



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

Despliegue de datos tabulares

Visualización de Información
IIC2026

Profesor: Denis Parra

Despliegue de datos tabulares

IIC2026 - Visualización de Información

Fernando Florenzano Hernández
faflorenzano@uc.cl

Planificación semanal

- *Deadline* formación de grupos de proyecto.
- *Deadline* Tarea 1: domingo 9 de septiembre, 19:59.
- *Deadline* Entrega 1: domingo 23 de septiembre, 19:59.

	Semana	Martes	Ayudantía	Jueves
7 y 9 ago	1	Intro + ¿Qué es visualización?	Tunear HTML/SVG/CSS (framework)	Javascript I (ayudantía)
14 y 16 ago	2	Data abstraction	feriado virgencita	Task abstraction
21 y 23 ago	3	Análisis y validación	d3 intro + versiones	Marcas y canales
28 y 30 ago	4	Percepción	d3 plot estáticos (enter, exit, update)	Rules of thumb
4 y 6 sept	5	Tablas (+altair): Fernando	d3 animaciones	Redes (I)
11 y 13 sept	6	Redes (2)	D3: grafos	Datos Espaciales
18 y 20 sept	7	feriado fiestas patrias	feriado fiestas patrias	Color
25 y 27 sept	8	Manipulación	D3: datos espaciales	Manipulación 2
2 y 4 oct	9	Presentación Hernán	D3: manipulacion/interactividad	Presentación Cristobal
9 y 11 oct	10	IR / Minería Texto	D3: texto	Visualización de Texto
16 y 18 oct	11	PRESENTACIONES		
23 y 25 oct	12	Series de Tiempo (Nebil)	D3: vistas coordinadas	Charla Invitada
30 oct y 1 nov	13	Casos de Estudio I	D3: Series de tiempo/streamgraphs	feriado dia de los muertos
6 nov y 8 nov	14	Casos de Estudio II		Visualización de Algoritmos
13 nov y 15 nov	15	Invitado de Socvis E. Graells		
20 nov y 22 nov	16			
27 nov y 29 nov	17	Presentaciones finales		

Contenidos

Organizar datos tabulares

Expresar valores cuantitativos

Regiones categóricas

Orientación de ejes

Densidad espacial

Contenidos

Organizar datos tabulares

Expresar valores cuantitativos

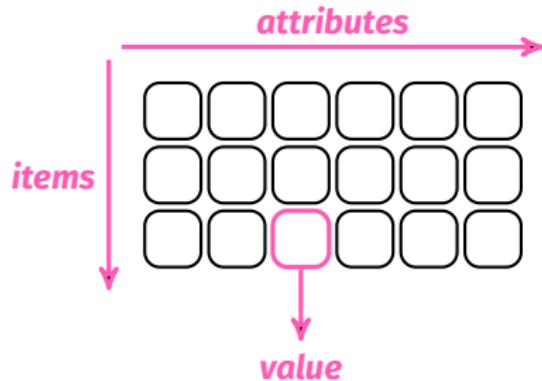
Regiones categóricas

Orientación de ejes

Densidad espacial

What? Tables

Tables es un tipo de **dataset**, compuesto de **objetos** con **valores** en **atributos** específicos.



Los valores encontrados pueden variar entre **categóricos, ordinales o cuantitativos**.

How? Arrange

En el contexto del **how?**, el cómo diseñar *idioms* de visualización, hay una familia de codificación visual llamada **arrange**.

Significados de **arrange**:

- organizar,
- ordenar,
- arreglar y
- disponer.

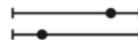
En el *framework*, **arrange** es toda **decisión de diseño** que cubre aspectos en el uso del **espacio** para codificar visualmente.

Ranking de expresividad de canales

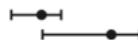
Channels: Expressiveness Types and Effectiveness Ranks

⊕ Magnitude Channels: Ordered Attributes

Position on common scale



Position on unaligned scale



Length (1D size)



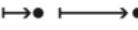
Tilt/angle



Area (2D size)



Depth (3D position)



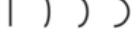
Color luminance



Color saturation



Curvature



Volume (3D size)



⊕ Identity Channels: Categorical Attributes

Spatial region



Color hue



Motion



Shape



Effectiveness ↑ ↓ Least Same ↓

How? Arrange

La **posición espacial** se relaciona con los canales visuales más efectivos en tanto atributos ordenados como categóricos.

El resto de los tipos de canales son efectivos exclusivamente entre magnitud o identidad.

Por lo tanto, *arrange* es el *encoding* visual más importante.

Arrange keys or values

Es necesario notar la diferencia entre **llaves (keys)** y **valores (values)**.

Arrange keys or values

Es necesario notar la diferencia entre **llaves (keys)** y **valores (values)**.

Llave: Atributo independiente que puede ser usado como **índice único** en el *dataset*. Los distintos valores que puede tomar se le llaman **niveles (levels)**. Puede ser de tipo:

Arrange keys or values

Es necesario notar la diferencia entre **llaves (keys)** y **valores (values)**.

Llave: Atributo independiente que puede ser usado como **índice único** en el *dataset*. Los distintos valores que puede tomar se le llaman **niveles (levels)**. Puede ser de tipo: **categórico** u **ordinal**.

Arrange keys or values

Es necesario notar la diferencia entre **llaves (keys)** y **valores (values)**.

Llave: Atributo independiente que puede ser usado como **índice único** en el *dataset*. Los distintos valores que puede tomar se le llaman **niveles (levels)**. Puede ser de tipo: **categórico** u **ordinal**.

Valor: Atributo dependiente, el **valor en una celda** de la tabla. Puede ser de tipo:

Arrange keys or values

Es necesario notar la diferencia entre **llaves (keys)** y **valores (values)**.

Llave: Atributo independiente que puede ser usado como **índice único** en el *dataset*. Los distintos valores que puede tomar se le llaman **niveles (levels)**. Puede ser de tipo: **categórico** u **ordinal**.

Valor: Atributo dependiente, el **valor en una celda** de la tabla. Puede ser de tipo: **categórico**, **ordinal** o **cuantitativo**.

