

# Visualización de información y analítica visual

**Clase 5:** Datos tabulares y manipulación de vistas

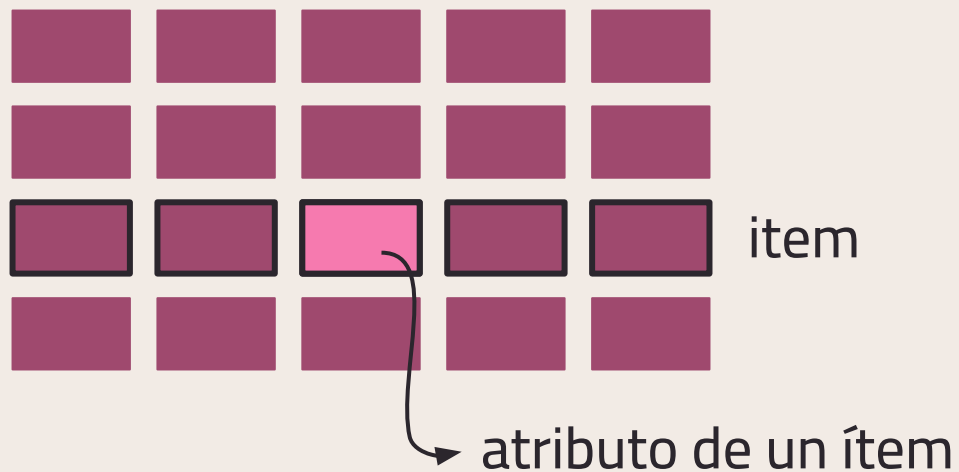


**Hoy revisaremos idioms espaciales  
para representar datos tabulares**



# Datos tabulares

---



## Acomodando datos tabulares...

Acomodar (arrange) datos tabulares cubre los aspectos del uso de canales espaciales en la visualización

- **Recordemos que los canales más efectivos para codificar atributos cuantitativos y ordenados son:**

- Posición en escalas alineadas



- Posición en escalas desalineadas





- Largo



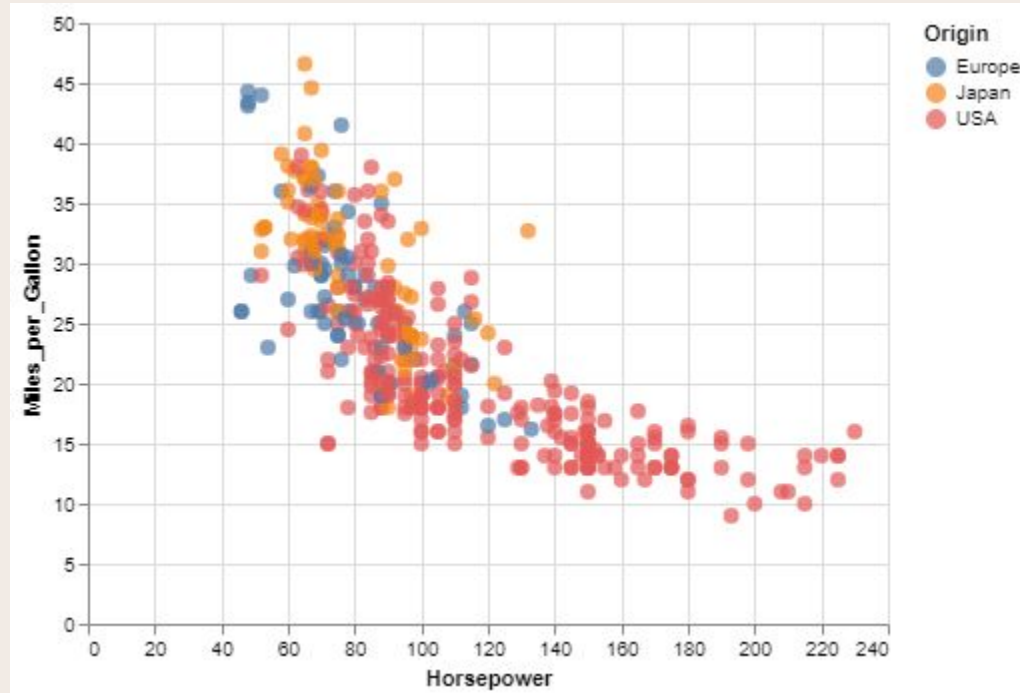
## Acomodando datos tabulares...

Un concepto especial es arreglar tablas en base a pares **llave-valor**

- **Llave**  : atributo independiente que puede usarse como un índice único para buscar ítems en una tabla
- **Valor**  : atributo dependiente , el valor de una de las celdas de la tabla

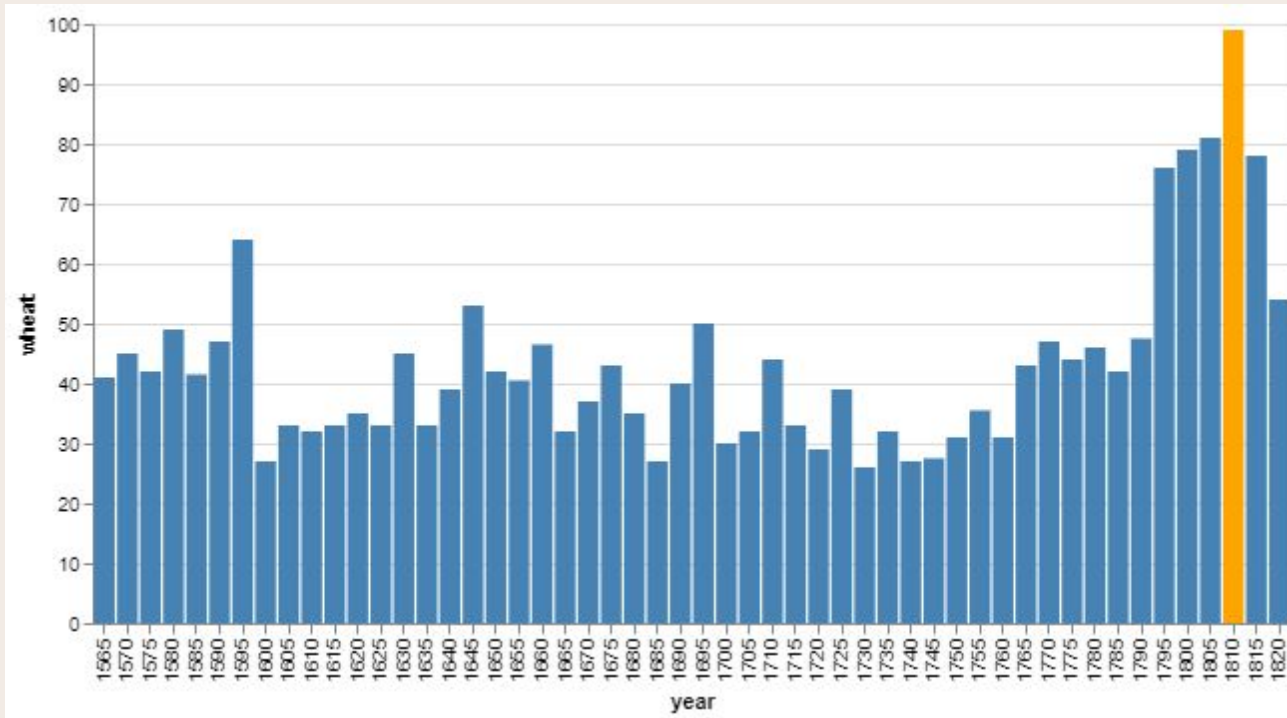
## Acomodando datos tabulares...

Hay algunos casos donde queremos codificar solo valores...



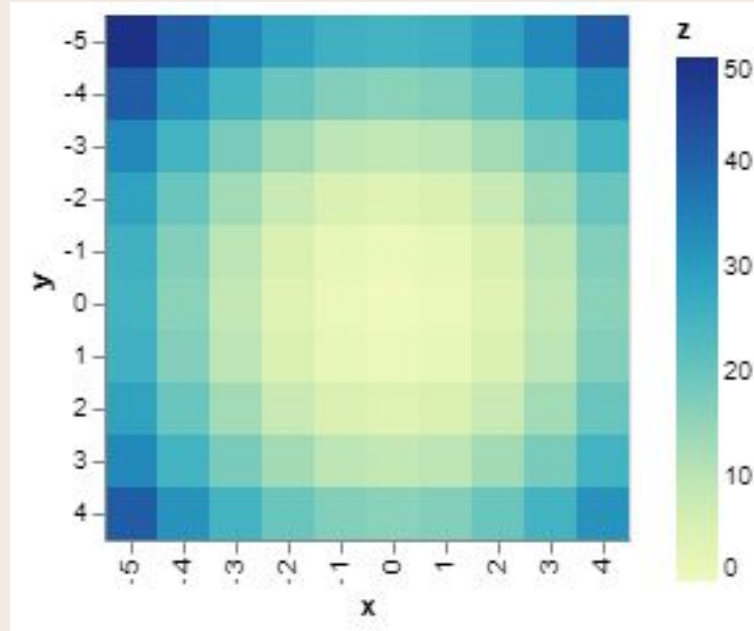
# Acomodando datos tabulares...

