Introducción a la visualización de datos

Juan Mascuñano Torres

1

Nombre de empresa o escuela Logo Etc
lunes, 17 de mayo de 2021

PROGRAMACIÓN EN PYTHON

Introducción

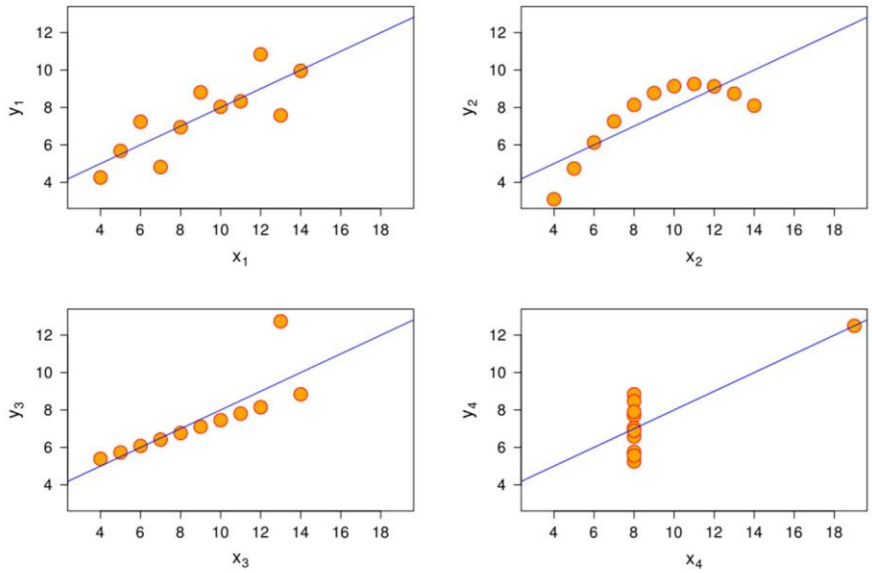
La visualización de datos es un ámbito de conocimiento en constante evolución que últimamente se ha visto impulsado debido a la gran cantidad de datos disponibles, los cuales esperan a ser analizados, interpretados y contextualizados mediante una narrativa que combina texto, imágenes y otros recursos interactivos, más allá de presentaciones estáticas que usan ciertos elementos gráficos como soporte. El *boom* de las redes sociales, el uso de dispositivos móviles y la práctica digitalización de todos los servicios (consumo, educación, ocio, etc.) permiten, en la práctica, disponer de datos sobre cualquier actividad humana que opere parcial o totalmente con la tecnología. Así, desde la aparición de internet a mediados de los años noventa, la capacidad de generar, procesar y compartir datos ha ido creciendo exponencialmente, se ha multiplicado por mil cada pocos años y ha obligado a adoptar nuevos prefijos (*mega-*, *giga-*, *tera-*, etc.) que reduzcan las cifras manejadas a cantidades razonables.

En este sentido, la visualización de datos es uno de los mecanismos de los cuales se dispone para presentar toda esta información de forma razonable para los usuarios finales, sin que estos se vean superados por semejante avalancha de datos. Una visualización de datos es un primer paso para el análisis y proyección de los datos disponibles, utilizando los recursos del sistema visual humano como procesador para detectar patrones, tendencias y/o anomalías. En este sentido, una visualización permite, de forma eficiente, medir y comparar datos, entre otras operaciones. Incluso un simple resumen de un conjunto de datos puede ser mejor transmitido y comprendido mediante una visualización que mediante el uso de texto y tablas numéricas.

Un ejemplo es el conocido como «cuarteto de Anscombe», creado por Francis Anscombe en 1973 para mostrar las limitaciones del uso de descriptores estadísticos para resumir un conjunto de datos, tal y como muestra la figura 1. Está compuesto por cuatro conjuntos de once puntos en el plano de la forma (*x*,

y) cada uno de ellos, de forma que la media y varianza de cada variable, así como la correlación entre ambas variables y el coeficiente de la recta de regresión óptima son idénticos para los cuatro conjuntos, siendo estos claramente diferenciables si se utiliza una representación visual. Obviamente cada conjunto representa el resultado de cuatro procesos diferentes: una colección de datos típica (figura superior izquierda), unos datos que siguen una relación no lineal (figura superior derecha), unos datos que siguen una relación lineal excepto uno de ellos, un posible *outlier* (figura inferior izquierda) y, finalmente, unos datos que muestran una relación no lineal entre las dos variables, pero donde un simple *outlier* genera un coeficiente de correlación elevado. Sin la visualización de estos datos usando un simple gráfico *x*-*y* es muy difícil hacerse a la idea de las cuatro distribuciones subyacentes en cada caso. Aunque se trata de un ejemplo sintético, muestra de forma convincente las limitaciones de los descriptores estadísticos más habituales en los trabajos de investigación y las posibilidades de la visualización como herramienta de análisis visual complementaria.

Figura1. El cuarteto de Anscombe.



Fuente: Wikipedia

Por lo tanto, acompañar (o incluso sustituir) los datos originales por una representación gráfica de estos puede ser muy efectivo para explicar el porqué de estos, especialmente por lo que respecta a la causalidad. No obstante, contar

historias por medio de los datos requiere combinar competencias y habilidades de diferentes áreas de conocimiento, incluyendo matemáticas y estadística, informática, psicología de la percepción y, por supuesto, diseño gráfico. Se trata de un ámbito claramente multidisciplinario donde normalmente se trabaja en equipo, aunque es necesario disponer de conocimientos básicos y de un vocabulario común en cada una de las áreas mencionadas.

Para ser un experto en visualización de datos es necesario, por lo tanto, disponer de un amplio abanico de conocimientos, lo cual no es sencillo ni rápido de adquirir, sino que exige un proceso continuo. Ya en 2010, Enrico Bertini (experto en visualización de datos y profesor de la New York University) lo describió desde su experiencia personal de la manera siguiente:

Enrico Bertini

Sitio web: http://fellinlovewithdata.com/
Twitter: @FILWD

Edward Tufte

Sitio web: https://www.edwardtufte.com
Twitter: @EdwardTufte

Nathan Yau

Sitio web: http://flowingdata.com/
Twitter: @flowingdata

1) Estudiar mucho, usando los recursos abiertos disponibles en la red y otros accesibles desde una biblioteca (p. ej. artículos en bases de datos). Bertini cita como ejemplo uno de los trabajos clave en el ámbito de la visualización escrito por Edward Tufte, llamado *The Visual Display of Quantitative Information*, que a pesar de ser antiguo (data de 1983) sigue siendo una piedra angular de cualquier trabajo relacionado con la visualización de datos, tal y como lo demuestran las casi diez mil citas, según Google Scholar, y la segunda edición del libro. **2) Robar** (o, mejor dicho, copiar) buenas prácticas, pero no limitándose a reproducirlas, sino absorbiendo los detalles y trucos que hacen que una visualización de datos funcione y sea una buena práctica. Bertini recomienda estar al corriente de las principales revistas y congresos internacionales del ámbito, así como conocer los trabajos de los principales autores (como Nathan Yau, por ejemplo). Actualmente, es posible seguir mediante el uso de redes sociales (principalmente Twitter) a los mejores especialistas del ámbito, dado que internet es mejor canal para difundir este tipo de trabajos especialmente cuando requieren interactividad.

3) Criticar, siendo capaz de detectar aquellos aspectos que diferencian lo que es una buena práctica de lo que no lo es, intentando en este caso detallar y corregir los aspectos

erróneos, lo que permitirá ver las dificultades del problema que hay que resolver.

4) Producir una buena visualización, poniendo en práctica los conocimientos adquiridos. Esto puede exigir el uso de herramientas más o menos complejas (desde Excel hasta Tableau, por ejemplo) o incluso lenguajes y bibliotecas de programación (Processing o D3, entre otros), lo cual determinará el grado de sofisticación alcanzable. Tal como dice Bertini, es posible crear visualizaciones excelentes independientemente de la tecnología usada. También es necesario disponer de datos que se visualizarán, aunque hoy día internet es un enorme repositorio de datos sobre los cuales hacerse preguntas interesantes y tratar de responderlas mediante una visualización.

Processing

Sitio web: https://processing.org/

5) Salir de la zona de confort, exponiendo la visualización creada al escrutinio público, especialmente mediante las redes sociales o el uso de repositorios y sitios web dedicados especialmente a compartir trabajos de este tipo. Los tres beneficios esperados son tener que darle más importancia y realizar un buen trabajo antes de exponerlo, tener que pensar sobre el uso que se le va a dar a la visualización y, especialmente, obtener *feedback* de otros usuarios interesados en la visualización de datos, de forma que sea posible detectar y corregir aquellos detalles que lo requieran.

Así pues, una manera de empezar con este plan descrito por Bertini es conocer algunos de los trabajos de autores clave en el ámbito. Con este objetivo, este material pretende guiar al lector a través de una serie de referencias básicas en el ámbito de la visualización de datos, con el objetivo de comprender qué es una visualización, los antecedentes históricos previos a la situación actual de abundancia de datos y tecnologías, el uso de visualizaciones interactivas para manipular y analizar dichos datos, así como los elementos que componen una visualización y determinan, tanto objetiva como subjetivamente, su percepción por parte del usuario final de la visualización. Se trata, por lo tanto, de proporcionar un vocabulario básico y una introducción a los principios que rigen la visualización de datos como mecanismo para la creación y transmisión de conocimiento. Es importante recordar en este momento la secuencia «datos, información,

conocimiento y sabiduría», y evitar la discusión sobre si se están visualizando datos o información, dado que siempre se trata de lo segundo, aunque se utilicen indistintamente los dos conceptos.

Esta guía no es, de ninguna manera, exhaustiva, sino que se trata de una serie de lecturas escogidas por su representatividad y por la importancia de sus autores dentro del ámbito y que, de alguna manera, están relacionadas entre sí por su contenido. Al contrario, constantemente se hará referencia a otros recursos (principalmente sitios web) que pueden usarse como punto de partida para alcanzar una visión más completa de este ámbito en constante evolución. Y, obviamente, esta guía no sustituye a las lecturas, sino que sirve de punto de entrada para ponerlas a todas en contexto.

