

ESTADÍSTICA INFERENCIAL

Librerías:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
from scipy.stats import shapiro, poisson, chisquare, expon, kstest
```

Generar un array con una distribución de datos normal

```
np.random.normal(loc, scale, size)
```

- loc: media
- scale: desviación estándar
- size: tamaño de la muestra

Ejemplo: `data = np.random.normal(0, 1, 1000)`

Un array con media = 0, desviación estándar = 1, 1000 muestras

DISTRIBUCIÓN NORMAL

Curva en forma de campana simétrica. El centro es la media, la distribución estándar cómo están distribuidos los datos alrededor de la media.

Test de Shapiro:

- **Hipótesis Nula (H0):** los datos provienen de una **distribución normal**.
- **Hipótesis Alternativa (H1):** los datos **no** provienen de una distribución **normal**.
- Interpretación:
 - p-value > 0.05: los datos podrían considerarse aproximadamente normales.
 - p-value < 0.05: los datos no siguen una distribución normal, hay desviaciones significativas.

Aplicación: muestras pequeñas

```
p_value = shapiro(df['columna']).pvalue
alpha = 0.05
if p_value > alpha → distribución normal
else → no se ajusta a una distribución normal
```

Test de Kolmogorov: evalúa si un conjunto de datos sigue una distribución de probabilidad específica, como la distribución normal. También se usa para comparar dos muestras y verificar si provienen de la misma población o distribución.

- **Hipótesis Nula (H0):** los datos siguen la distribución teórica, las muestras provienen de la misma población.
- **Hipótesis Alternativa (H1):** los datos no siguen la distribución teórica, las muestras no provienen de la misma población.
- Interpretación:
 - p-value > 0.05: los datos son consistentes con la distribución teórica o que las dos muestras provienen de la misma población.
 - p-value < 0.05: los datos no siguen la distribución teórica o que las dos muestras no provienen de la misma población

Aplicación: muestras grandes

```
p_value = kstest(df['columna'], 'norm').pvalue  
alpha = 0.05  
if p_value > alpha → distribución normal  
else → no se ajusta a una distribución normal
```

INTERVALO DE CONFIANZA

Rango de valores entre el que, con un porcentaje de confianza X, se encuentra un valor de una muestra.

Aplicación:

```
media = df["columna"].mean()  
error = stats.sem(df["columna"])  
nivel_confianza = 0.90  
grados_libertad = len(df["columna"]) - 1  
valor_critico = stats.t.ppf((1 + nivel_confianza) / 2, df=grados_libertad)  
limite_inferior = media - valor_critico * error  
limite_superior = media + valor_critico * error
```

Resultados:

```
Media Muestral: {np.round(media, 2)}  
Error Estándar: {np.round(error, 2)}  
Nivel de Confianza: {nivel_confianza }  
Valor Crítico: {np.round(valor_critico, 2)}  
Intervalo de Confianza: {np.round(limite_inferior, 2)}, {np.round(limite_superior, 2)}
```

PRUEBA DE HIPÓTESIS

Después de comprobar si los datos tienen una distribución normal o no (Test de Shapiro y Test de Kolmogorov), probar las hipótesis:

- Prueba de Levene: distribución no normal
- Prueba de Barlett: distribución normal