



PONTIFICIA  
UNIVERSIDAD  
CATÓLICA  
DE CHILE

# *Manipular Vista*

Visualización de Información  
IIC2026

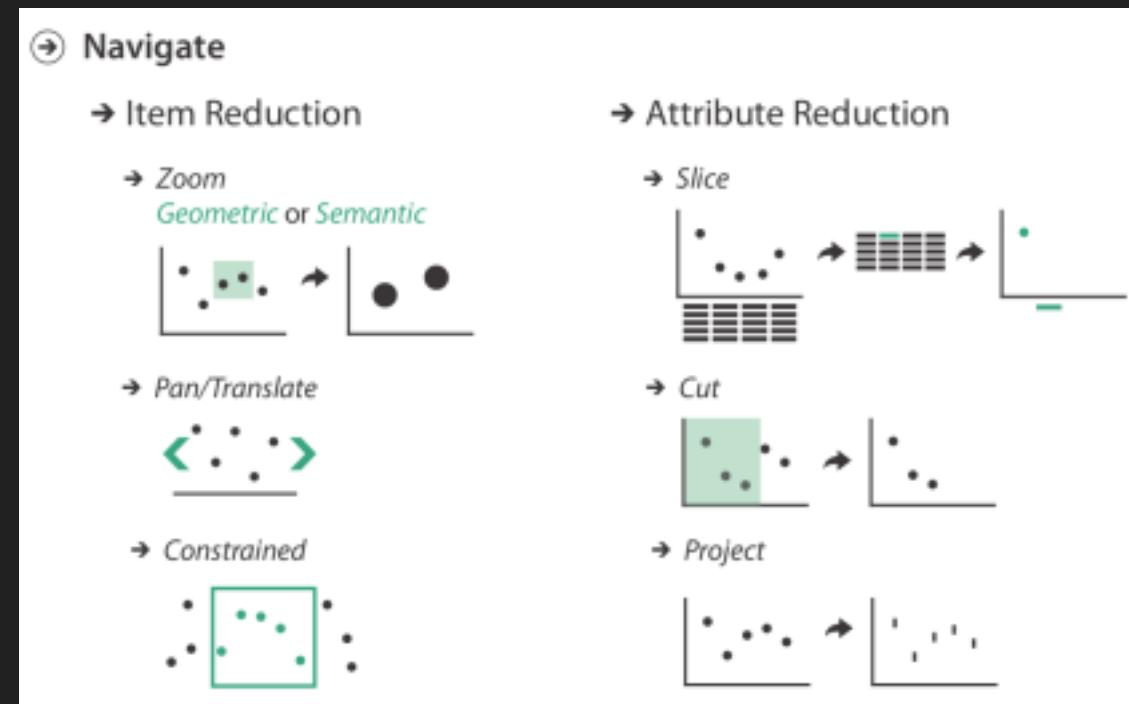
Profesor: Denis Parra

# Planificación semestral

		Pre: python/pandas	
Semana	Martes	Ayudantía	Jueves
1	Intro + ¿Qué es visualización?	Tunear HTML/SVG/CSS (framework)	Javascript I (ayudantía)
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 / Minería 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		<a href="#">Visualizacion de Algoritmos</a>
15	Invitado de Socvis E. Graells		
16			
		Presentaciones finales	

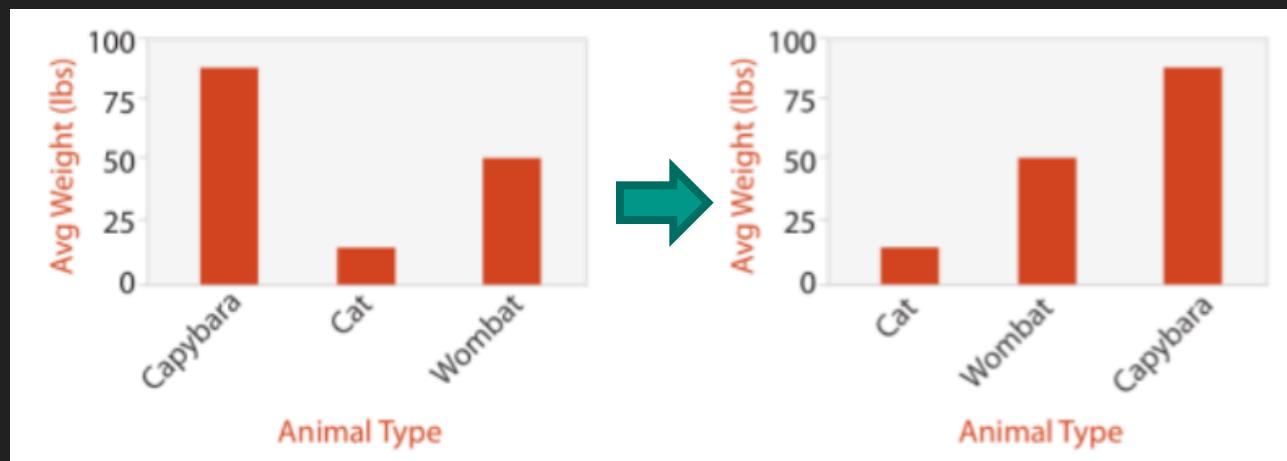
# Resumen

- En esta clase, veremos las decisiones de diseño que podemos tomar al momento de **manipular una vista**: cambiar cualquier elemento que estemos mostrando, seleccionar ítems o atributos dentro de ellos, y navegar a un punto de vista diferente.



# Resumen

- Todo aspecto de un *visual encoding* puede ser modificado, incluyendo el ordenamiento, elecciones relacionadas la disposición espacial, o el uso de canales visuales como el color.
- Los cambios también pueden estar vinculados a qué elementos se quieren filtrar, el nivel de detalle de alguna agregación, o cómo los datos se partitionan en múltiples vistas.

