# Instituto Politécnico Nacional

## Centro de Investigación en Cómputo

**Python en el Ámbito Científico**

Profesor: Alan Badillo Salas (alan@nomadacode.com)

## Manual de prácticas

## Temas

— **Estadística Descriptiva**
— **Visualización de Datos**
— **Regresión Lineal**
— **Regresión Logística**
— **Redes Neuronales**

Julio 26, 2025.

# Introducción

En el curso de Python Científico con Aplicaciones en el Ámbito Científico se han revisado diferentes temas relacionados a la Ciencia de Datos para aplicarlos en problemas y fenómenos reales relacionados a la ciencia y otras áreas industriales y empresariales.

Los problemas en la ciencia de datos parten del planteamiento de un fenómeno a través de los datos, en un flujo de trabajo que consiste en adquirir, sintetizar, procesar y visualizar los datos. Y para analizar los datos se utilizan diferentes técnicas y herramientas que usan modelos matemáticos y computacionales para obtener resultados automáticos y sobre el aprendizaje de los datos.

La **Estasdística Descriptiva** permite explicar los datos a través de los valores identificados dentro de los datos y los indicadores sobre estos valores, como la distribución de una variable, su valor central y los rangos proporcionales a una proporción de la muestra.

La **Visualización de Datos** permite entender el comportamiento estadístico de una forma más natural, por ejemplo, visualizar la distribución de una variable mediante una función de densidad o un histograma, ver la caja de sobre los rangos proporcionales y las correlaciones entre varios ejes de datos.

La **Regresión Lineal** permite predecir un valor para una variable de respuesta continua que tenga una fuerte correlación a otras covariables que la expliquen, por ejemplo, explicar el precio de la gasolina a partir del precio del dólar. A estos problemas se les conoce como problemas de regresión.

La **Regresión Logística** permite predecir una variable de tiene una respuesta binaria o probabilística y que tenga un comportamiento de activación respecto a las variables que la activan, por ejemplo, si una persona tiene trabajo a partir de su edad. A estos problemas se les conoce como problemas de clasificación.

La **Redes Neuronales** permite predecir variables de respuesta continuas o binarias por lo que resuelven problemas de regresión y clasificación según la función de activación marcada en su capa de salida. Por ejemplo, podrían aprender a predecir valor continuos como el precio de una casa a partir de las características de la casa o si habrá un fraude o no habrá fraude usando las características del cliente.

En este manual de prácticas desarrollaremos las habilidades necesarias para resolver un caso de uso aplicado al ámbito cientítifico, industrial y empresarial para cada tema, estructurando el problema en paso a paso para llegar a un reporte que explique y resuelva el problema planteado.

# Instrucciones

Selecciona una práctica según tu tema de interés y desarrolla paso a paso el procedimiento planteado para llegar a los mismos resultados de la práctica. En una Libreta de Python sigue el paso a paso hasta tener todos los pasos completados.

Al final de la práctica deberás entregar una Libreta de Python o su equivalente en formato PDF que deberá contener los siguientes puntos:

1. **Portada** - Debe incluir el nombre del Instituto Politécnico Nacional y el Centro de Investigación en Cómputo, así como el nombre del profesor y del estudiante, la fecha y ciudad (por ejemplo, Julio 2025. Ciudad de México.). También se deberá incluir el tema seleccionado para la práctica. Si usas Colab o Jupyter podrías poner:

```
# Instituto Politécnico Nacional

## Centro de Investigación en Cómputo

### Departamento de Diplomados y Extensión Profesional

Profesor: XXXX XXXX XXXX

Alumno: XXXX XXXX XXXX

Julio 2025. Ciudad de México.

## Práctica - Visualización de Datos
```

2. **Introducción** - Deberá incluir una breve introducción que describa lo que hiciste en la práctica, esta deberá escribirse al final ya que esté resuelta, por ejemplo, de qué se trata la práctica, cuales son los pasos más relevantes que se resolvieron y qué aplicaciones en la ciencia o la industria tiene.

3. **Desarrollo** - Deberá incluir el desarrollo de la práctica paso a paso, documentando en cada paso lo que se hace antes de cada celda de código, por ejemplo, si se importan librerías, si se adquieren datos, si se crea una gráfica o si se aplica un modelo de entrenamiento.