Arrange keys or values

Es necesario notar la diferencia entre **llaves (keys)** y **valores (values)**.

Llave: Atributo independiente que puede ser usado como **índice único** en el *dataset*. Los distintos valores que puede tomar se le llaman **niveles (levels)**. Puede ser de tipo: **categórico** u **ordinal**.

Valor: Atributo dependiente, el **valor en una celda** de la tabla. Puede ser de tipo: **categórico**, **ordinal** o **cuantitativo**.

Mismos atributos en distintos *datasets* pueden tener distintas semánticas de valor/llave.

Arrange keys or values

Hay una directa relación entre las decisiones para representar tablas en la semántica de sus atributos: ¿cuántos son llaves y cuantos son valores?

Arrange keys or values

Hay una directa relación entre las decisiones para representar tablas en la semántica de sus atributos: ¿cuántos son llaves y cuantos son valores?

- Solo dos valores:

Arrange keys or values

Hay una directa relación entre las decisiones para representar tablas en la semántica de sus atributos: ¿cuántos son llaves y cuantos son valores?

- Solo dos valores: **scatterplot**.

Arrange keys or values

Hay una directa relación entre las decisiones para representar tablas en la semántica de sus atributos: ¿cuántos son llaves y cuantos son valores?

- Solo dos valores: **scatterplot**.
- Un valor y una llave:

Arrange keys or values

Hay una directa relación entre las decisiones para representar tablas en la semántica de sus atributos: ¿cuántos son llaves y cuantos son valores?

- Solo dos valores: **scatterplot**.
- Un valor y una llave: **bar charts**.

Arrange keys or values

Hay una directa relación entre las decisiones para representar tablas en la semántica de sus atributos: ¿cuántos son llaves y cuantos son valores?

- Solo dos valores: **scatterplot**.
- Un valor y una llave: **bar charts**.
- Dos llaves y un valor:

Arrange keys or values

Hay una directa relación entre las decisiones para representar tablas en la semántica de sus atributos: ¿cuántos son llaves y cuantos son valores?

- Solo dos valores: **scatterplot**.
- Un valor y una llave: **bar charts**.
- Dos llaves y un valor: **heatmap**.

Arrange keys or values

Hay una directa relación entre las decisiones para representar tablas en la semántica de sus atributos: ¿cuántos son llaves y cuantos son valores?

- Solo dos valores: **scatterplot**.
- Un valor y una llave: **bar charts**.
- Dos llaves y un valor: **heatmap**.
- Cuatro o más valores:

Arrange keys or values

Hay una directa relación entre las decisiones para representar tablas en la semántica de sus atributos: ¿cuántos son llaves y cuantos son valores?

- Solo dos valores: **scatterplot**.
- Un valor y una llave: **bar charts**.
- Dos llaves y un valor: **heatmap**.
- Cuatro o más valores: **scatterplot matrix**.

Decisiones del *framework*

Se proponen cuatro decisiones a considerar dentro de ***arrange***:

1. Expresar valores cuantitativos.
2. Separar, ordenar y alinear regiones categóricas.
3. La orientación de un eje puede ser: rectilínea, paralela o radial.
4. El *layout* espacial puede ser denso o *space-filling*.

Contenidos

Organizar datos tabulares

Expresar valores cuantitativos

Regiones categóricas

Orientación de ejes

Densidad espacial

Expresar valores cuantitativos

En el caso de *encoding* de un único atributo, cada ítem tendrá una marca en alguna **posición a lo largo del eje**.

Expresar valores cuantitativos

En el caso de *encoding* de un único atributo, cada ítem tendrá una marca en alguna **posición a lo largo del eje**.



Expresar valores cuantitativos

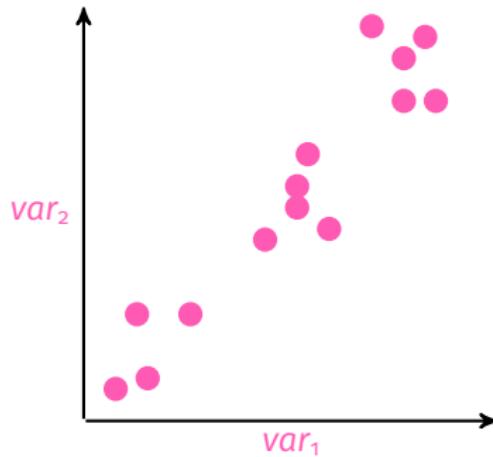
En el caso de *encoding* de un único atributo, cada ítem tendrá una marca en alguna **posición a lo largo del eje**.



Los atributos adicionales pueden ser expresados en la **misma marca** usando canales **no-espaciales**, como el tamaño o el color.

En un caso más complejo un objeto **glyph** puede ser usado, con una estructura interna que emerge a partir del uso de múltiples canales. Volveremos más adelante en el curso a hablar de los **glyphs**.

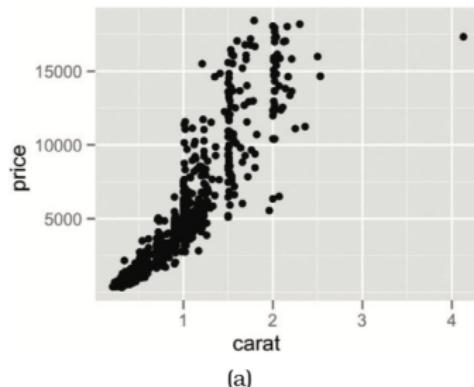
Scatterplot



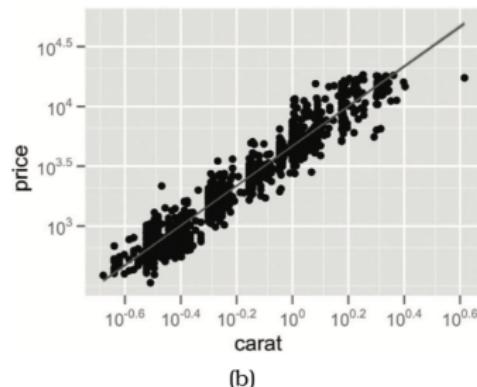
Los **scatterplots** logran expresar, mediante la marca de un punto, dos variables cuantitativas usando las **posiciones espaciales verticales y horizontales**.

