus/js/ES/VALORES_VARIABLE/115`


```
▼0:
    Id:           2
    Fk_Variable:  115
    Nombre:       "Araba/Álava"
    Codigo:       "01"
▼1:
    Id:           3
    Fk_Variable:  115
    Nombre:       "Albacete"
    Codigo:       "02"
▼2:
    Id:           4
    Fk_Variable:  115
    Nombre:       "Alicante/Alacant"
    Codigo:       "03"
▼3:
    Id:           5
    Fk_Variable:  115
    Nombre:       "Almería"
    Codigo:       "04"
```


Figura 4.5 Valores de la variable "Provincias" en la API JSON del INE.

4.2.3 Series temporales

Son un conjunto de observaciones medidas en determinados momentos del tiempo, ordenados cronológicamente y normalmente espaciados entre sí de manera uniforme.

Como entidad principal de la base de datos Tempus3, la serie tiene unas propiedades que la definen y que no cambian a lo largo del tiempo: Identificador único y características de la serie: id, nombre, periodicidad, escala, unidad, clasificación, decimales, etc.

Figura 4.6 Propiedades que definen una serie temporal en Tempus3.

Pero un literal o un id nos dice poco, así que, para dotar a la serie de significado, necesitamos definirla por una combinación de variables-valores, lo que llamaremos los metadatos de la serie.

La serie "Variación mensual del IPC en Andalucía" es el resultado del cruce de los valores Total (de la variable Grupos COICOP), Andalucía (de la variable CCAA), variación mensual (de la variable Tipo de dato), etc.

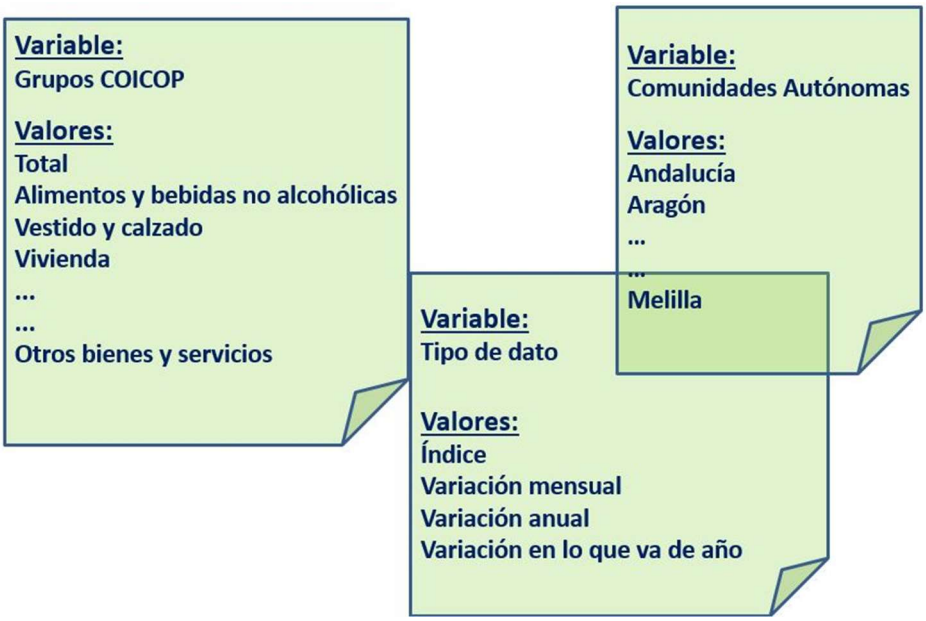

Figura 4.7 Ejemplo de serie temporal en Tempus3.

Otras distribuciones de los datos que se pueden obtener de la API son las tablas y las fechas de publicación:

Tablas

Una tabla es el resultado del cruce de grupos de valores contenidos en una o varias variables, es decir, son una agrupación de series temporales definidas por estos grupos.

Fechas de publicación

Cada operación tiene asociada una o varias publicaciones según sus diferentes periodicidades, por ejemplo, hay dos publicaciones para el IPC, una con periodicidad mensual y otra anual. Y éstas siguen el calendario de publicaciones del INE.

De esta manera, una publicación contiene los momentos en los que se publican los datos de una operación estadística: fechas de publicación.

4.3 Construcción de URLs

A continuación, se detallan las principales formas de obtención de metadatos, sin perder la referencia de la construcción de las URL:

```
http://servicios.ine.es/wstempus/js/{idioma}/{función}/{input}[?parámetros]
```

4.3.1 Operaciones estadísticas

Puede consultarse la lista de operaciones disponibles en cualquier momento a través de las siguientes funciones:

- OPERACIONES_DISPONIBLES
- OPERACION

Operaciones estadísticas disponibles del sistema

Todas las operaciones estadísticas disponibles.

- {URL} = http://servicios.ine.es/wstempus/js/ES/OPERACIONES_DISPONIBLES
- {función} = OPERACIONES_DISPONIBLES
- {input} = ninguno
- {output} = Se obtienen los identificadores del elemento operación estadística. Existen tres códigos para la identificación de la operación estadística "Índice de Precios de Consumo (IPC)":
    - código numérico Tempus3 interno (Id=25)
    - código de la operación estadística en el Inventario de Operaciones Estadísticas (IOE30138)
    - código alfabético Tempus3 interno (IPC)

4.3.2 Series

Serie

Por ejemplo, la serie IPC206449, recoge la variación mensual del Índice de precios de consumo. Base 2016.