4. **Conclusiones** - Deberá incluir una breve síntesis de lo que se hizo en general en la práctica, resaltando la utilidad y las aplicaciones directas que encuentras en tu entorno social o de trabajo, por ejemplo, explicar cómo se logró hacer la visualización de datos, qué resultados se obtuvieron y el problema en tu trabajo que se podría resolver usando esas técnicas similares, por ejemplo, visualizar cuántos clientes están satisfechos con el servicio que les brindó la empresa por mes y separados por grupo de edades.

## Comentarios

Cada práctica de cada tema ya está resuelta paso a paso, por lo que solo se deben reconstruir los mismos resultados, usando tu propio estilo de código y formatos o adaptaciones que consideres necesarias. Se recomienda elegir el tema que te sea de mayor interés y centrarte en esa práctica. Después tendrás tiempo para hacer más prácticas, pero es recomendable hacer solo una para entregarla a tiempo.

Opcionalmente, puedes aplicar los conocimientos a un proyecto para tu portafolio, donde deberás buscar un conjunto de datos en internet y proceder con su análisis similar al que se realizó en la práctica. Por ejemplo, si hiciste la práctica de regresión lineal, se recomienda analizar las dos variables que tengan mayor correlación en el conjunto de datos que encontraste en internet y limitarlo a conseguir el ajuste de los coeficientes de regresión. Intenta plantear el problema más sencillo posible que se apegue a lo aprendido en la práctica. Después tendrás tiempo de mejorar tu proyecto u obtener otros resultados.

Una vez que tengas las libretas con tu práctica, ponles la portada y las conclusiones como mínimo, luego descarga la libreta como PDF (puedes darle imprimir y guardar como PDF).

Envia los documentos PDF por correo con el asunto: **PyCien Julio 2025 - Práctica**, no es necesario incluir cuerpo en el correo, pero sería útil poner tus comentarios sobre qué te pareció el curso y la práctica en general (¿Qué mejorarías?).

# Práctica — Estadística Descriptiva

En esta práctica desarrollaremos las habilidades necesarias para analizar de forma estadística un conjunto de datos. Buscaremos entender cuáles son las variables asociadas al conjunto de datos, la naturaleza de cada variable y los estadísticos principales de cada variable según su naturaleza.

## Teoría

Al analizar un conjunto de datos nos encontramos con dos tipos de variables principales:

- **Variable Numérica** - Es una variable cuantitativa que representa un valor numérico asociado al objeto de análisis y que puede ser continuo o entero, pueden ser positivas, negativas o amboas. En las variables numéricas se estudia el rango (mínimo y máximo), el valor medio y los valores de sus cuartiles. Por ejemplo, el peso de un producto, la edad de una persona o el ingreso de un cliente.
- **Variable categótica** - Es una variable cualitativa que representa una característica asociada al objeto de análisis y que se representa como un valor categórico que puede ser binario o multiclase. Estas variables se separan en más variables binarias para cada clase o categoría y se analizan de forma individual. Por ejemplo, si la variable tiene dos clases (hombres y mujeres) se construye una variable binaria que indica si es hombre y otra variable binaria que indica si es mujer. Si la variable tiene más clases se hace lo mismo para cada clase, poniendo en esa variable derivada un 1 si pertenece a la clase o 0 si no pertenece a la clase.

Para el análisis estadístico de las variables numéricas construiremos un reporte sobre sus estadísticos principales:

- **Total** - Total de datos en la muestra, se representa por $n$.
- **Suma** - Suma del valor de todos los datos en la muestra, se representa por $S$.
- **Media o Promedio** - Promedio del valor de todos los datos, se representa por $\overline{x} = \sum_{i=1}^{n} x_i$
- **Cuartil 1 o Percentil 25** - Representa el valor que alcanza el 25% de todos los valores ordenados de la muestra del menor al mayor, se representa por $Q_1$
- **Cuartil 2, Percentil 25 o Mediana** - Representa el valor que alcanza el 50% de todos los valores ordenados de la muestra del menor al mayor, se representa por $Q_2$

- **Cuartil 1 o Percentil 75** - Representa el valor que alcanza el 75% de todos los valores ordenados de la muestra del menor al mayor, se representa por $Q_3$
- **Mínimo** - Representa el valor mínimo de todos los datos en la muestra, se representa por $x_{min}$
- **Máximo** - Representa el valor mínimo de todos los datos en la muestra, se representa por $x_{max}$

Con estos estadísticos podemos construir una tabla de datos eje por eje (para cada variable) y darnos una idea del espacio donde viven.

