

# Customs Visuals en Power BI con el lenguaje R

**Autora: Núria Pont Vilà**

Universitat Oberta de Catalunya (UOC)  
Master Universitario en Ciencia de Datos ( Data Science)  
Área: Salud pública y Medical Devices

**Tutora: Susana Pérez Alvarez**  
**Profesora: Angels Rius Gavidia**

23/06/2021



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

.

## **Agradecimientos:**

He tenido la suerte de enfocar mi vida profesional al análisis de datos, la cual cosa me apasiona, y ahora he podido realizar un máster dedicado a ello, con el que poder consolidar conocimientos que me son de gran ayuda.

He disfrutado en cada una de las asignaturas y como ‘broche de oro’, he podido trabajar en un proyecto que quiere aportar un granito de arena, para que cada vez más el mundo de los datos deje de ser tan misterioso y se vea como el gran tesoro que es.

Quiero agradecer a mi tutora, Susana Pérez Álvarez, sus observaciones, sus consejos, sus propuestas, sus rápidas respuestas ante mis dudas, gracias, sin ti no hubiera sido lo mismo.

También agradecer a Jordi Font, tutor del máster, a cada uno de los profesores y a la UOC por ofrecer la oportunidad de realizar este máster.

A nivel personal, quiero dar las gracias a mi pareja Xevi y a mis pequeñitos Miquel y Arnau, que han tenido la paciencia de adaptarse en muchos casos al ritmo que yo debía seguir.

## FICHA DEL TRABAJO FINAL

|                                    |  |
|------------------------------------|--|
| <b>Título del trabajo:</b>         | <i>Customs Visuals en Power BI con el lenguaje R</i>     |
| <b>Nombre de la autora:</b>        | <i>Núria Pont Vilà</i>                                   |
| <b>Nombre del consultora:</b>      | <i>Susana Pérez Alvarez</i>                              |
| <b>Nombre del PRA:</b>             | <i>Angels Rius Gavidia</i>                               |
| <b>Fecha de entrega (mm/aaaa):</b> | 06/2021  |
| <b>Titulación::</b>                | Master Universitario en Ciencia de Datos ( Data Science) |
| <b>Área del Trabajo Final:</b>     | Salud pública y Medical Devices                          |
| <b>Idioma del trabajo:</b>         | <i>Español</i>   |
| <b>Palabras clave</b>              | <i>Power BI<br/>R<br/>custom visual</i>                  |

**Resumen del Trabajo (máximo 250 palabras):** *Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.*

La finalidad de este proyecto, es mostrar la oportunidad de aportar nuevos gráficos basados en lenguajes de inteligencia artificial sin la necesidad que el usuario desarrollador del informe que los va a utilizar, conozca, ni deba interpretar, una sola línea de código.

Se integrarán dos herramientas como son Power BI (proporciona visualizaciones interactivas) y el lenguaje estadístico R, utilizando las librerías ggplot2 y plotly, con las que poder explorar gráficamente los datos, logrando descubrir “información oculta” que permita pronosticar y ver resultados estratégicos.

Para ello, se realiza un análisis de los elementos mencionados, observando e investigando sus fortalezas y debilidades, con lo que se elabora un documento dónde se detalla los procedimientos a seguir.

Power BI, utiliza unos elementos llamados objetos visuales, los cuales son un tipo de gráfico generado por diseñadores de Power BI. Los objetos visuales que no se incluyen automáticamente con Power BI se conocen como objetos visuales personalizados o también conocidos como custom visuals. Estos objetos, se pueden importar en Power BI desde sitios web externos, como Microsoft AppSource (contiene miles de aplicaciones empresariales y servicios), o desde orígenes internos, como el almacén de la organización.

Al final de este proyecto, hemos logrado crear nuestro propio custom visual y en breve se conseguirá ampliar la biblioteca AppSource de Power BI con un nuevo objeto diseñado en R.

Para finalizar, se desarrollará un panel o *dashboard* (nombre inglés mayoritariamente usado) en Power BI, con visualizaciones interactivas creadas a partir de nuestro custom.

**Abstract (in English, 250 words or less):**

The purpose of this project is to show the opportunity to provide new graphics based on artificial intelligence languages without the need for the user who develops the report that is going to use them, to know, or to interpret, a single line of code.

The two tools will be integrated such as Power BI (provides interactive visualizations) and the statistical language R, using the ggplot2 and plotly libraries, with which to graphically explore the data, discovering “hidden information” that allows forecasting and seeing strategic results.

To do this, an analysis of the aforementioned elements is carried out, observing and investigating their strengths and weaknesses, with which a document is prepared detailing the procedures to be followed.

Power BI uses elements called visual objects, which are a type of chart generated by Power BI designers. Visuals that are not automatically included with Power BI are known as custom visuals or also known as custom visuals. These objects can be imported into Power BI from external websites, such as Microsoft AppSource (contains thousands of business applications and services), or from internal sources, such as the organization store.

At the end of this project, we have managed to create our own visual custom and will soon be able to expand the Power BI AppSource library with a new object designed in R.

Finally, a dashboard will be developed in Power BI, with interactive visualizations created from our custom.

# Índice

|        |  |    |
|--------|--|----|
| 1.     | Introducción .....   | 1  |
| 1.1.   | Contexto y justificación del Trabajo .....                           | 1  |
| 1.2.   | Objetivos del Trabajo.....   | 2  |
| 1.3.   | Enfoque y método seguido .....                                       | 2  |
| 1.4.   | Planificación del Trabajo.....                                       | 4  |
| 1.5.   | Breve resumen de productos obtenidos .....                           | 5  |
| 1.6.   | Breve descripción de los otros capítulos de la memoria.....          | 5  |
| 2.     | Estado del arte.....   | 6  |
| 3.     | Diseño e implementación.....   | 13 |
| 3.1.   | Parte técnica:.....  | 13 |
| 3.1.1. | Prerrequisitos iniciales a instalar .....                            | 13 |
| 3.1.2. | Creación estructura del custom visual.....                           | 16 |
| 3.1.3. | Análisis y Diseño del custom visual.....                             | 20 |
| 3.2.   | Instrucciones de uso del custom visual en Power BI .....             | 40 |
| 3.2.1. | Incorporar datos a tratar .....                                      | 40 |
| 3.2.2. | Incorporar objeto Visual .....                                       | 43 |
| 3.2.3. | Selección tipo de gráfico .....                                      | 46 |
| 3.2.4. | Selección de características.....                                    | 47 |
| 3.2.5. | Filtros y relación con otros gráficos.....                           | 49 |
| 3.2.6. | Consideraciones al desarrollador Informes con objetos visual R ..... | 51 |
| 3.3.   | Requisitos y limitaciones de los paquetes R.....                     | 51 |
| 3.4.   | Consideraciones al desarrollador del custom .....                    | 52 |
| 3.5.   | Publicación objeto visual .....                                      | 52 |
| 3.5.1. | Preparar custom visual.....  | 53 |
| 3.5.2. | Obtención de un archivo XML del paquete de la aplicación .....       | 54 |
| 3.5.3. | Envío al Centro de partners .....                                    | 54 |
| 3.5.4. | Seguimiento del uso y estado del envío.....                          | 55 |
| 3.5.5. | Certificación del objeto visual .....                                | 55 |
| 3.5.6. | Problemática .....   | 55 |
| 3.6.   | Repositorio Código .....   | 56 |
| 4.     | Conclusiones .....   | 57 |
| 5.     | Glosario .....   | 61 |
| 6.     | Bibliografía.....  | 63 |
| 7.     | Anexos.....  | 66 |
| 7.1.   | <i>Manifest</i> XML (Microsoft).....                                 | 66 |
| 7.2.   | Ejemplos de gráficos del custom visual realizado.....                | 67 |

## Lista de figuras

|   |    |
|---|----|
| Figura 1: Esquema metodología SCRUM .....                         | 3  |
| Figura 2: Esquema metodología CRISP-DM.....                       | 3  |
| Figura 3: Cronograma TFM .....                                    | 4  |
| Figura 4: Diagrama de Gantt TFM.....                              | 4  |
| Figura 5: Logotipos Power BI y R.....                             | 6  |
| Figura 6: Marketplace.....  | 6  |
| Figura 7: Cuadro mágico 'Gartner Inc'.....                        | 7  |
| Figura 8: Evolución interés Power BI vs QlikTech vs Tableau ..... | 7  |
| Figura 9: Logos lenguajes R y Python.....                         | 8  |
| Figura 10: Tendencias lenguajes de programación.....              | 9  |
| Figura 11: Logos librerías ggplot2 y plotly .....                 | 9  |
| Figura 12: Elementos del Marketplace .....                        | 11 |
| Figura 13: Opciones activación R.....                             | 15 |
| Figura 14: Estructura VSC .....                                   | 16 |
| Figura 15: Estructura archivos.....                               | 17 |
| Figura 16: settings.json .....                                    | 18 |
| Figura 17: pbviz.json .....                                       | 18 |
| Figura 18: visual.ts .....  | 19 |
| Figura 19: pvibiz .....   | 21 |
| Figura 20: dependencies.json .....                                | 21 |
| Figura 21: Código desarrollo opciones .....                       | 23 |
| Figura 22: Imagen error por defecto .....                         | 23 |
| Figura 23: Imagen mensaje informativo .....                       | 24 |
| Figura 24: Código validación elementos .....                      | 24 |
| Figura 25: Menú custom visual.....                                | 25 |
| Figura 26: Objects .....  | 25 |
| Figura 27: settings_Global.....                                   | 26 |
| Figura 28: settings_Scatter.....                                  | 26 |
| Figura 29: settings_Box_Violin .....                              | 26 |
| Figura 30: settings.ts .....                                      | 26 |
| Figura 31: Menú objeto visual-settings_Global.....                | 27 |
| Figura 32: capabilities.json - gráficos .....                     | 27 |
| Figura 33: BoxPlot/Violin [20 ] .....                             | 28 |
| Figura 34: Scatter point .....                                    | 29 |
| Figura 35: Histograma vs Bar.....                                 | 30 |
| Figura 36: Scatter 3D .....                                       | 31 |
| Figura 37: Scatter 3D rotation .....                              | 31 |
| Figura 38: settings_Global.....                                   | 32 |
| Figura 39: settings-size .....                                    | 32 |
| Figura 40: settings-temas.....                                    | 33 |
| Figura 41: Paleta "ColorBrewer" por Cynthia Brewer .....          | 34 |
| Figura 42: Disposición.....                                       | 34 |
| Figura 43: Menú objeto visual-settings_Global.....                | 35 |
| Figura 44: settings.....  | 35 |
| Figura 45: ejemplo código gráfico histograma.....                 | 36 |
| Figura 46: Código resultado final.....                            | 36 |

|   |    |
|---|----|
| Figura 47: Viloin + jitter + trim.....                | 37 |
| Figura 48: Código Icon .....                          | 37 |
| Figura 49: Icon .....                                 | 38 |
| Figura 50: VSC-Terminal.....                          | 38 |
| Figura 51: Compilación .....                          | 38 |
| Figura 52: Paquete objeto visual.....                 | 39 |
| Figura 53: Imagen Inicial Power BI.....               | 40 |
| Figura 54: Imagen origen dataset Heart Attack.....    | 41 |
| Figura 55: Información elaborada en RStudio .....     | 42 |
| Figura 56: Obtención dataset .....                    | 42 |
| Figura 57: Campos dataset heart.csv.....              | 43 |
| Figura 58: Visualizations .....                       | 43 |
| Figura 59: Incorporar objeto visual .....             | 44 |
| Figura 60: Imagen nuevo icono .....                   | 44 |
| Figura 61: Elementos de los ejes .....                | 44 |
| Figura 62: Selección elementos .....                  | 45 |
| Figura 63: Modificación nomenclatura ejes .....       | 45 |
| Figura 64: Imagen de error.....                       | 46 |
| Figura 65: Imagen explicativa.....                    | 46 |
| Figura 66: Sección Format .....                       | 47 |
| Figura 67: Tipo de gráficos.....                      | 47 |
| Figura 68: Características settings_Global.....       | 48 |
| Figura 69: Características tendencia.....             | 48 |
| Figura 70: Características settings_Box_Violin .....  | 49 |
| Figura 71: Ejemplo scatter_point - loess .....        | 49 |
| Figura 72: Ejemplo con filtro.....                    | 50 |
| Figura 73: Ejemplo dos gráficas .....                 | 50 |
| Figura 74: Ejemplo con filtrado de otra gráfica ..... | 50 |
| Figura 75: Activación visuales script .....           | 51 |
| Figura 76: Instalación paquetes R.....                | 51 |
| Figura 77: Metadatos paquete pbiviz .....             | 53 |
| Figura 78: Página Inicial informe .....               | 53 |
| Figura 79: Estado requerimientos .....                | 54 |
| Figura 80: Repositorio Código TFM .....               | 56 |
| Figura 81: Gráfico con coordenadas geográficas.....   | 60 |
| Figura 82: Dashboard Power BI .....                   | 67 |
| Figura 83: Gráfico boxplot .....                      | 67 |
| Figura 84: Opciones Globales.....                     | 68 |
| Figura 85: Gráfico boxplot horizontal.....            | 68 |
| Figura 86: Elementos visuales disponibles.....        | 69 |
| Figura 87: Acción Zoom .....                          | 69 |
| Figura 88: Selección Zoom .....                       | 70 |
| Figura 89: resultado Zoom .....                       | 70 |
| Figura 90: Gráfico violín .....                       | 71 |
| Figura 91: Gráficos Histogramas/Barras .....          | 71 |
| Figura 92: Gráficos Scatter.....                      | 72 |
| Figura 93: Gráficos Scatter 3D .....                  | 72 |



# 1. Introducción

## 1.1. Contexto y justificación del Trabajo

La propuesta de generar gráficos estadísticos interactivos desde Power BI con el lenguaje R, genera mucho interés a nivel social en cualquiera de los ámbitos que se pueda tratar con datos.

Cualquier científico/analista que investigue/estudie datos querrá que sus conclusiones sean lo más entendibles posibles para el lector/receptor. Este lector querrá, a su vez, comprender por él mismo, el resultado obtenido pudiendo interactuar con dichos datos de forma pragmática y poder elaborar sus propias conclusiones.

Cuanto más claras y más opciones pueda aportar este científico al usuario lector más posibilidades tendrán ambos de conseguir sus logros.

Presentar datos estadísticos a través de gráficos es una tarea muy importante en el proceso de comunicación de los datos y para ello es muy importante la selección de las herramientas gráficas que se van a utilizar con el objetivo que el lector pueda comprender fácilmente y rápidamente lo que se está representando.

Este estudio pretende profundizar, revisar o extender los conocimientos ya existentes en esta área y poder aportar estos resultados a los futuros desarrollos de manera que se facilite la comprensión de la información elaborada científicamente, mediante la interacción realizada por el propio lector con los gráficos resultantes.

A nivel personal, siempre me ha apasionado el mundo de la lógica y encontrar o intentar encontrar un sentido a las cosas, me he dado cuenta de que muchas veces se puede encontrar, pero al mismo tiempo, muchas veces es muy difícil de explicarlo. He ido experimentando a lo largo de mi carrera profesional que una buena imagen, un buen gráfico puede explicar de forma muy sencilla lo que a veces en palabras o escritos sería mucho menos comprensible.

Siguiendo en esta línea, veo en este proyecto una oportunidad para facilitar esta 'traducción' de lo que está pasando, en un entorno donde el usuario pueda jugar interactuando estos datos e interpretando la información de manera mucho más intuitiva. El objetivo es aportar más visión en su trabajo para poder hacer análisis donde extraer conclusiones y/o futuras estrategias para sus necesidades.

Sinceramente me motiva pensar que alguna persona que pueda estar trabajando en el área donde aplica este trabajo de fin de master, 'Área: Salud pública y Medical Devices', pueda utilizar algunas de las funcionalidades que exploraremos y que pueda ayudarlo a comunicar/exponer más fácilmente los resultados en los que esté trabajando.

## **1.2.Objetivos del Trabajo**

El objetivo principal será estudiar y analizar la factibilidad de la conexión entre las herramientas Power BI y el lenguaje R.

Una vez logrado este objetivo se estudiará la viabilidad de la creación de gráficos interactivos. El objetivo es mostrar que se puede generar gráficos analíticos que permitan ver lo que está ocurriendo y poder interactuar con ellos. Se intentará generar los siguientes gráficos: boxplot, scatter-puntos, scatter-barra y scatter 3D

El siguiente objetivo será crear un objeto visual que pueda generar alguno de los gráficos mencionados.

Una vez conseguido el custom visual, se intentará pasar la validación para que sea publicado en el Marketplace de Microsoft, de esta manera estará disponible para la comunidad de desarrolladores de informes Power BI y también para programadores que usen R y quiera aproximar sus gráficos a los usuarios que no tengan conocimientos de programación.

Como objetivo general, no se perderá de vista los principios de entendibilidad, claridad, consistencia, eficiencia, necesidad y confiabilidad

Resumiendo:

- Conexión entre las herramientas Power BI y el lenguaje R
- Creación e Interacción de los gráficos
- Creación custom visual propio
- Parametrización elementos visuales
- Publicación custom visual en Marketplace
- Conseguir principios de entendibilidad, claridad, consistencia, eficiencia, necesidad y confiabilidad

## **1.3.Enfoque y método seguido**

El proyecto se va a basar en gran parte, en la investigación de estudiar el qué, el cómo y el dónde de las funcionalidades que aportan las herramientas con las que se va a trabajar, sin olvidar nunca para quién se va a realizar.

Al inicio del proyecto por mi parte existen conocimientos avanzados sobre el lenguaje de programación R pero básicos a nivel de Power BI y nulos respecto a la integración de ambos.

Con las premisas descritas y con el fin de cumplir los objetivos marcados y controlar posibles demoras, se ha pensado en usar una metodología que sea muy

dinámica, que se controle los avance, pudiéndose revisar posteriormente. La metodología que se ajusta a estas necesidades y es perfectamente aplicable es la Scrum [1].

Scrum anima a aprender a través de las experiencias, a autorganizarse mientras se trabaja en un problema y a reflexionar sobre sus aciertos y sus errores, para mejorar continuamente. Se irá creando y presentando el producto a la tutora (Susana) en intervalos cortos (sprints) y recibiendo feedbacks rápidos y precisos.



Figura 1: Esquema metodología SCRUM

Existen otras metodologías, que hemos visto en el máster, como podría ser CRISP-DM [2], que siguen un proceso definido, son más rigurosas en la planificación del proyecto, la documentación, control de requisitos, control implantación y son perfectamente válidas para proyectos y sectores ligados con la minería de datos como pueden ser los seguros, el financiero o el matemático. En este caso, como se ha comentado, me reafirmo con Scrum, segmentaré el proyecto en bloques con el fin de desarrollar un producto sencillo pero funcional e ir mejorándolo poco a poco, dando por sentado que dicho producto va a cambiar, evolucionando y mejorando constantemente



Figura 2: Esquema metodología CRISP-DM

## 1.4. Planificación del Trabajo

La planificación del proyecto ha sido elaborada a partir de las actividades definidas en el Máster de Ciencia de Datos de la UOC. ( PACs)

Los hitos principales serán precisamente las fechas de entrega de dichas PAC's.

En el siguiente cronograma se puede visualizar las diferentes etapas.

| ID | Nombre  | Duración | % Dedicación | Inicio        | Terminado      | Pr |
|----|---|----------|--------------|---------------|----------------|----|
| 1  | Lectura Plan Docente                                | 4 days   | 0%           | 13/02/21 8:00 | 16/02/21 17:00 |    |
| 2  | PAC1 - Definición y planificación del trabajo final | 12 days  | 5%           | 17/02/21 8:00 | 28/02/21 17:00 | 1  |
| 6  | PAC2 - Estado del arte o analisis del mercado       | 20 days  | 10%          | 1/03/21 8:00  | 20/03/21 17:00 | 2  |
| 11 | PAC3 - Diseño y implementación del trabajo          | 63 days  | 40%          | 21/03/21 8:00 | 22/05/21 17:00 | 6  |
| 21 | PAC4 - Redacción de la memoria                      | 14 days  | 25%          | 24/05/21 8:00 | 6/06/21 17:00  | 11 |
| 25 | PAC5 - Presentación y defensa del proyecto          | 7 days   | 10%          | 7/06/21 8:00  | 13/06/21 17:00 | 21 |
| 28 | Defensa pública                                     | 1 day    | 1 day        | 23/06/21 8:00 | 23/06/21 17:00 | 25 |

Figura 3: Cronograma TFM

En esta segunda imagen del cronograma, se puede visualizar con más detalles las sub-acciones

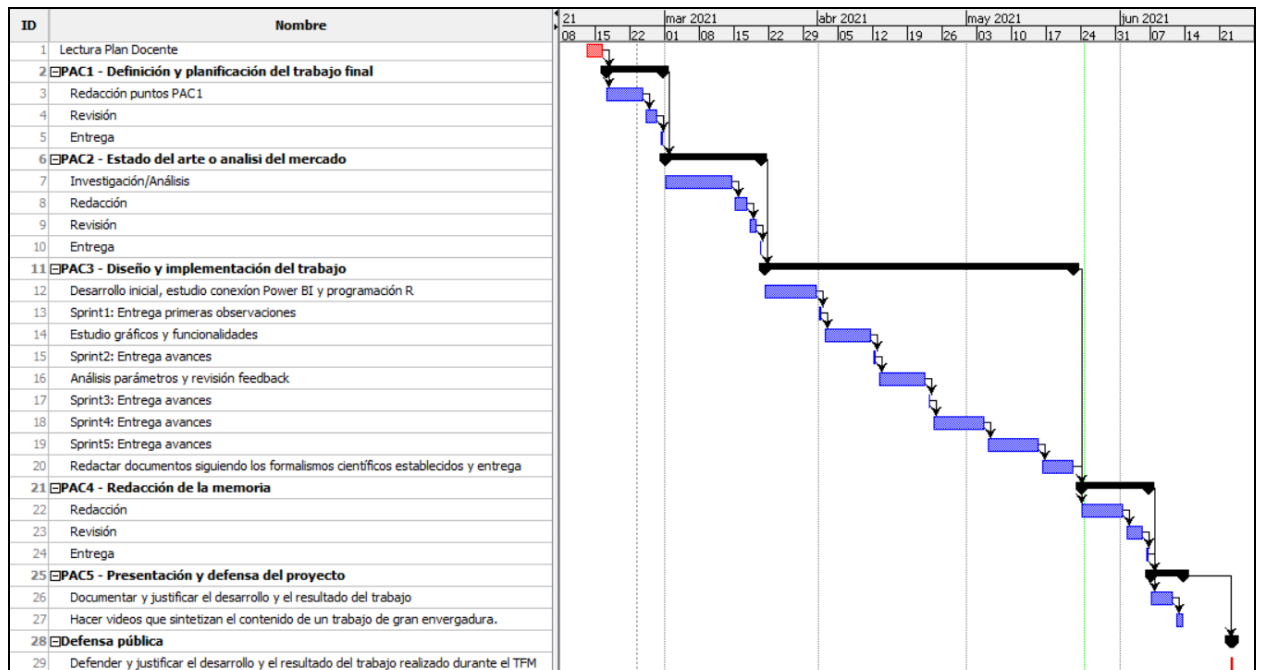


Figura 4: Diagrama de Gantt TFM

Nota: El cronograma está sujeto a actualizaciones en función de los progresos que se realicen. Los tiempos indicados en las sub-acciones, son orientativos, pudiéndose verse reducidos y/o ampliados, pero siempre dentro del marco de fechas que indica la acción principal.

Los recursos necesarios serán:

- RStudio
- VSC (Visual Studio Code)
- Power BI Desktop

En los siguientes apartados se dará el detalle de ellos.

## **1.5. Breve resumen de productos obtenidos**

El producto final será un custom visual u objeto visual (a lo largo del documento, se llamará indistintamente de las dos maneras), que se pondrá a disposición de la herramienta Power BI, para poder generar gráficos que han sido desarrollados con lenguaje de programación R.

## **1.6. Breve descripción de los otros capítulos de la memoria**

La memoria se divide en los siguientes capítulos:

- 1. Introducción.
- 2. Estado del arte: Investigación de los diferentes tipos de customs visuals que existen en la actualidad en Power BI. Concretar en los que se basan en desarrollos con lenguaje R. Estudiar sus debilidades y fortalezas.
- 3. Diseño e implementación: Es el gran capítulo en cuanto a tiempo dedicado, hay una primera parte esencial que es la preparación y ejecución técnica de todo lo necesario para crear la estructura del objeto. A partir de la estructura se realiza el desarrollo e implementación de los elementos que se quieren tratar.
- 4. Conclusiones: A partir de la comparativa de los objetivos previstos con los objetivos reales conseguidos se elabora la conclusión de los resultados
- 5. Glosario: Conceptos, acrónimos y siglas utilizadas en la memoria
- 6. Bibliografía: documentos y urls utilizados en el proyecto
- 7. Anexos: Documentos que se adjuntan como parte de la memoria que aportan detalles de los capítulos anteriores

## 2.Estado del arte

Se analizará la situación actual de los elementos principales que vamos a tratar, Power BI y R.