Los artículos que constituyen esta guía de lectura (por orden de aparición) son los siguientes:

- *The Visual Display of Quantitative Information*. Edward Tufte, 1983. Graphics Press (vol. 2, núm. 9), Chesire, CT, EE. UU.

- «A brief history of data visualization». Michael Friendly, 2006. *Handbook of Data Visualization* (págs. 15-56).

- «Visual Representation». Alan Blackwell, 2011. *The Encyclopedia of Human-Computer Interaction* (2.ª ed.), (cap. 5).

- «What is Visualization?». Lev Manovich, 2010.

- «The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations». Ben Shneiderman, 1996. *Proceedings of the IEEE Symposium on Visual Languages* (págs. 336-343).

- «Data Visualization for Human Perception». Stephen Few, 2011. *The Encyclopedia of Human-Computer Interaction* (2.ª ed.), (cap. 35).

PROGRAMACIÓN EN PYTHON

- «Interactive Dynamics for Visual Analysis: a taxonomy of tools that support the fluent and flexible use of visualizations». Jeffrey Heer y Ben Shneiderman, 2012. *ACM Queue* (vol. 10, núm. 2).

- «D^3: Data-Driven Documents». Michael Bostock, Vadim Ogievetsky y Jeffrey Heer, 2011. *IEEE Transactions on Visualization and Computer Graphics* (vol. 17, núm. 12, págs. 2301-2309).

1.Pioneros de la visualización

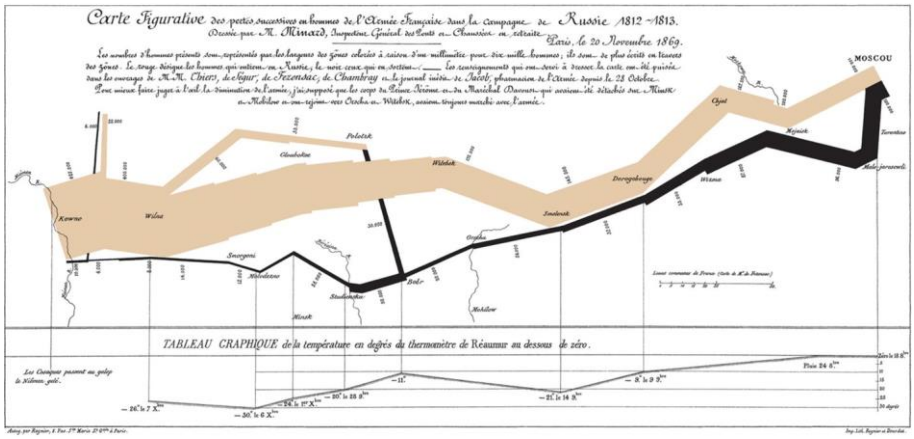
Uno de los primeros trabajos en el ámbito de la visualización de datos es *The Visual Display of Quantitative Information*, escrito por Edward Tufte en 1983. Tufte es profesor de la Universidad de Yale y está considerado como uno de los pioneros de la visualización de datos. Tufte introdujo el uso de diagramas como metodología habitual para la descripción de datos y su análisis preliminar, como una herramienta más parte de la estadística. Es el inventor del concepto *chartjunk* (diagrama basura), para criticar el mal uso de las visualizaciones cuando no aportan nada a los datos representados.

En este trabajo, Tufte desgrana una serie de visualizaciones de datos y las relaciona con el tipo de datos que se van a visualizar y el contexto en el cual fueron creadas, y también hace un repaso a algunas visualizaciones históricas, como el gráfico que describe las pérdidas del ejército francés durante la campaña de Napoleón en Rusia (1812-1813), creado por Charles Minard y mostrado en la figura 2. Tufte describe en detalle el uso de mapas, series temporales y su combinación, lo que él llama narrativas espacio-temporales, así como el uso de visualizaciones para mostrar relaciones entre elementos.

Pero lo más interesante del trabajo de Tufte es el concepto de *excelencia*, que define como la comunicación de ideas complejas con claridad, precisión y eficiencia. La excelencia es lo que proporciona al usuario de la visualización la mayor

cantidad de ideas en el más corto espacio de tiempo mediante el mínimo uso de tinta y en el espacio más pequeño posible. La excelencia es casi siempre multivariada, no depende de una sola variable. Y, finalmente, la excelencia requiere contar la verdad sobre los datos, algo que desafortunadamente parece haberse perdido en muchas visualizaciones de carácter político, en las que se utilizan las visualizaciones como herramienta de manipulación. En este sentido, cualquier persona interesada en ser un «periodista de datos» debería adoptar las ideas de Tufte como principios básicos.

Figura2. Mapa figurativo de las sucesivas pérdidas de hombres de la Armada Francesa en la campaña de Rusia 1812-1813, por Charles Minard (1869).



Fuente: Wikipedia

PROGRAMACIÓN EN PYTHON

2.Antecedentes históricos

Michael Friendly

Sitio web: http://www.datavis.ca/

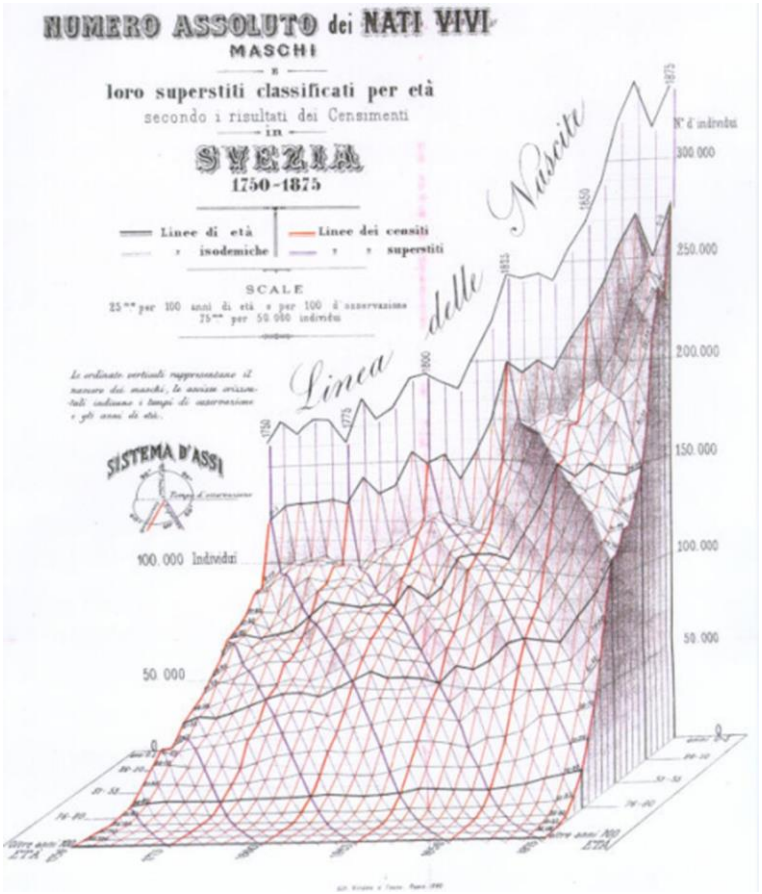
Aunque hay excelentes introducciones sobre qué es una visualización de datos y el significado del concepto a lo largo de la historia, la lectura recomendada para este apartado es el artículo «A brief history of data visualization», de Michael Friendly, publicado en el año 2006 como un capítulo de un manual de visualización de datos parte de una colección de libros de estadística, lo que da muestra de la importancia de la visualización como herramienta para el análisis de datos. Friendly es un autor muy prolífico en el ámbito de la visualización, impulsor del sitio web DataVis, en el cual se pueden encontrar muchos otros recursos relacionados, incluyendo artículos, libros y software.

El artículo de Friendly está estructurado según una línea temporal, que incluye desde las primeras visualizaciones (de hecho, mapas y diagramas) previas al siglo XVII hasta la actualidad (a partir de 1975), donde la tecnología ha hecho posible la creación masiva de visualizaciones. De hecho, una revisión del artículo hoy día seguramente añadiría una nueva sección dedicada al *boom* que ha tenido la visualización recientemente, por la disponibilidad masiva de datos y la aparición de tecnologías que permiten la creación de visualizaciones de datos dinámicas, con la utilización de la web como medio.

Una de las etapas más interesantes destacadas por Friendly es la ocurrida en la segunda mitad del siglo XIX, cuando se desarrollaron muchas técnicas para el análisis estadístico, que eran aplicadas a todos los ámbitos de la planificación social, la industrialización, el comercio y el transporte. Esto provocó la aparición de muchas innovaciones en la visualización de datos, necesarias para poder explicar los datos y fenómenos tan complejos de la sociedad del momento. Un primer paso fue la utilización de elementos 3D proyectados como vía de escape del plano que hasta el momento limitaba las posibilidades, como el ejemplo de la figura 3. Otro fue la combinación de mapas con datos de cada región, de forma que en una misma representación se combinaban datos

espaciales con otros temporales. Finalmente, el uso de gráficos para el análisis estadístico (la correlación era un concepto aún en desarrollo) permitió a Francis Galton y otros investigadores avanzar en la formalización de las observaciones realizadas y convertirlas en técnicas estadísticas, como muestra la figura 4. Un resultado de toda esta actividad fue la aparición de atlas estadísticos, es decir, informes de datos recopilados sobre casi todos los aspectos de la vida cotidiana acompañados de gráficos detallados. Friendly destaca la colección «Albums de Statistique Graphique», publicada anualmente por el Gobierno francés entre 1879 y 1897, y que fue discontinuada por su alto coste de producción, así como la realizada por el Gobierno de Estados Unidos entre 1872 y 1874.