Scatterplot

Los scatterplots son efectivos para los tasks de entregar **overviews**, caracterizar **distribuciones** y específicamente para encontrar **outliers** y **valores extremos**.



(a)



(b)

Además, nos permiten, con alta efectividad, juzgar la **correlación** entre dos atributos, ya que este **task** corresponde a percibir si es que los puntos forman una línea en la diagonal.

Scatterplot

Como mencionamos anteriormente, los **scatterplots** pueden ser aumentados usando **colores** para mostrar un atributo adicional. O bien, también se les puede agregar **tamaño** a las marcas para exponer otro atributo: este tipo de *idiom* se conoce generalmente como **bubble chart** o **bubble plot**.

La **escalabilidad** de un scatterplot está limitada por la necesidad de **distinguir** entre los puntos. Por lo tanto, un scatterplot debería ser capaz de aceptar hasta una centena de ítems.

Scatterplot: Resumen en *framework*

Idiom	Scatterplots
What: Data	Table: two quantitative value attributes.
How: Encode	Express values with horizontal and vertical spatial position and point marks.
Why: Task	Find trends, outliers, distribution, correlation; locate clusters.
Scale	Items: hundreds.

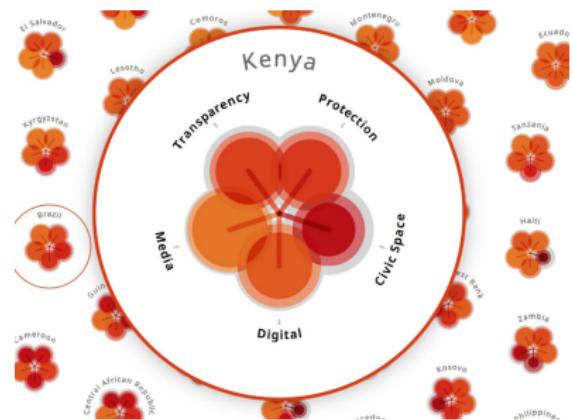
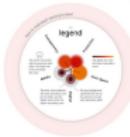
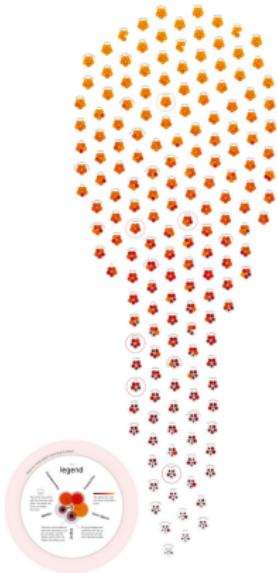
Scatterplot: Posición como composición



What city is the microbrew capital of the US?, pudding.cool

Glyphs

THE EXPRESSION AGENDA REPORT



The Freedom of Expression, Visual Cinnamon

Contenidos

Organizar datos tabulares

Expresar valores cuantitativos

Regiones categóricas

Orientación de ejes

Densidad espacial

Separar, ordenar y alinear: regiones categóricas

El **uso del espacio** para exponer atributos **categóricos** es más complejo que en el caso de los atributos cuantitativos.

Separar, ordenar y alinear: regiones categóricas

El **uso del espacio** para exponer atributos **categóricos** es más complejo que en el caso de los atributos cuantitativos.

Dado que la posición es un canal que expresa magnitudes **ordenadas**, los atributos categóricos **no tienen una semántica de orden**; por lo tanto, el **principio de expresividad** podría ser violado si son expuestos con una posición en el espacio.

Separar, ordenar y alinear: regiones categóricas

Sin embargo, la semántica de los atributos categóricos encaja bien con la idea de **regiones espaciales**.

Estas regiones son **áreas contiguas y limitadas**, que son **distintas** entre sí. Al dibujar todos los ítems con los mismos valores de un atributo categórico en una misma región, se utiliza la **proximidad espacial** para exponer información acerca de su **similaridad**.

Separar, ordenar y alinear: regiones categóricas

Veamos cómo este problema se vuelve más fácil al descomponerlo en tres operaciones:

- **separar**,
- **ordenar** y
- **alinear** las diferentes regiones.

La separación y el orden siempre deben ocurrir, no así con el alineamiento que es opcional.

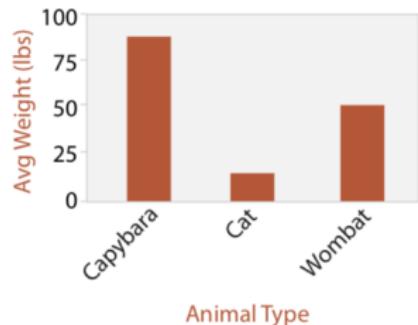
La separación debería ocurrir por un atributo **categórico**, mientras que los otros dos (i.e. ordenar y alinear) debe hacerse por un atributo que tenga inherentemente algún tipo de **orden**.

Bar chart

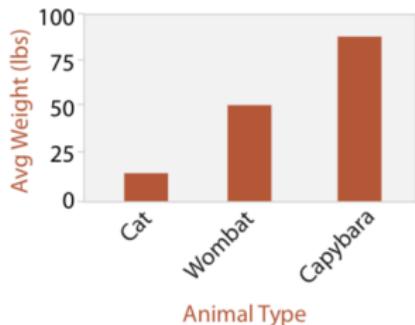
Este conocido *idiom*, llamado diagrama de barras, utilizando una marca de **línea**, con el atributo **cuantitativo** expuesto como una posición espacial.

Además, el otro atributo es **categórico** (el único *key*) utiliza una región **separada** por cada elemento. Las marcas están **alineadas** en un marco común para resaltar las diferencias del tamaño de las líneas.

Bar chart



(a)



(b)

Se utiliza algún tipo de **orden** para las regiones, dependiendo del uso. En el ejemplo, en (a) cada región está ordenada **alfabéticamente** por el nombre de las especies. Por otra parte, en (b) está ordenado por el **tamaño** de las barras. ¿Para que tarea es más conveniente cada orden?

