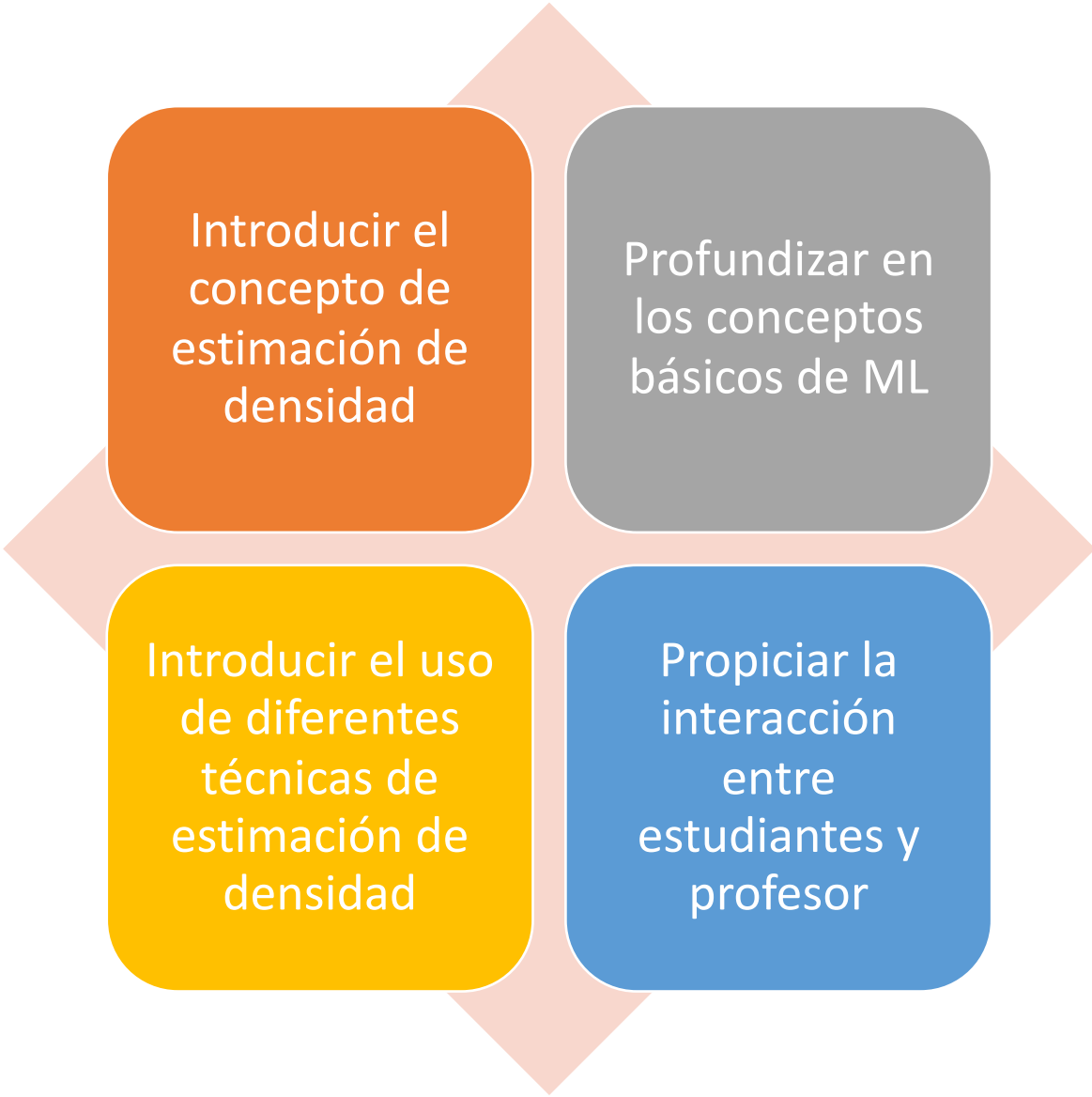

# Inteligencia Artificial LFIS419

Clase 8: Estimación de Densidad

Profesor: Jorge Arevalo (jorge.arevalo@uv.cl jab@meteo.uv.cl)