Figura3. Población de Suecia entre 1750 y 1875, realizada por Luigi Perozzo en 1880.

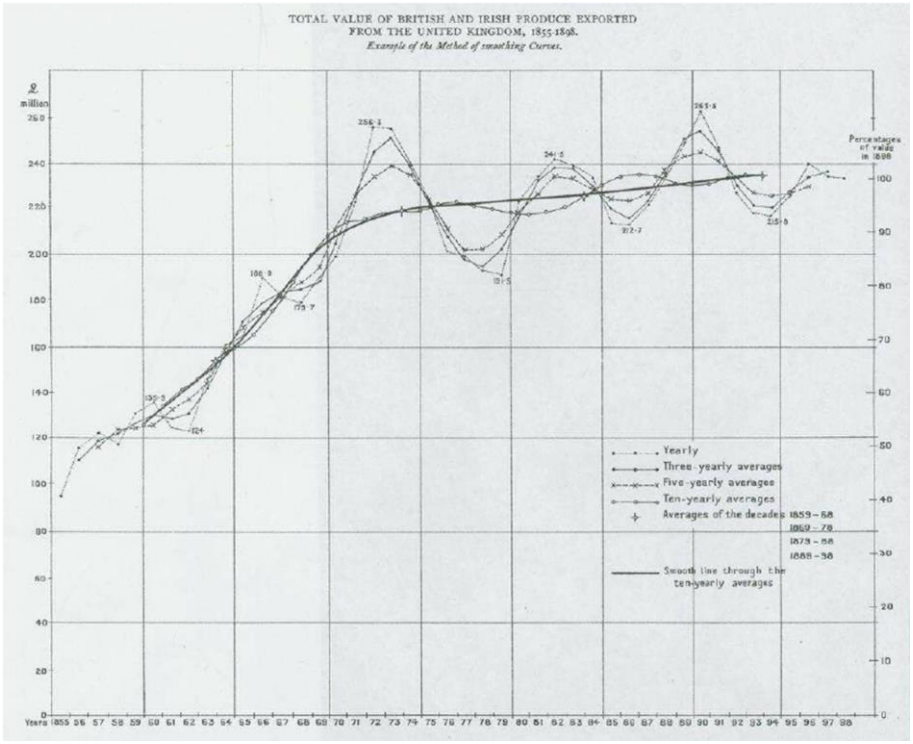


Fuente: datavis.ca

Friendly destaca varios elementos que han sido clave en la evolución de lo que actualmente se conoce como visualización de datos. Por una parte, el desarrollo y formalización de herramientas estadísticas para el análisis de datos, y la

consiguiente necesidad de representar los resultados obtenidos mediante estas. Por otra parte, la aparición de ordenadores y lenguajes de programación (como Fortran), que permitieron automatizar cálculos y crear las primeras representaciones gráficas para conjuntos de datos con cientos o miles de elementos.

Figura4. Visualización del efecto del suavizado en series temporales, por Arthur Bowley, 1901.



Finalmente, Friendly resume la evolución de la visualización de datos sobre la base de la necesidad de resolver problemas concretos, relacionados con el deseo de visualizar fenómenos y relaciones entre elementos de forma diferente, y esto último es posible gracias al desarrollo de las metodologías (análisis estadístico) y las tecnologías (ordenadores).

3.Representaciones visuales

Otra aproximación a la evolución del uso de la imagen a lo largo del tiempo se puede encontrar en el artículo «Visual Representation», de Alan Blackwell, publicado en 2011 como un capítulo de una enciclopedia del ámbito de la interacción

persona-ordenador (en inglés *human-computer interaction*), lo cual no resulta extraño en cuanto los seres humanos son, principalmente, visuales y los ordenadores han evolucionado para maximizar el uso de imágenes como interfaz para facilitar su uso.

Este trabajo tiene dos partes bien diferenciadas. En la primera, Blackwell describe mediante el uso de ejemplos los diferentes tipos de elementos que se han usado a lo largo de la historia para la representación de datos e información, desde los primeros textos hasta el uso de representaciones basadas en iconos para la creación de interfaces de usuario. En la segunda parte, Blackwell presenta una aproximación holística que engloba todos los elementos descritos en la primera, de acuerdo con tres dimensiones complementarias: el tipo de elemento o recurso utilizado, la correspondencia con la realidad representada y su uso en un contexto de diseño de la interacción.

Así pues, Blackwell inicia la primera parte mediante la descripción de los elementos que siempre han formado parte del conjunto de herramientas básicas para contar historias, empezando por el texto, el cual se modula mediante el uso de estructuras sencillas (párrafos, columnas y tablas en páginas) y sus propiedades (alineación, indentación, bordes y sombreados), junto con la elección de una tipografía adecuada (tamaño, familia y color). El concepto de texto es muy amplio, e incluye obviamente el uso de dígitos y otros símbolos propios de otros sistemas de representación, como por ejemplo el uso de letras griegas en el caso de ecuaciones y fórmulas matemáticas o la notación musical, un lenguaje en sí mismo fuertemente visual.

El siguiente elemento destacado por Blackwell son los mapas y diagramas, que son una evolución del concepto de símbolo, ya que permiten concentrar mayor cantidad de información y establecer relaciones entre los elementos que los componen, modificando parámetros como su posición y tamaño, por ejemplo. En el caso concreto de los mapas, usados desde tiempos ancestrales, el objetivo habitual es representar la realidad (una región de nuestro mundo 3D) mediante un conjunto de símbolos y etiquetas, que representan y describen los elementos que se desea destacar, como por ejemplo el mapa de

PROGRAMACIÓN EN PYTHON

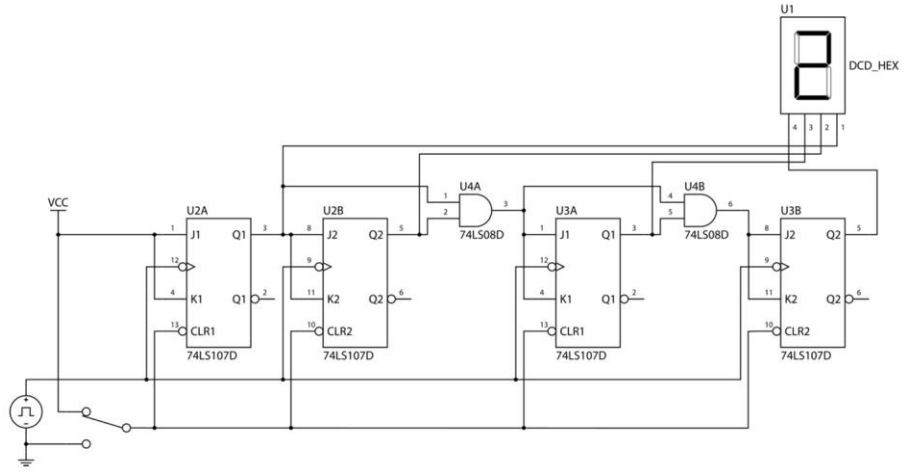
Juan de la Cosa, creado justo después del descubrimiento de América, mostrado en la figura 5.

Figura5. Mapamundi de Juan de la Cosa, hacia 1500, el primer mapa que representa el nuevo y recién descubierto continente, América.

Fuente: Wikipedia

Por su parte, los diagramas son representaciones esquemáticas de un elemento complejo, con el objetivo de simplificar su entendimiento, que se focalizan en aquellos aspectos que determinan su naturaleza, habitualmente con un conjunto de símbolos propio. Un ejemplo son los esquemas usados para la representación de circuitos electrónicos, como el mostrado en la figura 6. En los diagramas el tamaño y distancia relativos de los elementos que los componen no son tan importantes como en un mapa, dado que se está haciendo una abstracción de la realidad, eliminando detalles innecesarios.

Figura6. Diagrama de circuito de un contador TTL de 4 bits.



Fuente: Wikipedia

Un tipo especial de diagramas son aquellos que hacen énfasis en las relaciones entre los elementos que lo componen, de forma que la posición exacta pierde fuerza con respecto al orden o proximidad entre los elementos. Un ejemplo característico son los diagramas usados para la representación de las redes de transporte (especialmente el metro, como el mostrado en la figura 7), en los que la posición absoluta de los elementos no importa, y es la posición relativa (entre dichos elementos) la que aporta información. No son un mapa por lo que respecta a precisión, pero sirven igualmente de orientación. Los diagramas suelen representar estructuras tipo árbol o grafo, las cuales permiten describir relaciones (en algunos casos jerárquicas) entre elementos. Blackwell menciona la dificultad que puede comportar para algunos usuarios entender este tipo de visualización, especialmente si no se relativiza el concepto de distancia entre elementos.

Figura7. Diagrama de la red de transporte subterráneo de New York.

Fuente: Wikipedia

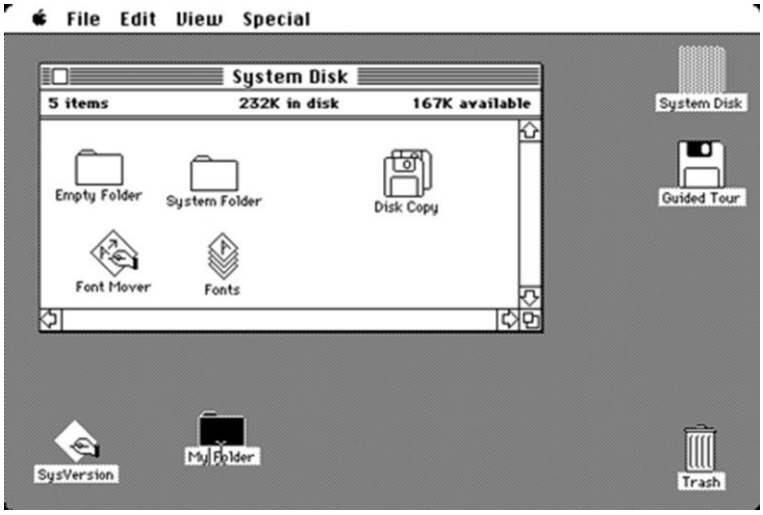
PROGRAMACIÓN EN PYTHON

El siguiente paso que describe Blackwell es no representar la realidad mediante un esquema, sino hacerlo directamente mediante una imagen, ya sea natural o sintética, de forma que sea posible hacerse una idea fidedigna del elemento representado. Por *natural* se entiende una imagen que reproduce un fragmento de la realidad («analógica»), mientras que por *sintética* se entiende una imagen que ha sido generada mediante un algoritmo o proceso («digital»). En el primer caso las imágenes se capturan mediante la tecnología existente, ya sea la pintura o la fotografía. En el segundo, se trata de imágenes generadas por ordenador, como las presentes en un videojuego o el resultado de un algoritmo. No obstante, la sofisticación de los gráficos generados por ordenador y su popularización en el mundo audiovisual han causado la difuminación de las barreras entre ambos tipos, por lo que en algunos casos es difícil determinar qué es real y qué no en una imagen.