- {URL} = http://servicios.ine.es/wstempus/js/ES/SERIE/IPC206449
- {función} = SERIE
- {input} = código identificativo de la serie (COD=IPC206449).
- {output} = Información de la serie: identificadores Tempus3 de la serie, objeto Tempus3 operación, nombre de la serie, número de decimales que se van a visualizar para los datos de esa serie, objeto Tempus3 periodicidad, objeto Tempus3 publicación, PubFechaAct * dentro de la publicación, objeto Tempsu3 clasificación *, objeto Tempus3 escala y objeto Tempus3 unidad.
- [?parámetros]= posibilidad de usar det=2 para ver dos niveles de detalle, en concreto para poder acceder al objeto PubFechaAct y tip=M para obtener los metadatos (cruce variables-valores) de la serie.

Series de una operación

Series pertenecientes a la operación IPC.

- {URL} = http://servicios.ine.es/wstempus/js/ES/SERIES_OPERACION/IPC?page=1
- {función} = SERIES_OPERACION
- {input} = código identificativo de la operación (IOE30138 / IPC / 25)
- {output} = Información de las series: identificadores Tempus3 de la serie, identificador Tempus3 de la operación, nombre de la serie, número de decimales que se van a visualizar para los datos de esa serie, identificador Tempus3 de la periodicidad, identificador Tempus3 de la publicación, identificador Tempsu3 de la clasificación, identificador Tempus3 de la escala e identificador Tempus3 de la unidad.
- [?parámetros]= posibilidad de usar det=2 para ver dos niveles de detalle, en contreto para poder acceder al objeto PubFechaAct y tip=M para obtener los metadatos (cruce variables-valores) de la serie.

Metadatos que definen una serie

Definición en términos de metadatos (variables y valores) de la serie que recoge los datos de la variación mensual del IPC, IPC206449.

- {URL} = http://servicios.ine.es/wstempus/js/ES/VALORES_SERIE/IPC206449
- {función} = VALORES_SERIE
- {input} = Código identificativo de la serie (IPC206449)
- {output} = Información de los metadatos que definen a la serie: identificador Tempus3 del valor, identificador Tempus3 de la variable a la que pertenece, nombre del valor y código oficial del valor.
- [?parámetros] = posibilidad de usar det=1 para acceder al objeto variable.

Series mediante cruce de metadatos

Series mensuales de la operación IPC cuya definición en términos de metadatos (variables y valores) cumple lo siguiente:

- "Provincias" = "Madrid"
- "Tipo de dato" = "Variación mensual"
- "Grupos ECOICOP" = Todos los grupos

La consulta es la siguiente:

- {URL} = http://servicios.ine.es/wstempus/js/ES/SERIE_METADATAOPERACION/IPC?g1=115:29&g2=3:84&g3=762:&p=1
- {función} = SERIE_METADATAOPERACION
- {input} = Código identificativo de la operación (IOE30138 /IPC/ 25) y códigos identificativos de las variables y valores:
    - "Provincias" (FK_VARIABLE=115) = "Madrid" (FK_VALOR=29) ⇒ g1=115:29
    - "Tipo de dato" (FK_VARIABLE=3) = "Variación mensual" (FK_VALOR=84) ⇒ g2=3:84
    - "Grupos ECOICOP" (FK_VARIABLE=762) = "Todos los grupos ECOICOP" (FK_VALOR=null) ⇒ g3=762:

    - "Serie mensual" (FK_PERIODICIDAD=1) ⇒ p=1 (Ver PUBLICACIONES_OPERACION)
- {output} = Información de las series cuya definición de metadatos cumple los criterios establecidos: identificadores Tempus3 de la serie, identificador Tempus3 de la operación, nombre de la serie, número de decimales que se van a visualizar para los datos de esa serie, identificador Tempus3 de la periodicidad, identificador Tempus3 de la publicación, identificador Tempsu3 de la clasificación, identificador Tempus3 de la escala e identificador Tempus3 de la unidad..
- [?parámetros] = posibilidad de usar det=2 para ver dos niveles de detalle, en contreto para poder acceder al objeto PubFechaAct y tip=M para obtener los metadatos (cruce variables-valores) de la serie.

Capítulo 5. INEbaseR y algunos ejemplos

INEbaseR es el nombre que recibe el desarrollo del package en R para la extracción y análisis de datos abiertos del Instituto Nacional de Estadística (INE).

Es una librería cuyo principal enfoque es permitir extraer operaciones y series de datos de forma estructurada y eficiente.

Para el desarrollo del package INEbaseR se han utilizado las tecnologías y metodologías que se describen en este apartado.

Entre sus principales características de esta librería, destacan:

- Es rápida.
- Es eficiente.
- Está bien documentada.
- Es versátil para utilizar junto a otras librerías.
- Se integra fácilmente con otras GUIs (por ejemplo, con RSquared project y Jupyter notebook).
- Permite obtener datos de forma estructurada.
- Incorpora funcionalidades para análisis y representación de datos estadísticos.
- Actualmente cuenta con varios interesados en el proyecto (Instituto Nacional de Estadística, Instituto Canario de Estadística y el Instituto de Estadística y Cartografía de Andalucía[26]).
- Ha recibido varios premios.

[26] http://www.juntadeandalucia.es/institutodeestadisticaycartografia/

Obtención de datos estructurados

Desde el servicio API del Instituto Nacional de Estadística (INE) se pueden extraer los datos abiertos en formato JSON, tal y como se muestra en la siguiente figura.

