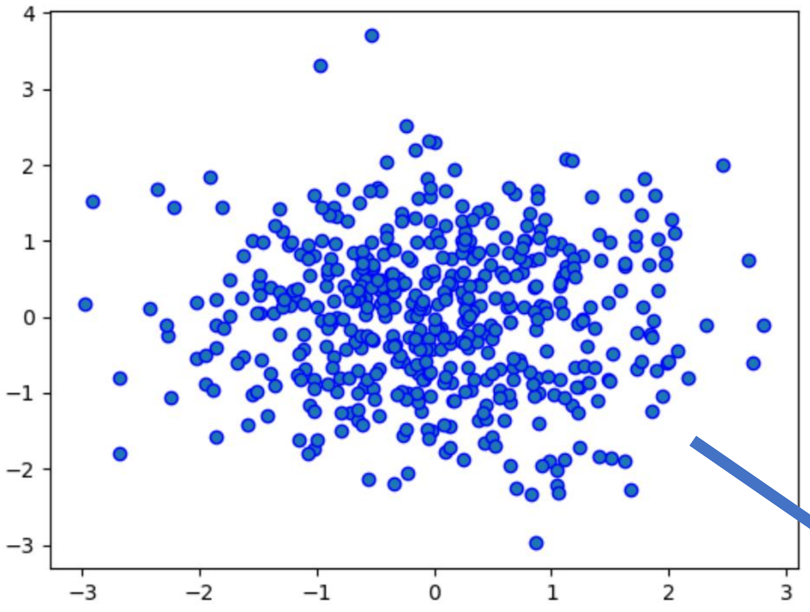
Ajuste de distribuciones

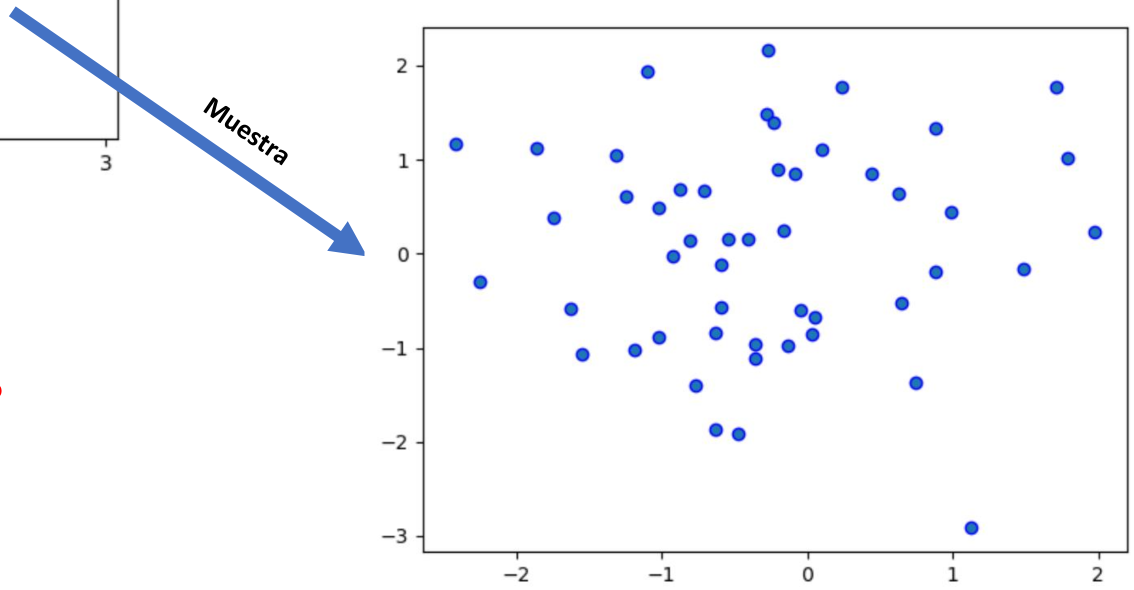
Rodrigo Maranzana

De los datos a los parámetros



Población con distribución Normal

Supuestos:
* Muestras aleatorias.
* Conozco la distribución de los datos.