El uso de imágenes para representar la realidad ha evolucionado también en lo que se conoce como *iconos*, es decir, símbolos que representan esquemáticamente un elemento de la realidad cotidiana, suplantándola y simplificándola, haciéndola fácilmente reconocible. Un ejemplo son los logos de las marcas comerciales, que sustituyen al propio nombre de la marca, como en el caso de Nike, cuyo logo (llamado Swoosh) es reconocido sin necesidad de texto alguno. En otro sentido, las interfaces de usuario actuales son un compendio de iconos que representan acciones (servicios) y/o contenidos (recursos), de acuerdo con ciertos criterios. Esto surge del concepto de metáfora visual, que fue explotado inicialmente por los investigadores de Xerox PARC, en Palo Alto (California), para el desarrollo de interfaces de usuario visuales, en un contexto en el que los ordenadores personales se usaban y programaban mediante una línea de comandos. Estas interfaces, basadas en el uso de iconos que representan elementos típicos de un entorno de trabajo o de una oficina, permitían un uso más sencillo al reducir la cantidad de texto que se tenía que leer y escribir (es decir, los comandos usados para las operaciones básicas). La metáfora de escritorio, desarrollada por Alan Kay en 1970, convierte la pantalla del ordenador en un espacio virtual que asemeja o reproduce un entorno de trabajo físico habitual, Apple fue

la que lo popularizó en 1984 con la interfaz de usuario del Apple Macintosh, mostrada en la figura 8.

Figura8. Escritorio del Apple Macintosh en 1984.



Fuente: Wikipedia

Yuri Engelhardt

Sitioweb: http://datagood.org/

Twitter: @YuriEngelhardt

Finalmente, Blackwell se basa en la tesis doctoral de Yuri Engelhardt para clasificar todos estos elementos según tres dimensiones: el tipo de recurso gráfico utilizado, su correspondencia con algún concepto del mundo real que desea ser representado y su utilización como parte del diseño que determina la visualización. Los elementos se agrupan en cuatro categorías:

- Marcas: son los atributos de más bajo nivel, incluyendo forma, orientación, tamaño, textura, saturación, color y tipo de línea. En este caso, la correspondencia con el concepto representado puede ser literal (una imitación de alguna de las características físicas), un mapeo a una escala relativa o convencional (arbitraria). Los usos de este tipo de recursos son marcar

  la posición, identificar categorías (mediante formas, texturas y colores), e indicar dirección y magnitud, mediante el uso de símbolos y códigos de color sencillos.

- Símbolos: incluye elementos geométricos, texto, logos e iconos, elementos pictóricos y elementos de conectividad. La correspondencia puede ser topológica (enlazando elementos), descriptiva (usando convencionalismos pictóricos), figurativa (mediante metonimia o bromas visuales), connotativa (asociada a aspectos culturales o

profesionales) o adquirida (en el caso de alfabetos especializados). Se usan para la representación de textos y el cálculo simbólico, diagramas, marcas o logos, retórica visual y la definición de regiones.

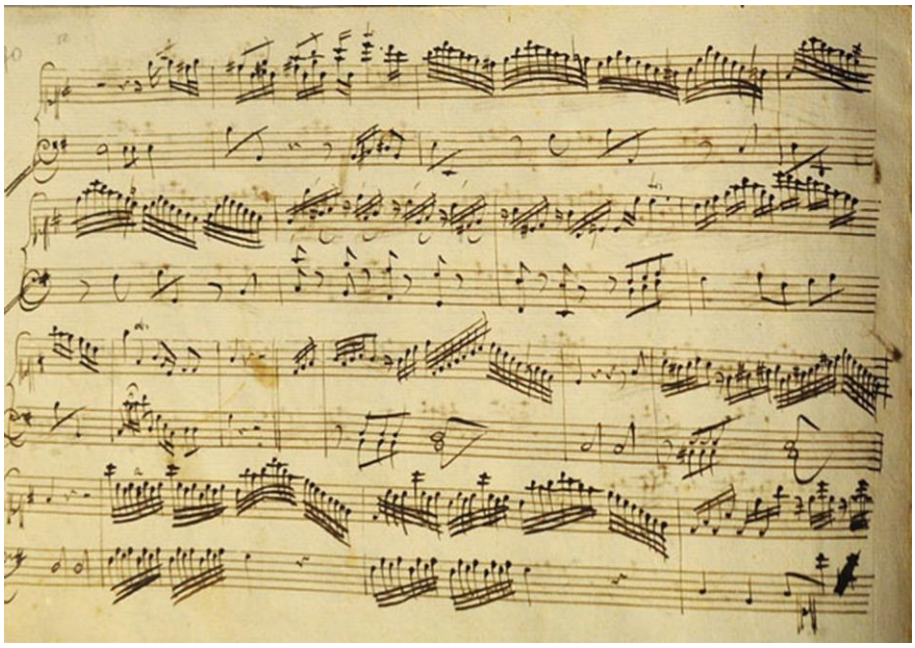
- Regiones: este concepto incluye rejillas para alinear elementos, bordes, marcos, rellenos, el uso del espacio en blanco y la idea de integración de la Gestalt para dar más valor al todo que a las partes que lo componen. Así, se pueden crear contenedores para elementos de más bajo nivel, separarlos, encuadrarlos (en un sentido fotográfico, integrándolos en una composición) y crear capas. Los usos son identificar una pertenencia compartida a un cierto conjunto o categoría, la separación de elementos diferentes en paneles o bien la colocación de etiquetas, subtítulos o leyendas.

- Superficies: es la generalización de más alto nivel, e incluye los objetos físicos (3D) donde se mapea (superpone) el elemento gráfico. En este caso, la correspondencia puede ser literal (como en un mapa), euclídea (respetando escalas y ángulos), métrica (según unos ejes cuantitativos), yuxtapuesta u ordenada (en regiones y/o categorías), esquemática o bien situada en un contexto. En este nivel, los usos son los habituales en la visualización de datos: la creación de diseños tipográficos, gráficos y diagramas, incluyendo los relacionales, interfaces visuales, etc.

Con esta clasificación, Blackwell proporciona unos criterios básicos para el análisis de cualquier representación visual, indicando qué elementos deben identificarse y cuál es su significado, siguiendo el orden propuesto. Un ejemplo indicado por el autor es una partitura (mostrada en la figura 9), la cual parte de unas formas básicas (líneas, círculos…) y una disposición (el pentagrama) para representar algo tan complejo como una sinfonía. Existe un orden determinado por la lectura del pentagrama, de izquierda a derecha y de arriba a abajo, incluyendo múltiples capas (p. ej. las anotaciones que realiza el director o autor sobre la obra).

Figura9. Pentagrama de una composición de Mozart en su infancia.

PROGRAMACIÓN EN PYTHON



Fuente: AP.

Nombre de empresa o escuela Logo Etc
lunes, 17 de mayo de 2021

PROGRAMACIÓN EN PYTHON

4.¿Qué es una visualización?

Lev Manovich

Sitio web: http://manovich.net/
Twitter: @manovich

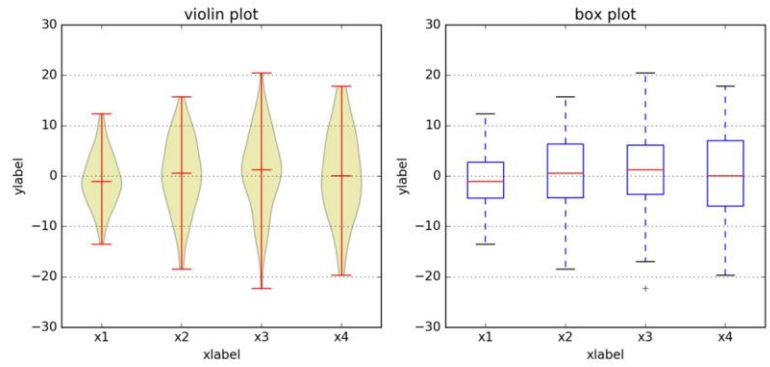
Este apartado toma el nombre de un famoso artículo del no menos famoso Lev Manovich, conocido teórico por sus trabajos sobre los medios de comunicación y las transformaciones motivadas por la adopción de las nuevas tecnologías. El artículo fue publicado en 2010, bajo el título «What is Visualization?». En dicho artículo el autor presenta un análisis de los principios que han sido clave en el desarrollo del ámbito, especialmente por lo que respecta a los nuevos medios que han impulsado y popularizado el uso de visualizaciones para narrar historias. El sitio web de Manovich es también una referencia para todos los interesados en la visualización de datos, pero desde una perspectiva más amplia, e incluye proyectos relacionados con el uso social de la imagen (por ejemplo, la moda del *selfie* o autofotos) así como otros de carácter más teórico.

