**Visualización de datos con Código.**

La visualización de datos a través de código consiste en escribir comandos en algún lenguaje de programación para que, a través de la programación, estos se adapten la información de un conjunto de datos a una estructura gráfica que nos permita comprender mejor la información. Este método se contrapone a la visualización a través de programas y plataformas de visualización como Tableau, RawData o el más usado de todos: visualización de datos en Excel[[1]](#footnote-1).

**Ventajas:**

Comparado con Excel, la visualización a través de código nos permite lo siguiente:

1. **Reproducibilidad.**

La reproducibilidad y el análisis es más transparente y ordenado que en Excel, ya que sólo hay que analizar y correr un solo archivo de texto.

1. **Automatización.**

Una vez programada una gráfica, podemos extender el proceso a una cantidad masiva de datos, generando múltiples gráficas de distintos datos sin mayor complicación.

1. **Opciones Online.**

La visualización programada nos permite crear objetos listos para su distribución en la web. Igualmente, podemos añadir opciones para generar interactividad entre gráficas y bases de datos, añadir ventanas para desplegar información extra o eventos para realizar acciones particulares.

**Desventajas:**

La principal desventaja para usar este tipo de visualizaciones es la barrera de entrada que representa el aprender a programar en cierto lenguaje particular, la cantidad de tiempo que se requiere invertir para hacer una gráfica una primera vez, la cantidad de excepciones que hay que manejar y que emular o superar la estética de programas con plantillas hechas puede ser muy complicado para los principiantes.

**Opciones.**

Para visualizar datos existen múltiples opciones, quizá al menos una por cada lenguaje de programación con apartado gráfico. Sin embargo, aquí revisaremos los más comunes en el entorno de las ciencias sociales.

* **Visualización en Stata.**

Stata cuenta con funciones de visualización nativas en su lenguaje de programación, permitiéndonos realizar una gran cantidad de gráficas.

* **Visualización en R.**

R es uno de los lenguajes de programación más usados para visualizar datos, contando con una gran variedad de librerías que permiten visualizaciones estáticas y dinámicas. Entre estos se encuentran r-base, ggpot2, plotly, los pertenecientes a la familia htmlwidgets y algunos para usos más particulares, como highcharter, leaflet y algunos basados en d3.js.

La fortaleza de ggplot2 consiste en la forma de trabajo: permite estruturar las gráficas en capas (layers) que se van sumando unas con otras añadiendo cada una características particulares a nuestras gráficas. Está construido a partir de la “Gramática de las Gráficas”, una forma de mantener una estructura común permitiendo una modificación de Todos y cada uno de sus componentes.

Otra ventaja de visualizar en R es la de tener nuestro análisis y exploración de datos dentro de un mismo script y flujo de trabajo.

* **Visualización en Python**

Python cuenta con varias paqueterias de visualización, entre las que se encuentran matplotlib y seaborn, contanto también con una distribución de plotly para visualización dinámica.

* **Visualización en JavaScript y D3.js**

A diferencia de las opciones anteriores, la visualización de D3.js suele ser más complicada al utilizar, al mismo tiempo, objetos y archivos \*.html, \*.css y \*.js. Mientras que en los programas anteriores nos enfocamos más en el código y la forma de ordenar los elementos gráficos de las gráficas, en D3.js tenemos que programar prácticamente cada uno de estos elementos desde cero. Aquí, ya no programamos gráficas de barras, sino líneas con rectángulos encima. El entrar a los elementos más primitivos de las gráficas de esta manera, podemos realizar gráficas con efectos extremadamente personalizados y altamente complejos. Desde mi punto de vista, esta librería es más adecuada para el uso de diseñadores y programadores web, así como para proyectos grandes en los que se trabaje en equipo.

1. Excluyendo la visualización de datos hecha a mano. [↑](#footnote-ref-1)