Otras veces tenemos una llave y un valor



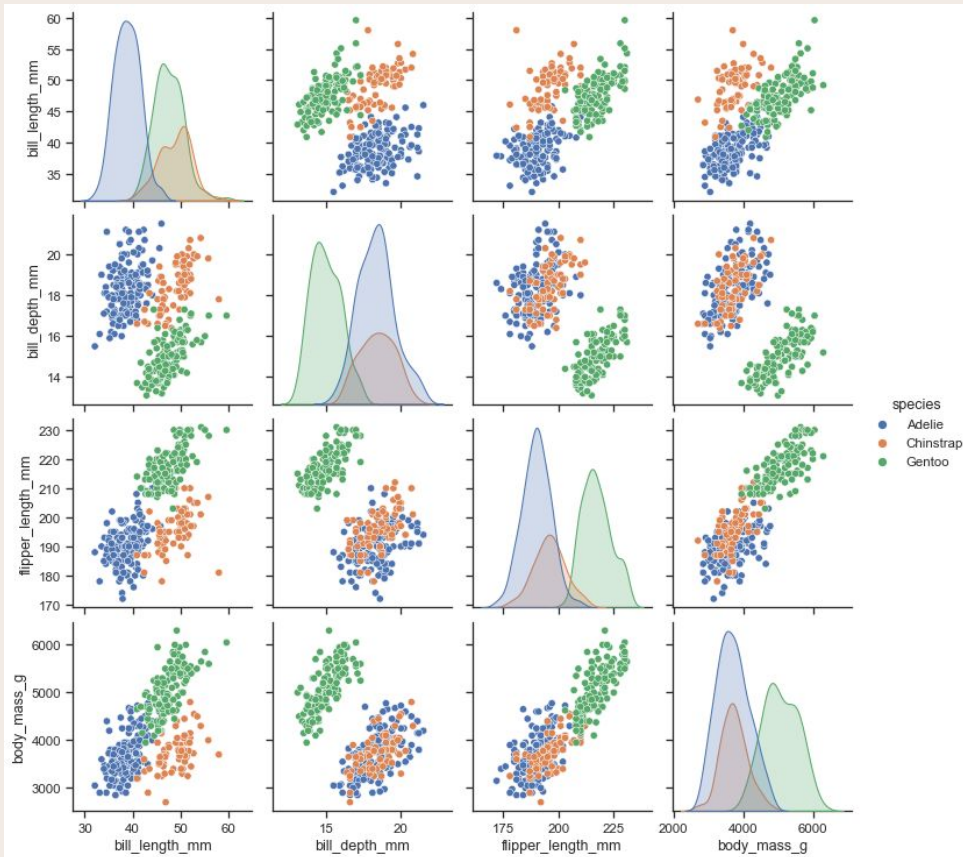
## Acomodando datos tabulares...

Y otras veces que tenemos dos llaves y un valor





# Acomodando datos tabulares...



Y podemos empezar a ir subdividiendo los espacios recursivamente para codificar más llaves y valores

## Expresar valores cuantitativos

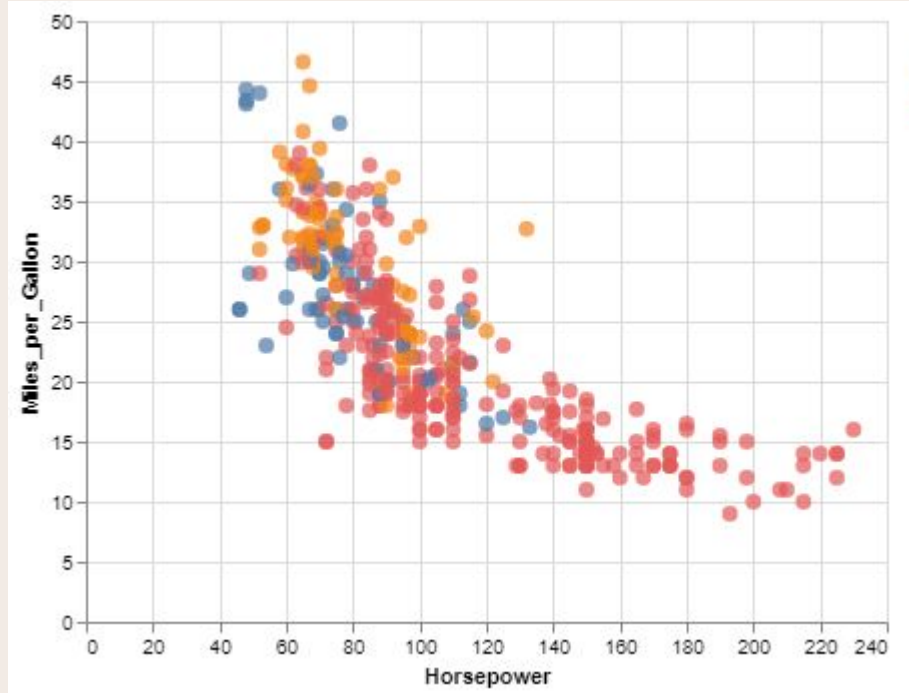
Lo más directo es ocupar el espacio para codificar valores cuantitativos:

***El atributo se mapea a la posición espacial a lo largo de un eje***

- Si hay solo un valor, entonces se ponen marcas en alguna posición del eje: otros atributos pueden codificarse en la misma marca agregando canales no espaciales...



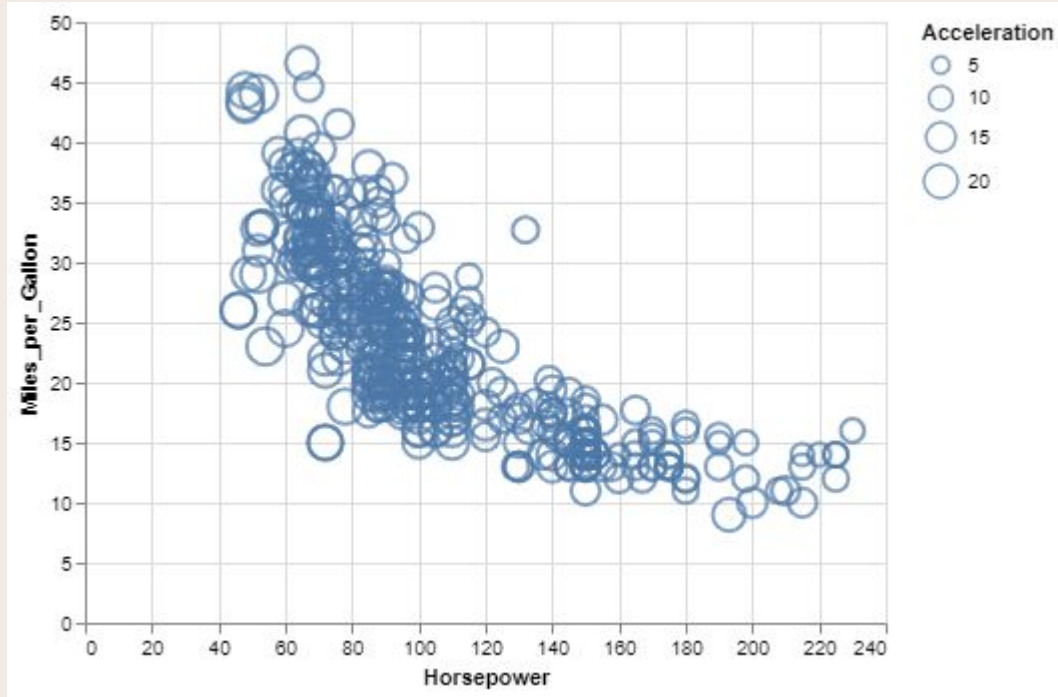
# Scatterplots



Un scatterplot permite codificar dos valores cuantitativos usando **los ejes XY**, y usando como **marca un punto**.

- buenos para dar un overview de los datos: distribución y outliers
- muy utiles tambien para la tarea de hallar correlaciones

# Scatterplots: bubbleplots



Podemos agregar otras codificaciones al scatterplot como color, o tamaño de las marcas.

Cuando agregamos el canal de tamaño entonces la visualización se conoce como **bubbleplot**

# Scatterplots

---

Idiom	Scatterplots
<b>Que:</b> Datos	Tabla: dos atributos con valores cuantitativos
<b>Como:</b> codificación	Los valores se codifican en los ejes XY y las marcas usadas son puntos
<b>Por qué:</b> tarea	Encontrar tendencias, outliers, distribuciones, correlaciones, identificar clusters
<b>Escala</b>	Cientos de items

## Separar, ordenar y alinear: regiones categóricas

El uso del espacio para codificar atributos categóricos es más complejo

- El espacio es inherentemente ordenado y de magnitud
- Los atributos categóricos tienen semánticas no-ordenadas

**Codificar valores categóricos directamente en canales espaciales  
viola el *principio de expresividad***

## Separar, ordenar y alinear: regiones categóricas

Para codificar valores categóricos usaremos **regiones espaciales**

**las regiones son áreas contiguas, acotadas y que son distintas entre sí**

- Ubicar distintos ítems en la misma región nos permite indicar similaridad
- Separar los datos en regiones nos da mucha flexibilidad para codificar los datos dentro de cada región



## Alineamiento de lista: una llave

Cuando tenemos una sola llave, armar las regiones según esa llave nos ubica un ítem en cada región.