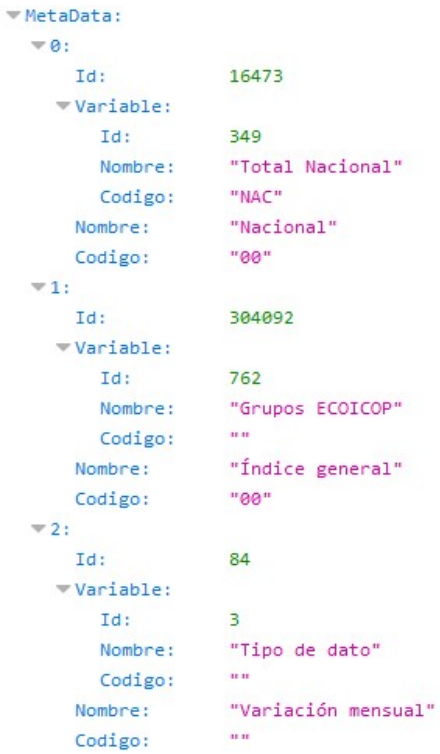

Figura 5.1 Datos abiertos en formato JSON extraídos de la API del INE.

Sin embargo, una de las principales ventajas de INEbaseR es que permite obtener en R los datos que se pueden obtener mediante el acceso a las URLs de la API del INE, pero de forma estructurada.

Filter

	Id	Variable.Id	Variable.Nombre	Variable.Codigo	Nombre	Codigo
1	16473	349	Total Nacional	NAC	Nacional	00
2	304092	762	Grupos ECOICOP		Índice general	00
3	84	3	Tipo de dato		Variación mensual	

Figura 5.2 Datos abiertos en R de forma estructurada extraídos de la API del INE con INEbaseR.

Esto es muy útil para el usuario del package, ya que le permite, por ejemplo, aplicar operaciones estadísticas sobre los datos extraídos de forma muy sencilla.

A continuación, la primera sección describe un procedimiento para la obtención de datos utilizando el package, en la segunda sección de este apartado se detalla una mejora planteada para obtención de series temporales mediante el uso de consultas estructuradas, y en el resto de secciones de este capítulo se definen otros aspectos de este proyecto tales como: documentación del package, utilización de caché de ficheros local, análisis de datos, licencia del package, repositorio donde está disponible el código, redes sociales y los premios y reconocimientos del proyecto.

5.1 Ejemplo de procedimiento de obtención de datos

En esta sección se define un ejemplo del procedimiento a seguir para la obtención de datos extraídos de la API JSON del INE partiendo desde la elección de una operación estadística hasta su representación y análisis.

Con la llamada al método `get_operations_all()` de INEbaseR podremos obtener todas las operaciones estadísticas publicadas en la API JSON del INE.

23	30417	Estadística de Defunciones según la Causa de Muerte	ECM
24	30048	Estadística Estructural de Empresas: Sector Industrial	EIE
25	30138	Índice de Precios de Consumo (IPC)	IPC
26	30050	Índices de Producción Industrial	IPI
27	30051	Índices de Precios Industriales	IPRI

Figura 5.3 Obtención de las operaciones disponibles con INEbaseR.

Suponiendo que elegimos la operación Índice de Precios de Consumo (IPC), ahora podremos obtener las series de esta operación con la llamada a la función `get_series_operation("IPC")` que recibe como parámetro el identificador de la operación.

COD	T3_Operacion	Nombre	Decimales	T3_Periodicidad	T3_Publicacion
IPC251603	Índice de Precios de Consumo (IPC)	Total Nacional. Alimentos y bebidas no alcohólicas...	3	Mensual	Índice de Precios de Consumo
IPC251602	Índice de Precios de Consumo (IPC)	Total Nacional. Bebidas alcohólicas y tabaco. Índice.	3	Mensual	Índice de Precios de Consumo
IPC251601	Índice de Precios de Consumo (IPC)	Total Nacional. Vestido y calzado. Índice.	3	Mensual	Índice de Precios de Consumo
IPC251600	Índice de Precios de Consumo (IPC)	Total Nacional. Vivienda, agua, electricidad, gas y ...	3	Mensual	Índice de Precios de Consumo
IPC251599	Índice de Precios de Consumo (IPC)	Total Nacional. Muebles, artículos del hogar y artíc...	3	Mensual	Índice de Precios de Consumo
IPC251598	Índice de Precios de Consumo (IPC)	Total Nacional. Sanidad. Índice.	3	Mensual	Índice de Precios de Consumo
IPC251597	Índice de Precios de Consumo (IPC)	Total Nacional. Transporte. Índice.	3	Mensual	Índice de Precios de Consumo
IPC251596	Índice de Precios de Consumo (IPC)	Total Nacional. Comunicaciones. Índice.	3	Mensual	Índice de Precios de Consumo
IPC251595	Índice de Precios de Consumo (IPC)	Total Nacional. Ocio y cultura. Índice.	3	Mensual	Índice de Precios de Consumo
IPC251594	Índice de Precios de Consumo (IPC)	Total Nacional. Enseñanza. Índice.	3	Mensual	Índice de Precios de Consumo
IPC251593	Índice de Precios de Consumo (IPC)	Total Nacional. Restaurantes y hoteles. Índice.	3	Mensual	Índice de Precios de Consumo

Figura 5.4 Obtención de las series de la operación IPC con INEbaseR.