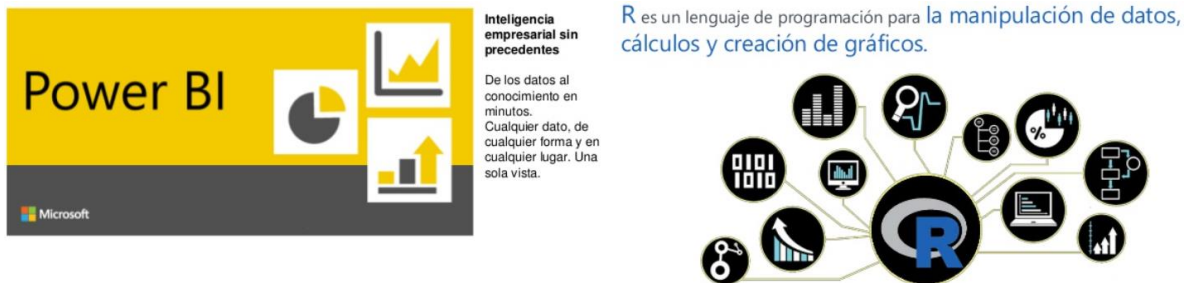


Figura 5: Logotipos Power BI y R

A su vez se investigará qué hay en el Marketplace de Microsoft referente a R y qué aportaciones da la unión de Power BI y R.

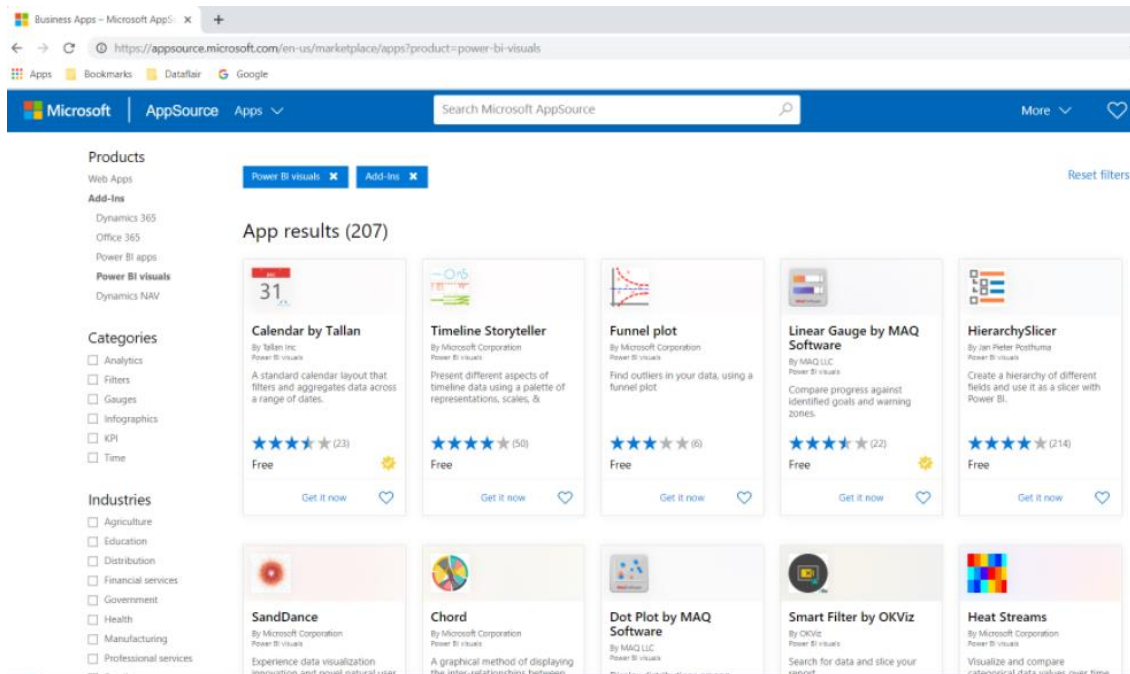


Figura 6: Marketplace

Secciones de este capítulo:

1. Power BI y lenguajes que utiliza
2. R y librerías de visualización de datos
3. Power BI & R
4. Marketplace Power BI
5. Otros puntos a investigar y resolución del estudio

## 1) Power BI y lenguajes que utiliza

Uno de los elementos principales de este proyecto, Power BI [3], es un servicio de análisis empresarial de Microsoft. Su principal objetivo es proporcionar visualizaciones interactivas y capacidades de inteligencia empresarial, también conocidas por 'Business Intelligence' más conocido por su acrónimo 'BI', tratando una interface sencilla para ser utilizada.

Como se puede ver en la siguiente imagen, el 'cuadrado mágico' presentado en Febrero de 2021 por la consultora 'Gartner Inc.' [4], de referencia para profesionales y empresas del ámbito de las TIC, sitúa como líder el programa de Microsoft, destacando su papel visionario junto con la facilidad de ejecución.



Figura 7: Cuadrado mágico 'Gartner Inc'

Para realizar el estudio se ha buscado una comparativa con más detalles sobre softwares de análisis de datos que hay en el mercado con sus pros y contras, principalmente Qlik y Tableau. En la siguiente tabla se puede visualizar un resumen del interés en el último año de estos programas. Para ver más detalles, en el enlace [5] de la bibliografía se podrá acceder y consultar la información.

● power bi ● QlikTech ● Tableau Software

Evolución del interés

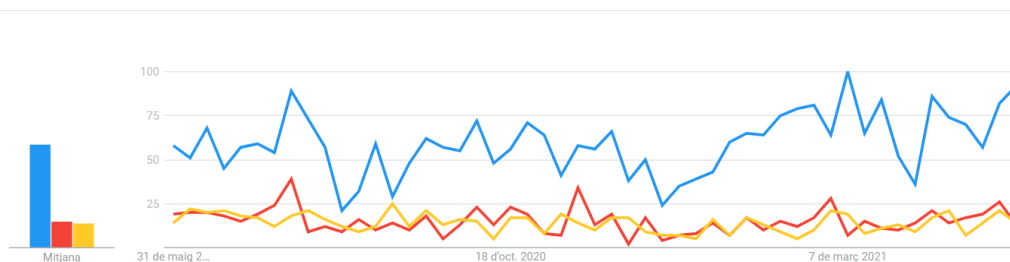


Figura 8: Evolución interés Power BI vs QlikTech vs Tableau

Un elemento clave, será la elección del lenguaje de programación que se utilizará. Se ha realizado un estudio de los lenguajes mayoritarios con los que puede trabajar la herramienta Power BI: [6]

- DAX(*Data Analysis Expressions*): es un lenguaje de cálculo de datos analíticos que se puede utilizar para un análisis de datos en profundidad durante la fase de visualización de datos.
- M : se puede considerar como un lenguaje de fórmulas de consulta y se puede usar en el editor de consultas de Power BI para preparar los datos antes de que se puedan cargar en el modelo de Power BI
- SQL: lenguaje de consulta estructurado (SQL)
- MDX: (Multi-Dimensional expressions) es un lenguaje de consulta usado para sacar datos de una base de datos multidimensional. Más concretamente, MDX se usa para consultar datos de una base de datos OLAP con Analysis Services y otras herramientas que soporten este lenguaje.
- R (*Microsoft R Open*): lenguaje Open Source orientado a análisis estadístico con el que aplicar multitud de algoritmos y a su vez graficar los resultados.
- Python: es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código.
- Visual Script: es un lenguaje interpretado por el Windows Script Host de Microsoft. Su sintaxis refleja su origen como variación del lenguaje de programación Visual Basic.
- TypeScript( lenguaje de desarrollo de JavaScript de la era moderna. Es un lenguaje compilado estáticamente para escribir código Javascript claro y simple.)

## 2) R y librerías de visualización de datos

De la lista anterior, los lenguajes más utilizados en la ciencia de datos son Python y R. Ambos son excelentes lenguajes de programación, si bien, para el proyecto en el que trabajamos, diseño de gráficos, se realizará con el lenguaje R, el cual está orientado al análisis estadístico y es más potente en visualización de información y datos que Python.



Figura 9: Logos lenguajes R y Python

Observando las tendencias de los lenguajes de programación se puede observar en diferentes fuentes redmonk [7] ,tiobe[8] ,pypl[9] como el lenguaje Python está situado en los primeros puestos, y R está entre los 13 primeros, es muy relevante la posición de R ya que el público objetivo es una comunidad muy especializada que requiere un mayor conocimiento de conceptos propios de la estadística, mientras que Python es un



lenguaje más sencillo de aprender y utilizar con lo que es más generalista, aunque no podrá obtener los resultados que consigue R.



Figura 10: Tendencias lenguajes de programación

El objetivo es generar un objeto visual que obtenga gráficos, para ello se ha investigado qué librerías de R serían las más indicadas. Las librerías más habituales y que permiten interacción al usuario serían ggplot2 y plotly

GGPLOT2 [10], es un paquete dedicado a la visualización de datos. Puede mejorar en gran medida la calidad y la estética de sus gráficos personalizando los atributos como textos, colores, forma, tamaño y lo hará mucho más eficiente al crearlos . GGPLOT2 se vincula con la biblioteca PLOTLY [11] la cual crea gráficos interactivos con calidad de publicación entre ellos diagramas de líneas, diagramas de dispersión, gráficos de áreas, gráficos de barras, barras de error, diagramas de caja, histogramas, mapas de calor, subtramas, múltiples ejes y gráficos 3D



Figura 11: Logos librerías ggplot2 y plotly

En el desarrollo del trabajo final, se va a investigar cómo crear una salida HTML en R, usando el paquete HTMLWidgets, el cual admite guardar en HTML la salida que fue creada por otros paquetes, como los que se usaran GGPLOT2 y PLOTLY. La salida se crea como un widget que se guarda en un archivo HTML y muy importante destacar que no existirán dependencias externas.

En el siguiente enlace se puede observar intrucciones útiles de ggplot2 que serán utilizadas en el proyecto.

<https://github.com/rstudio/cheatsheets/blob/master/data-visualization-2.1.pdf>

En el punto [12] de la bibliografía, se detalla la creación de gráficas para publicación con GGLOT2

### 3) Power BI & R

Algunas puntos genéricos a favor para tener en cuenta el uso de estos dos elementos serían:

- Licencias:
  - o R es gratuito. No necesitan piratear o comprar licencias de software comercial.
  - o Power BI es Freemium
- Soporte de comunidades/áreas:
  - o R tiene una comunidad académica detrás que provee una muy buena documentación en línea.
  - o R es empleado por investigadores de distintas áreas
  - o Power BI cuenta con una enorme comunidad.
  - o Power BI es implementarle en todas las áreas.
- Evolución
  - o Constantemente aparecen nuevos paquetes gratuitos que expande la capacidad de R para solucionar diferentes problemas
  - o R es uno de los paquetes estadísticos de mayor crecimiento en su uso en diferentes disciplinas.
  - o Power BI se actualiza todos los meses.
  - o Power BI ha crecido enormemente desde su inicio en junio 2015.
- Multiplataforma
  - o R es multiplataforma, es decir funciona en Mac, Windows o Linux
  - o Power BI es consumible de cualquier plataforma.

Analizando cómo puede interactuar Power BI con el lenguaje R vemos:

- Se puede ejecutar directamente código R en Power BI Desktop
- Importar conjuntos de datos resultantes en un modelo de datos
- Realizar transformaciones, en el caso de requerir alguna modificación, o para su visualización, con el editor de Power Query .
- Power BI también permite importar visualizaciones a través de un objeto visual que ya tengamos configurado previamente o también a través de su propia 'marketplace'.

Este último punto, se va a desarrollar con más detalle, ya que uno de los objetivos de este proyecto es la generación de un objeto visual y ponerlo a disposición de la comunidad en el Marketplace.

#### 4) Marketplace Power BI

Los elementos visuales de Marketplace son elementos visuales personalizados probados y aprobados por Microsoft y sus compañeros miembros de la comunidad. Estos elementos visuales personalizados se desarrollan y cargan en AppSource. Esto se realiza como un servicio comunitario para beneficiar a los usuarios de Power BI, ya que pueden explorar nuevos aspectos de sus datos mediante el uso de estos elementos visuales personalizados.

Por una lado Marketplace ofrece diferentes elementos que no están disponibles en el menú inicial de Power BI, por otro, da la ventaja que todos los usuarios obtienen la misma versión del objeto visual, que está controlada por el administrador de Power BI.

Se investiga qué objetos visuales existen actualmente en el Marketplace Power BI, el resultado en Marzo 2021 es de 322, de los cuales realizando un análisis se ha visto que hay únicamente 17 implementados en R, 3 en Python y el resto incorpora aplicaciones concretas. Por ejemplo Mapbox Studio, es una aplicación para administrar datos geoespaciales y diseñar estilos de mapas personalizados o mediante la creación de nuevos elementos a partir de TypeScript.

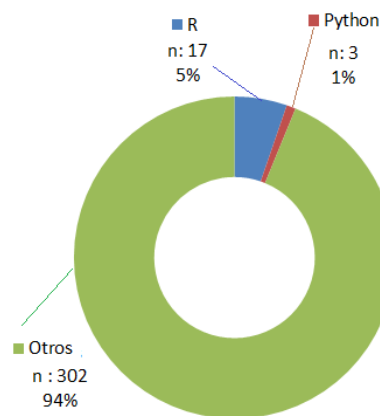


Figura 12: Elementos del Marketplace

En la siguiente URL se accede a la lista de elementos del Marketplace [13]

<https://appsource.microsoft.com/ca/marketplace/apps?product=power-bi-visuals>

[consultado marzo 2021].

También se ha estudiado, la galería de scripts en R que hay en Power BI, son gráficos personalizados que no necesitan estar publicados en el Marketplace, con lo que no hay información detallada de su diseño ni de su validación.

Concretamente, en la siguiente URL[14]:

[R Script Showcase - Microsoft Power BI Community](#) [consultada marzo-mayo 2021]

Se ha consultado los siguientes temas, para obtener conocimientos a nivel general:

- Aprender cómo funcionan y ver posibilidades de los gráficos en general en Power BI, realizados con otras tipologías diferentes a R.

Se adjunta una lista de los gráficos visuales que se muestran de ejemplo, en la URL[15]:

[Power BI visuals samples - Power BI / Microsoft Docs](#) [marzo-mayo 2021]

- Objetos visuales personalizados interactivos de Power BI con R[16]  
[Interactive Power BI custom visuals with R / by Freddie Tasker / Towards Data Science](#) [consultada marzo-mayo 2021]
- Cómo utilizar R Script (no a través de custom visual) en Power BI [17]  
[R Pubs - Utilizando R en Power BI](#) [consultada marzo-mayo 2021]
- Estudio avanzado de una aplicación de una Artificial Neural Network (ANN) en Power BI adaptando el lenguaje de programación R. [18]  
[Aplicación de una Artificial Neural Network \(ANN\) en Power BI adaptando el lenguaje de programación R - CepoBIA](#) [consultada marzo-mayo 2021]
- Estudio de buenas prácticas (best practice), sobre la visualización de datos. Con este documento se obtienen ideas y métodos que ayudan a diseñar una herramienta más enfocada a las necesidades del usuario final.[19]

##### 5) Otros puntos a investigar y resolución del estudio

En Power BI, uno de los problemas que pueden limitarnos a la hora de trabajar es la transformación de datos en Power BI ya que no tiene una biblioteca de funciones para la ciencia de datos, estadística, PNL y muchas otras áreas, pero queda solventado ya que el editor de consultas de Power BI admite la ejecución de scripts de R para transformar datos.

Analizando toda la información que se ha ido extrayendo no debería existir problemas en demostrar la factibilidad del uso R en Power BI y sus beneficios. Pero, cierto es, que hace falta estudiar bien, si existe un motivo por el cuál no se encuentra muchos customs visuals en el mercado, es aquí el aspecto que se debe abordar y/o demostrar que no debería existir dicha limitación. Quizás se podría lograr dando más recursos a las personalizaciones en los customs visuals.

Dónde también hará falta más estudio es en concreto con el paquete PLOTLY, si bien, existen evidencias de su incorporación con Power BI, se ha detectado algunas opiniones de desarrolladores donde describen algunos problemas, aunque ya han sido solventadas gracias a la mejora continua, pero ara falta ver si en el proyecto se detecta algún problema de incompatibilidad.

Por ejemplo, uno de los principales problemas, que ya se ha resuelto, era que anteriormente, solo permitía convertir la gráfica en formato png, y esta trama no era interactiva, ahora ya permite la generación del archivo html, con sus correspondientes ventajas de interacción con el usuario.

## 3. Diseño e implementación

Se sabe que los datos van incrementándose exponencialmente. Cada año que avanza, las empresas se centran más en los datos y en cómo sacarles valor para vender más productos, conseguir más clientes, aumentar la eficiencia del proceso, etc.

A menudo se tienen necesidades muy específicas de visualización y utilización de los datos que no pueden ser explotados a no ser que se tengan conocimientos en programación.

Como se ha comentado, uno de los objetivos de este proyecto es poder aportar a la comunidad una manera de trabajar con desarrollos realizados con el lenguaje estadístico R, sin la necesidad de conocer el lenguaje, con lo que no será necesario programar y se podrá lograr obtener los gráficos estadísticos que se generan gracias a librerías con las que se va a trabajar, estos gráficos se podrán utilizar en Microsoft Power BI, una herramienta de visualización que es capaz de manejar Big Data y al mismo tiempo cambiar la forma en que interactuamos con los gráficos. Esta aportación se va a realizar creando un objeto visual que se podrá incluir en los informes que se diseñen.

Este capítulo se dividirá en las siguientes secciones:

- 3.1. Parte técnica
- 3.2. Instrucciones de uso del custom visual en Power BI
- 3.3. Requisitos y limitaciones de los paquetes R
- 3.4. Consideraciones al desarrollador del custom visual
- 3.5. Publicación objeto visual: Se describe cómo realizar la publicación de un objeto visual al universo de Power BI, para que pueda ser utilizado por toda la comunidad.
- 3.6. Repositorio código

### 3.1. Parte técnica:

En la parte técnica se definirán:

- 3.1.1. Prerrequisitos iniciales a instalar
- 3.1.2. Creación estructura del custom visual
- 3.1.3. Análisis y diseño del custom visual

#### 3.1.1. Prerrequisitos iniciales a instalar

La creación del custom visual requiere de unos preparativos. En esta parte técnica se indica las herramientas y operaciones que se deben instalar para poder desarrollar un custom visual. Una vez preparado el entorno, se podrá ejecutar las instrucciones que crearán la estructura que se genera por defecto.

**a) Power BI Desktop**

Power BI Desktop, es una aplicación gratuita que se puede instalar en el equipo local y que permite conectarse a los datos, transformarlos y visualizarlos

<https://Power BI.microsoft.com/es-es/desktop/>



**b) Instalación de node.js**

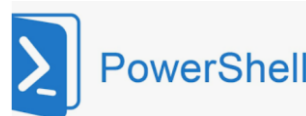
Node.js es un entorno de ejecución de JavaScript basado en la versión 8 del motor de JavaScript de Chrome. Permite a los desarrolladores ejecutar cualquier aplicación creada en JavaScript

<https://nodejs.org/en/>



**c) PowerShell**

La versión 4 de Windows PowerShell o una posterior para los usuarios de Windows



o Terminal para los usuarios de OSX.

**d) Instalar pbiviz**

La herramienta pbiviz, compila el código de origen del objeto visual del paquete pbiviz que es un proyecto de objeto visual de Power BI comprimido, con todos los scripts y recursos necesarios.

Abrir PowerShell y escribir: `npm i -g Power BI-visuals-tools`

**e) Crear e instalar un certificado.**

Para que el equipo y el servidor (el servicio Power BI) interactúen de forma segura, se requiere un certificado de Capa de sockets seguros (SSL). Sin un certificado para garantizar la seguridad de las interacciones, el explorador las bloqueará.

En PowerShell, escribir: `pbiviz --install-cert`

Para ver el detalle se puede consultar la siguiente página web:

<https://docs.microsoft.com/es-es/power-bi/developer/visuals/environment-setup?tabs=windows>

**f) Configurar servicio Power BI**

Configurar el servicio Power BI para desarrollar un objeto visual. Se debe conectar a la dirección:

<https://www.powerbi.com>

Y en el apartado de configuración activar la casilla:

*<Habilitar la depuración de objetos visuales personalizados con el objeto visual de desarrollador >*

**g) Instalar RStudio-R**

Se realiza el diseño y las pruebas de los programas realizados en R, en alguna herramienta de entorno de desarrollo integrado, en este caso se utiliza RStudio.

*<https://www.rstudio.com/products/rstudio/download/>*

**h) Habilitar objetos R**

Después de instalar R, Power BI Desktop lo habilita de forma automática. Si existe más de una instalación local en R, debemos indicar la ruta de acceso que nos interesa en las opciones de Power BI Desktop

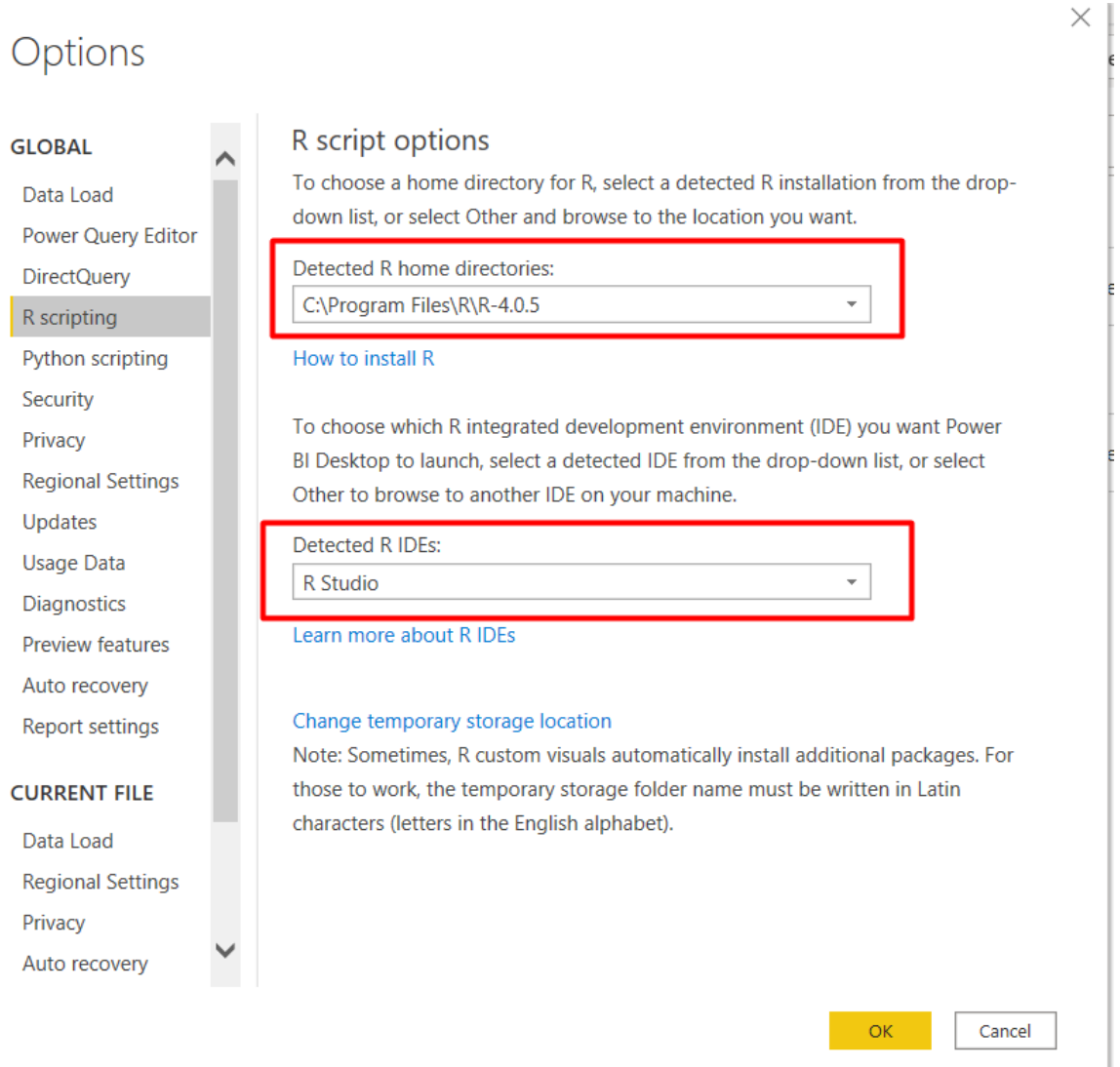


Figura 13: Opciones activación R

Nota importante: Para utilizar R en **Power BI Desktop**, se necesita tener R instalado en el equipo local. Pero cuando los usuarios quieren ver un objeto visual con tecnología R en el **servicio Power BI**, no necesitan tener instalado R localmente.

### i) Visual Studio Code (VS Code).

VS Code es un entorno de desarrollo integrado (IDE) gratuito, ideal para desarrollar aplicaciones JavaScript y TypeScript.