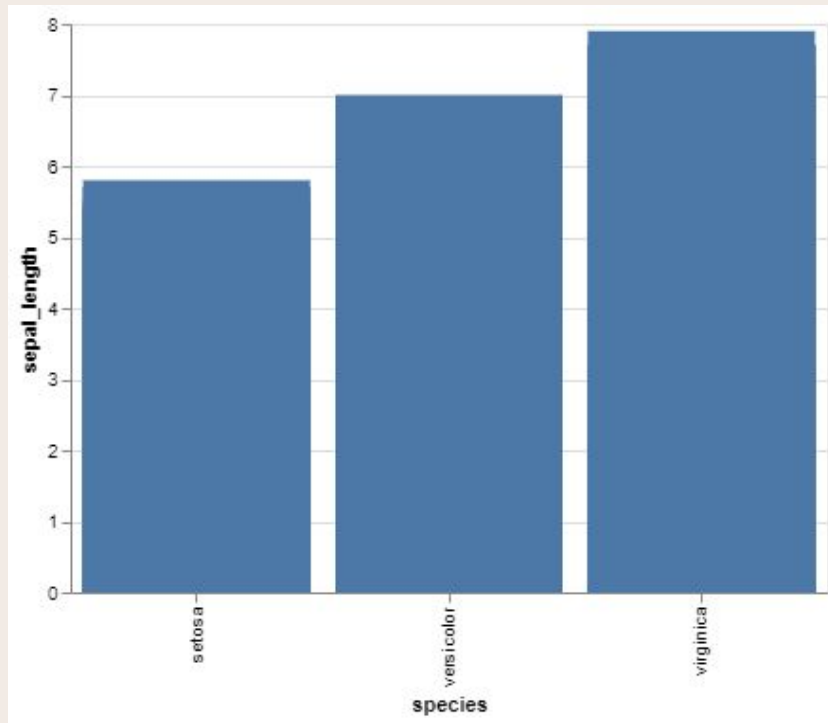
- Las regiones se arman en base a una línea, normalmente horizontal o vertical
- La vista completa tiene dos dimensiones, la lista alineada de items por una parte, y por otra la región donde se muestran los valores

**Veamos como ejemplo un bar chart**



## Bar chart

---



En el eje X creamos regiones para cada especie distinta y en el eje Y codificamos el atributo (en este caso largo del sépalo)

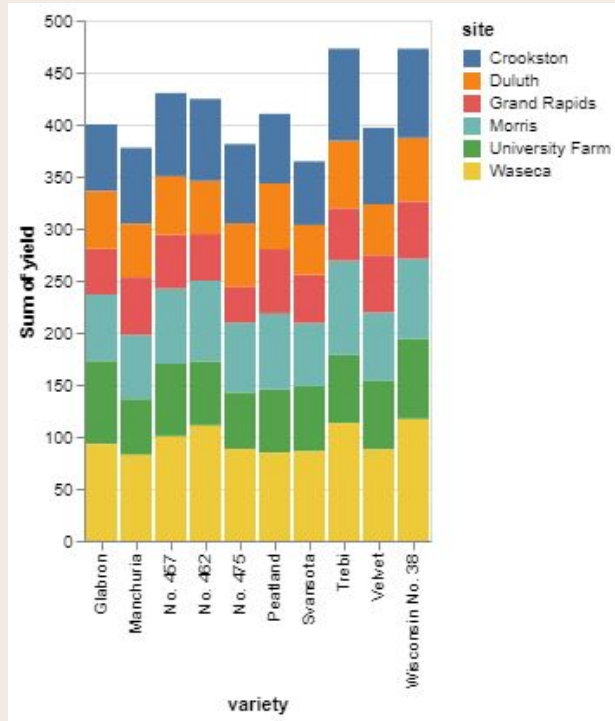
La marca utilizada en este caso es una **línea** y se alinean **para codificar de la manera más efectiva posible el valor cuantitativo.**

# Bar chart

---

Idiom	Bar chart
<b>Qué:</b> datos	Tabla: un atributo de valor cuantitativo y un atributo llave categórico
<b>Cómo:</b> codificación	Las marcas son líneas. El atributo de valor se expresa con posición en un eje alineado (vertical), los atributos llave se separan con la posición horizontal
<b>Por qué:</b> tarea	Buscar y comparar valores
<b>Escala</b>	Decenas - centenas de llaves distintas

# Stacked bar chart



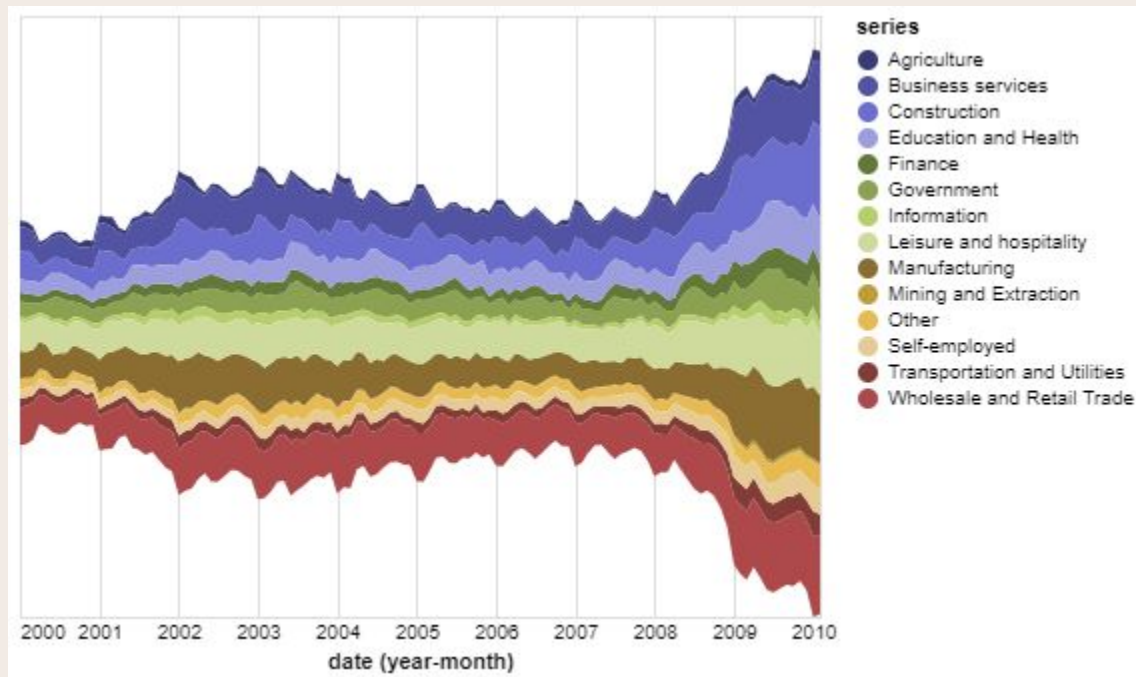
Un **stacked bar chart** usa un glifo más complejo en cada barra, donde en realidad se montan distintas barras para armar cada una.

- El largo de cada barra-glifo codifica un valor, pero también cada sub-barra.
- Un stacked bar chart permite codificar tablas multidimensionales, específicamente tablas con dos llaves

# Stacked bar chart

Idiom	Stacked bar chart
<b>Qué:</b> datos	Tabla multidimensional: un atributo valor cuantitativo y dos atributos categóricos de llave
<b>Cómo:</b> codificación	Glifo barra con subcomponentes codificados a través del largo que representan cada key secundaria. Barras separadas por categoría en la llave principal.
<b>Por qué:</b> tarea	Apreciar la relación parte-todo, buscar valores, encontrar tendencias
<b>Escala</b>	Atributo llave en el eje principal: decenas a cientos. Atributo llave stackeado: hasta 10

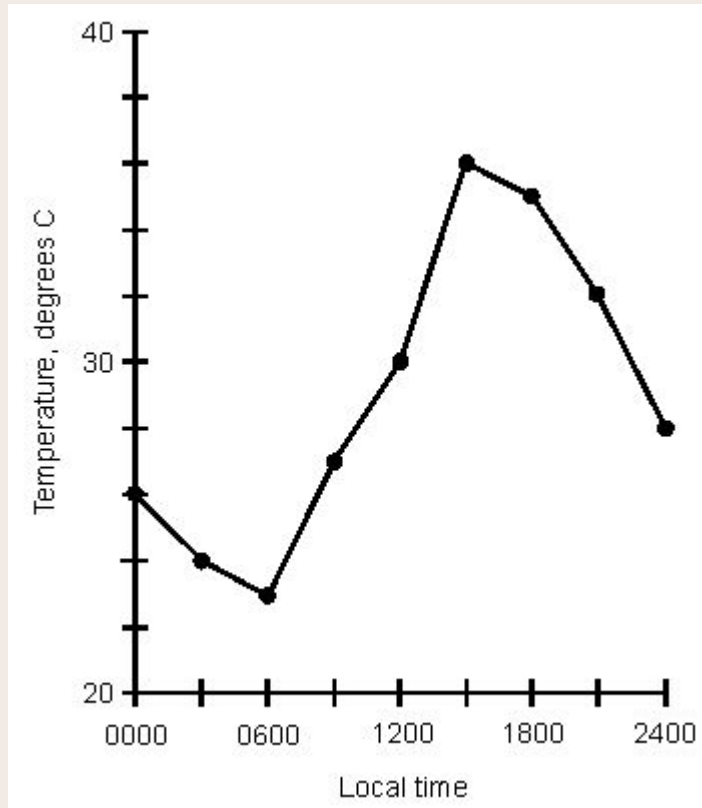
# StreamGraphs



# StreamGraphs

Idiom	StreamGraphs
<b>Qué:</b> datos	Tabla multidimensional: un valor cuantitativo (conteo), un atributo llave ordenado (tiempo), un atributo categórico llave (industria)
<b>Qué:</b> derivado	Un atributo cuantitativo (por capa)
<b>Cómo:</b> codificación	Usa la geometría derivada para mostrar una capa por industria a lo largo del tiempo, el alto de la barra codifica
<b>Por qué:</b> tarea	Mostrar tendencias
<b>Escala</b>	Atributo llave (tiempo), cientos de puntos de tiempo. Atributos llave (industria): decenas a cientos

## Dot and line charts



El gráfico de puntos codifica un atributo cuantitativo usando posición espacial versus uno cualitativo usando como marca el **punto**.