En este artículo, Manovich revisa su definición inicial de 2002, en la cual define una visualización como «una transformación de datos cuantificados no visuales en una representación visual (de estos)». El autor se preocupa por definir «visualización de información» (*infovis*, abreviadamente) de la forma más inclusiva posible, teniendo en cuenta la diversidad de trabajos que podrían cobijarse bajo semejante término. Manovich empieza mostrando la diferencia entre visualización de información y visualización científica, e indica que esta última (según otros autores) se limita a datos numéricos, mientras que la primera engloba otros conceptos de semántica más compleja, como el texto o las redes y grafos. Manovich no hace semejante distinción según dicho criterio, ya que, según él, la mayoría de las visualizaciones combinan datos numéricos y no numéricos. Para Manovich, la principal diferencia entre visualización de información y visualización científica es el uso de tecnologías diferentes y el hecho de que provengan de culturas diferentes, del diseño en el primer caso y del ámbito científico en el segundo. Igualmente, Manovich se pregunta si la visualización de información es diferente del diseño de información, en este caso (abusando del lenguaje) es una

cuestión de visualizar datos *versus* visualizar información, respectivamente. No obstante, Manovich rechaza una distinción tajante y considera que todos los términos se solapan.

Según Manovich, desde la segunda mitad del siglo XVIII hasta la actualidad han existido dos principios clave que han dado forma a la visualización de información. El primero es el principio de reducción, que consiste en el uso de primitivas gráficas (puntos, líneas, formas geométricas simples…) para la representación de elementos y sus relaciones, que revelan patrones y estructuras subyacentes, sin necesidad de visualizar los datos originales. Esto ha conllevado una pérdida de importancia de los datos con respecto a sus representaciones, demasiado esquemáticas en algunos casos. Un ejemplo es el resumen de un conjunto de datos mediante descriptores estadísticos, como el ya mencionado cuarteto de Anscombe. El más sencillo es la media, acompañado habitualmente de la varianza, que indica cómo de centrados alrededor de la media están los datos. El siguiente paso es usar *box-plots* para describir los cuartiles, mostrando la distribución de los datos y la existencia de posibles *outliers*. Actualmente se usan los *violin plots*, que integran el histograma como parte de la visualización, añadiendo información sobre la distribución real de los datos, tal y como muestra la figura 10. En función del nivel de detalle deseado y de la naturaleza de los datos, se puede optar por una representación u otra.

Figura10. Ejemplo de *violin plot* como extensión del *box-plot* equivalente.



Este reduccionismo, presente en todos los ámbitos de las ciencias, propone que el mundo puede ser analizado sobre la base de los elementos simples que lo componen y las reglas que rigen sus interacciones, de forma que pueda ser posible

comprender la totalidad mediante una descripción simplificada o reducida. Así, durante el siglo XIX se desarrollaron todos los gráficos típicos para representar estos datos «reducidos» que permiten explicar aspectos sociales, demográficos, etc. Fue en esta época cuando aparecieron los gráficos de barras y de tarta, los histogramas, etc., todos ellos conceptualizados desde dicha visión reduccionista y usando los mismos elementos.

Así, el segundo principio es del uso de variables espaciales (posición, tamaño, forma, etc.) para representar diferencias en los datos y revelar así los patrones y relaciones existentes más importantes. En el ejemplo (ficticio) de la figura 10 se pueden observar las diferencias entre cuatro clases diferentes por lo que respecta a la distribución de una cierta variable con respecto a cada clase. Manovich hace notar que la visualización de información privilegia las dimensiones espaciales sobre otras, dando más importancia a la topología y a la geometría, y menos a otros aspectos como el color, la saturación o la transparencia. Así, para representar un conjunto de datos, las dimensiones más importantes son asignadas a la disposición espacial (llamada *layout*), mientras que el resto de las dimensiones se mapean habitualmente al resto de las variables visuales (color, etc.). En este caso, el color y/o la forma se usan para particionar los elementos de un conjunto de datos en diferentes clases.

Manovich hace una reflexión teórica sobre el porqué de esta distinción, es decir, ¿por qué la organización geométrica de los elementos en una representación ha de ser más importante (para la percepción humana) que las otras dimensiones visuales (color, etc.)? Manovich menciona el hecho de que todo objeto ocupa una única parte del espacio como posible razón, dado que el cerebro utiliza esta información para segmentar el mundo tridimensional en una colección de objetos diferentes que probablemente conforman identidades diferentes (por ejemplo, gente, el cielo, el suelo, edificios, etc.). El propio concepto de arte moderno y contemporáneo en corrientes como la abstracción se ha basado en romper con la tradición de identificar y representar entidades en su contexto, dando más importancia al color sobre la forma, por ejemplo. Manovich menciona también la dificultad de

reproducir representaciones gráficas mediante la tecnología existente, lo cual limita el uso del color, la transparencia, etc. Han sido los ordenadores los que han permitido crear y manipular representaciones más complejas, potenciando el uso de otras dimensiones visuales.

Manovich prosigue entonces con el concepto de visualización sin reducción (o visualización directa), en la que los datos adoptan una mayor importancia que en el caso anterior. Un ejemplo son los *tag clouds* o nubes de palabras clave, popularizados por diferentes herramientas y redes sociales aparecidas con la web 2.0, como Flickr o los blogs, entre otros. Un *tag cloud* muestra la frecuencia de aparición de cada palabra en un texto, de forma que rápidamente es posible hacerse una idea de los conceptos que aparecen en él. La figura 11 muestra una nube de palabras con las doscientas palabras más frecuentes del artículo de Manovich. Aunque sería posible usar un gráfico de barras para representar la misma información, la visualización directa proporcionada por la nube de palabras es mucho más rica, al mismo tiempo que estéticamente más agradable. No es importante saber el porcentaje de veces que aparece la palabra «visualization», sino que se puede ver rápidamente que es una de las más usadas. De hecho, una visualización directa interactiva podría proporcionar esa información si el usuario se sitúa encima de una palabra en concreto, por ejemplo, es decir, aportando detalles solamente cuando estos son requeridos, a diferencia de una visualización basada en la reducción en la que son estos detalles, precisamente, los que gobiernan la visualización.

Figura11.*Tag cloud* con las palabras más usadas en el artículo de Manovich, «What is Visualization?».

Fuente: Elaboración propia (Tagxedo).

Manovich describe otros ejemplos de visualización directa o sin reducción. **Brendan Dawes**

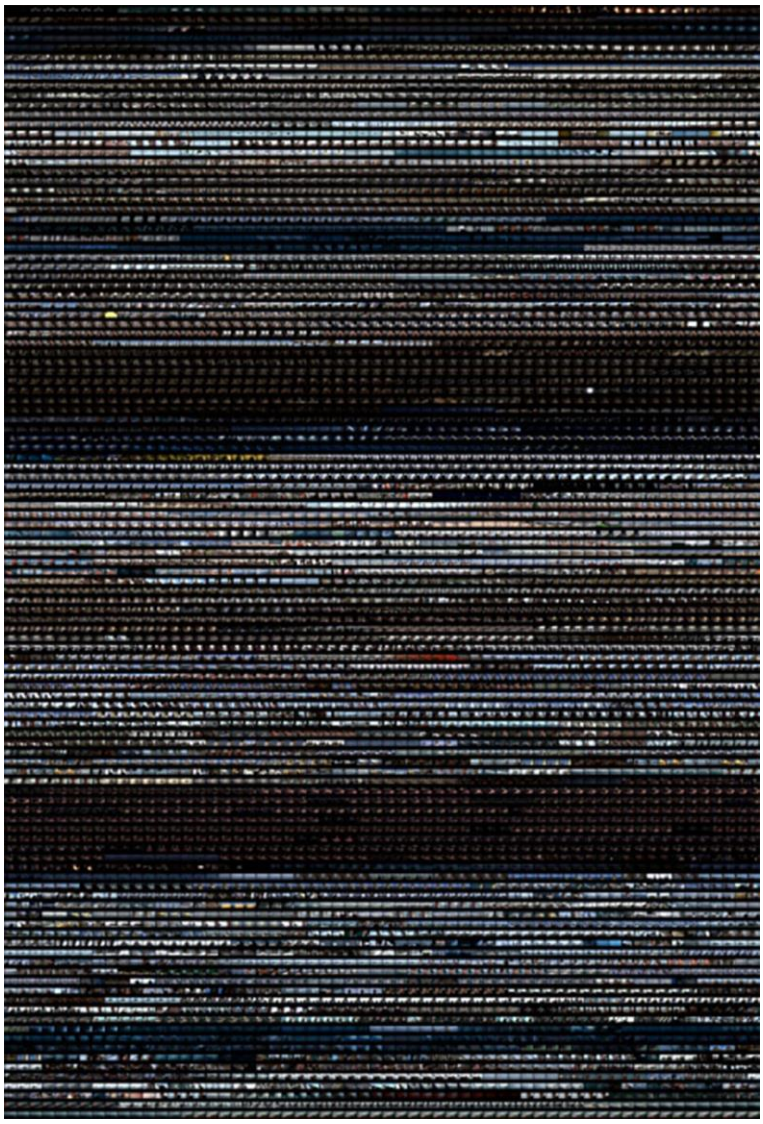
Uno de ellos es el famoso «Cinema Redux» de Brendan Dawes, creado en 2004.

Sitio web: http://

Dawes toma un fotograma por segundo de una película y lo reduce a una ima- www.brendandawes.com/ gen de un tamaño de 8 x 6 píxeles. Luego toma sesenta de estas imágenes y Twitter: @brendandawes las ordena como elementos de una línea, añadiendo tantas líneas como mi- Cinema Redux: www.brendandawes.com/prohttp:// nutos dura la película, tal y como muestra la figura 12. Así, Dawes visualiza jects/cinemaredux toda una película en una sola imagen, pero sin recurrir a una reducción (aun el título de la obra). Como bien indica Manovich, no se trata de una reducción propiamente dicha, sino un muestreo o *sampling*, de forma que solamente se muestran algunos de los datos originales (un fotograma de cada veinticuatro –a 24 fps–, un pixel de cada siete mil doscientos –a una resolución de 720 x 480–). Si el autor hubiera utilizado el color promedio de cada fotograma o la imagen promedio de cada veinticuatro *frames* sí que se trataría entonces de una visualización reduccionista.

