

Universidad Europea de Madrid  
Escuela de Arquitectura, Ingeniería y Diseño

---

# APUNTES INTELIGENCIA ARTIFICIAL

GRADO EN INGENIERÍA INFORMÁTICA

INTELIGENCIA ARTIFICIAL

---

**Profesor: Isabel Fuentes**

Daniel Cuesta Sanz de Madrid

22 de diciembre de 2024



# Tema 1

El tema 1 es una introducción a la inteligencia artificial así como una breve historia de la misma. Se explican los conceptos básicos y se introducen los sistemas inteligentes.

## 1.1. ¿Qué es la inteligencia artificial?

La inteligencia artificial no tiene una definición exacta como tal, pero se puede entender como la capacidad de una máquina para imitar las funciones cognitivas humanas, tales como el aprendizaje y la resolución de problemas.

### 1.1.1. Historia de la IA

La inteligencia artificial ha tenido (por ahora) una historia muy corta, se podría dividir su vida hasta hoy en 5 etapas principales:

- **1943-1956 Genesis de la IA:** Surge el término 'Inteligencia Artificial' y los primeros computadores y juegos de ajedrez.
- **1952-1969 Entusiasmo Inicial:** Numerosos proyectos surgen, y se crean nuevas herramientas.
- **1966-1973 Etapa de crisis:** Decepción ante los resultados en los últimos años, se redimensionan los problemas.
- **1969-1979 Resurgimiento:** Se cambia el enfoque por completo, ya no se intentan resolver problemas generales, sino que solo los de dominios concretos.
- **1980-2024 Industria Actual:** La última frontera de la IA, se invierte más dinero que nunca en la tecnología.

## 1.2. ¿Qué es un sistema inteligente?

Un sistema inteligente es aquel que puede percibir su entorno y tomar decisiones para maximizar sus posibilidades de éxito en alguna tarea o conjunto de tareas. Estos sistemas pueden ser diseñados para aprender y adaptarse a nuevas situaciones, mejorando su rendimiento con el tiempo. Los sistemas inteligentes pueden ser clasificados en varias categorías, tales como:

- **Sistemas basados en conocimiento:** Utilizan un conjunto de reglas predefinidas para tomar decisiones. Son fáciles de entender y diseñar, pero tienen limitaciones en cuanto a su capacidad de adaptación.
- **Sistemas basados en aprendizaje:** Utilizan algoritmos de aprendizaje automático para mejorar su rendimiento a partir de datos. Estos sistemas pueden adaptarse a nuevas situaciones y mejorar con el tiempo.
- **Sistemas híbridos:** Combinan características de los sistemas basados en reglas y los sistemas basados en aprendizaje para aprovechar las ventajas de ambos enfoques.

Un ejemplo de sistema inteligente es un asistente virtual, como Siri o Alexa, que puede entender comandos de voz, buscar información en internet, y realizar tareas como enviar mensajes o configurar recordatorios. Estos asistentes utilizan técnicas de procesamiento de lenguaje natural y aprendizaje automático para mejorar su capacidad de entender y responder a las solicitudes de los usuarios.

**1.2.1. Sistema basado en el conocimiento**

Un sistema basado en el conocimiento presenta una arquitectura con tres componentes principales que interactúan entre ellos para conseguir un objetivo:

1. **Motor de inferencia:** Se encarga del control del problema buscando en la base de conocimiento.
2. **Base de Conocimiento:** Almaneca las reglas de conocimiento
3. **Base de hechos:** Es la que contiene los datos iniciales del problema

En base a estos tres componentes, el motor de inferencia y la base de conocimiento va creando nuevos hechos que luego se almacenan en la base de hechos.

**1.3. Aplicaciones de la IA**

La IA tiene muchas aplicaciones posibles, algunas de las cuales se detallan en la Tabla 1.1. Estas aplicaciones abarcan desde el control avanzado de sistemas hasta la minería de datos, pasando por el diagnóstico y reparación, los agentes inteligentes y el asesoramiento. Cada una de estas categorías tiene casos de uso específicos que demuestran el potencial de la inteligencia artificial para transformar diversas industrias y mejorar la eficiencia y efectividad de numerosos procesos.