- Es como un scatterplot, pero en vez de dos atributos cuantitativos, hay uno que es categórico.
- También es como un bar chart, pero en vez de líneas, usamos puntos

# Dot and line charts

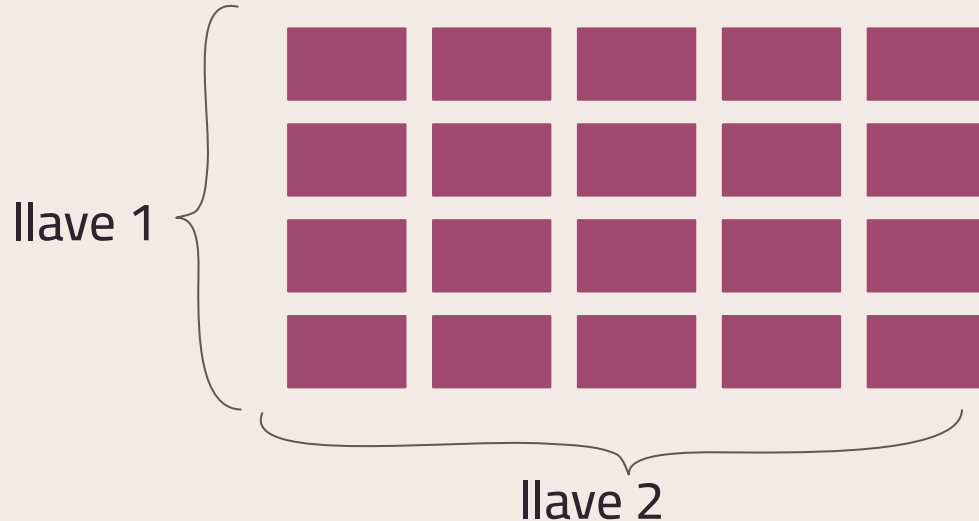
Idiom	Dot and line chart
<b>Qué:</b> datos	Tabla: un atributo de valor cuantitativo y un atributo llave <b>ordenado</b> .
<b>Cómo:</b> codificación	Expresamos los valores de los atributos con posición vertical alineada y marcas de puntos. Separamos y/o ordenamos las regiones horizontales según el valor del atributo llave. <ul style="list-style-type: none"><li>• Si <b>unimos los puntos</b> tenemos un line chart</li></ul>
<b>Por qué:</b> tarea	Mostrar tendencias
<b>Escala</b>	Cientos de atributos llaves



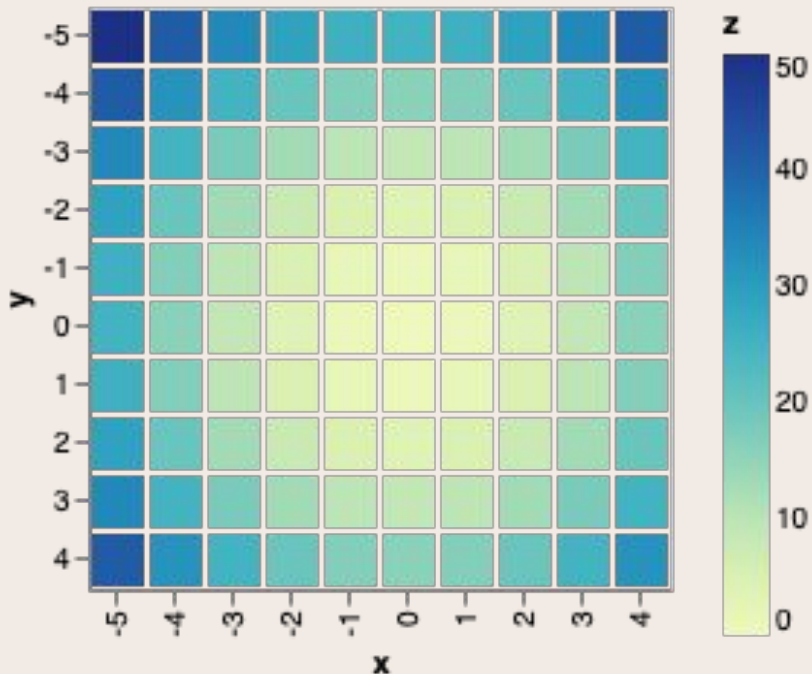
## Alineamiento de matriz: dos llaves

Cuando hay dos llaves, lo más común es usar arreglos de tipo matriciales, ubicando una llave en las filas y otra en las columnas.

- cada celda entonces de la matriz es la región para ese par de matrices



# Heatmaps



El arreglo matricial de datos más directo es el **heatmap**

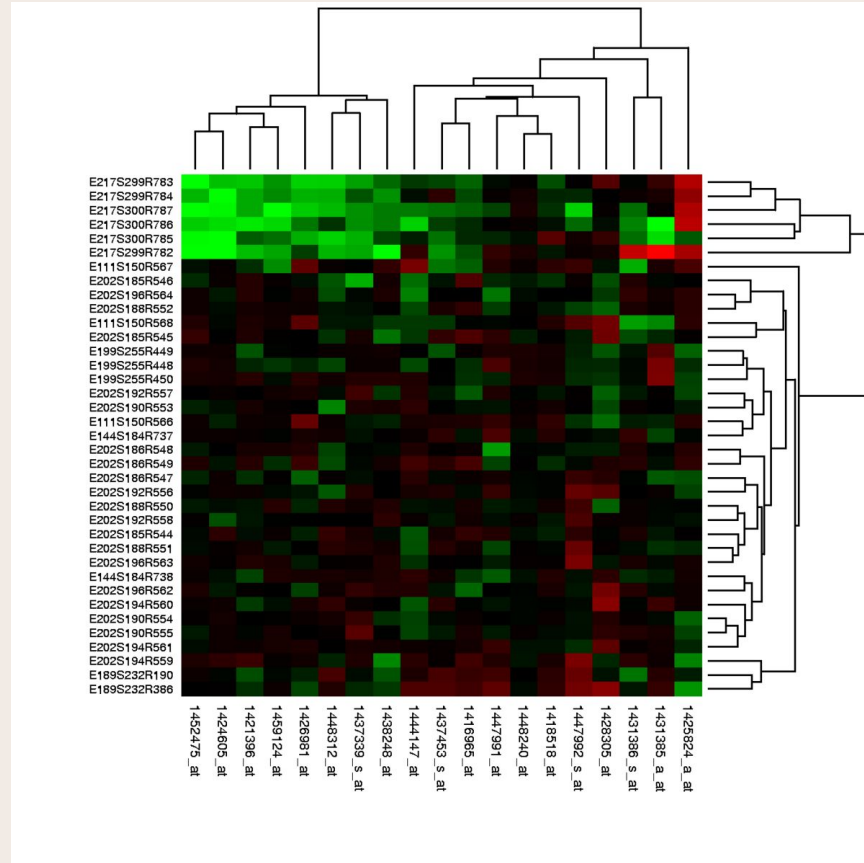
Cada celda ocupa completamente la región asignada y codifica un valor cuantitativo (con color, normalmente colormaps divergentes)

# Heatmaps

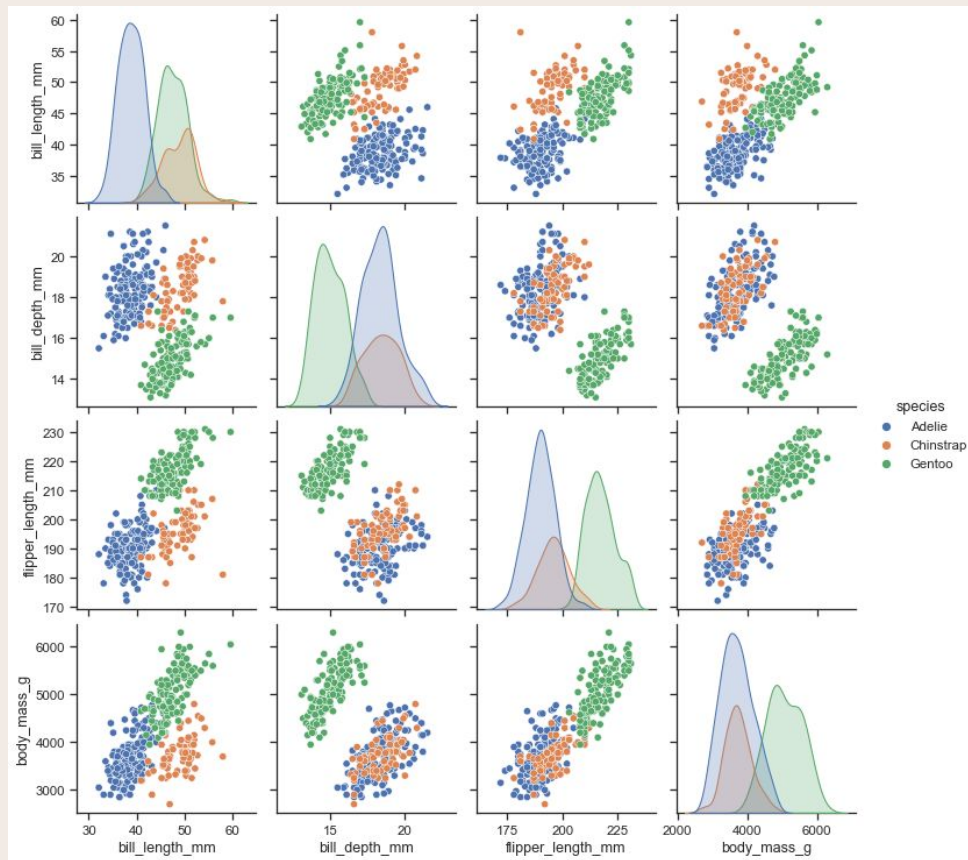
---

Idiom	Heatmaps
<b>Qué:</b> datos	Tabla con dos atributos llaves categóricos y un atributo de valor cuantitativo
<b>Cómo:</b> codificación	Matriz 2D con marcas de tipo área y un colormap divergente
<b>Por qué:</b> tarea	Encontrar clusters, outliers, resumir
<b>Escala</b>	Atributos categoricos, hasta cientos, atributos cuantitativos entre 3 y 11 (distinguibles, por diferencia de color)

# Cluster heatmaps



# Scatterplot matrix



Otra manera de arreglar los datos matricialmente es un **scatterplot matrix**: en cada celda de la matriz tenemos un scatterplot (la diagonal normalmente se reemplaza por otra visualización).