<https://code.visualstudio.com/>

La herramienta Visual Studio Code, será dónde se desarrollará la gran parte del proyecto, su estructura es la que se puede ver en la imagen <Estructura VSC>, que es la que se origina al crear el objeto visual. Para acceder a ella, se debe indicar el directorio dónde se han generado los elementos. En este proyecto el directorio se llama → RBOXSCATTER

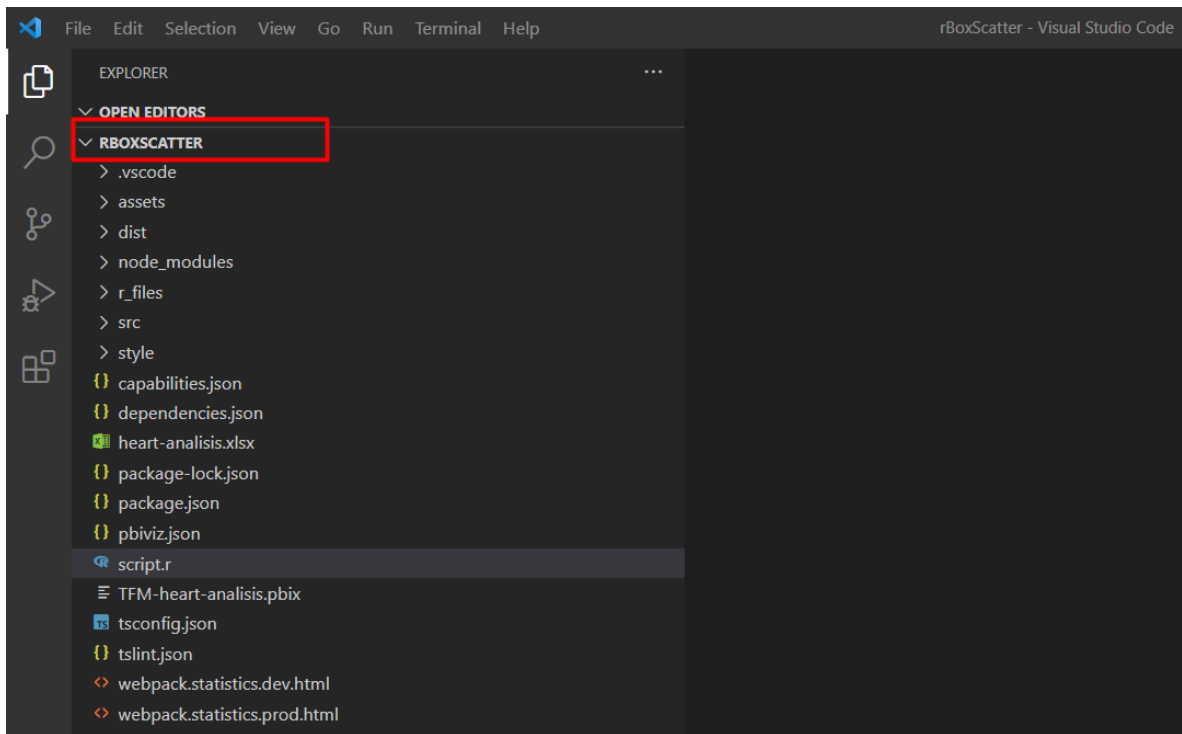


Figura 14: Estructura VSC

### 3.1.2. Creación estructura del custom visual

El objeto visual está diseñado en base a un esquema de carpetas. Para generar dicho esquema, se necesita ejecutar una serie de comandos, previamente se debe pensar en que tecnología se quiere crear el objeto visual:

- Basado en PNG (Retorna una imagen sin respuesta al desplazamiento del mouse)
- Basado en HTML (Retorna un gráfico que sí responde al desplazamiento del mouse, se puede acercar el zoom, etc.,)

Se muestra cómo se ejecutan las dos opciones, pero en este trabajo, se optará por la que está basada en HTML, debido a que aportará mucha más interacción al usuario.



### a) En PowerShell

Se ejecuta en PowerShell o terminal, las siguientes líneas de comando. (según tecnología PNG o HTML)

El paquete *pbiviz* es un proyecto de objeto visual de Power BI comprimido, con todos los scripts y recursos necesarios.

- ✓ PNG
  - >cmd
  - >pbiviz new rVisualSample -t rvisual
  
- ✓ HTML
  - >cmd
  - >pbiviz new rVisualSample -t rhtml

### b) Estructura archivos por defecto

La herramienta *pbiviz* crea el siguiente esquema de archivos:

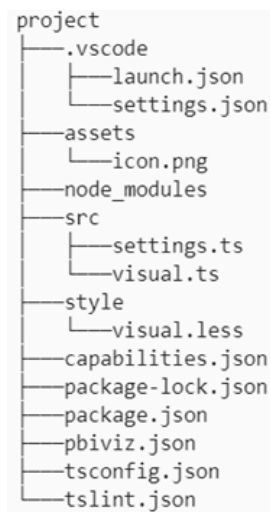


Figura 15: Estructura archivos

Descripción archivos:

**.vscode** : Esta carpeta se utiliza para configurar el área de trabajo, en ella se encuentra el archivo → **settings.json**.

Se puede observar los tres directorios principales dónde se distribuye el esquema del proyecto:

```

1 {
2   "editor.tabSize": 4,
3   "editor.insertSpaces": true,
4   "files.eol": "\n",
5   "files.watcherExclude": {
6     "**/.git/objects/**": true, "**/node_modules/**": true, ".tmp": true
7   },
8   "files.exclude": {
9     ".tmp": true
10  },
11  "files.associations": {
12    "*.resjson": "json"
13  },
14  "search.exclude": {
15    ".tmp": true,
16    "typings": true
17  },
18  "json.schemas": [
19    {
20      "fileMatch": [
21        "/pbiviz.json"
22      ],
23      "url": "./node_modules/powerbi-visuals-api/schema.pbiviz.json"
24    },
25    {
26      "fileMatch": [
27        "/capabilities.json"
28      ],
29      "url": "./node_modules/powerbi-visuals-api/schema.capabilities.json"
30    },
31    {
32      "fileMatch": [
33        "/dependencies.json"
34      ],
35      "url": "./node_modules/powerbi-visuals-api/schema.dependencies.json"
36    }
37  ]
38 }

```

-/pbiviz.json  
 - /capabilities.json  
 -/dependencies.json

Figura 16: settings.json

**pbiviz.json** Contiene los metadatos del objeto visual.

```

{} pbiviz.json > ...
1 {"visual":
2   {"name": "rBoxScatter",
3     "displayName": "R_Box_Scatter",
4     "guid": "rBoxScatter0E79A72CF15D432082926D847A65F8D7",
5     "visualClassName": "Visual",
6     "version": "1.0.0",
7     "description": "R_Box_Sacatter",
8     "supportUrl": "http://community.powerbi.com/",
9     "githubUrl": ""
10  },
11  "apiVersion": "2.6.0",
12  "author": {"name": "Nuria Pont Vila", "email": "nuriapontvila@uoc.edu"},
13  "assets": {"icon": "assets/icon.png"},
14  "externalJS": null,
15  "style": "style/visual.less",
16  "capabilities": "capabilities.json",
17  "dependencies": "dependencies.json",
18  "stringResources": []
19 }

```

Figura 17: pbiviz.json

**capabilities.json** Este archivo contiene las propiedades y las configuraciones principales (o funcionalidades) del objeto visual. Permite que el objeto visual declare características, objetos, propiedades y asignaciones de vistas de datos admitidos.

**dependencies.json** El archivo contiene las librerías que se van a instanciar.

**.assets** : Esta carpeta contiene el archivo → **icon.png**. Será el icono que representará al custom visual.

**.src**: Esta carpeta contiene el código fuente del objeto visual.

**visual.ts**: el código fuente principal del objeto visual. En este archivo se puede ver los procedimientos genéricos para poder generar los elementos específicos del objeto visual. No se debe confundir con el archivo `script.r`, que será el que contenga la programación específica del custom visual.

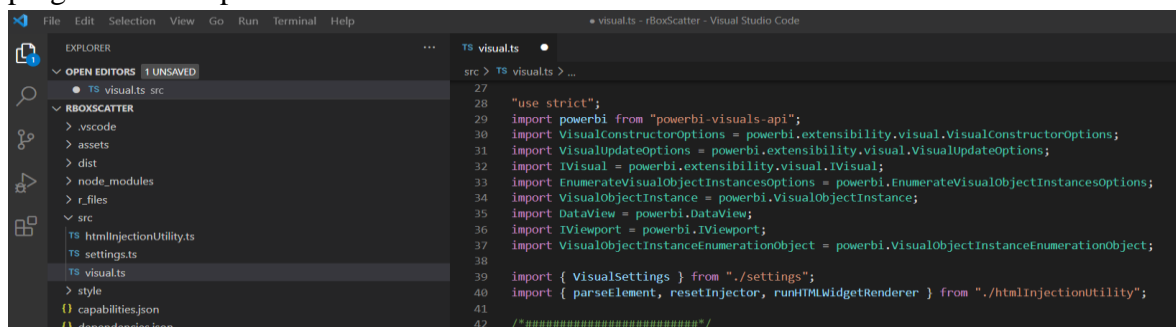


Figura 18: visual.ts

**settings.ts**: el código de la configuración del objeto visual. Las clases del archivo proporcionan una interfaz para definir las propiedades del objeto visual.

**.style** Esta carpeta contiene el archivo `visual.less`, que incluye los estilos del objeto visual.

**package-lock.json** Este archivo se genera automáticamente para las operaciones donde `npm` modifica el árbol `node_modules` o el archivo `package.json`.

**package.json** Este archivo describe el paquete del proyecto. Contiene información sobre el proyecto, como los autores, la descripción y las dependencias del proyecto.

**tsonfig.json** Un archivo de configuración para TypeScript.

Este archivo debe contener la ruta de acceso al archivo `*.ts` donde está ubicada la clase principal del objeto visual, como se especifica en la propiedad `visualClassName` del archivo `pbviz.json`.

**tslint.json** El archivo contiene la configuración de TSLint.

**script.r**: Cuando se usa `pbviz` para crear el objeto visual con tecnología de R basado en la plantilla `rvisual`, se crea un archivo denominado `script.r` en la carpeta raíz del objeto visual. En este archivo se definen las funciones y procedimientos que van a

poder utilizarse en el custom visual, por ejemplo es dónde se define la generación del gráfico seleccionado.

### 3.1.3. Análisis y Diseño del custom visual

En este apartado se desarrolla cómo y dónde se ha ido añadido y actualizando el código necesario para incorporar los diferentes parámetros y funciones que estarán disponibles para el usuario final en los diseños que genere en los informes Power BI.

El núcleo de trabajo está basado en las librerías de R, *ggplot2* y *plotly*.

GGPLOT2 permite de manera sencilla crear gráficos complejos a partir de un conjunto de datos. Este paquete provee una estructura para que se pueda especificar qué variables graficar, cómo deben ser presentadas y otras propiedades visuales generales. De esta manera, sólo se necesita realizar cambios mínimos si los datos sufren alguna modificación o si se decide pasar, por ejemplo, de un gráfico de barras a un diagrama de dispersión. Esto ayuda a crear gráficos con calidad para publicación con pocos ajustes adicionales.

PLOTLY es una biblioteca de gráficos interactiva, de código abierto, construida sobre plotly.js, son gráficos declarativos de alto nivel, incluidos gráficos científicos, gráficos 3D, gráficos estadísticos, mapas SVG, gráficos financieros y más.

La información de los elementos del diseño se estructurará, según las siguientes secciones:

- a) Configuración inicial
- b) Referencia a las librerías R
- c) Elementos de los ejes
- d) Creación de menús
- e) Lista de gráficos
- f) Elementos de menú globales
- g) Elementos de menú concretos
- h) Etiquetas/Labels
- i) Icono custom visual
- j).Compilación

#### a) Configuración inicial

La configuración inicial se indica en el archivo pvibiz.

- Pvibiz

Es necesario especificar los datos seleccionados para poder compilar correctamente el objeto, el resto de datos son los que se generan por defecto al crear el custom visual.

Datos necesarios:

- ❖ Descripción
- ❖ URL de soporte
- ❖ Autor
- ❖ email

```
pbiviz.json > ...
1  {"visual":{"name":"rBoxScatter","displayName":"R_Box_Scatter",
2    "guid":"rBoxScatter0E79A72CF15D432082926D847A65F8D7","visualClassName":"Visual","version":"1.0.0",
3    "description":"R_Box_Scatter","supportUrl":"www.google.com","githubUrl":""},
4    "apiVersion":"2.6.0",
5    "author":{"name":"Nuria Pont Vila","email":"nuriapontvila@uoc.edu"},
6    "assets":{"icon":"assets/icon.png"},"externalJS":null,"style":"style/visual.less",
7    "capabilities":"capabilities.json","dependencies":"dependencies.json","stringResources":[]}
8
```

Figura 19: pvibiz

## b) Referencias a las librerías R

En el archivo dependencies.json, se define la ubicación de las librerías que se van a utilizar → ggplot2 y plotly

Las librerías seleccionadas se buscarán en Comprehensive R Archive Network (CRAN) que es el repositorio de software central de R, respaldado por R Foundation. Contiene un archivo de las versiones anteriores y más recientes de la distribución de R, la documentación y los paquetes de R contribuidos.

```
dependencies.json > ...
{
  "cranPackages": [
    {
      "name": "ggplot2",
      "displayName": "GG Plot 2",
      "url": "https://cran.r-project.org/web/packages/ggplot2/index.html"
    },
    {
      "name": "plotly",
      "displayName": "Plotly",
      "url": "https://cran.r-project.org/web/packages/plotly/index.html"
    },
    {
      "name": "htmlwidgets",
      "displayName": "HTML Widgets",
      "url": "https://cran.r-project.org/web/packages/htmlwidgets/index.html"
    },
    {
      "name": "XML",
      "displayName": "XML",
      "url": "https://cran.r-project.org/web/packages/XML/index.html"
    }
  ]
}
```

Figura 20: dependencies.json

### c) Elementos de los ejes (dimensiones)

Para crear los gráficos, es necesario indicar los datos que los van a alimentar. Se ha preparado el custom para trabajar según los tres ejes de coordenadas cartesianas, el X, Y y Z.

Nota: Se remarca, que para este proyecto se ha optado por estas especificaciones, pero podría fácilmente ampliarse, o denominarse diferente, por ejemplo podría incluirse un nuevo elemento que se denominara 'leyenda'.

Los archivos implicados son:

- capabilities.json
- script.r

- capabilities.json

Para cada opción que se quiera tratar, se debe cumplimentar las secciones:

- ❖ dataRoles (Definición de los campos de datos que espera el objeto visual)
  - name: nombre interno de este campo de datos
  - kind: tipo de campo:
    - Grouping: valores discretos que se usan para agrupar campos de medida.
    - Measure: valores de datos numéricos.
    - GroupingOrMeasure: valores que se pueden usar como una agrupación o una medida.
  - displayName: nombre que se muestra al usuario en el panel
- ❖ dataViewMappings (Definición de la forma en que se quieren asignar los datos)

En este proyecto, por ejemplo se ha definido que sólo se quiere un elemento para cada eje, pero se cómo se puede ver, se puede modificar fácilmente en caso que las especificaciones lo requieran.
- ❖ scriptResult (recuperación de los valores)

```
"dataRoles": [  
  {  
    "displayName": "eje_x",  
    "description": "eje_x",  
    "kind": "GroupingOrMeasure",  
    "name": "eje_x"  
  }  
]
```

```
"dataViewMappings": [  
  {  
    "conditions": [  
      {  
        "eje_x": {
```

```
"max": 1
},
```

```
"scriptResult": {
  "dataInput": {
    "table": {
      "rows": {
        "select": [
          {
            "for": {
              "in": "eje_x"
            }
          }
        ],
      }
    }
  },
}
```

Figura 21: Código desarrollo opciones

- script.r

Este archivo tratará el cuerpo del custom visual.

Se inicializa el dataset con los elementos seleccionados.

```
dataset = data.frame(eje_x, eje_y,eje_z)
```

En este proyecto, se realizará una validación, dónde se comprobará que al menos se ha seleccionado los elementos X e Y. Normalmente, en este punto, el programa daría un mensaje genérico de error como el siguiente:

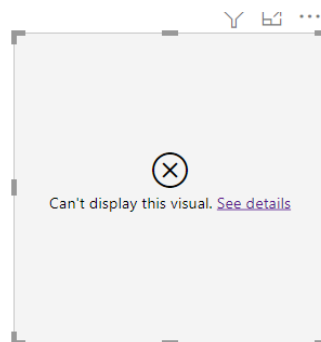


Figura 22: Imagen error por defecto

Si ha querido modificar esta imagen y en caso de no haber introducido los datos, se muestra la siguiente imagen:

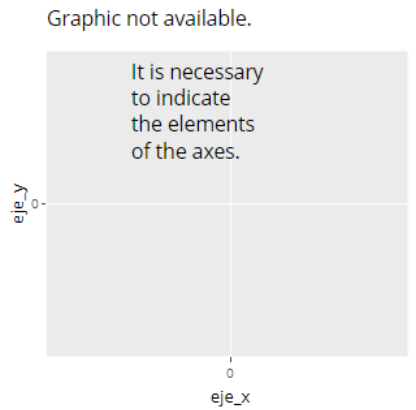


Figura 23: Imagen mensaje informativo

El siguiente código, muestra cómo se ha realizado, comprobando si los datos han sido seleccionados.

```

if (!exists("eje_x") || !exists("eje_y") )
{
  eje_x=0
  eje_y=0
  eje_z=0

  dataset=data.frame(eje_x,eje_y)

  p2=ggplot(data = dataset, aes(x = eje_x, y = eje_y))+ scale_x_continuous(breaks=seq(0, 1, 1))+ scale_y_continuous(breaks=seq(0, 1, 1))

  It is necessary
  to indicate
  the elements
  of the axes.")
  p2 = ggplotly(p2);
  internalSaveWidget(p2, 'out.html');
  stop_work==TRUE
}

```

Figura 24: Código validación elementos

#### d) Creación de menús

Con el objetivo de facilitar el uso de la herramienta al desarrollador de informes, se ha estructurado el custom en diferentes carpetas según funcionalidades.



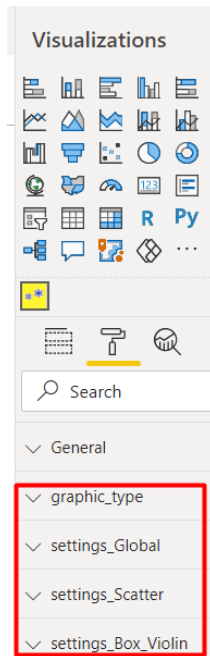


Figura 25: Menú custom visual

Los archivos implicados son:

- capabilities.json
- settings.ts

- capabilities.json

En este archivo se define con el tipo 'objects', cada una de las carpetas que se quieren generar.

En este custom se generarán las siguientes:

- ❖ graphic\_type
- ❖ settings\_Global
- ❖ settings\_Scatter
- ❖ settings\_Box\_Violin

```
"objects": {  
  "graphic_type": {  
    "properties": {  
      "graphic_type": {  
        "displayName": "graphic_type",  
        "description": "graphic_type",  
        "type": {  
          "type": "..."  
        }  
      }  
    }  
  }  
}
```

Figura 26: Objects

```
"settings_Global": {
  "displayName": "settings_Global",
  "properties": {
    "show": {
      "type": {"bool": true}
    },
    "color": {

```

Figura 27: settings\_Global

```
"settings_Scatter": {
  "properties": {
    "tipo_Scatter": {

```

Figura 28: settings\_Scatter

```
"settings_Box_Violin": {
  "properties": {

```

Figura 29: settings\_Box\_Violin

- settings.ts

Este archivo indica los parámetros que se quieren mostrar en cada carpeta y el valor por defecto.

En el recuadro verde se generan las carpetas, en los recuadros azules los parámetros implicados.

```
"use strict";

import { dataViewObjectsParser } from "powerbi-visuals-utils-dataviewutils";
import DataViewObjectsParser = dataViewObjectsParser.DataViewObjectsParser;

export class VisualSettings extends DataViewObjectsParser {
  public graphic_type: graphic_type = new graphic_type();
  public settings_Global: settings_Global = new settings_Global();
  public settings_Scatter: settings_Scatter = new settings_Scatter();
  public settings_Box_Violin: settings_Box_Violin = new settings_Box_Violin();
  public rcv_script: rcv_scriptSettings = new rcv_scriptSettings();
}

export class graphic_type {
  public graphic_type: string = "boxplot";
}

export class settings_Global {
  public colour: string = "dodgerblue";
  public themes: string = "theme_bw";
  public size: number = 1;
  public fill: string = "RdPu";
  public Disposicion: string = "Vertical";
}

export class settings_Scatter {
  public metodo = "loess";
}

export class settings_Box_Violin {
  public jitter: string = "FALSE";
  public notch: string = "FALSE";
  public trim: string = "TRUE";
}
}
```

Figura 30: settings.ts

Dando como resultado:

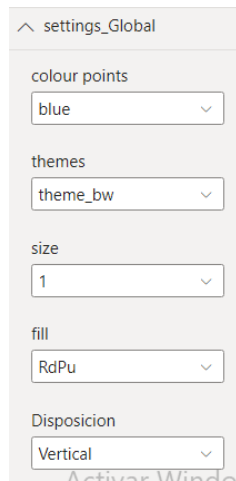


Figura 31: Menú objeto visual-settings\_Global

### e) Lista de gráficos

Los archivos implicados son:

- capabilities.json
- settings.ts
- script.r
- capabilities.json

En la carpeta de tipos de gráficos se define la siguiente lista:

```
"objects": {
  "graphic_type": {
    "properties": {
      "graphic_type": {
        "displayName": "graphic_type",
        "description": "graphic_type",
        "type": {
          "enumeration": [
            {
              "displayName": "violin", "value": "violin"
            },
            {
              "displayName": "boxplot", "value": "boxplot"
            },
            {
              "displayName": "violin&boxplot", "value": "both"
            },
            {
              "displayName": "scatter_point", "value": "scatter_point"
            },
            {
              "displayName": "scatter_line", "value": "scatter_line"
            },
            {
              "displayName": "Histogram", "value": "Histogram"
            },
            {
              "displayName": "Bar", "value": "Bar"
            },
            {
              "displayName": "Bar_2vars(Separadas)", "value": "Bar_2vars(Separadas)"
            },
            {
              "displayName": "Bar_2vars", "value": "Bar_2vars"
            },
            {
              "displayName": "scatter3d mesh3d", "value": "scatter3d mesh3d"
            },
            {
              "displayName": "scatter3d lines", "value": "scatter3d lines"
            },
            {
              "displayName": "scatter3d markers", "value": "scatter3d markers"
            }
          ]
        }
      }
    }
  }
}
```

Figura 32: capabilities.json - gráficos

Nota: Para el tipo scatter, se podría haber realizado un único punto y posteriormente en la carpeta de settings\_Scatter indicar la modalidad requerida, histograma o diagrama de barras, pero para facilitar su uso, se ha creado directamente en el punto de selección indicando el 'subtipo' de scatter.

A continuación se indica brevemente cada uno de ellos:

- ❖ **“Boxplot”**: diagrama también nombrado como diagrama de caja y bigotes es un método estandarizado para representar gráficamente una serie de datos numéricos a través de sus cuartiles. El Cuadro de Recuadro y Bigote o Box and Whisker Chart visualiza un conjunto de datos en un resumen de cinco números (media, mediana, cuartiles, min/máx). Este es una visual interesante porque proporciona una visión que permanece oculta cuando sólo miramos las estadísticas de resumen. Estas pueden sugerir que todo está bien, aunque en realidad hay cosas que necesitan ser optimizadas.
- ❖ **“Violín”**: diagrama que se utiliza para visualizar la distribución de los datos y su densidad de probabilidad. Este gráfico es una combinación de un diagrama de cajas y bigotes y un diagrama de densidad girado y colocado a cada lado, para mostrar la forma de distribución de los datos.