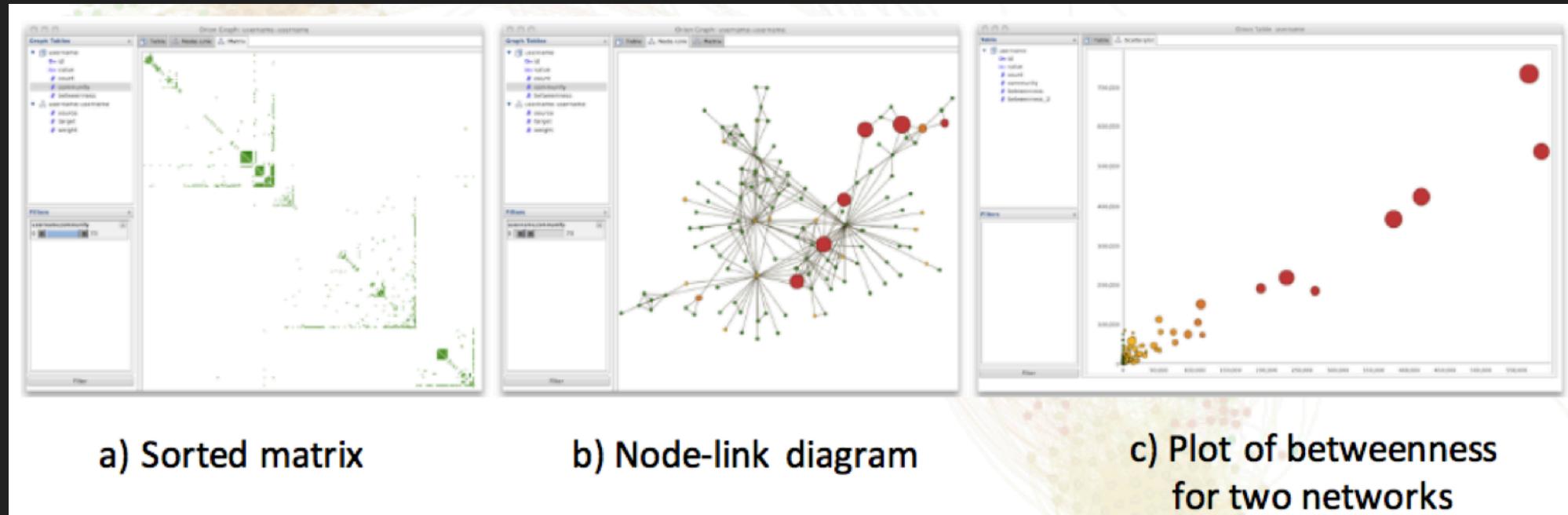


# ¿Por qué cambiar?

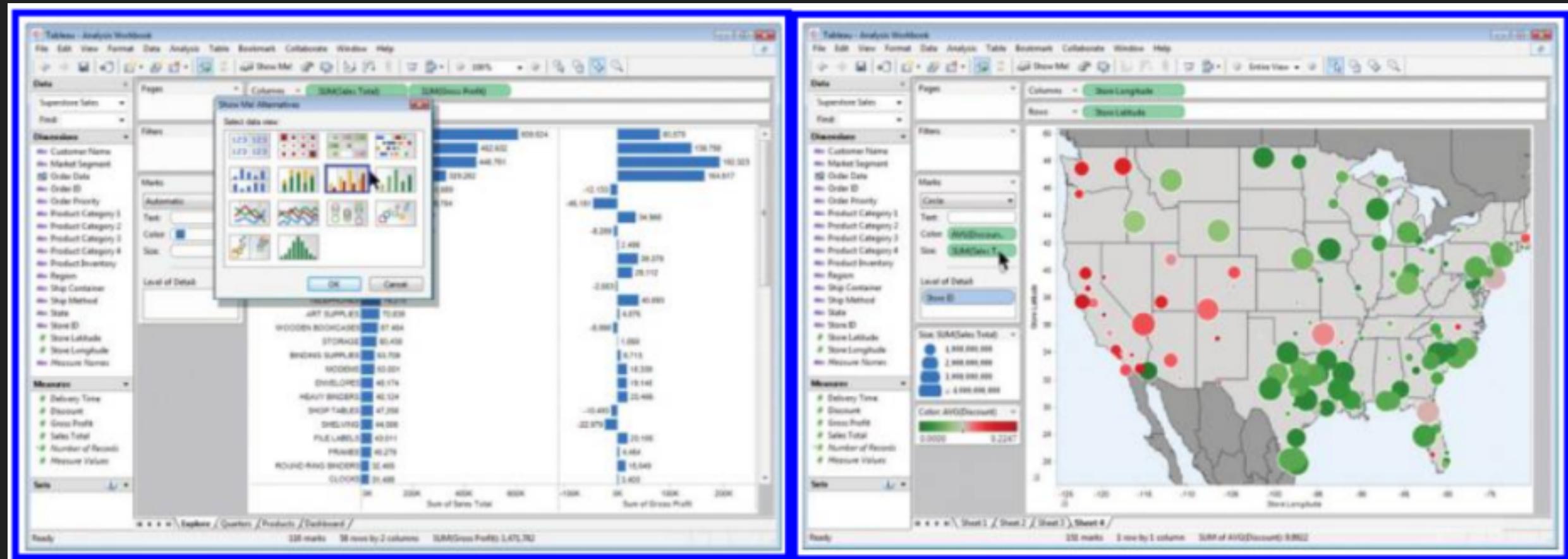
- Generalmente, los *datasets* con los que trabajaremos son suficientemente grandes y complejos, por lo que mostrar todo en una misma vista estática podría conducir a **visual clutter**.
- Para manejar la complejidad del *dataset*, existen varias opciones, siendo una de ellas **cambiar la vista de los datos a través del tiempo**. El resto de las opciones serán vistas en clases posteriores.
- El **aspecto más importante** entregado en una visualización **por computador** (con respecto a una **impresa**) es la posibilidad de **interacción**: una vista que cambia en el tiempo puede responder dinámicamente al *input* del usuario.

# Cambio en el tiempo

- Las posibilidades de **cómo** cambiar una vista están basadas en las mismas decisiones de cómo construimos un *idiom*: cambiar la codificación, cambiar la disposición, cambiar el ordenamiento, cambiar el punto de vista, cambiar qué atributos estamos filtrando, cambiar el nivel de agregación...

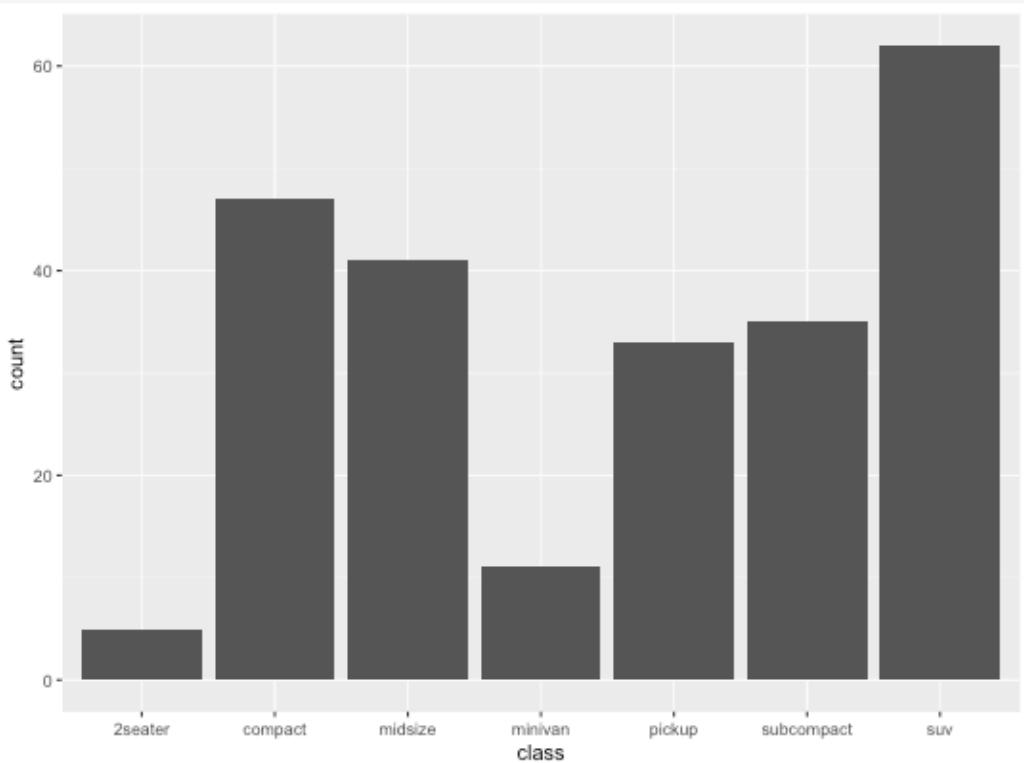


# Ejemplo: distintos *idioms* con Tableau

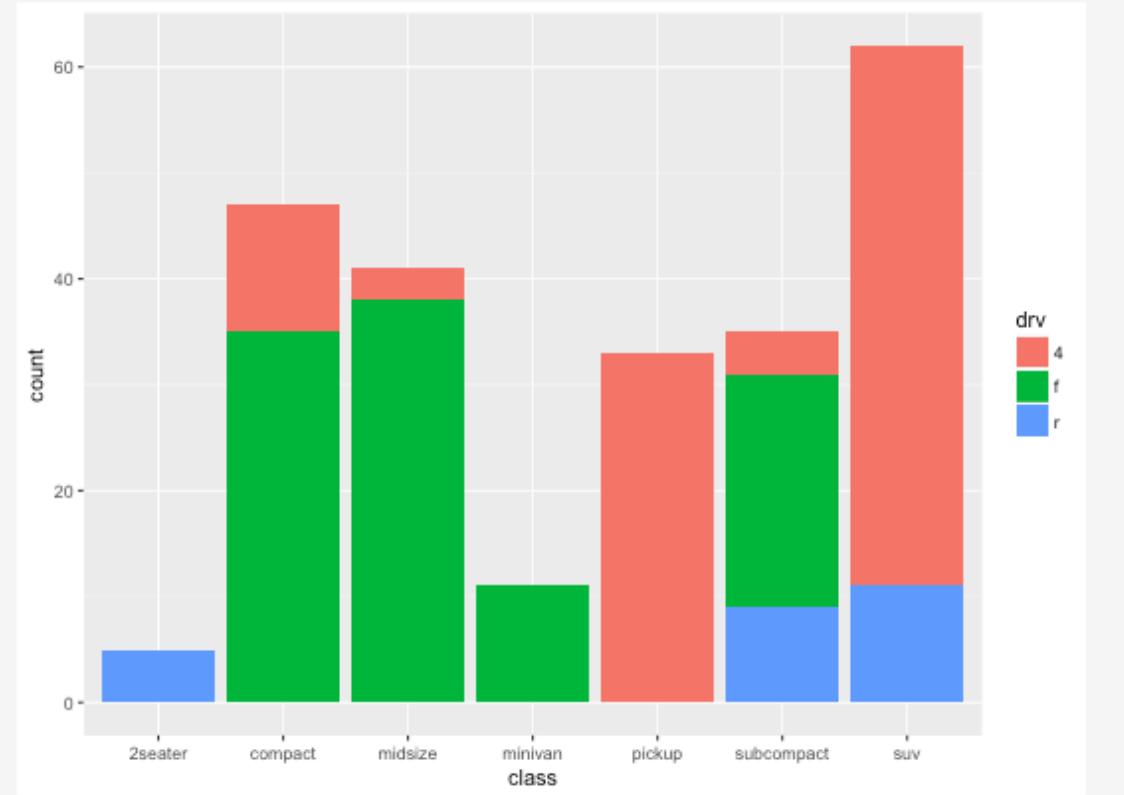


# ggplot2 en R

```
# geom_bar is designed to make it easy to create bar charts that show  
# counts (or sums of weights)  
g <- ggplot(mpg, aes(class))  
# Number of cars in each class:  
g + geom_bar()
```