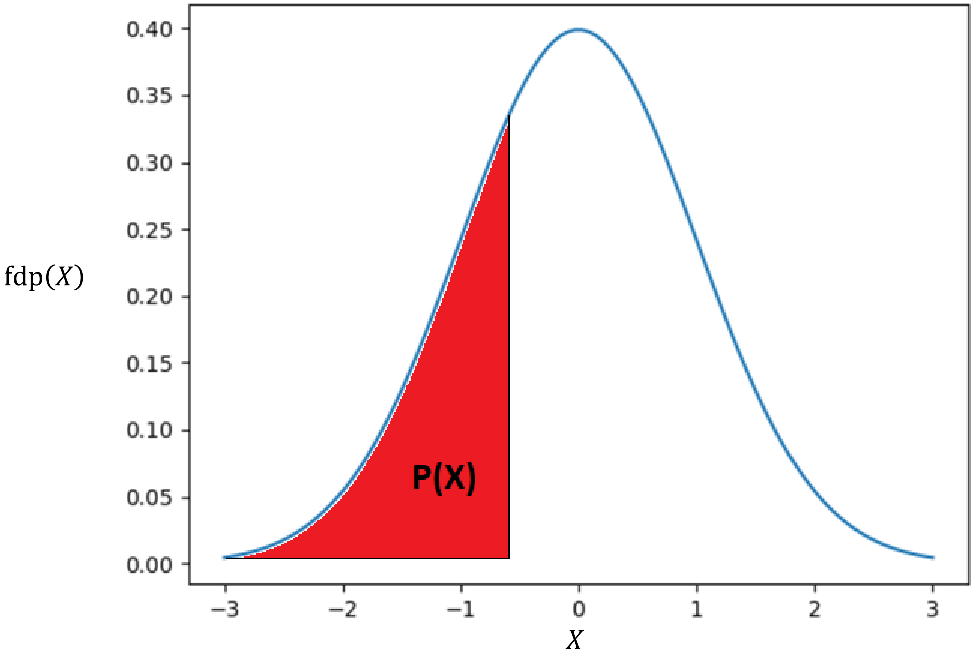


¿Cómo conozco sus parámetros μ y σ?

Probabilidad

Recordemos: probabilidad de X $\rightarrow$ $P(X) = P(X|\theta)$

Tal que conozco sus parámetros θ



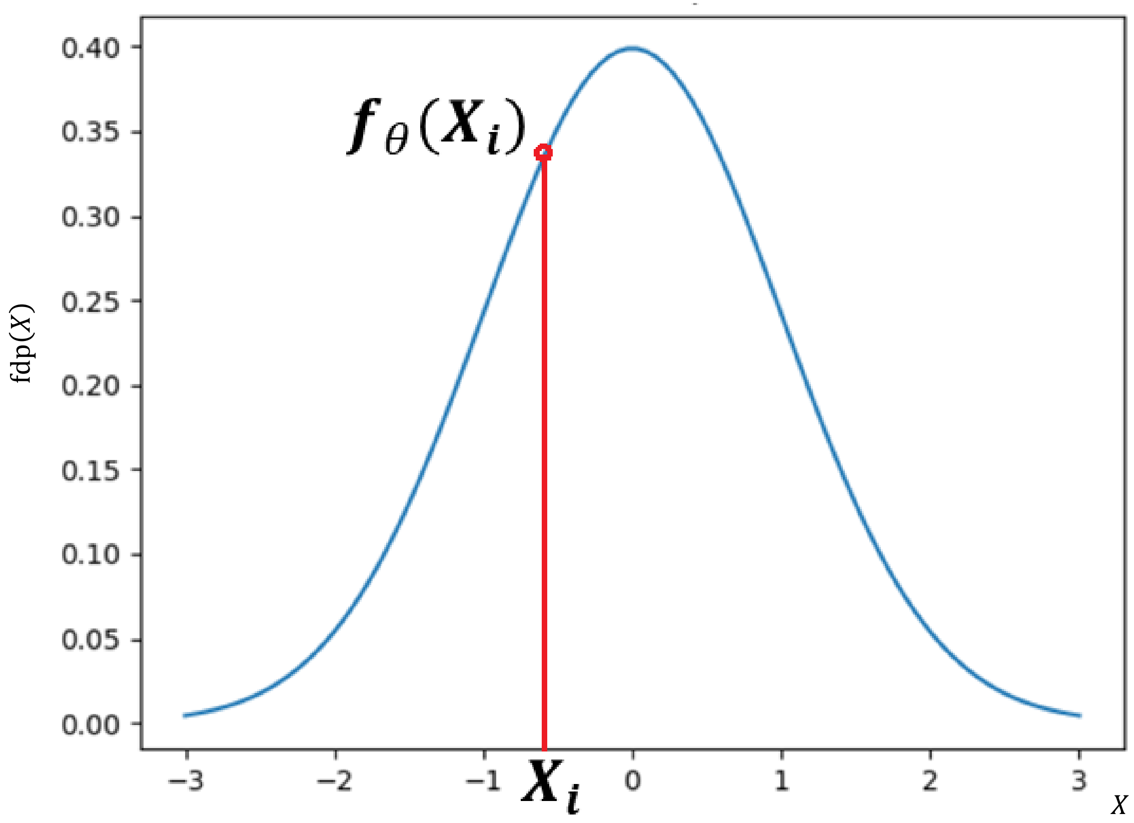
*fdp: función de densidad de probabilidad.