Dependiendo de la operación estadística, puede haber cientos de miles de series por cada operación estadística que pueden ralentizar la extracción de estos registros desde la API del INE e incluso, en ocasiones, este proceso puede tardar hasta más de 30 minutos.

Es por esto por lo que se ha desarrollado para este proyecto una caché de ficheros local que permite que esta consulta se realice de forma ágil y eficiente, y la cual se hablará con más detalles en la sección 5.3.2.

Ahora supongamos que elegimos la serie `Total Nacional. Índice general. Variación mensual (IPC206449)` de la operación Índice de Precios de Consumo (IPC). Podemos obtener los datos de esta serie mediante la llamada a la función `get_data_serie("IPC206449", nult = 5)`.

	Fecha	FK_TipoDato	FK_Periodo	Anyo	Valor	Secreto
1	1.512083e+12	1	12	2017	0.0	FALSE
2	1.514761e+12	1	1	2018	-1.1	FALSE
3	1.517440e+12	1	2	2018	0.1	FALSE
4	1.519859e+12	1	3	2018	0.1	FALSE
5	1.522534e+12	1	4	2018	0.8	FALSE

Figura 5.5 Extracción de datos de la serie IPC206449 de la operación IPC con INEbaseR.

Por último, podemos representar estos datos utilizando la función `highcharts_series("IPC206449", nult = 5)`.

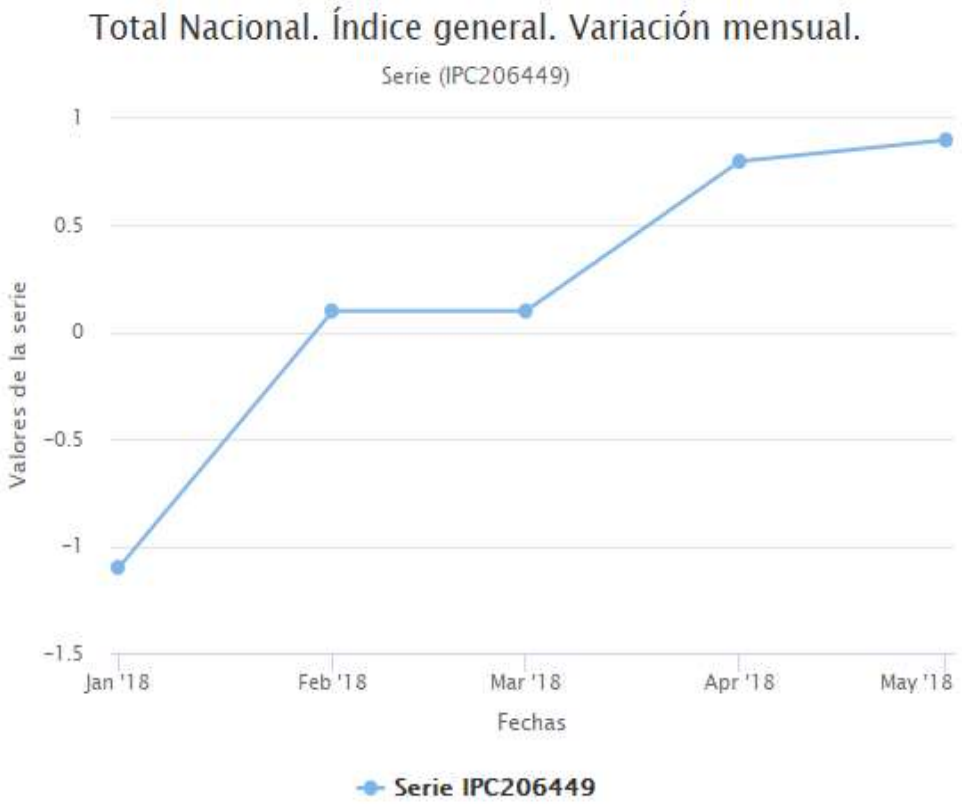

Figura 5.6 Representación de los datos de la serie IPC206449 con highcharter.

Highcharter es una librería utilizada en INEbaseR (dependencia) que permite el interactuar con los datos representados para obtener más información y permitir un análisis de estos. En la sección 3.1 se habla con más detalle sobre esta librería.

5.2 Obtención de las series estadísticas mediante lenguaje de consulta estructurada

Desde la API JSON del Instituto Nacional de Estadística (INE) es posible obtener series mediante el cruce de metadatos utilizando información de las variables y valores utilizadas las series de una operación.

Por ejemplo, para obtener la serie IPC mensual de precios de hoteles y restaurantes en Madrid utiliza es necesario realizar el cruce de metadatos siguiendo el esquema: `variable = valor`.

A continuación, se detalla cómo es el procedimiento actual para resolver esta consulta con los recursos disponibles en la API JSON del INE.

En el caso de este ejemplo, primero hay que realizar una consulta que permita saber las variables disponibles de la operación IPC, esto es posible en INEbaseR mediante la llamada a la función `get_variables_operation("IPC")`.

	Id	Nombre	Codigo
1	3	Tipo de dato	
2	70	Comunidades y Ciudades Autónomas	CCAA
3	115	Provincias	PROV
4	269	Grupos especiales 2001	
5	270	Rúbricas 2001	
6	349	Totales Territoriales	NAC
7	544	Corrección de efectos	
8	762	Grupos ECOICOP	
9	763	Subgrupos ECOICOP	
10	764	Clases ECOICOP	
11	765	Subclases ECOICOP	

Figura 5.7 Variables disponibles para la operación IPC.