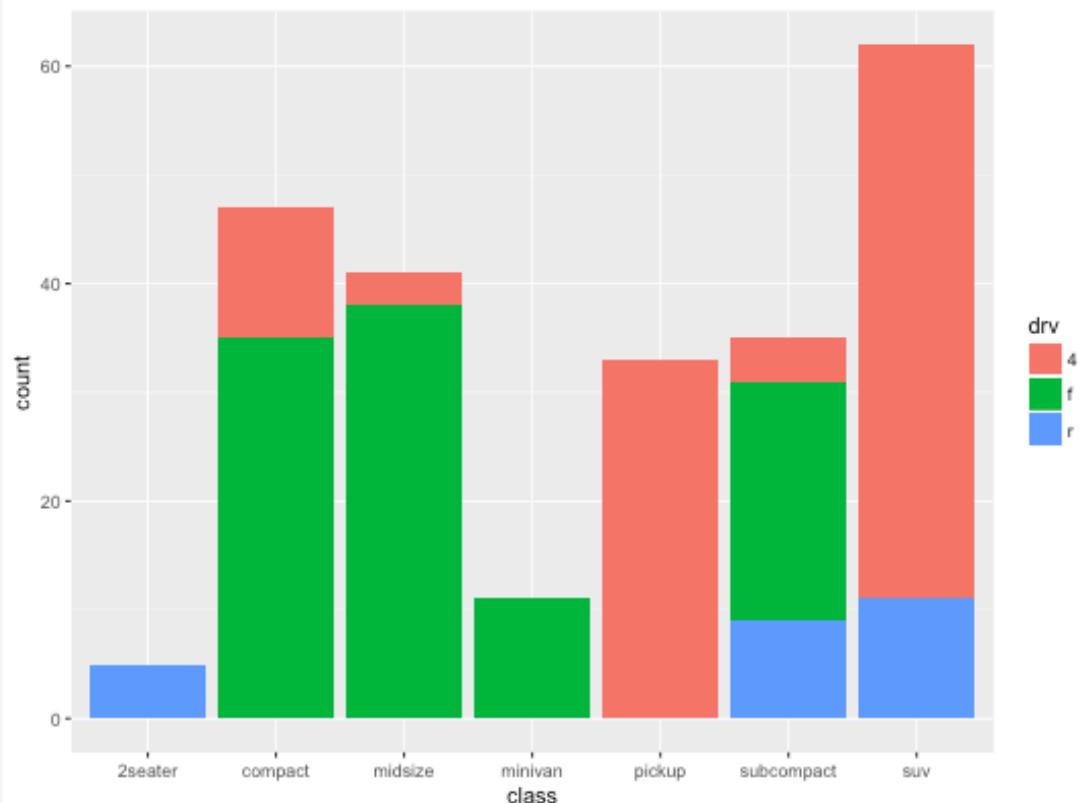


```
# Bar charts are automatically stacked when multiple bars are placed  
# at the same location. The order of the fill is designed to match  
# the legend  
g + geom_bar(aes(fill = drv))
```

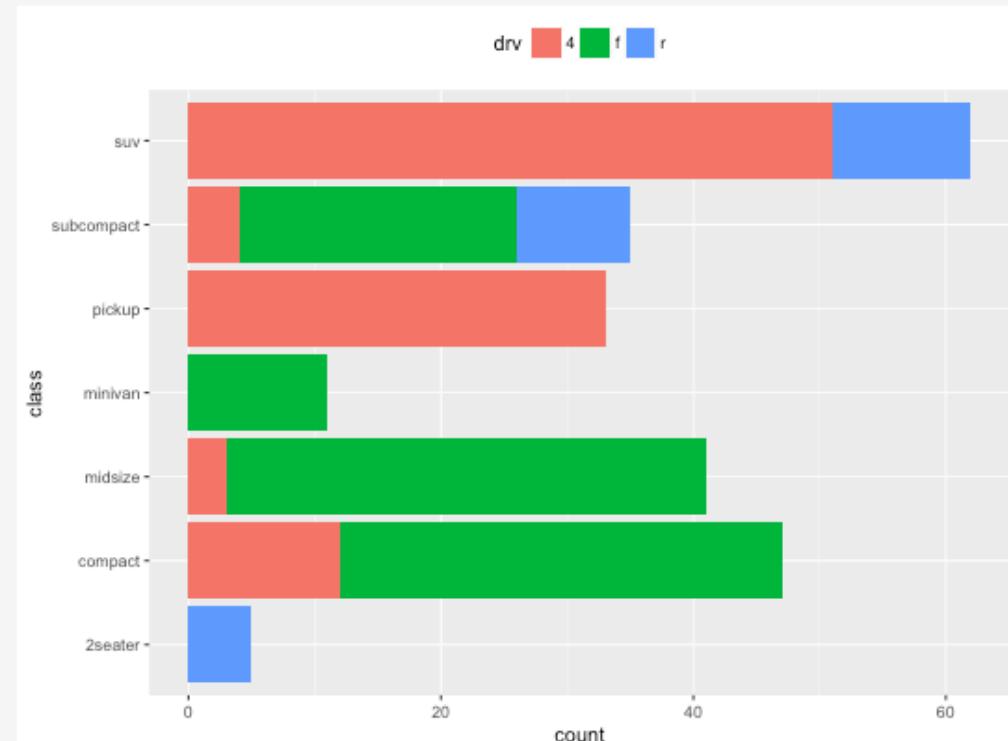


# ggplot2 en R

```
# Bar charts are automatically stacked when multiple bars are placed  
# at the same location. The order of the fill is designed to match  
# the legend  
g + geom_bar(aes(fill = drv))
```

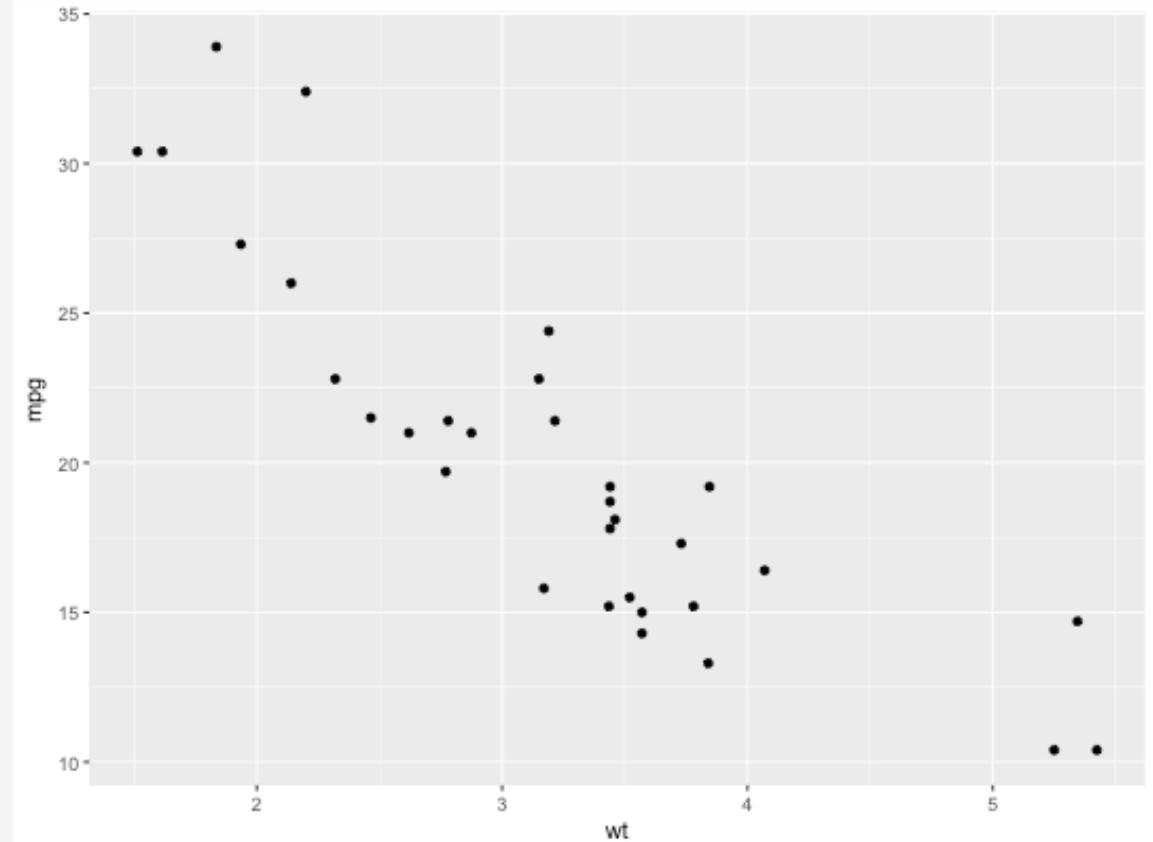


```
# If you need to flip the order (because you've flipped the plot)  
# call position_stack() explicitly:  
g +  
  geom_bar(aes(fill = drv), position = position_stack(reverse = TRUE)) +  
  coord_flip() +  
  theme(legend.position = "top")
```



# ggplot2 en R

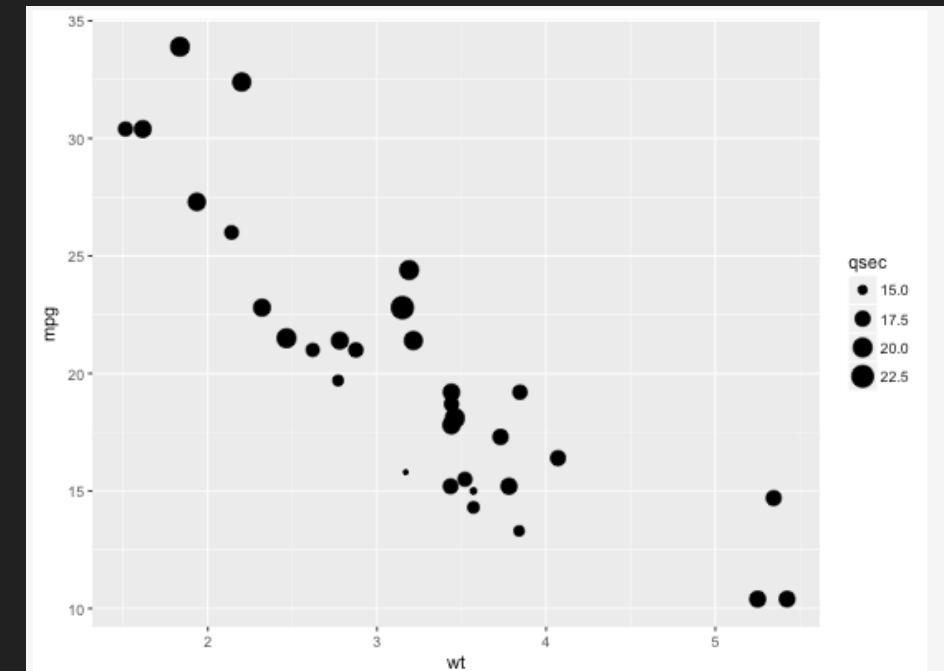
```
p <- ggplot(mtcars, aes(wt, mpg))  
p + geom_point()
```



```
# Add aesthetic mappings  
p + geom_point(aes(colour = factor(cyl)))
```

```
p + geom_point(aes(shape = factor(cyl)))
```