Para el análisis estadístico de las variables categóricas, primero separaremos cada variable categórica en más variables binarias que se represeten con 1 si pertenece a la categoría o clase y con 0 sino, estas variables aumentarán rápidamente, por ejemplo, si hay 12 categorías, tendremos 12 variables binarias adicionales.

Para las variables binarias, usaremos un análisis diferente y solo estimaremos las proporciones de 1s y 0$s$ como estadísticos principales:

- **Total** - Total de datos en la muestra, se representa por $n$.
- **Suma de** 1**s** - Suma del valor de todos los datos en la muestra que valen 1, se representa por $S_1$.
- **Suma de** 0**s** - Suma del valor de todos los datos en la muestra que valen 0, se representa por $S_0 = n - S_1$.
- **Porcentaje de** 1**s** - Representa la proporción de 1s en la muestra, se representa por $P_1 = S_1/n$
- **Porcentaje de** 0**s** - Representa la proporción de 0s en la muestra, se representa por $P_0 = S_0/n$

Aunque para las variables binarias los estadísticos son menos, los porcentajes de 1s y 0s se asocian a la probabilidad de pertener o no pertenecer a la categoría asociada. Por ejemplo, si $P_1 = 75\%$ y la variable está asociada a si es hombre podría significar que hay 75% de hombres.

## Planteamiento del Problema

En esta práctica analizaremos los datos de los ataques al corazón mediante la construcción de variables continuas (para los ejes de datos numéricos enteros y decimales) y binarias (para los ejes de datos categóricos).

## Desarrollo

El desarrollo consiste en adquirir el conjunto de datos desde el repositorio, inspeccionar las variables categóricas, construir cada una de las variables de análisis, generar los estadísticos para cada tipo de variable y construir los reportes sobre los estadísticos de cada variable.

### Paso 1 - Aquisición del conjunto de datos

Primero obtendrémos los datos de ataques al corazón directamente desde el repositorio y mostraremos las primeras 5 muestras.

```
import pandas

url = "https://github.com/dragonnomada/ipn-cic-pycien-abril-2025/raw/refs/heads/main/datasets/practicas/heart_attack.csv"

heart_attack = pandas.read_csv(url)

heart_attack.head()
```

**Código** 1: Adquisición del conjunto de datos de ataques al corazón

Preguntas:

1. ¿Cuántas columnas se observan?
2. ¿Qué tipo de valores tiene cada columna?

### Paso 2 - Información general del conjunto de datos

Como segundo paso mostraremos la información general sobre el conjunto de datos.

```
heart_attack.info()
```

**Código** 2: Información general del conjunto datos de ataques al corazón

Preguntas:

1. ¿Cuántos registros hay?
2. ¿Cuántas columnas totales hay?
3. ¿Cuáles columnas tienen datos enteros?
4. ¿Cuáles columnas tienen datos decimales?
5. ¿Cuáles columnas tienen datos de texto?
6. ¿Cuánta memoria usa el conjunto de datos?

**Paso 3 - Inspeccionar los ejes de datos categóricos**

Dado que existen dos ejes de datos categóricos, inspeccionaremos cada uno obtener las clases o categorías asociadas en cada eje. Así podremos construir una variable binaria para cada clase o categoría en cada eje de datos.

```
heart_attack["Gender"].unique()
```

**Código** 3: Clases asociadas al género

```
heart_attack["Result"].unique()
```

**Código** 4: Clases asociadas al resultado

Preguntas:

1. ¿Cuántos clases hay para el género y con qué valores?
2. ¿Cuántas clases hay para el resultado y con qué valores?
3. ¿Cuántas variables totales se derivaran de esas clases?