Figura12. Resumen visual de «Jaws», por Brendan Dawes, 2004.

PROGRAMACIÓN EN PYTHON



Fuente: Brendan Dawes.

Este tipo de visualización es imposible de realizar sin una tecnología que permita acceder al contenido y manipularlo según unas reglas. Dawes utilizó el lenguaje de programación Processing para capturar los fotogramas de cada película, reducirlos de tamaño y reordenarlos en forma de matriz. Estas nuevas visualizaciones aparecen de las posibilidades que ofrece la tecnología para manipular datos en su totalidad, no mediante el uso de las reducciones habituales. Manovich destaca que Dawes usa fotogramas reales, no el color promedio. No es necesario reducir los datos (a un solo número, usando descriptores estadísticos) para destacar patrones en estos datos, sino que se puede utilizar muestreo, usando los datos originales. Además, no es necesario seguir ninguna configuración espacial concreta, sino que se utiliza el orden natural de los datos.

PROGRAMACIÓN EN PYTHON

Manovich finaliza su artículo con una reflexión alrededor del concepto de visualización directa, y se plantea si es o no un método diferente de la visualización de información, la cual sigue actualmente basándose en el uso de primitivas gráficas. Dado que las visualizaciones directas también permiten extraer patrones y revelar estructuras en los datos, Manovich plantea que pueden ser consideradas visualizaciones tradicionales, aun cuando en la mayoría de los casos provengan de autores con *backgrounds* muy diferentes al tradicional del ámbito estadístico o científico. Ha sido la disponibilidad de tecnologías como Processing lo que ha permitido a otros autores crear visualizaciones en las que los parámetros que determinan su aspecto no son los tradicionales, aprovechando las posibilidades de manipulación por igual en todos ellos. No obstante, sigue existiendo una limitación en forma del ancho de banda necesario para transmitir imágenes (mapas de bits, datos completos), así que los gráficos tradicionales (vectoriales, basados en reducciones) seguirán siendo omnipresentes, aunque cada vez más complejos. En este sentido, lenguajes como Processing o mejor aún D3 son la prueba de que Manovich tiene razón cuando dice que la visualización directa o sin reducción es un nuevo paradigma para ser explorado, especialmente desde ámbitos como las ciencias sociales y las humanidades.

PROGRAMACIÓN EN PYTHON

5.Tipos de datos y operaciones

Aunque relativamente antiguo (data de 1996), el artículo «The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations» de Ben Shneiderman sigue siendo una referencia (ha sido citado más de tres mil setecientas veces según Google Scholar) por lo que respecta a diseño de interfaces visuales para la manipulación de datos. Citando al crítico de arte Ernst Hans Josef Gombrich, Shneiderman presentó este trabajo en un simposio sobre lenguajes visuales, mostrando la necesidad de categorizar los elementos que componen una representación visual y las operaciones que se derivan de ellos.

En este trabajo, Shneiderman introduce su famoso mantra sobre cómo debe plantearse la búsqueda visual de información: «Overview first, zoom and filter, then details-on-demand», es decir, primero una visión general, después seleccionar y ampliar la zona de interés, y finalmente añadir el detalle necesario según las necesidades del usuario. A partir de esta idea, Shneiderman define siete tareas que pueden ser aplicadas a conjuntos de datos, que también son (coincidentemente) de siete tipos diferentes:

- Datos 1-dimensionales: datos organizados secuencialmente, que son consumidos según el orden como que se han generado. Por ejemplo, documentos de texto, listas de elementos, etc.

- Datos 2-dimensionales: datos que representan una cierta estructura espacial, por lo tanto, incluyendo el concepto de posición. Un ejemplo obvio son los mapas o diagramas.

- Datos 3-dimensionales: en algunos casos, se desea representar la posición de un conjunto de elementos en el mundo real, y en este caso es necesario utilizar tres dimensiones. Muchos de los primeros esfuerzos en el ámbito de la visualización de datos pertenecían a esta categoría, como por ejemplo los sistemas de información geográfica, los entornos de diseño asistidos por ordenador (CAD) para

el modelado de piezas o en la arquitectura, o las imágenes resultantes de procesos de barrido del cuerpo humano.

- Datos temporales: son aquellos relacionados con eventos que tienen un inicio y un fin, y que pueden solaparse. Es importante no confundirlos con datos 1-dimensionales que son generados por un proceso con un cierto ritmo, como por ejemplo un conjunto de mediciones de la temperatura diaria en un lugar.

- Datos multidimensionales o n-dimensionales: en la mayoría de los casos, los ítems almacenados en una base de datos se describen mediante *n* atributos, lo cual dificulta su visualización dadas las limitaciones de nuestro mundo tridimensional.

- Árboles (datos jerárquicos): en este caso se trata de datos que tienen una relación entre ellos, en la que cada elemento o nodo tiene un enlace a su (único) *ancestor*, excepto en el caso del nodo raíz.

- Redes: cuando las relaciones entre nodos son más generales y no existen restricciones, los árboles generalizan en grafos, los cuales pueden ser de muchos tipos diferentes (dirigido, bipartido, acíclico, etc.). Actualmente el análisis y visualización de datos de redes es una de las áreas con mayor interés (por ejemplo, el análisis del flujo de contenido en Twitter).

Sobre estos tipos de datos, Shneiderman define siete operaciones básicas, según la idea repetida en el mantra («Overview first, zoom and filter, then details-on-demand»). Obviamente para cada tipo de datos, el significado de la operación puede ser ligeramente diferente:

- *Overview*: se trata de crear una vista general de todos los datos disponibles, como punto de entrada a la exploración de estos. La vista general puede no incluir todos los datos al mismo tiempo, pero entonces debe proporcionar un mecanismo para que el usuario pueda seleccionar otros datos al mismo nivel de detalle.

- *Zoom*: se trata de un ajuste de la vista anterior, seleccionando un subconjunto de datos, pero manteniendo la sensación de posición y el contexto, acercándose o alejándose de los datos. El zoom complementa la operación anterior para poder hacerse una idea de la estructura subyacente en los datos.

- *Filter*: el usuario puede seleccionar un subconjunto de los datos, de forma que solamente visualice aquellos que cumplan unos ciertos criterios, eliminando el resto. Los criterios pueden especificarse de muchas maneras, dependiendo del tipo de datos: en el caso de una variable numérica 1-dimensional, se puede especificar un rango, mientras que en una 2-dimensional se utiliza el concepto de *bounding box*, o caja que encierra un conjunto de datos.

- *Details-on-demand*: una vez, mediante las operaciones anteriores, ya se ha seleccionado un subconjunto reducido de datos, el usuario puede solicitar información de cada uno de ellos, normalmente mediante el uso del ratón, pasando por encima o haciendo clic en los datos, de forma que aparezca la información adicional contenida en cada uno de ellos.

- *Relate*: en el caso de datos n-dimensionales, es posible establecer criterios de distancia entre los elementos del conjunto, según su tipología y el uso que se le quiera dar a la visualización. Mediante esta distancia es posible entonces establecer relaciones entre elementos (cercanos) que comparten una o más características.

- *History*: se trata de mantener una lista de las operaciones realizadas, de forma que sea posible deshacer alguno de los pasos realizados durante el proceso de visualización de los datos.

- *Extract*: finalmente, una vez se han seleccionado los datos y la información adicional requerida, se trata de poder volcarlos para poder reutilizarlos en otro contexto, almacenándolos como un nuevo conjunto de datos.

Finalmente, Shneiderman propone una manera sencilla para que usuarios no expertos puedan crear selecciones complejas (es

decir, filtros) a partir de los datos disponibles, combinando operadores booleanos sencillos de forma dinámica, obteniendo visualizaciones de datos adaptadas a sus necesidades. Si la visualización no es usable y no permite al usuario responder a sus preguntas, entonces no es útil. Shneiderman, de forma casi profética, finaliza su artículo destacando que el uso de ordenadores ha generado una explosión de datos difícilmente gestionable, pero que a la vez es posible utilizar los ordenadores para visualizar dichos datos mediante diferentes herramientas conocidas (que combinan las operaciones descritas anteriormente) así como otras (usando palabras textuales del autor) que todavía deben ser domadas y validadas.

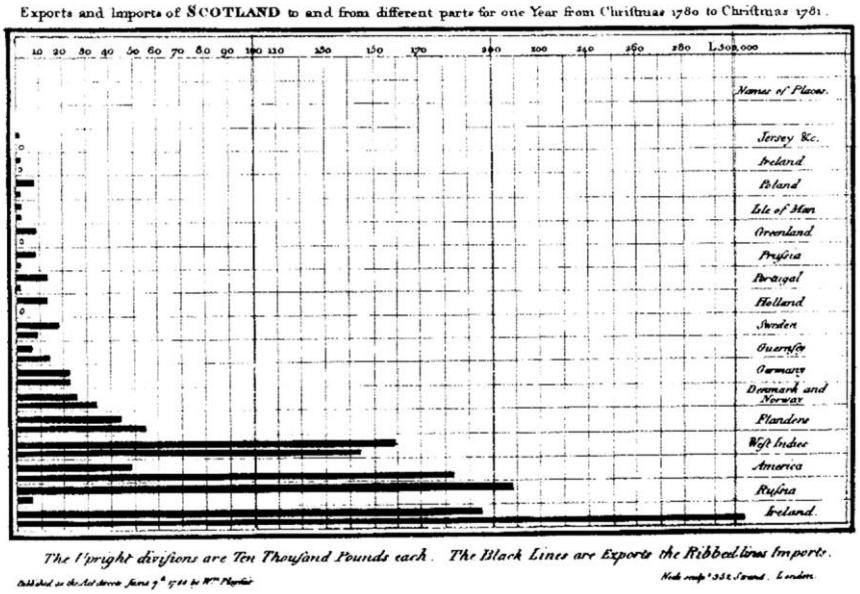
6.Principios de diseño

También forma parte de la enciclopedia de la interacción persona-ordenador el artículo «Data Visualization for Human Perception», publicado en 2011, que presenta qué aspectos relacionados con la percepción son importantes a la hora de tomar decisiones respecto al diseño de una visualización. En este artículo, Stephen Few describe unos principios básicos de percepción visual y cognición que se han de respetar, de forma que sea posible traducir una idea abstracta a un conjunto de atributos físicos, entre otros la forma, posición, tamaño y el color, de los elementos que compondrán la visualización.

