

# Visualización de Datos



UNIVERSIDAD DE  
**MURCIA**

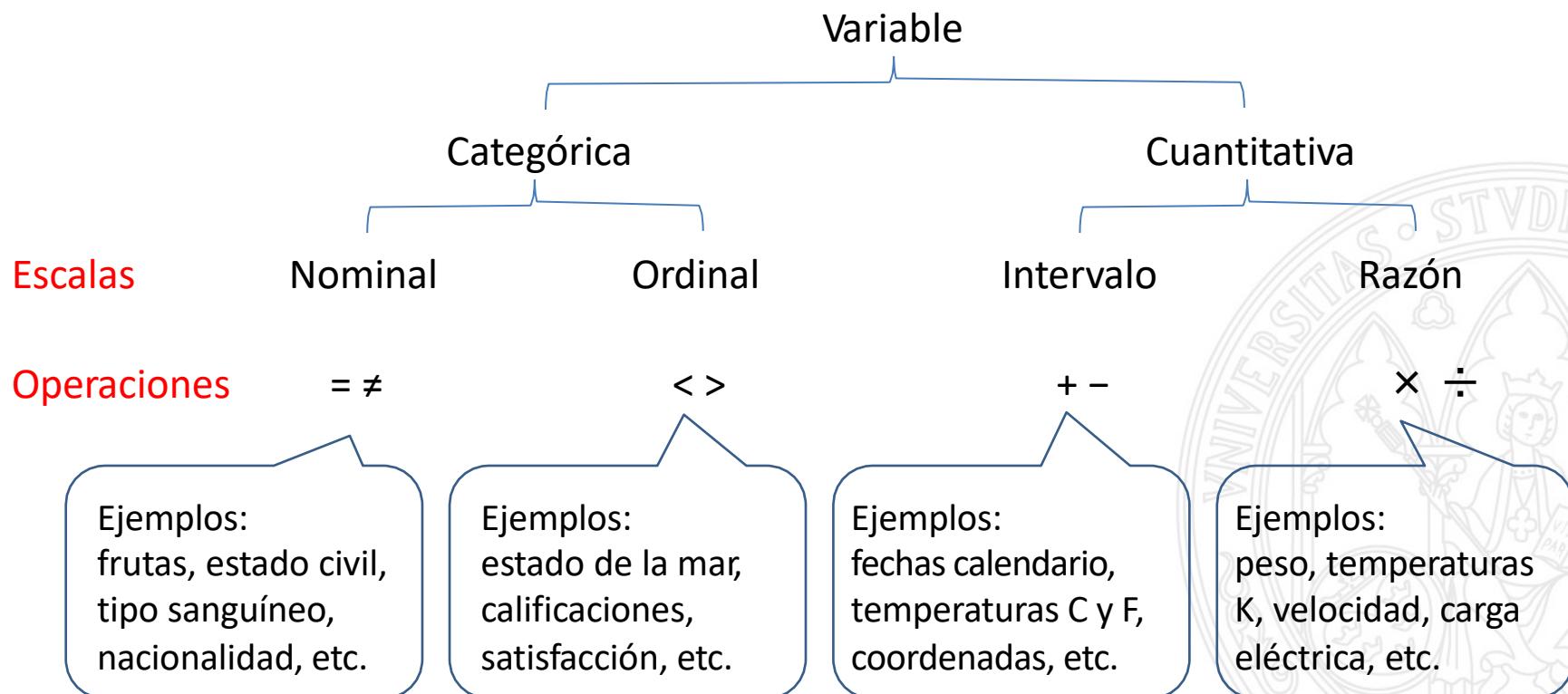
# Tema 2. Básicos

“The purpose of visualization is insight, not pictures”  
— Ben Shneiderman

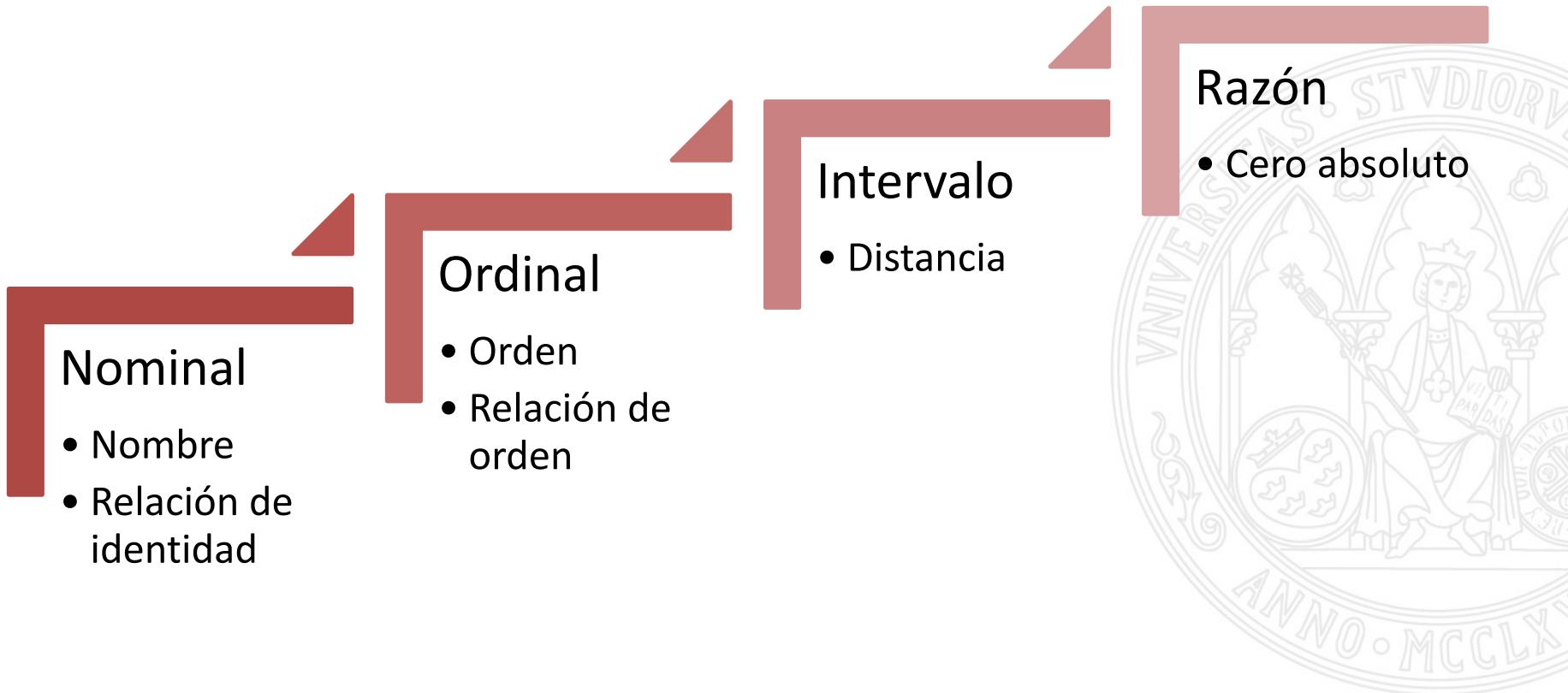
The screenshot shows a web page titled "Treemaps for space-constrained visualization of hierarchies". The page includes the HCIL logo, a navigation bar with links to the HCIL website and current HCIL news, and a main content area with a treemap visualization of a directory tree. The treemap consists of various colored rectangles of different sizes, representing file structures. Below the visualization, there is descriptive text about the history of treemap research at the University of Maryland, including details about the first technical report and the development of the algorithm.



# Tipos de medidas (Stevens, 1946)



Podemos realizar conversiones entre escalas; por ejemplo, podemos convertir un intervalo en un ordinal o en un nominal:  $80^{\circ}\text{C} \rightarrow$  muy caliente  $\rightarrow$  quemado



- Mackinlay, J. (1986). Automating the design of graphical presentations of relational information. ACM Trans. Graph. 5(2), 110–141. <https://doi.org/10.1145/22949.22950>

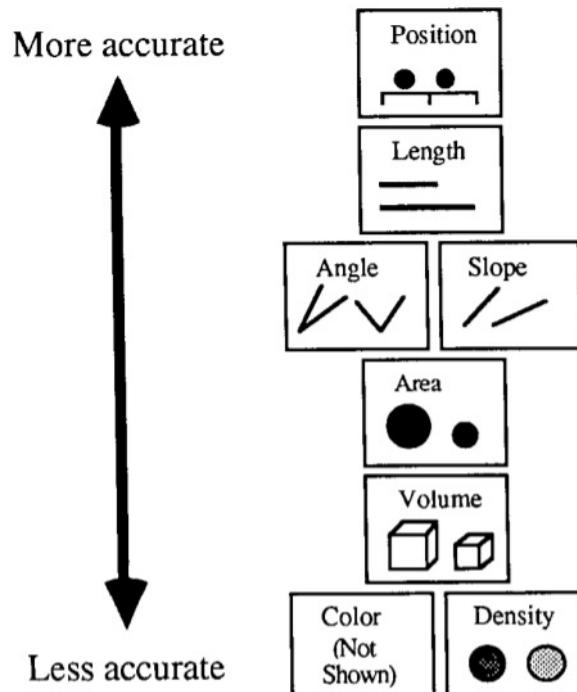


Fig. 14. Accuracy ranking of quantitative perceptual tasks. Higher tasks are accomplished more accurately than lower tasks. Cleveland and McGill empirically verified the basic properties of this ranking.

- Mackinlay, J. (1986). Automating the design of graphical presentations of relational information. ACM Trans. Graph. 5(2), 110–141. <https://doi.org/10.1145/22949.22950>

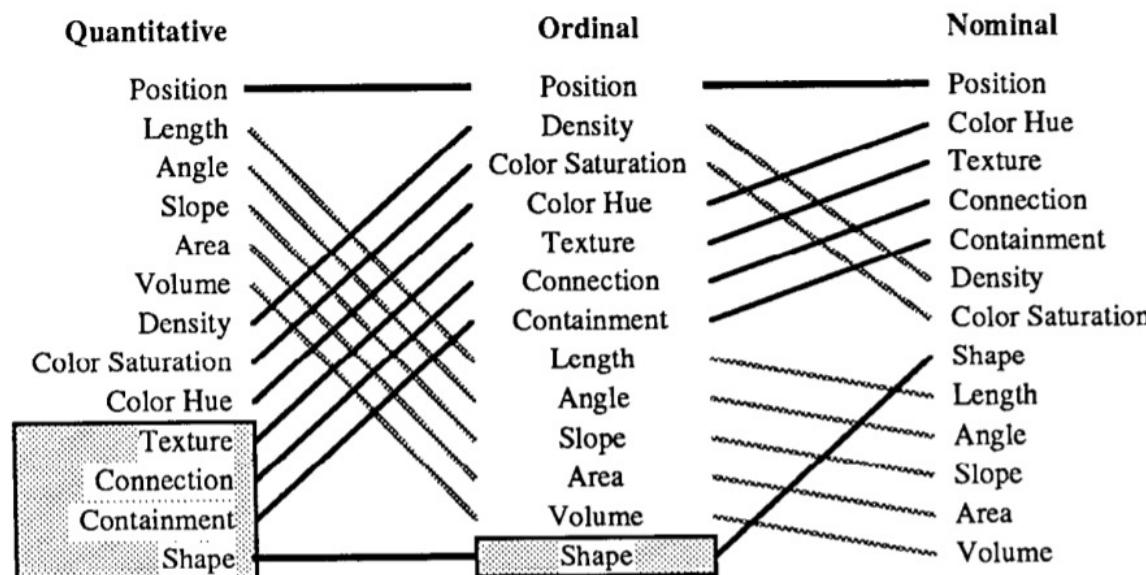
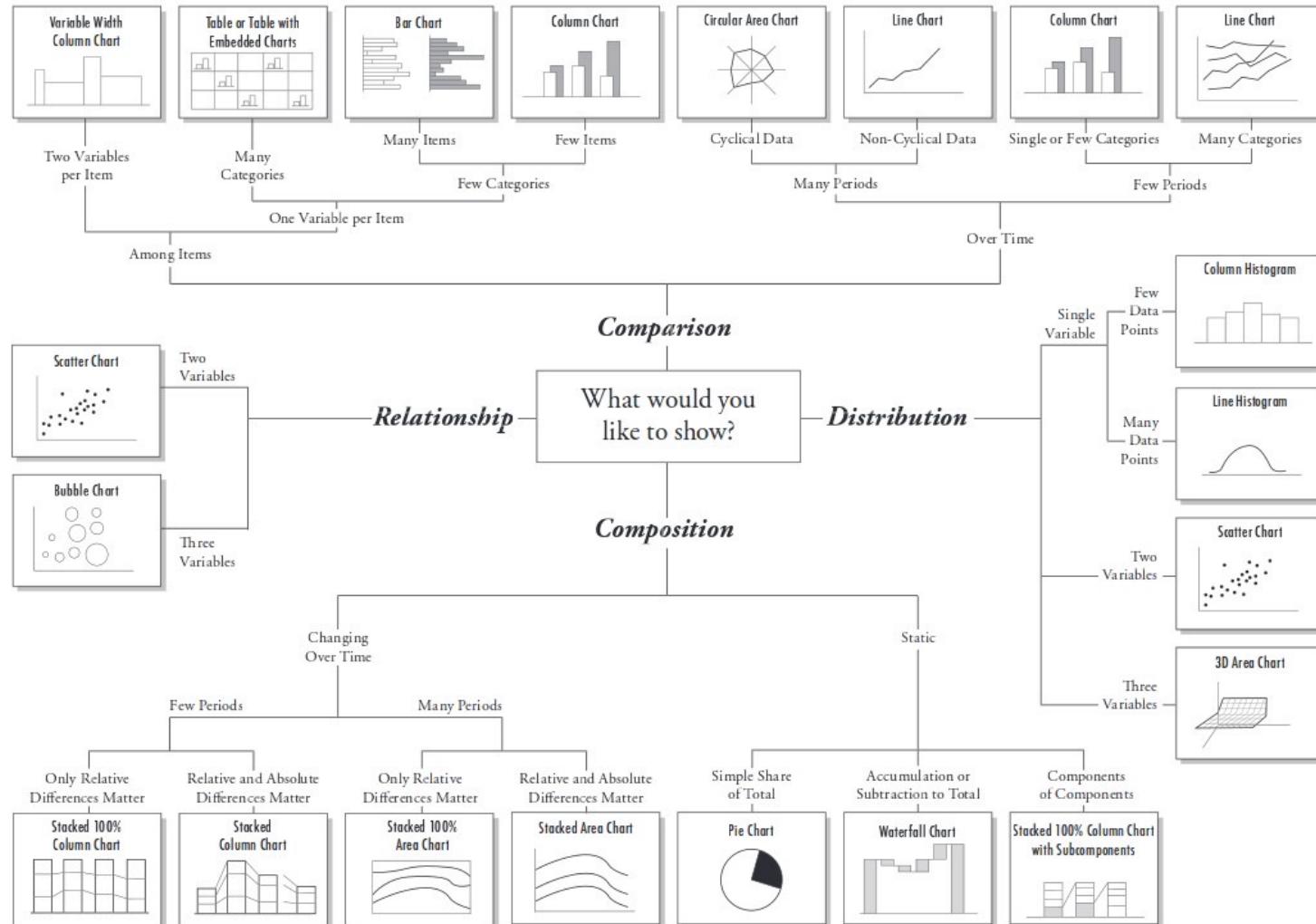


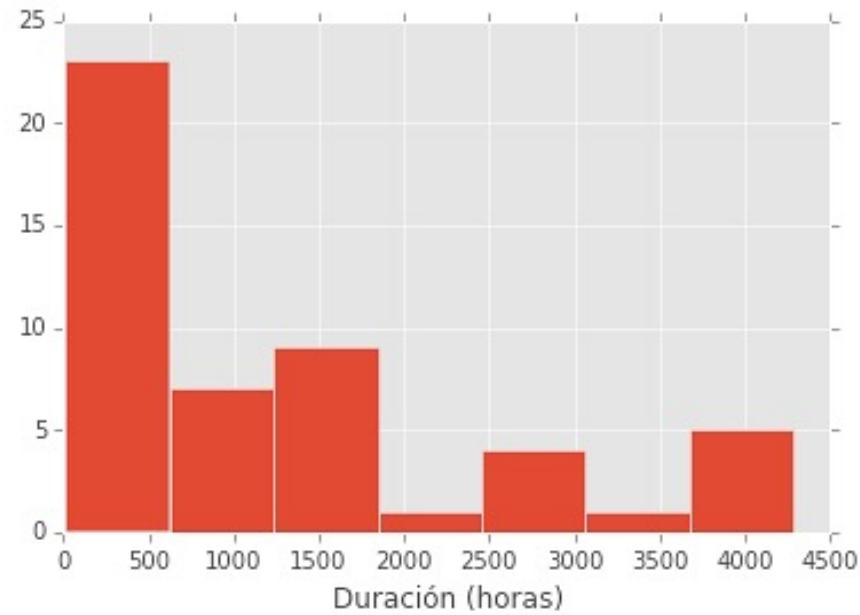
Fig. 15. Ranking of perceptual tasks. The tasks shown in the gray boxes are not relevant to these types of data.