Categoría	Casos de uso
Control avanzado de sistemas	Unidades de transporte ferroviario Planificación de procesos complejos Transbordador espacial
Diagnóstico y reparación	Medicina Locomotoras eléctricas Detección de yacimientos
Agentes Inteligentes	Comercio electrónico automatizado Videojuegos
Asesoramiento	Entidades de crédito Detección de fraudes
Minería de datos	Modelado financiero Astronomía Biología

Cuadro 1.1: Aplicaciones de la IA

# Tema 2

El tema 2 sirve como una introducción un poco mas detalladas a los sistemas inteligentes y los agentes inteligentes.

## 2.1. Agentes inteligentes

Los agentes inteligentes son un *subset* de los sistemas inteligentes que tienen la capacidad de percibir su entorno y actuar de manera autónoma para alcanzar sus objetivos. Estos agentes pueden ser simples o complejos, y su comportamiento puede ser predefinido o aprendido a través de la experiencia. Estos agentes inteligentes no tienen porque ser físicos o virtuales, ya que pueden ser ambos. La arquitectura de un sistema inteligente normalmente tiene la siguiente estructura:

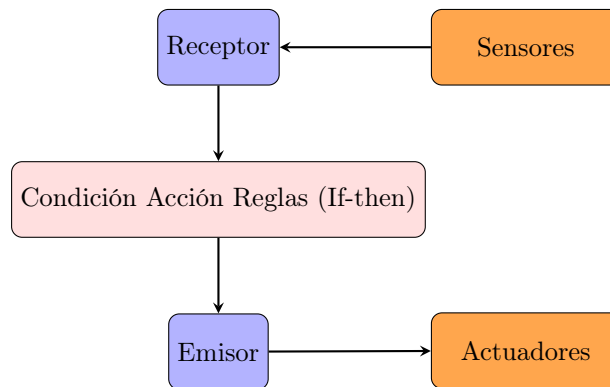


Figura 2.1: Arquitectura agentes inteligentes

Dicho de la manera más sencilla, un agente inteligente es: Un sistema informático que interpreta su **entorno** mediante **sensores** e interactúa con él usando **actuadores**. Este entorno como he dicho antes puede ser virtual o físico, ya sea un salón para una aspiradora robótica, un proyecto para el copilot... E igual con los sensores, de la misma manera que con los actuadores.

### 2.1.1. Características de los agentes inteligentes

Las características principales de los agentes inteligentes incluyen:

- **Reactividad:** La capacidad de un agente para percibir su entorno y responder a los cambios en tiempo real.
- **Proactividad:** La capacidad de un agente para tomar la iniciativa y actuar de manera autónoma para alcanzar sus objetivos.
- **Adaptabilidad:** La capacidad de un agente para aprender de la experiencia y mejorar su comportamiento a lo largo del tiempo.
- **Comunicación:** La capacidad de un agente para interactuar y colaborar con otros agentes o sistemas para lograr objetivos comunes.
- **Autonomía:** La capacidad de un agente para operar sin intervención humana directa, tomando decisiones por sí mismo.

- **Racionalidad:** La capacidad de un agente para tomar decisiones basadas en la lógica y la información disponible para maximizar sus objetivos.

### 2.1.2. Tipos de entornos

Los entornos pueden ser de muchos tipos, y es además el aspecto más importante de un sistema inteligente, ya que este luego decidirá los sensores y actuadores que necesita, además de todas las acciones que puede hacer o hace. Los principales entornos que existen son:

- **Entorno completamente observable vs. parcialmente observable:** En un entorno completamente observable, el agente tiene acceso a toda la información necesaria para tomar decisiones. En un entorno parcialmente observable, el agente solo tiene acceso a una parte de la información.
- **Entorno determinista vs. estocástico:** En un entorno determinista, las acciones del agente determinan de manera precisa el estado siguiente del entorno. En un entorno estocástico, hay un grado de incertidumbre en los resultados de las acciones del agente.
- **Entorno estático vs. dinámico:** En un entorno estático, el entorno no cambia mientras el agente está decidiendo su acción. En un entorno dinámico, el entorno puede cambiar mientras el agente está decidiendo su acción.
- **Entorno discreto vs. continuo:** En un entorno discreto, hay un número finito de estados y acciones posibles. En un entorno continuo, los estados y acciones pueden variar de manera continua.
- **Entorno episódico vs. secuencial:** En un entorno episódico, la experiencia del agente se divide en episodios independientes. En un entorno secuencial, las decisiones del agente afectan estados futuros y se deben considerar a largo plazo.
- **Entorno conocido vs. desconocido:** En un entorno conocido, el agente tiene conocimiento previo de las reglas del entorno. En un entorno desconocido, el agente debe aprender las reglas a través de la interacción.