**Paso 4 - Construir las variables de análisis**

Ahora que entendemos todos los ejes de datos numéricos (enteros y decimales) y los ejes de datos categóricos (y sus clases o categorías), podemos construir las variables de análisis derivadas, para cada eje de datos numérico tendremos una variable de análisis continua y para cada categoría en cada eje de datos categórico tendremos una variable de análisis binaria.

```
# Edad - Enteros
x1 = heart_attack["Age"]

# Hombre - Binaria
x2 = heart_attack["Gender"] == 1

# Mujer - Binaria
x3 = heart_attack["Gender"] == 0

# Latidos del corazon - Enteros
x4 = heart_attack["Heart rate"]

# Presion sistolica de la sangre - Enteros
x5 = heart_attack["Systolic blood pressure"]

# Presion diastolica de la sangre - Enteros
x6 = heart_attack["Diastolic blood pressure"]

# Azucar en la sangre - Decimales
x7 = heart_attack["Blood sugar"]

```

```
# Enzima cardiaca en el musculo daniado - Decimales
x8 = heart_attack["CK-MB"]

# Troponina - Decimales
x9 = heart_attack["Troponin"]

# Resultado positivo - Binaria
x10 = heart_attack["Result"] == "positive"

# Resultado negativo - Binaria
x11 = heart_attack["Result"] == "negative"
```

**Código** 5: Construcción de las variables de análisis según su naturaleza

Preguntas:

1. ¿Por qué las variables $x_2, x_3, x_{10}, x_{11}$ usa los símbolos ==?
2. ¿Por qué tenemos que poner entre comillas el nombre exacto del nombre de la columna incluyendo los espacios?
3. ¿Qué pasa si usamos minúsculas o escribimos mal un nombre de columna?
4. ¿Qué pasa si en lugar de *positive* o *negative* escribimos *positivo* o *negativo*?
5. ¿Qué pasaría si en genero hubieramos tenido el valor de 3 que representa a otros? ¿Influiría o no en el análisis y cómo?

### Paso 5 - Análisis estadístico de las variables continuas

Como se analizan de forma diferente las variables continuas a las binarias, construiremos los estadísticos similares solo para las variables que son continuas (enteras y decimales).

```
# Variables continuas:
# x1 - Edad - Enteros
# x4 - Latidos del corazon - Enteros
# x5 - Presion sistolica de la sangre - Enteros
# x6 - Presion diastolica de la sangre - Enteros
# x7 - Azucar en la sangre - Decimales
# x8 - Enzima cardiaca en el musculo daniado - Decimales
# x9 - Troponina - Decimales

estadisticos = []

for xi in [x1, x4, x5, x6, x7, x8, x9]:
  n = xi.count()
  s = xi.sum()
  p = xi.mean()
  q1 = xi.quantile(0.25)
  q2 = xi.quantile(0.50)
```

```
  q3 = xi.quantile(0.75)
  a = xi.min()
  b = xi.max()

  estadisticos.append({
        "Total": n,
        "Suma": s,
        "Media": p,
        "Cuartil 1": q1,
        "Cuartil 2": q2,
        "Cuartil 3": q3,
        "Minimo": a,
        "Maximo": b,
  })

pandas.DataFrame(estadisticos,
                 index=["x1", "x4", "x5", "x6",
                        "x7", "x8", "x9"])
```

**Código** 6: Estadísticos para variables continuas

Adicionalmente podemos transponer los datos para mostrar el mismo reporte, pero visualizado de forma inversa.

```
pandas.DataFrame(estadisticos,
                 index=["x1", "x4", "x5", "x6",
                        "x7", "x8", "x9"]).T
```

**Código** 7: Estadísticos para variables continuas (transpuesto)

Preguntas:

1. ¿Por qué se excluyen las variables $x_2, x_3, x_{10}, x_{11}$?
2. ¿Qué pasa si las incluimos las variables $x_2, x_3, x_{10}, x_{11}$?
3. ¿Qué representa la lista vacía llamada `estadísticos`?
4. En cada iteración para cada variable se construyen sus estadísticos y se forma un diccionario con esos estadísticos que se agrega directamente a la lista de estadísticos. ¿Cómo incluiríamos un nuevo estadístico?
5. La librería de *pandas* construye fácilmente un *DataFrame* a partir de la lista de estadísticos y los reporta en forma de tabla. ¿Qué pasa si no especificamos los índices? ¿Qué representa cada columna en la tabla? ¿Qué representa cada fila en la tabla?
6. ¿Cómo crees que se visualizan mejor los estadísticos, en forma normal o transpuesta?
7. En el reporte transpuesto, ¿Qué representa cada columna en la tabla? ¿Qué representa cada fila en la tabla?