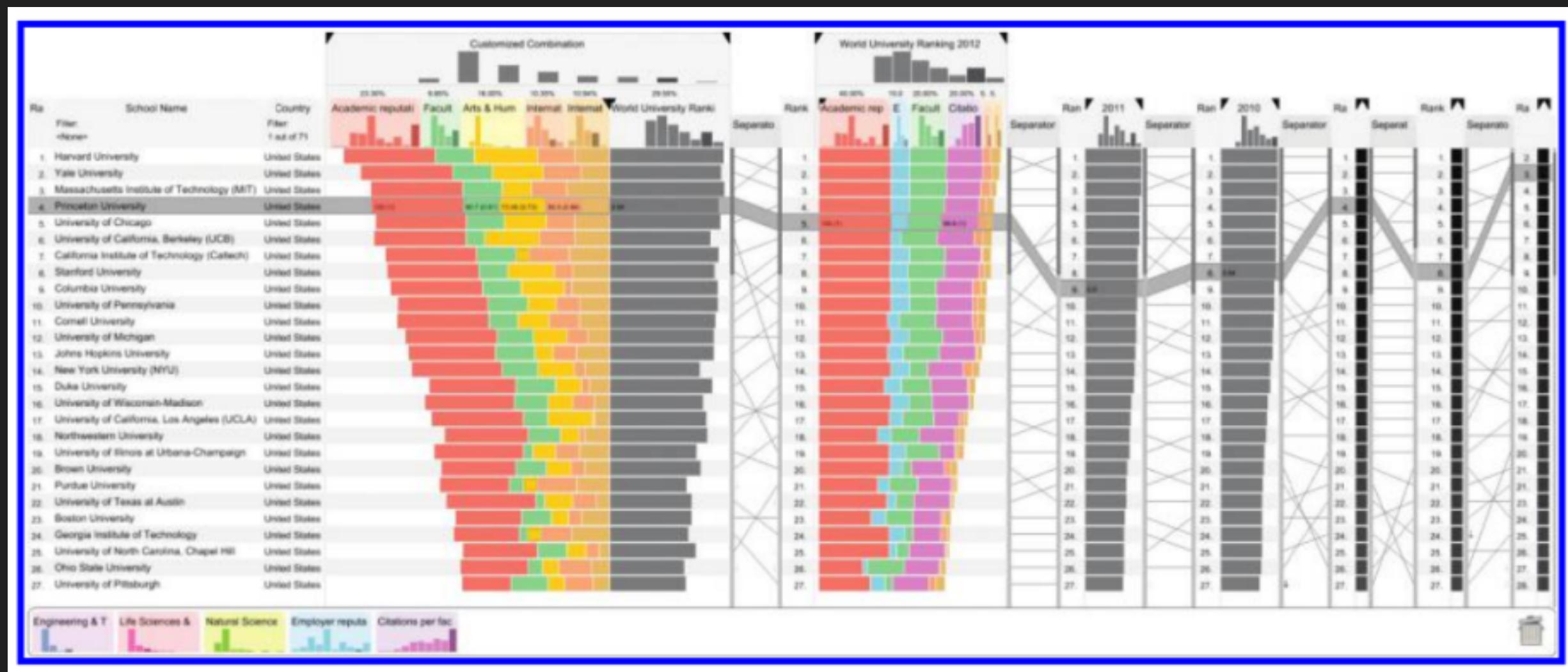
```
p + geom_point(aes(size = qsec))
```



# Cambio de vista en el tiempo

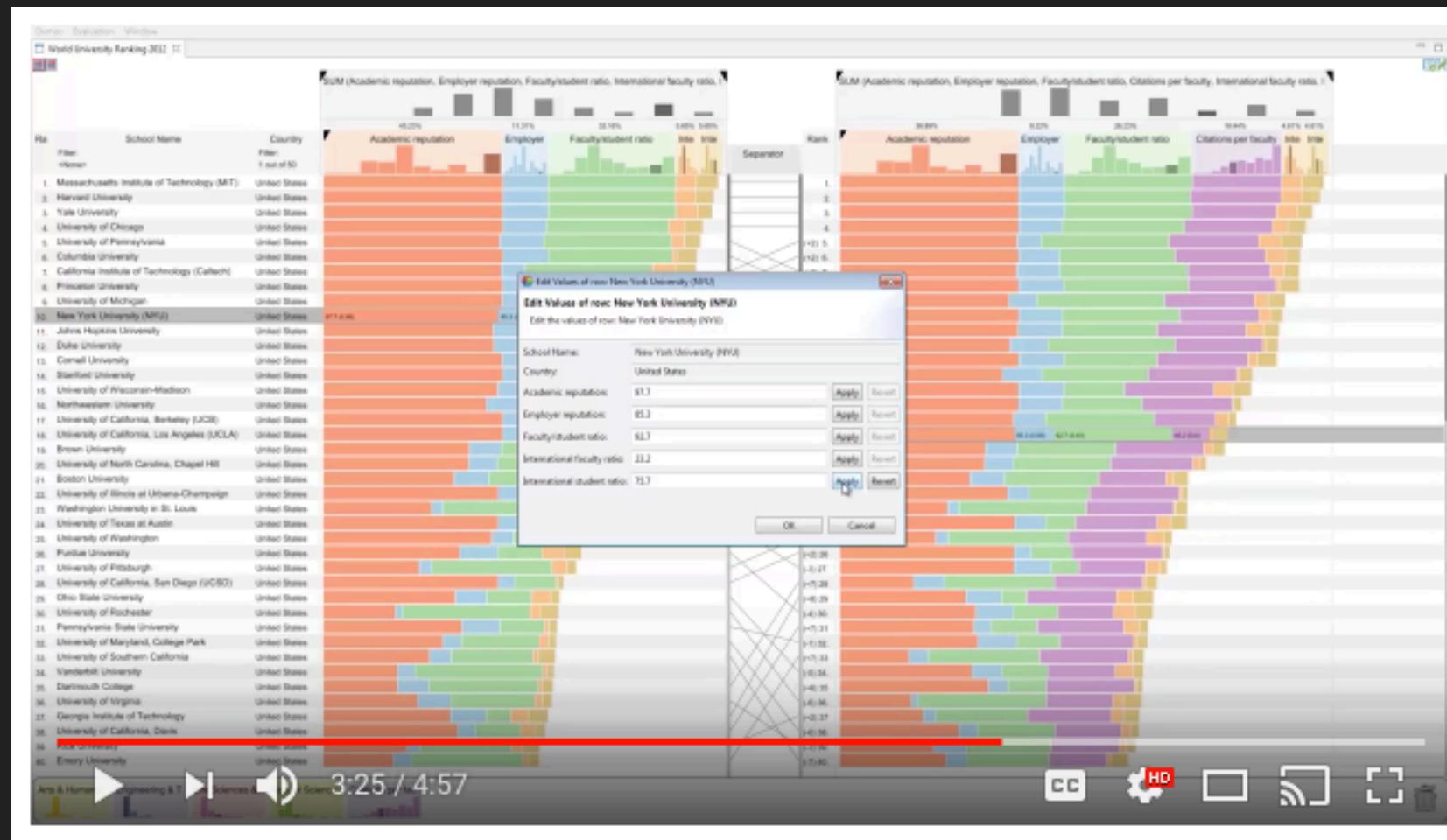
- Otros tipos de cambios están relacionados con el **reordenamiento** de algunos o todos los ítems en la vista. También conocido como *sorting*, esta es una acción poderosa para encontrar patrones en un *dataset*, ya que nos permite interactivamente ajustar el atributo con el que se ordenan los datos.
- El ejemplo más típico es el reordenamiento de una tabla, según una columna específica. Sin embargo, este mismo principio se puede utilizar para otros *visual encodings*.

# Ejemplo: LineUp



# LineUp Demo

- <https://youtu.be/iFqCBI4T8ks>



# Resumen: LineUp

System	LineUp
What: Data	Table.
What: Derived	Ordered attribute: weighted combination of selected attributes.
How: Encode	Stacked bar charts, slope graphs.
How: Manipulate	Reorder, realign, animated transitions.
Why: Task	Compare rankings, distributions.