# Tema 3

El tema 3 es una introducción a la minería de datos, especialmente técnicas como el web scrapping y además el aprendizaje automático ligado a estas técnicas.

## 3.1. Introducción a la minería de datos y aprendizaje automático

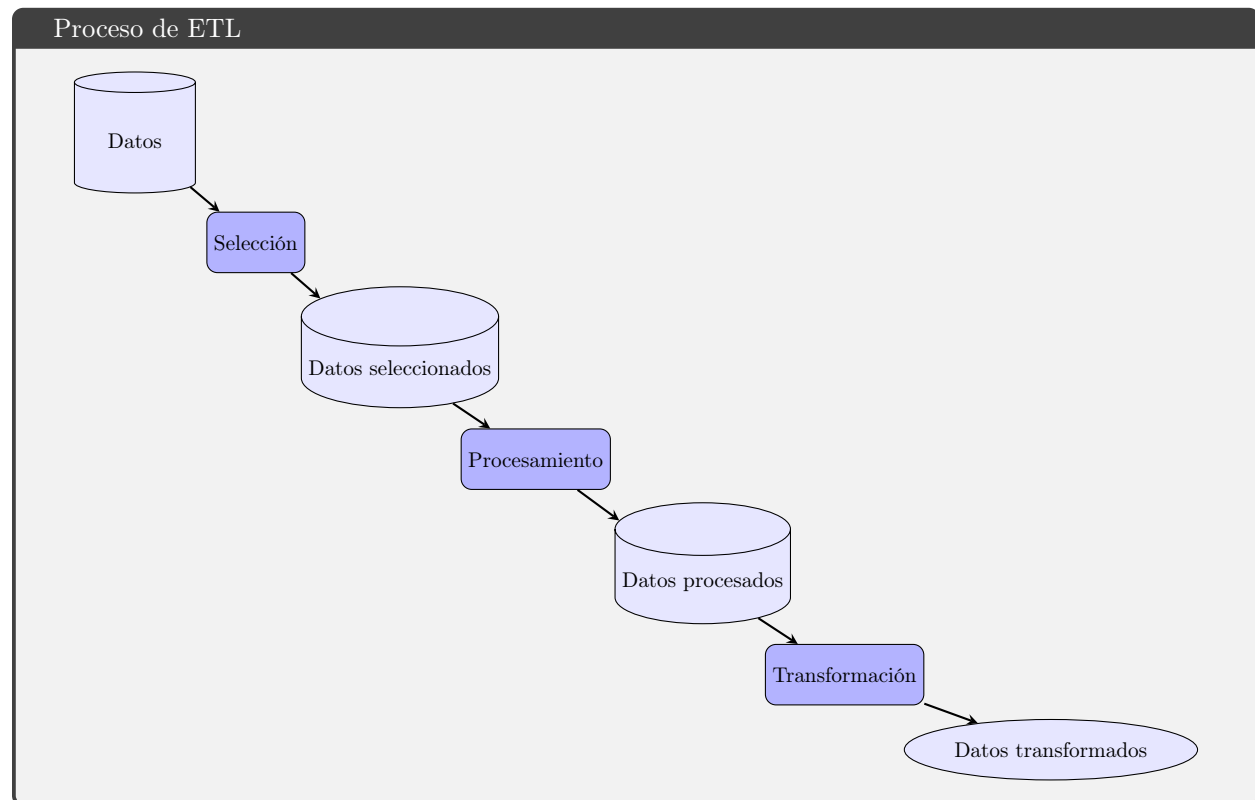
### 3.1.1. Minería de datos

El proceso de minería de datos es un proceso iterativo que consiste en descubrir patrones en grandes volúmenes de datos, inicialmente se encuentran procesos para preparar y limpiar los datos. El proceso de descubrimiento tiene las siguientes fases:

- **Selección:** Se seleccionan los datos que se van a utilizar.
- **Procesamiento:** Se procesan y limpian para que puedan ser usados de una manera eficiente.

### 3.1.2. Extracción, transformación y carga de datos (ETL)

Básicamente, el proceso ETL se encarga de extraer datos de diferentes fuentes, transformarlos y cargarlos en un almacén de datos. Esta será una palabra que se escuchará mucho en el mundo de la minería de datos, ya que es un proceso fundamental para este.