Verosimilitud (likelihood)

Verosimilitud (Likelihood) $\xrightarrow{\text{De 1 valor } X_i}$ $\boldsymbol{L(\theta|X_i) = f_\theta(X_i)}$

Densidad de probabilidad

Tal que conozco un valor X_i de la muestra



Sample Likelihood

Verosimilitud (Likelihood)

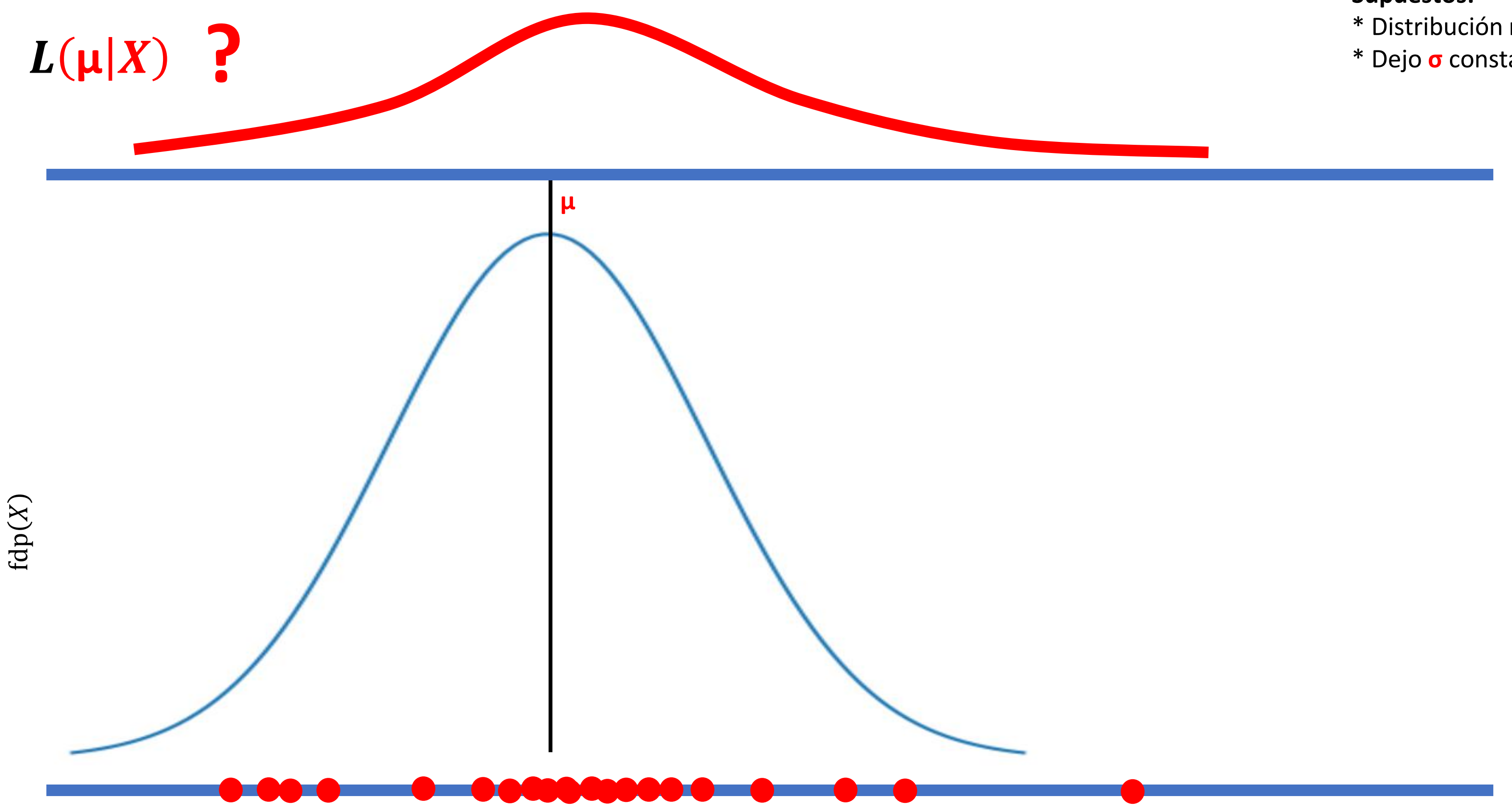
De la muestra:

$$L(\theta|X_1, X_2, \dots, X_n)$$

$$= L(\theta|X_1)L(\theta|X_2)\dots L(\theta|X_n)$$

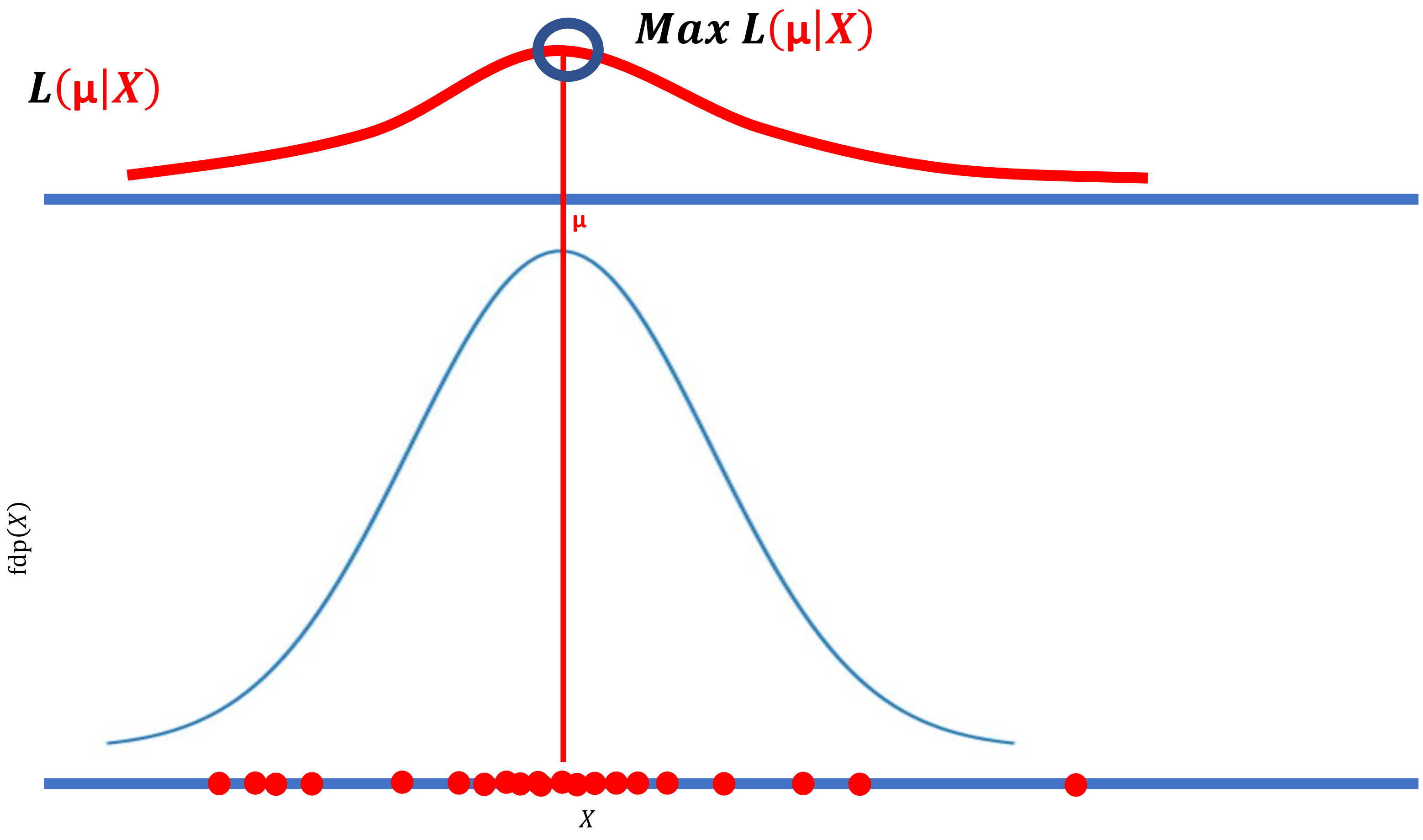
$$= \prod_{i}^{n} L(\theta|X_i)$$

