

Contents

| | |
|---|----------|
| Clase 4: Distribuciones No Paramétricas y Simulación de Datos de Entrada en R | 1 |
| Datos de tiempo de falla | 3 |
| 1. Construir la ECDF | 3 |
| Graficar la ECDF (opcional) | 3 |
| 2. Generar 1000 muestras aleatorias de la distribución empírica | 3 |
| 3. Calcular la media y la desviación estándar de las muestras | 3 |
| 4. Comparar con los datos originales | 3 |
| Datos de tiempo de falla (mismos que antes) | 3 |
| Instalar y cargar el paquete "density" (si no lo has hecho ya) | 3 |
| install.packages("density") | 3 |
| Construir el modelo KDE con diferentes anchos de banda | 4 |
| Graficar los modelos KDE | 4 |
| Función para generar muestras aleatorias de una KDE (aproximación) | 4 |
| Generar 1000 muestras aleatorias del KDE (con ancho de banda alto) | 4 |
| Calcular y comparar estadísticas | 4 |
| Comparar con los datos originales (como en el ejercicio anterior) | 4 |
| Clase 4: Distribuciones No Paramétricas y Simulación de Datos de Entrada en R | |
| 1. Objetivos de la Clase: | |
| • Comprender la necesidad y aplicabilidad de distribuciones no paramétricas para el modelado de datos de entrada. | |
| • Aprender a generar distribuciones empíricas a partir de datos observados usando R. | |
| • Explorar técnicas de remuestreo (bootstrap) para obtener estimaciones robustas de parámetros y realizar inferencias. | |
| • Aplicar las funciones de simulación de R para generar datos basados en distribuciones ajustadas. | |
| • Validar modelos de datos de entrada mediante la comparación de datos simulados y reales. | |
| 2. Contenido Teórico Detallado: | |
| • Distribuciones No Paramétricas: | |
| – Introducción: Limitaciones de las distribuciones paramétricas (supuestos sobre la forma de la distribución). Ventajas de las distribuciones no paramétricas (flexibilidad, no requieren supuestos sobre la forma). | |
| – Distribución Empírica: Definición de la distribución empírica como la distribución de probabilidad basada directamente en los datos observados. Cálculo de la función de distribución acumulativa empírica (ECDF). | |
| – Kernel Density Estimation (KDE): Introducción al KDE como una técnica para suavizar la distribución empírica. Selección del ancho de banda (bandwidth) y su impacto en la suavidad de la estimación. | |

- **Remuestreo (Bootstrap):**

- **Concepto:** Descripción del bootstrap como una técnica de remuestreo con reemplazo para estimar la distribución de un estadístico.
- **Aplicaciones:** Estimación de intervalos de confianza para parámetros, evaluación de la precisión de los estimadores.
- **Implementación en R:** Uso de paquetes como `boot` para realizar análisis bootstrap.

- **Simulación de Datos de Entrada en R:**

- **Funciones de Simulación:** Uso de las funciones `rnorm()`, `rexp()`, `rgamma()`, `runif()`, etc. para generar datos aleatorios basados en las distribuciones ajustadas previamente.
- **Generación de Datos a partir de Distribuciones Empíricas:** Uso de la función `sample()` para generar datos a partir de la distribución empírica. Implementación de simulación basada en KDE.

- **Validación de Modelos de Datos de Entrada:**

- **Comparación Visual:** Uso de histogramas, Q-Q plots y P-P plots para comparar la distribución de los datos simulados con la distribución de los datos reales.
- **Pruebas Estadísticas:** Aplicación de pruebas de bondad de ajuste (Kolmogorov-Smirnov, Chi-cuadrado) para comparar la distribución de los datos simulados con la distribución de los datos reales.
- **Análisis de Sensibilidad:** Evaluación del impacto de las diferentes distribuciones de entrada en los resultados de la simulación.

3. Ejemplos o Casos de Estudio:

- **Caso 1: Modelado del Tiempo de Reparación de una Máquina:**

- Datos: Tiempo de reparación de una máquina recolectados durante un período de tiempo.
- Análisis:
 1. Generar una distribución empírica de los tiempos de reparación.
 2. Aplicar KDE para obtener una estimación suavizada de la distribución.
 3. Usar bootstrap para estimar el intervalo de confianza del tiempo medio de reparación.
 4. Simular datos de tiempo de reparación basados en la distribución empírica y en KDE.
 5. Comparar los datos simulados con los datos reales usando histogramas y pruebas de Kolmogorov-Smirnov.

- **Caso 2: Modelado del Tiempo entre Llegadas de Clientes a una Tienda:**

- Datos: Tiempo entre llegadas de clientes a una tienda durante un día.
- Análisis:
 1. Generar la distribución empírica de los tiempos entre llegadas.
 2. Simular nuevos tiempos entre llegadas usando la función `sample()` basada en la distribución empírica.
 3. Comparar la distribución de los datos simulados con la distribución de los datos reales.
 4. Evaluar si el modelo empírico es una buena representación de los datos.

