

# Contents

Clase 5: Modelado de Datos de Entrada Multivariados y Dependencias . . . . .	1
<b>Ejemplo: Ajustar una cópula Gaussiana</b>	<b>2</b>
<b>Cargar los datos</b>	<b>3</b>
<b>Análisis exploratorio</b>	<b>3</b>
<b>Ajuste de distribuciones marginales</b>	<b>3</b>
<b>Ajuste de cópulas</b>	<b>4</b>
<b>Comparación con AIC y BIC (pseudocódigo - necesita implementacion)</b>	<b>4</b>
<b>Simulación de datos</b>	<b>4</b>
<b>Validación</b>	<b>4</b>
“markdown	

## Clase 5: Modelado de Datos de Entrada Multivariados y Dependencias

### Objetivos de la Clase:

- Comprender la necesidad de modelar dependencias entre variables de entrada en simulaciones.
- Aprender técnicas para identificar y representar dependencias estadísticas.
- Aplicar cópulas para modelar dependencias entre distribuciones marginales en R.
- Validar modelos multivariados de datos de entrada.

### Contenido Teórico Detallado:

#### 1. Introducción a los Datos de Entrada Multivariados:

- En muchas simulaciones, las variables de entrada no son independientes. Ignorar las dependencias puede llevar a resultados de simulación incorrectos o engañosos.
- Ejemplos comunes:
  - Tiempo de llegada de clientes a un banco y el número de transacciones que realizan.
  - Tiempo de reparación de una máquina y el costo de las piezas de repuesto.
  - Volumen de ventas de diferentes productos en una tienda.
- La necesidad de modelar correctamente estas dependencias para una representación precisa del sistema.

#### 2. Identificación de Dependencias:

- **Análisis de Correlación:**
  - Coeficiente de correlación de Pearson (lineal).
  - Coeficiente de correlación de Spearman (monotónica).
  - Limitaciones de los coeficientes de correlación: solo capturan dependencias lineales o monotónicas.
- **Diagramas de Dispersión (Scatter Plots):**
  - Visualización de las relaciones entre pares de variables.
  - Identificación de patrones no lineales.
- **Medidas de Dependencia No Lineal:**
  - Información Mutua.
  - Distance Correlation.

#### 3. Cópulas:

- **Introducción al Concepto de Cópula:**
  - Una cópula es una función que une distribuciones marginales univariadas para formar una distribución conjunta multivariada.
  - El Teorema de Sklar establece que cualquier distribución multivariada puede ser expresada en términos de sus marginales univariadas y una cópula que describe la estructura de dependencia.
- **Tipos Comunes de Cópulas:**
  - **Cópula Gaussiana:** Adecuada para dependencias elípticas.
  - **Cópulas Arquimedianas:** (Clayton, Gumbel, Frank) - Capturan diferentes tipos de dependencia en las colas de la distribución.
    - \* Clayton: Dependencia fuerte en la cola inferior.
    - \* Gumbel: Dependencia fuerte en la cola superior.
    - \* Frank: Dependencia moderada en ambas colas.
- **Selección de la Cópula Apropriada:**
  - Análisis de la estructura de dependencia observada en los datos.
  - Pruebas de bondad de ajuste para cópulas.

#### 4. Implementación de Cópulas en R (Paquete `copula`):

- **Instalación del Paquete `copula`:** `R`      `install.packages("copula")`      `library(copula)`
- **Ajuste de Cópulas:**
  - Estimar los parámetros de la cópula (e.g., correlación en la cópula Gaussiana, parámetro de dependencia en cópulas arquimedianas) usando el método de Máxima Verosimilitud (MLE).

### Ejemplo: Ajustar una cópula Gaussiana

```
normal.cop <- normalCopula(param = 0.5, dim = 2, dispstr = "ex") # Define la cópula
fit.normal.cop <- fitCopula(normal.cop, data = your_data, method = "ml") # Ajusta la cópula
a los datos coef(fit.normal.cop) # Muestra el parámetro estimado “
```

- **Simulación de Datos Usando Cópulas:** `R`      # Simular datos de la cópula ajustada  
`u <- rCopula(1000, normal.cop)` # Genera valores uniformes basados en la cópula
- **Creación de Datos Multivariados Sintéticos:** `R`      # Transformar los valores uniformes a las distribuciones marginales deseadas      `x <- qexp(u[,1], rate = 1/10)` # Distribución exponencial marginal para la primera variable      `y <- qgamma(u[,2], shape = 2, rate = 1/5)` # Distribución gamma marginal para la segunda variable      `synthetic_data <- data.frame(x = x, y = y)` # Datos multivariados simulados

#### 5. Validación de Modelos Multivariados de Datos de Entrada:

- **Comparación Visual:**
  - Comparar diagramas de dispersión de los datos reales con los datos simulados.
- **Pruebas Estadísticas:**
  - Pruebas de bondad de ajuste para distribuciones marginales.
  - Pruebas para evaluar la estructura de dependencia (e.g., pruebas de hipótesis sobre los parámetros de la cópula).
- **Análisis de Sensibilidad:**
  - Evaluar el impacto de las dependencias modeladas en los resultados de la simulación.
  - Comparar los resultados de la simulación con y sin modelado de dependencias.

