Data abstraction

Visualización de Información
IIC2026

Profesor: Denis Parra

Plan semestral

		Pre: python/pandas	
Semana	**Martes**	**Ayudantía**	**Jueves**
1	**Intro + ¿Qué es visualización?**	**Tunear HTML/SVG/CSS (framewor**	**Javascript I (ayudantia)**
2	**Data abstraction**	feriado virgencita	**Task abstraction**
3	**Análisis y validación**	**Javascript II**	**Marcas y canales**
4	**Percepción**	**d3 introducción**	***Rules of thumb***
5	***Tablas***	**d3 plot estáticos**	***Redes (1)***
6	***Redes (2)***	D3: networks	***Datos Espaciales***
7	feriado fiestas patrias	feriado fiestas patrias	**Color**
8	**Manipulación**	D3: manipulacion	***Manipulación 2***
9	***Presentación Hernán***	D3: interactividad	***Presentación Cristobal***
10	**IR / Mineria Texto**		Visualización de Texto
11	PRESENTACIONES	PRESENTACIONES	PRESENTACIONES
12	Series de Tiempo (Nebil)		*Charla Invitada*
13	**Casos de Estudio I**		feriado dia de los morts
14	**Casos de Estudio II**		VIsualizacion de Algoritmos
15	Invitado de Socvis E. Graells		
16			
	Presentaciones finales		

Definición de Visualización de Información

(o de sistemas de visualización de información)

Según Tamara Munzner:

Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.

Según Tamara Munzner:

Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.

Why?...

Visualization (vis) defined & motivated

Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.

Visualization is suitable when there is a need to augment human capabilities rather than replace people with computational decision-making methods.

- human in the loop needs the details & no trusted automatic solution exists
  - doesn't know exactly what questions to ask in advance
  - exploratory data analysis
    - *speed up* through human-in-the-loop visual data analysis
  - present known results to others
  - stepping stone towards automation
    - before model creation to provide understanding
    - during algorithm creation to refine, debug, set parameters
    - before or during deployment to build trust and monitor

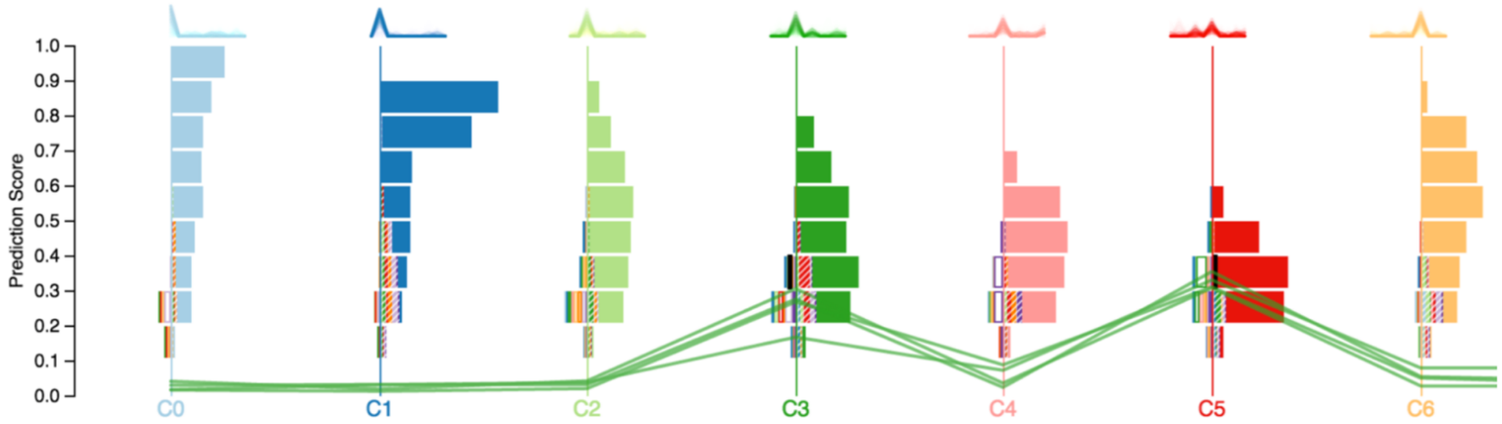
Why use an external representation?

Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.

- external representation: replace cognition with perception

- IEEE VIS 2016:

Ren et al.
Squares: Supporting Interactive Performance Analysis for Multiclass Classifiers



Dataset: **scenario.randomforest.csv** | 10 Classes, 5000 Instances, 5000 Shown | Acc.: 0.87, Prec.: 0.87, Recall: 0.87, TP: 436.0, FP: 64.0, TN: 4436.0, FN: 64.0

★	Image	TRUE	Assigned	Correct	Prediction Score	C0	C1	C2
☆		C3	C5	0	0.3094	0.0426	0.0208	0.0437
☆		C3	C5	0	0.3561	0.0315	0.0339	0.0331
☆		C3	C5	0	0.3121	0.0180	0.0144	0.0218
☆		C3	C5	0	0.3320	0.0188	0.0326	0.0396
☆		C7	C7	1	0.5849	0.0103	0.0226	0.0646
☆		C1	C1	1	0.3576	0.0111	0.3576	0.1180