Para este ejemplo, debemos obtener los identificadores de las variables Tipo de dato (id = 3), Provincias (id = 115) y Grupos ECOICOP (id = 762).

Para realizar la obtención de los valores de cada variable en INEbaseR, se puede hacer mediante la llamada a la función `get_values_all(id_variable)`.

Para realizar el ejemplo planteado hay que obtener:

Valores de la variable Tipo de dato

Elegimos el valor Variación mensual, que tiene como identificador el código 74.

	Id	Fk_Variable	Nombre	Codigo
1	70	3	Datos brutos	
2	71	3	Datos corregidos de efectos estacionales y de calend...	
3	72	3	Dato base	
4	73	3	Variación trimestral	
5	74	3	Variación anual	
6	75	3	Euros	
7	76	3	Tasa de variación interanual	
8	77	3	Porcentaje	
9	81	3	Tasa de variación anual	
10	82	3	Número de Horas	
11	83	3	Índice	
12	84	3	Variación mensual	
13	85	3	Media anual	M
14	86	3	Variación anual (para series anuales)	
15	87	3	Variación en lo que va de año	
16	88	3	Media en lo que va de año	

Figura 5.8 Valores de la variable Tipo de dato en INEbaseR.

Valores de la variable Provincias

Elegimos el valor Madrid, que tiene como identificador el código 28.

	Id	Fk_Variable	Nombre	Codigo
27	28	115	Lugo	27
28	29	115	Madrid	28
29	30	115	Málaga	29
30	31	115	Murcia	30
31	32	115	Navarra	31
32	33	115	Asturias	33
33	34	115	Palencia	34
34	35	115	Palmas, Las	35
35	36	115	Pontevedra	36
36	37	115	Salamanca	37
37	38	115	Santa Cruz de Tenerife	38
38	39	115	Cantabria	39
39	40	115	Segovia	40
40	41	115	Sevilla	41
41	42	115	Soria	42

Figura 5.9 Valores de la variable Provincias en INEbaseR.

Valores de la variable Grupos ECOICOP.

Elegimos todos los valores de esta variable.

	Id	Fk_Variable	Nombre	Codigo
1	304092	762	Índice general	00
2	304093	762	Alimentos y bebidas no alcohólicas	01
3	304094	762	Bebidas alcohólicas y tabaco	02
4	304095	762	Vestido y calzado	03
5	304096	762	Vivienda, agua, electricidad, gas y otros combustibles	04
6	304097	762	Muebles, artículos del hogar y artículos para el mant...	05
7	304098	762	Sanidad	06
8	304099	762	Transporte	07
9	304100	762	Comunicaciones	08
10	304101	762	Ocio y cultura	09
11	304102	762	Enseñanza	10
12	304103	762	Restaurantes y hoteles	11
13	304104	762	Otros bienes y servicios	12

Figura 5.10 Valores de la variable Grupos ECOICOP en INEbaseR.

Siguiendo este proceso, se consigue realizar la siguiente consulta que permite el cruce de metadatos para obtener un conjunto de series sujetas a este filtro personalizado.

"Provincias" *(FK_VARIABLE=115)* = "Madrid" *(FK_VALOR=29)* ⇒ *g1=115:29*
"Tipo de dato" *(FK_VARIABLE=3)* = "Variación mensual" *(FK_VALOR=84)* ⇒ *g2=3:84*
"Grupos ECOICOP" *(FK_VARIABLE=762)* = "Todos los grupos ECOICOP" *(FK_VALOR=null)* ⇒ *g3=762:*
"Serie mensual" *(FK_PERIODICIDAD=1)* ⇒ *p=1* *(Ver PUBLICACIONES_OPERACION)*

Figura 5.11 Consulta para realizar cruce de metadatos en la API JSON del INE.

Con lo que nuestra URL final para obtener los datos es la siguiente:

http://servicios.ine.es/wstempus/js/ES/SERIE_METADATAOPERACION/IPC?g1=115:29&g2=3:84&g3=762:&p=1

Sin embargo, debido al elevado número de series, es conveniente utilizar un lenguaje que permita hacer consultas más amigables al usuario.

Es por esto que INEbaseR contiene una mejora contemplada, y este proceso se puede realizar haciendo una llamada a la función `get_series_metadataoperation()` utilizando el parámetro "query" con una estructura similar a esta:

```
query = "Provincias = Madrid AND Tipo de dato = Variación
mensual AND Grupos ECOICOP = NULL"
```