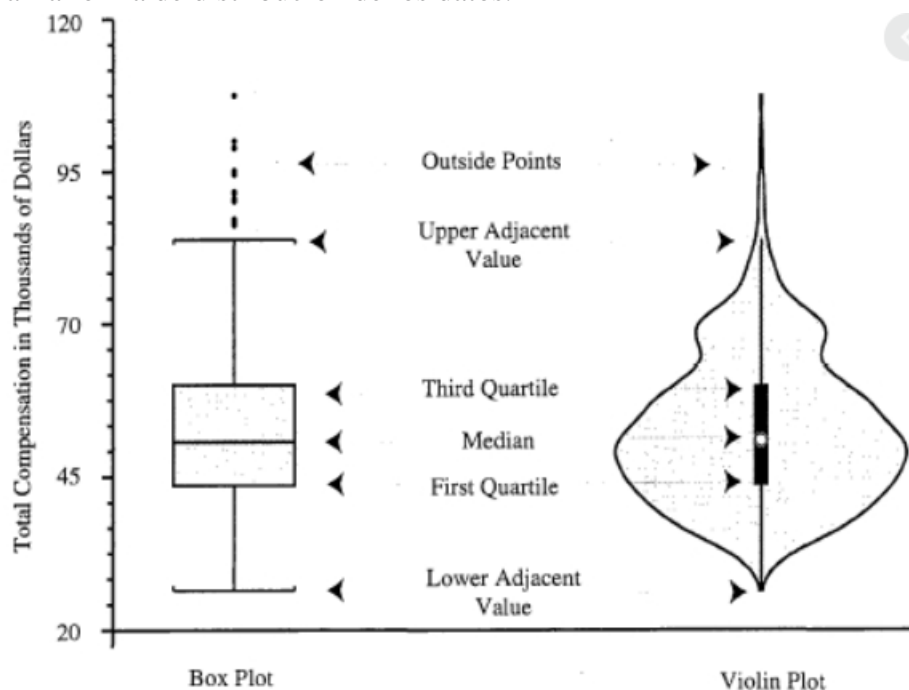


Figura 33: BoxPlot/Violin [20 ]

- ❖ **“Violin&Boxplot”**: Combina la funcionalidad del box plot -mostrando los cuartiles de la distribución- y la de la estimación de densidad kernel.
- ❖ **“scatter\_point”**: Hace referencia al tipo de diagrama matemático de puntos o gráfico de dispersión, el cual utiliza las coordenadas cartesianas para mostrar los valores de dos variables para un conjunto de datos.

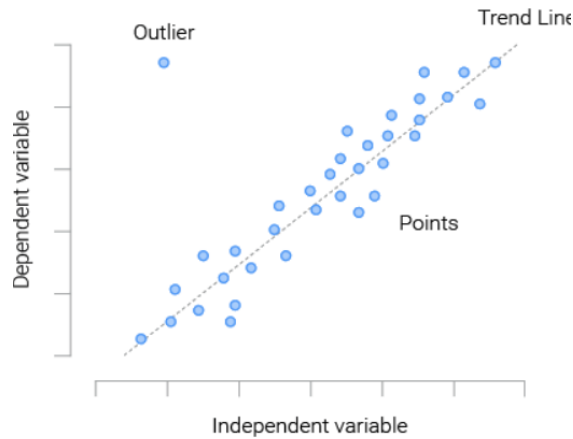


Figura 34: Scatter point

❖ **“scatter\_line”**

Lo mismo que el anterior, unificando las líneas. No es muy usado, pero se muestra como ejemplo de las posibilidades que se pueden realizar en el propio custom visual que se está creando, pero también para mostrar la multitud de opciones que se pueden generar en un nuevo objeto visual.

❖ **“Histograma”**

En estadística, un histograma es una representación gráfica de una variable en forma de barras, donde la superficie de cada barra es proporcional a la frecuencia de los valores representados.

❖ **“Bar”**

Un gráfico de barras o gráfico de columnas (conocido como *bar* en inglés), es una forma de representar gráficamente un conjunto de datos o valores mediante barras rectangulares de longitud proporcional a los valores representados. Un diagrama de barras se utiliza para representar datos cualitativos o datos cuantitativos de tipo discreto.

Nota: la diferencia entre gráfico de barras e histograma: el segundo tiene las barras juntas y sin espacio porque representa una variable continua. En cambio, el gráfico de barras las tiene separadas porque representa la frecuencia de categorías

## Histogram vs. Bar Chart

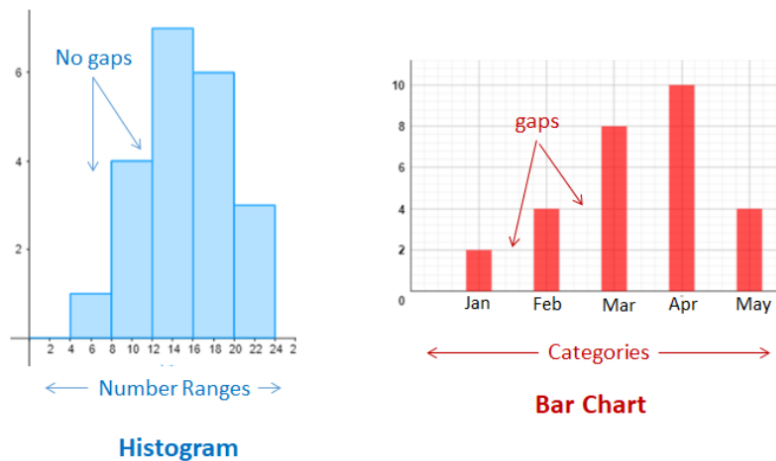


Figura 35: Histograma vs Bar

### ❖ “*Bar\_2vars(Separadas)*”

Gráfico para representar dos variables en forma BAR, este caso se muestran de forma separada.

### ❖ “*Bar\_2vars*”

Ídem, caso anterior pero de forma apilada. Se verá cada categoría en un color diferente, pero todas en la misma columna

### ❖ “*scatter3d mesh3d*”

En general, una gráfica de dispersión 3D se usa para explorar las relaciones potenciales entre tres variables. Normalmente los ejes X y Y son las variables predictoras y la Z la respuesta. Por ejemplo si queremos saber si la edad y el colesterol afectan a una ataque cardíaco.

Hay diferentes formas de mostrar las dimensiones, con mesh3d da un formato voluminoso, dibuja un conjunto 3D de triángulos con vértices dados por x, y y z

### ❖ “*scatter3d lines*”

Relaiza la representación uniendo los puntos

### ❖ “*scatter3d markers*”

Dibuja el espacio tridimensional.

Nota: Se puede girar una gráfica de dispersión 3D para apreciarla desde diferentes ángulos. Esto permite explorar más extensamente las relaciones entre las variables.

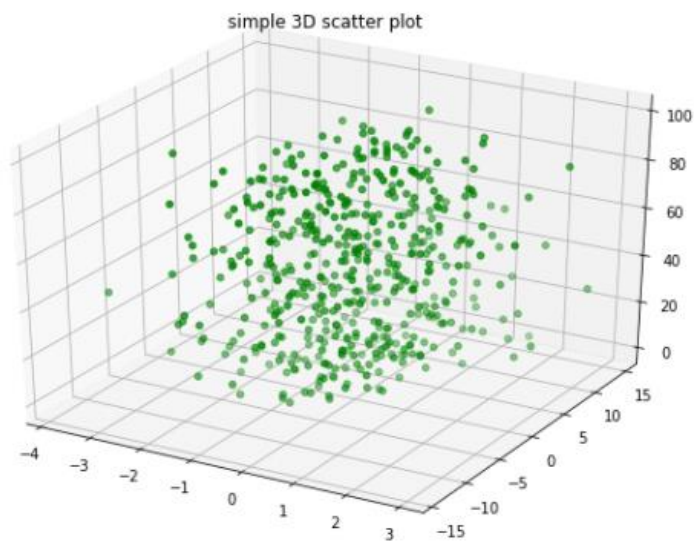


Figura 36: Scatter 3D

Mencionar que al utilizar la librería plotly, el scatter 3d permitirá rotar entre sus ejes:

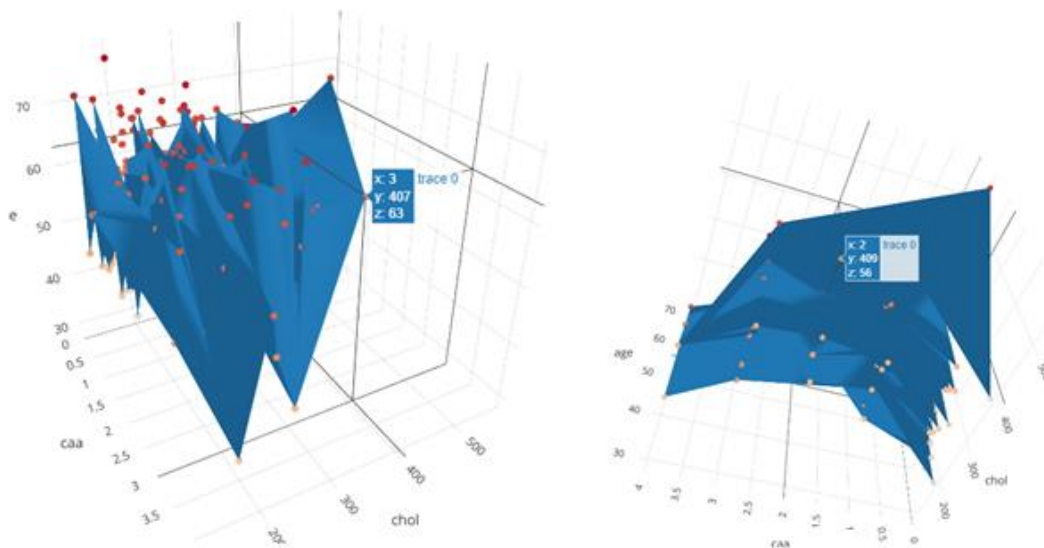


Figura 37: Scatter 3D rotation

#### f) Elementos de menú globales

Los archivos implicados son:

- capabilities.json
- settings.ts
- script.r

Es importante conocer previamente la estructura y la nomenclatura de los parámetros implicados en las funciones que se quieren tratar para que funcionen correctamente.

- **capabilities.json**

En este archivo se define la lista para cada uno de los parámetros que se quieren dar a escoger para su utilización en el custom visual.

Como parámetros globales para la utilización en todos los gráficos se ha seleccionado:

- ❖ color

El siguiente código indica los valores:

```
"settings_Global":{
  "displayName": "settings_Global",
  "properties": {
    "show": {
      "type": {"bool": true}
    },
    "colour":{
      "displayName": "colour points",
      "description": "colour points",
      "type": {
        "enumeration": [
          {
            "displayName": "red",
            "value": "firebrick2"
          },
          {
            "displayName": "green",
            "value": "chartreuse3"
          },
          {
            "displayName": "blue",
            "value": "dodgerblue"
          },
          {
            "displayName": "yellow",
            "value": "gold"
          }
        ]
      }
    }
  }
},
```

En este custom se ha propuesto 4 posibles colores, los más comunes: rojo, verde, azul y amarillo. Se puede ver en el código, que en la descripción existen estos valores, pero internamente se ha indicado unos tonos más suaves.

Esta variable se podrá utilizar para mostrar los colores de las observaciones, o por ejemplo, dar color a las rectas.

En este punto, se puede oberbar la facilidad para incrementar/disminuir esta lista, según las especificaciones que se detallen en el desarrollo del custom visual.

Figura 38: settings\_Global

- ❖ tamaño

Como en el caso anterior la lista podría ser más larga, para este proyecto se ha definido los siguientes valores.

```
"size":{
  "displayName": "size",
  "description": "size",
  "type": {
    "enumeration": [
      {
        "displayName": "0.5",
        "value": "0.5"
      },
      {
        "displayName": "1",
        "value": "1"
      },
      {
        "displayName": "2",
        "value": "2"
      },
      {
        "displayName": "4",
        "value": "4"
      },
      {
        "displayName": "6",
        "value": "6"
      }
    ]
  }
},
```

Figura 39: settings-size



#### ❖ tema

Los temas son una forma de personalizar los componentes que no son datos de sus gráficos: es decir, títulos, etiquetas, fuentes, fondo, líneas de cuadrícula y leyendas. Los temas se pueden utilizar para dar a las tramas un aspecto personalizado coherente.

Se ha preparado la siguiente lista para poder ser seleccionados:

```
"themes":{
  "displayName": "themes",
  "description": "themes",
  "type": {
    "enumeration": [
      {
        "displayName": "theme_gray", "value": "theme_gray"
      },
      {
        "displayName": "theme_linedraw", "value": "theme_linedraw"
      },
      {
        "displayName": "theme_bw", "value": "theme_bw"
      },
      {
        "displayName": "theme_dark", "value": "theme_dark"
      },
      {
        "displayName": "theme_minimal", "value": "theme_minimal"
      },
      {
        "displayName": "theme_classic", "value": "theme_classic"
      },
      {
        "displayName": "theme_void", "value": "theme_void"
      },
      {
        "displayName": "theme_test", "value": "theme_test"
      }
    ]
  }
},
```

En la siguiente URL, se detalla el contexto de cada una:

<https://ggplot2.tidyverse.org/reference/ggtheme.html>

Figura 40: settings-temas

#### ❖ relleno/fill

En muchas ocasiones, nos resulta visulamente muy útil que se diferencien categorías de una variable según colores. Del mismo modo, muchas veces vemos paletas que definen estos colores.

En este apartado, recordando la frase que se suele decir, *para gustos los colores*, se ha creído oportuno dar un amplio abanico. La lista original que se preparó era más reducida, pero a raíz de mostrar este proyecto a usuarios potenciales y ver puntos de posibles mejora, se ha repensado, un ejemplo lo ha indicado una persona que es daltónica y que ampliando las posibilidades de paletas, tiene más opciones de escoger y trabajar con los informes.

La lista final que se ofrece está basada en la paleta de colores “ColorBrewer” por Cynthia Brewer.:

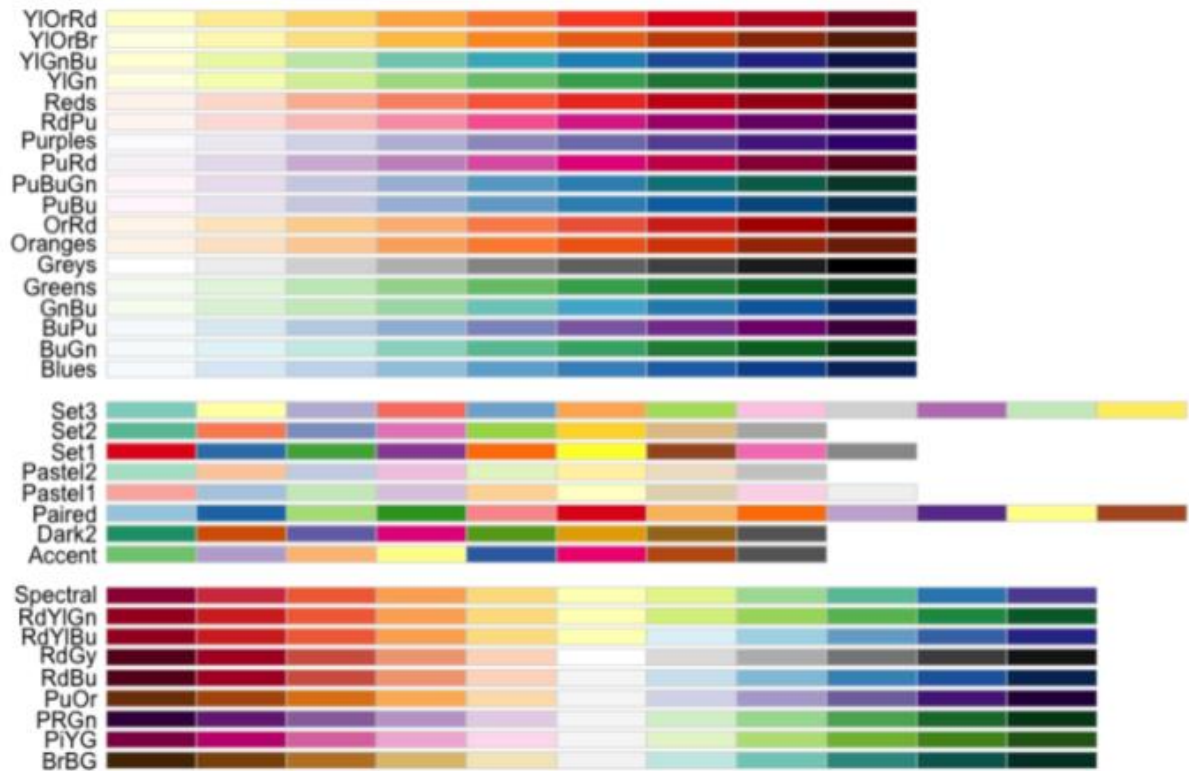


Figura 41: Paleta "ColorBrewer" por Cynthia Brewer

❖ Disposicion

Se podrá seleccionar, en los gráficos que sea pertinente, entre la opción vertical, por defecto, o la horizontal.

```

"Disposicion":{
  "displayName": "Disposicion",
  "description": "Disposicion",
  "type": {
    "enumeration": [
      {
        "displayName": "Horizontal",
        "value": "Horizontal"
      },
      {
        "displayName": "Vertical",
        "value": "Vertical"
      }
    ]
  }
},

```

Figura 42: Disposición

El resultado es el siguiente:

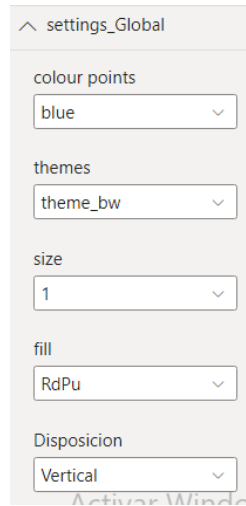


Figura 43: Menú objeto visual-settings\_Global

- **settings**

Este archivo contiene los parámetros que se quieren visualizar en Power BI, se harán públicos los menús y los valores por defecto.

En la siguiente imagen, se ha indicados dos ejemplos, el punto de menú settings\_Global y los elementos que tiene en su interior, color rojo y en azul, el punto de menú que hace referencia a sttings\_Box\_Violin.

```
"use strict";

import { dataViewObjectsParser } from "powerbi-visuals-utils-dataviewutils";
import DataViewObjectsParser = dataViewObjectsParser.DataViewObjectsParser;

export class VisualSettings extends DataViewObjectsParser {
  public graphic_type: graphic_type = new graphic_type();
  public settings_Global: settings_Global = new settings_Global();
  public settings_Scatter: settings_Scatter = new settings_Scatter();
  public settings_Box_Violin: settings_Box_Violin = new settings_Box_Violin();
  public rcv_script: rcv_scriptSettings = new rcv_scriptSettings();
}

export class graphic_type {
  public graphic_type: string = "boxplot";
}

export class settings_Global {
  public colour: string = "dodgerblue";
  public themes: string = "theme_bw";
  public size: number = 1;
  public fill: string = "RdPu";
  public Disposicion: string = "vertical";
}

export class settings_Scatter {
  public metodo = "loess";
}

export class settings_Box_Violin {
  public jitter: string = "FALSE";
  public trim: string = "TRUE";
}
```

Figura 44: settings

- **script.r**

Este archivo contiene la definición de cada uno de los gráficos con los parámetros que se pueden seleccionar.

La siguiente imagen describe la parte del código en que se define uno de los gráficos, en concreto se muestra el de histograma.

```
p=ggplot(dataset, aes(x=val, fill=val y)) +  
  geom_histogram(color=settings_Global_colour) +  
  geom_freqpoly(color=settings_Global_colour) +  
  xlab(n1) + ylab(n2) + labs(title = graphic_type graphic_type)+  
  scale_fill_brewer(palette=settings_Global_fill)  
if (settings_Global_Disposicion=='Horizontal')  
  {  
    p=p+coord_flip()  
  }
```

Figura 45: ejemplo código gráfico histograma

Y en la siguiente imagen, común a todos los gráficos, la orden de visualización:

```
##### Create and save widget #####  
p = ggplotly(p);  
  
internalSaveWidget(p, 'out.html');  
  
#####
```

Figura 46: Código resultado final

Cualquier widget creado a partir de cualquier paquete **htmlwidgets** (por ejemplo, **plotly**) se puede guardar como un archivo HTML independiente a través de la `htmlwidgets::saveWidget()` función. De forma predeterminada, produce un archivo HTML completamente autónomo, lo que significa que todos los archivos de dependencia de JavaScript y CSS necesarios se incluyen dentro del archivo HTML.

**g) Elementos de menú concretos**

En este proyecto se ha definido dos menús con características propias a cada tipo de gráfico seleccionado. En concreto se sugieren unas funciones para el tipo Box/Violin y otra para los que son gráficos de tipo Scatter

- ❖ Tipo Box/Violin

- Jitter → se dibuja un punto por cada observación, lo cual puede ayudar a entender mejor los datos que se tienen.
- Trim → Si TRUE(predeterminado), recorta las colas de los violines al rango de los datos. Si FALSE, no cortes las colas.

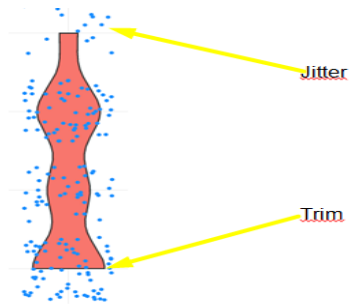




Figura 47: Viloin + jitter + trim

## h) Etiquetas/Labels

La nomenclatura de los ejes por defecto viene definida por el nombre de su etiqueta, es decir, *eje\_x*, *eje\_y*, pero se ve claramente que no es suficientemente descriptivo, al entregar un informe se necesita conocer a qué datos se está refiriendo, es por eso que se actualiza el valor con el nombre del campo de datos que utilizamos. Como se verá posteriormente, Power BI permite al diseñador del informe especificar otro nombre más concreto en caso de ser necesario.

## i) Icono custom visual

Por defecto el icono que se ofrece es el siguiente: 

Ha sido remplazado por: 

Archivo implicado:

- **pbviz.json** es dónde se indica la ubicación del icono

```

{} pbviz.json > ...
1  {"visual":{"name":"rBoxScatter","displayName":"R_Box_Scatter",
2    "guid":"rBoxScatter0E79A72CF15D432082926D847A65F8D7","visualClassName":"Visual","version":"1.0.0",
3    "description":"R_Box_Scatter","supportUrl":"www.google.com","githubUrl":""},
4    "apiVersion":"2.6.0",
5    "author":{"name":"Nuria Pont Vila","email":"nuriapontvila@uoc.edu"},
6    "assets":{"icon":"assets/icon.png"} "externalJS":null,"style":"style/visual.less",
7    "capabilities":"capabilities.json","dependencies":"dependencies.json","stringResources":[]}
8
  
```

Figura 48: Código Icon

En la siguiente imagen, podemos ver el icono en la estructura :

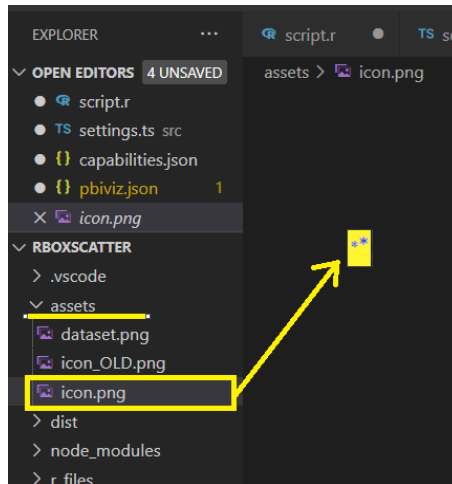


Figura 49: Icon

## j) Compilación

Este punto se realiza en el terminal del VSC.

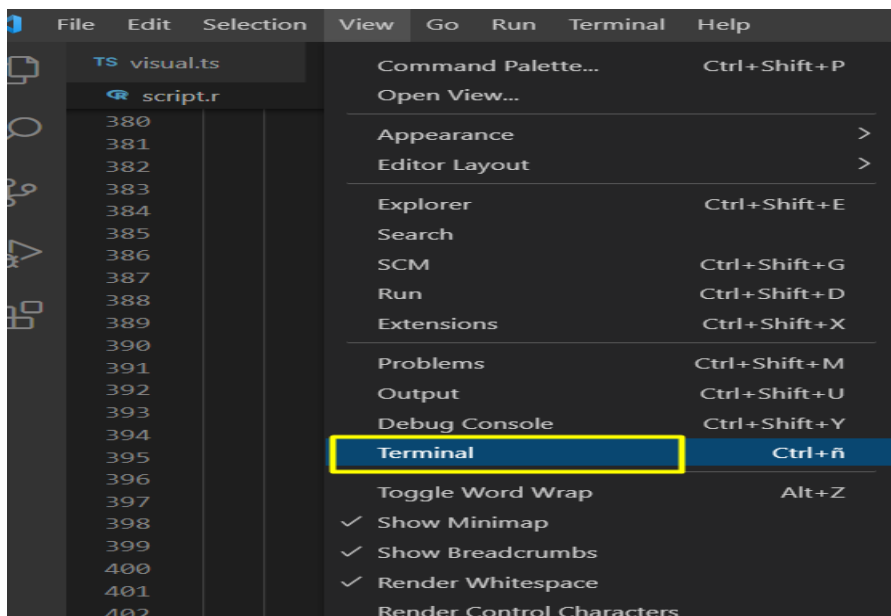


Figura 50: VSC-Terminal

En la siguiente línea de comandos se indicará la instrucción →  
*pbiviz Package*

```
PS E:\Nuria\UOC\PowerBI\CustomVisuals\rBoxScatter> pbiviz package
```

El resultado esperado es:

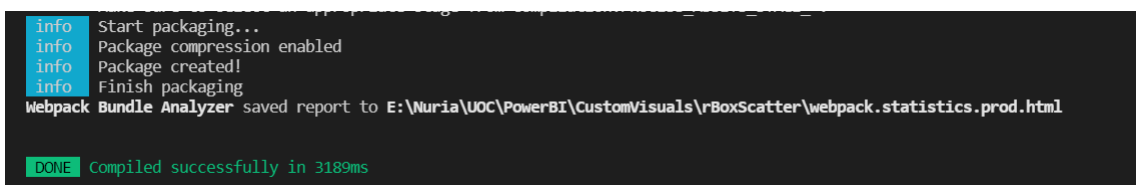


Figura 51: Compilación

A partir de este momento el objeto visual estará disponible para su uso. El objeto se ubica en la carpeta dist.