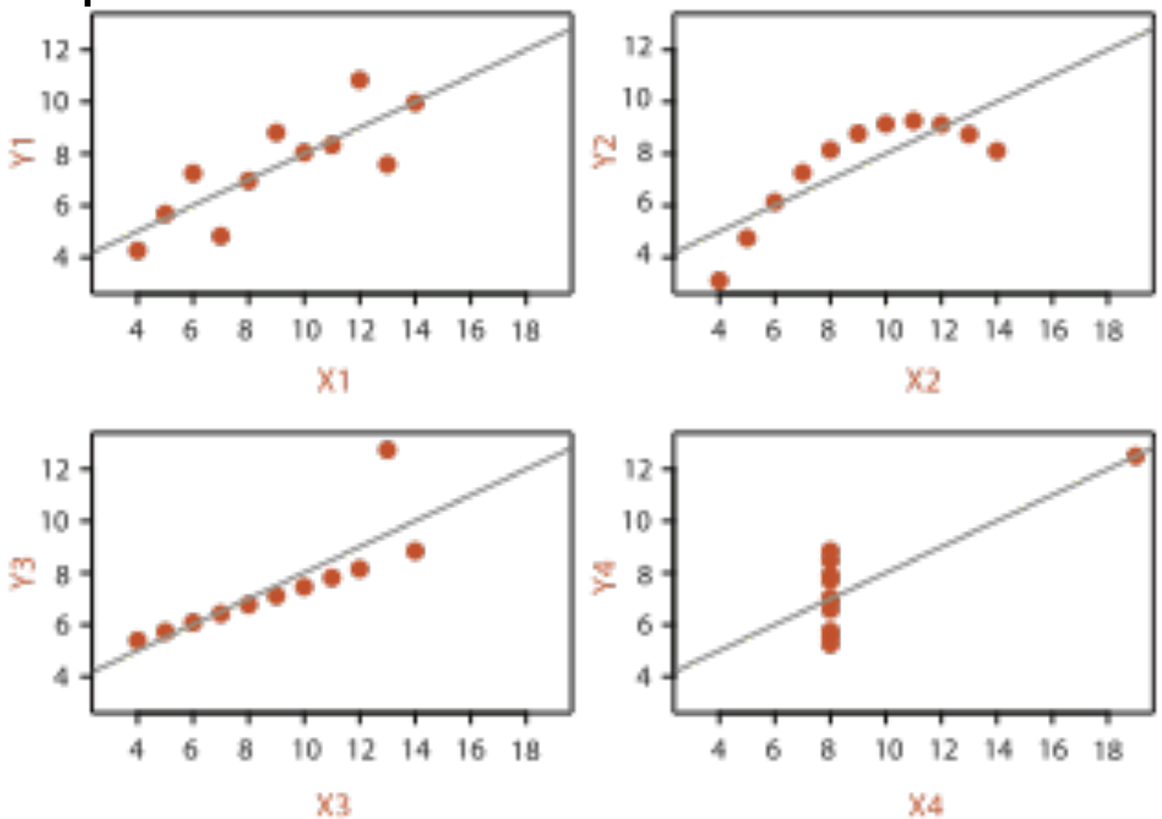
Why represent all the data?

Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.

- summaries lose information, details matter
  - confirm expected and find unexpected patterns
  - assess validity of statistical model

Anscombe's Quartet

Identical statistics	
x mean	9
x variance	10
y mean	7.5
y variance	3.75
x/y correlation	0.816



https://www.youtube.com/watch?v=DbJyPELmhJc

Same Stats, Different Graphs

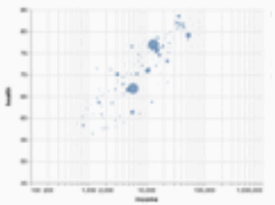
Why focus on tasks and effectiveness?

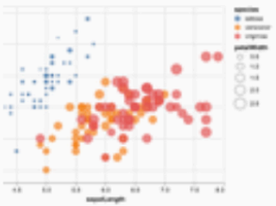
Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively

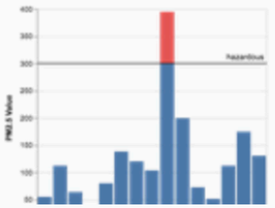
- effectiveness requires match between data/task and representation
  - set of representations is huge
  - many are ineffective mismatch for specific data/task combo
  - increases chance of finding good solutions if you understand full space of possibilities

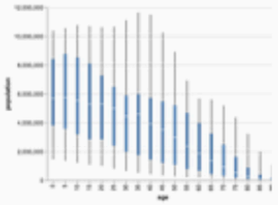

Lines

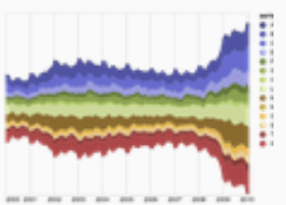

Dot Dash Plot

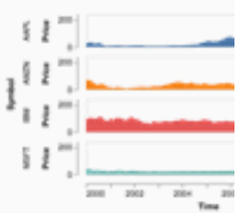

Gapminder Bubble Plot


Multifeature Scatter Plot


Bar Chart with Highlight


Box Plot with Min/Max Whiskers


Streamgraph


Trellis Area

Why focus on tasks and effectiveness?

Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively

- effectiveness requires match between data/task and representation
  - set of representations is huge
  - many are ineffective mismatch for specific data/task combo
  - increases chance of finding good solutions if you understand full space of possibilities
- what counts as effective?
  - novel: enable entirely new kinds of analysis
  - faster: speed up existing workflows
- how to validate effectiveness
  - many methods, must pick appropriate one for your context

What resource limitations are we faced with?

Vis designers must take into account three very different kinds of resource limitations: those of computers, of humans, and of displays.

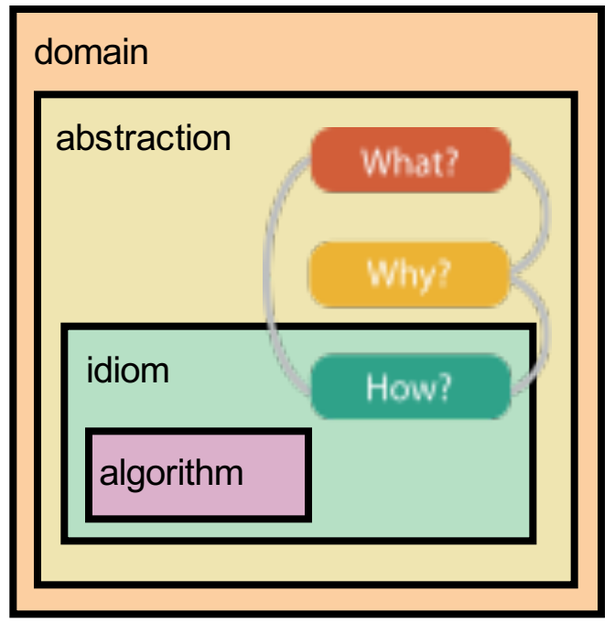
- computational limits
  - processing time
  - system memory
- human limits
  - human attention and memory
- display limits
  - pixels are precious resource, the most constrained resource
  - **information density**: ratio of space used to encode info vs unused whitespace
    - tradeoff between clutter and wasting space, find sweet spot between dense and sparse

Framework de T. Munzner

- Tamara propone este modelo para realizar visualizaciones efectivas:

  - *domain situation*
    - who are the target users?
  - *abstraction*
    - translate from specifics of domain to vocabulary of vis
      - **what** is shown? **data** abstraction
      - **why** is the user looking at it? **task** abstraction
  - *idiom*
    - **how** is it shown?
      - **visual encoding** idiom: how to draw
      - **interaction** idiom: how to manipulate
  - *algorithm*
    - efficient computation

[A Nested Model of Visualization Design and Validation.
Munzner. IEEE TVCG 15(6):921-928, 2009 (Proc. InfoVis 2009).]

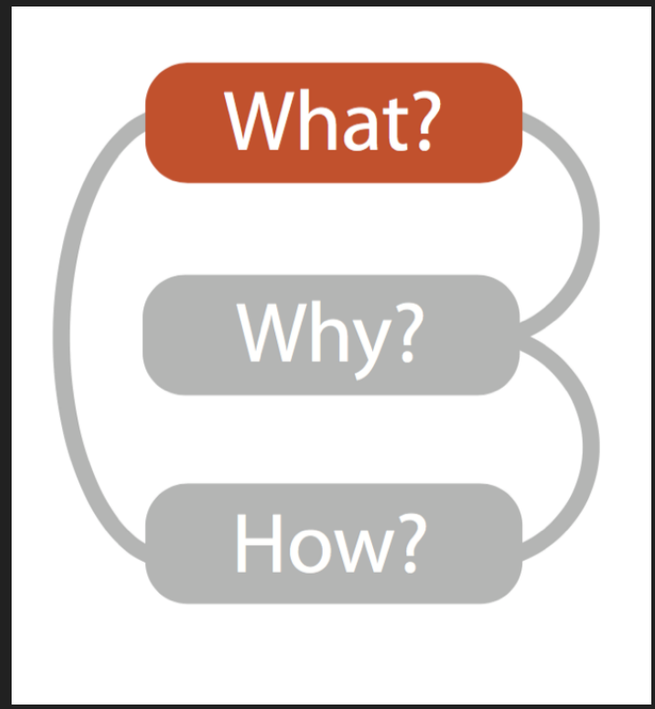


[A Multi-Level Typology of Abstract Visualization Tasks
Brehmer and Munzner. IEEE TVCG 19(12):2376-2385, 2013 (Proc. InfoVis 2013).]

Tres preguntas: qué, por qué, cómo

Partamos con **qué**

- Tipos de datos
- Tipos de *datasets*



Tres preguntas: qué, por qué, cómo

- Muchos aspectos que guían el diseño de una visualización son impulsados por el tipo de datos que tenemos a nuestra disposición.
- Hay que preguntarse, entonces, qué tipo de datos tenemos, qué información podemos obtener directamente, y qué sentido tienen realmente.

Semántica de los datos

14, 2.8, 30, 30, 15, 1001

Semántica de los datos

Santiago, 3, N, Nacimiento

Semántica de los datos

- Para salir de las adivinanzas, es necesario saber dos tipos de información: la semántica y el tipo de dato
- La semántica es su significado en el mundo real (¿qué es? ¿un nombre de una persona, una ciudad, una abreviación de un punto cardinal?)