# ¿Qué tipo de gráfico escoger?



- Es una gráfica de barras verticales. Cada dato es parte de una sola barra, a la que también llamamos **categoría**
- La anchura es habitualmente constante, y la altura es **proporcional a la frecuencia**, esto es, al número de datos en el intervalo. Si normalizamos a 1, obtenemos la **frecuencia relativa**
- Un histograma es una representación de la **distribución de probabilidad** de una variable discreta. No es raro confundirlo con un diagrama de barras

## Histograma (2/3)

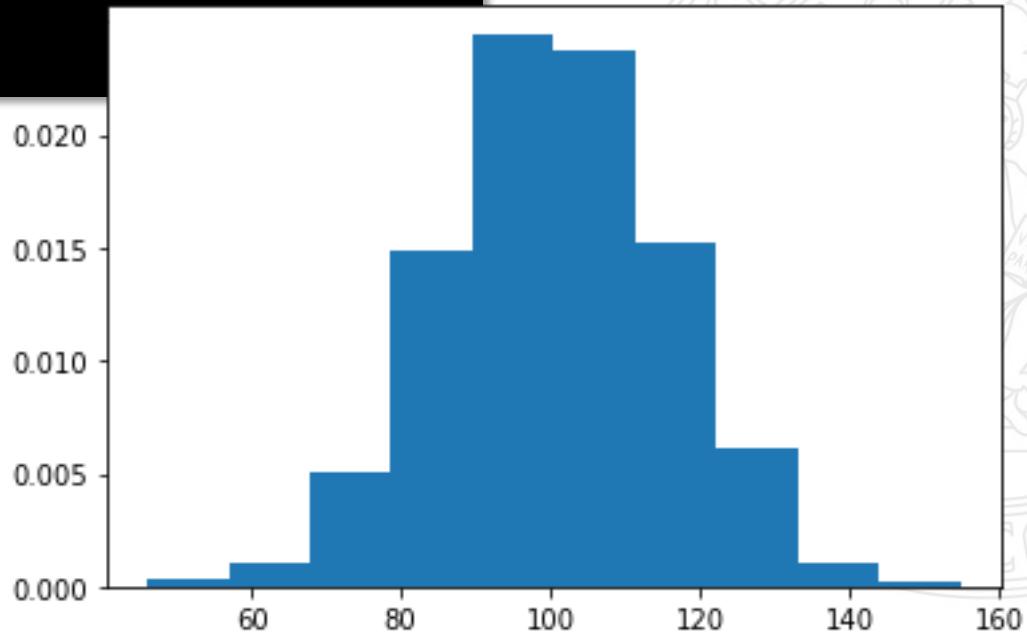


## Histograma (3/3)

```
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline

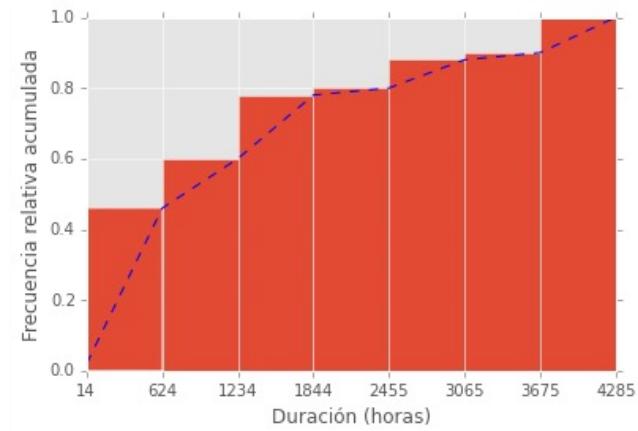
mu, sigma = 100, 15 # media y desviación estándar
x = np.random.normal(mu, sigma, 1000)

plt.hist(x, density=True)
```



# Gráfica de distribución acumulada

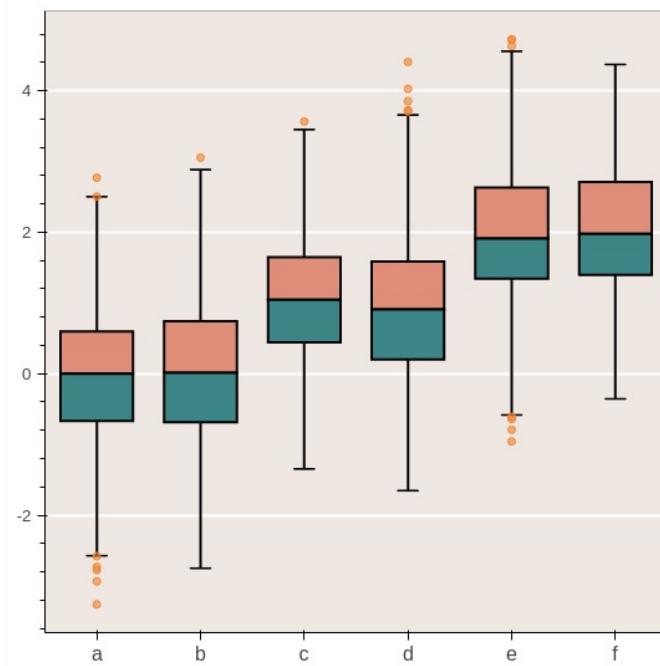
- Para cada categoría del histograma podemos calcular su **frecuencia relativa**, como la fracción de datos representada respecto al total
- Las gráficas de distribución acumulada se obtienen a partir del histograma, donde cada barra acumula el porcentaje de datos correspondiente a **cada categoría y a las anteriores**
- Permite responder a preguntas como:  
*¿Cuál es el porcentaje aproximado de baterías que fallará durante las primeras 1500 horas de operación?*  
*¿Qué representa una frecuencia relativa de 0.5?*



- Medidas como la **media, la varianza y la desviación típica** se utilizan habitualmente para representar los datos, pero sufren una **distorsión** por la presencia de valores atípicos
- Se proponen otros **estadísticos**, similares a los primeros, que sean **robustos** respecto a estos problemas
- La **mediana** es una medida robusta de **tendencia central**. Si los datos están ordenados de menor a mayor, la mediana es la observación central (número impar de datos) o el promedio de los dos datos centrales (número par de datos)
- Algunas medidas robustas de **dispersión** son los **cuartiles** (dividen la muestra en 4 partes iguales), los **deciles** (10 partes iguales) y los **percentiles** (100 partes iguales)

## Diagrama de caja y bigotes (2/3)

- El **primer y el tercer cuartil** ( $Q_1$  y  $Q_3$ ) delimitan la caja central, cuya longitud recibe el nombre de **rango intercuartílico** ( $RI = Q_3 - Q_1$ )
- Los **límites inferior y superior**, se calculan como:
  - $LI = \max\{\min\{x_i\}, Q_1 - 1.5 \cdot RI\}$
  - $LS = \min\{\max\{x_i\}, Q_3 + 1.5 \cdot RI\}$
- La **mediana** ( $Q_2$ ) se representa mediante una línea que cruza la caja central

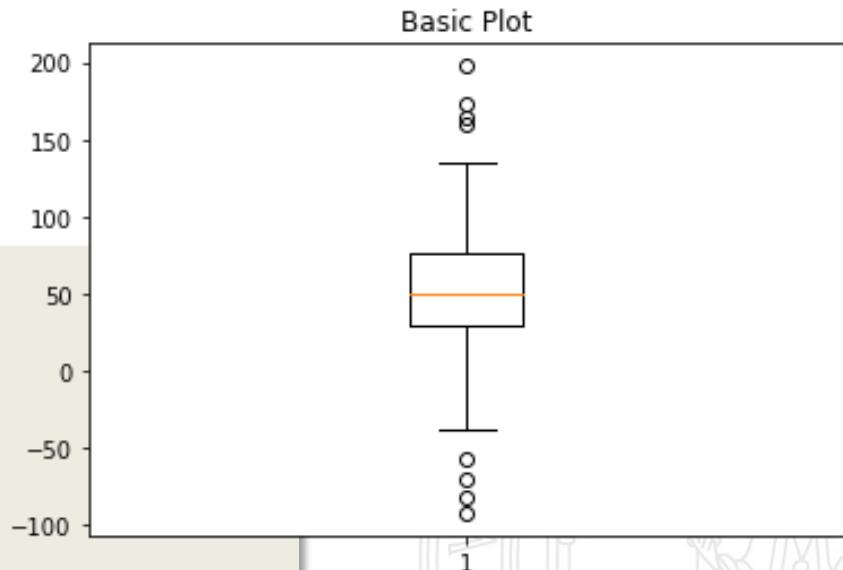


## Diagrama de caja y bigotes (3/3)

```
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline

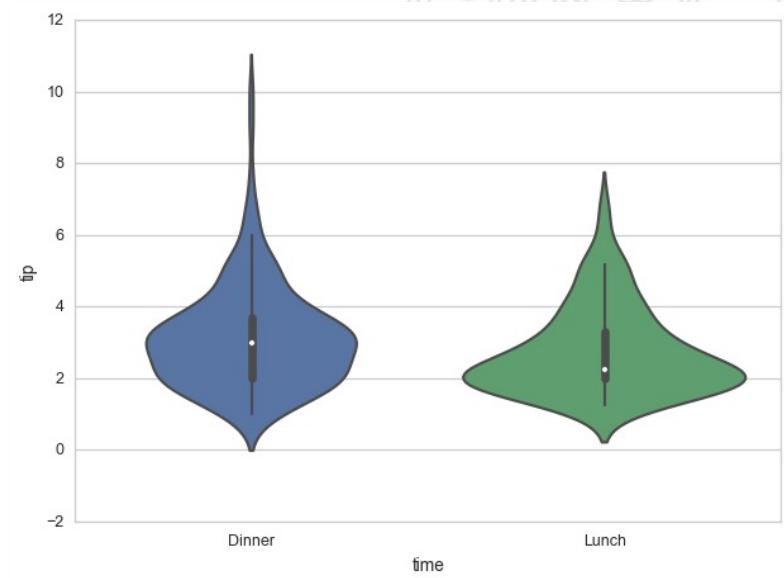
spread = np.random.rand(50) * 100
center = np.ones(25) * 50
flier_high = np.random.rand(10) * 100 + 100
flier_low = np.random.rand(10) * -100
data = np.concatenate((spread, center, flier_high,
                      flier_low))

fig1, ax1 = plt.subplots()
ax1.set_title('Basic Plot')
ax1.boxplot(data)
```



## Gráfico de violín (1/2)

- Son una extensión de los diagramas de caja y bigotes que muestran además la densidad de probabilidad de cada variable aleatoria contemplada en el experimento
- La densidad de probabilidad se puede simplificar mediante su **histograma** o se puede estimar mediante algún **método no paramétrico basado en núcleos** (*kernel density estimation*), funciones cuya integral es uno y su media es cero



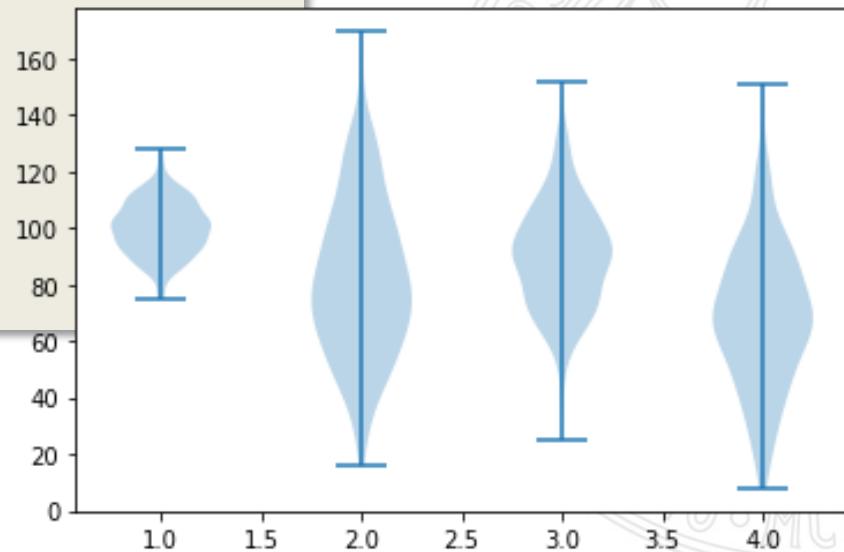
## Gráfico de violín (2/2)

```
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline

col_1 = np.random.normal(100, 10, 200)
col_2 = np.random.normal(80, 30, 200)
col_3 = np.random.normal(90, 20, 200)
col_4 = np.random.normal(70, 25, 200)

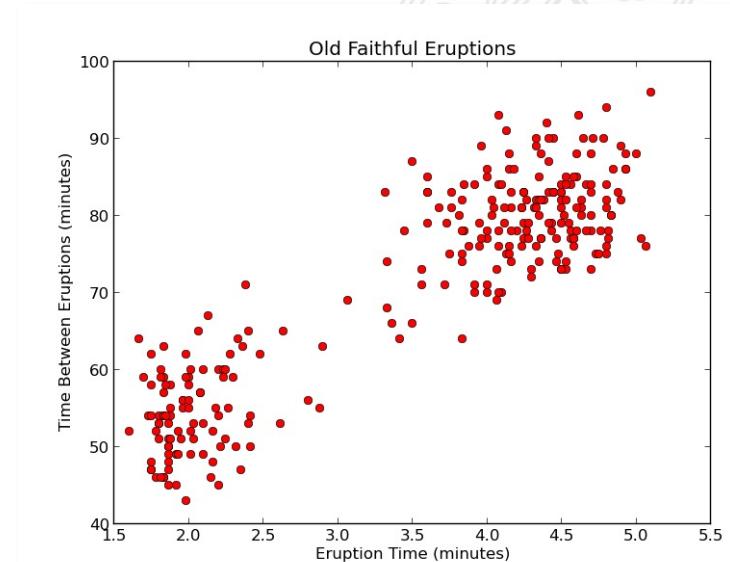
# Combinar las colecciones de datos
data = [col_1, col_2, col_3, col_4]

plt.violinplot(data)
```