4. Problemas Prácticos o Ejercicios con Soluciones:

- **Ejercicio 1:**

- **Problema:** Dados los siguientes datos de tiempo de falla de un componente electrónico (en horas): [120, 150, 180, 200, 220, 250, 280, 300, 320, 350].
 1. Construya la función de distribución acumulativa empírica (ECDF) en R.
 2. Genere 1000 muestras aleatorias a partir de la distribución empírica.
 3. Calcule la media y la desviación estándar de las muestras generadas.
 4. Compare los resultados con la media y la desviación estándar de los datos originales.
- **Solución:**

“R

Datos de tiempo de falla

```
datos <- c(120, 150, 180, 200, 220, 250, 280, 300, 320, 350)
```

1. Construir la ECDF

```
ecdf_datos <- ecdf(datos)
```

Graficar la ECDF (opcional)

```
plot(ecdf_datos, main="ECDF del Tiempo de Falla")
```

2. Generar 1000 muestras aleatorias de la distribución empírica

```
set.seed(123) # Para reproducibilidad muestras <- sample(datos, size=1000, replace=TRUE)
```

3. Calcular la media y la desviación estándar de las muestras

```
media_muestras <- mean(muestras) sd_muestras <- sd(muestras)
```

4. Comparar con los datos originales

```
media_original <- mean(datos) sd_original <- sd(datos)
```

```
cat("Media de las muestras:", media_muestras, "\n") cat("Desviación estándar de las muestras:",  
sd_muestras, "\n") cat("Media de los datos originales:", media_original, "\n") cat("Desviación estándar  
de los datos originales:", sd_original, "\n") “
```

- **Ejercicio 2:**

- **Problema:** Utilice los datos del Ejercicio 1 para construir un modelo KDE utilizando R. Experimente con diferentes valores de ancho de banda (bandwidth) y observe cómo afecta la suavidad de la estimación. Genere 1000 muestras aleatorias del modelo KDE.
- **Solución:**

“R

Datos de tiempo de falla (mismos que antes)

```
datos <- c(120, 150, 180, 200, 220, 250, 280, 300, 320, 350)
```

Instalar y cargar el paquete "density" (si no lo has hecho ya)

```
install.packages("density")
```

```
library(density)
```

Construir el modelo KDE con diferentes anchos de banda

```
kde_datos_anchos_alto <- density(datos, bw = 30) # Ancho de banda alto
kde_datos_anchos_bajo <- density(datos, bw = 5) # Ancho de banda bajo
```

Graficar los modelos KDE

```
plot(kde_datos_anchos_alto, main = "KDE con Ancho de Banda Alto (30)", xlab = "Tiempo de Falla")
plot(kde_datos_anchos_bajo, main = "KDE con Ancho de Banda Bajo (5)", xlab = "Tiempo de Falla")
```

Función para generar muestras aleatorias de una KDE (aproximación)

```
generar_muestras_kde <- function(kde, n) { sample(kde$x, size = n, replace = TRUE, prob = kde$y) }
```

Generar 1000 muestras aleatorias del KDE (con ancho de banda alto)

```
set.seed(456) # Para reproducibilidad
muestras_kde <- generar_muestras_kde(kde_datos_anchos_alto, 1000)
```

Calcular y comparar estadísticas

```
media_kde <- mean(muestras_kde) sd_kde <- sd(muestras_kde)
cat("Media de las muestras KDE:", media_kde, "\n") cat("Desviación estándar de las muestras KDE:",
sd_kde, "\n")
```

Comparar con los datos originales (como en el ejercicio anterior)

““

Explicación de la Solución del Ejercicio 2:

- Se utilizan los mismos datos de tiempo de falla del Ejercicio 1.
- **Modelo KDE:** La función `density()` se utiliza para construir un modelo de estimación de densidad de kernel (KDE). El argumento `bw` controla el ancho de banda, que determina la suavidad de la estimación. Un ancho de banda más grande produce una curva más suave, mientras que un ancho de banda más pequeño puede capturar más detalles, pero también ser más sensible al ruido.
- **Graficar los Modelos KDE:** Los modelos KDE se visualizan usando la función `plot()`. Observa cómo el ancho de banda afecta la suavidad de la curva.
- **Generación de Muestras:** Como `density` no es una función de distribución directamente muestreable, se provee la función `generar_muestras_kde` para crear muestras aleatorias. Esto se hace creando una muestra de los valores `x` de la `kde`, con probabilidad proporcional al valor `y` de la KDE en ese punto.

5. Materiales Complementarios Recomendados:

- **Libros:**
 - "Simulation Modeling and Analysis" by Averill M. Law.
 - "Discrete-Event System Simulation" by Jerry Banks et al.
- **Artículos:**
 - Artículos sobre el uso de KDE y bootstrap en el modelado de datos de entrada.

- **Paquetes de R:**
 - `fitdistrplus`: Para ajuste de distribuciones paramétricas.
 - `boot`: Para análisis bootstrap.
 - `MASS`: Contiene funciones para KDE y otras técnicas estadísticas.
 - `dplyr`: Para manipulación de datos.