### 3.1.3. Aprendizaje automático

El aprendizaje automático es una rama de la inteligencia artificial que permite a las máquinas aprender de los datos y hacer predicciones o tomar decisiones sin ser programadas explícitamente para ello. Existen varios tipos de aprendizaje automático:

- **Aprendizaje supervisado:** Se entrena un modelo con datos etiquetados, es decir, datos que ya tienen una respuesta conocida.
- **Aprendizaje no supervisado:** Se entrena un modelo con datos no etiquetados y el objetivo es encontrar patrones o estructuras ocultas en los datos.
- **Aprendizaje semi-supervisado:** Combina una pequeña cantidad de datos etiquetados con una gran cantidad de datos no etiquetados durante el entrenamiento.
- **Aprendizaje por refuerzo:** Un agente aprende a tomar decisiones mediante la interacción con un entorno y la obtención de recompensas o castigos.

## 3.2. Extracción, transformación y carga de datos

Antes de realizar web scrapping; **¿Es legal?**, sorprendentemente sí, pero siguiendo diferentes reglas y normas:

- La información tiene que ser pública.
- Puedo hacer uso de los datos siempre que no lesione los derechos de autor.
- Tienes que aceptar explícitamente los términos de uso.

### 3.2.1. APIs

Una API es un canal de comunicación, que permite que 2 software se comuniquen entre sí. Tiene 4 tipos diferentes de peticiones:

#### Peticiones API

- **GET:** Se utiliza para obtener datos de un servidor.
- **POST:** Se utiliza para enviar datos a un servidor.
- **PUT:** Se utiliza para actualizar datos en un servidor.
- **DELETE:** Se utiliza para eliminar datos de un servidor.

Estas peticiones se realizan contra endpoints, que son formas de interactuar con este. También cabe destacar **REST**, que es un estilo de arquitectura de software que define un conjunto de restricciones para crear servicios web.

#### Ejemplo:

```
import requests
url = 'https://pokeapi.co/api/v2/pokemon/ditto'

response = requests.get(url)
print(response.json())
```

En el ejemplo de encima hacemos un request a la API de Pokemon, y obtenemos la información del Pokemon Ditto. Dependiendo de la respuesta que nos encontremos, esta nos informa de si la petición ha sido correcta o no. Estas respuestas se dividen en diferentes códigos de estado:

- 1xx: Información
- 2xx: Éxito
- 3xx: Redirección
- 4xx: Error del cliente
- 5xx: Error del servidor

Pero aquí nos encontramos un problema, ya que no es común que las APIs tengan una documentación clara y concisa, o directamente ni siquiera ni nos permitan hacer uso de ellas. Para acceder a una API que no tiene una documentación clara, podemos hacer uso de la técnica de web scrapping o parsear el HTML directamente con BS4.