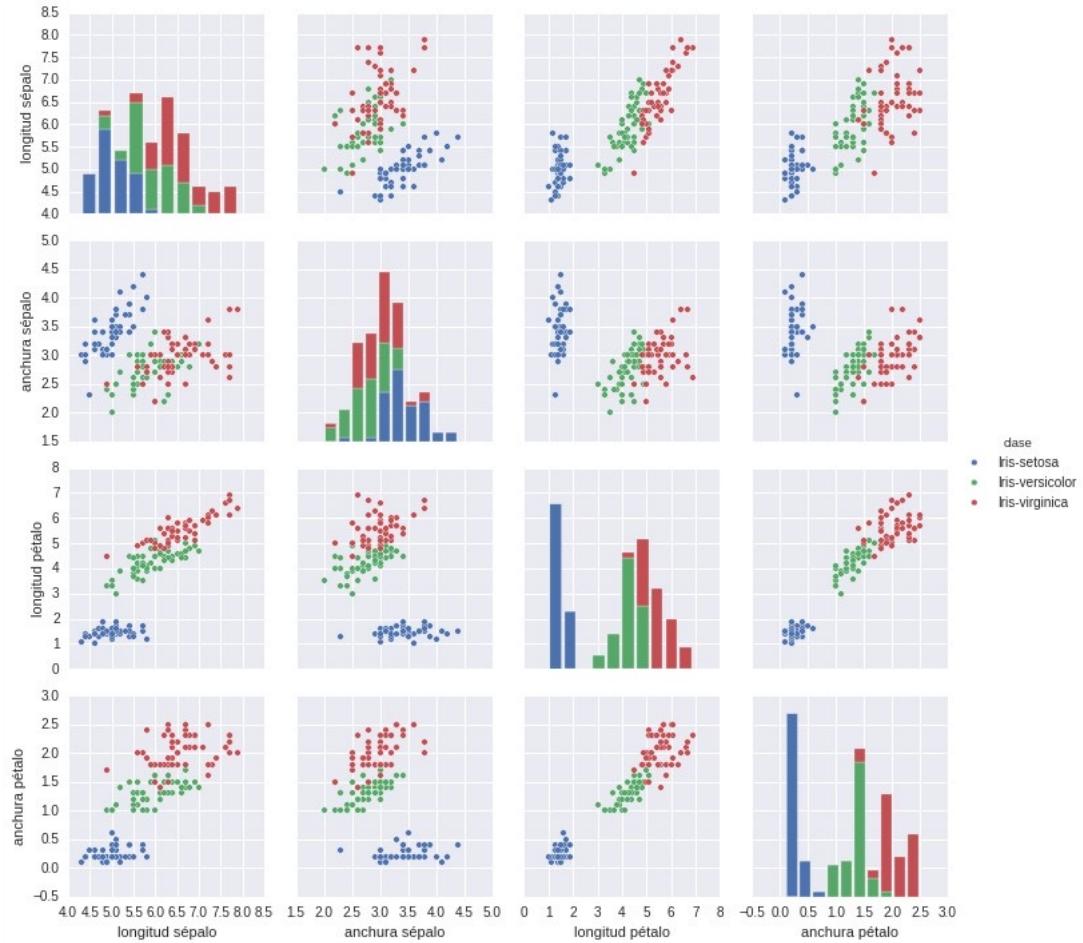


## Diagrama de dispersión (1/4)

- Permiten conocer la **distribución** de valores de los datos en espacios de múltiples dimensiones
- Representan por parejas los distintos atributos de los datos
- Son especialmente útiles como exploración previa en problemas de **clasificación** o **agrupamiento**



# Diagrama de dispersión (2/4)



# Diagrama de dispersión (3/4)

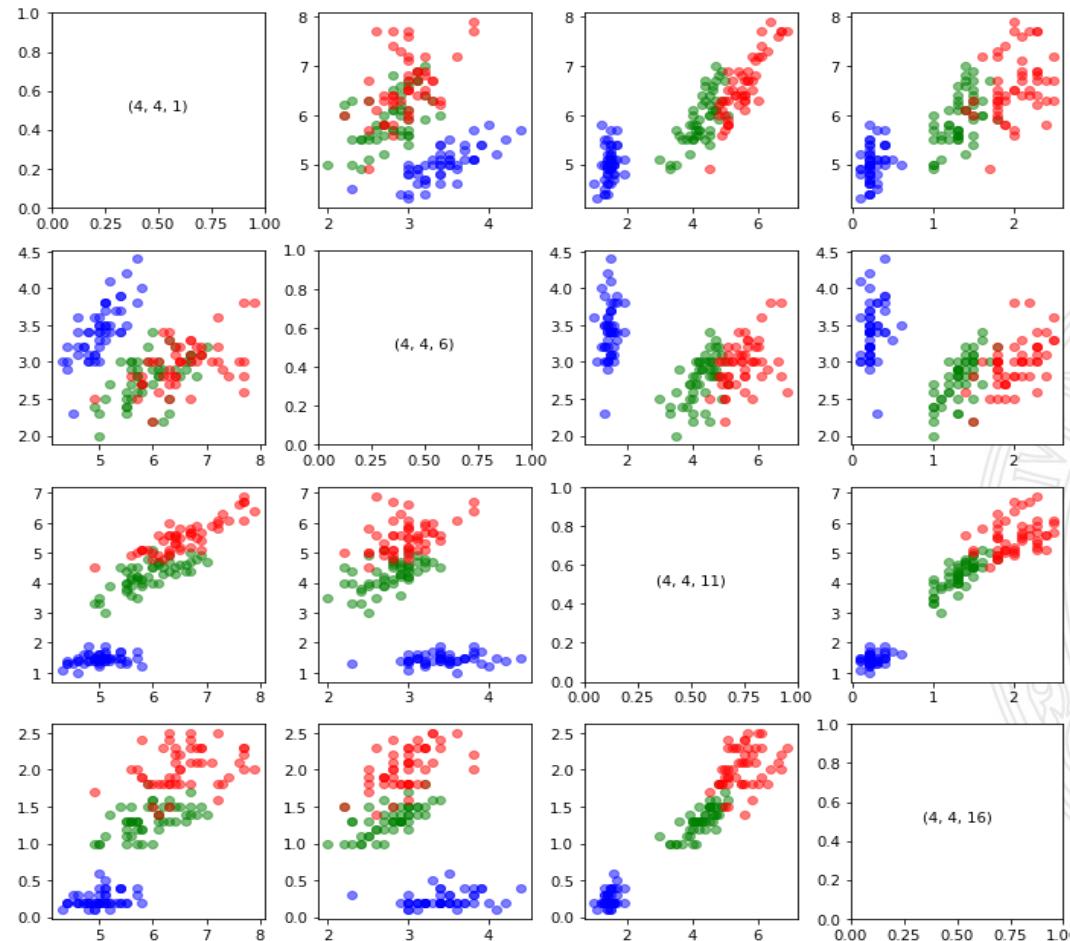
```
from sklearn import datasets
import matplotlib.pyplot as plt
import pandas as pd
%matplotlib inline

data = datasets.load_iris()
df = pd.DataFrame(data['data'], columns=data['feature_names'])
df['target'] = data['target']

colours = ['blue', 'green', 'red']
species = data.target_names

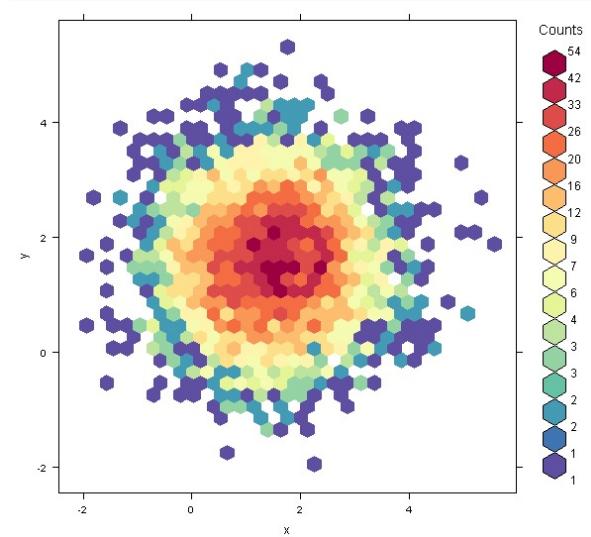
plt.figure(figsize=(10,10))
for i in range(1, 17):
    plt.subplot(4, 4, i)
    if ((i-1)%4 == int((i-1)/4)):
        plt.text(0.5, 0.5, str((4, 4, i)), ha='center')
    else:
        for k in range(3): # Hay tres especies
            species_df = df[df['target'] == k]
            plt.scatter(species_df.iloc[:, (i-1)%4], species_df.iloc[:, int((i-1)/4)],
                        color=colours[k], alpha=0.5, label=species[k])
            #plt.legend(loc = 'lower right')
plt.tight_layout()
```

# Diagrama de dispersión (4/4)



# Gráfico de agrupamiento hexagonal (1/2)

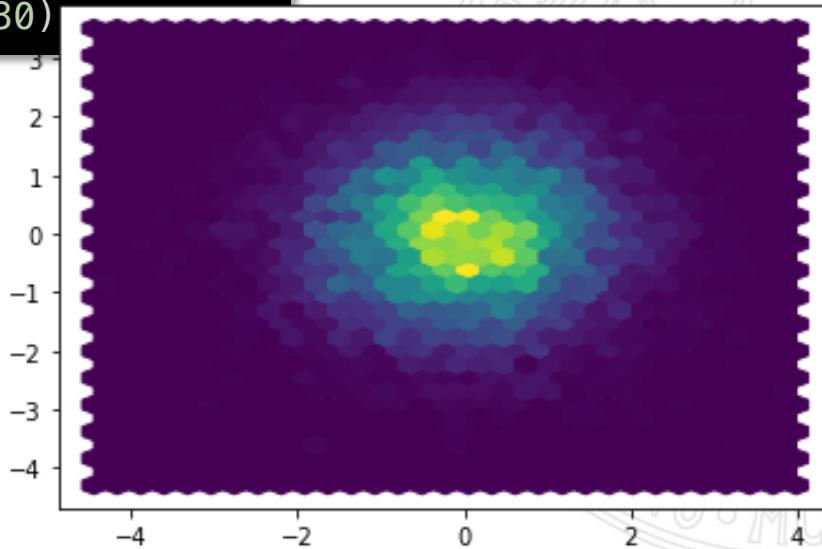
- Son una forma de histograma en dos dimensiones donde se realiza un conteo del número de muestras que se encuentran en cada uno de los hexágonos que permite teselar el plano
- Suele ser más informativo que un diagrama de dispersión



## Gráfico de agrupamiento hexagonal (2/2)

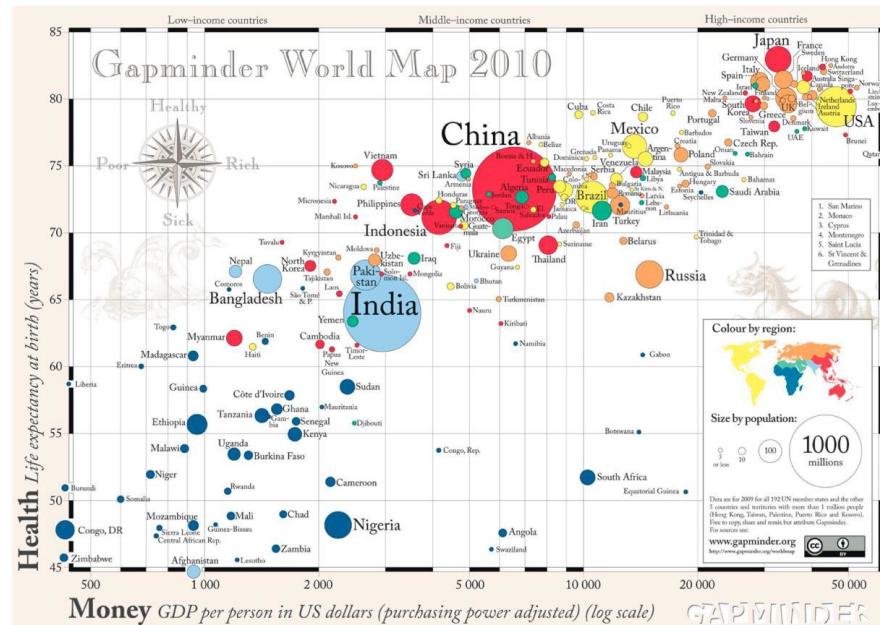
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
%matplotlib inline

df = pd.DataFrame({'x': np.random.randn(10000),
                   'y': np.random.randn(10000)})
plt.hexbin(df['x'], df['y'], gridsize=30)
```



# Gráfico de burbujas (1/2)

- Son un tipo de gráficos de **dispersión** en los que aparece una tercera componente de los datos, en forma de **radio de una burbuja** centrada en las coordenadas de las dos primeras componentes del diagrama de dispersión



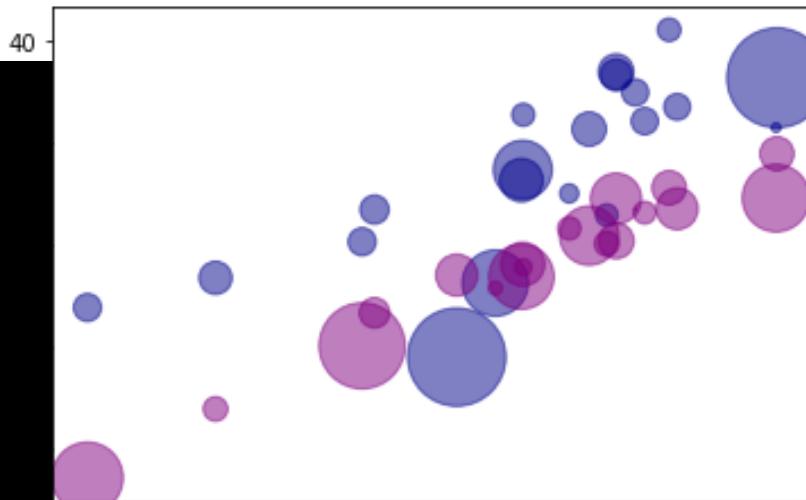
## Gráfico de burbujas (2/2)

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
%matplotlib inline

x = np.random.normal(8, 10, 20)
y = x + np.random.normal(15, 10, 20)
z = x - np.random.normal(10, 5, 20)

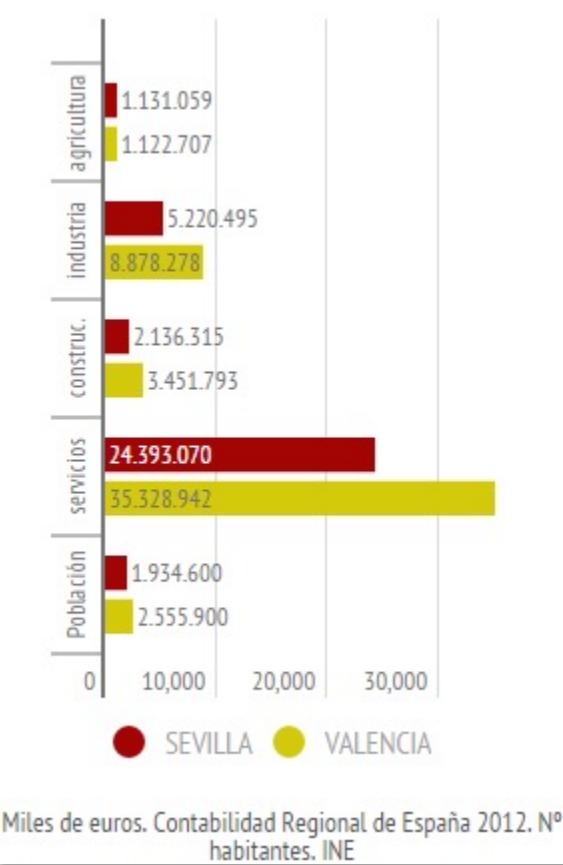
area_y = np.random.exponential(np.random.rand() * 1000, 20)
area_z = np.random.exponential(np.random.rand() * 1000, 20)

plt.scatter(x, y, color='darkblue', alpha=0.5, s = area_y)
plt.scatter(x, z, color='purple', alpha=0.5, s = area_z)
```