Bar chart: Resumen en framework

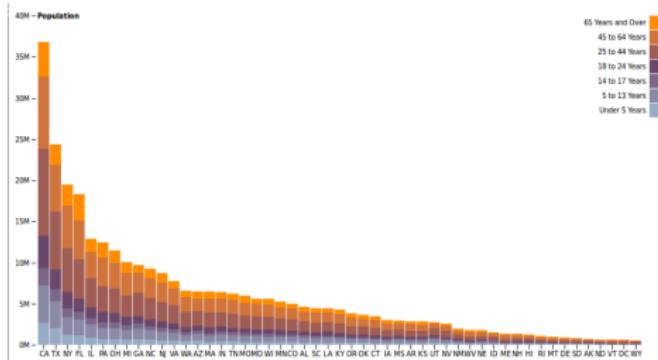
Idiom	Bar Charts
What: Data	Table: one quantitative value attribute, one categorical key attribute.
How: Encode	Line marks, express value attribute with aligned vertical position, separate key attribute with horizontal position.
Why: Task	Lookup and compare values.
Scale	Key attribute: dozens to hundreds of levels.

Stacked bar chart

Los *stacked bar charts* son similares a los *bar charts*, pero usan un **glifo** más complejo para cada barra, donde múltiples sub-barras son apiladas verticalmente. El **largo** del glifo compuesto todavía **codifica un valor** (como un *bar chart*), pero también lo hace cada **subcomponente**.

Los *stacked bar charts* ofrecen información sobre **tablas multidimensionales**, específicamente una tabla bidimensional con **dos llaves**. Los glifos compuestos están ordenados por la **llave primaria**, mientras que la **llave secundaria** sirve para construir la estructura vertical de cada glifo.

Stacked bar chart



Ese *idiom* generalmente también utiliza el **color**, además del tamaño, para cada glifo. De esta forma, cada **subcomponente se colorea según la misma llave** que está determinada por el ordenamiento vertical.

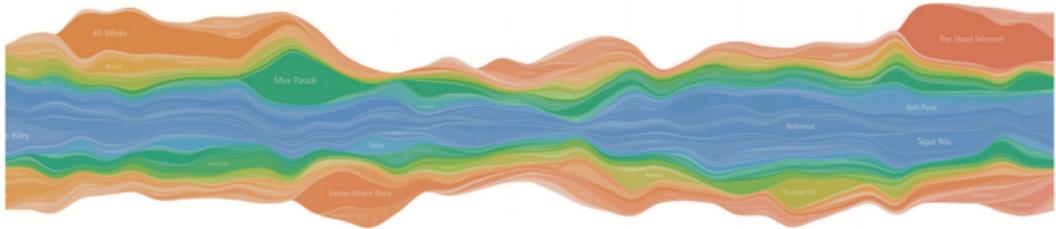
Este color **no agrega ninguna información adicional**, ya que el orden vertical ya muestra a qué llave secundaria pertenece. Sin embargo, **comparar** subcomponentes entre diferentes glifos sería mucho **más difícil**.

Stacked bar chart: Resumen en framework

Idiom	Stacked Bar Charts
What: Data	Multidimensional table: one quantitative value attribute, two categorical key attributes.
How: Encode	Bar glyph with length-coded subcomponents of value attribute for each category of secondary key attribute. Separate bars by category of primary key attribute.
Why: Task	Part-to-whole relationship, lookup values, find trends.
Scale	Key attribute (main axis): dozens to hundreds of levels. Key attribute (stacked glyph axis): several to one dozen

La **escalabilidad** de este *idiom* es similar al de los diagramas de barras, en término del número de llaves primarias, pero sí está **más limitada** con las **llaves secundarias**, que no deberían ser más de una docena.

Streamgraph



Los *streamgraphs* (gráfico de corriente, torrente o flujo) ofrecen una geometría que hace énfasis en la **continuidad** de los datos. Si usáramos un *stacked bar chart*, veríamos glifos verticales que darían énfasis sobre la popularidad en **tiempos específicos**.

Es importante notar que tanto los stacked bar charts como los *streamgraphs* pueden ser orientados tanto horizontalmente como verticalmente.

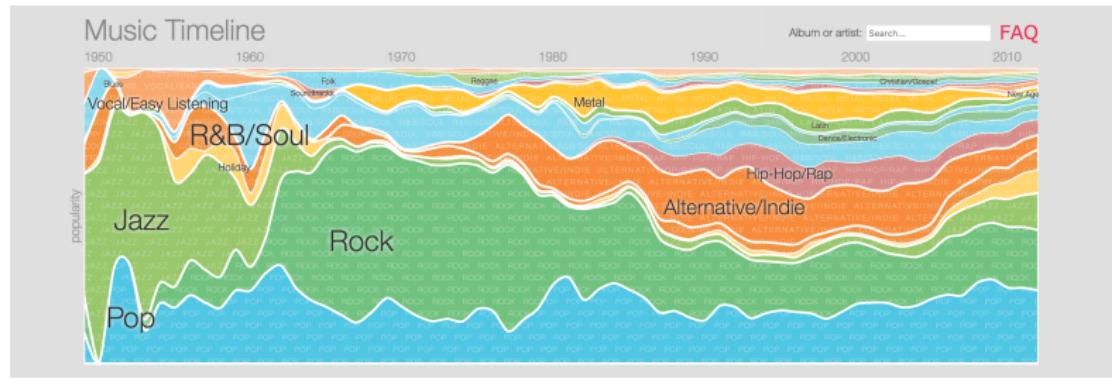
Streamgraph

Los *streamgraphs* suelen usarse para representar una **serie de tiempo**, simplificando el *task* del usuario de rastrear de ítemes individuales a través del tiempo en un flujo continuo de datos.

Este tipo de *idiom* es generalmente fácil de entender, ya que es algo familiar y reduce el esfuerzo cognitivo para interpretar la visualización.

De esta forma, se privilegia la **legibilidad** de cada torrente con un silueta orgánica, dejando de lado el **baseline** que ofrece generalmente el eje horizontal.