Martes 6 de junio de 2023

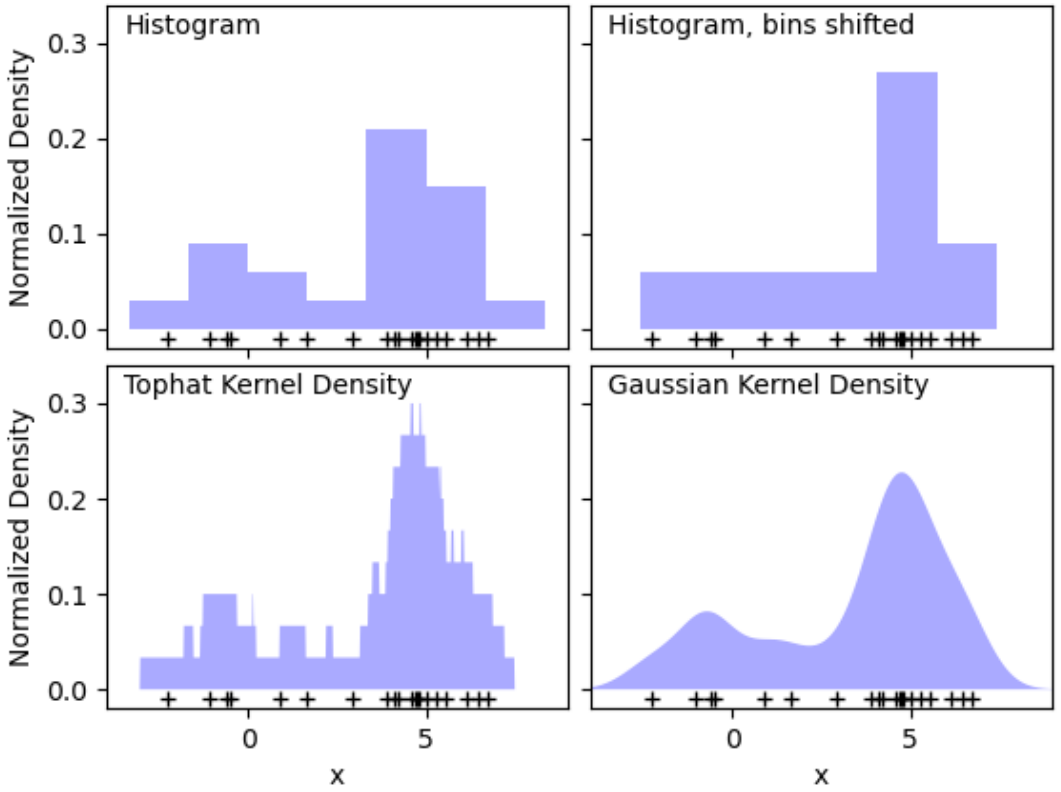
# Objetivos de la sesión



# Problema

- Tenemos un conjunto de datos (muestra) elegidos “aleatoriamente” desde un universo de datos mucho mayor (población).
- Se desea tener una estimación de la densidad de probabilidad de la población.
- Para describir datos
- Para identificar propiedades de los datos
- Para guiar nuevos experimentos (recolección de datos)
- Para simular datos nuevos