# Reorder.js Reordering tables and networks

- <https://vimeo.com/136253887>



# Reorder.js Reordering tables and networks

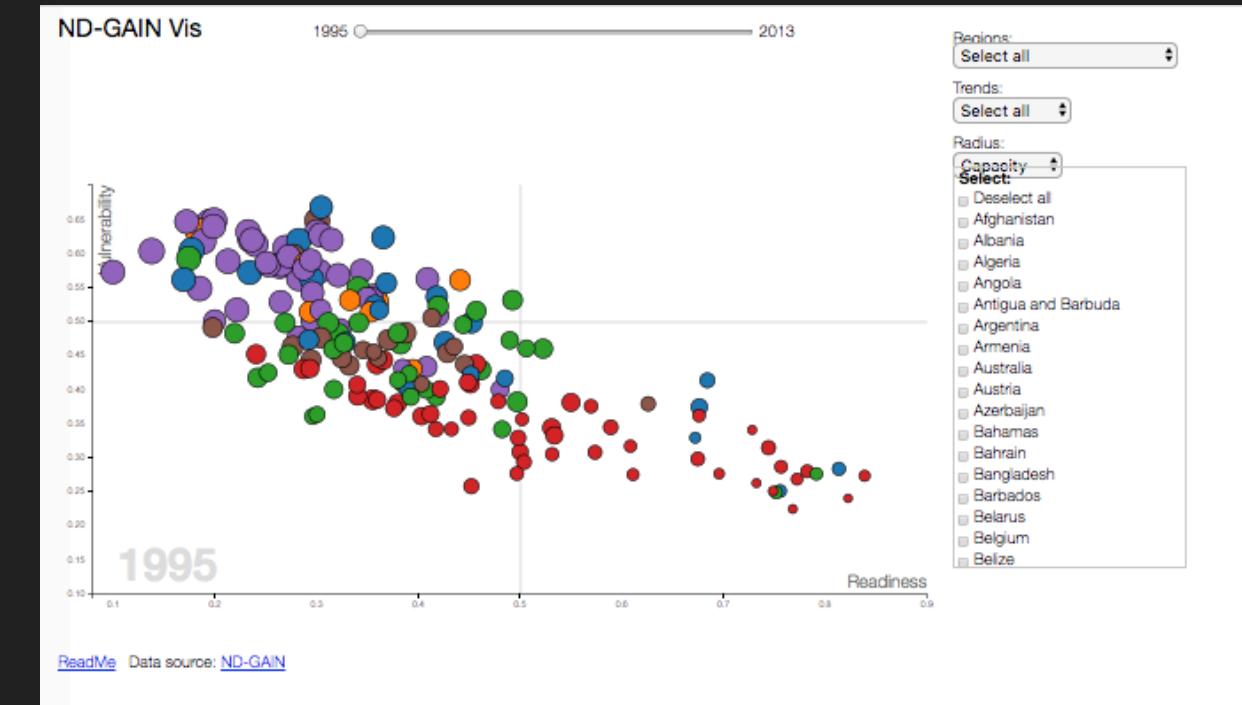
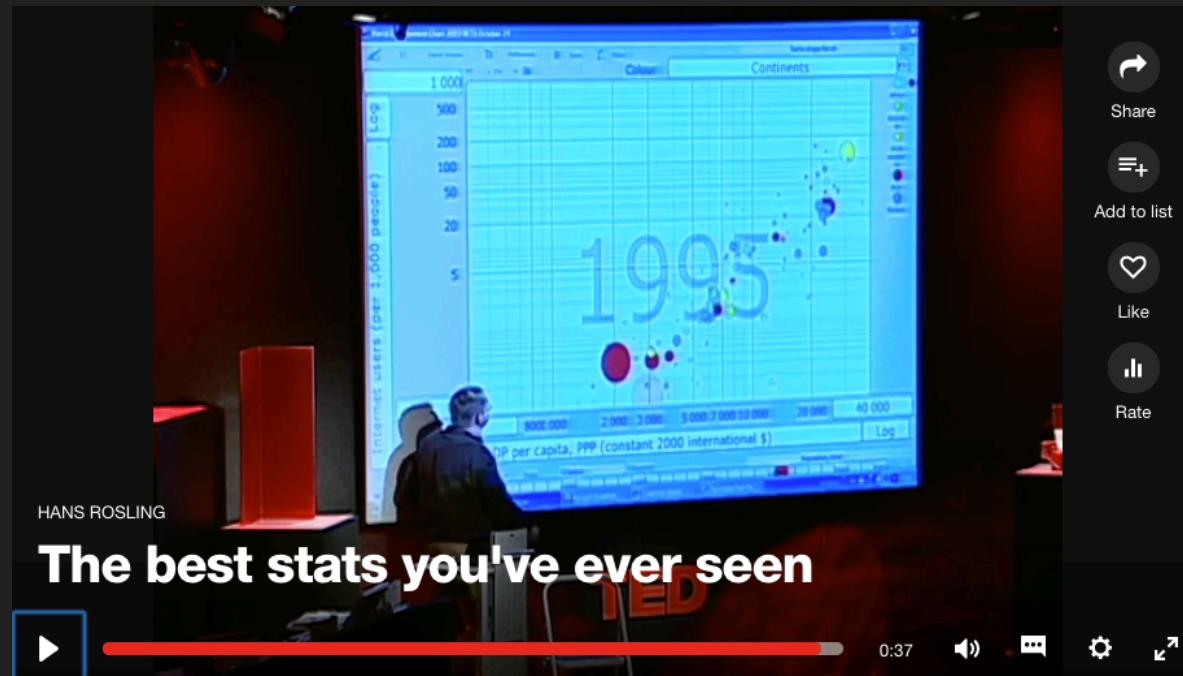
- <https://vimeo.com/136253887>



<b>Idiom</b>	<b>Animated Transitions</b>
<b>What: Data</b>	Compound network.
<b>How: Manipulate</b>	Change with animated transition. Navigate between aggregation levels.

# Hans Rosling

- Demo: <https://www3.nd.edu/~cwang11/nd-gain/vis.html>



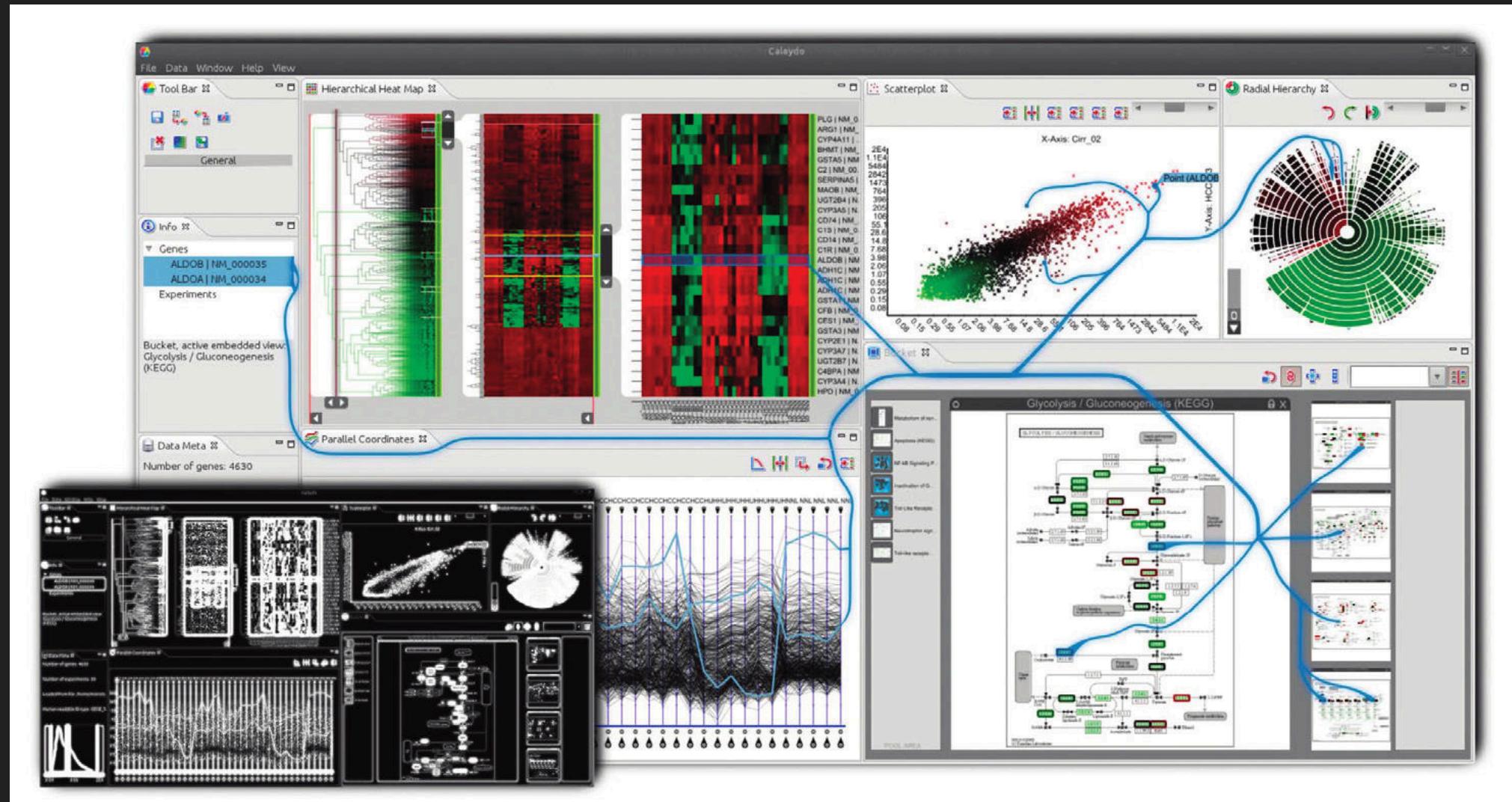
# Seleccionar elementos

- Una de las acciones fundamentales que aparece en (casi) todas las visualizaciones interactivas es permitirle al usuario **seleccionar uno o más elementos** de interés.
- El resultado de una selección es, generalmente, el *input* para realizar una operación subsecuente.