Streamgraph

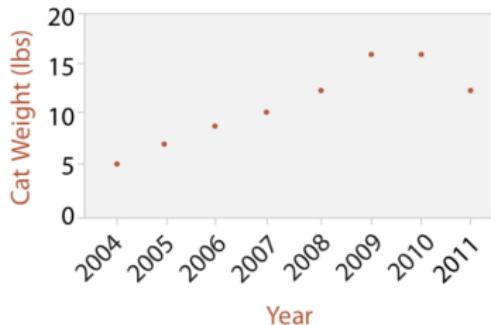


Music Timeline, Google Research

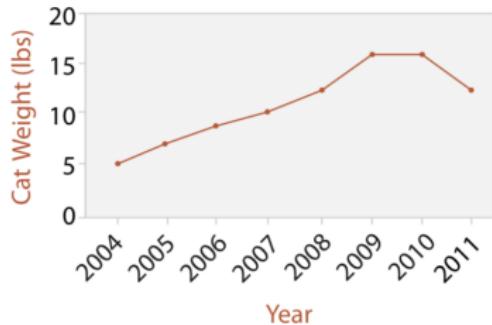
Streamgraph: Resumen en framework

Idiom	Streamgraphs
What: Data	Multidimensional table: one quantitative value attribute (counts), one ordered key attribute (time), one categorical key attribute (artist).
What: Derived	One quantitative attribute (for layer ordering).
How: Encode	Use derived geometry showing artist layers across time, layer height encodes counts.
Scale	Key attributes (time, main axis): hundreds of time points. Key attributes (artists, short axis): dozens to hundreds

Dot chart & line chart



(a)



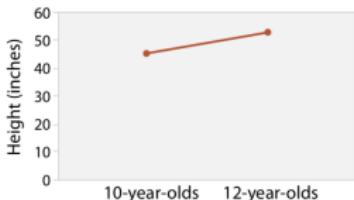
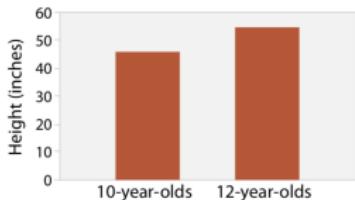
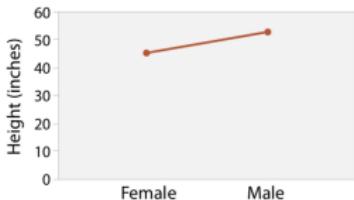
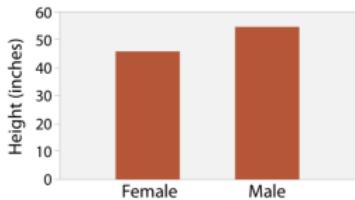
(b)

Un *dot chart* es un *visual encoding* de un atributo **cuantitativo** usando la posición espacial en frente a un atributo **categórico u ordinal**.

Por otra parte, un *line chart* aumenta un *dot chart*, al **conectar** los puntos con líneas. Estas líneas sirven para darle énfasis a una cierta **tendencia** que queramos mostrar desde los datos.

Dot chart & line chart

Es similar a un *bar chart*, en donde el atributo cuantitativo es codificado por **puntos en vez de líneas**. O también puede ser visto como un **scatterplot** en donde uno de los ejes muestra un atributo categórico.



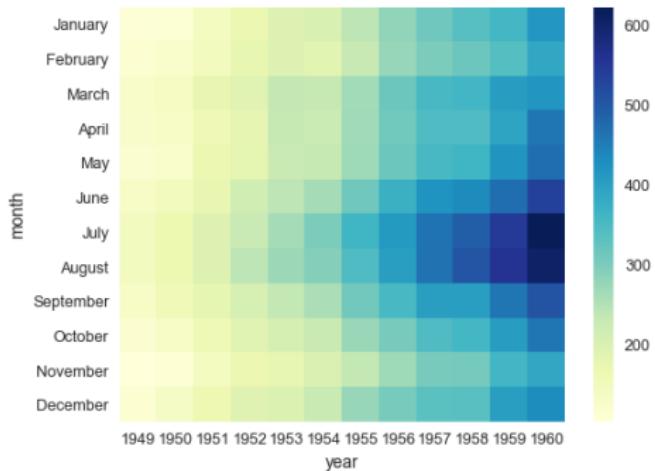
Dot chart & line chart : Resumen en framework

Idiom	Dot Charts
What: Data	Table: one quantitative value attribute, one ordered key attribute.
How: Encode	Express value attribute with aligned vertical position and point marks. Separate/order into horizontal regions by key attribute.

Idiom	Line Charts
What: Data	Table: one quantitative value attribute, one ordered key attribute.
How: Encode	Dot chart with connection marks between dots.
Why	Show trend.
Scale	Key attribute: hundreds of levels.

Matrix Alignment

Los datasets tabulares que están compuestos por **dos llaves** son, generalmente, mostrados en una **matriz bidimensional**, con una llave distribuida en las filas y otra en las columnas. Así, la celda en esta matriz es la región para codificar el valor del ítem.



