

Sesión 3: Distribuciones y visualización de datos con Matplotlib (2 horas)

3.1 Introducción a las distribuciones

3.1.1 Distribuciones discretas: binomial, geométrica, Poisson

Las **distribuciones discretas** describen la probabilidad de eventos discretos en un experimento. Ejemplos de distribuciones discretas son:

- **Distribución binomial:** describe la probabilidad de obtener un número específico de éxitos en un número fijo de ensayos independientes de Bernoulli.
- **Distribución geométrica:** describe la probabilidad de la cantidad de ensayos necesarios para obtener el primer éxito en una secuencia de ensayos independientes de Bernoulli.
- **Distribución de Poisson:** describe la probabilidad de un número específico de eventos que ocurren en un intervalo fijo de tiempo o espacio.

3.1.2 Distribuciones continuas: uniforme, normal, exponencial

Las **distribuciones continuas** describen la probabilidad de eventos continuos en un experimento. Ejemplos de distribuciones continuas son:

- **Distribución uniforme:** describe una distribución de probabilidad continua donde todos los valores en un intervalo tienen la misma probabilidad.
- **Distribución normal:** describe una distribución de probabilidad continua en forma de campana, donde los valores cercanos a la media tienen mayor probabilidad.
- **Distribución exponencial:** describe la distribución de tiempos entre eventos en un proceso de Poisson.

3.2 Estimación de parámetros

3.2.1 Método de los momentos

El **método de los momentos** es una técnica de estimación de parámetros que consiste en igualar los momentos de una distribución teórica a los momentos de una distribución empírica, basada en datos observados.

3.2.2 Método de máxima verosimilitud

El **método de máxima verosimilitud** es una técnica de estimación de parámetros que maximiza la función de verosimilitud, que es una medida de cuán probable es observar los datos dados los parámetros de la distribución.

3.3 Visualización de distribuciones con Matplotlib

3.3.1 Gráficos de función de probabilidad y función de densidad

Utilice `plt.plot` para crear gráficos de función de probabilidad (PMF) para distribuciones discretas y función de densidad (PDF) para distribuciones continuas.

```
1  from scipy.stats import binom, norm
2
3  # Gráfico de función de probabilidad (PMF) para distribución binomial
4  n = 10
5  p = 0.5
6  x_binom = np.arange(binom.ppf(0.01, n, p), binom.ppf(0.99, n, p))
7  pmf_binom = binom.pmf(x_binom, n, p)
8
9  plt.plot(x_binom, pmf_binom, "bo", ms=8)
10 plt.title("Función de probabilidad (PMF) - Distribución binomial")
11 plt.xlabel("Valores")
12 plt.ylabel("Probabilidad")
13 plt.show()
14
15 # Gráfico de función de densidad (PDF) para distribución normal
16 mu = 0
17 sigma = 1
18 x_norm = np.linspace(norm.ppf(0.01, mu, sigma), norm.ppf(0.99, mu, sigma),
19                       100)
19 pdf_norm = norm.pdf(x_norm, mu, sigma)
20
21 plt.plot(x_norm, pdf_norm, "r-", lw=5)
22 plt.title("Función de densidad (PDF) - Distribución normal")
23 plt.xlabel("Valores")
24 plt.ylabel("Densidad")
25 plt.show()
```

3.3.2 Gráficos de función de distribución acumulativa

Utilice `plt.plot` para crear gráficos de función de distribución acumulativa (CDF) tanto para distribuciones discretas como continuas.

```
1  # Gráfico de función de distribución acumulativa (CDF) para distribución
   binomial
2  cdf_binom = binom.cdf(x_binom, n, p)
3
4  plt.plot(x_binom, cdf_binom, "bo", ms=8)
5  plt.title("Función de distribución acumulativa (CDF) - Distribución
   binomial")
6  plt.xlabel("Valores")
7  plt.ylabel("Probabilidad acumulada")
8  plt.show()
9
10 # Gráfico de función de distribución acumulativa (CDF) para distribución
   normal
```

```

11 cdf_norm = norm.cdf(x_norm, mu, sigma)
12
13 plt.plot(x_norm, cdf_norm, "r-", lw=5)
14 plt.title("Función de distribución acumulativa (CDF) - Distribución normal")
15 plt.xlabel("Valores")
16 plt.ylabel("Probabilidad acumulada")
17 plt.show()

```

3.3.3 Comparación de distribuciones y ajuste de datos

Utilice las funciones de `scipy.stats` para ajustar los parámetros de una distribución teórica a los datos observados y comparar las distribuciones utilizando gráficos y medidas de bondad de ajuste.

```

1  from scipy.stats import kstest
2
3  # Generar datos aleatorios y ajustar a una distribución normal
4  data = np.random.normal(loc=5, scale=2, size=1000)
5  mu_fit, sigma_fit = norm.fit(data)
6
7  # Comparar la distribución empírica con la distribución teórica
8  x_empirical = np.sort(data)
9  cdf_empirical = np.arange(1, len(x_empirical) + 1) / len(x_empirical)
10 cdf_theoretical = norm.cdf(x_empirical, mu_fit, sigma_fit)
11
12 plt.plot(x_empirical, cdf_empirical, label="Empírica")
13 plt.plot(x_empirical, cdf_theoretical, label="Teórica")
14 plt.title("Comparación de distribuciones")
15 plt.xlabel("Valores")
16 plt.ylabel("Probabilidad acumulada")
17 plt.legend()
18 plt.show()
19
20 # Prueba de bondad de ajuste (Kolmogorov-Smirnov)
21 statistic, p_value = kstest(data, "norm", args=(mu_fit, sigma_fit))
22 print("Estadístico de prueba:", statistic)
23 print("Valor p:", p_value)

```

El resultado del test Kolmogorov-Smirnov (estadístico de prueba y valor p) nos ayuda a evaluar qué tan bien se ajusta la distribución teórica a los datos observados.

3.4 Ejercicios prácticos con Python y Matplotlib

1. Genere un conjunto de datos aleatorios siguiendo una distribución binomial con `n = 20` y `p = 0.3`. Grafique la función de probabilidad y la función de distribución acumulativa utilizando Matplotlib.
2. A partir de un conjunto de datos de edades de un grupo de personas, calcule las medidas de tendencia central (media, mediana, moda) y las medidas de dispersión (rango, varianza, desviación estándar). Visualice estos datos utilizando un histograma y un gráfico de caja y bigotes.

3. Genere un conjunto de datos aleatorios siguiendo una distribución exponencial con una tasa de $\lambda = 0.5$. Ajuste los datos a una distribución exponencial y compare la distribución empírica con la distribución teórica utilizando un gráfico de función de distribución acumulativa y la prueba Kolmogorov-Smirnov.