### Paso 6 - Análisis estadístico de las variables binarias

Para analizar las variables binarias, tenemos estadísticos distintos a los del **Paso 5**, ahora tendremos conteos y proporciones (probabilidades de pertenecer a la categoría).

```
# Variables binarias:
# x2 - Hombre - Binaria
# x3 - Mujer - Binaria
# x10 - Resultado positivo - Binaria
# x11 - Resultado negativo - Binaria

estadisticos = []

for xi in [x2, x3, x10, x11]:
  n = xi.count()
  s1 = xi.sum()
  s0 = n - s1
  p1 = s1 / n
  p0 = s0 / n

  estadisticos.append({
        "Total": n,
        "Suma 1s": s1,
        "Suma 0s": s0,
        "Porcentaje 1s": p1,
        "Porcentaje 0s": p0,
  })

pandas.DataFrame(estadisticos,
                 index=["x2", "x3", "x10", "x11"])
```

**Código** 8: Estadísticos para variables binarias

Adicionalmente podemos transponer los datos para mostrar el mismo reporte, pero visualizado de forma inversa.

```
pandas.DataFrame(estadisticos,
                 index=["x2", "x3", "x10", "x11"]).T
```

**Código** 9: Estadísticos para variables binarias (transpuesto)

Preguntas:

1. ¿Qué pasa si analizamos variables no binarias con estos estadísticos?
2. ¿Por qué la suma de 0s es el complemento de $n$ (el total) menos la suma de 1s?
3. ¿Cuál es la probabilidad de ser hombre?
4. ¿Cuál es la probabilidad de ser mujer?

5. ¿Cuál es la probabilidad de tener un resultado positivo?
6. ¿Cuál es la probabilidad de tener un resultado negativo?
7. ¿Cuántos casos con resultado positivo hay en total?
8. Si hubieran 3 categorías, ¿La suma de ceros representaría el complemento a otra categoría o el complemento a las otras categorías?
9. ¿Crees que hay otro estadístico importante a considerar en las variables binarias?

## Conclusiones

En la **Práctica — Estadística Descriptiva** hemos analizado cada eje de datos del conjunto de datos de ataques al corazón, para determinar los ejes de datos numéricos y categóricos y derivar las variables de análisis continuas y binarias. Las variables continuas (enteras o decimales) pueden expresar estadísticos más completos como el promedio o los cuartiles que nos darán una idea del cómo se distribuyen los datos en cada variable, mientras que las variables binarias (de perenencia a una categoría) expresan estadísticos más dirigidos a conteos y probabilidades (o propociones), como el porcentaje de pertenencia a una categoría, el cual se puede expresar como la probabilidad de pertenecer a una categoría y el conteo total de los que si pertenecen y los que no pertenecen.

Con esto hemos desarrollado las habilidades fundamentales para poder procesar conjuntos de datos más complejos relacionados a nuestra área de interés, por ejemplo, analizar el conjunto de clientes que compran productos en nuestra tienda, analizar el conjunto de clientes que tienen una cuenta en nuestro banco, analizar el conjunto de productos de nuestra tienda o analizar los datos sobre el rendimiento de los proyectos de nuestra empresa.

# Práctica — Visualización de Datos

En esta práctica desarrollaremos las habilidades necesarias para analizar un conjunto de datos de forma visual. Buscaremos entender la visualización de las diferentes variables de análisis de forma individual y cruzada según su naturaleza.

## Teoría

Las variables de análisis pueden ser continuas o binarias derivadas de los ejes de datos numéricos y categóricos, como se expone en la teoría de la **Práctica — Estadística Descriptiva**.

Para visualizar de forma individual las variables de análisis utilizaremos:

- **Variable continua** - La gráfica de histograma y densidad, las cuales nos orientan acerca de cómo se distribuyen los datos a lo largo del eje. La gráfica de violín nos mostrará en una forma más compacta y simétrica la misma información, pero nos permitirá identificar los aglutinamientos. Y la gráfica de caja que mostrará la distribución de forma sintetizada en sus cuartiles ($Q_1, Q_2, Q_3$) y los mínimos y máximos.
- **Variable binaria** - Cada variable binaria (o *dummy*) poviene de un eje de datos categórico y una categoría en particular, por lo que cada variable binaria explica si un elemento pertenece o no a una categoría. Para visualizar esto, podemos construir la matriz de conteos y la matriz de probabilidas y luego visualizar los datos en forma de una gráfica de conteos.