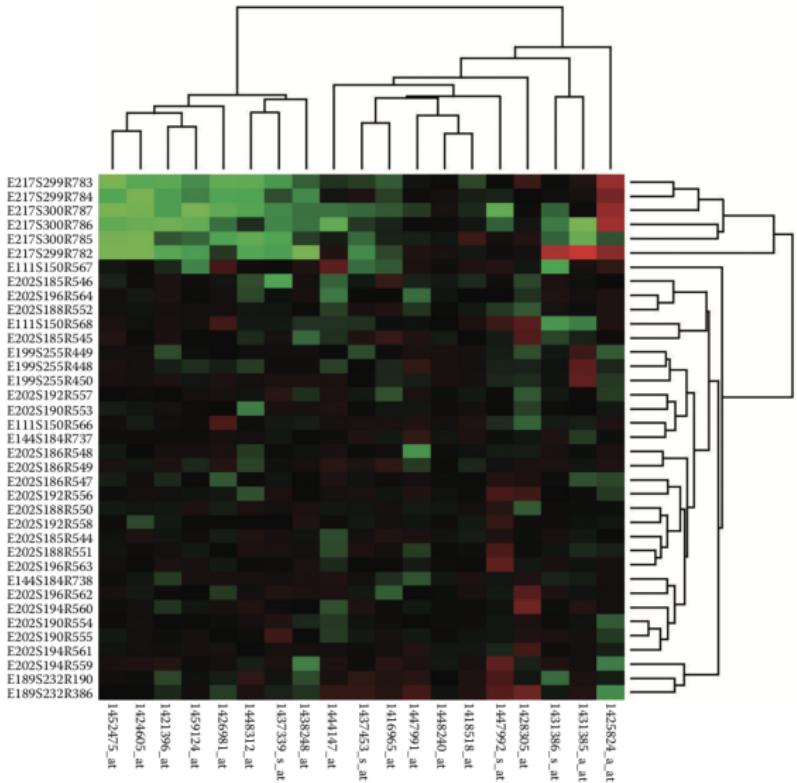
Heatmap

Este *idiom* es uno de los más simples de este tipo: cada celda está ocupada por una marca que codifica un **único valor cuantitativo**.

La ventaja de los **heatmaps** es que usar este tipo de marcas cromáticas para codificar valores es compacta, por lo que permite ofrecer overviews con alta densidad de información. Los principales *tasks* que es posible desarrollar a partir de un heatmap son:

- Encontrar **outliers**
- Encontrar **clusters**
- Tener un **overview**

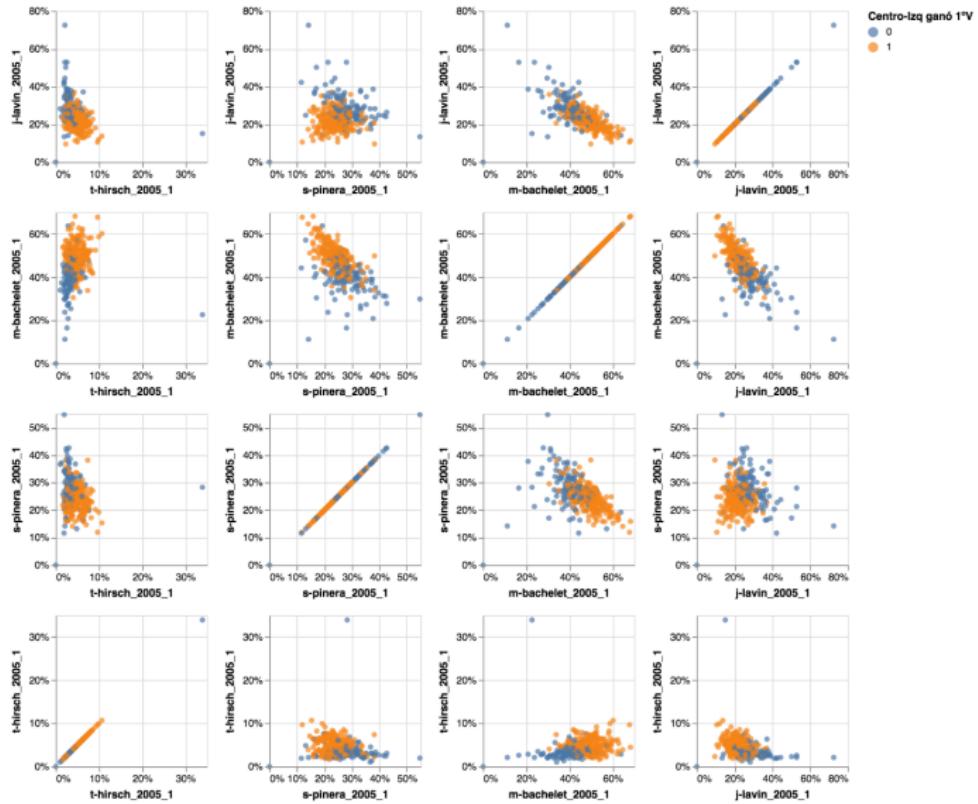
Cluster Heatmap



Heatmap: Resumen en framework

Idiom	Heatmaps
What: Data	Table: two categorical key attributes (genes, conditions), one quantitative value attribute (activity level for gene in condition).
How: Encode	2D matrix alignment of area marks, diverging color-map.
Why: Task	Find clusters, outliers; summarize.
Scale	Items: one million. Categorical attribute levels: hundreds. Quantitative attribute levels: 3–11.
Idiom	Cluster Heatmaps
What: Derived	Two cluster hierarchies for table rows and columns.
How: Encode	Heatmap: 2D matrix alignment, ordered by both cluster hierarchies. Dendrogram: connection line marks for parent-child relationships in tree.

Scatter plot matrix



Scatter plot matrix: Resumen en framework

Idiom	Scatterplot Matrix (SPLOM)
What: Data	Table.
What: Derived	Ordered key attribute: list of original attributes.
How: Encode	Scatterplots in 2D matrix alignment.
Why: Task	Find correlation, trends, outliers.
Scale	Attributes: one dozen. Items: dozens to hundreds.

¿Dos o más llaves?

Así como los datos pueden ser alineados en una lista (1D) o en una matriz (2D), es posible hacer lo mismo entonces con una grilla volumétrica (3D).

Sin embargo, esta alternativa de diseño **no es recomendada**, ya que provoca problemas de percepción como **occlusión** y **distorsión perspectiva**.

Una buena opción es utilizar **subdivisión recursiva**, que analizaremos en un capítulo futuro.

Contenidos

Organizar datos tabulares

Expresar valores cuantitativos

Regiones categóricas

Orientación de ejes

Densidad espacial

Orientación de ejes

Hay tres opciones para esta decisión de diseño:

- *Rectinilear*

- *Parallel*

- *Radial*

Rectilinear layout

En esta disposición, las regiones o ítems se distribuyen en **dos ejes perpendiculares**, con posiciones espaciales horizontales y verticales, que van desde un valor **mínimo** hasta uno **máximo**.