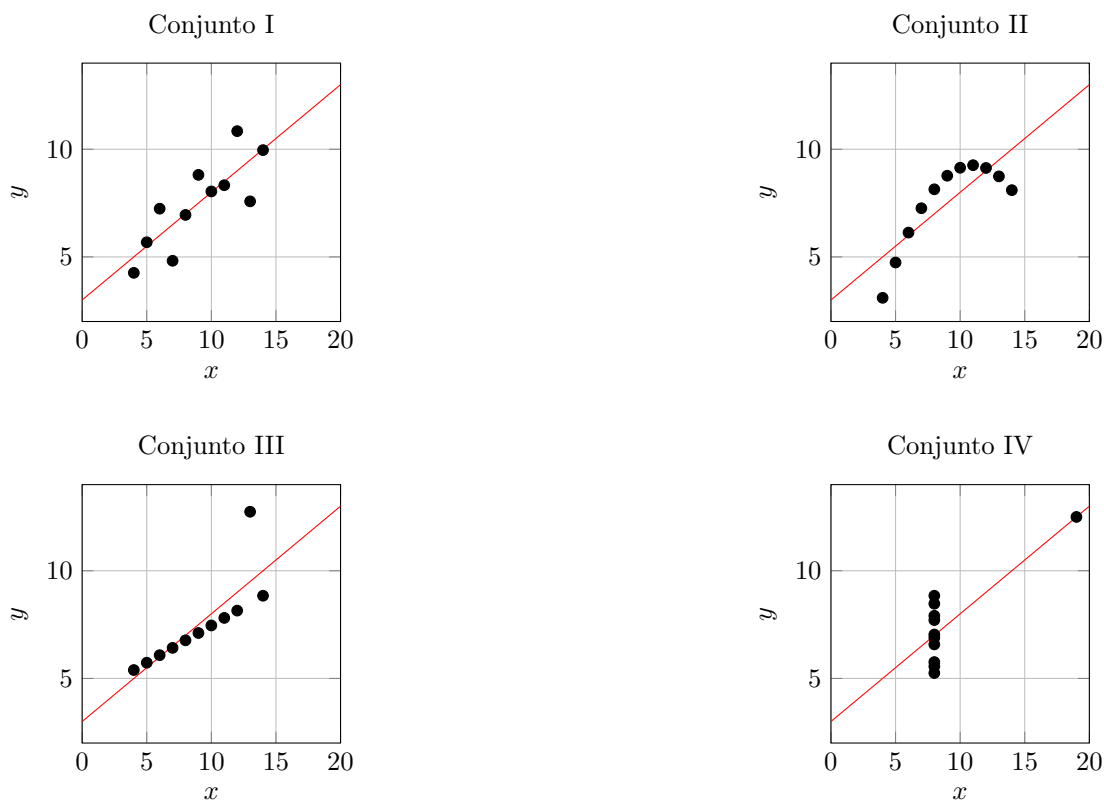
### Ejemplo con BS4:

```
#Ejemplo con BeautifulSoup y la wikipedia de la UEM
from bs4 import BeautifulSoup

url = 'https://es.wikipedia.org/wiki/Universidad_Europea_de_Madrid'
response = requests.get(url)
soup = BeautifulSoup(response.content, "html.parser")
```

## 3.3. Representación y visualización de datos

La representación de datos es una parte fundamental en la minería de datos, ya que nos permite visualizar los datos y darles un enfoque diferente. Ya que aunque un análisis pueda tener datos muy similares, una vez visualizados estos pueden tener un enfoque diferente. Esto, por ejemplo, se hace muy fácil de ver en el **Cuarteto de Anscombe**:





Todos estos gráficos tienen datos muy similares (media, varianza, correlación...) pero sus representaciones gráficas son totalmente diferentes. Esto nos demuestra la importancia de la representación de datos para poder comprender una función o un conjunto de datos.

# Tema 4

El tema 4 por fin introduce la inteligencia artificial, comenzando por el aprendizaje supervisado

## 4.1. Introducción al machine learning

El machine learning, o aprendizaje automático, es una rama de la inteligencia artificial que se centra en el desarrollo de algoritmos y técnicas que permiten a las computadoras aprender de y hacer predicciones sobre datos. A través del uso de modelos matemáticos y estadísticos, las máquinas pueden identificar patrones y tomar decisiones con mínima intervención humana.

Todo el proceso tiene **5 pasos principales**:

1. Recoger Datos.
2. Preprocesamiento de datos
3. Elegir el mejor modelo.
4. Entrenar y ajustar los parámetros del modelo.
5. Ver si los resultados del modelo predicen correctamente nuevos datos.

### 4.1.1. Recoger Datos

Los datos se pueden obtener de muchas maneras: Mediante web scraping, desde una API o base de datos... Pero como vamos a realizar aprendizaje supervisado necesitamos **datos etiquetados**.

age	gender	height	weight	ap_hi	ap_lo	cholest	gluc	smoke	alco	cardio
46.0	F	172	112	120	80	1	1	0	0	YES
44.0	M	170	69	120	70	1	1	0	1	NO
45.5	M	159	49	120	70	1	1	0	0	NO
39.7	F	164	48	110	70	1	2	1	1	YES
63.3	F	180	104	120	85	2	2	0	0	NO

Cuadro 4.1: Ejemplo de datos etiquetados

Por ejemplo en la tabla 4.1 tenemos datos de pacientes y si han tenido un ataque al corazón o no. Los datos etiquetado se elegirán dependiendo de la variable que queramos predecir:

$$\{(x[i], y[i]) \mid x[i] \in \mathbb{R}^n, y[i] \in \{c_1, c_2, \dots, c_m\}, i = 1, \dots, N\}$$

Todo esto con el objetivo de generar un modelo acorde a la fórmula:  $\hat{Y} = f(x, w)$  Siendo  $x_n$  cada variable y  $w$  los pesos de cada variable.