Para visualizar de forma cruzada las variables de análisis utilizaremos:

- **Dos variable continuas ($x, y$)** - La gráfica de puntos y regresión nos permitirá visualizar un comportamiento lineal o no lineal entre los datos de dos variables de análisis $x, y$.
- **Múltiples variables continuas** - La gráfica de pares permite analizar al mismo tiempo múltiples variables continuas para encontrar correlaciones entre parejas rápidamente.
- **Una variable binaria ($x$) y una varibla continua** $y$ - La gráfica de caja nos permitirá segmentar los datos de la variable continua $y$ mediante el valor 0 o 1 de la variable binaria $x$, explicando así si hay diferencias significativas entre ambos segmentos (uno para si pertenece a la categoría y otro si no pertenece).
- **Una variable continua ($x$) y una varibla binaria** $y$ - La gráfica de puntos y regresión nos permitirá entender si hay un comportamiento de activación entre los valores continuos $x$ y los valores binarios $y$. Esto es útil en la regresión logística, para determinar si hay un patrón de

activación de la variable de respuesta binaria $y$, dado que la varaible continua (covariable) $x$ supere un valor límite o umbral.

- **Dos variables binarias** - Al poseer dos variables binarias, podemos construir una matriz de conteos y una matriz de probabilidades para cuando la primera variable es 0 y la segunda es 0 y cuando las variables cambian a 1 (ambas 1 o diferentes). Con la gráfica de calor podemos mostrar estos resultados.

- **Dos variable continuas $(x, y)$ y una variable binaria** - La gráfica de puntos nos permitirá visualizar un comportamiento lineal o no lineal entre los datos de dos variables de análisis $x, y$ y colorear los segmentos respecto a la variable binaria.

## Planteamiento del Problema

En esta práctica visualizaremos los datos de los ataques al corazón mediante la construcción de variables continuas (para los ejes de datos numéricos enteros y decimales) y binarias (para los ejes de datos categóricos), en forma individual y cruzada.

## Desarrollo

El desarrollo consiste en adquirir el conjunto de datos desde el repositorio, construir cada una de las variables de análisis, generar las visualizaciones para variables de análisis individuales y cruzadas, según su naturaleza.

### Paso 1 - Aquisición del conjunto de datos

Primero obtendrémos los datos de ataques al corazón directamente desde el repositorio y mostraremos las primeras 5 muestras.

```
import pandas

url = "https://github.com/dragonnomada/ipn-cic-pycien-abril-2025/raw/refs/heads/main/datasets/practicas/heart_attack.csv"

heart_attack = pandas.read_csv(url)

heart_attack.head()
```

**Código** 10: Adquisición del conjunto de datos de ataques al corazón

Preguntas:

1. ¿Cuáles columnas tienen datos numéricos y cuáles categóricos?

2. ¿Cómo se visualizaría cada columna?

**Paso 2 - Construir las variables de análisis**

Siguiendo el análisis de la **Práctica — Estadística Descriptiva** podemos construir las variables de análisis derivadas de los ejes de datos y su naturaleza.

```
# Edad - Enteros
x1 = heart_attack["Age"]

# Hombre - Binaria
x2 = heart_attack["Gender"] == 1

# Mujer - Binaria
x3 = heart_attack["Gender"] == 0

# Latidos del corazon - Enteros
x4 = heart_attack["Heart rate"]

# Presion sistolica de la sangre - Enteros
x5 = heart_attack["Systolic blood pressure"]

# Presion diastolica de la sangre - Enteros
x6 = heart_attack["Diastolic blood pressure"]

# Azucar en la sangre - Decimales
x7 = heart_attack["Blood sugar"]

# Enzima cardiaca en el musculo daniado - Decimales
x8 = heart_attack["CK-MB"]

# Troponina - Decimales
x9 = heart_attack["Troponin"]

# Resultado positivo - Binaria
x10 = heart_attack["Result"] == "positive"

# Resultado negativo - Binaria
x11 = heart_attack["Result"] == "negative"
```

**Código** 11: Construcción de las variables de análisis según su naturaleza

Preguntas:

1. ¿Cuáles variables de análisis conviene entender de forma singular?
2. ¿Cuáles variables de análisis conviene entender de forma cruzada?
3. ¿Cómo analizarías a los hombres (variable $x_2$) respecto a si tienen un resultado positivo (variable $x_{10}$)?