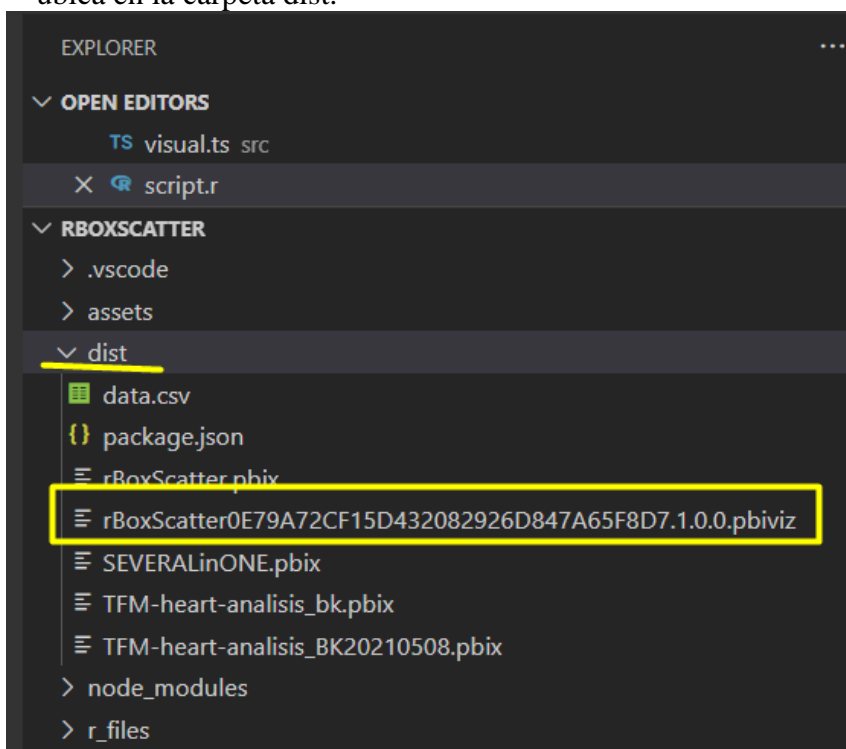


Figura 52: Paquete objeto visual

## 3.2. Instrucciones de uso del custom visual en Power BI

En este apartado se indica los pasos a seguir para la utilización del custom visual generado para este proyecto.

La información se estructurará, según las siguientes secciones:

- 3.2.1. Incorporar datos a tratar
- 3.2.2. Incorporar objeto Visual
- 3.2.3. Selección tipo de gráfico
- 3.2.4. Selección de características
- 3.2.5. Filtros y relación con otros gráficos
- 3.2.6. Consideraciones al desarrollador de informes

### 3.2.1. Incorporar datos a tratar

Una vez iniciado Power BI Desktop, se puede observar en la siguiente imagen, la pantalla inicial que se mostrará:

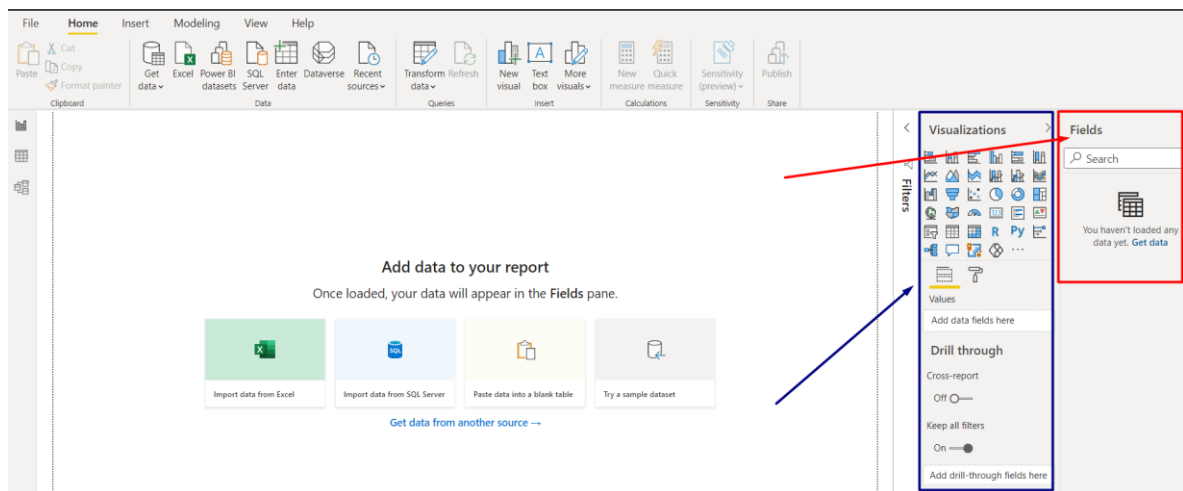


Figura 53: Imagen Inicial Power BI

En el panel rojo, <Fields>, se incorpora el conjunto de datos que queremos tratar. Como información, se indica que estos datos pueden tener diferentes orígenes, pudiéndolos tratar conjuntamente.

En el caso de este proyecto se utilizará los datos del siguiente dataset: heart.csv

<https://www.kaggle.com/rashikrahmanpritom/heart-attack-analysis-prediction-dataset>

El dataset contiene de un conjunto de datos de análisis y predicción de ataques cardíacos.





Figura 54: Imagen origen dataset Heart Attack

Dataset heart.csv:

| Name     | Desc SP   | Desc EN   |
|----------|---|---|
| age      | Edad de la persona  | Age of the person                                       |
| sex      | Sexo de la persona  | Gender of the person                                    |
| cp       | Pecho Tipo de dolor en el pecho                                       | Chest Pain type chest pain type                         |
| trtbps   | Presión arterial en reposo (en mm Hg)                                 | resting blood pressure (in mm Hg)                       |
| chol     | colestoral en mg / dl obtenido a través del sensor de IMC             | cholesterol in mg/dl fetched via BMI sensor             |
| fbs      | (azúcar en sangre en ayunas > 120 mg / dl) (1 = verdadero; 0 = falso) | (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false) |
| restecg  | Resultados electrocardiográficos en reposo                            | resting electrocardiographic results                    |
| thalachh | Frecuencia cardíaca máxima alcanzada                                  | maximum heart rate achieved                             |
| exng     | Angina inducida por el ejercicio (1 = sí; 0 = no)                     | exercise induced angina (1 = yes; 0 = no)               |
| oldpeak  | Pico anterior   | Previous peak   |
| slp      | Pendiente   | Slope   |
| caa      | Número de pendiente de vasos principales (0-3)                        | number of major vessels (0-3)                           |
| thall    | Frecuencia Thal   | Thal rate   |
| output   | Variable objetivo   | Target variable   |

Entrando en el detalle se puede indicar que las siguientes variables podrían tratarse como factorizables entre:

[0,1] → [sex] [fbs] [exng] [output]

[0,1,2] → [restecg] [slp]

[0,1,2,3] → [cp] [thall]

[0,1,2,3,4] → [caa]

En el caso de [cp], tipo dolor de pecho, los valores descriptivos son:

- Value 0: typical angina
- Value 1: atypical angina
- Value 2: non-anginal pain
- Value 3: asymptomatic

En el caso output:

- Value 0: Sin enfermedad cardiaca
- Value 1: Con enfermedad cardiaca

Realizando un pequeño análisis, obtenemos:

```
> describe(heart)
      vars  n  mean  sd median trimmed  mad min  max range skew kurtosis  se
age      1 303 54.37 9.08  55.0  54.54 10.38 29  77.0 48.0 -0.20  -0.57 0.52
sex      2 303  0.68 0.47   1.0   0.73  0.00  0   1.0  1.0 -0.78  -1.39 0.03
cp       3 303  0.97 1.03   1.0   0.86  1.48  0   3.0  3.0  0.48  -1.21 0.06
trtbps   4 303 131.62 17.54 130.0 130.44 14.83 94 200.0 106.0 0.71   0.87 1.01
chol     5 303 246.26 51.83 240.0 243.49 47.44 126 564.0 438.0 1.13   4.36 2.98
fbs      6 303  0.15 0.36   0.0   0.06  0.00  0   1.0  1.0  1.97   1.88 0.02
restecg  7 303  0.53 0.53   1.0   0.52  0.00  0   2.0  2.0  0.16  -1.37 0.03
thalachh 8 303 149.65 22.91 153.0 150.98 22.24 71 202.0 131.0 -0.53  -0.10 1.32
exng     9 303  0.33 0.47   0.0   0.28  0.00  0   1.0  1.0  0.74  -1.46 0.03
oldpeak 10 303  1.04 1.16   0.8   0.86  1.19  0   6.2  6.2  1.26   1.50 0.07
slp     11 303  1.40 0.62   1.0   1.46  1.48  0   2.0  2.0 -0.50  -0.65 0.04
caa     12 303  0.73 1.02   0.0   0.54  0.00  0   4.0  4.0  1.30   0.78 0.06
thall   13 303  2.31 0.61   2.0   2.36  0.00  0   3.0  3.0 -0.47   0.25 0.04
output  14 303  0.54 0.50   1.0   0.56  0.00  0   1.0  1.0 -0.18  -1.97 0.03
-----
> str(heart)
'data.frame':  303 obs. of  14 variables:
 $ age      : int  63 37 41 56 57 57 56 44 52 57 ...
 $ sex      : int  1 1 0 1 0 1 0 1 1 1 ...
 $ cp       : int  3 2 1 1 0 0 1 1 2 2 ...
 $ trtbps   : int 145 130 130 120 120 140 140 120 172 150 ...
 $ chol     : int 233 250 204 236 354 192 294 263 199 168 ...
 $ fbs      : int  1 0 0 0 0 0 0 0 1 0 ...
 $ restecg  : int  0 1 0 1 1 1 0 1 1 1 ...
 $ thalachh : int 150 187 172 178 163 148 153 173 162 174 ...
 $ exng     : int  0 0 0 0 1 0 0 0 0 0 ...
 $ oldpeak  : num 2.3 3.5 1.4 0.8 0.6 0.4 1.3 0 0.5 1.6 ...
 $ slp      : int  0 0 2 2 2 1 1 2 2 2 ...
 $ caa      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ thall    : int  1 2 2 2 2 1 2 3 3 2 ...
 $ output   : int  1 1 1 1 1 1 1 1 1 1 ...
```

Figura 55: Información elaborada en RStudio

Para incorporar el dataset al Power BI desktop, hay diferentes opciones que se pueden observar en el centro de la aplicación, al abrir Power BI Desktop, en este caso se ha seleccionado la opción de otro tipo de origen, en concreto .csv:

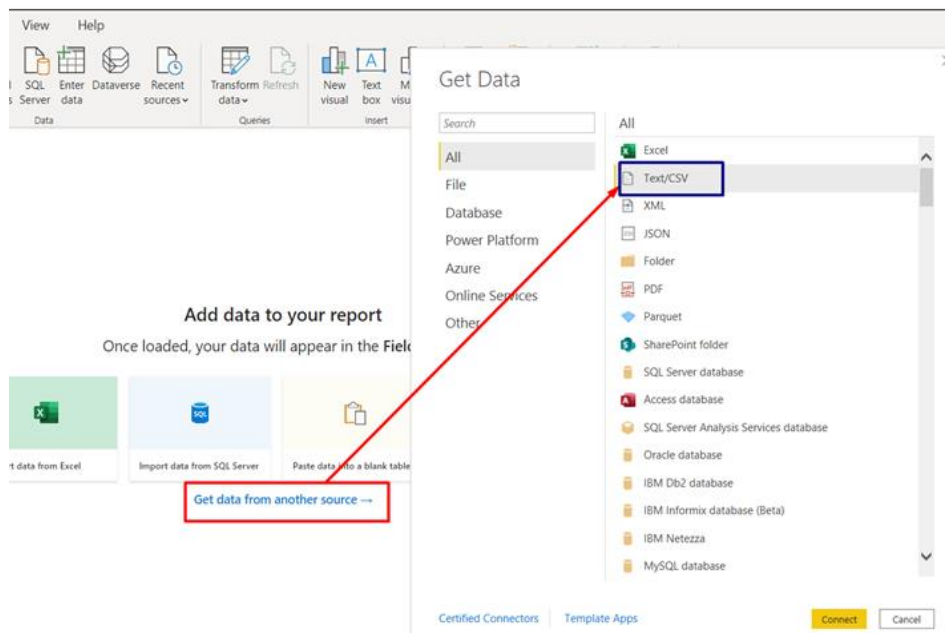


Figura 56: Obtención dataset

Dando como resultado final, el conjunto de campos del dataset:

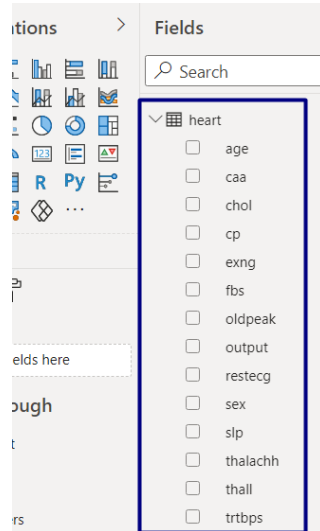


Figura 57: Campos dataset heart.csv

Nota: Cada R Visual aplica el <uniqueoperator> a su tabla de entrada. Para evitar que se eliminen filas idénticas, se debe considerar agregar un campo de entrada adicional con una ID única y hay que ignorarlo en el código R.

En este proyecto se ha creado el campo <índice>, que es una clave única, es un simple contador único de registro.

### 3.2.2. Incorporar objeto Visual

En el panel azul, <Visualizations>, de la imagen, <Imagen Inicial Power BI>, se puede ver los objetos visuales que por defecto se ofrecen.

Para incorporar uno nuevo se debe seguir los siguientes pasos:

- Ir a los tres puntos indicados en el recuadro rojo<Imagen Visualizations>, se desplegará una listado de cuatro opciones.

La opción del recuadro azul <Get more visuals>, abrirá las opciones que hay en el Marketplace de Microsoft (objetivo final de este proyecto).

La opción del recuadro rojo, <Import a visual from a file>, se utiliza para incorporar un custom visual directamente desde los archivos locales.

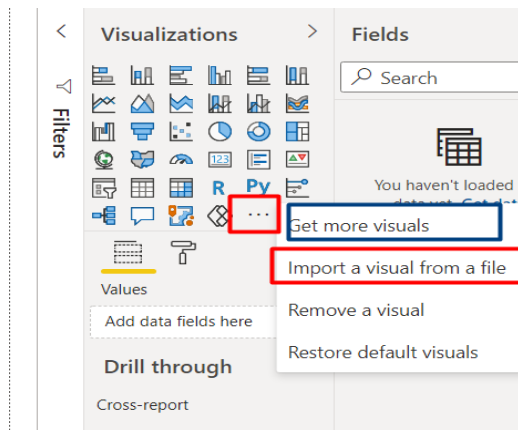


Figura 58: Visualizations

En este punto se realiza la incorporación del archivo:

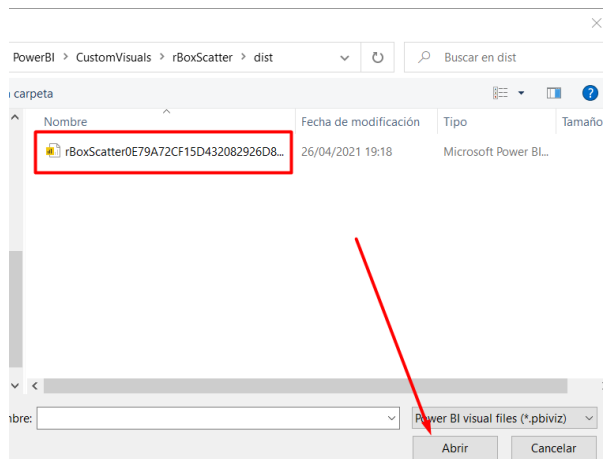


Figura 59: Incorporar objeto visual

Se puede observar que aparece un nuevo icono en el panel de <Visualizations>:

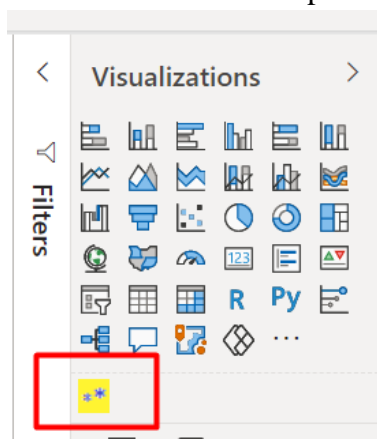


Figura 60: Imagen nuevo icono

En este momento, el objeto está disponible como uno más de los que se ofrece en el panel.

Al pulsar sobre él se muestra la siguiente imagen:



Figura 61: Elementos de los ejes

Se selecciona los campos de edad y sexo, con el fin de ver fácilmente como trabajan los gráficos que se van a tratar.

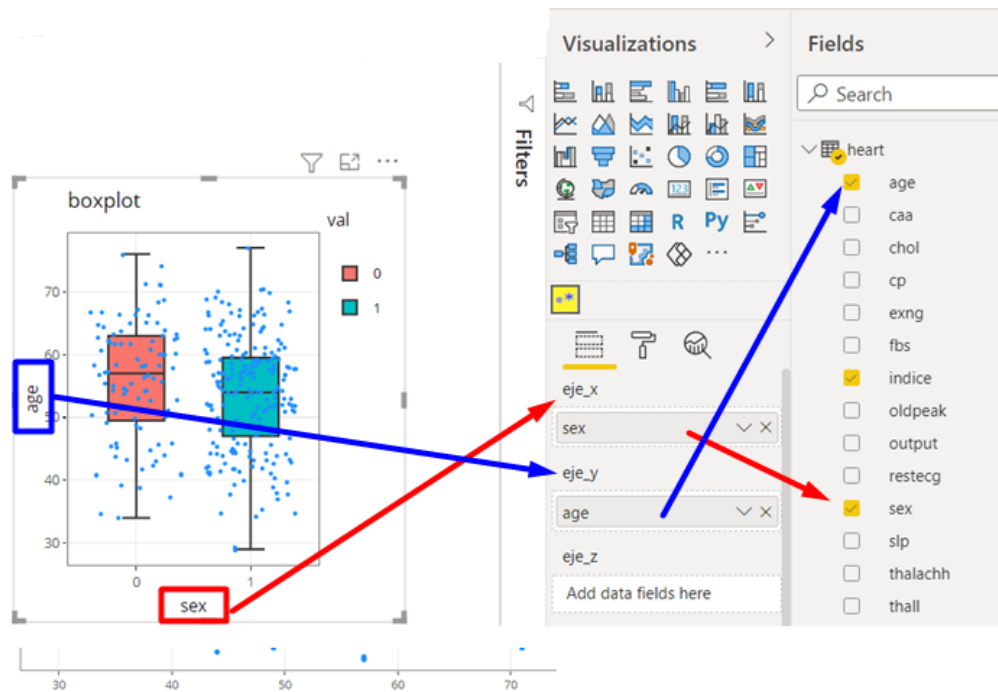


Figura 62: Selección elementos

Como se ha indicado en la parte técnica, se da la posibilidad de entrar tres elementos, indicados como eje\_x, eje\_y y eje\_z, que se utilizarán en función del gráfico seleccionado. Como obligatorios en este proyecto se han considerado el eje\_x y el eje\_y. Observando el gráfico resultante, se puede ver que el título de los ejes queda substituido por el nombre del campo, el cuál también se podrá modificar en caso de ser necesario por parte del usuario que diseña el informe.

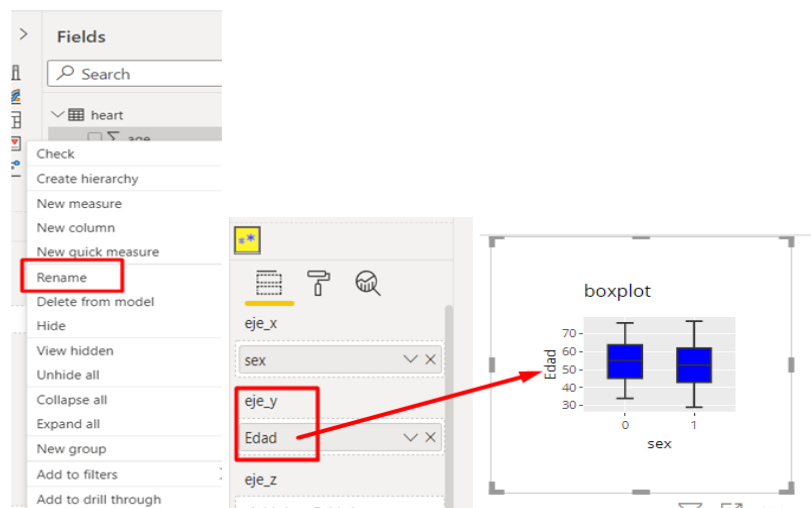


Figura 63: Modificación nomenclatura ejes

¿Qué ocurre, sino se indican los elementos?  
Normalmente, aparecería este tipo de ventana indicando que hay un error:

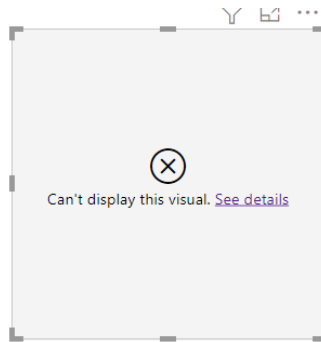


Figura 64: Imagen de error

Pero, el público objetivo quizás no está familiarizado con el language R, con lo que se ha realizado otro tipo de ventana informativa:

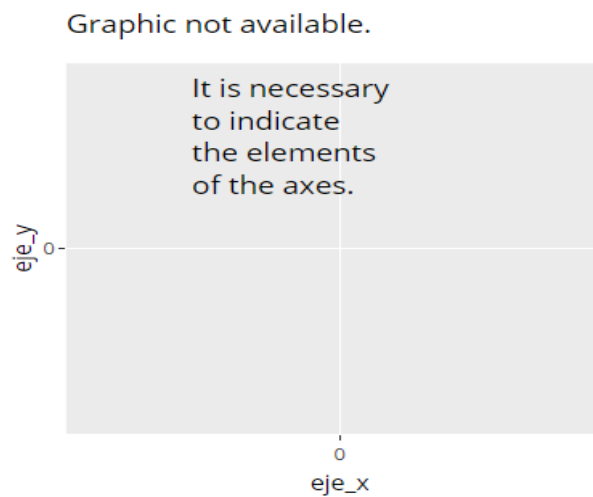
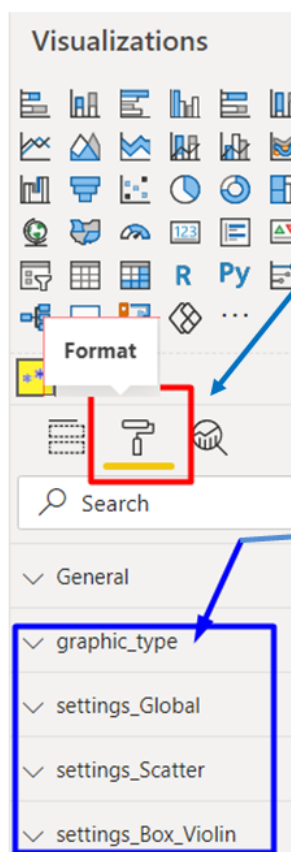


Figura 65: Imagen explicativa

### 3.2.3. Selección tipo de gráfico

Una vez seleccionado los elementos, se puede escoger diferentes modelos de gráficos.

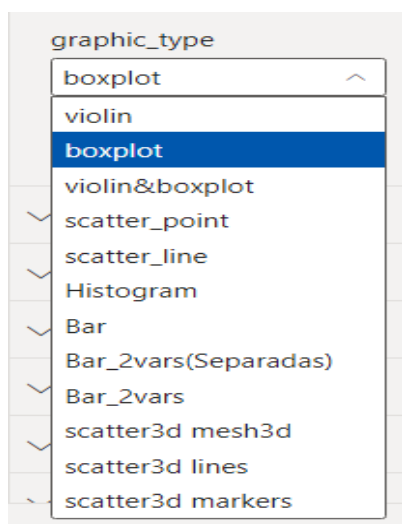
Para visualizar los tipos y ver los parámetros a escoger se deberá acceder a la sección de formato, icono justo al lado del que se utiliza para incorporar los datos:



Las características con las que se va a poder interactuar están agrupadas en las secciones indicadas en el recuadro azul

Figura 66: Sección Format

Al extender la sección de <Tipo\_Grafico> se pueden visualizar las siguientes opciones:



Para este proyecto se ha optado por estas opciones, pero se podrían extender a muchas más, según las especificaciones que se detallen en diferentes proyectos. En la parte técnica se especifica cómo se ha realizado. En un punto posterior se muestran ejemplos de estas opciones.