	COD	T3_Operacion	Nombre	Decimales	T3_Periodicidad
1	IPC217343	Índice de Precios de Consumo (IPC)	Madrid. Restaurantes y hoteles. Variación mensual.	1	Mensual
2	IPC217355	Índice de Precios de Consumo (IPC)	Madrid. Transporte. Variación mensual.	1	Mensual
3	IPC217373	Índice de Precios de Consumo (IPC)	Madrid. Alimentos y bebidas no alcohólicas. Variac...	1	Mensual
4	IPC217364	Índice de Precios de Consumo (IPC)	Madrid. Vivienda, agua, electricidad, gas y otros c...	1	Mensual
5	IPC217370	Índice de Precios de Consumo (IPC)	Madrid. Bebidas alcohólicas y tabaco. Variación m...	1	Mensual
6	IPC217346	Índice de Precios de Consumo (IPC)	Madrid. Enseñanza. Variación mensual.	1	Mensual
7	IPC217340	Índice de Precios de Consumo (IPC)	Madrid. Otros bienes y servicios. Variación mensu...	1	Mensual
8	IPC217361	Índice de Precios de Consumo (IPC)	Madrid. Muebles, artículos del hogar y artículos p...	1	Mensual
9	IPC217367	Índice de Precios de Consumo (IPC)	Madrid. Vestido y calzado. Variación mensual.	1	Mensual
10	IPC217352	Índice de Precios de Consumo (IPC)	Madrid. Comunicaciones. Variación mensual.	1	Mensual
11	IPC216458	Índice de Precios de Consumo (IPC)	Madrid. Índice general. Variación mensual.	1	Mensual
12	IPC217349	Índice de Precios de Consumo (IPC)	Madrid. Ocio y cultura. Variación mensual.	1	Mensual
13	IPC217358	Índice de Precios de Consumo (IPC)	Madrid. Sanidad. Variación mensual.	1	Mensual

Figura 5.12 Obtención de series mediante el cruce de metadatos en INEbaseR utilizando lenguaje de consulta estructurada.

Esto permite que el usuario pueda obtener la misma información sin tener que conocer los identificadores de cada variable y cada valor.

El proceso que sigue esta función para generar la URL es el siguiente:

- Separar variables y valores de la consulta obtenida.
- Obtener identificadores de cada variable y valor.
- Generar la URL correspondiente.
- Obtener los datos.

El problema, es que igualmente el usuario desconoce las variables y valores de una operación para poder hacer esta consulta.

INEbaseR incorpora una función que permite obtener una tabla con todas las variables y valores de una operación, esta función se llama `get_var_values_operation()`, así que supongamos que queremos obtener todas las variables y los valores de la operación Índice de Precios de Consumo (IPC), sólo debemos de pasarle el parámetro op = "IPC".

	Id	Operacion	Fk_Variable	Nombre variable	Codigo	Nombre
49	19	25	115	Provincias	18	Granada
50	20	25	115	Provincias	19	Guadalajara
51	21	25	115	Provincias	20	Gipuzkoa
52	22	25	115	Provincias	21	Huelva
53	23	25	115	Provincias	22	Huesca
54	24	25	115	Provincias	23	Jaén
55	25	25	115	Provincias	24	León
56	26	25	115	Provincias	25	Lleida
57	27	25	115	Provincias	26	Rioja, La
58	28	25	115	Provincias	27	Lugo
59	29	25	115	Provincias	28	Madrid
60	30	25	115	Provincias	29	Málaga
61	31	25	115	Provincias	30	Murcia
62	32	25	115	Provincias	31	Navarra

Figura 5.13 Obtención de relación variable-valor de una operación estadística en INEbaseR.

5.3 Aspectos que destacar en el desarrollo de INEbaseR

Además del lenguaje de consulta estructurada detallado en la sección anterior, se han implementado varias mejoras que facilitan al usuario su utilización.

Es importante, al ser una librería en un software con licencia libre, que exista una documentación estandarizada y suficientemente clara. Esto se ha realizado mediante roxygen2 y se define en la siguiente sección.

Asimismo, para la optimización de tiempos de carga de las series de una operación y para agilizar las consultas de datos a la API, se ha implementado una caché de ficheros local que se verá en la sección 5.3.2.

5.3.1 Documentación

Como se ha comentado en el capítulo 3.5 Documentación con roxygen2, la documentación es uno de los aspectos más importantes del buen código.

Este package ha sido documentado con roxygen2, tal y como se puede observar en el siguiente ejemplo donde se muestra un fragmento de código de la función `get_serie()`:

```
#' @title Get serie
#' @description This function returns a data frame with a serie
from an id or code
#' @param code serie identification
#' @param det \code{det = 2} to see two levels of depth,
specifically to access the \code{PubFechaAct} object, \code{det
= 0} by default
#' @param tip \code{tip = M} to obtain the metadata (crossing
variables-values) of the series.
#' @param lang language used to obtain information
#' @param cache used to load data from local cache instead API,
\code{cache = FALSE} by default.
#' @param benchmark used to measure the performance of the
system, \code{benchmark = FALSE} by default.
#' @examples
#' get_serie("IPC206449")
#' get_serie("IPC206449", det = 2, tip = "M")
#' get_serie("IPC206449", det = 2, tip = "M", cache = FALSE,
benchmark = TRUE)
#' @export
get_serie <- function(code, det = 0, tip = NA, lang = "ES",
cache = FALSE, benchmark = FALSE) {
          ... código de la función ...
}
```

A continuación, se define cada etiqueta propia de roxygen2 utilizadas para documentar el código:

- title: título de la función.
- description: descripción de la función.
- param: parámetros que recibe la función.
- examples: ejemplos de uso de la función.
- export: permite exportar una función, es decir, permite que cuando se busque su documentación con el comando ?nombre_de_la_funcion en R, esta sea encontrada.

5.3.2 Caché

La caché en INEbaseR es una parte del desarrollo del package que se utiliza para almacenar datos y que las solicitudes futuras de esos datos se puedan atender con mayor rapidez.

El objetivo de este desarrollo es almacenar en caché de ficheros la información de la API del INE que es muy poco eficiente en las consultas a tiempo real. Por tanto, se trata de un archivo que tiene el usuario que instala la librería.