- En general en cada celda podemos tener un gráfico completo

# Heatmaps

Idiom	SPLOM
<b>Qué:</b> datos	Tabla
<b>Qué:</b> datos derivados	Atributo llave ordenado: lista de atributos originales
<b>Cómo:</b> codificación	Scatterplots en alineaciones 2D
<b>Por qué:</b> tarea	Ver correlaciones, tendencias, outliers
<b>Escala</b>	Una decena de atributos Items: decenas a cientos

## Grilla volumétrica: tres llaves

---

Al igual que podemos usar una o dos dimensiones, también podemos acomodar los datos en tres dimensiones: en una grilla volumétrica.



Sin embargo, esta elección de diseño típicamente no es recomendada porque tiene muchos problemas perceptuales (clase 4).

**Ahora veremos algunas alternativas para codificar datasets con tres llaves** 🔑 🔑 🔑

## Para múltiples llaves: subdivisión recursiva

Para codificar múltiples llaves, podemos hacer subdivisión recursiva:

- subdividimos las celdas de una lista
- subdividimos las celdas de una matriz

Dentro de cada celda, podemos incluir nuevas llaves, usando el concepto de **contención**

**Veremos esto en más detalle en algunas clases más**



## Orientación de los ejes espaciales

Otro aspecto de diseño a considerar cuando estamos haciendo uso del espacio es la orientación de los ejes espaciales. Podemos usar un layout:

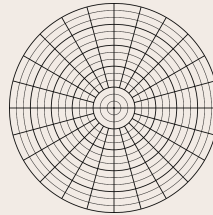
- Rectilíneo



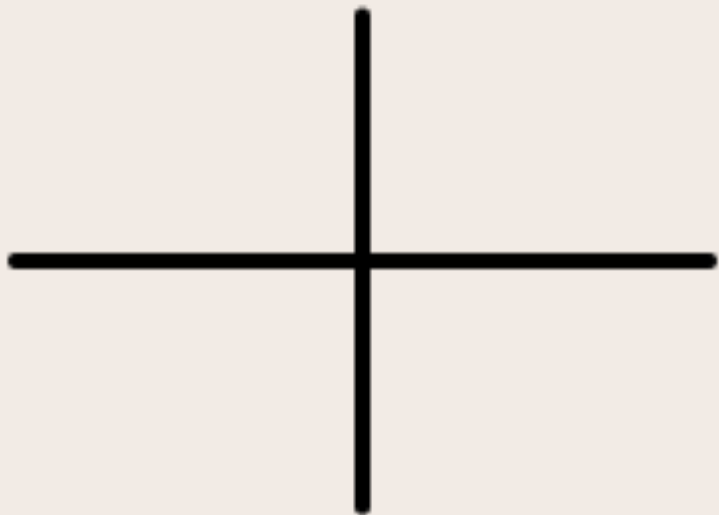
- Paralelo



- Radial



# Layout rectilíneo

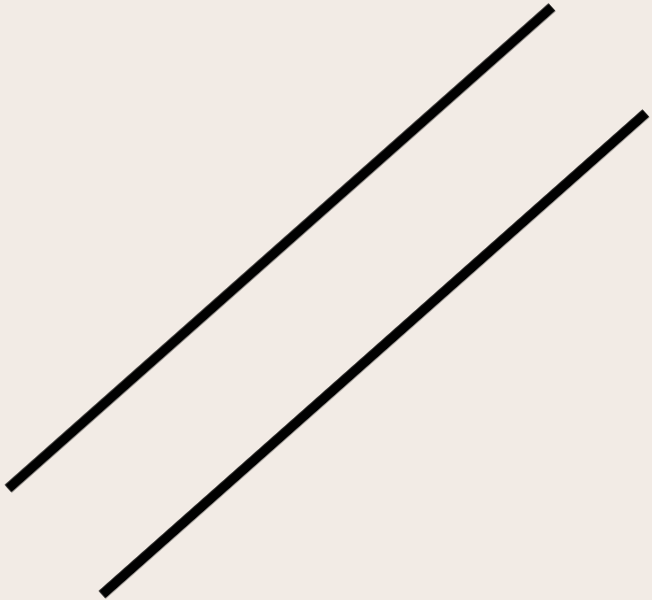


En el **layout rectilíneo** los ítems y/o regiones se distribuyen a lo largo de dos ejes.

El layout rectilíneo es el más frecuentemente utilizado en visualizaciones en general

💡 ***Todos los ejemplos que vimos hoy usan layout rectilíneo***

## Layout paralelo



Un **layout paralelo** nos permite codificar varios atributos cuantitativos simultáneamente usando posición espacial.

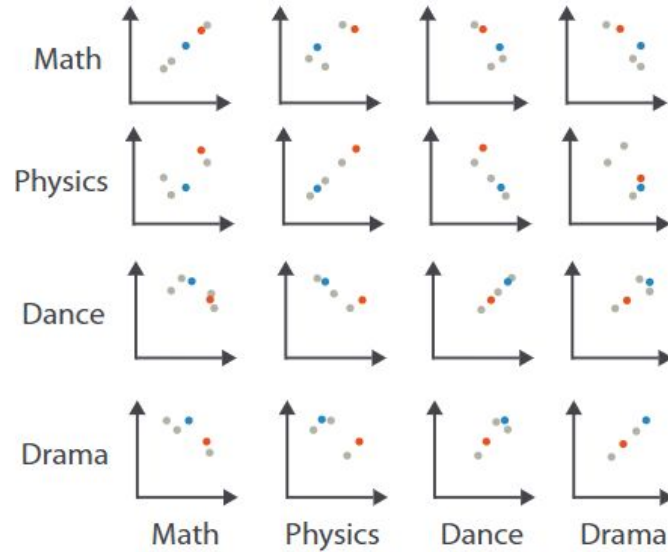
- Los ejes se ubican de manera paralela entre ellos
- En cada eje podemos codificar un atributo distinto