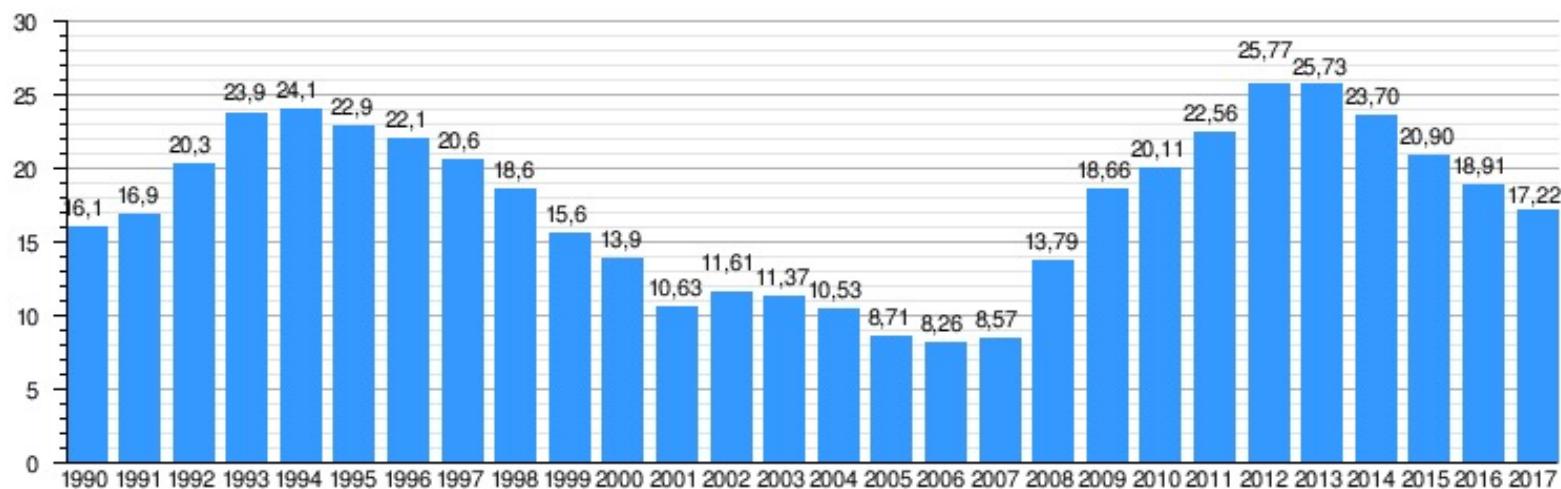
## Gráfico de barras (1/3)

- Permiten **comparar** múltiples entidades entre sí o múltiples categorías



## Gráfico de barras (2/3)

- A veces se usa para representar datos **a lo largo del tiempo**
- Se busca realizar **comparaciones** entre variables tomando como base de representación el **tiempo**

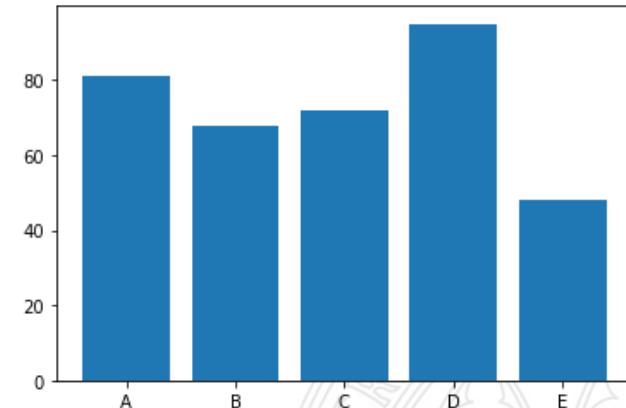


## Gráfico de barras (3/3)

```
import matplotlib.pyplot as plt
%matplotlib inline

weight = [81, 68, 72, 95, 48]
name = ('A', 'B', 'C', 'D', 'E')

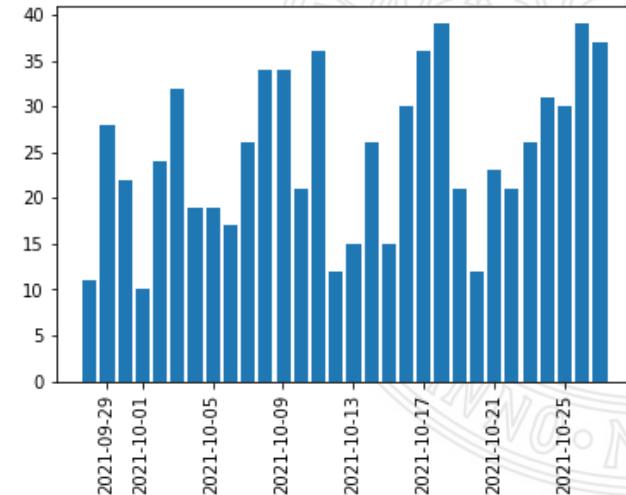
plt.bar(name, weight)
```



```
import datetime as dt
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
%matplotlib inline

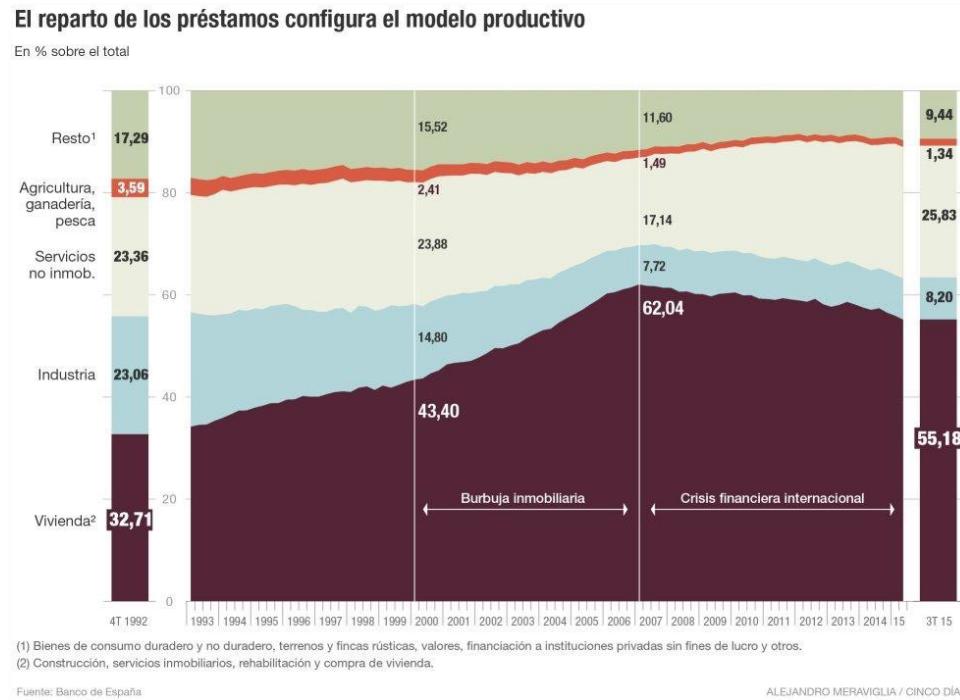
datelist = pd.date_range(dt.datetime.today(),
    periods=30).tolist()
values = np.random.randint(10, high=40,
size=30)

plt.bar(datelist, values)
plt.xticks(rotation=90)
```

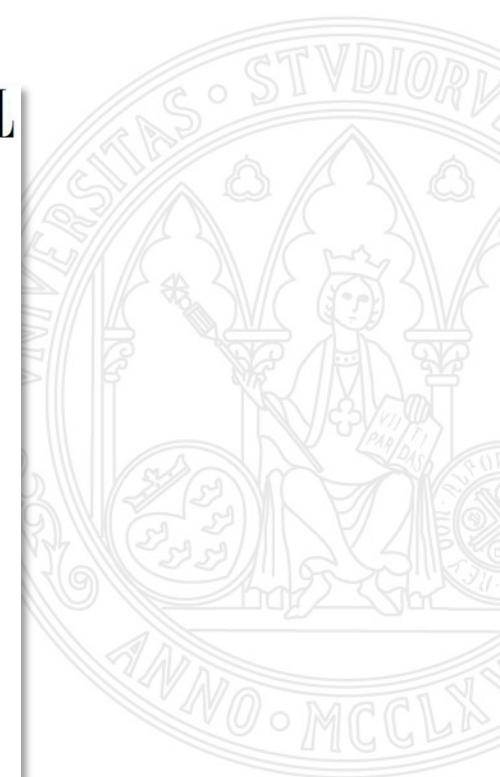
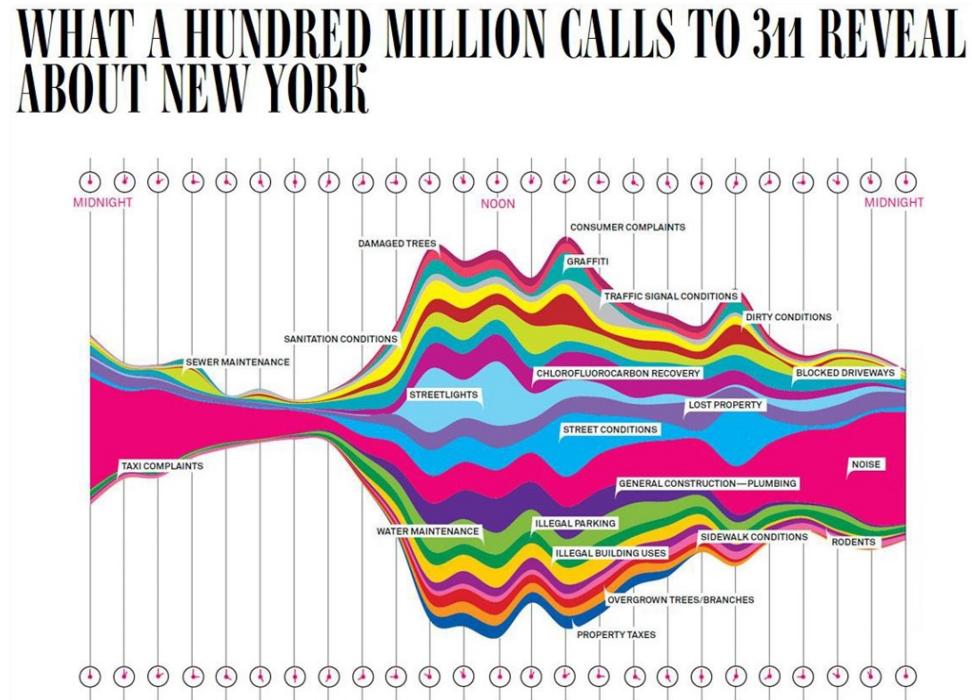


# Gráfico de área

- A veces es particularmente interesante **apilar los valores de las categorías**, cuando interesa mostrar **incrementos**, o cuando representan una **parte de un total**

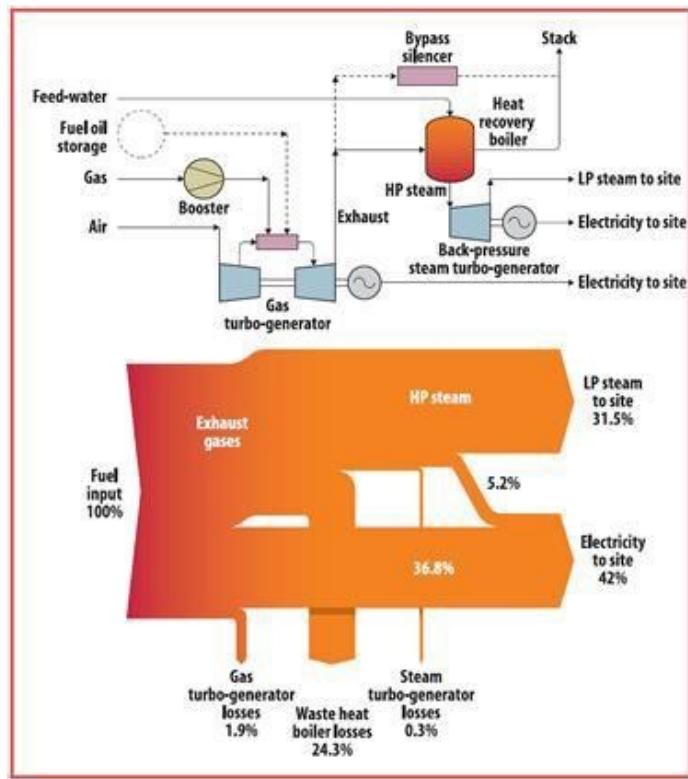


- Se construye partir de los gráficos de barras o de líneas apilados, donde la **información está contenida en el área** y se **centra la figura en un eje horizontal**



## Gráfico Sankey (1/2)

- Es un tipo de **diagrama de flujo** donde el **ancho de las flechas** es **proporcional al tamaño del flujo**

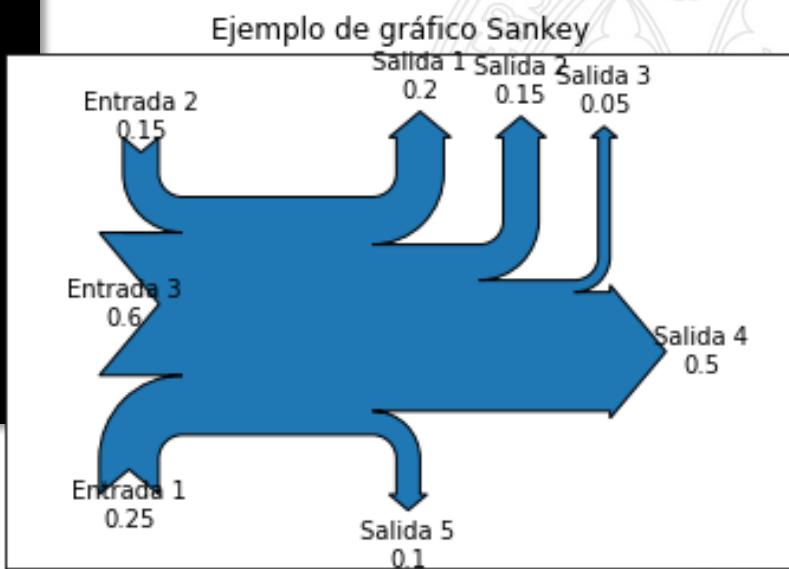


## Gráfico Sankey (2/2)

```
import matplotlib.pyplot as plt
from matplotlib.sankey import Sankey

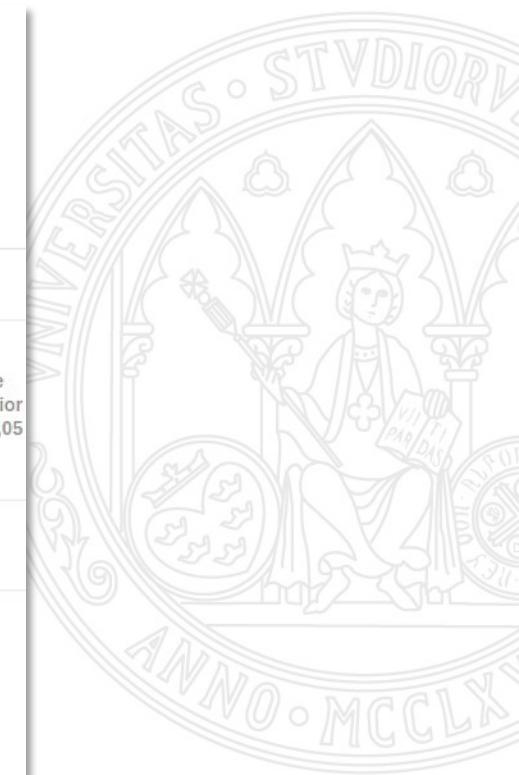
flows=[0.25, 0.15, 0.60, -0.20, -0.15,
-0.05, -0.50, -0.10]
labels=['Entrada 1', 'Entrada 2',
'Entrada 3', 'Salida 1', 'Salida 2',
'Salida 3', 'Salida 4', 'Salida 5']
orientations=[-1, 1, 0, 1, 1, 1, 0, -1]

s = Sankey(flows=flows,
           labels=labels,
           orientations=orientations)
s.finish()
plt.title('Gráfico Sankey')
```