Los datos almacenados en esta caché son el resultado del duplicado de datos extraídos desde la API JSON del Instituto Nacional de Estadística (INE), principalmente de datos de series temporales que abarcan más espacio.

Se produce un acierto de caché cuando los datos solicitados se pueden encontrar en esta, mientras que un error de caché ocurre cuando no están dichos datos. La lectura de la caché es más rápida que volver a leer los datos desde la API JSON del Instituto Nacional de Estadística (INE).

Cuantas más solicitudes se puedan atender desde la memoria caché, más rápido funcionará el sistema.

La caché desarrollada en el package INEbaseR consta de las siguientes funciones ajenas al usuario:

- **build_cache_directory**: Crea el directorio "cache" si no existe. Es en este directorio donde se almacenarán los ficheros .RData que conforman los datos en memoria.
- **check_cache**: Comprueba si ya existe la información en caché.
- **clean_cache**: Vacía de la caché un determinado conjunto de datos almacenados en un fichero .RData. Esto es útil si el usuario ha elegido extraer los datos directamente desde la API (cache = FALSE) y ya existen estos datos en la memoria caché (en este caso, se borra el archivo anterior y se actualiza con los últimos datos obtenidos para así mantener la caché actualizada)
- **build_cache**: Guarda los datos obtenidos desde la API en memoria caché.
- **get_cache**: Obtiene los datos desde la memoria caché.

Benchmark: pruebas de rendimiento

Tras el desarrollo de una memoria caché en el package INEbaseR, se han hecho una serie de pruebas para medir el rendimiento (tiempo en segundos) de esta con respecto a la extracción de datos directamente de la API JSON del Instituto Nacional de Estadística (INE).

La siguiente tabla muestra una comparativa de tiempos entre la API del INE y el package INEbaseR (utilizando caché), de la extracción de las series de diferentes operaciones estadísticas:

Operación (id)	Registros	API del INE (t)	INEbaseR (t)	Mejora (%)
241	5.255	43.80 segundos	0.09 segundos	99.79 %
139	12.151	99.24 segundos	0.16 segundos	99.84 %
10	25.178	264.33 segundos	0.67 segundos	99.75 %

Tabla 5.1 Pruebas de rendimiento de INEbaseR con caché.

Las pruebas de la tabla anterior han sido realizadas con las siguientes características técnicas:

- Conexión de banda ancha de 10 Mb/s.
- Procesador i3-3110M a 2.40GHz.
- 8Gb de memoria RAM.
- Disco duro de estado sólido (SSD).

5.3.3 Análisis de datos

Además de extracción de datos, este proyecto también permite realizar representación y análisis de datos.

ggplot

ggplot2 es un package de visualización de datos para el lenguaje de programación estadístico R.

Este package fue creado por Hadley Wickham[20] en 2005, ggplot2 es una implementación de Gramática de gráficos de Leland Wilkinson: un esquema general para la visualización de datos que divide los gráficos en componentes semánticos como escalas y capas.

ggplot2 puede servir como reemplazo de los gráficos base en R y contiene una serie de valores predeterminados para la visualización web e impresión de escalas comunes.

Desde 2005, ggplot2 ha crecido en uso para convertirse en uno de los paquetes R más populares y está licenciado bajo GPLv2.

highcharter

highcharter es un wrapper de la librería 'Highcharts' que incluye funciones de acceso directo para trazar objetos R.

La librería original 'Highcharts', es una librería de gráficos implementada en el lenguaje de programación JavaScript en 2009 por la empresa Highsoft, y ofrece numerosos tipos de gráficos con una sintaxis de configuración simple.

En este proyecto, la librería highcharter es utilizada para analizar los datos abiertos extraídos de la API JSON del Instituto Nacional de Estadística (INE) de las series temporales de una operación estadística.

5.4 Licencia: GPLv3.0

Este proyecto está desarrollado bajo la licencia GNU General Public License v3.0.

Los permisos de esta licencia de copyleft están condicionados a poner a disposición un código fuente completo de obras y modificaciones con licencia, que incluyen obras más grandes que utilizan un trabajo bajo licencia, bajo la misma licencia. Los derechos de autor y avisos de licencia deben ser preservados. Los colaboradores proporcionan una concesión expresa de derechos de patente.

5.5 Repositorio

El código del proyecto está disponible libre y gratuitamente en el repositorio GitHub del autor: oddworldng/INEbaseR.

5.6 Redes sociales

Para facilitar el seguimiento de los avances del proyecto por parte de la comunidad y para facilitar su búsqueda en buscadores de internet, este proyecto puede encontrarse en:

- Blog oficial: https://inebaser.wordpress.com/
- RSS: https://inebaser.wordpress.com/feed/
- Twitter: https://twitter.com/INEbaseR

5.7 Premios y reconocimientos

Este proyecto ha ganado los siguientes premios y reconocimientos:

- PREMIO AL MEJOR PROYECTO en el VIII Concurso Universitario de Software Libre local[27] de la Universidad de La Laguna[28].

- PREMIO AL MEJOR PROYECTO DE INVESTIGACIÓN en el XII Concurso Universitario de Software Libre nacional[29] en la Universidad de Sevilla[30].

[27] http://cusl.osl.ull.es/
[28] https://www.ull.es/
[29] https://concursosoftwarelibre.org/1718/
[30] http://www.us.es/

Capítulo 6. Conclusiones y trabajos futuros

En conclusión, este proyecto abarca grandes aspectos relacionados con la extracción y análisis de datos abiertos publicados por el Instituto Nacional de Estadística (INE) que facilita mucho al usuario su labor, y en la actualidad consta de varios premios e interesados.