# Layout paralelo: ejemplo

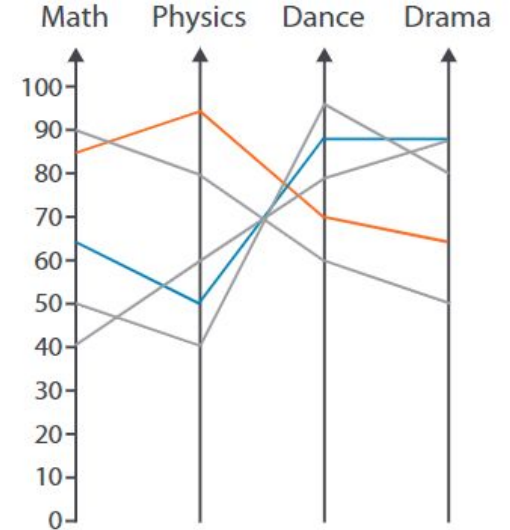
Table

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

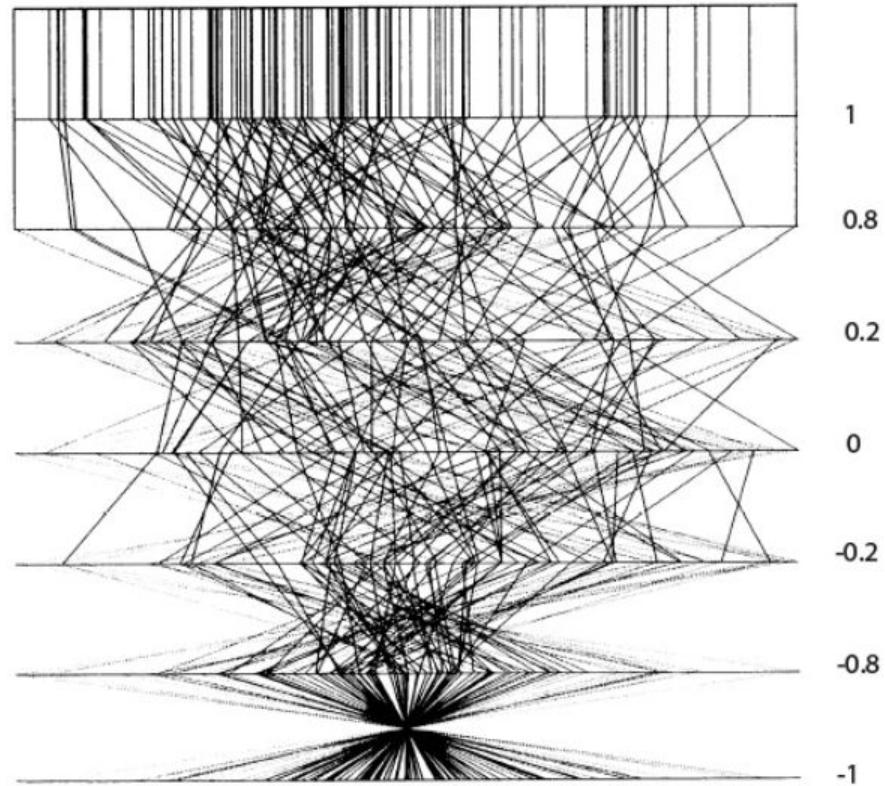
Scatterplot Matrix



Parallel Coordinates



## Layout paralelo: ejemplo para mostrar correlación

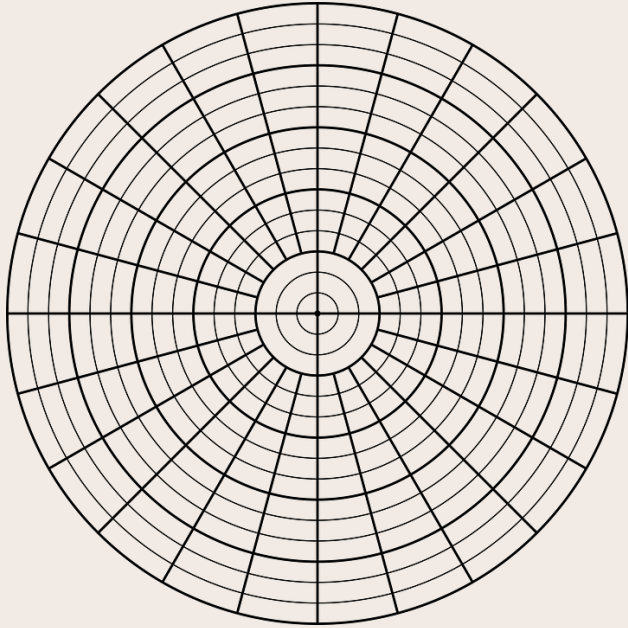


# Radial bar chart

---

Idiom	Coordenadas paralelas
<b>Qué:</b> datos	Tabla: muchos atributos
<b>Cómo:</b> codificación	Layout paralelo: el espacio horizontal se usa para separar los ejes, la posición horizontal para marcar los valores en cada eje con puntos y se conectan con líneas entre sí.
<b>Por qué:</b> tarea	Encontrar tendencias, outliers, extremos y correlaciones
<b>Escala</b>	Docenas de atributos, cientos de items

## Layout radial

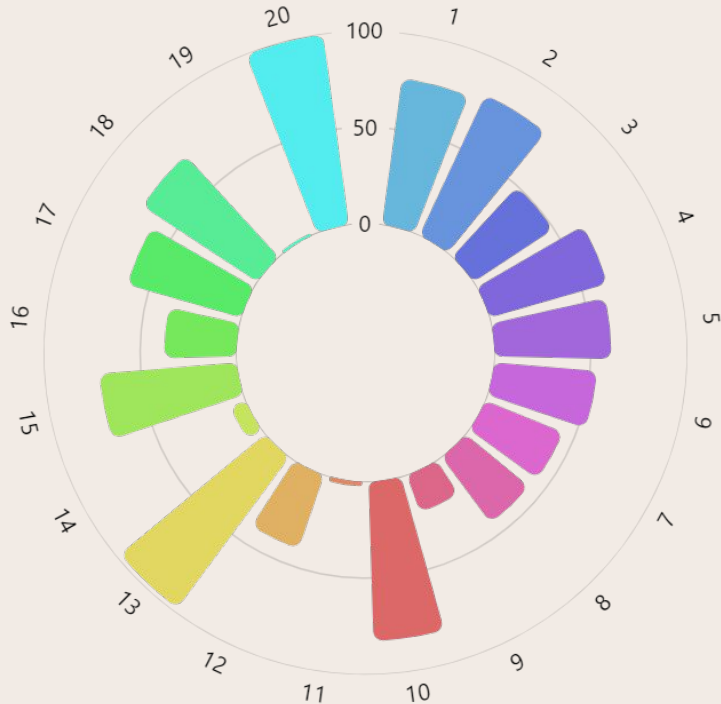


En el **layout radial** los ítems se distribuyen alrededor de un círculo usando el canal de **ángulo**, además de uno o más canales espaciales lineales.

- La manera más natural de usar un layout radial es con el sistema de coordenadas polares.

# Radial bar chart

---

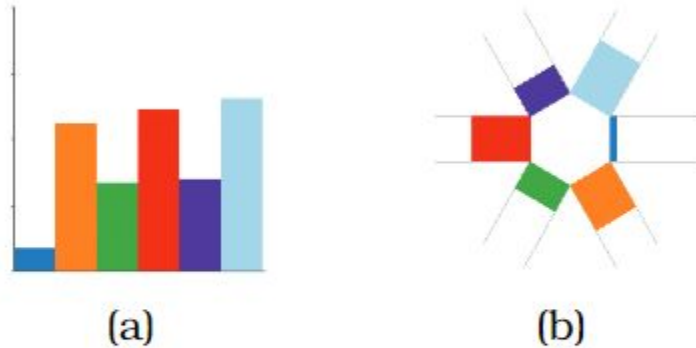


El radial bar chart es como un bar chart, usa líneas como marcas y como canal de magnitud usa el largo.

- sin embargo, en este caso las barras se ubican de manera circular
- no tenemos barras alineadas



## Radial bar chart

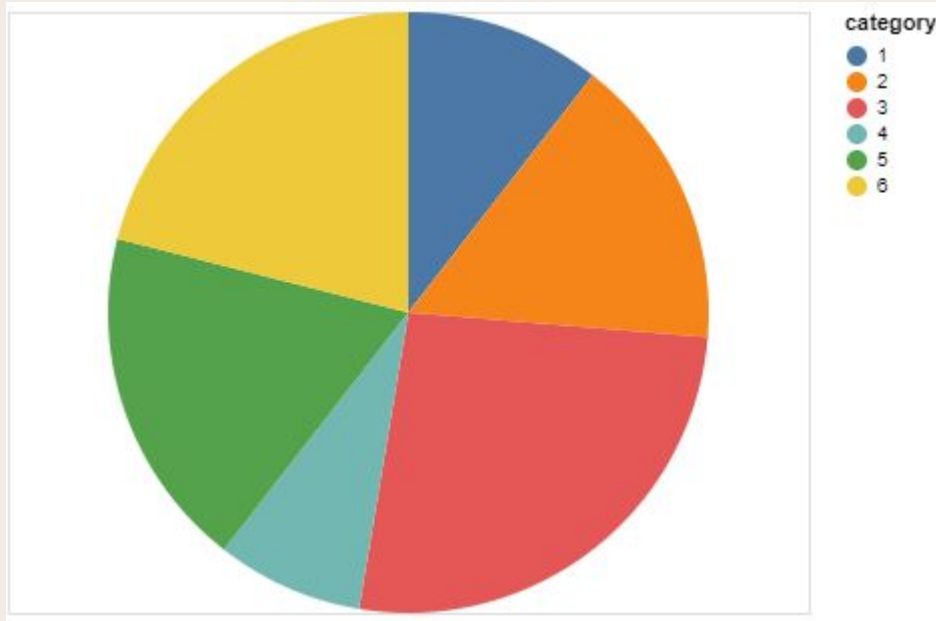


**Figure 7.16.** Radial versus rectilinear layouts. (a) Rectilinear bar chart. (b) Radial bar chart. After [Booshehrian et al. 11, Figure 4].

# Radial bar chart

Idiom	Radial bar chart
<b>Qué:</b> datos	Tabla: un atributo cuantitativo y uno categórico.
<b>Cómo:</b> codificación	Se usa líneas como marcas y el largo como canal de magnitud, se dispone esto en un layout radial
<b>Por qué:</b> tarea	Buscar y comparar valores
<b>Escala</b>	Decenas de ítems

# Pie chart



El gráfico más conocido de layout radial es el **pie chart**.

El pie chart codifica un único atributo con **marca de tipo área** y con **canal angulo**.

- Pese a su popularidad, los pie charts tienen problemas de efectividad al compararlos con otras alternativas.

# Pie chart

---

Idiom	Pie chart
<b>Qué:</b> datos	Tabla: un atributo cuantitativo y otro cualitativo
<b>Cómo:</b> codificación	Marcas de área y canal angular, layout radial
<b>Por qué:</b> tarea	Apreciar la relación parte-todo
<b>Escala</b>	A lo mas 12 categorías distintas

## Densidad espacial

El último aspecto que vamos a revisar en esta clase es el de la densidad espacial: el layout de nuestra visualización va a ser denso o *sparse*?

- En un **layout denso** usamos marcas pequeñas para poder ofrecer un overview de la mayor cantidad de ítems posibles. Un layout máximamente denso, usa marcas de tamaño 1 pixel
- Por otra parte, un **layout sparse** permite usar marcas más complejas y grandes. De por sí, cuando no hay tantos datos es mejor usar un layout sparse.