### 4.1.2. Preprocesamiento de datos

El preprocesamiento de datos es una etapa crítica en el proceso de machine learning, ya que los datos sin procesar pueden contener errores, valores atípicos o información redundante que puede afectar la precisión de los modelos. Algunas técnicas comunes de preprocesamiento de datos incluyen la limpieza de datos, la normalización de datos y la selección de características. A continuación se detallarán las técnicas de preprocesamiento a usar.

#### Missing Values

Primero se mira si hay *missing values*. Dependiendo del tamaño del dataset, se puede eliminar las muestras del dataset (si es muy grande) o intentar predecir los con otros modelos para datasets pequeños.

#### Visualizar para ver si hay outliers

Los *outliers* son valores atípicos que pueden afectar negativamente el rendimiento de los modelos de machine learning. Se pueden visualizar mediante gráficos de caja y bigotes, histogramas o diagramas de dispersión. Las maneras más típicas de visualizar para ver si hay outliers es:

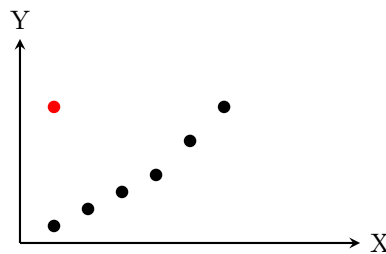


Figura 4.1: Visualización de los datos

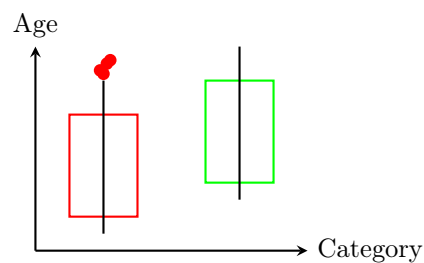


Figura 4.2: Box-plot

Frente a los outliers de las figuras (destacados en rojo), ¿Qué se puede hacer? Si son varios hay que verificar si hay datos mal recogidos, o que se interpretan mal en la base de datos, con pocos outliers puede ser:

- Casos con comportamiento distinto pero con razón: ***True Informative Outliers***
- Casualidad: ***False informative Outliers***
- Que no tengamos suficientes datos y sean normales.

**Codificar la variables categóricas**

Las variables categóricas son aquellas que representan una categoría o clase en lugar de un valor numérico. Para poder utilizar estas variables en los modelos de machine learning, es necesario codificarlas en un formato numérico. Algunas técnicas comunes de codificación de variables categóricas incluyen la codificación one-hot y la codificación ordinal. El problema que tenemos con esto es el *skewnewss* o asimetría, que es una medida de la simetría de la distribución de los datos. Una distribución puede ser simétrica, sesgada a la derecha (positiva) o sesgada a la izquierda (negativa). A continuación se muestran ejemplos de distribuciones con diferentes tipos de asimetría.

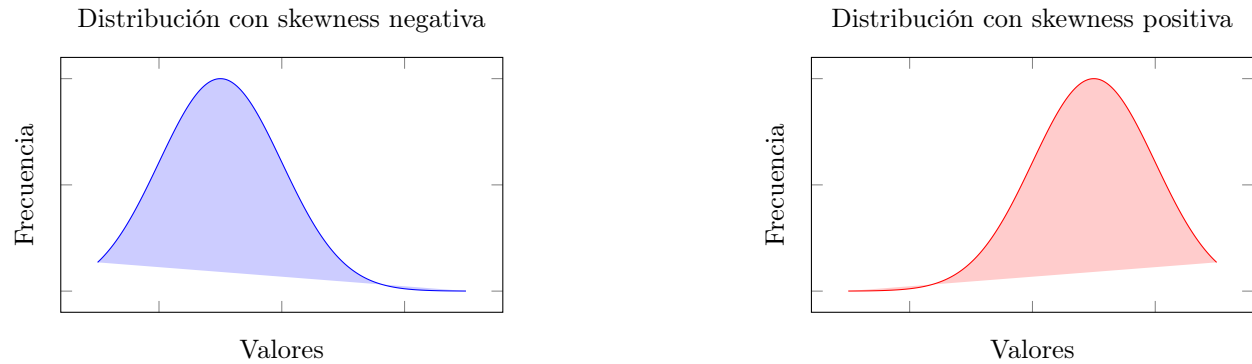


Figura 4.3: Ejemplos de distribuciones con skewness negativa (izquierda) y positiva (derecha)