Figura 67: Tipo de gráficos

### 3.2.4. Selección de características

Como se ha indicado anteriormente, según el tipo de gráfico elegido existen unas características particulares, pero también es cierto que hay algunas que son comunes a todos los gráficos, como podría ser el color elegido para una recta o un punto o el tamaño y el tipo de tema, es decir, dar un formato predefinido.

### - Características Generales

Para estas características generales se seleccionará la sección <Settings\_Global>, en la siguiente imagen se representa como un diagrama de árbol las diferentes opciones:

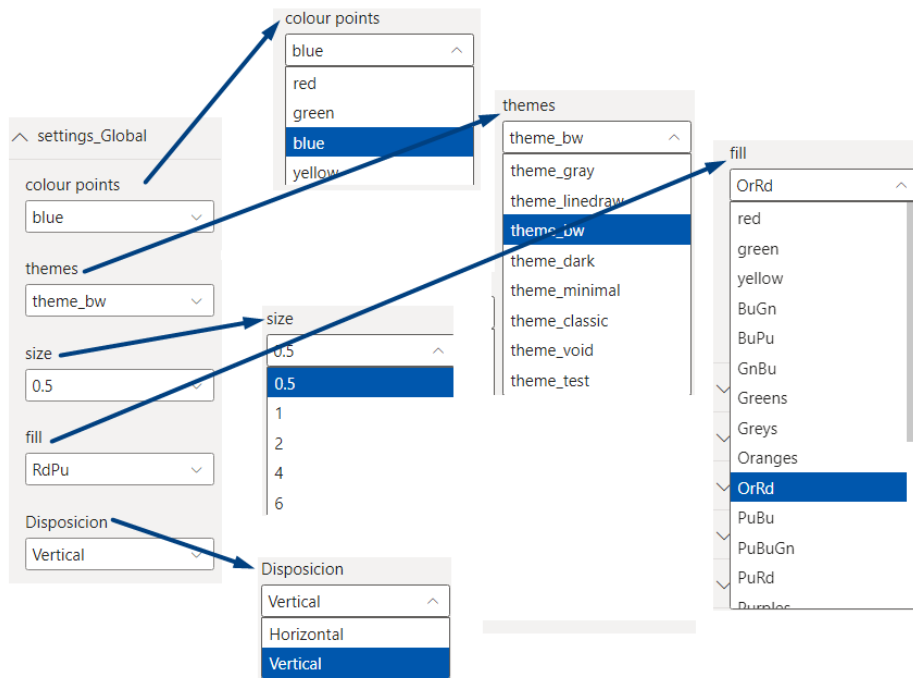


Figura 68: Características settings\_Global

### - Características Scatter

En esta sección se incluye la siguiente opción:

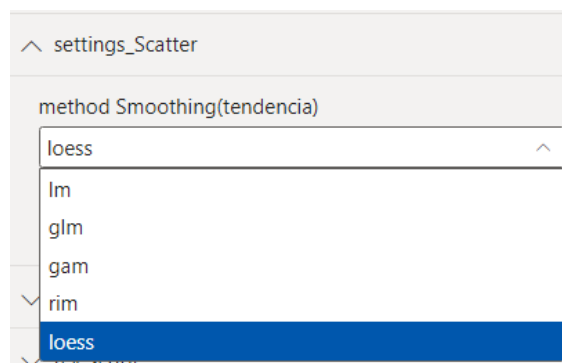


Figura 69: Características tendencia

Esta opción permite focalizar la tendencia. Se puede escoger entre los diferentes tipos programados. Ayuda a ver patrones en presencia de sobretrazado.

Por ejemplo, el suavizado de loess es un proceso mediante el cual muchos softwares estadísticos suavizan. En ggplot2 debe hacerse cuando se tiene menos de 1000 puntos, de lo contrario puede llevar mucho tiempo.



Por otro lado, gam es el suavizado en modo aditivo generalizado. Funciona con una gran cantidad de puntos.

- Características Box\_Violin

Las características disponibles en gráficos box o violín son:

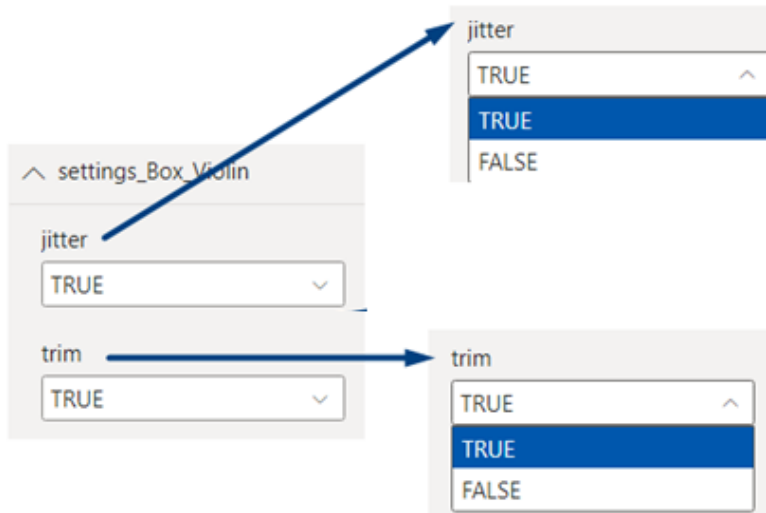


Figura 70: Características settings\_Box\_Violin

### 3.2.5. Filtros y relación con otros gráficos

Una vez obtenido el resultado del objeto visual, se podrá utilizar con las mismas características que se ofrecen al resto de objetos que son originales de Power BI. Estas opciones serían, principalmente, el uso de los filtros y la interacción entre gráficos, en caso de que el diseñador del informe lo cree oportuno.

Se muestra un ejemplo concreto:

A partir de los datos del dataset heart.csv, se trabaja para estudiar la relación entre la edad y el colesterol, en este caso la gráfica resultante sería:

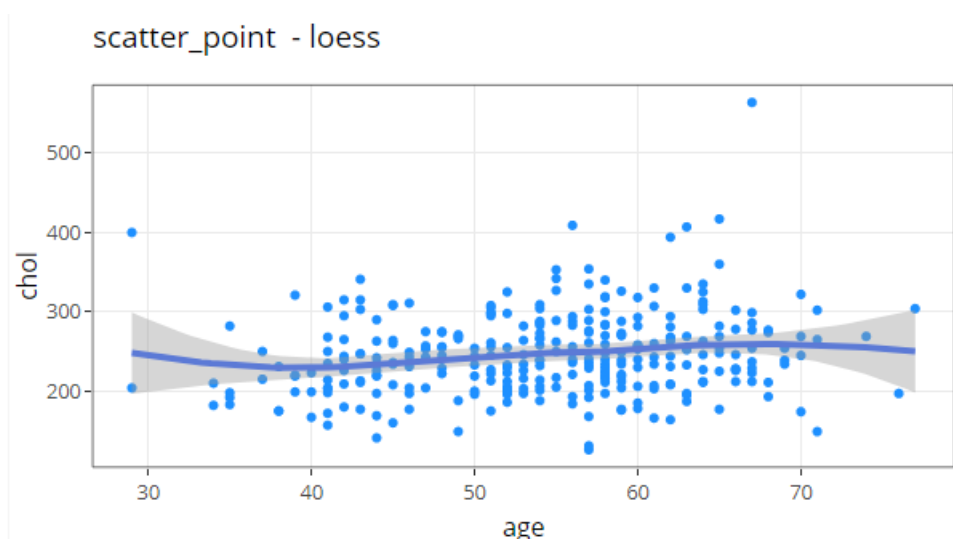


Figura 71: Ejemplo scatter\_point - loess

Se podría analizar, una franja de edad, que se quiere un análisis más específico por ejemplo menores de 40.  
Dando como resultado:

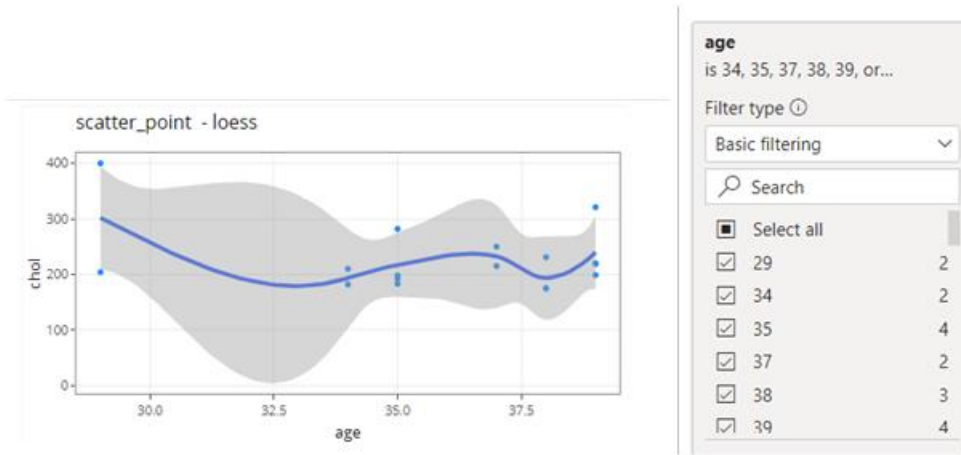


Figura 72: Ejemplo con filtro

Y si se quiere que se actualice en función de otra gráfica:

Imaginemos las siguientes gráficas:

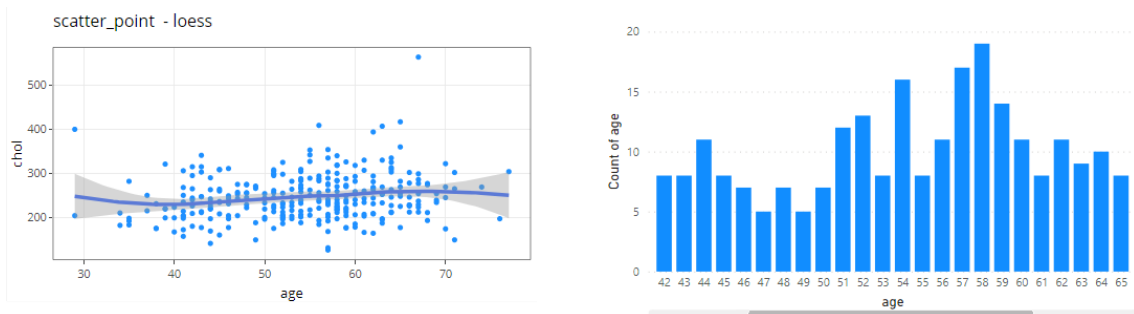


Figura 73: Ejemplo dos gráficas

Si se selecciona en el histograma, por ejemplo el valor 57,58,59,60, el resultado es el siguiente:

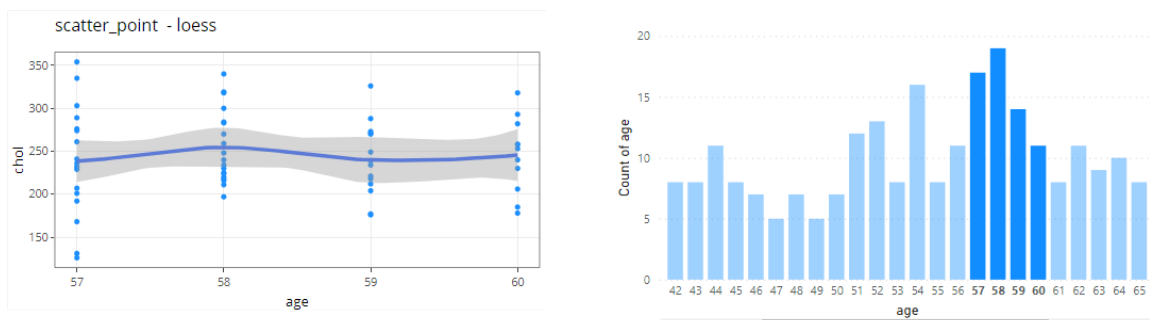


Figura 74: Ejemplo con filtrado de otra gráfica

Con lo que se demuestra que se puede generar un desarrollo en R, e interactuar con elementos que no se han desarrollado en este lenguaje y lo más interesante, la persona que generará el informe no ha escrito ni una línea de código.

### 3.2.6. Consideraciones al desarrollador Informes con objetos visual R

Podría pasar que al incorporar un objeto visual en R, aunque se tenga instalado R (cómo se ha indicado es necesario), no se haya instalado alguna librería que se requiere para ejecutar el custom visual, o que no haya habilitado los objetos visuales de script, en estos casos solo hay que habilitar e instalar:

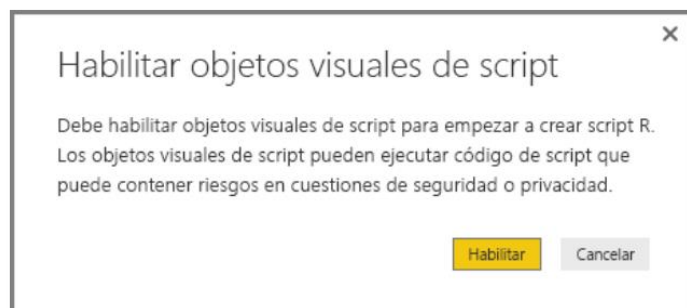


Figura 75: Activación visuales script

En el caso de librerías,

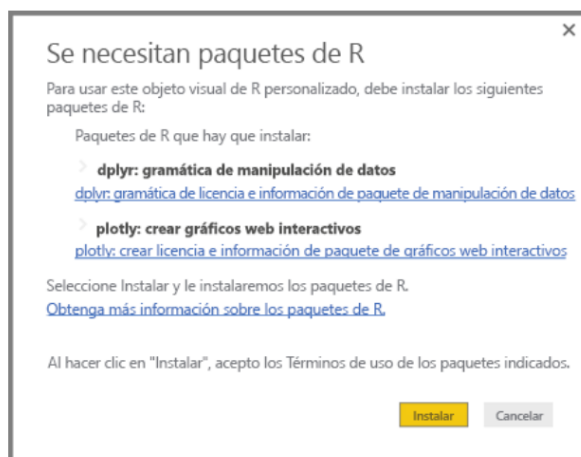


Figura 76: Instalación paquetes R

## 3.3. Requisitos y limitaciones de los paquetes R

El servicio Power BI admite paquetes publicados en CRAN. El servicio no es compatible con paquetes R personalizados o privados. Aunque se puede poner los paquetes privados a disposición de CRAN antes de solicitar que el paquete esté disponible en el servicio Power BI.

En la siguiente URL se puede ver la gran cantidad de paquetes (sin parar de actualizarse) que hay disponibles:

<https://docs.microsoft.com/es-es/power-bi/connect-data/service-r-packages-support>

### **3.4. Consideraciones al desarrollador del custom**

Cómo se ha podido comprobar a lo largo del proyecto se puede realizar gráficos en R totalmente transparentes para el usuario que desarrolle los informes. El diseñador que prepara el custom visual, puede implementarlo sin problemas, siempre que se sigan los pasos o etapas que se han indicado. Es muy importante no perder nunca de vista el esquema general de los archivos.

Conociendo la estructura en R del gráfico que se quiere realizar, se puede customizar prácticamente cualquier parámetro. Será decisión de las especificaciones del desarrollador de informes cómo desea visualizar agrupadas estas opciones, o si desea tratar un gráfico concreto con unas especificaciones determinadas.

En este proyecto no se ha profundizado en el análisis concreto de un ejemplo, ya que el objetivo era mostrar la factibilidad de crear un objeto visual en R y ponerlo a disposición de la comunidad.

### **3.5. Publicación objeto visual**

Una vez que se ha creado el objeto visual de Power BI, se puede publicar en AppSource para que otros usuarios lo detecten y lo usen. AppSource, es el lugar donde podrá encontrar aplicaciones SaaS y complementos para sus productos y servicios de Microsoft.

Para realizar la publicación se debe realizar una serie de pasos entre los que hay varias comprobaciones con el objetivo de verificar la utilidad del custom visual.

#### 3.5.1. Preparar custom visual

- Paquete Pbviz
- Archivo de informe .pbix de ejemplo
- Icono
- Capturas de pantalla
- Vínculo de descarga de soporte técnico
- Vínculo de documento de privacidad
- Contrato de licencia para el usuario final (CLUF)

#### 3.5.2. Obtención de un archivo XML del paquete de la aplicación

#### 3.5.3. Envío al Centro de partners

#### 3.5.4. Seguimiento del uso y estado del envío

#### 3.5.5. Certificación del objeto visual

### 3.5.1. Preparar custom visual

❖ Paquete pbviz : Se debe empaquetar con los siguientes metadatos:

```
pbviz.json > {} visual > supportUrl
1  {"visual":
2  {
3    "name": "rBoxScatter",
4    "displayName": "R_Box_Scatter",
5    "guid": "rBoxScatter0E79A72CF15D432082926D847A65F8D",
6    "visualClassName": "Visual",
7    "version": "1.0.0",
8    "description": "R_Box_Scatter",
9    "supportUrl": "http://community.powerbi.com/",
10   "githubUrl": ""
11  },
12  "apiVersion": "2.6.0",
13  "author": { "name": "Nuria Pont Vila", "email": "nuriapontvila@uoc.edu" },
14  "assets": { "icon": "assets/icon.png" },
15  "externalJS": null,
16  "style": "style/visual.less",
17  "capabilities": "capabilities.json",
18  "dependencies": "dependencies.json",
19  "stringResources": []
20 }
```

- Nombre de objeto visual:
- Nombre para mostrar
- GUID
- Versión
- Descripción
- Nombre y correo electrónico del autor

Figura 77: Metadatos paquete pbviz

❖ Archivo de informe .pbix de ejemplo

Se ha realizado el informe rBoxScatter.pbix, en el que hay las siguientes páginas:

- Página Inicial explicativa

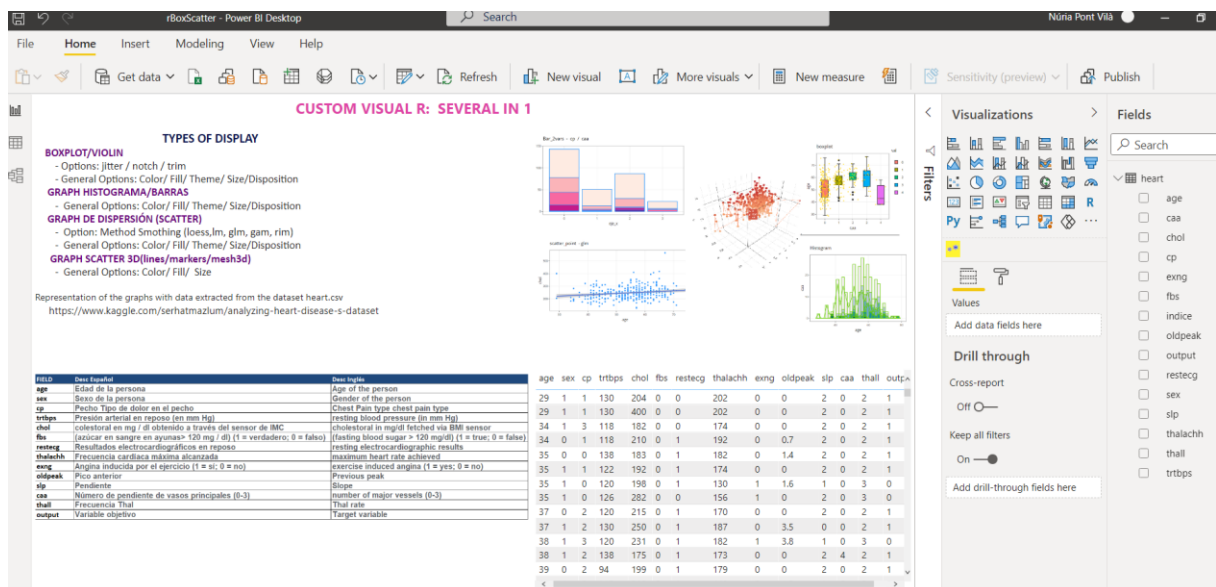
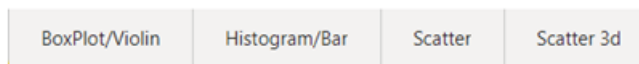


Figura 78: Página Inicial informe

- Páginas de demostración del objeto visual

Se muestra los diferentes tipos de gráficos con las características que se han preparado para ser utilizadas.



Nota: Debe funcionar sin ninguna conexión externa.

#### ❖ Icono

Se ha creado un icono respetando las características demandadas: Su formato puede ser .png, .jpg, .jpeg o .gif. Debe ser exactamente de 300 px (ancho) x 300 px (alto).

#### ❖ Capturas de pantalla

Se ha generado 3 capturas de pantalla, una inicial mostrando la página inicial del archivo pbix y dos de gráficos mostrando cómo se puede interactuar con los gráficos, gracias al uso de la librería plotly.

El formato de cada imagen puede ser .png, .jpg, .jpeg o .gif. Las dimensiones deben ser exactamente 1366 px (ancho) por 768 px (alto). El tamaño del archivo no puede superar los 1024 KB.

#### ❖ Vínculo de descarga de soporte técnico

<http://community.Power BI.com/>

#### ❖ Vínculo de documento de privacidad

Este vínculo se especifica como parte de la lista del Centro de partners y es visible para los usuarios cuando acceden a la lista de objetos visuales en AppSource.

#### ❖ Contrato de licencia para el usuario final (CLUF)

Se adjunta el contrato estándar que Microsoft proporciona para los objetos visuales de Power BI

### 3.5.2. Obtención de un archivo XML del paquete de la aplicación

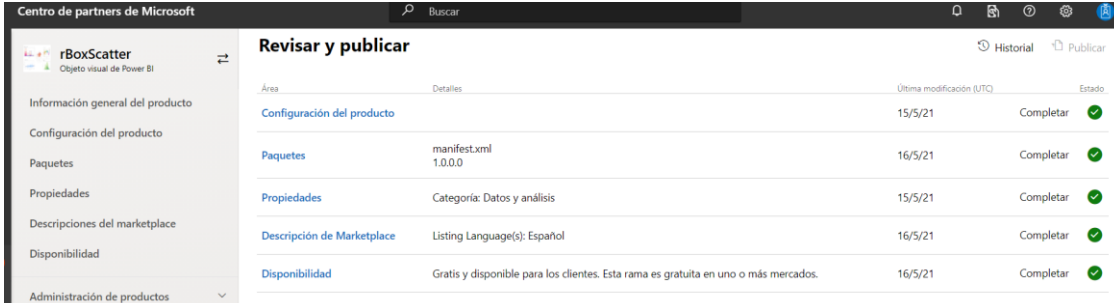
Previamente, se ha enviado a la dirección → [pbivizsubmit@microsoft.com](mailto:pbivizsubmit@microsoft.com) , tanto el archivo pbiviz.json como el archivo pbix del informe de ejemplo.

Por parte de Microsoft, se ha obtenido el archivo de manifiesto. (ver documento anexo: 7.1. *Manifest XML* (Microsoft)) . El archivo ‘manifest’, describe el nombre, número de versión, licencia y los archivos que forman parte del programa.