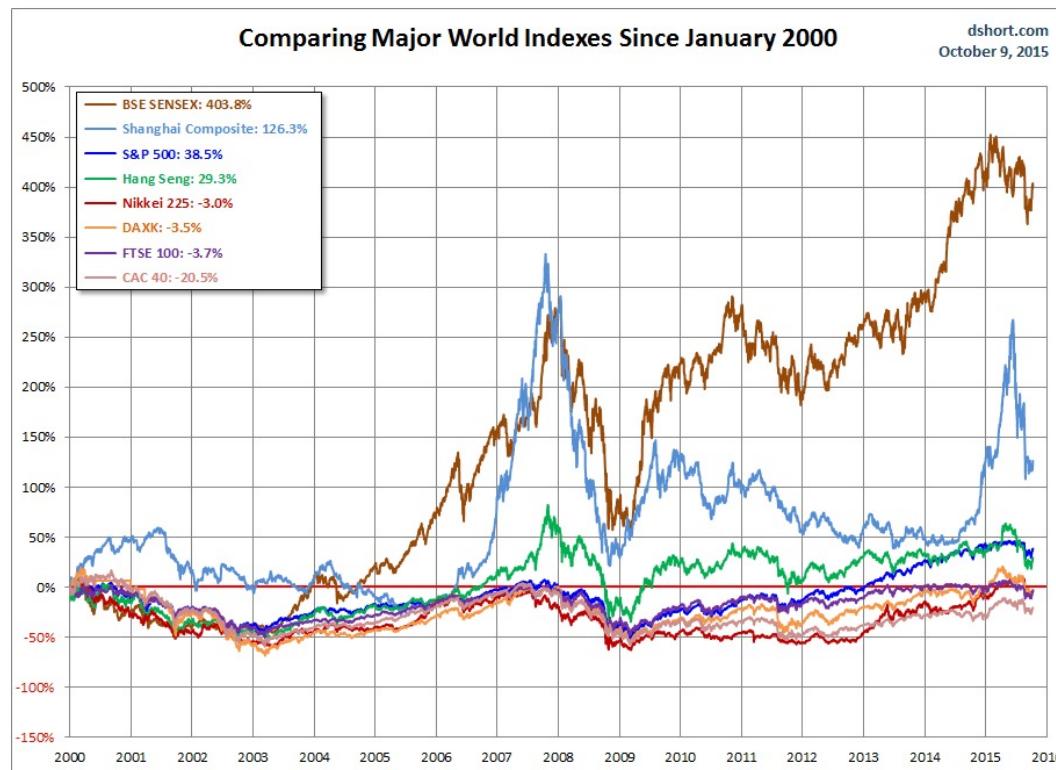
## Diagrama de líneas (1/3)

- Se puede usar cuando se desean **analizar tendencias** y se **desprecia el error de interpolación**



## Diagrama de líneas (2/3)

- Particularmente útiles para **comparar la evolución** de múltiples parámetros en el **tiempo**



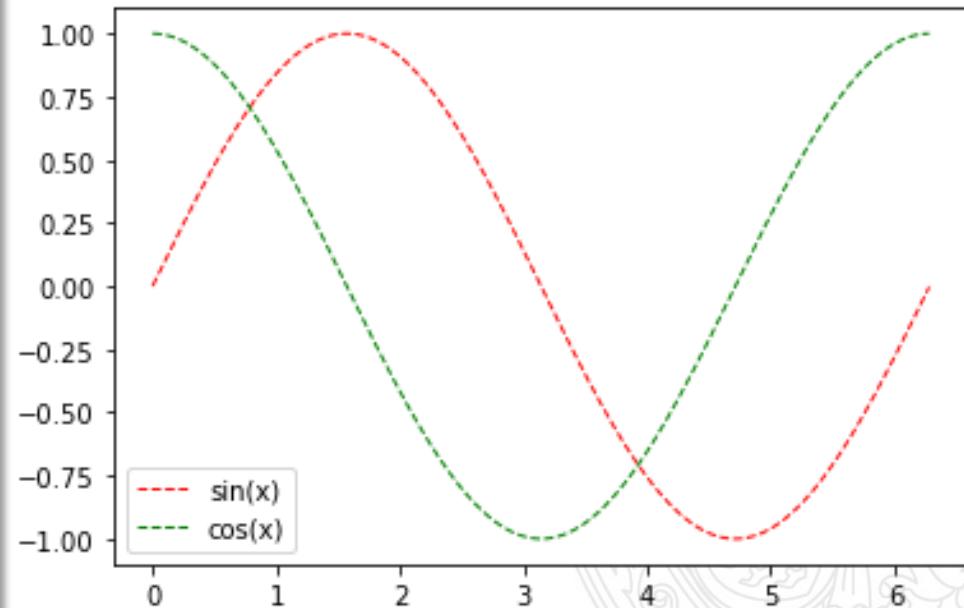
## Diagrama de líneas (3/3)

```
import matplotlib.pyplot as plt
import numpy as np

x = np.arange(0, 2*np.pi, 0.01)
y1 = np.sin(x)
y2 = np.cos(x)

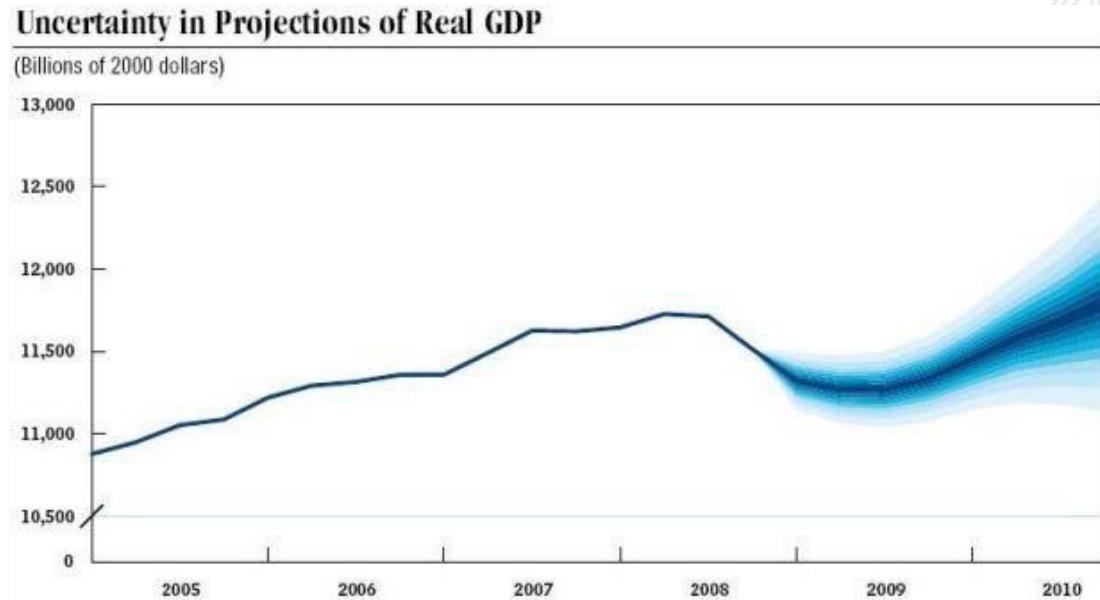
lines = plt.plot(x, y1, x, y2)

plt.setp(lines, linestyle='--')
plt.setp(lines[0], linewidth=1,
color='red', label='sin(x)')
plt.setp(lines[1], linewidth=1,
color='green', label='cos(x)')
plt.legend()
```



# Diagramas de proyección (1/2)

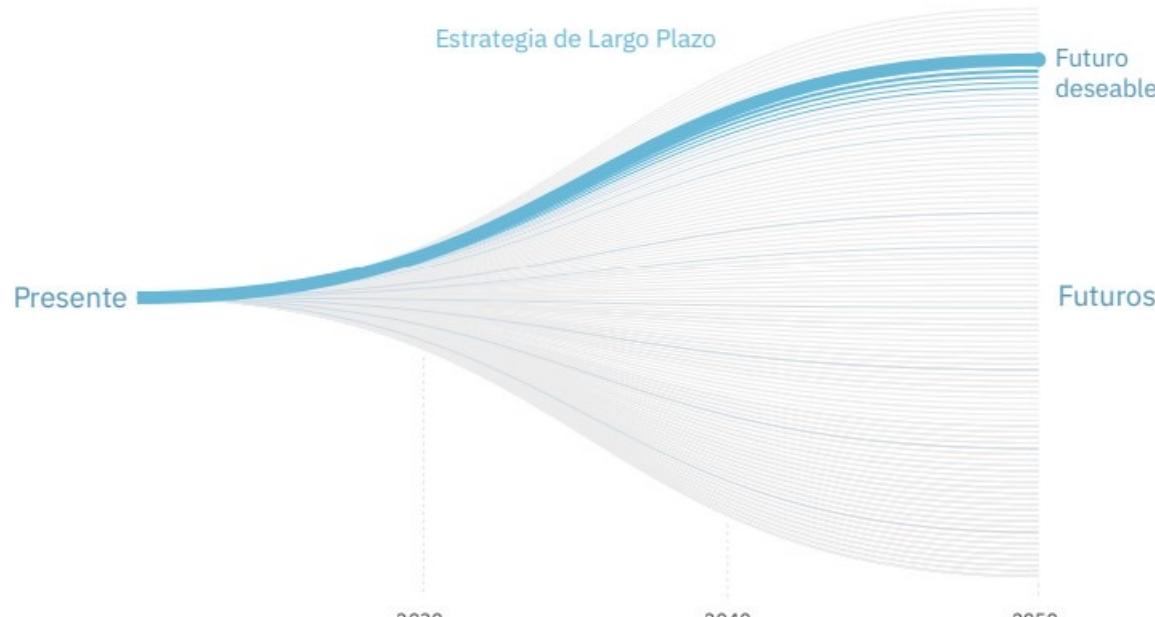
- Son una **extensión de los diagramas de líneas** en los que se dispone de un **modelo de predicción con incertidumbre**
- Se proyecta en el futuro **distintas predicciones** donde su **probabilidad** se traduce en un **esquema de colores**



## Diagramas de proyección (2/2)

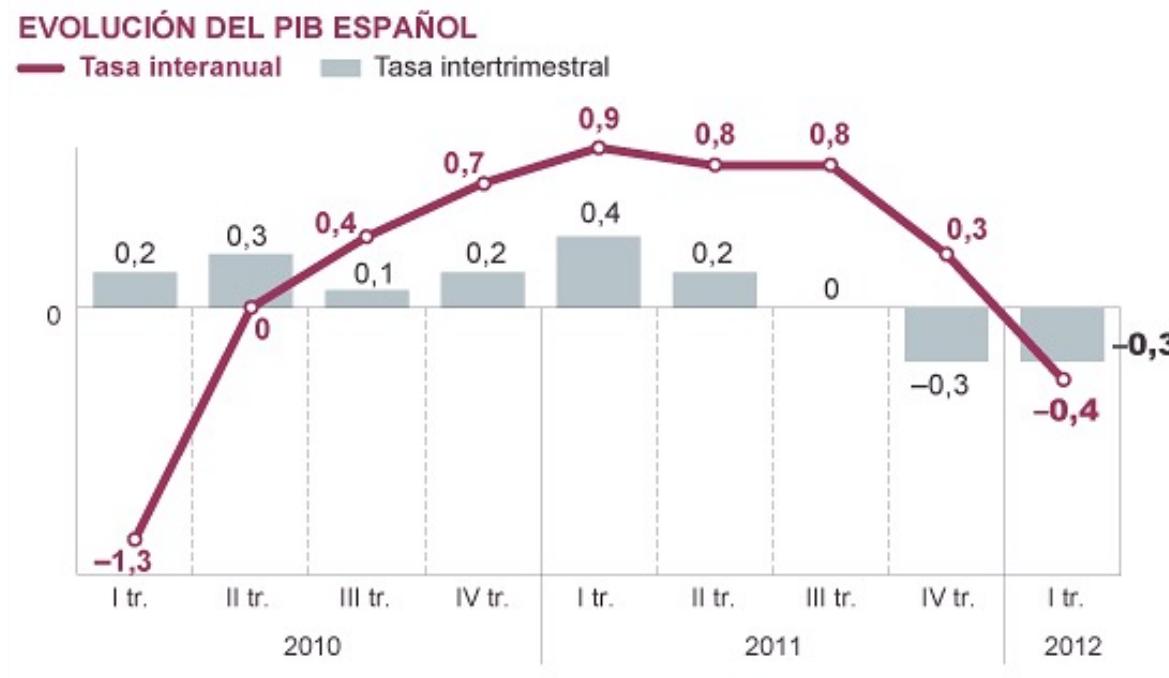
- Estrategia nacional de largo plazo ‘España 2050’

Fig. 2. Diseño de la estrategia (fase II del ejercicio)



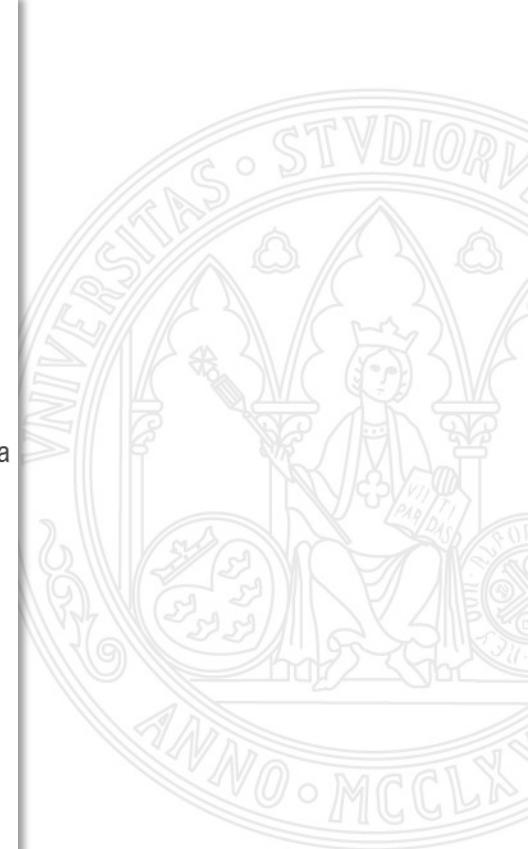
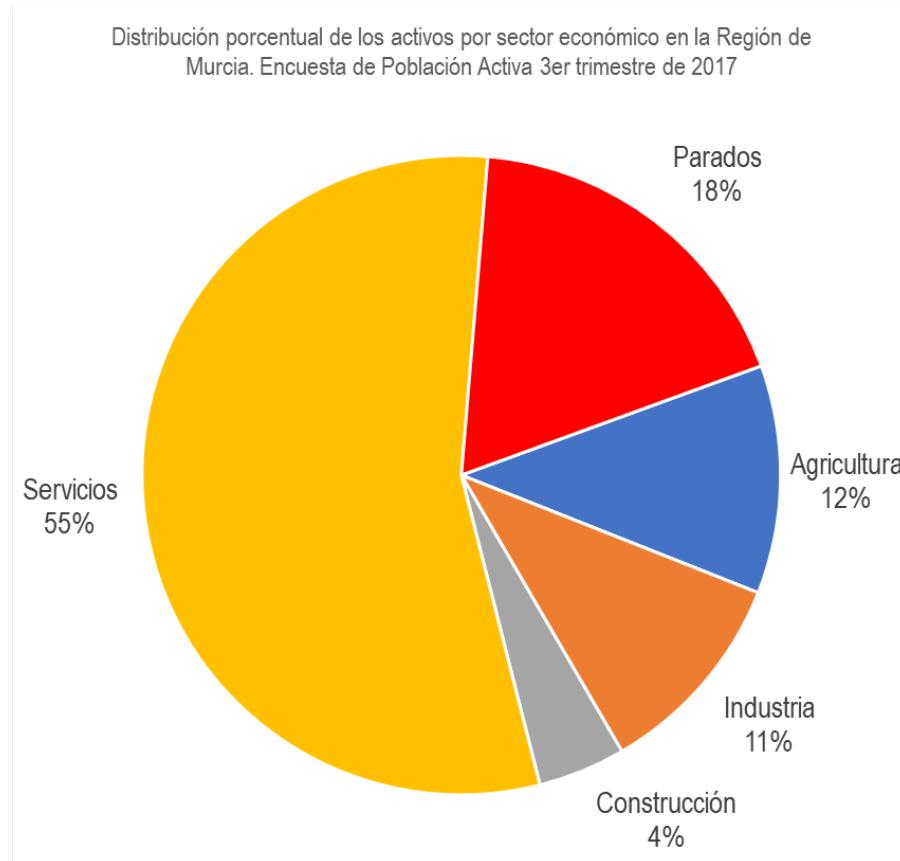
# Diagramas combinados