**Paso 3 - Visualizaciones individuales**

Primero analizaremos algunas variables de análisis continuas y binarias de forma individual, lo primero será importar las librerías para visualizar los datos y luego construir las gráficas adecuadas para variables continuas y binarias.

```
import matplotlib.pyplot as pyplot
import seaborn
```

**Código** 12: Importar las librerías de visualización

```
pyplot.hist(x1)
pyplot.show()
```

**Código** 13: Histograma de una variable continua

```
seaborn.kdeplot(x1)
pyplot.show()
```

**Código** 14: Densidad de una variable continua

```
pyplot.hist(x1, density=True)
seaborn.kdeplot(x1)
pyplot.show()
```

**Código** 15: Histograma y densidad de una variable continua

```
seaborn.violinplot(x1)
pyplot.show()
```

**Código** 16: Densidad en forma de violín de una variable continua

```
seaborn.boxplot(x1)
pyplot.show()
```

**Código** 17: Caja estadística (cuariles y rango) de una variable continua

```
x2.value_counts()
```

**Código** 18: Matriz de conteos para una variable binaria

```
x2.value_counts(normalize=True)
```

**Código** 19: Matriz de probabilidades (proporciones) para una variable binaria

```
seaborn.countplot(x=x2)
pyplot.show()
```

**Código** 20: Visualización de los conteos una variable binaria

```
probs = x2.value_counts(normalize=True)
seaborn.barplot(x=probs.index, y=probs)
pyplot.show()
```

**Código** 21: Visualización de las probabilidades una variable binaria

Preguntas:

1. ¿Qué crees que sea mejor para entender la distribución de una variable continua, un histograma o una gráfica de densidad?
2. ¿Qué crees que sea mejor para entender la distribución de una variable continua, un gráfica de violín o una caja estadística?
3. ¿Qué crees que sea mejor para entender la proporción de datos en una variable binaria, la gráfica de los conteos o la gráfica de las probabilidades?

### Paso 4 - Visualizaciones cruzadas

Ahora visualizaremos múltiples variables de forma cruzada según su naturaleza.

```
pyplot.scatter(x1, x5)
pyplot.show()
```

**Código** 22: Correlación entre dos variables continuas

```
pyplot.scatter(x5, x6)
pyplot.show()
```

**Código** 23: Correlación entre otras dos variables continuas

```
seaborn.regplot(x=x5, y=x6, line_kws={"color": "red"})
pyplot.show()
```

**Código** 24: Regresión lineal entre dos variables continuas

```
import numpy

X = numpy.array([x1, x4, x5, x6, x7, x8, x9]).T

seaborn.pairplot(pandas.DataFrame(X))
```

**Código** 25: Correlación en parejas entre múltiples variables continuas

```
seaborn.boxplot(x=x10, y=x1)
pyplot.show()
```

**Código** 26: Cruce entre una variable binaria (eje $x$) y una variable continua (eje $y$)

```
seaborn.regplot(x=x1, y=x10, logistic=True, line_kws={"color
    ": "red"})
pyplot.show()
```

**Código** 27: Regresión logística entre una variable continua (eje $x$) y una variable binaria (eje $y$)

```
conteos = pandas.DataFrame(
    numpy.array([x2, x10]).T,
    columns=["Hombre", "Positivo"]
).groupby(["Hombre", "Positivo"]).size().unstack()

conteos
```

**Código** 28: Matriz de conteos entre dos variables binarias

```
probabilidades = conteos / conteos.values.sum()

probabilidades
```

**Código** 29: Matriz de probabilidades entre dos variables binarias

```
seaborn.heatmap(conteos, annot=True, fmt="d")
pyplot.show()
```

**Código** 30: Gráfica de calor de los conteos entre dos variables binarias

```
seaborn.heatmap(probabilidades, annot=True)
pyplot.show()
```

**Código** 31: Gráfica de calor de las probabilidades entre dos variables binarias

```
seaborn.heatmap(probabilidades * 100, annot=True, fmt=".1f",
    cmap="viridis")
pyplot.show()
```

**Código** 32: Gráfica de calor de las probabilidades entre dos variables binarias (mejorado)

```
seaborn.scatterplot(x=x1, y=x6, hue=x10)
pyplot.show()
```

**Código** 33: Cruce entre dos variables continuas y coloreo por una variable binaria

Preguntas:

1. ¿Cómo es la correlación entre las variables $x_1, x_5$?
2. ¿Cómo es la correlación entre las variables $x_5, x_6$?