# Opciones al seleccionar

- Selección
  - Qué elementos son seleccionables (¿datos, atributos?)
  - Tipos de selección (¿seleccionado/no seleccionado?) hover, click, double-click, etc. / por niveles
  - Tipos de selección pueden cambiar por dispositivo (touch-screens)
  - Cuántos elementos: ¿Uno o varios ítems?
- Énfasis (highlighting)
  - Idiom de interacción : cómo el usuario selecciona elementos
  - Idiom de visual encoding: cómo los elementos seleccionados son enfatizados

# Conectar/coordinar vistas



# SetFusion

- Denis Parra and Peter Brusilovsky. 2015. User-controllable personalization. *Int. J. Hum.-Comput. Stud.* 78, C

<http://halley.exp.sis.pitt.edu/cn3/recsys.php?conferenceID=114&taction=system>

Tune weights of the recommender methods:

- Most bookmarked papers 0.4
- Similar to your favorite articles 0.6
- Frequently cited authors in ACM DL 0.4

Update Recommendation List →

\* Hover over circles to explore articles  
\* Click on the diagram to highlight subsets

Similar to your favorite articles

Most bookmarked papers

Frequently cited authors

3. Recommendation with Differential Context Weighting

1. Adaptive Visualization of Research Communities

by Patrick M. Dudas, Peter Brusilovsky, Martijn de Jongh

Type: Workshop Paper [\[see abstract\]](#)

Rate the relevance of this talk: (not relevant) 1 2 3 4 5 (strongly relevant)

2. Interaction Based Content Recommendation in Online Communities

by Surya Nepal, Cecile Paris, Payam Aghaei Pour, Jill Freyne, Sanat Kumar Bista

Type: Long Paper [\[see abstract\]](#)

Rate the relevance of this talk: (not relevant) 1 2 3 4 5 (strongly relevant)

3. Recommendation with Differential Context Weighting

by Robin Burke, Bamshad Mobasher, Yong Zheng

Type: Long Paper [\[see abstract\]](#)

Rate the relevance of this talk: (not relevant) 1 2 3 4 5 (strongly relevant)

4. Scrutable User Models and Personalised Item Recommendation in Mobile Lifestyle Applications

by Rainer Wasinger, James Wallbank, Luiz Pizzato, Judy Kay, Bob Kummerfeld, Matthias Bohmer, Antonio Kruger

Type: Long Paper [\[see abstract\]](#)

Rate the relevance of this talk: (not relevant) 1 2 3 4 5 (strongly relevant)

5. Personalized access to scientific publications: from recommendation to explanation

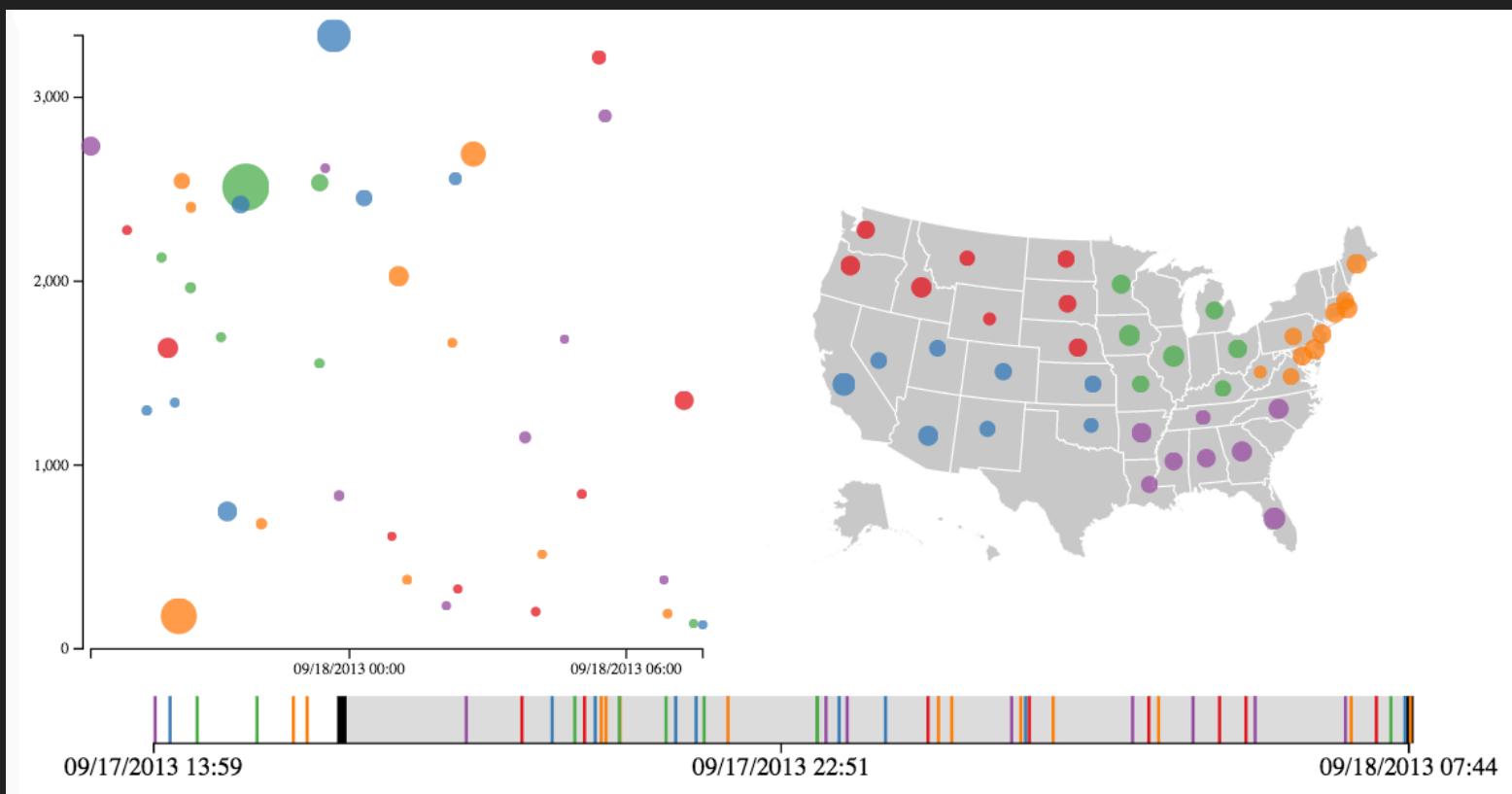
by Felice Ferrara, Dario De Nart, Carlo Tasso

Type: Short Paper [\[see abstract\]](#)

Rate the relevance of this talk: (not relevant) 1 2 3 4 5 (strongly relevant)

# Demo

- <http://bl.ocks.org/dukevis/raw/6768900/>

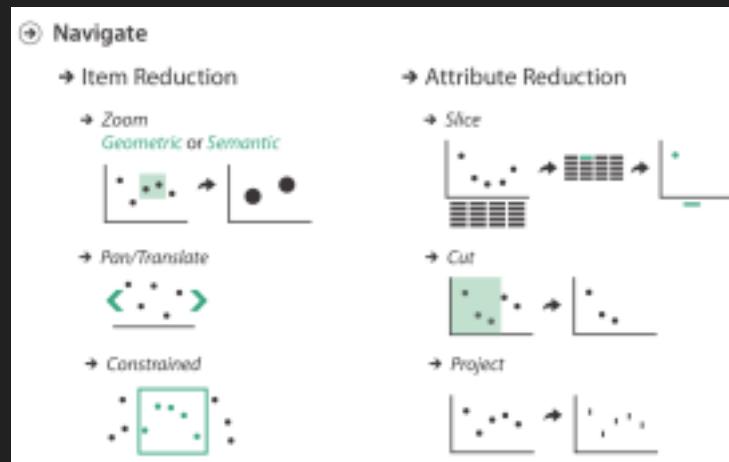


# Conectar/coordinar vistas

Idiom	Context-Preserving Visual Links
What: Data	Any data.
How: Encode	Any encoding. Highlight with link marks connecting items across views.
How: Manipulate	Select any element.
(How: Coordinate)	Juxtaposed multiple views.

# Navegar: cambiando el punto de vista

- Como ya fue mencionado, los *datasets* de gran tamaño generalmente no pueden ser comprendidos desde un único punto de vista.
- La gran mayoría de los sistemas interactivos realizan una **metáfora de navegación**, análoga a la navegación por el mundo físico. En estos sistemas, el *layout* especial queda fijo y la navegación actúa como un **cambio del *viewpoint***.