- Es importante valorar la utilización de **múltiples esquemas de representación** para mejorar la claridad de la información proporcionada



## Diagrama de sectores (1/2)

- Se utiliza para representar **cantidades fraccionarias**

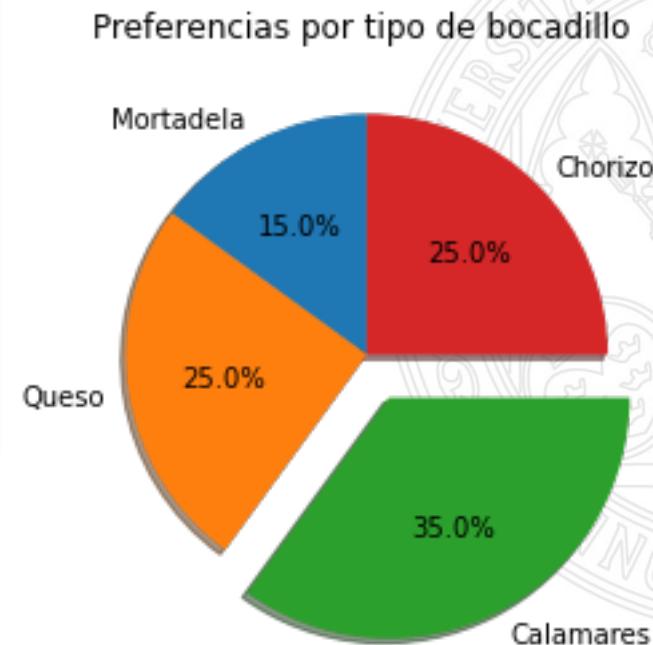


## Diagrama de sectores (2/2)

```
import matplotlib.pyplot as plt

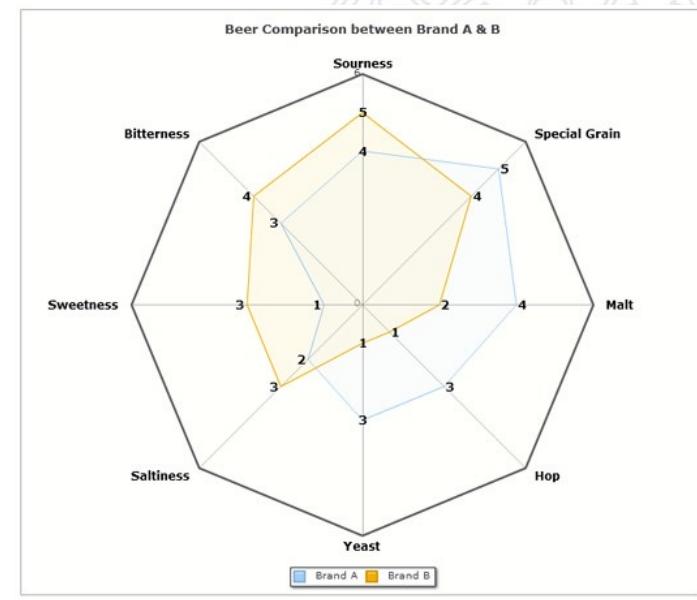
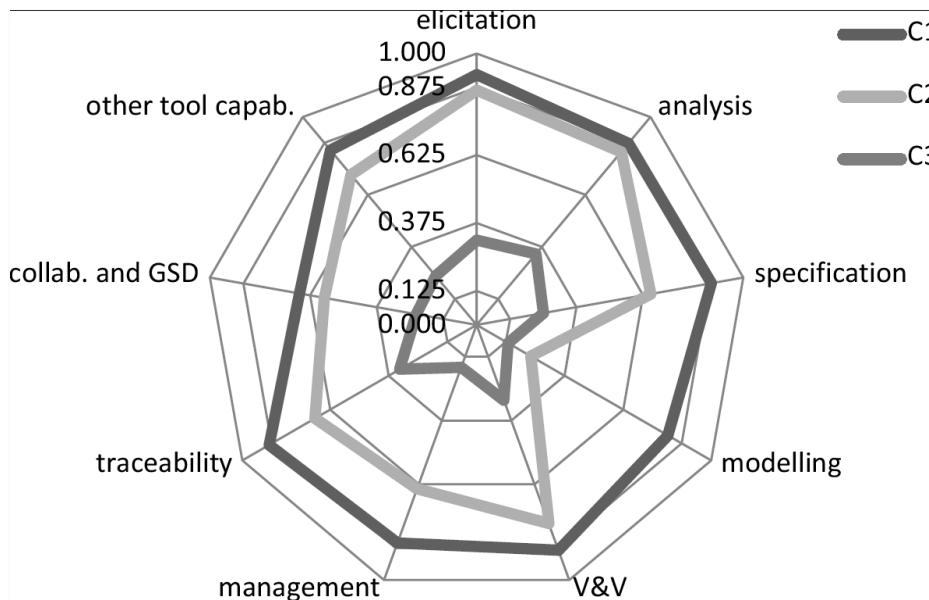
labels = ['Mortadela', 'Queso',
          'Calamares', 'Chorizo']
sizes = [15, 25, 35, 25]
explode = (0, 0, 0.2, 0)

fig1, ax1 = plt.subplots()
ax1.pie(sizes, explode=explode,
         labels=labels, autopct='%.1f%%',
         shadow=True, startangle=90)
ax1.axis('equal')
fig1.suptitle('Preferencias por tipo de bocadillo')
```



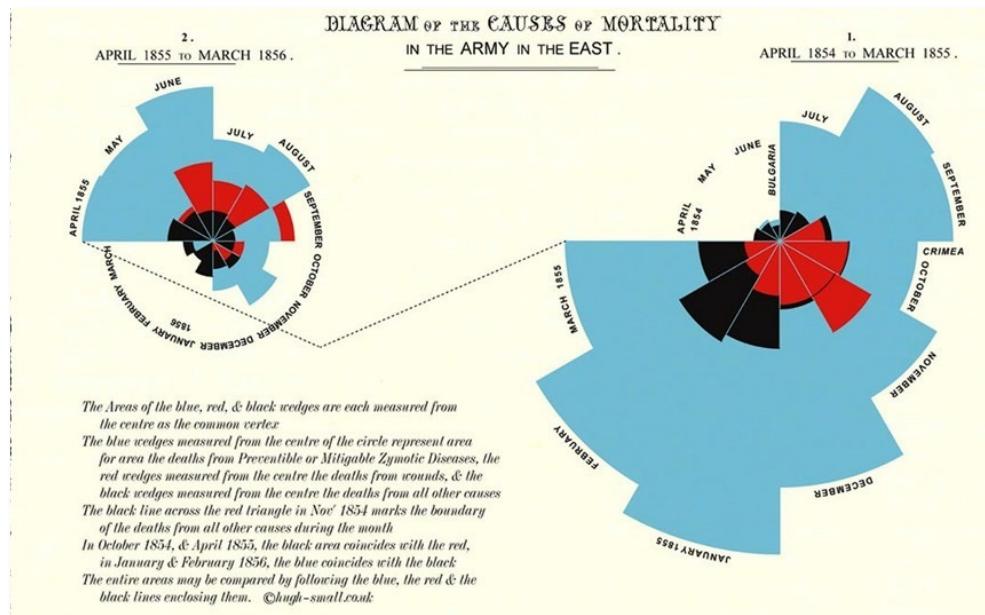
## Gráfico de tela de araña

- También llamado *gráfico de radar* y *diagrama de Kiviat*, muestra **múltiples dimensiones con distintas escalas**
- El orden de representación de los ejes y sus escalas permite visualizar **patrones relevantes**



# Gráfico de área polar (1/2)

- También permite visualizar **múltiples dimensiones con distintas escalas** en un mismo gráfico en el plano
- El valor de cada observación es **proporcional a la distancia** de cada sector respecto al **centro del círculo**

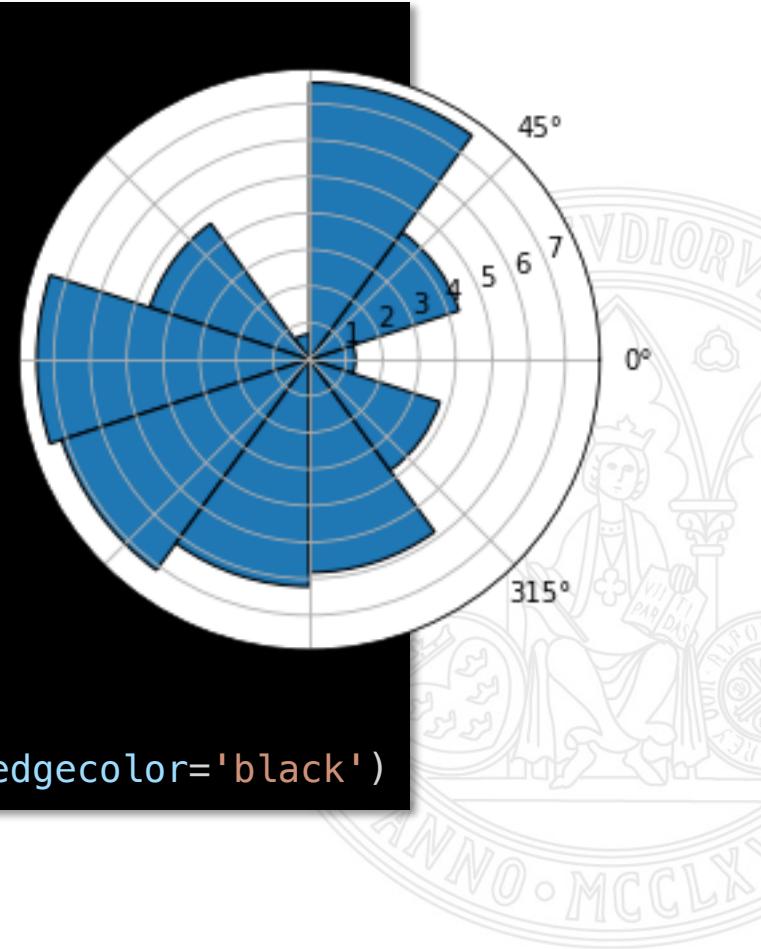


## Gráfico de área polar (2/2)

```
import numpy as np
import matplotlib.pyplot as plt

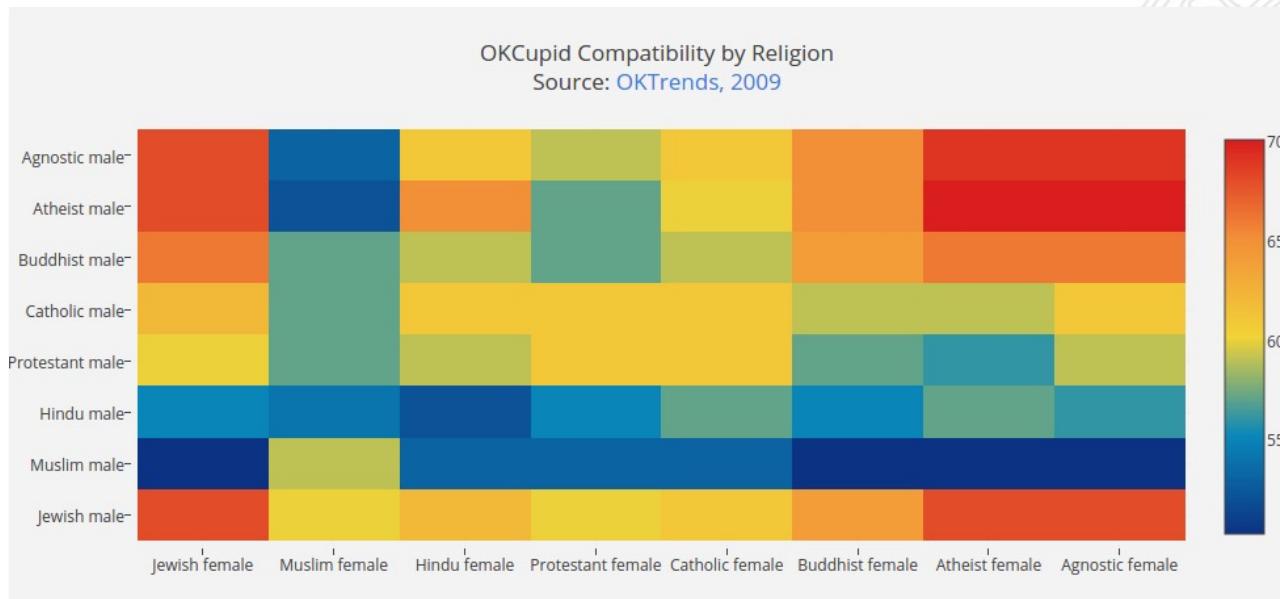
N = 10
# Circunferencia en radianes
c = 2 * np.pi
# Distribuimos los elementos a representar
theta = np.linspace(0.0, c, N, endpoint=False)
# Los valores serán aleatorios
radii = 10 * np.random.rand(N)
# Ancho de los elementos
width = (2 * np.pi) / N

ax = plt.subplot(projection='polar')
ax.bar(theta, radii, width=width, bottom=0.0, edgecolor='black')
```



# Mapa de calor

- Matriz donde los **valores numéricos** son transformados en **códigos de colores**
- Facilitan tareas de **agrupamiento visual** si podemos realizar **cambios en la ordenación** de filas y columnas



# Mapa jerárquico (1/2)

- Proceden de la **clasificación jerárquica** realizada a partir del resultado de un **árbol de decisión (treemaps)**
- A cada **rama** del árbol le **corresponde un rectángulo**, que se subdivide en tantos rectángulos como sub-ramas