### 3.5.3. Envío al Centro de partners

Se ha realizado todos los requerimientos necesarios, los cuales han sido validados correctamente. (16/05/2021)

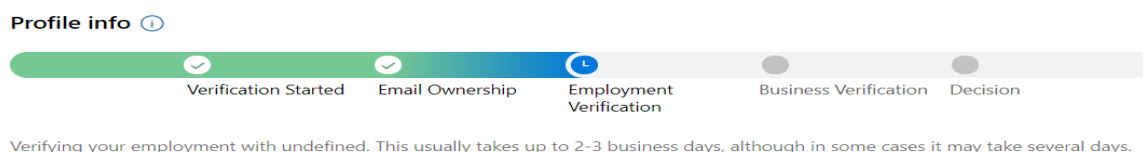
<https://partner.microsoft.com/en-us/dashboard/office/products/357bfb3c-f925-4dd2-8171-f0faed2733b5/submit>



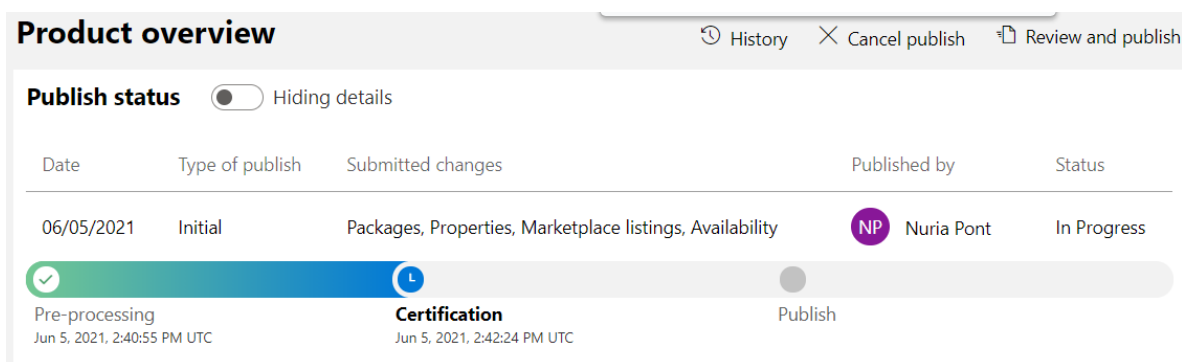
| Área                       | Detalles  | Última modificación (UTC) | Estado      |
|----------------------------|---|---------------------------|-------------|
| Configuración del producto |   | 15/5/21                   | Completar ✓ |
| Paquetes                   | manifest.xml<br>1.0.0.0   | 16/5/21                   | Completar ✓ |
| Propiedades                | Categoría: Datos y análisis   | 15/5/21                   | Completar ✓ |
| Descripción de Marketplace | Listing Language(s): Español  | 16/5/21                   | Completar ✓ |
| Disponibilidad             | Gratis y disponible para los clientes. Esta rama es gratuita en uno o más mercados. | 16/5/21                   | Completar ✓ |

Figura 79: Estado requerimientos

Quedando pendiente de validación de cuenta 'Profile'



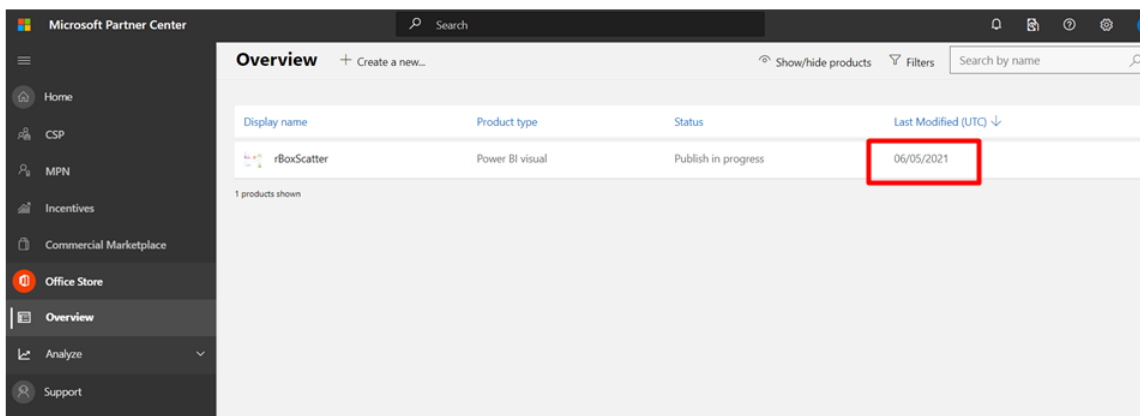
Nota importante: La validación debería haber sido de unos días y el siguiente proceso se ha podido realizar el día 05/06/2021:



### 3.5.4. Seguimiento del uso y estado del envío

En esta etapa se realiza el seguimiento del estado del custom visual.

Nota: Se adjunta la imagen de la situación a día 05/06/21, antes de la liberación del documento.



### 3.5.5. Certificación del objeto visual

Última etapa en la que se puede optar a la certificación del objeto.

### 3.5.6. Problemática

El proceso de validación ha durado más de lo esperado.

Se adjunta un link, dónde se puede ver que parece que este problema no es puntual y hay más usuarios que se muestran bloqueados a la hora de poder publicar el custom.

<https://www.microsoftpartnercommunity.com/t5/Feedback-Support-Discussions/Joining-MPN-stuck-on-Employment-verification/m-p/15133>

### 3.6.Repositorio Código

El código de este proyecto estará disponible en un repositorio GitHub <https://github.com/nuriapontvila/RBoxScatter>

Se podrá acceder a informe rBoxScatter.pbix, en la raíz del github

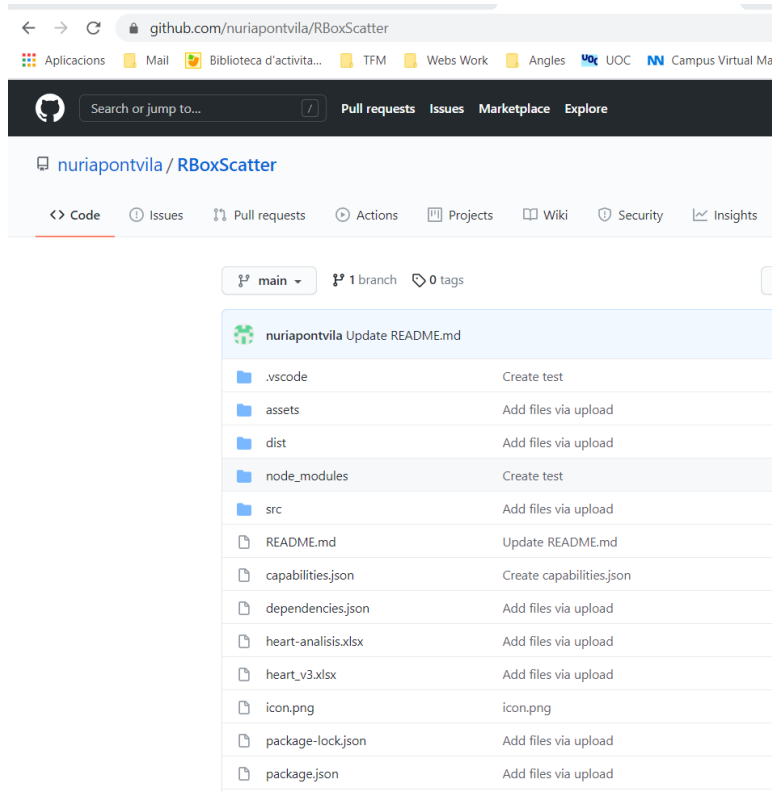


Figura 80: Repositorio Código TFM



## 4. Conclusiones

Este apartado de conclusiones lo estructuraré en las siguientes secciones:

1. Descripción conclusiones
2. Logros de los objetivos
3. Análisis de la planificación
4. Líneas de trabajo futuro

### 1. Descripción conclusiones

Este trabajo ha estado pensado para ofrecer la posibilidad de visualizar e interpretar cálculos complejos estadísticos mediante gráficos desarrollados en lenguaje R sin la necesidad de programarlos.

Para llevarlo a cabo este proceso ha sido necesario ser riguroso en los procedimientos, viendo paso a paso los aciertos y los errores que se iban generando para ir corrigiendo y/o adaptando nuevas opciones.

Se ha podido observar que el número de custom visuals actuales de R en Power BI es muy minoritario, lo cual hace pensar que quizás se debe difundir más el tipo de gráficos que puede ofrecer dicho lenguaje y los conceptos que puede ofrecer pero a un nivel más comprensible para una comunidad de usuarios con un perfil menos científico. Para poner un ejemplo, se vuelve a remitir al tipo de gráfico BoxPlot, que resume la distribución de los datos mostrando la posición de sus cuartiles y de los valores atípicos (outliers), es un gráfico muy útil para que permite cuantificar cómo de concentrada está una variable numérica, ver valores extremos y sobretodo facilitar la comparación entre grupos, con lo que explicado de manera más didáctica llegaría a ser mucho más utilizado.

### 2. Logros de los objetivos

A continuación, se describirá el resultado de los objetivos que se habían marcado para este proyecto:

- Conexión entre las herramientas Power BI y el lenguaje R
- Creación e Interacción de los gráficos
- Creación custom visual propio
- Parametrización elementos visuales
- Publicación custom visual en Marketplace
- Conseguir principios de entendibilidad, claridad, consistencia, eficiencia, necesidad y confiabilidad

Se pueden describir las siguientes conclusiones:

- ✓ La conexión entre las herramientas Power BI y el lenguaje R es perfectamente realizable. Solo hay que seguir el procedimiento y el orden lógico para realizar la conexión entre los elementos.

- ✓ La creación de gráficos en R y la posibilidad de que estos puedan ser interactivos por parte del usuario, también es posible. Power BI mantiene y permite las funcionalidades que aporta la librería plotly, la cual permite interactuar con el mouse facilitando información al usuario de manera mucho más amena.
- ✓ Se ha realizado la creación del custom visual, Para lograr el éxito de este punto es clave seguir los procedimientos con las instrucciones descritas. Se ha creado un objeto visual con código de programación R, una vez compliado el paquete que se ha tratado, se ha generado un único archivo que será nuestro custom visual, que será importable a la herramienta Power BI.
- ✓ Se ha comprobado la posibilidad de parametrizar los elementos visuales asociados a uno o varios gráficos. Se inició el proyecto, pensando en un gráfico sencillo como el boxplot y ver las posibilidades que se podían realizar, pero al ver las posibilidades y quererlas demostrar, se fue ampliando las opciones. Esta ampliación pasa que con en el mismo objeto visual, dar la posibilidad de escoger más tipos de gráficos. A su vez, se ha podido investigar cómo ofrecer al usuario desarrollador de informes, diferentes opciones para configurar el gráfico seleccionado. Se ha podido estudiar, cómo organizar dichas funcionalidades en diferentes carpetas según el tipo de gráfico y crear una de general para parámetros que pueden afectar de manera general a todos los gráficos, como podría ser el color o el tamaño.  
Como es de imaginar, solo se han implementado algunas de las opciones, pero suficientes para demostrar la eficacia del trabajo.
- ✓ La publicación del custom visual en el Marketplace de Microsoft ha sido un logro que por ahora se ha conseguido parcialmente, si bien por aspectos técnicos parece haber pasado los requerimientos, se está esperando aspectos legales para su validación.  
En cualquier caso el custom visual estaría disponible para una organización, es decir, podría publicar el objeto en una empresa y podría ser utilizable por cualquier desarrollador de dicha organización.
- ✓ Conseguir principios de entendibilidad, claridad, consistencia, eficiencia, necesidad y confiabilidad. Estos objetivos son intrínsecos en el desarrollo de cualquier proyecto que se desee compartir con otros colaboradores. Este objetivo es muy subjetivo, por eso lo he compartido con algunos compañeros que me han podido dar su punto de vista y ayudar a corregir aspectos que debían ser especificados más detalladamente.

### 3. Análisis de la planificación

Para realizar el seguimiento y que no hubiera desviaciones en los objetivos marcados de proyecto, se ha seguido la metodología scrum, y ha sido perfecta para este proyecto.

Ha habido una interacción con Susana (tutora del proyecto), que ha facilitado los avances de forma progresiva.

Con esta metodología, se ha podido ampliar las opciones que en un principio se habían propuesto sin que el global del proyecto se viera afectado.

Hay que mencionar que ha habido un imprevisto en el desarrollo del proyecto, este ha sido referente a la publicación en el Marketplace de Microsoft. Microsoft, estima un plazo de unas cuatro semanas para poder publicar el objeto visual, lo cual hizo acelerar la parte técnica para poder tener el objeto correctamente diseñado para su implantación y ver si era posible su disponibilidad para la fecha de la presentación del proyecto. En este punto se envió a Microsoft, pero sigue en proceso de validación.

### 4. Líneas de trabajo futuras

Las líneas futuras de trabajo las dividiría en dos momentos, uno sería el inmediato con la continuación del trabajo actual y el otro momento sería las acciones futuras para potenciar este tipo de trabajos.

El inmediato trabajo es finalizar la publicación en el Marketplace de Power BI de Microsoft. Como se ha visto seguir con temas de documentación legal para la publicación

Una vez publicado el custom será realizar su certificado, lo cual dará más prestigio al custom visual, en este punto se exigen una serie de comprobaciones más específicas, Microsoft somete algunos elementos visuales seleccionados a pruebas de calidad y riesgo aún más rigurosas.

Las pruebas del proceso de certificación incluyen, pero no se limitan a:

- Revisiones de código
- Análisis de código estático
- Pérdida de datos
- Pruebas de vulnerabilidad ante datos aleatorios o inesperados
- Pruebas de penetración
- Pruebas de acceso XSS
- Inserción de datos malintencionados
- Validación de la entrada
- Pruebas funcionales

Los elementos visuales que pasan tales pruebas se denominan elementos visuales certificados. Estos elementos visuales también proporcionan algunas funciones adicionales, como suscripciones por correo electrónico, exportación a PowerPoint, etc.

Como líneas de futuro que se podrían tratar, y quizás no en un tipo de trabajo de un master de ciencia de datos pero si relacionado con él, sería la divulgación del conocimiento que se puede extraer del resultado de los gráficos que aportan librerías como son ggplot2 y plotly.

Otra de las líneas para explorar es analizar otras librerías de R para el análisis de series temporales o algoritmos de clasificación y agrupamiento.

Aunque una línea muy atractiva sería analizar los datos de un sistema de información geográfica (GIS) es un entorno para recopilar, gestionar y analizar datos. Arraigado en la ciencia de la geografía, el GIS integra muchos tipos de datos. Analiza la ubicación espacial y organiza capas de información en visualizaciones usando mapas y escenas 3D. Con esta capacidad única, los GIS revelan conocimientos más profundos de los datos, como patrones, relaciones y situaciones, ayudando a los usuarios a tomar decisiones más inteligentes.

```
ggplot(data = municipios) +  
  geom_sf() +  
  coord_sf(crs = "+proj=laea +lat_0=52 +lon_0=10 +x_0=4321000  
    +y_0=3210000 +ellps=GRS80 +units=m +no_defs ")
```



Figura 81: Gráfico con coordenadas geográficas

## 5. Glosario

**Big Data:** Los macrodatos, también llamados datos masivos, inteligencia de datos, datos a gran escala terminología en idioma inglés utilizada comúnmente, es un término que hace referencia a conjuntos de datos tan grandes y complejos que precisan de aplicaciones informáticas no tradicionales de procesamiento de datos para tratarlos adecuadamente.

**Cran:** Comprehensive R Archive Network. es el repositorio de software central de R , respaldado por R Foundation.

**Crisp-DM:** es un modelo de proceso estándar abierto que describe los enfoques comunes utilizados por los expertos en minería de datos

**Custom visual:** Los objetos visuales que no se incluyen automáticamente con Power BI se conocen como objetos visuales personalizados o también conocidos como custom visuals.

**Dashboard:** Documento en el que se reflejan, mediante una representación gráfica, las principales métricas o KPI que intervienen en la consecución de los objetivos de una estrategia

**Freemium:** es un modelo de negocio que funciona ofreciendo servicios básicos gratuitos, mientras se cobra dinero por otros servicios más avanzados o especiales

**Ggplot2:** es un paquete de visualización de datos de código abierto para el lenguaje de programación estadística R.

**GIS:** Geographical Information System sistema de información capaz de integrar, almacenar, editar, analizar, compartir y mostrar la información geográficamente referenciada.

**HTML:** siglas en inglés de HyperText Markup Language ('lenguaje de marcado de hipertexto'), hace referencia al lenguaje de marcado para la elaboración de páginas web.

**IDE:** Integrated Development Environment, en español ambiente de desarrollo integrado o entorno de desarrollo integrado

**SVG:** SVG es un formato de archivo basado en XML que ha sido diseñado específicamente para la visualización en la Web. SVG puede contener información tanto vectorial como de ráster

**Objeto visual:** Es un tipo de gráfico generado por diseñadores de Power BI que utiliza los datos de informes y conjuntos de datos.

**PAC o PEC:** Prueba de evaluación continua

**Plotly:** es un paquete de visualización de datos de código abierto para el lenguaje de programación estadística R.

**PNL:** La Programación Neurolingüística (PNL) es un conjunto de modelos, habilidades cognitivas y técnicas para actuar, pensar y sentir de forma efectiva en todos los aspectos de nuestra vida. PNL se encarga del estudio de la conducta humana, centrándose en la comprensión de los procesos mentales y emocionales.

**Power BI:** nombre colectivo para una variedad de aplicaciones y servicios basados en la nube que ayudan a las organizaciones a recopilar, administrar y analizar datos de una variedad de fuentes, a través de una interfaz fácil de usar

**Power BI desktop:** Power BI Desktop es una aplicación de escritorio gratuita que funciona con el servicio de Power BI. Se puede instalar directamente en sus equipos y nos ayuda a explorar los datos, de una manera más profunda y avanzada.

**R:** R es un entorno y lenguaje de programación con un enfoque al análisis estadístico.

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

**Scrum:** Scrum es un proceso de gestión que reduce la complejidad en el desarrollo de productos para satisfacer las necesidades de los clientes. La gerencia y los equipos de Scrum trabajan juntos alrededor de requisitos y tecnologías para entregar productos funcionando de manera incremental usando el empirismo.

**Uoc:** Universitat Oberta de Catalunya

**VSC:** Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.

**Widget:** En informática, un widget o artilugio es una pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de widgets o Widget Engine. Entre sus objetivos están dar fácil acceso a funciones frecuentemente usadas y proveer de información visual.

## 6. Bibliografía

Todos estos registros han sido consultados entre febrero y junio de 2021

[1] El día a día de un proyecto ágil: Scrum.

Disponible es:

[http://cv.uoc.edu/annotation/ebc1adfc61836d7205ad7dde343367b5/603266/PID\\_00236248/PID\\_00236248.html](http://cv.uoc.edu/annotation/ebc1adfc61836d7205ad7dde343367b5/603266/PID_00236248/PID_00236248.html)

[2] What is CRISP DM?. ¿Qué es Crisp-dm?

Disponible en : <https://www.datascience-pm.com/crisp-dm-2/>

[3] Power BI (Página principal)

Disponible en : <https://powerbi.microsoft.com/en-us/>

[4] Gartner Magic Quadrant for Analytics and Business Intelligence Platforms

Disponible en : <https://www.gartner.com/en/documents/2020/3980852-magic-quadrant-for-meeting-solutions1>

[5] Tendencia Power BI vs Qlik vs Tableau: ¿Cuál es el mejor software de análisis de datos?

Disponible en : <https://vandalytic.com/power-bi-vs-qlik-vs-tableau-cual-es-el-mejor-software-de-analisis-de-datos/>

[6] Mike Morris . Una guía completa paso a paso para principiantes en la comprensión del Power BI. Mike Morris.Power BI,

[7] Tendencias de los lenguajes de programación (redmonk)

Disponible en : [https://redmonk.com/sogrady/2021/03/01/language-rankings-1-21/?utm\\_source=rss&utm\\_medium=rss&utm\\_campaign=language-rankings-1-21](https://redmonk.com/sogrady/2021/03/01/language-rankings-1-21/?utm_source=rss&utm_medium=rss&utm_campaign=language-rankings-1-21)

[8] Tendencias de los lenguajes de programación (Tiobe)

Disponible en : <https://www.tiobe.com/tiobe-index/>

[9] Tendencias de los lenguajes de programación (PYPL)

Disponible en : <https://pypl.github.io/PYPL.html>

[10] Conceptos ggplot2

Disponible en : [ggplot2.tidyverse.org](https://ggplot2.tidyverse.org)

[11] Carson Sievert.Interactive web-based data visualization with R, plotly, and shiny.2019-12-19

Disponible en : <https://plotly-r.com/index.html>

[12] Creando gráficas con calidad para publicación con ggplot2

<https://swcarpentry.github.io/r-novice-gapminder-es/08-plot-ggplot2/>

[13] Elementos del Marketplace

<https://appssource.microsoft.com/ca/marketplace/apps?product=power-bi-visuals>  
[consultado marzo 2021].

[14] Galería scripts en R que hay en Power BI

[R Script Showcase - Microsoft Power BI Community](#)

[15] Ejemplos visuales Power Bi

[Power BI visuals samples - Power BI | Microsoft Docs](#)

[16] Objetos visuales personalizados interactivos de Power BI con R

[Interactive Power BI custom visuals with R | by Freddie Tasker | Towards Data Science](#)

[17] Cómo utilizar R Script (no a través de custom visual) en Power BI

[RPubs - Utilizando R en Power BI](#)

[18] Estudio avanzado de una aplicación de una Artificial Neural Network (ANN) en Power BI adaptando el lenguaje de programación R.

[Aplicación de una Artificial Neural Network \(ANN\) en Power BI adaptando el lenguaje de programación R - CepoBIA](#)

[19] Guía de buenas prácticas de visualización

[http://atenciociudadana.gencat.cat/web/.content/manuals/visualitzacions/guia\\_visualitzacio\\_es.pdf](http://atenciociudadana.gencat.cat/web/.content/manuals/visualitzacions/guia_visualitzacio_es.pdf)

[20] Información académica Violin.BoxPLot

<https://towardsdatascience.com/violin-plots-explained-fb1d115e023d>

## **Documentación utilizada para realizar el proyecto entre febrero y junio 2021:**

- Hadley Wickham and Garre Golemund. R for Data Science.  
Disponibile en : <https://r4ds.had.co.nz/index.html> [Consulta: abril-mayo 2021]
- Teo Lachev. APPLIED MICROSOFT POWER BI (5th Edition) (BRING YOUR DATA TO LIFE)
- Ware, Colin. Information Visualization, Third Edition: Perception for Design. Morgan Kaufmann, 2012  
Wikipedia. Interactive Storytelling. [Five Ws - Wikipedia](#) [Consulta: abril-mayo 2021]
- Yihui Xie, J. J. Allaire, Garre Golemund. R Markdown: The Definitive Guide. 2020-12-14.  
Disponibile en : <https://bookdown.org/yihui/rmarkdown> [Consulta: abril-mayo 2021]
- Comunidad Microsoft Power BI:  
[Home - Microsoft Power BI Community](#)



- Utilizando R en Power BI  
[https://rpubs.com/rubefbc/powerbi\\_gapminder](https://rpubs.com/rubefbc/powerbi_gapminder)
- ggplot2  
<https://www.r-graph-gallery.com/ggplot2-package.html>
- Cheatsheet ggplot2  
<https://ggplot2.tidyverse.org/>
- Ejemplos de Power BI visuals  
<https://docs.microsoft.com/ca-es/power-bi/developer/visuals/samples>
- Join the Microsoft Partner Network  
<https://partner.microsoft.com/>
- Create custom data visualizations for Power BI  
<https://powerbi.microsoft.com/en-us/developers/custom-visualization/>
- Publicación de objetos visuales de Power BI en el Centro de partners  
<https://docs.microsoft.com/es-es/power-bi/developer/visuals/office-store>

## 7. Anexos

### 7.1. Manifest XML (Microsoft)

Documento enviado por parte de Microsoft, que valida la información del custom. Este archivo es imprescindible para poder publicar el custom en el AppSource. El nombre del archivo es *'Manifest XML'*

#### Manifest XML (Microsoft)

```
<?xml version="1.0" encoding="utf-8"?>
<OfficeApp xmlns="http://schemas.microsoft.com/office/appforoffice/1.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="ContentApp">
  <Id>FF944500-B088-11EB-96AE-E58AF9FF7A67</Id>
  <Version>1.0.0.0</Version>
  <ProviderName>Nuria Pont Vila</ProviderName>
  <DefaultLocale>en-US</DefaultLocale>
  <DisplayName DefaultValue="R_Box_Scatter"/>
  <Description DefaultValue="R_Box_Scatter"/>
  <IconUrl DefaultValue="https://visuals.azureedge.net/app-store/Power
BI_icon32x32.png"/>
  <SupportUrl DefaultValue="http://community.Power BI.com"/>
  <Hosts>
    <Host Name="Power BI"/>
  </Hosts>
  <Requirements>
    <Sets>
      <Set Name="BuiltWithR.Power BI" MinVersion="1.1"/>
    </Sets>
  </Requirements>
  <DefaultSettings>
    <SourceLocation DefaultValue="https://appsource.Power
BI.com/rBoxScatter.rBoxScatter0E79A72CF15D432082926D847A65F8D7.1.0.0.0.pbi
viz"/>
    <RequestedWidth>100</RequestedWidth>
    <RequestedHeight>100</RequestedHeight>
  </DefaultSettings>
  <Permissions>ReadWriteDocument</Permissions>
</OfficeApp>
```

## 7.2. Ejemplos de gráficos del custom visual realizado

Seguidamente se muestran algunos gráficos que se pueden realizar con este visual de Power BI, parametrizándolo según las características escogidas.