Tipo de datos

- El tipo de dato es interpretación estructural o matemática del dato (¿es un ítem, un enlace o un atributo?)
- Por ejemplo, si tenemos un número que representa cajas de azúcar, sí hace sentido sumarlas, ya que estamos hablando de una cantidad. Por otra parte, si el número fue el código postal, no tiene sentido sumarlos, ya que no es una cantidad, sino un código.
- A veces, se necesita leer información adicional (conocida como *metadata*) para poder interpretar correctamente un dato.

Dataset and data types

Data and Dataset Types

Tables	Networks & Trees	Fields	Geometry	Clusters, Sets, Lists
Items	Items (nodes)	Grids	Items	Items
Attributes	Links	Positions	Positions	
	Attributes	Attributes		

Data Types

→ Items → Attributes → Links → Positions → Grids

Dataset Availability

→ Static → Dynamic

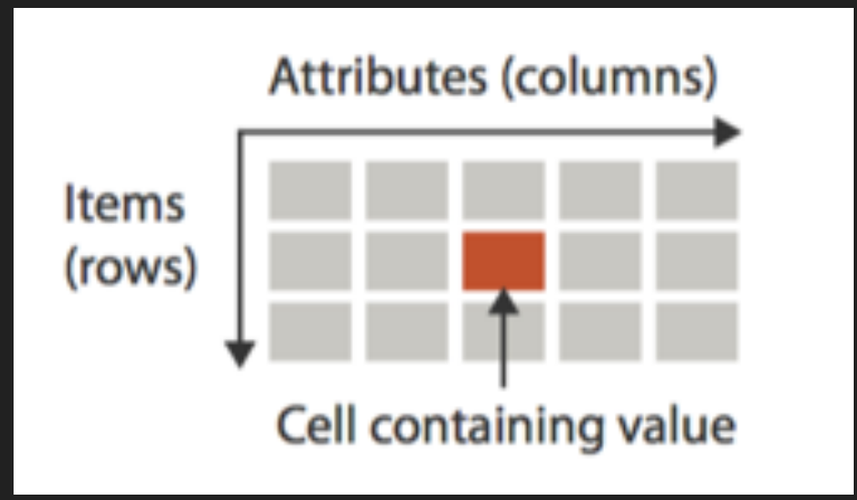
Tipos de dato (*data types*)

Según Munzner (2014), hay cinco tipos básicos de datos:

- Ítems
- Atributos
- Vínculos
- Posiciones
- Grillas

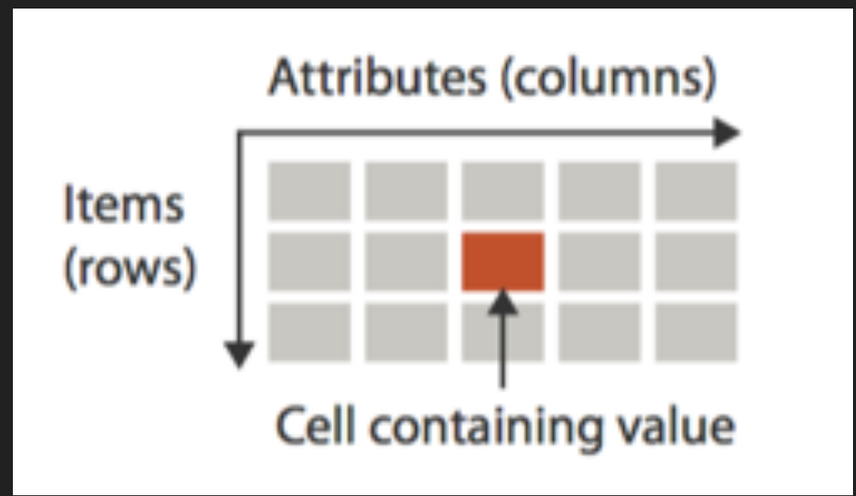
Ítems

- Es una entidad discreta (e.g. fila en una tabla, nodo en un grafo)
- Por ejemplo: personas, ciudades, tiendas de computación



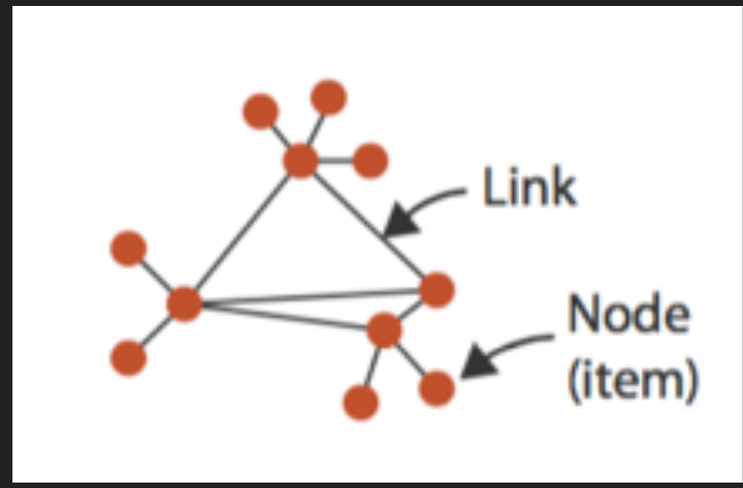
Atributos

- Es una propiedad específica que puede ser medida, observada o registrada
- Por ejemplo: temperatura, salario, precio, número de ventas, etcétera
- También se le conoce como *variable* o *dimensión*