Es un dicho popular que «una imagen vale más que mil palabras», pero esto solamente es cierto si la historia se puede narrar mejor visualmente y está bien diseñada. Por ejemplo, analizar tablas con datos numéricos no da una idea de su estructura, es mucho mejor visualizarlos y detectar rápidamente tendencias, patrones, máximos y mínimos, etc. Este es el poder de la visualización de datos, poder captar rápidamente los aspectos más característicos de un conjunto de datos mediante las capacidades del sistema visual humano. Lo mismo sirve para otros tipos de datos, como las redes o estructuras de elementos relacionados entre sí. Few describe la evolución del uso de gráficos para la representación de datos desde el trabajo pionero de William Playfair (1759-1823), quien usó por primera vez gráficos de diferentes tipos (de líneas, de barras y de tarta), como muestra el ejemplo de la figura 13.

Figura13. Importaciones y exportaciones de Escocia a otras regiones del mundo, por William Playfair, 1786.

PROGRAMACIÓN EN PYTHON



Fuente: Wikipedia

Usando diferentes ejemplos, Few introduce las ideas que deberían regir cualquier visualización de datos:

- Una visualización debería indicar claramente cómo los valores mostrados se relacionan (comparan) los unos con otros, o cada uno con la totalidad.

- Representa las cantidades de forma precisa.

- Facilita la comparación entre cantidades.

- Facilita establecer el orden de los elementos según la cantidad que representan, es decir, detectar máximos y mínimos.

- Deja claro cómo debería usarse la visualización y cuáles son sus objetivos y para qué debería usarse.

Así, se puede observar que un gráfico tan extendido como el gráfico de tarta no es eficiente, puesto que no facilita la comparación entre cantidades ni permite reordenar fácilmente los elementos una vez representados. Es mucho más efectivo un gráfico más sencillo como el de barras, principalmente porque se percibe mejor una dimensión lineal (una barra) que una angular (un segmento de tarta).

PROGRAMACIÓN EN PYTHON

Según Few, la visualización de datos funciona porque cambia el equilibrio entre la percepción y la cognición, aprovechando las capacidades visuales del cerebro humano. Few se basa en la psicología de la Gestalt, una corriente de la psicología moderna que trata (entre otros) del concepto de percepción, para describir cómo los seres humanos perciben patrones, sus formas y la organización de estos cuando se visualiza algo. Algunos de los principios que determinan estos aspectos son:

- Principio de proximidad: los objetos que se encuentran cerca unos de otros se perciben como un grupo.

- Principio de similitud: los objetos que comparten atributos similares (por ejemplo, su forma o color) se perciben como un grupo.

- Principio de adjunción: los objetos que parecen tener una frontera o borde alrededor de ellos (una línea o un área de color) se perciben como un grupo.

- Principio de clausura: una estructura abierta (parcialmente) se percibe como cerrada, completa y regular siempre que exista la posibilidad razonable de ser interpretada de dicha manera.

- Principio de continuidad: los objetos que están alineados o bien aparecen uno a continuación de otro se perciben como un grupo.

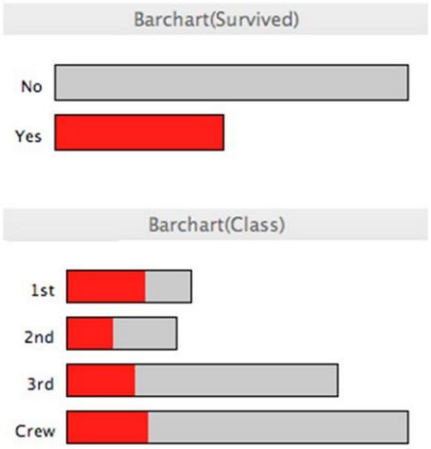
- Principio de conectividad: los objetos que están conectados (por ejemplo, mediante una línea) se perciben como grupo.

Estos principios y las ideas anteriormente mencionadas se pueden entender mejor mediante los avances en dos ámbitos de estudio: el procesamiento visual preventivo (en inglés, *preemptive*) y los mecanismos y limitaciones de la atención y la memoria. Es sabido que el procesamiento visual es más rápido que el verbal. En parte, porque el sistema visual humano realiza tareas de bajo nivel muy rápidamente dado que están codificadas mediante circuitos neuronales específicos, para detectar atributos básicos como longitud, tamaño, color (tono e intensidad), ángulo, textura y forma, entre otros. Además, las limitaciones para procesar y recordar múltiples elementos simultáneamente hacen que sea más

eficiente utilizar una visualización. Según Few, la visualización de datos puede extender la capacidad de análisis, ya sea mediante el uso de visualizaciones simples pero efectivas o bien, en un futuro, mediante nuevos usos, incluyendo el uso de interfaces complejas para interactuar con visualizaciones de forma sencilla, integrando el análisis estadístico y el uso de minería de datos para la extracción de conocimiento.

Finalmente, se puede destacar un comentario de Robert Kosara respecto al concepto de metáfora visual, muy extendido actualmente y mencionado con anterioridad en el artículo de Alan Blackwell. Kosara, uno de los responsables actuales de la herramienta de visualización de datos Tableau y del blog eagereyes, presenta brevemente la idea de metáfora visual como una manera de forzar la visualización, de acuerdo con otros criterios (quizás puramente estéticos), especialmente por lo que respecta a la interacción. Kosara describe diferentes operaciones que se pueden usar para manipular datos en una visualización interactiva, desde las más sencillas asociadas al movimiento del ratón sobre los diferentes elementos que la componen hasta el *linking* (cuando una visualización muestra el mismo subconjunto de datos en diferentes vistas mediante los mismos atributos, estableciendo una relación o enlace entre ellos) y el *brushing* (cuando el usuario selecciona diferentes elementos para crear un subconjunto de datos de interés), tal y como muestra la figura 14. Así, una vez se han seleccionado (*brushing*) los pasajeros que sobrevivieron al accidente del *Titanic* en la figura superior izquierda, la visualización muestra los mismos pasajeros repartidos según las otras categorías (dimensiones) usadas para clasificarlos, lo que permite ver, por ejemplo, que el porcentaje de mujeres que sobrevivieron es mucho mayor que el de hombres.

visualización de datos sobre los pasajeros y la tripulación del *Titanic*.



Robert Kosara

Sitio web: http://kosara.net/ y https://eagereyes.org/ Twitter: @eagereyes **Figura14**. Ejemplo de *brushing* y *linking* en una

7.Herramientas de análisis visual

Jeffrey Heer

Sitio web: http://homes.cs.washington.edu/~jheer/

Twitter: @jeffrey_heer

Junto con Ben Shneiderman, Jeffrey Heer revisa en el artículo «Interactive Dynamics for Visual Analysis: a taxonomy of tools that support the fluent and flexible use of visualizations» (publicado en 2012) las operaciones básicas descritas anteriormente, pero desde una perspectiva de las herramientas y tecnologías con las que se ejecutan. De hecho este artículo desarrolla el trabajo previo de Shneiderman mencionado con anterioridad y clasifica doce acciones o dinámicas en tres grandes bloques, añadiendo algunas nuevas y revisando otras respecto a la lista inicial de siete tareas del artículo de Shneiderman:

- Especificación de los datos y la vista: incluye visualizar datos mediante una representación pictórica, filtrar los datos no relevantes para focalizarse en los que sí lo son, ordenar los datos para exponer los posibles patrones y, finalmente, derivar valores o modelos a partir de los datos.

- Manipulación de la vista: incluye seleccionar elementos para poder destacarlos, aplicar sobre ellos filtros o manipularlos, navegar por los datos para poder detectar patrones de alto nivel así como el detalle a bajo nivel, coordinar vistas para explorar datos multidimensionales y organizar espacios de trabajo y ventanas múltiples.

- Procesado y procedencia de los datos: incluye almacenar el histórico de análisis realizados para poder revisarlo, revisitarlo y compartirlo, crear anotaciones sobre la base de los patrones descubiertos, compartir vistas y anotaciones para promover la colaboración y guiar a los usuarios a través de tareas e historias (narrativas).

Los autores describen cada una de las operaciones y actualizan el trabajo anterior de Shneiderman, teniendo en cuenta los avances tecnológicos que ha habido desde entonces. Así, para cada una de las operaciones mencionadas, los autores proporcionan también ejemplos usando diferentes herramientas para la visualización de datos, incluyendo R y ggplot2,

Protovis (el precursor de D3), Tableau o IBM Many Eyes, entre otros. El objetivo de los autores es mostrar la visualización de datos como una herramienta más para el análisis y la toma de decisiones, así como la narración de historias basadas en los datos.

En este sentido, uno de los objetivos de cualquier visualización de datos es permitir un análisis preliminar de estos, de forma que sea posible detectar patrones, tendencias, etc. No obstante, esto puede ser difícil con los datos originales, debido a su naturaleza, por ejemplo. La visualización debería integrar la transformación de los datos originales, la creación de nuevas variables, el cálculo e integración (en la propia visualización) de descriptores estadísticos, así como la creación de modelos (estadísticos o de minería de datos) para la extracción de conocimiento, facilitando la detección de grupos o de las variables más relevantes, por ejemplo. Los autores destacan que esta es una de las áreas (conocida como *visual analytics*) en las que aún es necesario avanzar, combinando (superponiendo) dos ámbitos que hasta ahora se habían planteado secuencialmente, el análisis y la visualización.