### ✓ Gráficos BOX/Violin (diagramas de caja)

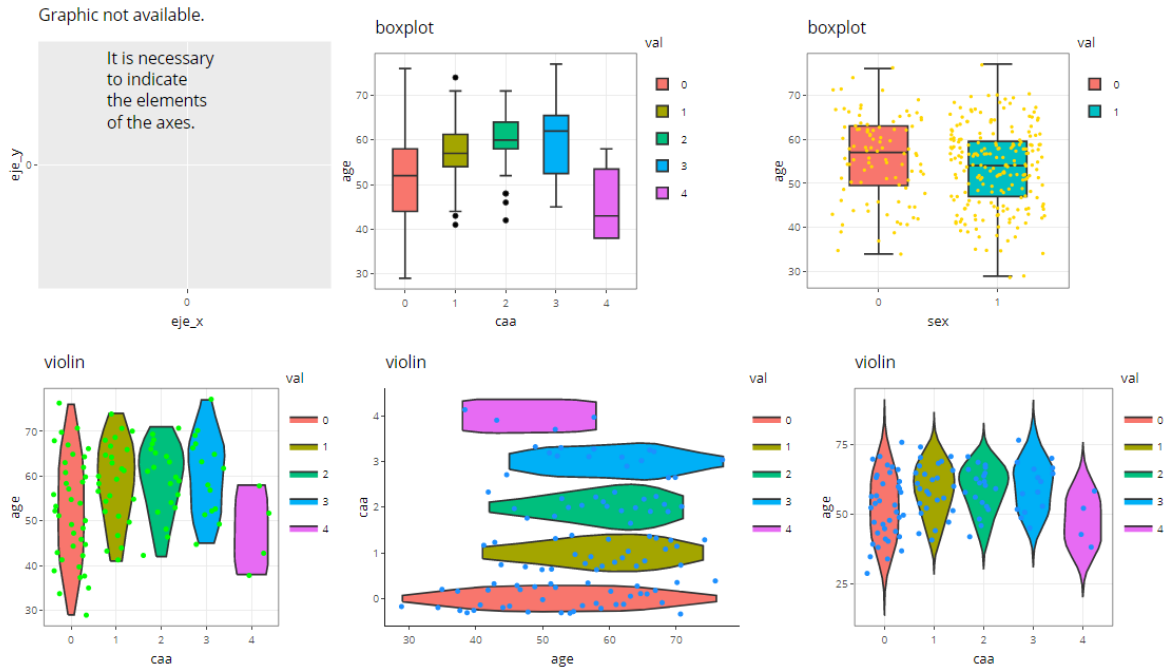


Figura 82: Dashboard Power BI

El primer ejemplo, muestra un mensaje en caso de no haber seleccionado los elementos que corresponden a las dimensiones X e Y.

Una vez seleccionado los valores a tratar, se puede indicar una paleta de colores con el que se mostrarán las diferentes categorías. Esta paleta, se puede escoger de la lista de opciones generales, se muestran varias alternativas con el que el diseñador del informe podrá trabajar.

Se muestra un caso en detalle, analizamos en boxplot la edad respecto al sexo:

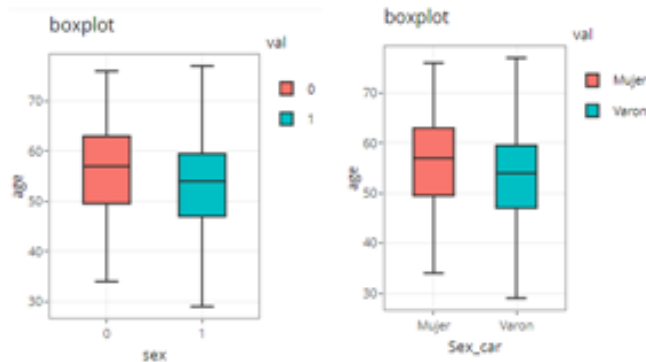


Figura 83: Gráfico boxplot

Se puede ver, que se puede añadir, actualizar los datos que se incorporen, en este caso se ha creado una nueva columna con la que se identifica el sexo con una descripción,

cómo es normal, el diseñador del informe en cada caso escogerá la mejor forma de trabajar o tratar el tema.

Siguiendo con el ejemplo, si se quiere ver todas las observaciones se selecciona la opción jitter, con los puntos de color verde y se desea que sea horizontal:

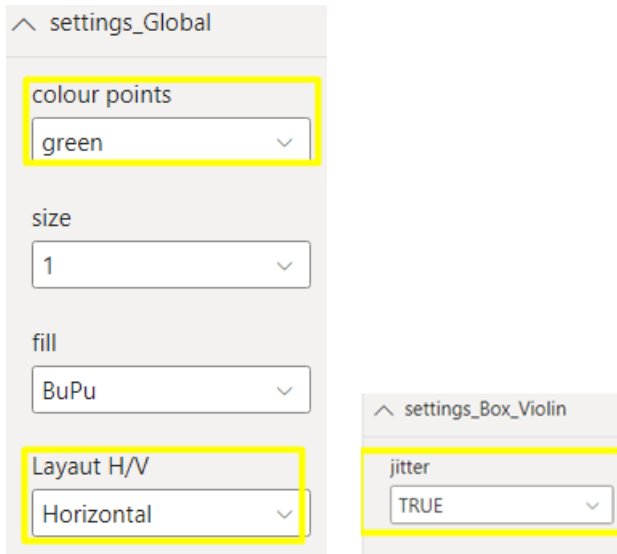


Figura 84: Opciones Globales

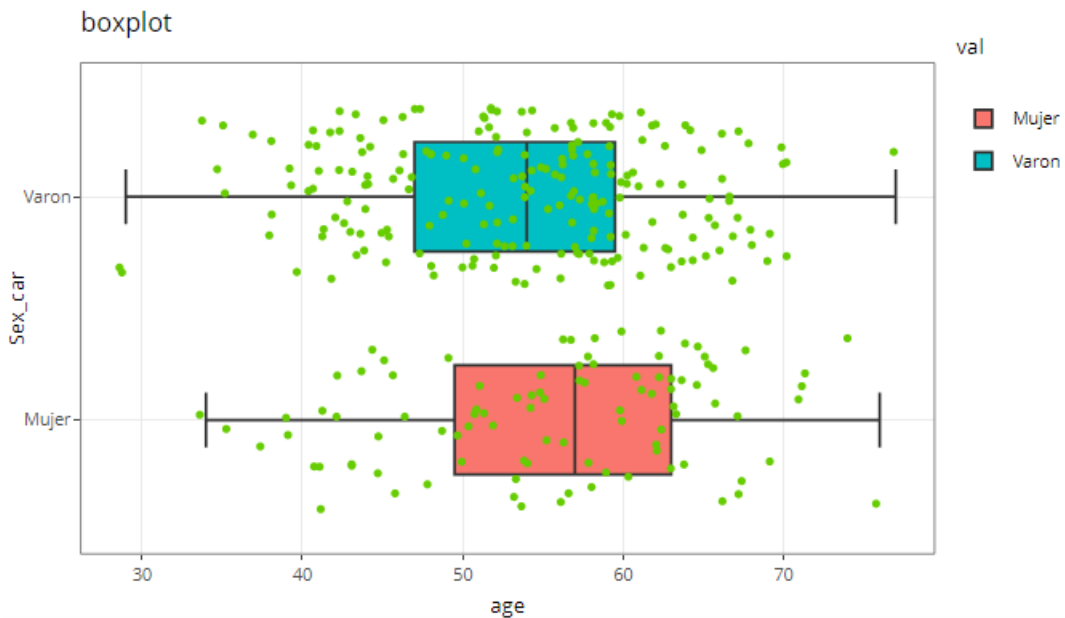


Figura 85: Gráfico boxplot horizontal

Al trabajar con gráficos tipo plotly, se permite la interacción, haciendo zoom, mostrando los valores, etc , como se puede ver en las siguientes imágenes, solo hay que posicionarse con el mouse:

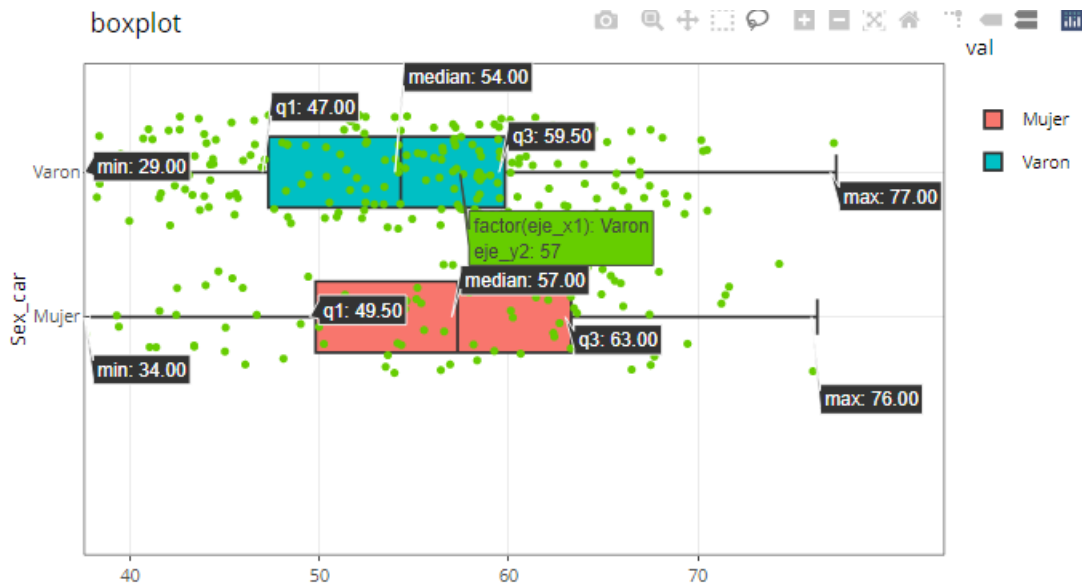


Figura 86: Elementos visuales disponibles

Si se desea, puede ser útil en casos poder realizar un zoom:

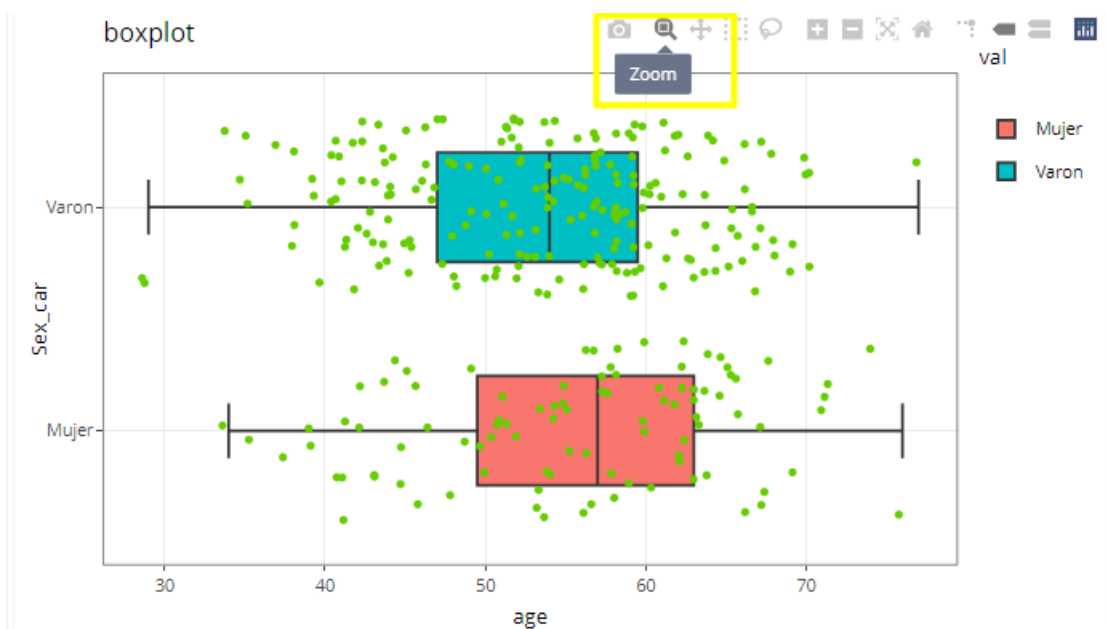


Figura 87: Acción Zoom

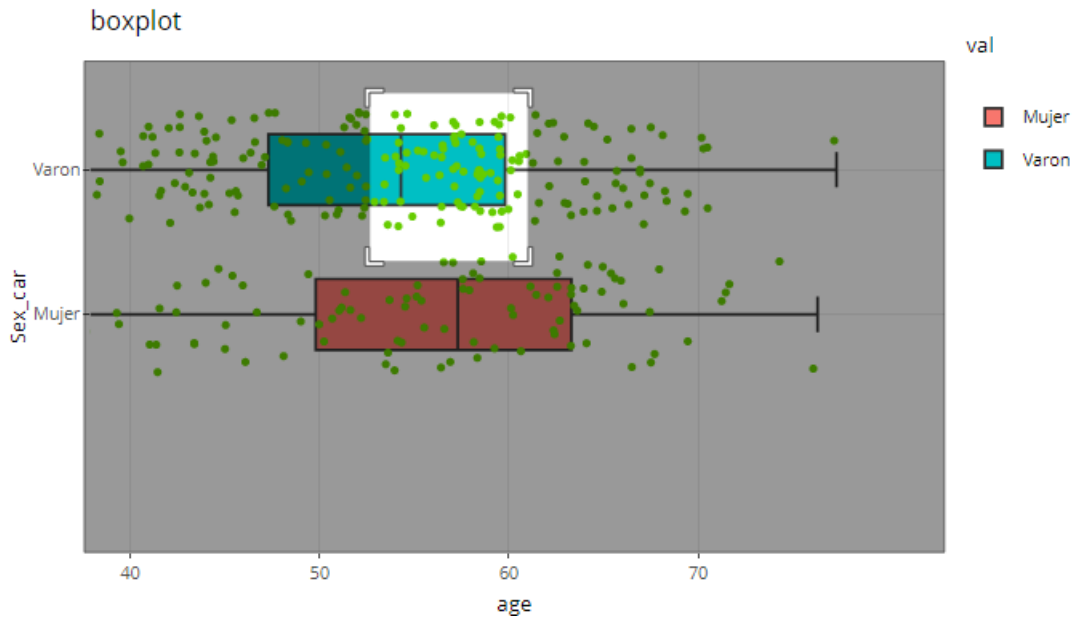


Figura 88: Selección Zoom

Dando como resultado:

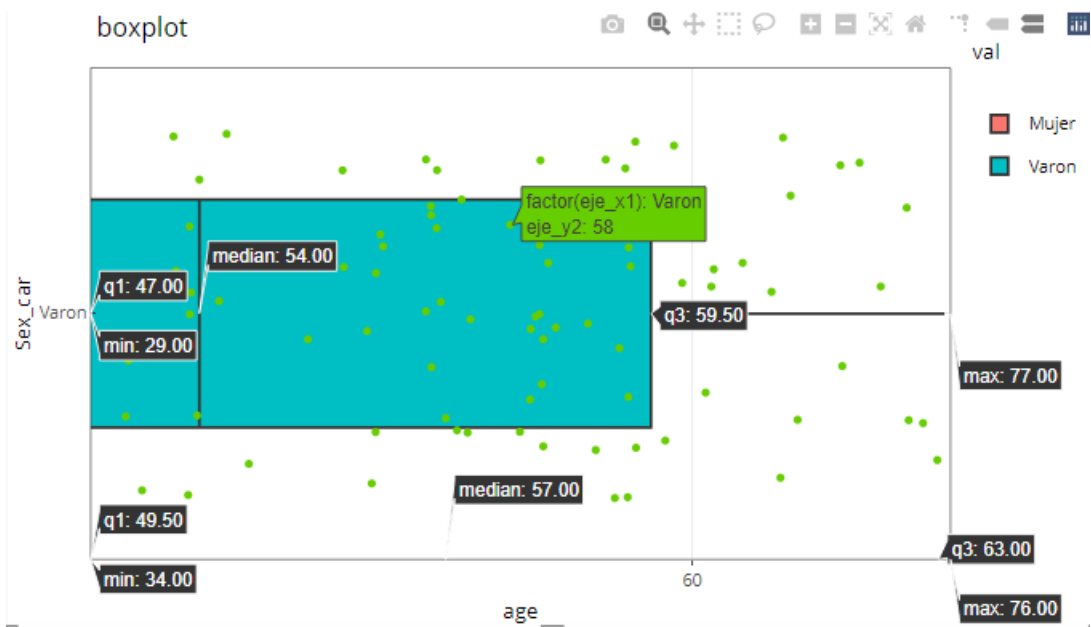


Figura 89: resultado Zoom

En el caso del gráfico violín, (violín es una presentación compacta de una distribución continua. Es una mezcla de `geom_boxplot()` y `geom_density()`), se puede indicar si se quiere recortar las colas con la opción `TRIM`, se puede ver que el último gráfico, la opción esta indicada como `FALSE`.

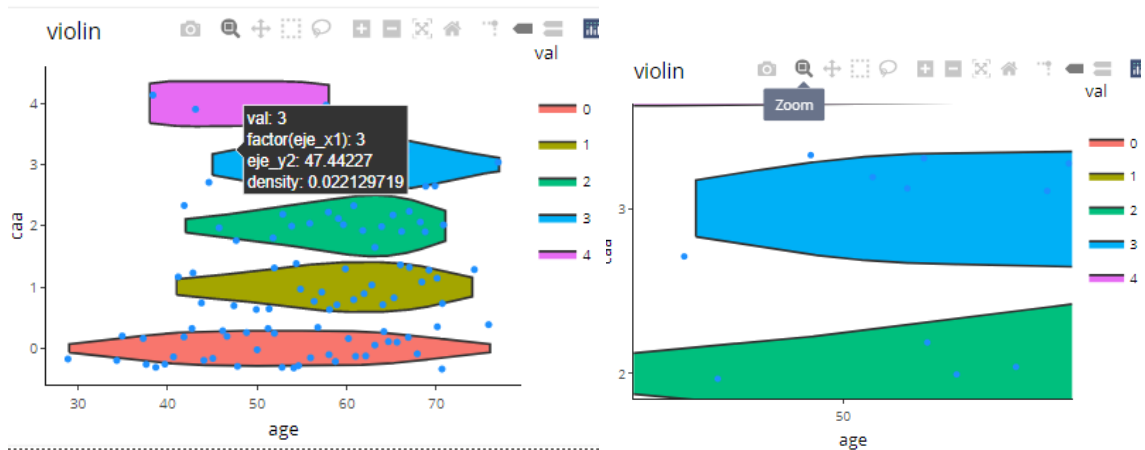


Figura 90: Gráfico violín

### ✓ Gráficos Histogramas/Barras

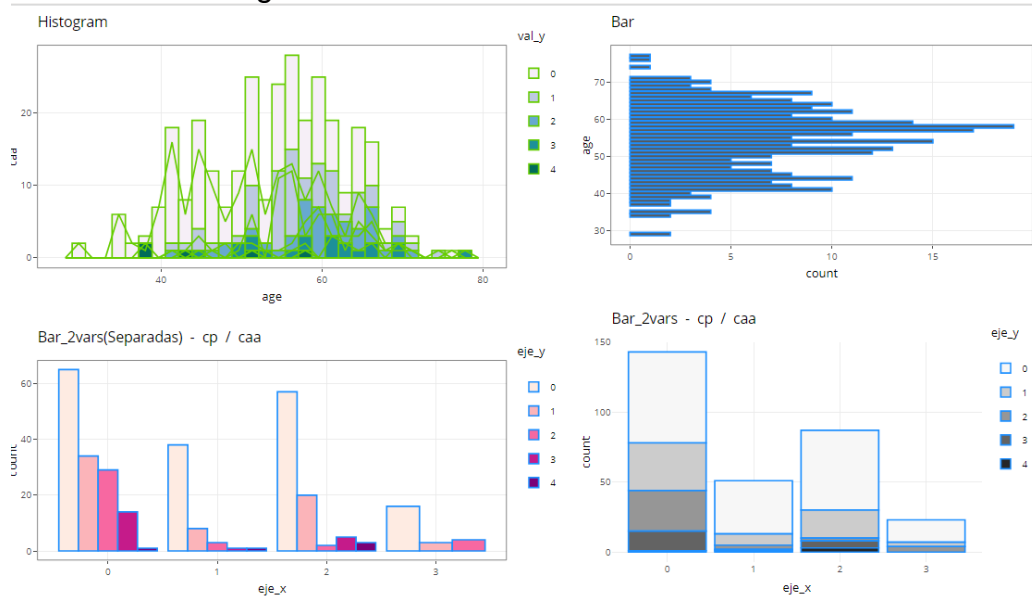


Figura 91: Gráficos Histogramas/Barras

Estos gráficos, permiten trabajar según se quieran ver datos con el tipo barras o con el tipo histograma.

Se puede observar que también se puede escoger la disposición visual del gráfico.

En el caso de escoger tratar con dos variables, se puede decidir, si verlo representado de forma apilada o separada.

De igual modo, se puede seleccionar el tema, de la lista predefinida, que dará formato general al gráfico.

## ✓ Gráficos Scatter

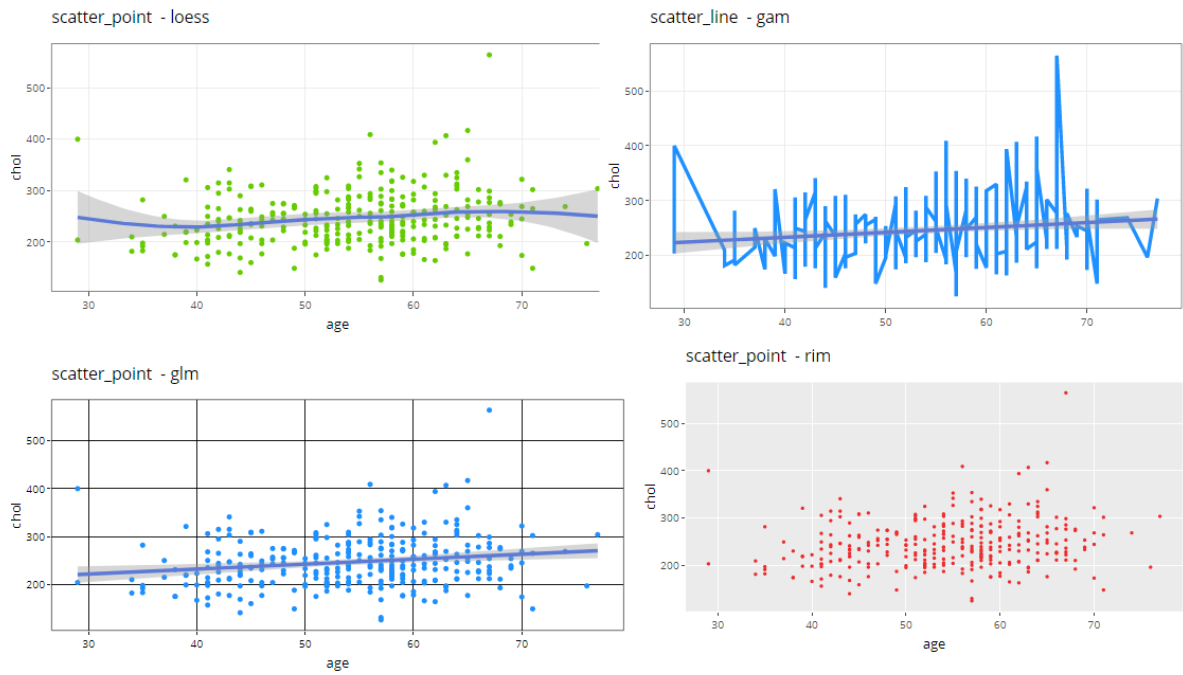


Figura 92: Gráficos Scatter

## ✓ Gráficos Scatter3D

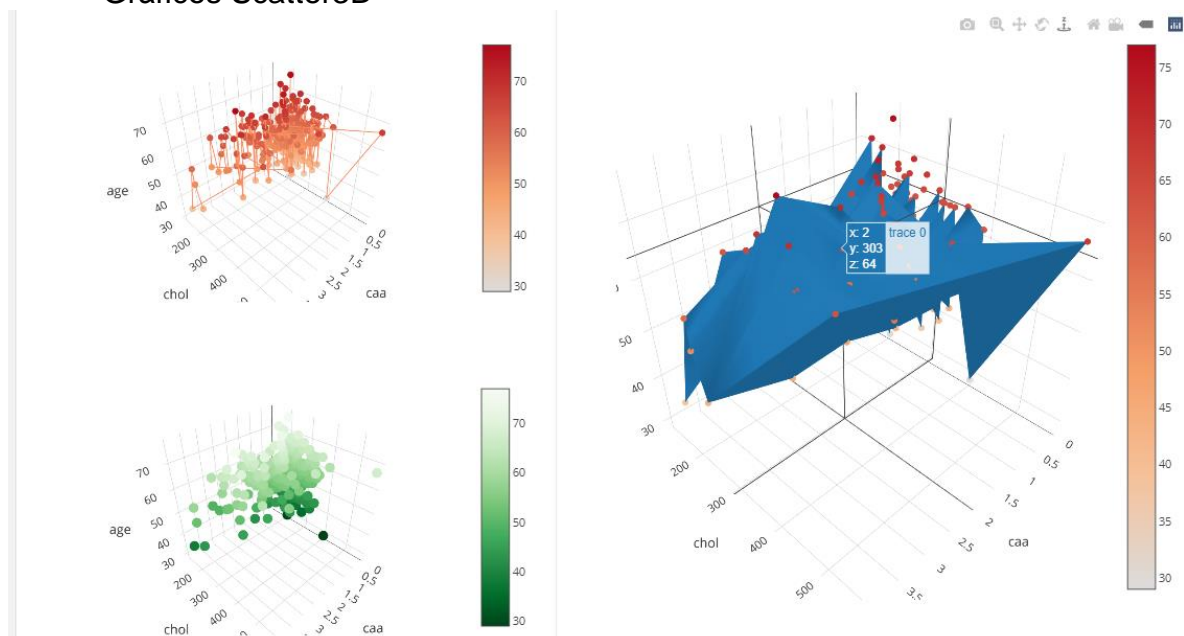


Figura 93: Gráficos Scatter 3D

Se podrá rotar y ver según el punto de vista que se desee.