La posibilidad de acceso a múltiples fuentes y orígenes de datos en la red supone una gran oportunidad para realizar un gran número de estudios y análisis en una gran multitud de áreas. Sin embargo, la extracción de dichos datos puede llegar a representar un gran problema cuando los usuarios interesados no disponen de los conocimientos necesarios para obtener la información de una forma estructurada que le permita trabajar con ellos directamente utilizando las herramientas de análisis habituales. Por esta razón, el objetivo del presente proyecto ha consistido en desarrollar una aplicación en forma de librería que facilite estos procedimientos de acceso a los datos que publica un organismo público, en este caso el Instituto Nacional de Estadística.

El desarrollo realizado se integra en un software ampliamente utilizado y conocido (R) proporcionando una serie de métodos para la extracción de datos que el usuario puede ejecutar de forma sencilla. La librería permite no sólo obtener los datos en formato tabular, sino que además es posible realizar representaciones gráficas de gran interés.

Como trabajos futuros, cabe destacar que aún se podrían realizar algunas mejoras tales como:

- Desarrollar un ajuste predictivo a los datos (para predecir el futuro de las series).
- Detectar valores anómalos en las series.
- Representar dos o más series de forma conjunta en un mismo gráfico.

Capítulo 7. Summary and conclusions

In conclusion, this project covers large aspects related to the extraction and analysis of open data by the National Institute of Statistics (INE) that greatly facilitate the user's work, and currently consists of several awards and stakeholders.

The access to multiple data sources in internet supposes a real opportunity to make a high number of studies and analyses in many areas. However, the extraction of such data can be represented a huge problem when the users interested in them do not have the knowledge about how to obtain the information in a structured way to work using the most popular data analysis tools. For this reason, the objective of the current project has consisted in developing a library to ease the use of these procedures of accessing to data published by a public organism, as the “Instituto Nacional de Estadística”.

The development is perfectly integrated in a widely and very well-known software, R, providing a collection of methods to data extraction that can be executed by the users in an easy way. The library allows not only to obtain tabular data even to make highly interesting graphical representations.

As a future work, it should be noted that some improvements can still be made, such as:

- Develop a predictive fit to the data (to predict the future of the series).
- Detect anomalous values in the series.
- Represent two or more series jointly in the same chart.

Bibliografía

[1] ‘Gobierno de España’. [Online]. Available: http://datos.gob.es/es.

[2] ‘Gobierno de Canarias’, 2010. [Online]. Available: http://www.gobiernodecanarias.org/.

[3] Jefatura del Estado, *LEY 37/2007, de 16 de noviembre, sobre reutilización de la información del sector público.* 2007, p. 6.

[4] Parlamento Europeo, *Directiva 2003/98/CE del Parlamento europeo y del Consejo*. 2003, p. 7.

[5] Ministerio de la Presidencia, *Real Decreto 1495/2011*. 2011.

[6] W3C, ‘World Wide Web Consortium’, 2013. [Online]. Available: http://www.w3.org/standards/webdesign/i18n.

[7] Ministerio de Hacienda y Administraciones Públicas, *Norma Técnica de Interoperabilidad de Reutilización de recursos de la información*. 2013, p. 27.

[8] G. Van Rossum, [Online]. Available: https://gvanrossum.github.io/Resume.html.

[9] Free Software Foundation, ‘Licencia GPL’, 29-Jun-2007. [Online]. Available: http://www.gnu.org/licenses/gpl.html.

[10] F.-S. Krah, ‘rusda’, 03-Apr-2016. [Online]. Available: https://cran.r-project.org/web/packages/rusda/index.html.

[11] K. R. Ryberg and A. V. Vecchia, ‘waterData’, 28-Apr-2017. [Online]. Available: https://cran.rstudio.com/web/packages/waterData/index.html.

[12] R. McTaggart, G. Daroczi and C. Leung, ‘Quandl’, 23-Apr-2016. [Online]. Available: https://cran.rstudio.com/web/packages/Quandl/index.html.

[13] G. Thor Briem, ‘rdatamarket’, 24-Nov-2014. [Online]. Available: https://cran.rstudio.com/web/packages/rdatamarket/index.html.

[14] E. Szöcs, ‘webchem’, 07-Apr-2018. [Online]. Available: https://cran.rstudio.com/web/packages/webchem/index.html.

[15] A. Nacimiento and C. J. Pérez, ‘INEbaseR’, 2018. [Online]. Available: https://github.com/oddworldng/INEbaseR.

[16] C. González, ‘istacr’, 2018. [Online]. Available: https://github.com/rOpenSpain/istacr.

[17] H. Parker, ‘Writing an R package from scratch’, 2014. [Online]. Available: https://hilaryparker.com/2014/04/29/writing-an-r-package-from-scratch/.

[18] H. Wickham, ‘Roxygen2’, 06-Feb-2017. [Online]. Available: https://cran.r-project.org/web/packages/roxygen2/vignettes/roxygen2.html.

[19] R. Pawson, *Clean Code: A Handbook of Agile Software Craftsmanship*, vol. 17, no. 2. 2011.

[20] H. Wickham, ‘R packages’, 2018. [Online]. Available: http://r-pkgs.had.co.nz.