Este tipo de disposición es utilizado **generalmente** en visualizaciones. De hecho, todos los ejemplos vistos hasta ahora en este capítulo utilizan un **rectilinear layout**.

Parallel layout

El *approach* rectilíneo de un *scatterplot* sólo nos permite analizar *datasets* con **dos atributos**. Si bien podemos codificar más atributos con canales no-espaciales, seguiremos limitados para atributos cuantitativos.

Para solucionar este problema, es posible usar un **parallel layout** con el propósito de visualizar atributos multidimensionales. Como el nombre lo sugiere, los ejes están dispuestos de forma paralela entre ellos, generalmente de forma equidistante.

Parallel coordinates

Un *idiom* específico de esta disposición son los **parallel coordinates**.

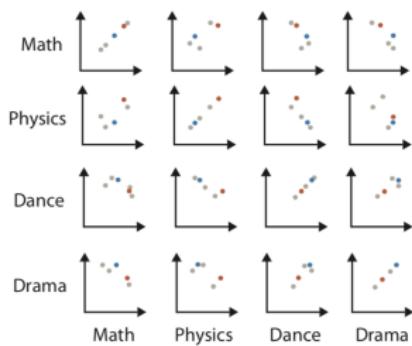
Mientras un dato en un scatterplot es representado con un punto, acá cada ítem se simboliza con una **línea** que hace zigzags a través de los **distintos ejes** (cruzando exactamente una vez) justo en el lugar que representa el valor del ítem. Al igual que los scatterplots, este *idiom* también permite ver rápidamente si existe alguna **correlación** entre los atributos.

Parallel coordinates

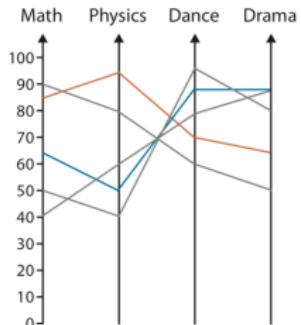
Table

Math	Physics	Dance	Drama
85	95	70	65
90	80	60	50
65	50	90	90
50	40	95	80
40	60	80	90

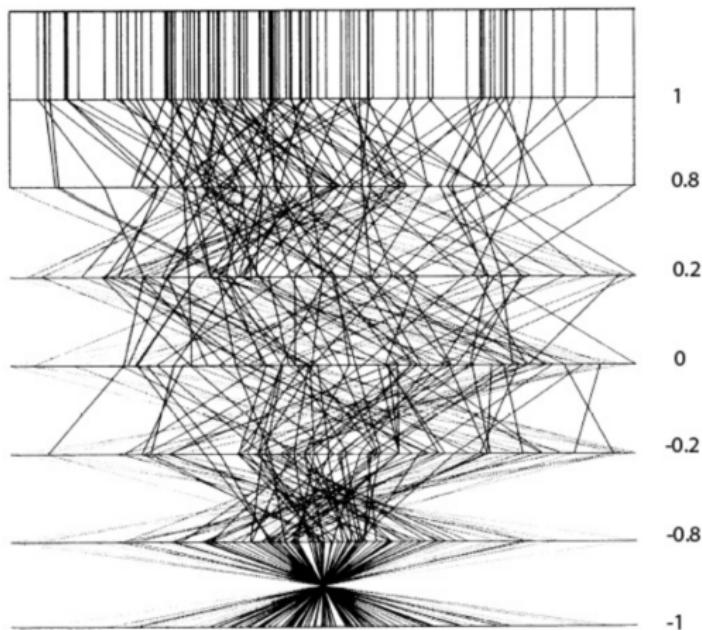
Scatterplot Matrix



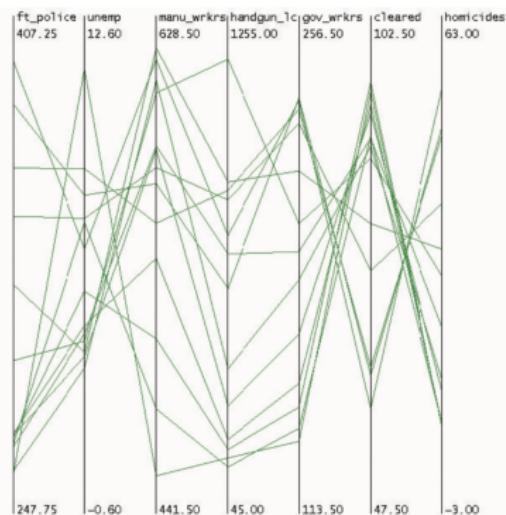
Parallel Coordinates



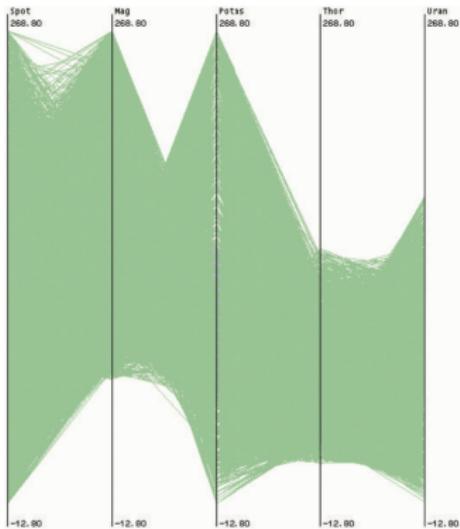
Parallel coordinates



Parallel coordinates



(a)



(b)

Parallel coordinates: Resumen en framework

Idiom	Parallel Coordinates
What: Data	Table: many value attributes.
How: Encode	Parallel layout: horizontal spatial position used to separate axes, vertical spatial position used to express value along each aligned axis with connection line marks as segments between them.
Why: Tasks	Find trends, outliers, extremes, correlation.
Scale	Attributes: dozens along secondary axis. Items: hundreds.

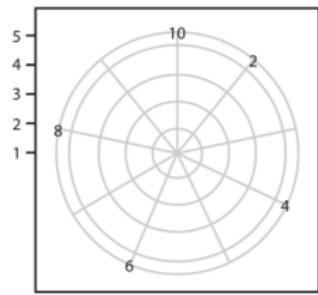
Radial layout

En una disposición espacial radial, los ítems están distribuidos alrededor de un círculo, utilizando como canal el **ángulo** entre ellos.