La navegación por los datos que forman parte de la visualización es otra de las operaciones que toma mayor importancia, especialmente en escenarios *big data*, donde existe un gran volumen de datos muy diversos y cambiantes en el tiempo. El mantra de Shneiderman («Overview first, zoom and filter, then details-on-demand») puede no ser adecuado si la primera operación debe lidiar con un hipercubo (el producto cartesiano de las tres dimensiones mencionadas, volumen, variedad y velocidad de los datos, las tres uves del *big data*) enorme, por lo que es necesario establecer mecanismos para facilitar una aproximación inversa: «search, show context, expand on demand», es decir, ir del elemento seleccionado (encontrado en una búsqueda) hasta la totalidad. El cambio de paradigma provocado por lo que se conoce como *big data* ha causado también la necesidad de repensar lo que se entiende por una visualización, dado que no es posible (ni tiene sentido) visualizarlo todo. Aunque la capacidad de cálculo de los ordenadores es brutal, lo es aún más el volumen de datos que se generan y/o se capturan, por lo que es necesario replantearse todas las operaciones descritas por los autores, para hacerlas eficaces y eficientes.

Finalmente, los autores plantean esta taxonomía como un punto de partida para interesados en el ámbito de la visualización de datos, y proporcionan una gran cantidad de referencias bibliográficas y ejemplos, identificando los aspectos clave para el desarrollo de visualizaciones y las áreas de investigación de más interés, como nuevos métodos para la especificación de vistas interactivas, la integración de análisis y visualización de datos o el uso de anotaciones para añadir semántica a las visualizaciones, con lo que facilitan también su utilización.

8.Introducción a D3.js

Finalmente, una pregunta obvia una vez se llega a este punto es «¿cómo?», es decir, de qué manera se pueden construir estas visualizaciones más allá del uso de una herramienta concreta, desde la perspectiva de un desarrollador de aplicaciones o, mejor dicho, de visualizaciones de datos. Aunque ya existían diferentes bibliotecas de software para crear gráficos y visualizaciones más o menos complejas (por ejemplo, prefuse o Protovis, este último es el antecesor de D3, o el lenguaje de programación Processing), es a finales de 2011 cuando Michael (Mike) Bostock, Vadim Ogievetsky y Jeffrey Heer presentan D3 en el artículo «D^3: Data-Driven Documents», publicado en la revista *IEEE Transactions on Visualization and Computer Graphics*, una de las más prestigiosas del ámbito.

El nombre dado a esta nueva biblioteca, D3 (o también D3.js, puesto que se trata de una biblioteca escrita en JavaScript), no es gratuito, ya que lleva implícito un cambio de paradigma en la manera como se construye una visualización. D3 permite crear gráficos interactivos combinando elementos de las diferentes capas que componen una página web: lenguaje HTML, manipulación de los objetos que constituyen el documento mediante nodos DOM (*document object model*), aplicación de estilos mediante CSS (*cascading style sheets*) y, obviamente, JavaScript, con lo que se obtienen gráficos en formato SVG (*scalable vector graphics*) que pueden ser visualizados directamente por cualquier navegador web. Así, *data-driven documents* (D3) hace referencia al hecho de enlazar

directamente datos con elementos del DOM, que permite su actualización inmediata según los cambios que se produzcan en los datos y genera de esta manera visualizaciones interactivas y dinámicas. D3 no es un nuevo lenguaje de programación para la creación de gráficos, la aproximación habitual de otras bibliotecas predecesoras, sino que permite pensar en el documento (es decir, la página web resultante) como una representación visual de un conjunto de datos, según una configuración (*layout*) y un conjunto de parámetros que la determinan.

En este artículo, de carácter predominantemente técnico, los autores describen los aspectos más importantes de D3 por lo que respecta a su funcionamiento interno:

- Selección: permite acceder a un conjunto de elementos (uno, varios o todos) del documento, ya sea por nombre, clase, identificador, atributo, etc.
  Es posible combinar selecciones usando intersecciones y/o uniones.

- Operadores: sobre los elementos seleccionados, se pueden aplicar operadores como establecer y/o cambiar atributos, estilos, propiedades y contenidos, tanto textuales como HTML.

- Datos: el operador *data* permite establecer vínculos entre los datos de entrada y los elementos que conformarán la visualización. Los datos pueden ser ordenados y filtrados según diferentes criterios. El modo como D3 procesa los datos y los vincula a los elementos es una de las cuestiones clave que hay que entender para dominar su funcionamiento. Los datos pueden «entrar» (*enter*) en el documento (es decir, ser mapeados a los nodos), ser actualizados (*update*) o bien ser eliminados (*exit*).

- Eventos: se proporcionan mecanismos para supervisar la interacción mediante los elementos habituales (teclado y ratón).

- Configuraciones (*layouts*): uno de los aspectos más interesantes de D3 es la existencia de diferentes *layouts* para crear visualizaciones, como si fueran plantillas, incluyendo la mayoría de los más habituales, entre otros

diagramas de Sankey, *treemaps*, grafos de fuerza dirigidos y mapas, tal y como muestra la figura 15.

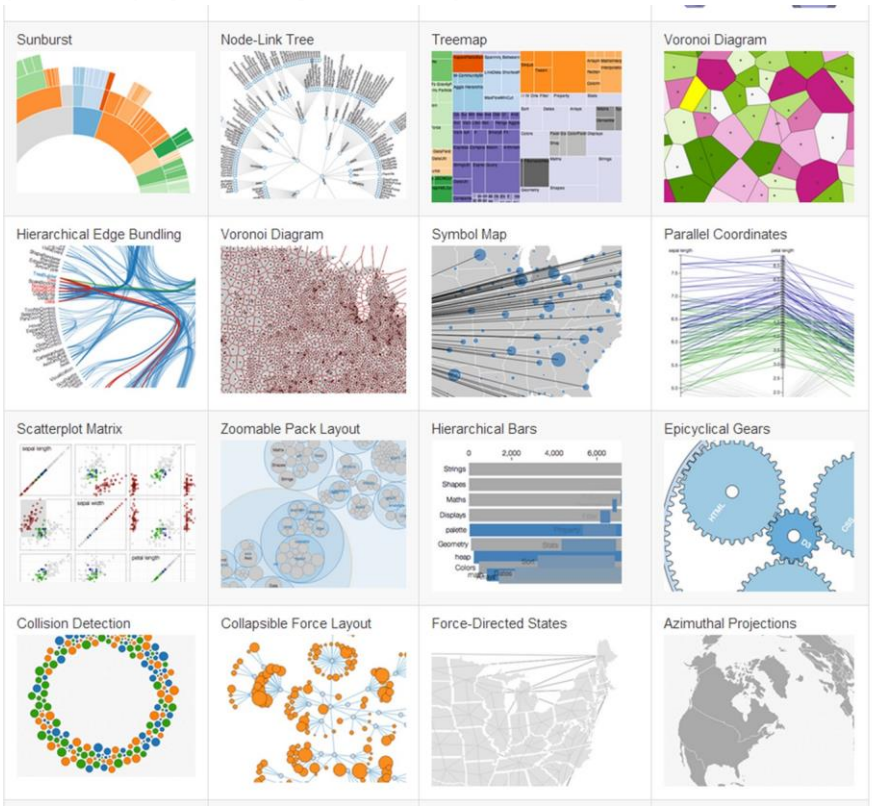
- Otros: finalmente, D3 también incluye una colección de primitivas gráficas, el uso de transiciones animadas, o el uso de interpolación para crear nuevos elementos a partir de unos básicos (colores, fuentes, etc.).

Los autores también analizan un aspecto clave en el desarrollo de visualizaciones de datos interactivas (la razón por la cual usar D3): el rendimiento o coste computacional necesario para visualizar un conjunto de datos. Es obvio que una visualización que necesite mucho tiempo para renderizar y mostrar un conjunto de datos no será bien recibida por los usuarios, así que D3 ha sido pensado para ser muy eficiente en este aspecto, tanto en la carga inicial como en la actualización de los datos. No obstante, D3 no está pensado para visualizar conjuntos de datos con cientos de miles o millones de elementos, no hay que olvidar que se ejecuta en el cliente (es decir, el navegador web), por lo que es necesario transmitir y cargar los datos en memoria, así como procesarlos, lo cual puede ser costoso para ciertos *layouts* como los grafos de fuerza dirigidos, por ejemplo.

Mike Bostock

Sitio web: https://bost.ocks.org
Twitter: @mbostock
D3.js: https://d3js.org/

Figura 15. Ejemplos de *layouts* en D3.js.



PROGRAMACIÓN EN PYTHON

Fuente: D3js.org

D3 se ha convertido en el estándar de facto para la creación de visualizaciones interactivas en línea, en parte debido a la gran cantidad de buenas prácticas que se han desarrollado a partir de él y que lo han popularizado, junto con un acertado diseño visual muy ligero y muy moderno. El propio Mike Bostock lideró diversos proyectos de visualización de datos mientras estuvo en el New York Times, como por ejemplo el análisis de la disputa electoral en el año 2012 por la presidencia de Estados Unidos entre el demócrata Barack Obama y el republicano Mitt Romney.

D3 sigue siendo un proyecto vivo que ha sido actualizado recientemente y el número de tutoriales y ejemplos de uso sigue creciendo desde su lanzamiento en 2011, por lo que es una muy buena apuesta para aquellos desarrolladores que quieran crear visualizaciones de datos interactivas. Aunque la curva de aprendizaje de D3 es empinada (en el sentido de que es complicado empezar a usarlo desde cero), sí que es posible reutilizar los ejemplos existentes, adaptando los datos disponibles a los requerimientos de cada *layout*. El hecho de ejecutarse en un navegador web también facilita la creación de visualizaciones sin necesidad de disponer de un entorno de desarrollo complejo.