Vínculos

- Es una relación entre los ítems, generalmente en un grafo

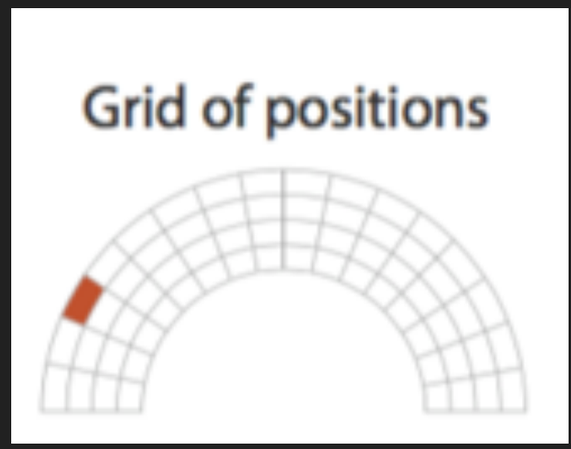


Posiciones

- Es un dato *espacial*, que provee una ubicación en un espacio 2D o 3D
- Por ejemplo: un par latitud-longitud mostrando una ubicación en la Tierra, o también podría ser la ubicación en la región de un escáner médico.

Grilla

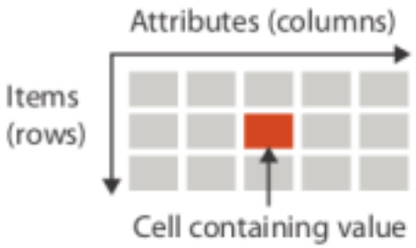
- Es una estrategia para obtener una muestra de datos continuos
- Esto se traduce tanto en términos de relaciones geométricas como topológicas entre distintas celdas



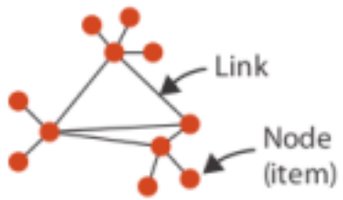
Tipos de datasets

Dataset Types

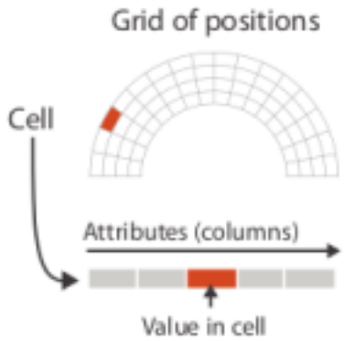
→ Tables



→ Networks

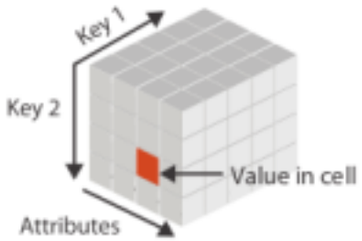


→ Fields (Continuous)



→ Geometry (Spatial)

→ *Multidimensional Table*



→ *Trees*

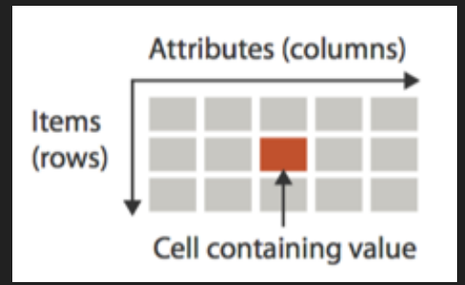
Tipos de *dataset* (*dataset types*)

Según Munzner, hay cuatro tipos básicos de *datasets*:

- Tablas
- Redes (grafos) y árboles
- Campos (*fields*)
- Geometría

Cada uno de ellos, está compuesto por los cinco tipos de dato recién vistos.

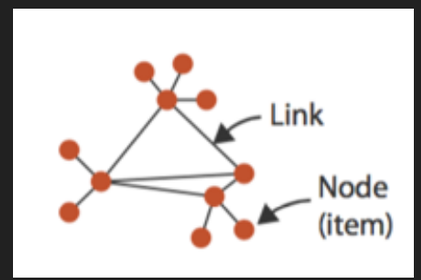
Tablas

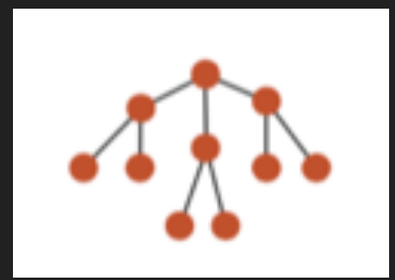


- Es el tipo de *dataset* más común
- Viene en forma de filas y columnas (e.g. *spreadsheet*)
- Los tipos de datos son: ítems y atributos
  - Generalmente, una fila representa un ítem,
  - Y una columna representa un atributo.
- Cada celda de la tabla es un **valor** para la combinación ítem-atributo
- Además, existen las tablas multidimensionales, que tienen múltiples llaves

Redes y árboles

- Este tipo de *dataset* es apropiado para mostrar que existe algún tipo de **relación** entre dos o más ítems
- Un ítem en una red es llamado nodo o vértice
- Una relación entre dos o más nodos se llama enlace o vínculo