Visualización de función de Likelihood



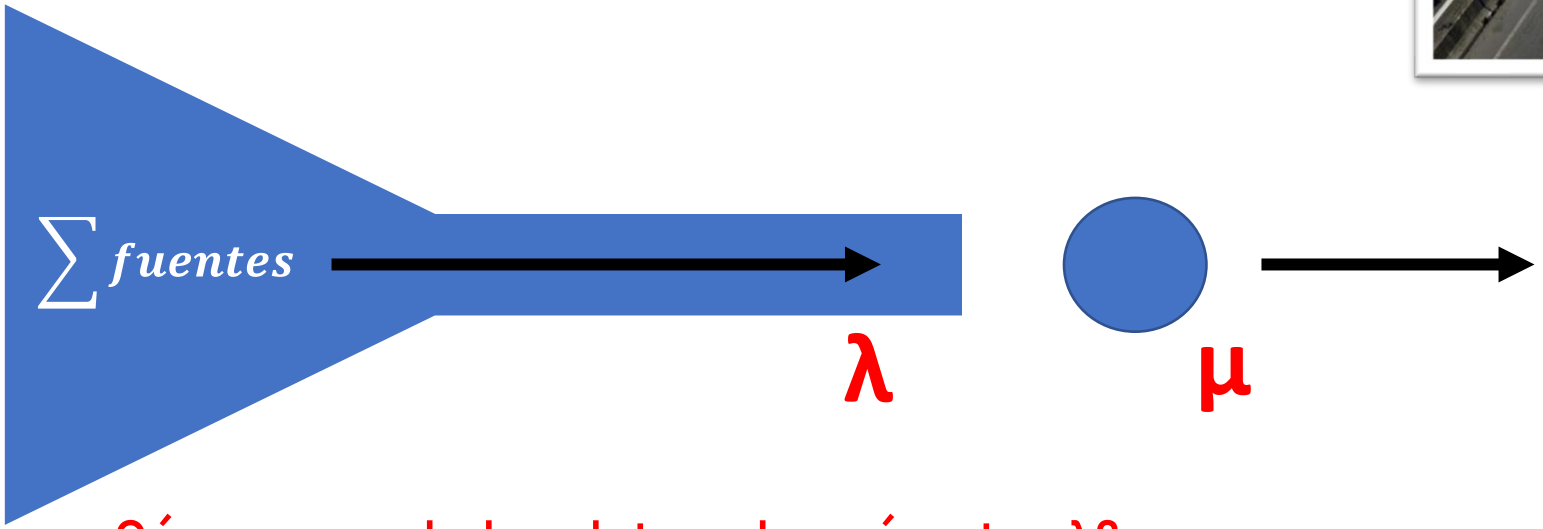
Quiero visualizar $L(\mu|X)$
Supuestos:
* Distribución normal.
* Dejo **σ** constante y varío **μ**.