# Histograma




https://scikit-learn.org/stable/modules/density.html

# Histograma

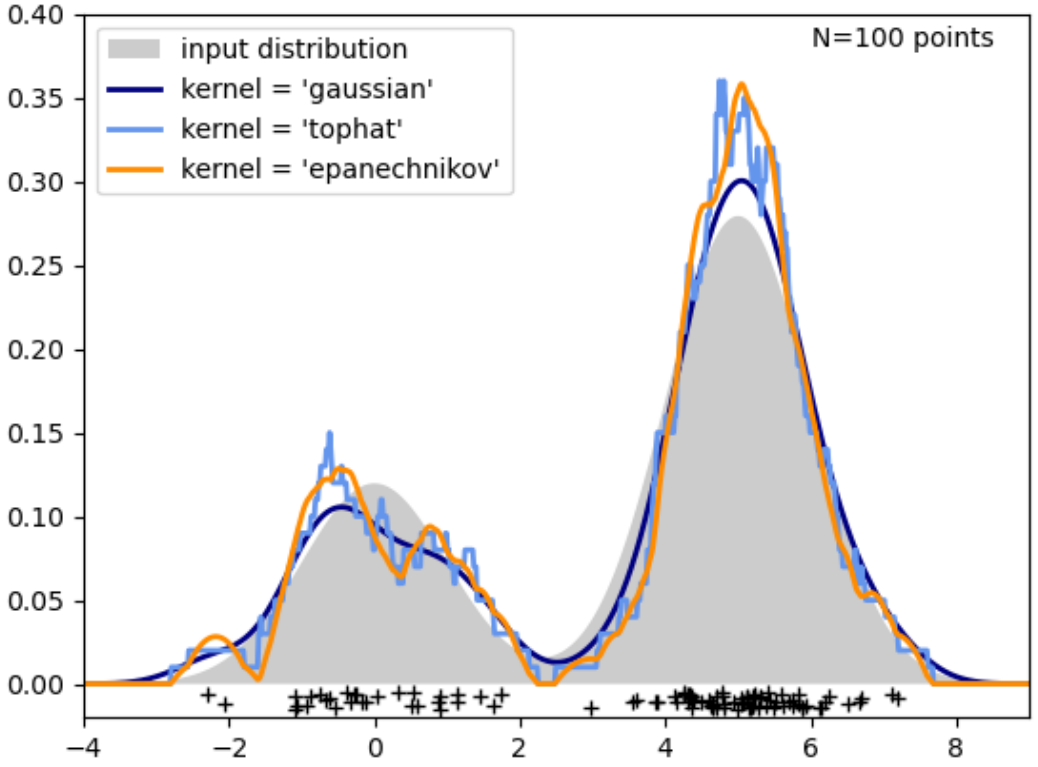
- Se cuenta la frecuencia de elementos de la muestra en diferentes “bins” a lo largo del rango de la variable.
- Es muy sensible a la elección de bins
- Diferentes bins pueden tener aspectos muy diferentes, llevando a conclusiones diferentes.

# Top-hat kernel

- Se parece al histograma
- Centrado en el valor de cada muestra se construye un rectángulo y luego se suman las áreas (incrementando el valor de la densidad)
- Produce densidades que no son suaves
- Es fácil de implentar y de entender

$$K(x;h) = \frac{1}{2h} \quad for \quad |x| < h$$

# Kernel density estimation




https://scikit-learn.org/stable/modules/density.html

# Kernel density estimation

- Centrado en cada valor de la muestra se apila una función kernel y se agrega verticalmente para construir la densidad
- Típicos kernel incluyen:

- Gaussian kernel (`kernel = 'gaussian'`)

  $K(x; h) \propto \exp(-\frac{x^2}{2h^2})$

- Tophat kernel (`kernel = 'tophat'`)

  $K(x; h) \propto 1$ if $x < h$

- Epanechnikov kernel (`kernel = 'epanechnikov'`)

  $K(x; h) \propto 1 - \frac{x^2}{h^2}$

- Exponential kernel (`kernel = 'exponential'`)

  $K(x; h) \propto \exp(-x/h)$

- Linear kernel (`kernel = 'linear'`)

  $K(x; h) \propto 1 - x/h$ if $x < h$

- Cosine kernel (`kernel = 'cosine'`)

  $K(x; h) \propto \cos(\frac{\pi x}{2h})$ if $x < h$

https://scikit-learn.org/stable/modules/density.html

# Kernel density estimation

```
>>> from sklearn.neighbors import KernelDensity
>>> import numpy as np
>>> X = np.array([[-1, -1], [-2, -1], [-3, -2], [1, 1], [2, 1], [3, 2]])
>>> kde = KernelDensity(kernel='gaussian', bandwidth=0.2).fit(X)
>>> kde.score_samples(X)
array([-0.41075698, -0.41075698, -0.41076071, -0.41075698, -0.41075698,
       -0.41076071])
```

https://scikit-learn.org/stable/modules/density.html

# Ejemplos

- https://scikit-learn.org/stable/auto_examples/neighbors/plot_kde_1d.html#sphx-glr-auto-examples-neighbors-plot-kde-1d-py

- https://scikit-learn.org/stable/auto_examples/neighbors/plot_digits_kde_sampling.html#sphx-glr-auto-examples-neighbors-plot-digits-kde-sampling-py

- https://scikit-learn.org/stable/auto_examples/neighbors/plot_species_kde.html#sphx-glr-auto-examples-neighbors-plot-species-kde-py