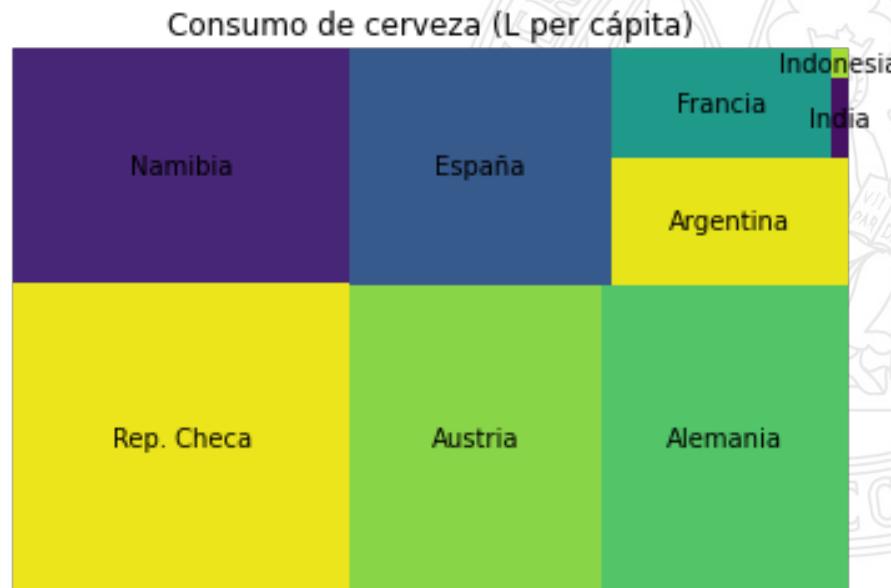


## Mapa jerárquico (2/2)

```
import numpy as np
import matplotlib.pyplot as plt
import squarify

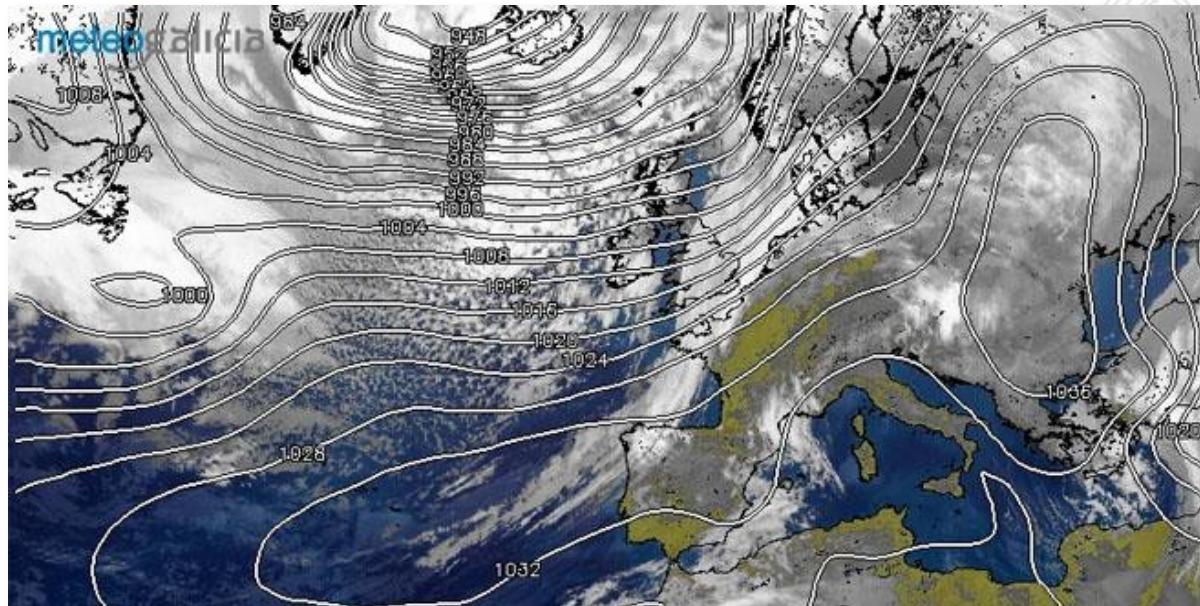
litres = [143.3, 108, 106, 104.2,
          84.8, 41.3, 33, 2, 0.7]
labels = ['Rep. Checa', 'Namibia',
          'Austria', 'Alemania',
          'España', 'Argentina',
          'Francia', 'India',
          'Indonesia']

squarify.plot(sizes=litres,
              label=labels)
plt.axis('off')
plt.title('Consumo de cerveza (L per cápita)')
```



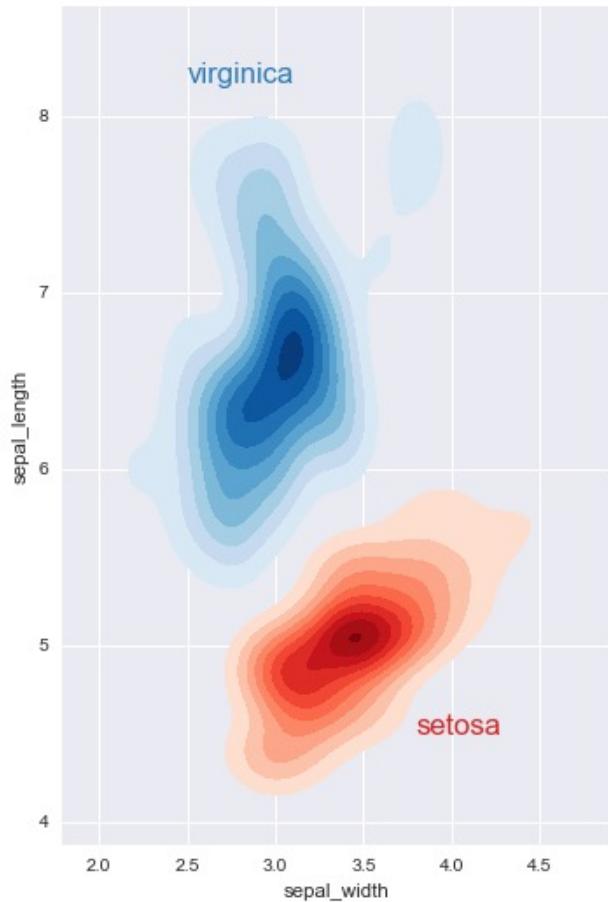
## Mapa de contorno (1/2)

- Se utiliza una **curva para unir puntos** que comparten un **mismo valor** para una función determinada
- Son habituales en meteorología, para representar isobaras, o en representaciones cartográficas



## Mapa de contorno (2/2)

- El uso de mapas de contorno se ha extendido a la representación de histogramas y funciones de densidad de probabilidad (*kernel density estimation*)

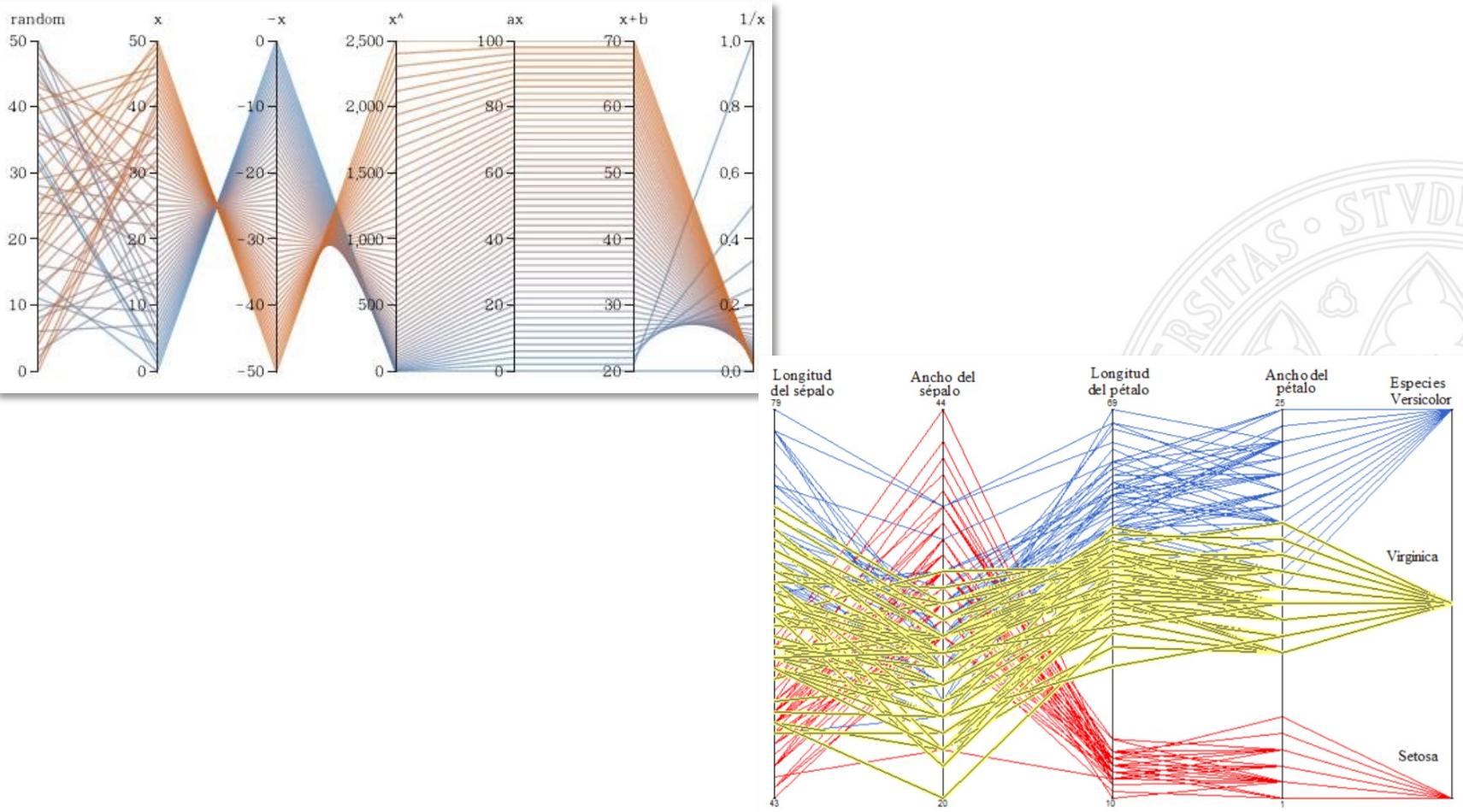


## Diagrama de coordenadas paralelas (1/2)

- Se utilizan para visualizar datos en espacios de **múltiples dimensiones**
- Se representa el conjunto de dimensiones como **ejes paralelos**, de modo que un punto se representa mediante una línea quebrada que une los valores que toma en cada una de las coordenadas
- Permite el **reconocimiento de patrones**

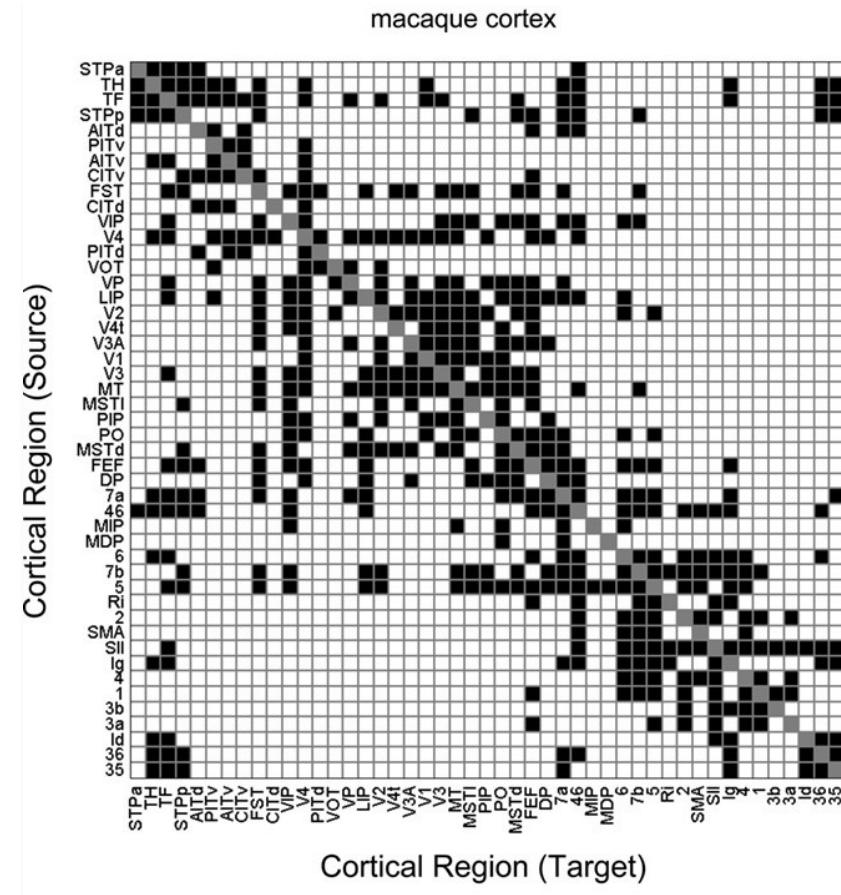


# Diagrama de coordenadas paralelas (2/2)

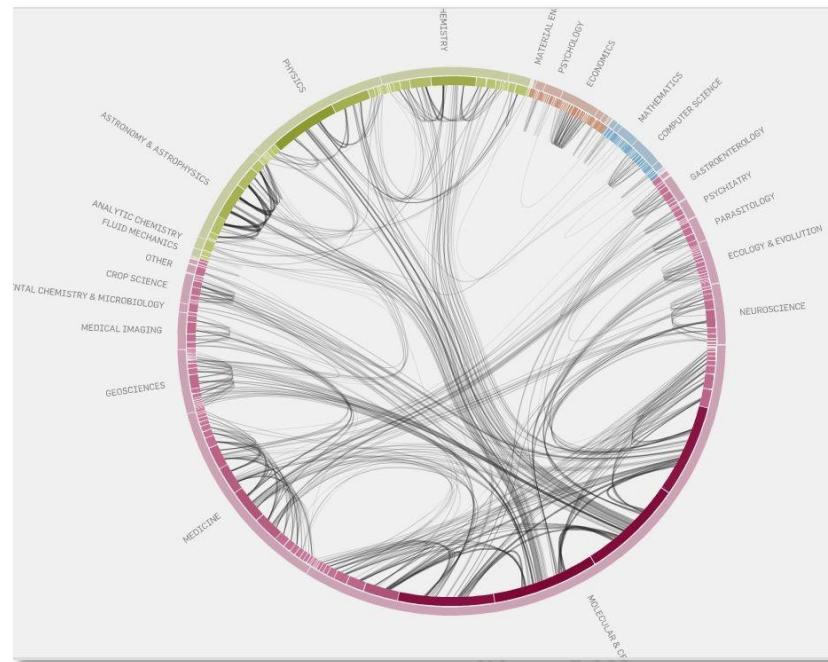


# Mapa de adyacencia

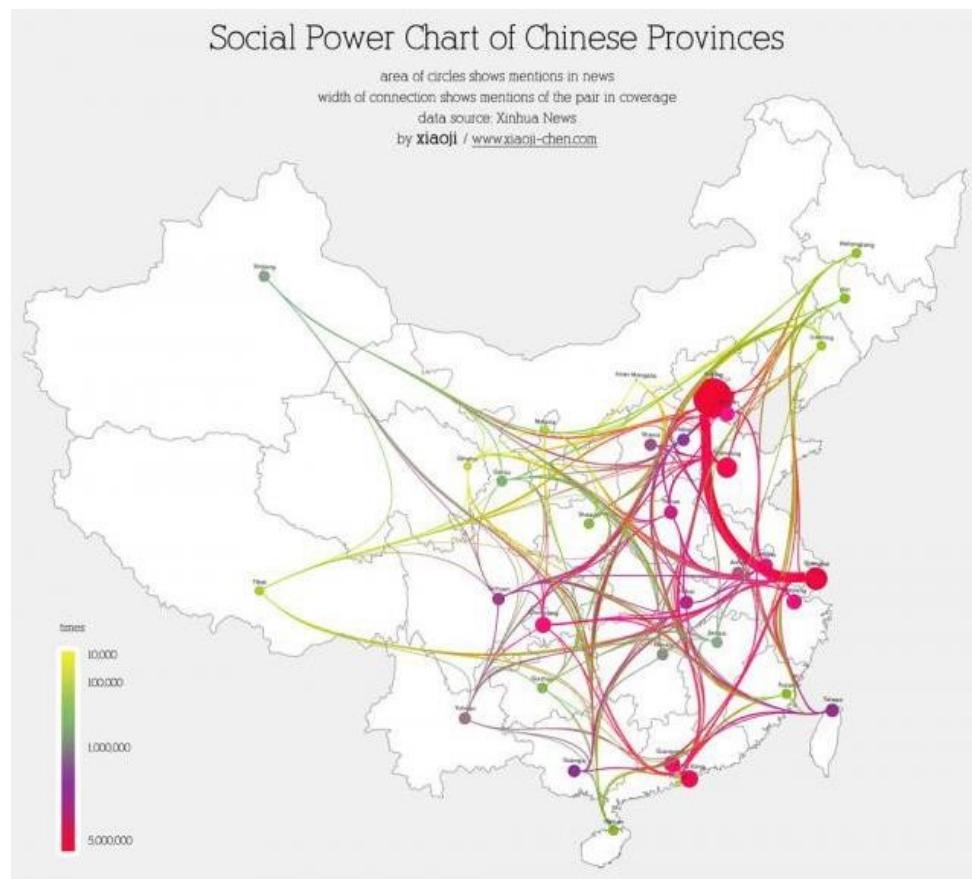
- Se utiliza para representar **grafos complejos** mediante una matriz que **colorea los nodos que están conectados**
- Una vez que se dispone de la **matriz de adyacencia** esta se reordena para separar aquellos subgrupos formados por nodos interconectados entre sí, realizando una permutación de filas (y por consiguiente, columnas)