Visualización de función de Likelihood



Ejemplo: estimación de flujo vehicular

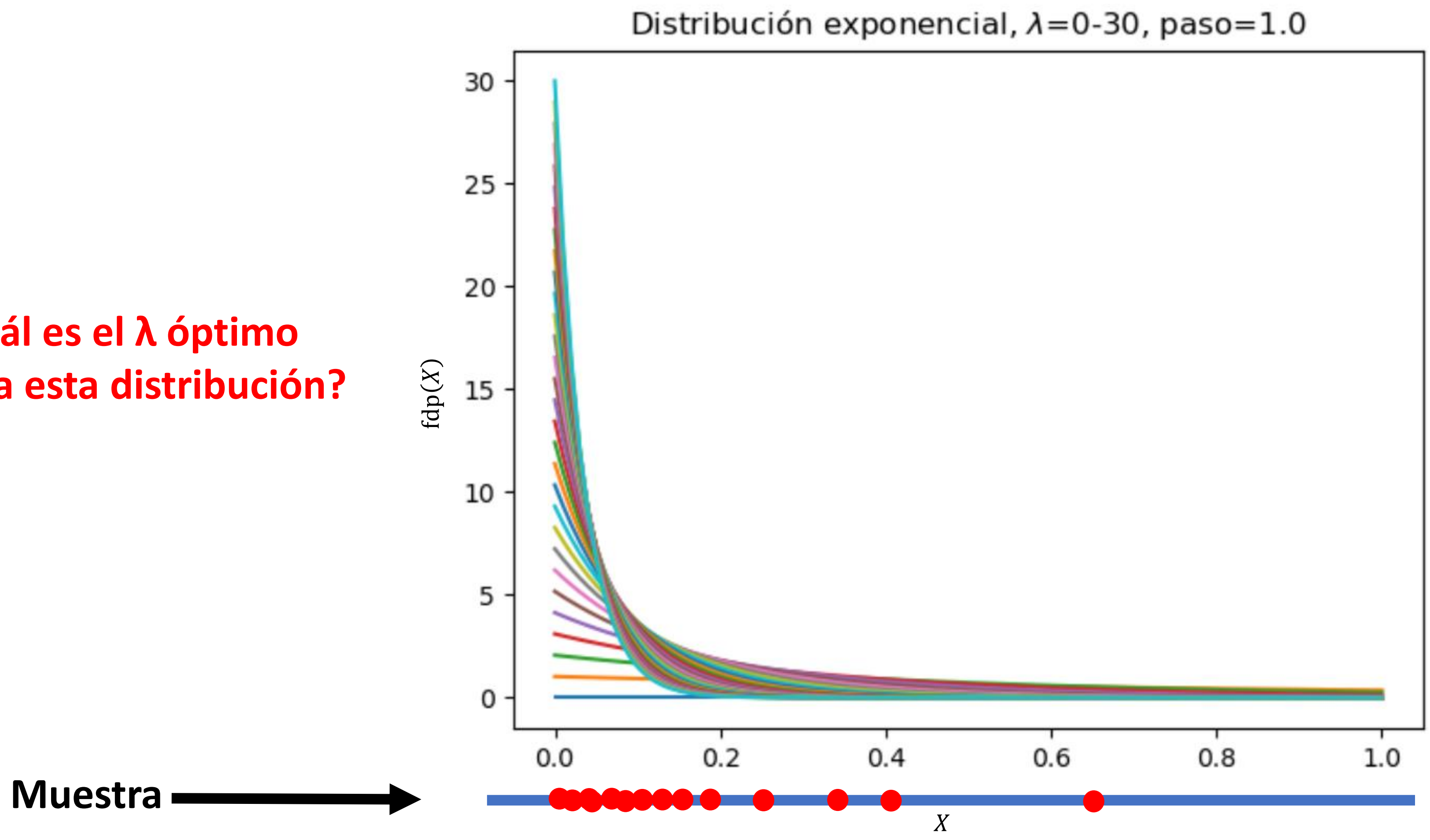
- Conozco el parámetro de servicio de agentes μ, ya que es un proceso controlable.
- En un control de tránsito, necesito estimar el parámetro de llegada de vehículos λ.
- La distribución de "tiempo entre llegadas" se supone exponencial.
- Puedo aplicar ajuste paramétrico de distribución.