# Navegar: cambiando el punto de vista

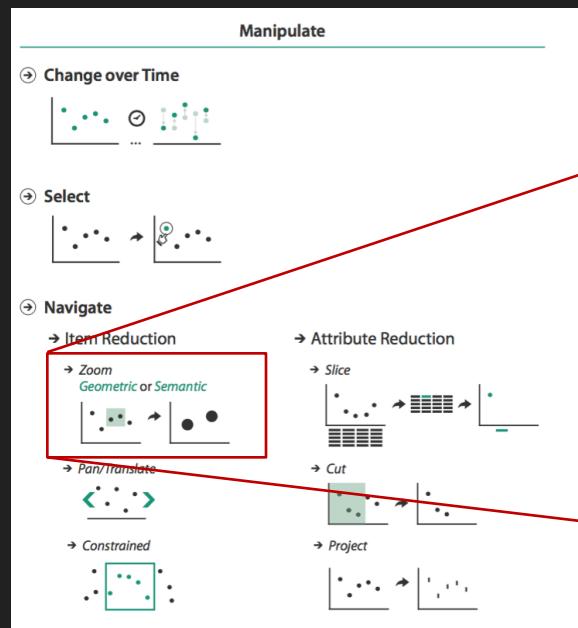
- La navegación puede ser descompuesta en **tres componentes**.
- *Zooming* es la acción que aleja o acerca la cámara. Al alejarse, deberíamos menos ítems, pero con mayor tamaño, y lo contrario al acercarse.
- *Panning* es la acción que mueve la cámara de forma paralela al plano de la imagen, ya sea verticalmente u horizontalmente. Cuando estamos en un contexto tridimensional, se hablará de *translating*.
- *Rotating* es la acción que gira la cámara alrededor de su propio eje, siendo esto muchísimo más predominante en un sistema 3D.

# Navegar: cambiando el punto de vista

- El efecto de cambiar el punto de vista cambiará también el conjunto de los elementos visibles, logrando un resultado que podría combinar filtro y agregación.
- Por ejemplo, al hacer *zoom in* o *panning* (con una cámara *zoomed-in*), filtra según el *layout* espacial.

# Tipos de *zooming*

- Existen dos tipos de *zooming*: geométrico y semántico.
- La forma más intuitiva es la geométrica, ya que corresponde casi exactamente a nuestra experiencia de la vida real.

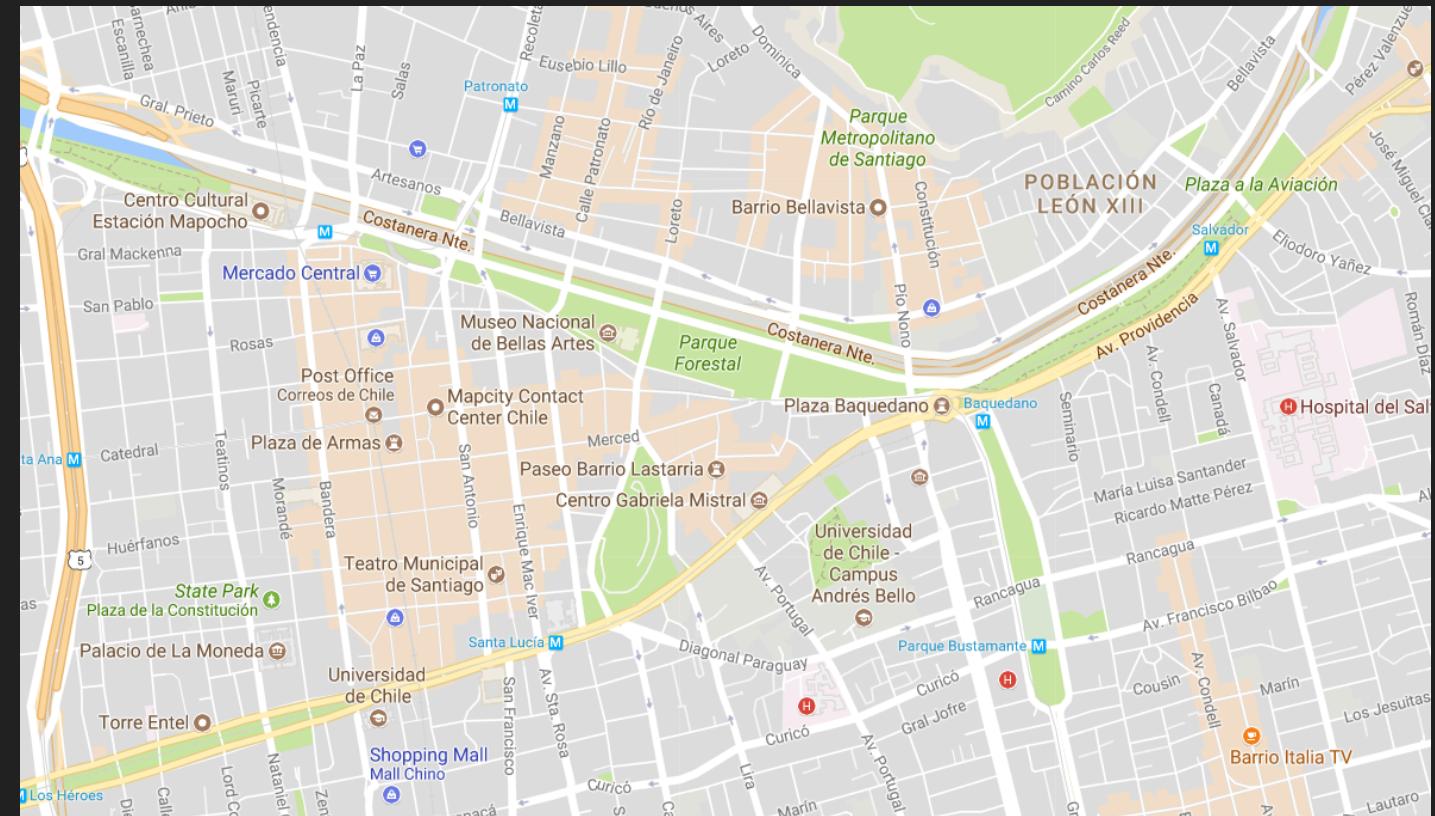
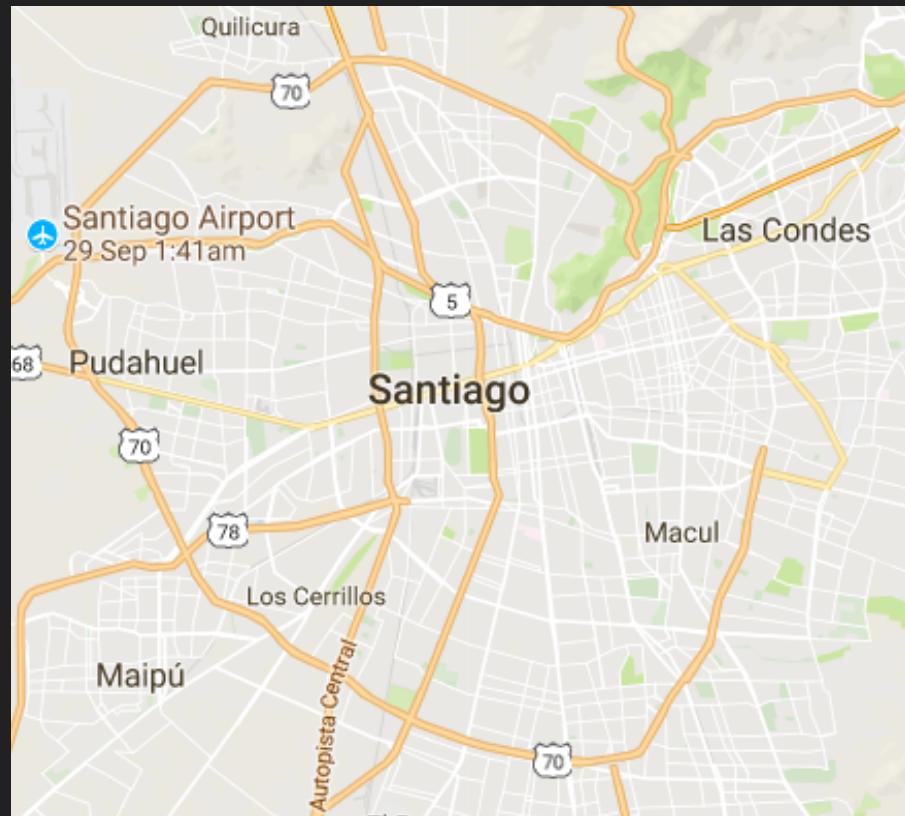