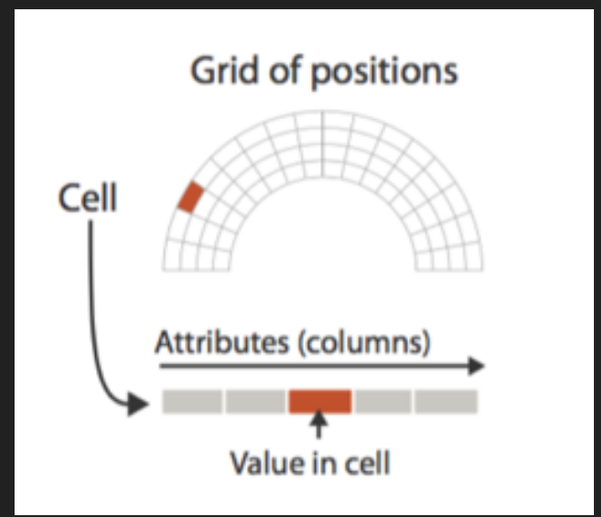


Redes y árboles II

- Por ejemplo: las personas pueden ser representadas como nodos y su relación de amistad entre ellas como vínculos
- Adicionalmente, es posible asociar atributos a cada nodo y enlace.
- Un árbol es un caso específico de un grafo, en donde no existen ciclos (e.g. árbol de jerarquía en una organización)
- Es importante distinguir que nos referimos al concepto abstracto de una red y no a un *layout* en particular (con las posiciones en el espacio) de esta red.

Campos (*fields*)

- Un *dataset* de tipo *field* contiene atributos asociados a celdas
- Luego, cada celda contiene algún tipo de medida o cálculo de un dominio continuo: existen conceptualmente infinitos valores que se podrían medir, ya que puedes siempre medir uno entre dos ya existentes.



Campos (*fields*) II

- Dependemos entonces del sistema de *sampling* utilizado, ya que esto afectará la resolución y los valores obtenidos, cuán frecuente serán las medidas tomadas, y las técnicas de interpolación
- Esto se estudia en disciplinas como procesamiento de señales y estadísticas
- La diferencia entre *scivis* e *infovis* (dos áreas de especialización en visualización) reside en si la posición espacial fue un dato entregado o no

Campos (*fields*) III

- Si hacemos un *sampling* (o muestreo) en intervalos regulares, entonces las celdas forman una grilla uniforme
- Por otra parte, podríamos tener una grilla rectilínea en donde el muestreo no es uniforme, lo que permite almacenar información de forma eficiente, según la complejidad de la región que en donde hacemos el *sampling*
- Una grilla estructurada, además, nos permite tener figuras curvilíneas, en donde la ubicación de cada celda necesita especificarse
- Finalmente, una grilla no estructurada entrega flexibilidad completa, pero la información topológica acerca de cómo se conectan las celdas también debe ser explícita —además de la ubicación.

Geometría

- Habla sobre la forma de ítems con posiciones explícitas
- Los ítems pueden ser puntos, curvas, superficies o volúmenes
- Los *datasets* geométricos son intrínsecamente espaciales
- Este tipo de *dataset* puede que no tenga atributos, a diferencia del resto
    - Aquí es interesante saber cómo codificar información
- También es necesario saber con qué nivel de detalle se generan las formas (*shapes*) desde datos geográficos crudos
    - Por ejemplo, la frontera de un bosque, o de una ciudad, o también la curva de una carretera

Otros tipos de *dataset I*

- Existen múltiples formas de agrupar ítems, además de una tabla
  - Un conjunto (*set*) es grupo sin orden de ítems
  - Una lista (*list*, *array*) es un grupo ordenado de ítems
  - Un clúster (*cluster*) es un grupo basado en la similaridad de un atributo específico