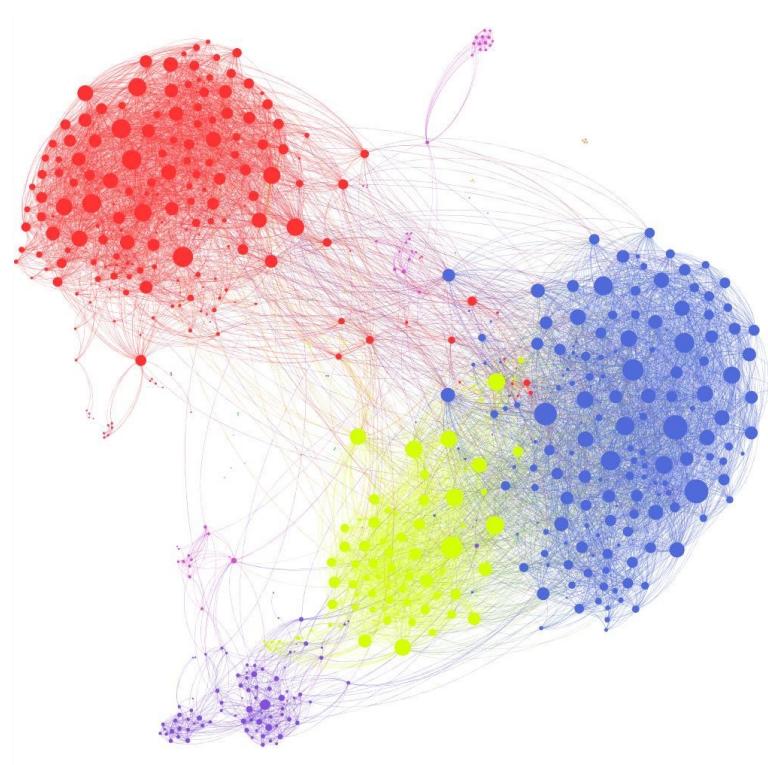
- Representación alternativa a un **grafo no dirigido**
- Útil para la identificación de **patrones** a partir de una **acumulación de arcos** cuando el número de nodos y arcos resulta muy alto
- *Grafo que representa citas entre distintas publicaciones (círculo interior) y distintas disciplinas (círculo exterior)*



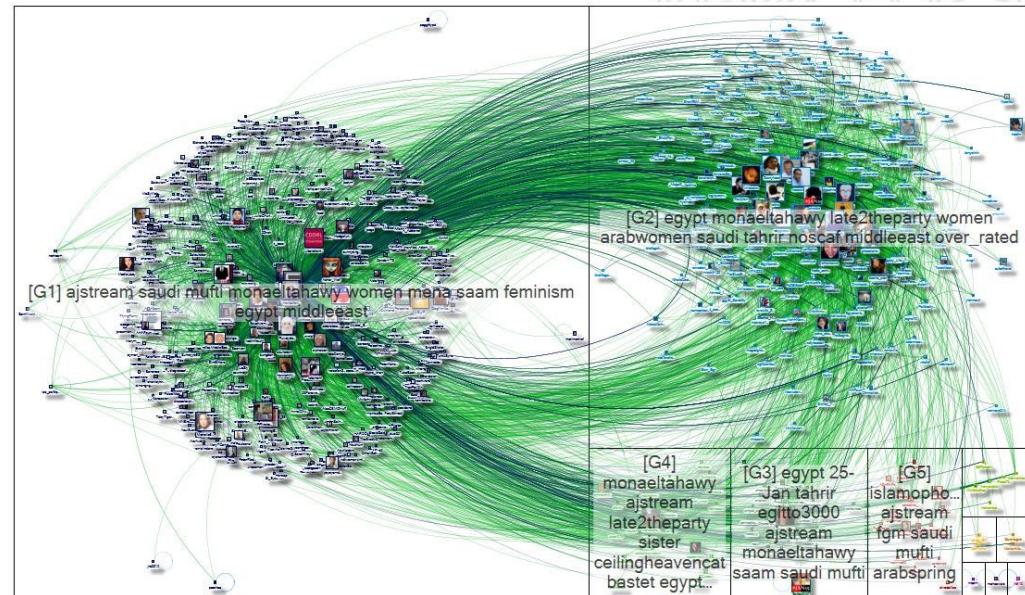
- Son una **representación extendida de un grafo**, al que se añade más información parametrizando los elementos de la visualización (tamaño de los nodos, escalas de tonalidad, grueso de los arcos, etc.)



- El análisis de redes sociales es útil para realizar un gran número de **preguntas**, o realizar operaciones de **agrupamiento o clasificación**
- *Grafo que representa relaciones entre individuos de una red social*
- *Podemos preguntar por las relaciones entre individuos de un sexo, de una edad, etc.*



- Es importante la **búsqueda de patrones** a partir de la visualización de redes, en particular de redes sociales
- *Se muestra un ejemplo de patrón de polaridad en la red social Twitter ante un artículo sobre la represión de la mujer en Oriente Medio*



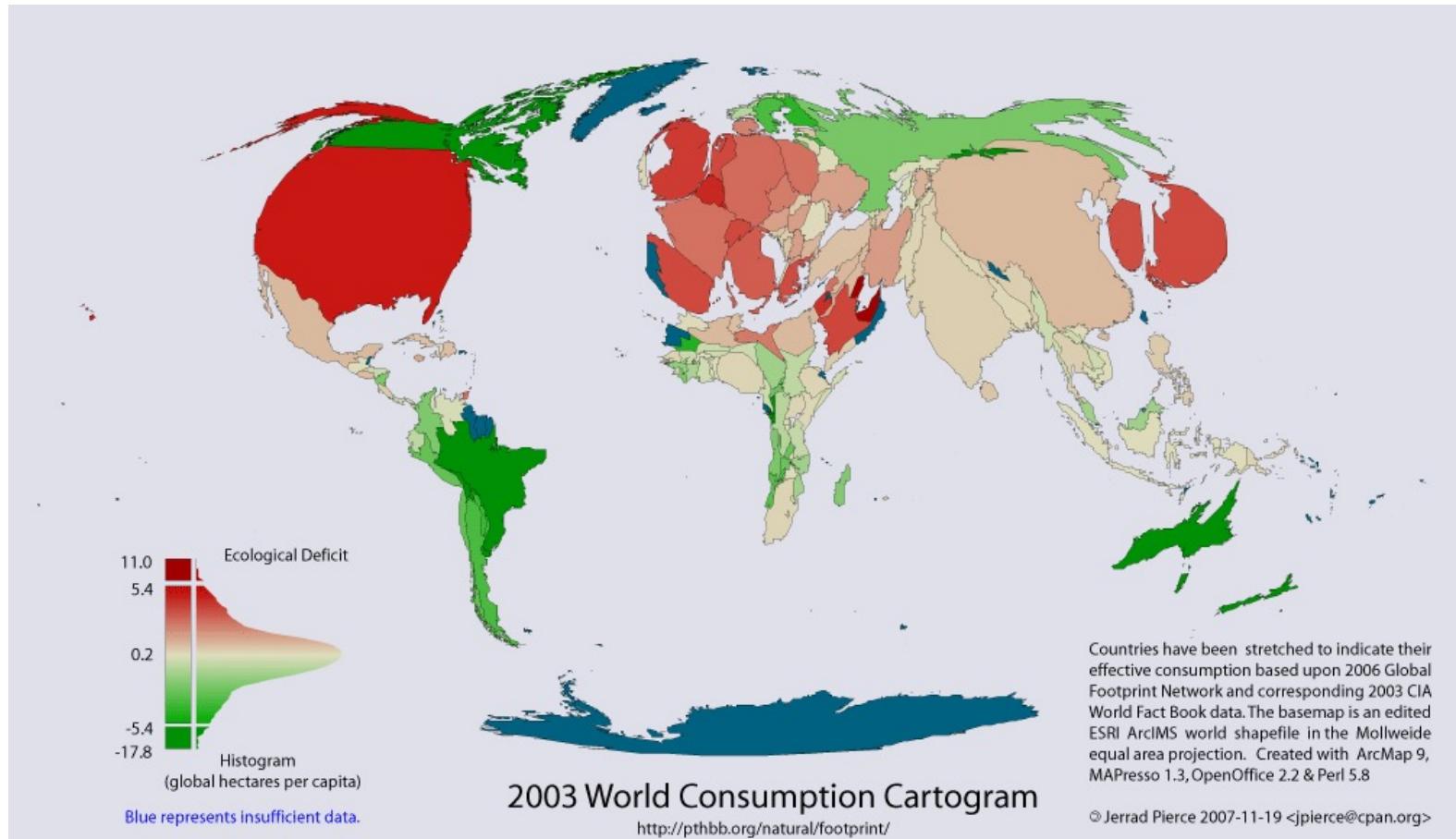
- D3 <https://d3js.org>
- Gephi <http://gephi.org>
- NodeXL  
<http://nodexl.codeplex.com>
- Cytoscape  
<http://cytoscape.org>
- Network Workbench  
<http://nwb.cns.iu.edu>
- Sci2  
<https://sci2.cns.iu.edu>
- VOS viewer  
<http://www.vosviewer.com>
- UCINET  
<https://sites.google.com/site/ucinetsoftware/home>
- GUESS  
<http://graphexploration.com.org>
- R
- SigmaJS <http://sigmajs.org>
- Circos <http://circos.ca>

# Cartograma (1/2)

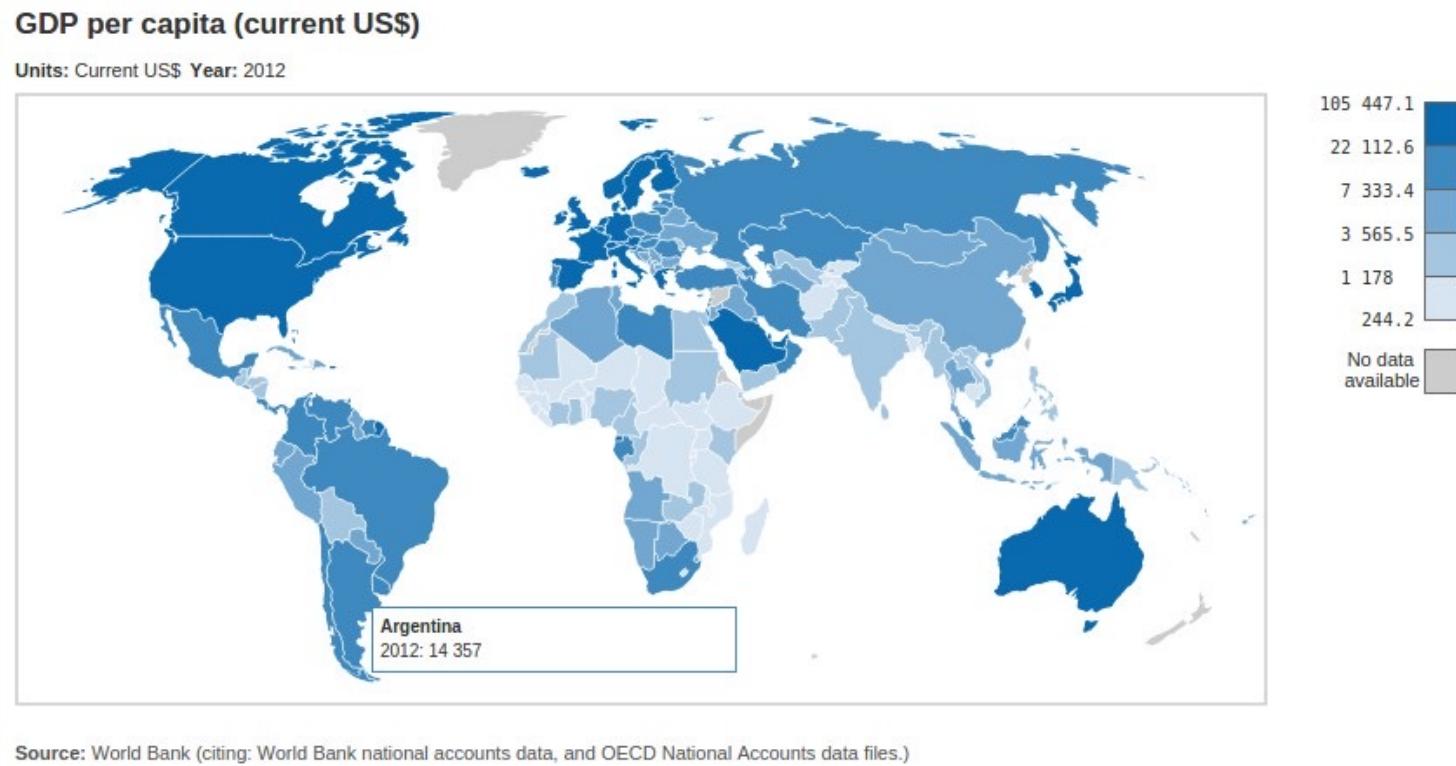
- Muestran un **mapa** con algún esquema de representación de alguna variable relevante en términos de **área o distancia**
- En algunos casos la representación del mapa resulta proporcionalmente **distorsionada**



## Cartograma (2/2)



- Son **cartogramas** donde se **colorean las regiones** en función de alguna variable de interés



- ArcGIS
- QGIS
- Tableau Public
- CartoDB
- Google Fusion Tables
- Google Earth
- GeoCommons
- JavaScript
  - D3 <http://d3js.org>
  - Leaflet <http://leafletjs.com>
  - Kartograph  
<http://kartograph.org>
  - Polymaps <http://polymaps.org>
  - Google Maps API

- E. Forsyth and L. Katz, A matrix approach to the analysis of sociometric data: preliminary report, *Sociometry*, 9(4): 340-347, 1946.
- J. Mackinlay, Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics*, 5(2):110-141, 1986.
- S.S. Stevens, On the theory of scales of measurement. *Science*, 103(2684), 1946