→ Zoom  
Geometric or Semantic



# Zooming semántico

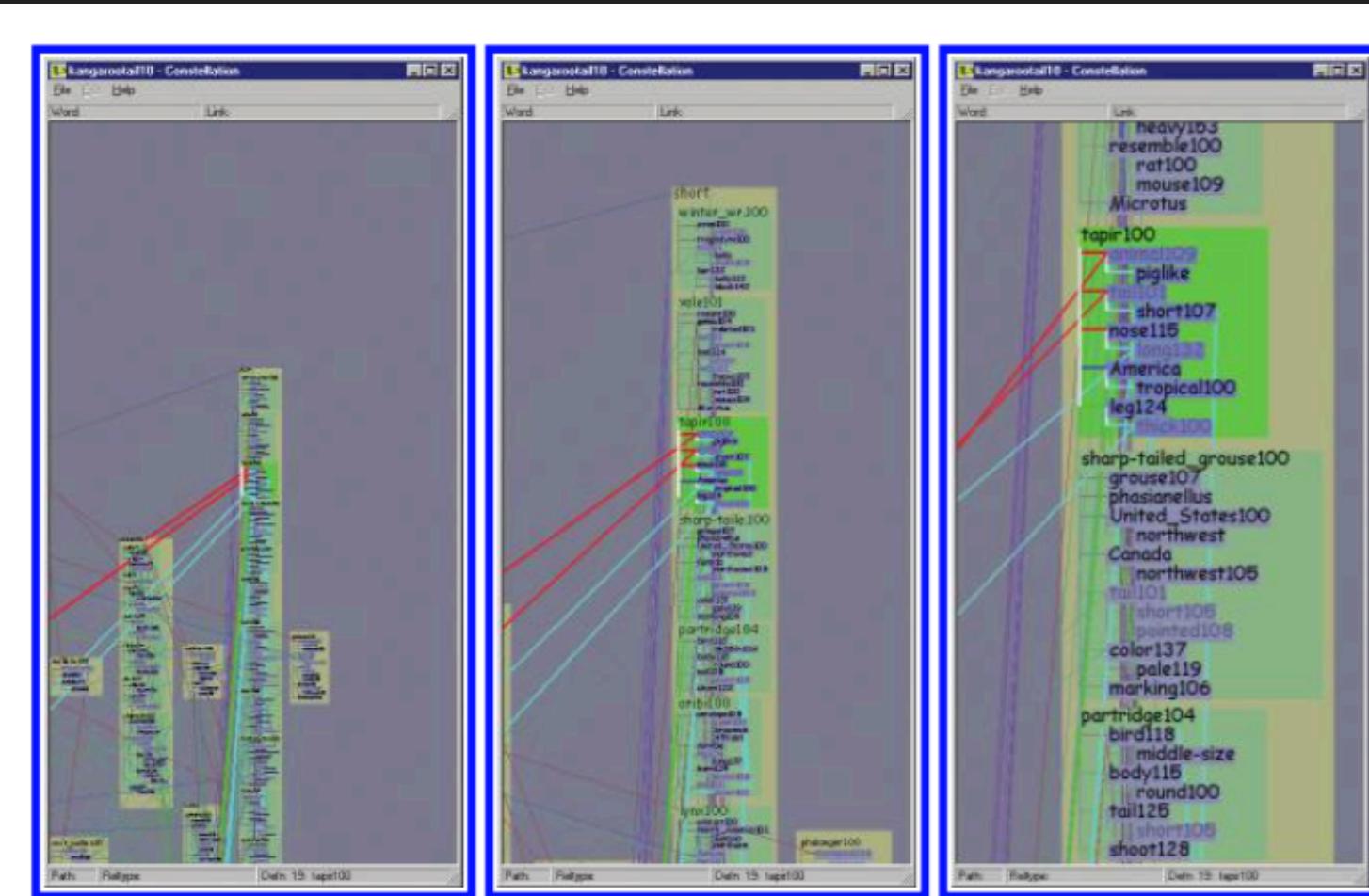
- Nivel de detalle cambia con distintos niveles de zoom



# Navegación condicionada

- Cuando la navegación es completamente libre (unconstrained) la "cámara" pueda moverse a cualquier lugar, pero encontrar un punto de visión adecuado no es trivial.
- Si bien el problema se presenta más comúnmente en 3D, también es posible encontrar problemas de navegación en 2D, por ejemplo al perderse con panning y zooming excesivo.
- La navegación condicionada (constrained) permite limitar la vista de la cámara para simplificar el problema, por ejemplo: limitar el zoom de la cámara.

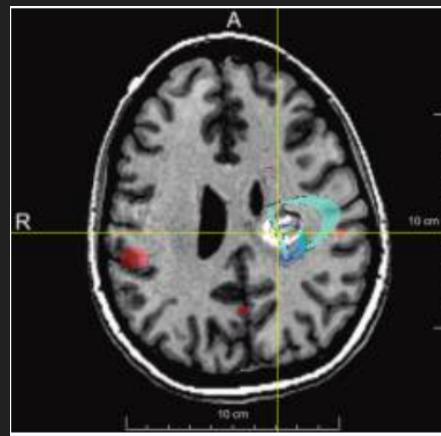
# Ejemplo: Manipulación en Constellation



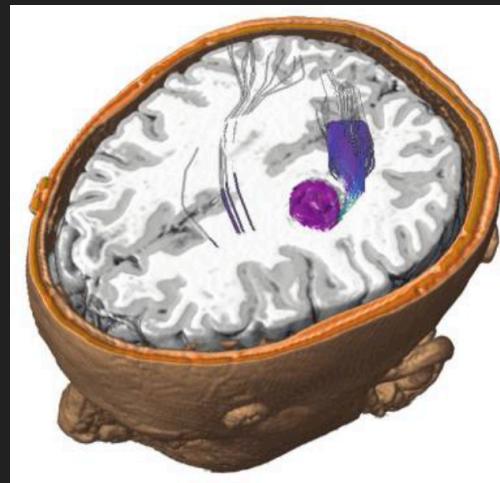
**Figure 15.20.** Constellation uses a subtle version of the semantic zooming design choice, where the space allocated for the first word versus the rest of the definition changes according to the zoom level. From [Munzner 00, Figure 5.19].

# Navegación: Reducción de Atributos

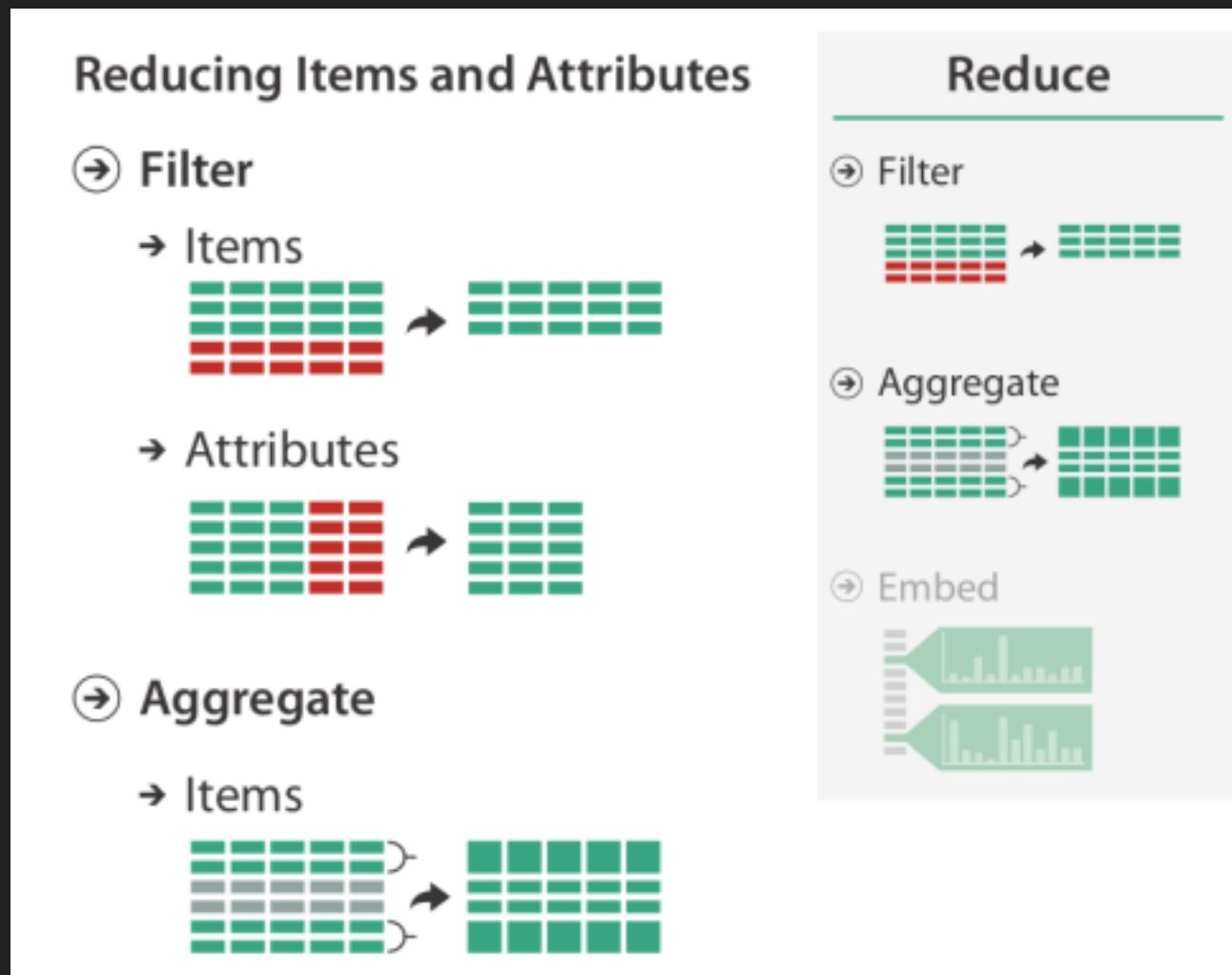
- Slice: Se elige un sólo valor de la dimensión a eliminar y se muestre el resto de las otras dimensiones.



- Cut: En este idiom el usuario puede posicionar un plano de forma que el volumen de visión se divide en dos, mostrando la parte más alejada y escondiendo la más cercana.



# Navegación: reducir atributos o items



# Reducir atributos (capítulo 13)

- Ahora veremos cómo reducir lo que mostramos en una vista, mediante las técnicas de *filtering* y *aggregation*. El primero elimina elementos, mientras que el segundo se encarga de combinarlos.
- Ambas técnicas pueden ser aplicadas tanto a ítems como a atributos.

# Referencias

- *Visualization: Analysis & Design*, Tamara Munzner