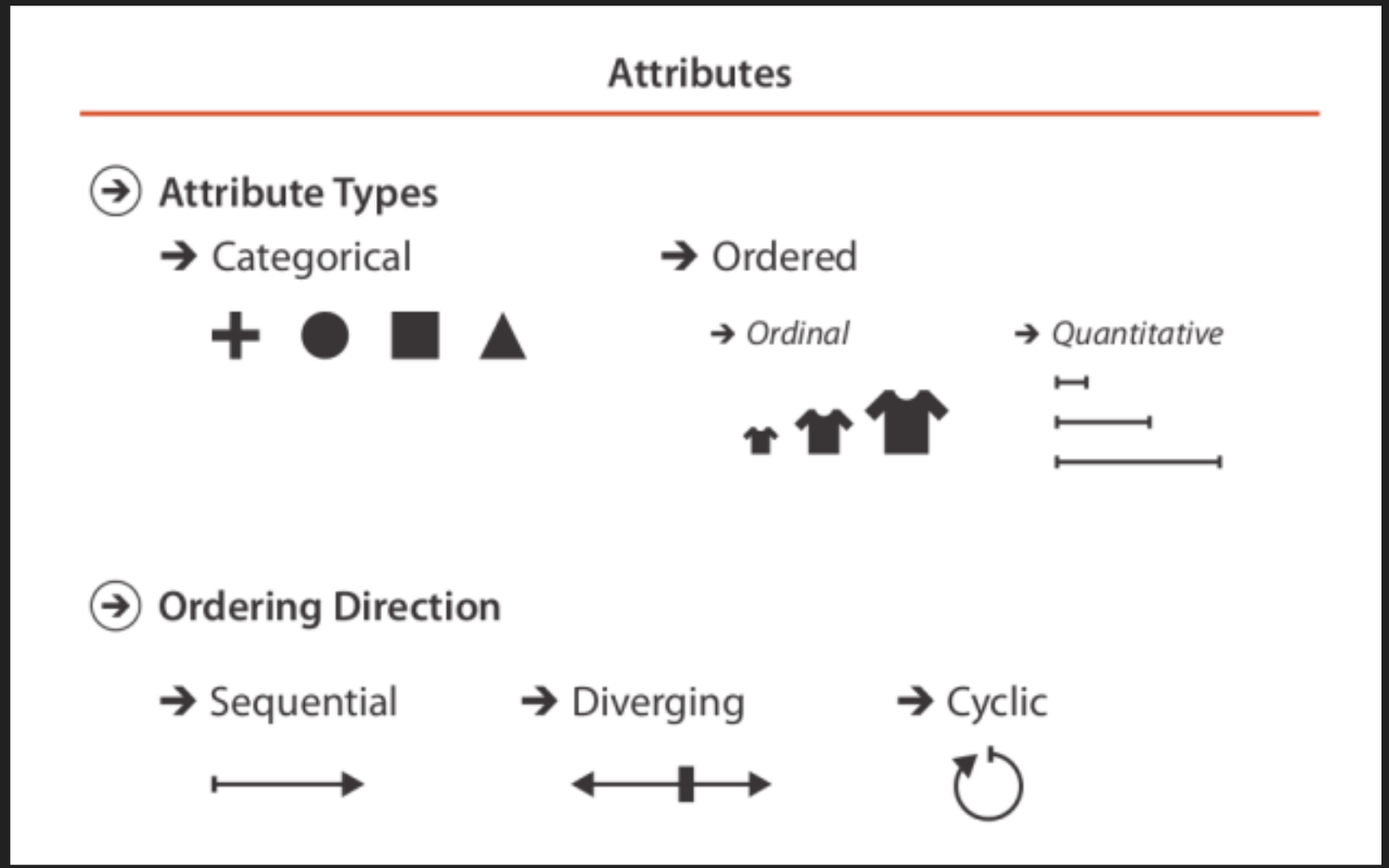
Otros tipos de *dataset II*

- También se pueden construir estructuras a partir de un grafo
  - Por ejemplo, se pueden mostrar **caminos**, que son listas de vínculos que conectan nodos
  - O podríamos tener también un ***compound network***, que es una red que tiene asociado un árbol: todos los nodos de la red son las hojas del árbol, mientras que los nodos interiores del árbol proveen cierta estructura jerárquica para estos nodos de la red.
- Además, es posible crear estructuras híbridas y más complejas que intentan modelar aplicaciones de la vida real: esto es sólo un punto inicial del análisis de *data abstraction*

Disponibilidad del *dataset*

- Existen dos categorías: *datasets* **estáticos** y *datasets* **dinámicos**
  - Estático (*offline*) es cuando el *dataset* está disponible *all at once* (i.e. todo en un instante)
  - Dinámico (*online*) es cuando nueva información llega a través del tiempo (*streaming data*)
- Cuando el *dataset* es dinámico, nuevos datos pueden ser agregados, otros eliminados o también actualizados
- Esto agrega complejidad en varios aspectos al proceso de visualización comparado a un dataset estático

Tipos de atributos

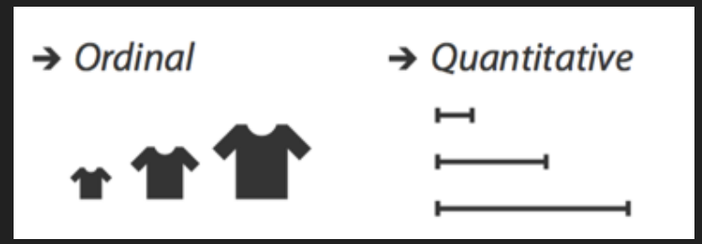


Tipos de atributos: categóricos

- La primera distinción que haremos entre los datos son los de tipo categóricos (o también conocidos como nominales)
- No tienen un orden implícito, pero generalmente sí existe una jerarquía
- Podrían, eso sí, ser ordenados de forma arbitraria por datos externos
- Ejemplo: nombres de frutas

Tipos de atributos: ordenados

- Los datos que no son categóricos, se conocen como datos ordenados.
- Esto puede ser subdividido en: datos ordinales y datos cuantitativos.
- En los datos ordinales, no existe una aritmética bien definida entre sus componentes, pero sí es posible ofrecer un orden (e.g. tallas de poleras)
- Por otra parte, en los cuantitativos, existe una magnitud que sí permite una comparación aritmética. Ejemplos: altura, peso, temperatura, etcétera.



Tipos de atributos: secuencial, divergente o cíclico

- Entre los datos ordenados, podemos distinguir los datos secuenciales, en donde existe un rango homogéneo desde un valor mínimo hasta uno máximo (ejemplo: altura de montañas, que va desde el nivel del mar hasta el Everest)
- Por otra parte, también podemos hablar de datos divergentes, que puede ser descompuesto en dos secuencias que van en direcciones opuestas, que se encuentran en un punto en común: el cero (ejemplo: un *dataset* de elevación, en donde los valores van hacia arriba para las montañas y hacia abajo para los valles submarinos, siendo el nivel del mar, el valor cero)
- Por último, podrían ser cíclicos, en donde los valores *wrap around* hacia el punto inicial, en vez de crecer indefinidamente.