# Manipulación de vista



## ¿Cómo manipular vistas?

En muchas visualizaciones vamos a querer manipular las vistas, ya sea:

- cambiar algún aspecto de la visualización
- seleccionar ítems o atributos
- navegar hacia distintos POV

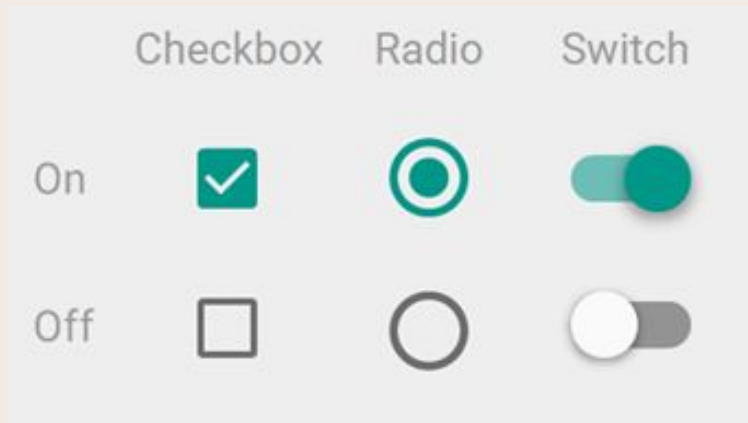
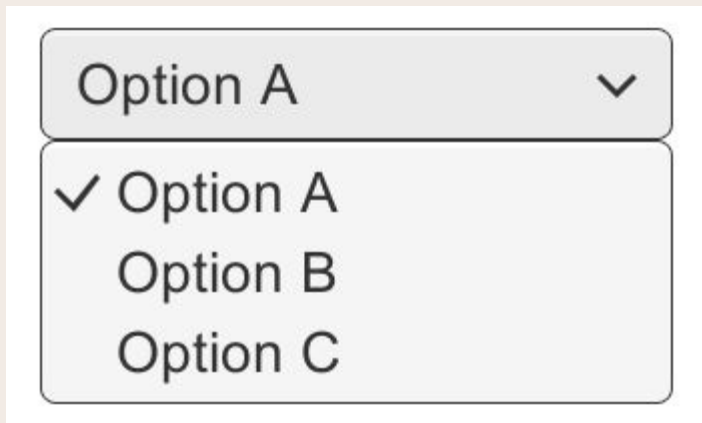
Esto incluye, cambios de idiomas, orden, arreglos espaciales, uso de nuevos canales visuales.



## ¿Cómo manipular vistas?

Normalmente un cambio va a requerir un **input** por parte del usuario:

El **cómo recibimos ese input** es un aspecto muy importante de una interfaz, pero es un aspecto que se estudia más en el diseño UX/UI de interfaces.





## ¿Por qué ir cambiando la vista?

Hay distintas razones por las cuales quisiéramos cambiar una vista

1. No podemos mostrar todos los datos “de una” porque son muchos
2. Los datos cambian en el tiempo
3. Tenemos un idiom de interactividad y las interacciones tienen impacto en lo que el usuario está viendo

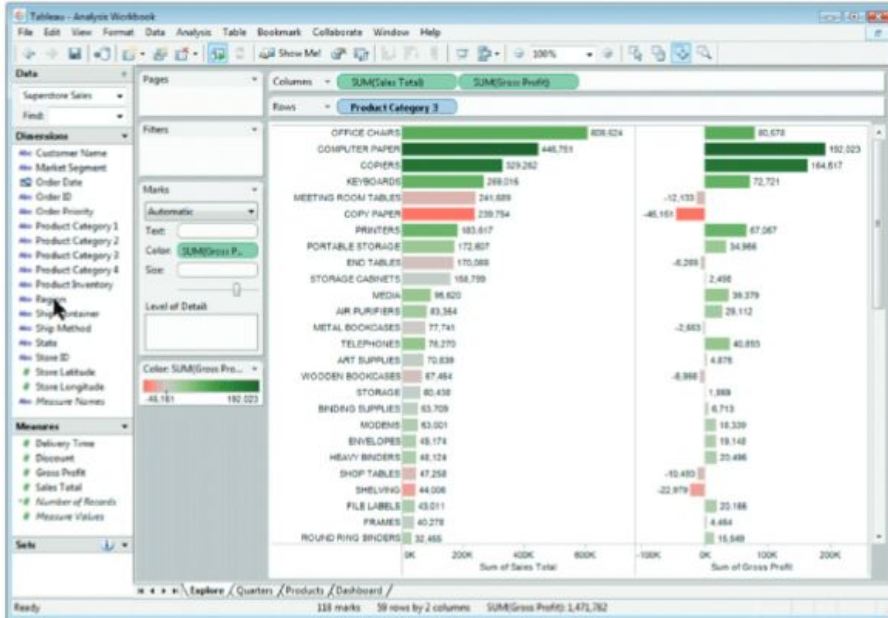


## Cambiar la vista en el tiempo

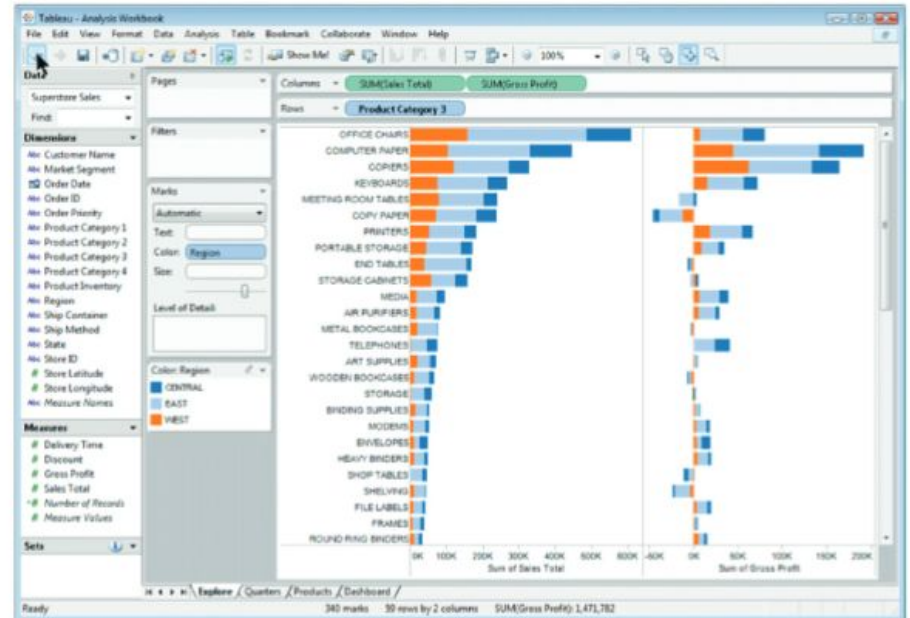
Las posibilidades de cambiar la vista pueden estar relacionadas a cualquier decisión del idiom:

- podemos cambiar el idiom: cambiar marcas, encodings, el orden, el punto de vista
- podemos filtrar atributos, niveles de agregación
- y en general cualquier decisión de diseño

# Cambiar la vista en el tiempo



(a)



(b)



## Cambiar la vista en el tiempo

Muchos cambios que podemos hacer implican re-arreglar los datos: algo muy típico que vemos es la posibilidad de ordenar una visualización. Por ejemplo: ordenar una tabla según un valor.

- Darle al usuario la posibilidad de reordenar datos le permite poder encontrar nuevos patrones en los datos.

## Selección de elementos

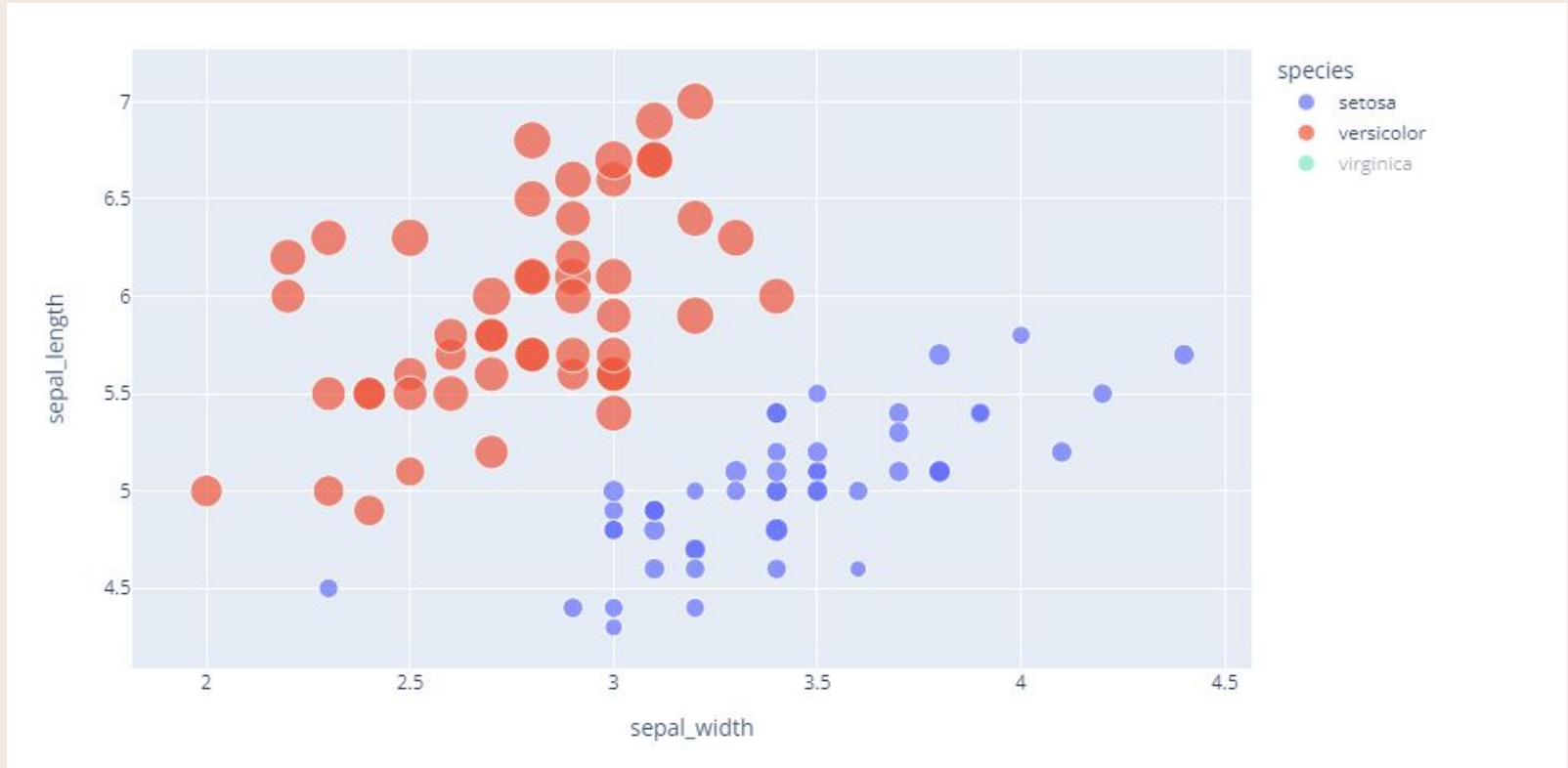
Permitir al usuario **seleccionar** uno o más elementos de interés en una visualización es una acción fundamental que es soportada por casi todos los sistemas interactivos.