#### Ejemplos o Casos de Estudio:

1. Modelado del Tiempo de Llegada y Tiempo de Servicio en un Centro de Llamadas:

- Datos: Tiempo entre llegadas de llamadas y duración de las llamadas.
- Análisis: Observar si existe una correlación entre el tiempo entre llegadas y la duración de las llamadas. Por ejemplo, si las llamadas más largas tienden a llegar en grupos más cercanos.
- Modelado:
  - Ajustar distribuciones marginales a cada variable (e.g., exponencial para el tiempo entre llegadas, log-normal para la duración de las llamadas).
  - Ajustar una cópula para modelar la dependencia entre las variables.
  - Simular datos usando la cópula y las distribuciones marginales.
- Validación: Comparar los diagramas de dispersión y realizar pruebas de bondad de ajuste.

## 2. Modelado de la Demanda de Múltiples Productos en una Tienda:

- Datos: Datos de ventas diarias de diferentes productos.
- Análisis: Investigar si la demanda de algunos productos está correlacionada (e.g., la demanda de café y leche).
- Modelado:
  - Ajustar distribuciones marginales a la demanda de cada producto.
  - Usar cópulas para modelar las dependencias entre las demandas de diferentes productos.
- Validación: Comparar las matrices de correlación y realizar pruebas de bondad de ajuste.

## Problemas Prácticos o Ejercicios con Soluciones:

### 1. Ejercicio:

- **Datos:** Se proporciona un conjunto de datos que contiene información sobre el tiempo de espera en una fila y el tiempo de servicio para cada cliente.
- **Tarea:**
  - Cargue los datos en R.
  - Realice un análisis exploratorio para identificar posibles dependencias entre el tiempo de espera y el tiempo de servicio (diagrama de dispersión, coeficiente de correlación).
  - Ajuste distribuciones marginales para el tiempo de espera y el tiempo de servicio utilizando `fitdistrplus`.
  - Ajuste una cópula Gaussiana y una cópula Clayton a los datos utilizando el paquete `copula`.
  - Compare los resultados de los ajustes utilizando criterios de información (AIC, BIC).
  - Simule nuevos datos utilizando la cópula mejor ajustada y las distribuciones marginales.
  - Valide el modelo comparando gráficamente los datos simulados con los datos originales.
- **Solución (Esquema):**

“R

## Cargar los datos

```
data <- read.csv("your_data.csv") # Reemplaza "your_data.csv" con el nombre de tu archivo
wait_time <- data$wait_time
service_time <- data$service_time
```

## Análisis exploratorio

```
plot(wait_time, service_time, main = "Diagrama de Dispersión")
cor(wait_time, service_time) # Coeficiente de correlación de Pearson
```

## Ajuste de distribuciones marginales

```
library(fitdistrplus)
fit_wait <- fitdist(wait_time, "gamma")
fit_service <- fitdist(service_time, "lognormal")
```

## Ajuste de cópulas

```
library(copula) normal.cop <- normalCopula(dim = 2) clayton.cop <- claytonCopula(dim = 2)
fit_normal <- fitCopula(data.frame(pobs(wait_time), pobs(service_time)), normal.cop, method
= "ml") fit_clayton <- fitCopula(data.frame(pobs(wait_time), pobs(service_time)), clayton.cop,
method = "ml")
```

## Comparación con AIC y BIC (pseudocódigo - necesita implementación)

```
AIC(fit_normal) BIC(fit_normal) AIC(fit_clayton) BIC(fit_clayton)
```

## Simulación de datos

```
copula_to_use <- ifelse(AIC(fit_normal) < AIC(fit_clayton), normal.cop, clayton.cop)
#Pseudocódigo - elija el mejor modelo copula_fit_to_use <- ifelse(AIC(fit_normal) <
AIC(fit_clayton), fit_normal, fit_clayton)
u <- rCopula(1000, copula_to_use) # simular datos usando la mejor cópula
sim_wait <- qgamma(u[,1], shape = fit_wait$estimate["shape"], rate = fit_wait$estimate["rate"])
# transformar a la marginal ajustada sim_service <- qlnorm(u[,2], meanlog = fit_service$estimate["meanlog"],
sdlog = fit_service$estimate["sdlog"])
sim_data <- data.frame(wait_time = sim_wait, service_time = sim_service)
```

## Validación

```
plot(wait_time, service_time, main = "Datos Reales") plot(sim_data$wait_time, sim_data$service_time,
main = "Datos Simulados") ““
```

### Materiales Complementarios Recomendados:

- **Libros:**
  - "Simulation Modeling and Analysis" by Averill M. Law
  - "Handbook of Simulation: Modeling and Theory" by Jerry Banks
  - "An Introduction to Copulas" by Roger B. Nelsen
- **Artículos Científicos:** Buscar artículos sobre modelado de datos de entrada multivariados y aplicaciones de cópulas en simulaciones en bases de datos como IEEE Xplore, ScienceDirect y JSTOR.
- **Paquetes de R:**
  - **fitdistrplus:** Para el ajuste de distribuciones univariadas.
  - **copula:** Para el modelado con cópulas.
  - **VineCopula:** Para cópulas vine (una extensión de cópulas).
- **Tutoriales en Línea:** Buscar tutoriales sobre el uso de cópulas en R para el modelado de datos de entrada.