Atributos jerárquicos

- Puede existir una estructura jerárquica entre uno o múltiples atributos.
- Por ejemplo, los precios de acciones recolectados a lo largo de una década es un ejemplo de un *time-series dataset*, en donde uno de los atributos es el tiempo. Este atributo puede ser agregado de forma jerárquica.
- Muchos tipos de datos tienen esta propiedad: por ejemplo, el atributo geográfico de un código postal podría ser agregado a nivel de ciudades, como de regiones, o incluso países.

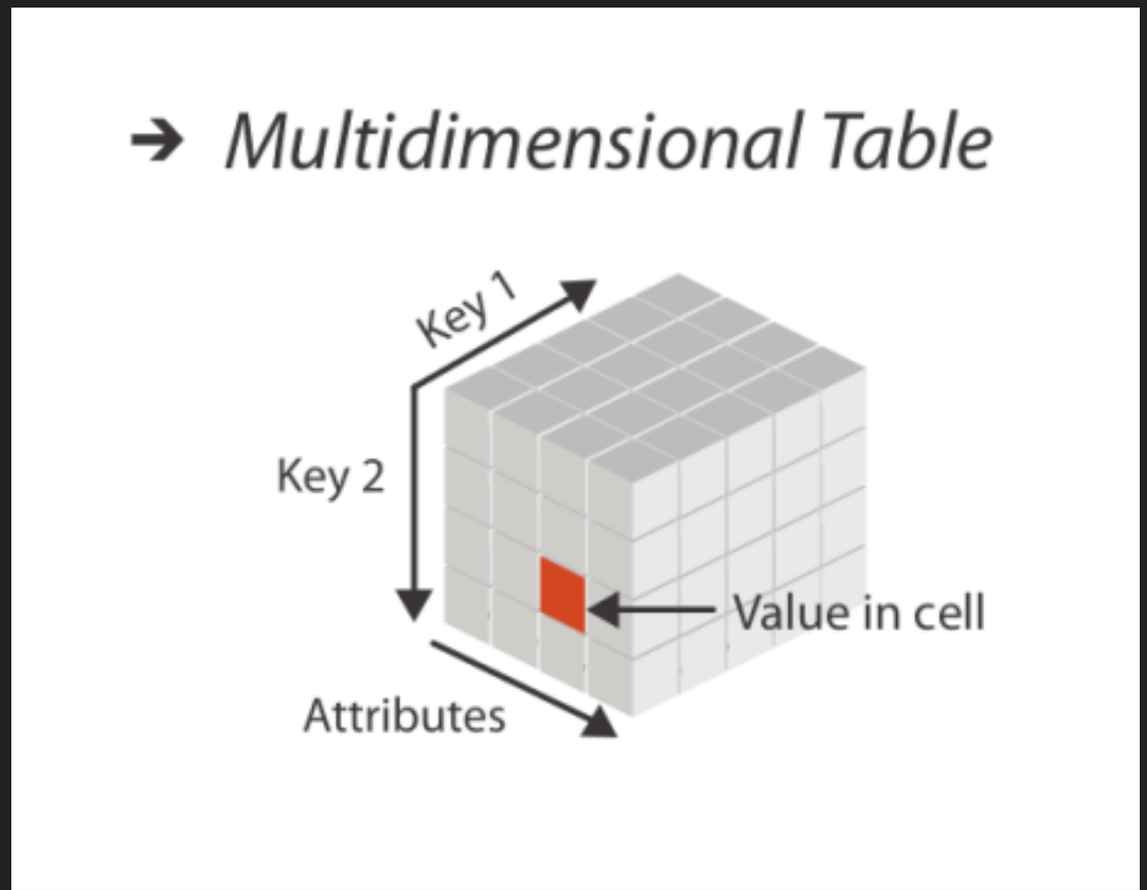
Semántica

- Saber el tipo de dato de un atributo no nos habla de su semántica, ya que son preguntas independientes: uno no impone el significado del otro.
- *Key versus value*: una llave se considera como un atributo independiente, en donde esta distinción es importante en un dataset tabular. Por otra parte, el valor vendría siendo el valor dependiente de la llave.

Semántica

- Saber el tipo de dato de un atributo no nos habla de su semántica, ya que son preguntas independientes: uno no impone el significado del otro.
- *Key versus value*: una llave se considera como un atributo independiente, en donde esta distinción es importante en un dataset tabular. Por otra parte, el valor vendría siendo el valor dependiente de la llave.

Semántica


➔ Multidimensional Table
Key 1
Key 2
Value in cell
Attributes

Semántica temporal

- Igualmente, es importante distinguir una semántica temporal en los datos, que es cualquier tipo de información que se relacione con el tiempo
- No es sencillo manejar un *dataset* con una semántica temporal, dada la riqueza jerárquica que tiene el tiempo, tanto como la posible periodicidad
- Además, también existen algunos problemas con las escalas, ya que no calzan perfectamente (e.g. semanas en un mes)
- Puede ser considerado como un atributo cuantitativo (ya que es posible hacer aritmética con el tiempo), pero si la duración no es de interés, entonces podemos tratarlo como un atributo ordenado