- El output de una selección (es decir, los ítems seleccionados) son normalmente el input de una operación que se operará después.



# Selección de elementos

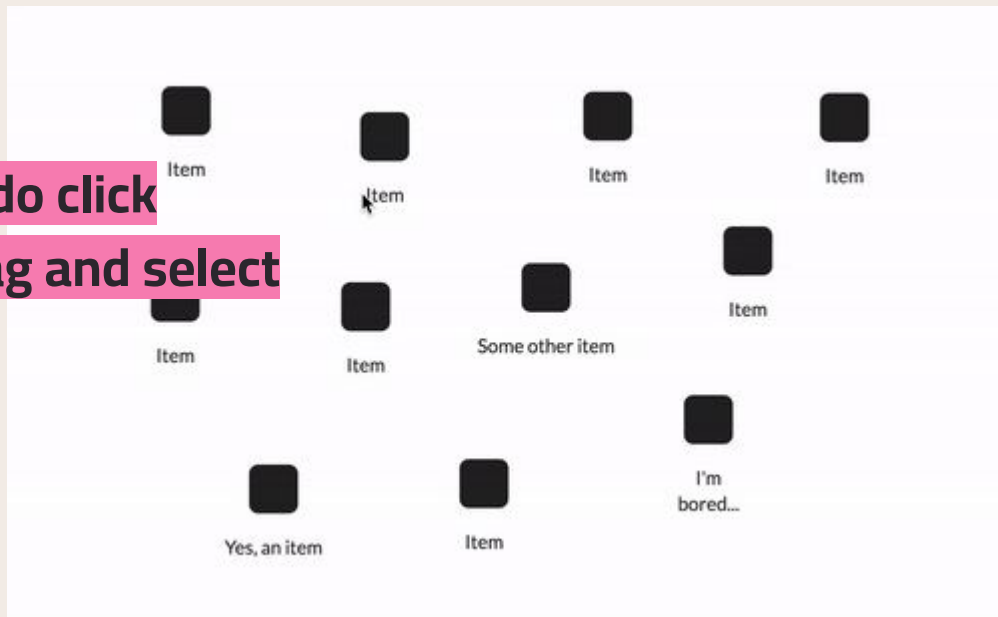
---



# Selección de elementos: elecciones de diseño

- Lo más básico es que cada elemento pueda estar seleccionado o no seleccionado.

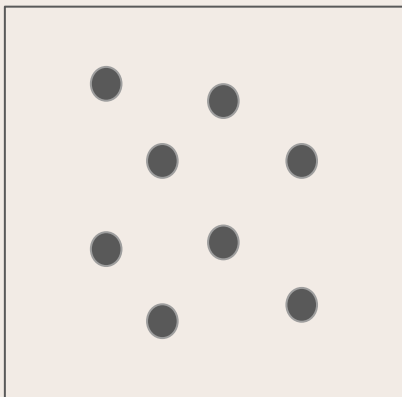
- Seleccionar haciendo click
- Seleccionar con drag and select





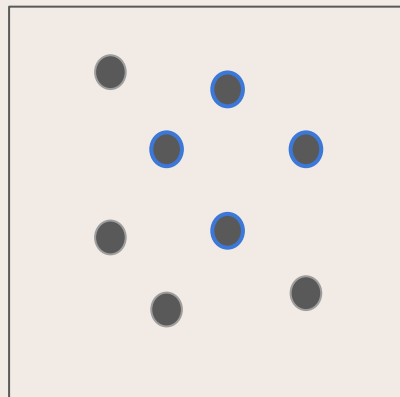
## Selección de elementos: highlighting

El usuario puede querer seleccionar los datos para **ubicarlos de mejor manera en la visualización**: para esto sirve el **highlighting**



A

B



A

B

## Selección de elementos: highlighting

Lo conversamos antes, pero es bueno recalcarlo:

**Las acciones del usuario deben tener feedback inmediato: en el caso de la selección ese feedback puede ser el highlighting.**

*Algo eficiente sería generar efecto pop-out (clase 2)*

## Selección de elementos y highlighting

**Si bien relacionados, la elección de interacción (como el usuario selecciona los ítems) versus el efecto de la interacción (el highlighting) no son dependientes entre sí, es decir, son dos elecciones de diseño independientes**

## Selección como input de otra acción

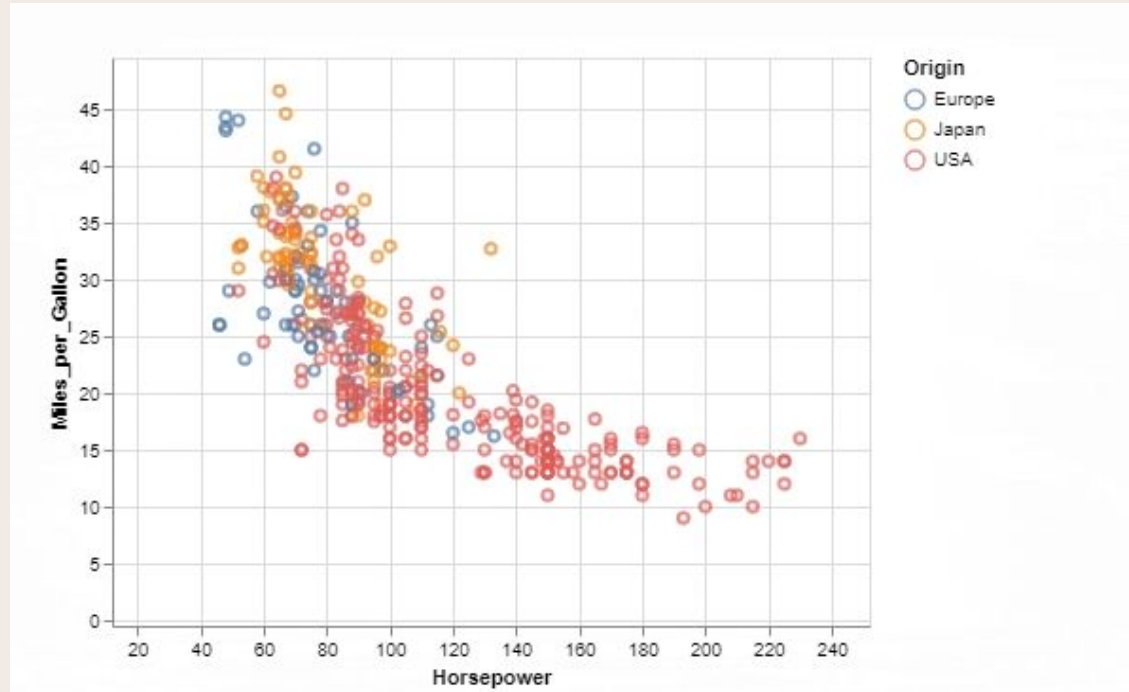
El highlight es el output inmediato de la selección, pero a veces podemos querer hacer otras cosas:

- Seleccionar ítems para hacer algo con ellos: por ejemplo, crear un gráfico solo con los items seleccionados

# Navegar

---

Además de seleccionar, podemos **navegar a través de la visualización**






# Navegar

---

- **Navegación**

Cambiar el punto de vista desde el cual las cosas se ven. Cuando el punto de vista cambia, también cambia el conjunto de elementos visibles para el usuario.

## Tres tipos de navegación




-  **Zooming**
-  **Panning/Translating**
-  **Rotating**

# Navegar

---

- Navegación

## Tres tipos de navegación

-  Zooming
-  Panning/Translating
-  Rotating

*La navegación se puede mezclar con filtros y agregación*

# Geometric y semantic zooming

## -Geometric zooming-

Forma intuitiva de navegación, que emula cómo nos movemos IRL

- En 2D es cómo acercar o alejar una hoja en frente de nuestros ojos
- La apariencia de los objetos se mantiene fija

## -Semantic zooming-

En este caso la forma de los objetos se adapta al movimiento

- la apariencia de los objetos se ajusta al número de píxeles disponibles
- los textos pueden ir desapareciendo
- las marcas pueden cambiar



## Navegación limitada

La navegación limitada es lo contrario a la **navegación ilimitada (poder moverse infinitamente hacia cualquier lado)**

- En las visualizaciones, sin embargo, tener navegación ilimitada puede hacer que el usuario se pierda y no encuentre el POV que necesita

**Una aplicación de navegación limitada en 2D es tener zooming hasta ciertos puntos: un mínimo y un máximo.**



## Reducción de atributos

La navegación en visualizaciones comparte fundamentos con los actos de reducir atributos: slice, cut and project

- **Slice**  
Seleccionamos algún atributo y valor y eliminamos los items que hacen match con esa seleccion
- **Cut**  
Seleccionar una porción de un plano y dejar de ver lo que no esta en la dicha porción
- **Project**  
Seleccionamos un atributo y lo dejamos de mostrar, de todos modos, seguimos mostrando todos los items

# Control #1:

# Visualizando datos tabulares



# Visualización de información y analítica visual

**Clase 5:** Datos tabulares y manipulación de vistas