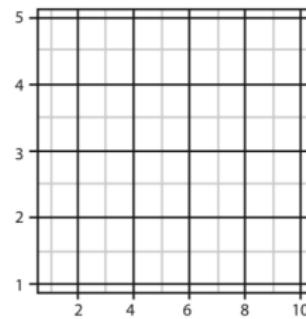
El sistema natural de coordenadas en este tipo de *layout* es el de coordenadas **polares**, en donde una dimensión es medida como el **ángulo** desde la línea de partida y la otra como una **distancia desde el centro**.

Desde un punto de vista matemático, los *layouts* rectilíneos y radiales son equivalentes después de cierta transformación.

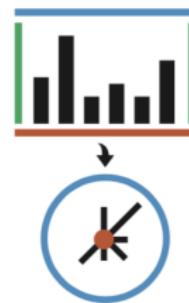
Radial layout



(a)



(b)



(c)

Radial layout

Sin embargo, desde un punto de vista **perceptual**, estas disposiciones son bastante diferentes. El cambio de canal visual tiene dos principales consecuencias según los principios ya estudiados.

Primero, el ángulo es percibido con **menor precisión** que un canal perteneciente al *layout* rectilíneo. Segundo, el **canal angular** es inherentemente **cíclico** ya que el punto de inicio y de término es el mismo. Esto se opone a la **naturaleza lineal** del canal de posición. Por esta misma razón, un *layout* radial es más efectivo para **mostrar periodicidad** de ciertos patrones. Por otra parte, codificar datos no-periódicos con este tipo de canal podría ser **engañoso**.

Contenidos

Organizar datos tabulares

Expresar valores cuantitativos

Regiones categóricas

Orientación de ejes

Densidad espacial

Densidad espacial

Otra decisión de diseño relacionada con *encoding* visual espacial. Esta cubre dos sub-decisiones:

- *Dense or sparse layout*
- *Space-filling layout*

Layout denso

Un *layout denso* usa marcas pequeñas y empaquetadas de forma densa para dar un *overview* de la mayor cantidad posible de ítems.

Un *layout* alcanza un máximo de densidad si utiliza marcas de punto de **1 pixel de tamaño** o líneas de **1 pixel de ancho**. Luego, solo las posiciones planares y los colores se pueden usar para agregar *encoding* visuales. No hay espacio para áreas, curvas, ángulos, etc...

Un sínonimo de *layout denso* es ***pixel-oriented***.

Layout denso



Visualización de líneas de código de programas en empresas.

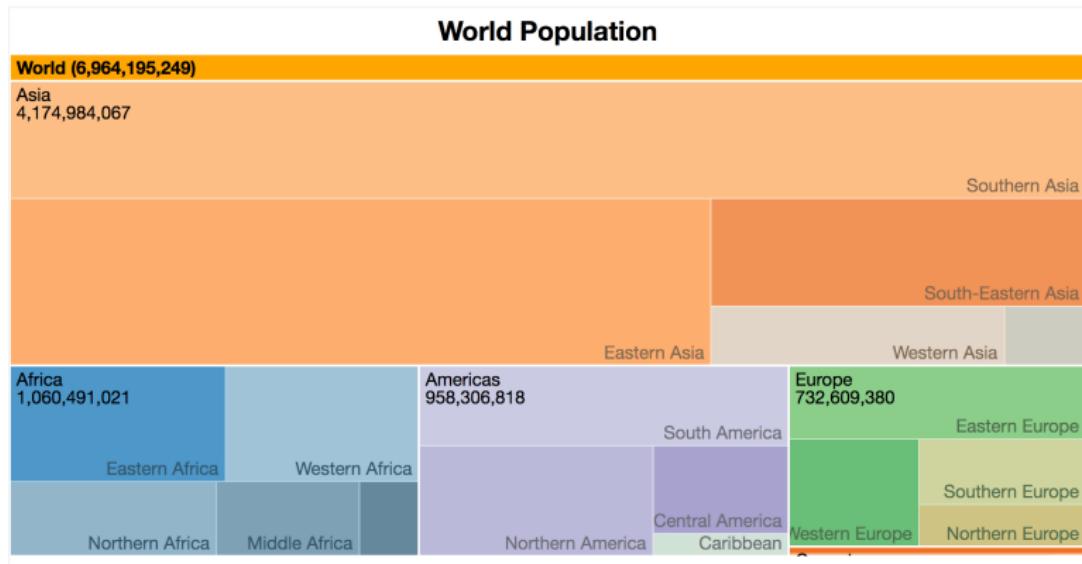
Space-filling layout

Como se esperan, un ***space-filling layout*** es aquel que llena todo el espacio ocupado, **sin espacios en blanco**.

Generalmente utilizan **áreas** y **contención** entre áreas para mostrar elementos y realciones, en vez de marcas de puntos y líneas.

Como ventaja, tiene que al utilizar todo el espacio, aumenta la probabilidad de que el espacio ocupado sea sobresaliente para el usuario y haya suficiente para agregar texto. Como desventaja, no hay espacio en blanco para ser utilizado, el cual en otros *idioms* es aprovechado.

Space-filling layout: Treemap



Zoomable treemap, ganeshv's Block

Referencias

- *Visualization Analysis & Design*, Tamara Munzner
- *The Freedom of Expression*, Visual Cinnamon
- *What city is the microbrew capital of the US?*, pudding.cool
- *Music Timeline*, Google Research
- *Zoomable treemap*, ganeshv's Block

Actividad

- Podrán encontrar archivo `.ipynb` base de actividad en
`syllabus-2018/material actividades`
- Enlace de invitación para repositorio de entrega:
<https://classroom.github.com/a/LuwJACuN>

¡Muchas gracias!

¡Muchas gracias!

¿Alguna pregunta?

Agradecimientos

- Presentación y contenidos base, por **Denis Parra** y **Nebil Kawas**.
- Código base de presentación, por **Cristian Riveros**.