¿Cómo paso de los datos al parámetro λ?

Distribución exponencial gráficamente

¿Cuál es el λ óptimo para esta distribución?



Solución analítica de Maximum Likelihood

Para este caso: ¡existe solución analítica de la Verosimilitud Máxima!

$$L(\theta|X_i \dots X_n) = \prod_i^n L(\theta|X_i) \qquad Dado\ que\colon\ L(\theta|X_i) = \boldsymbol{f_\theta(X_i)}$$

En la exponencial:

$$\boldsymbol{f_\lambda(X) = \lambda e^{-\lambda x}}$$

$$L(\lambda|X_i \dots X_n) = \prod_i^n L(\lambda|X_i) = \prod_i^n \boldsymbol{\lambda e^{-\lambda x_i}}$$

Solución analítica de Maximum Likelihood

$$L(\lambda|X_i \dots X_n) = \prod_{i}^{n} \boldsymbol{\lambda e^{-\lambda x_i}}$$

Queremos calcular:

$$\max L(\lambda|X_i \dots X_n)$$

¿Cómo encontramos el máximo analíticamente?

$$\boldsymbol{\frac{\mathrm{d}\, L(\lambda|X_i \dots X_n)}{d\lambda} = 0}$$

Solución analítica de Maximum Likelihood

$$L(\lambda|X_i \ldots X_n) = \prod_{i}^{n} \boldsymbol{\lambda e^{-\lambda x_i}}$$

$$= \boldsymbol{\lambda e^{-\lambda x_1} \lambda e^{-\lambda x_2} \ldots \lambda e^{-\lambda x_n}}$$

$$= \boldsymbol{\lambda^n e^{-\lambda x_1 - \lambda x_2 - \cdots - \lambda x_n}}$$

$$L(\lambda|X_i \ldots X_n) = \boldsymbol{\lambda^n e^{-\lambda(x_1 + x_2 + \cdots + x_n)}}$$

$$\frac{dL(\lambda|X_i \ldots X_n)}{d\lambda} = \frac{d\left[\boldsymbol{\lambda^n e^{-\lambda(x_1 + x_2 + \cdots + x_n)}}\right]}{d\lambda} = 0$$

Solución analítica de Maximum Likelihood

Sabemos que:

$$\max L(\lambda|X_i \ldots X_n) \;=\; \max \log L(\lambda|X_i \ldots X_n)$$

¡Las propiedades del logaritmo facilitan la optimización!

$$\log L(\lambda|X_i \ldots X_n) = \log \boldsymbol{\lambda^n e^{-\lambda(x_1+x_2+\cdots+x_n)}}$$

$$= \log \boldsymbol{\lambda^n} + \log \boldsymbol{e^{-\lambda(x_1+x_2+\cdots+x_n)}}$$

$$\log L(\lambda|X_i \ldots X_n) = \mathrm{n} \log \boldsymbol{\lambda} - \boldsymbol{\lambda(x_1 + x_2 + \cdots + x_n)}$$

Solución analítica de Maximum Likelihood

$$\frac{dL(\lambda|X_i \dots X_n)}{d\lambda} = \frac{d(\mathrm{n}\log\lambda - \lambda(x_1 + x_2 + \cdots + x_n))}{d\lambda} = 0$$

$$= \frac{n}{\lambda} - (x_1 + x_2 + \cdots + x_n) = 0$$

$$\boxed{\boldsymbol{\lambda = \frac{(x_1 + x_2 + \cdots + x_n)}{n}}}$$

En otras distribuciones

- ¡No todas son tan simples!
- No todas tienen solución analítica
- Uso de métodos numéricos de ajuste -> Distribución Beta
- Lo importante es: conocer cómo armar la Función de densidad

Resumen

- Puente entre **datos -> parámetros** de una Distribución
- Ajuste de datos a una densidad de probabilidad **conocida**
- Parámetros **desconocidos**

Ajuste en Python

Distribución Beta:

```
from scipy.stats import beta

a, b, loc, scale = beta.fit(x)
```

** Documentación: Solución numérica, óptimos locales!

https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.rv_continuous.fit.html