3. ¿Que tanto se ajusta la recta de regresión sobre las variables $x_5, x_6$?

4. ¿Cuáles son las variables que más se correlacionan de todas las parejas?

5. ¿Cómo explicas la diferencia entre las cajas generadas de las variables $x_{10}, x_1$? ¿Crees que es significativo el aumento de edad cuando hay ataques al corazón?

6. ¿Cómo explicas la regresión logística de las variables $x_1, x_{10}$? ¿Crees que la probabilidad de sufrir un ataque al corazón aumenta al avanzar la edad?

7. ¿Cuántos hombres sufrieron ataques al corazón y cuántos no?

8. ¿Cuál es la probabilidad de ser hombre y sufrir un ataque al corazón?

9. ¿Cómo se podría calcular la probabilidad de sufrir un ataque al corazón dado que se es hombre?

10. ¿Qué significa que para las variables $x_1, x_6$ (edad contra presión diastólica) esté coloreado más naranja a la derecha y más azúl a la izquierda, pero no pase así arriba y abajo?

## Conclusiones

En la **Práctica — Visualización de Datos** hemos generado diferentes formas de visualizar los datos, según la naturaleza de los ejes de análisis, para las variables continuas hemos buscado explorar el espacio donde residen mediante su distribución y densidad y para las variables binarias hemos intentado explorar las probabilidades de ser verdadero (1) o falso (0) para entender cuántos valores pertenecen o no a la categoría asociada, usando el conjunto de datos sobre ataques de corazón.

Con esto hemos desarrollado las habilidades fundamentales para poder visualizar conjuntos de datos más complejos relacionados a nuestra área de interés, por ejemplo, visualizar cuántos de nuestros clientes son hombres, cuántos realizaron una compra, cómo se relaciona el precio de una venta con el tiempo que pasa el cliente en nuestra tienda o cuántos clientes han cometido fraude en nuestro banco según su edad.

# Práctica — Regresión Lineal

En esta práctica desarrollaremos las habilidades necesarias para analizar la relación lineal entre dos o más variables continuas del conjunto de datos.

## Teoría

La regresión lineal consiste en predecir el valor de una variable de respuesta continua $y$ a partir de las covariables continuas $x_1, x_2, \ldots, x_k$. En el modelo simple, solo se tiene una única covariable $x$ y se genera un modelo $x, y$, pero en el caso general tendremos:

$$y = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \ldots + \beta_k \cdot x_k \tag{1}$$

Donde $\beta_0, \beta_1, \beta_k, \ldots, \beta_k$, son llamados los coeficientes de regresión que tenemos encontrar para que se ajusten mejor a nuestros datos.

En el modelo de regresión lineal simple, se busca que $x$ se encuentre altamente correlacionada a $y$, de no ser así el modelo no tiene sentido, en el modelo general, buscamos que las covariables se correlacionen fuertemente a la $y$ y no guarden realación entre ellas (problema de multicolinealidad).

Si logramos encontrar los valores de $\beta_0, \beta_1, \beta_k, \ldots, \beta_k$ mediante algún algoritmo de optimización, entonces, podremos predecir para cada valor de $x_1, x_2, \ldots, x_k$, cuál debería ser el valor de respuesta de $y$.

## Planteamiento del Problema

En esta práctica ajustaremos un modelo de regresión lineal simple y múltiple para explicar una variable continua a partir de otra u otras variables continuas.

## Desarrollo

El desarrollo consiste en adquirir el conjunto de datos desde el repositorio, construir cada una de las variables de análisis, generar los modelos de regresión y visualizar su comportamiento.

### Paso 1 - Aquisición del conjunto de datos

Primero obtendrémos los datos de ataques al corazón directamente desde el repositorio y mostraremos las primeras 5 muestras.

```
1 import pandas
2
3 url = "https://github.com/dragonnomada/ipn-cic-pycien-abril
      -2025/raw/refs/heads/main/datasets/practicas/heart_attack
      .csv"
4
```

```
heart_attack = pandas.read_csv(url)

heart_attack.head()
```

**Código** 34: Adquisición del conjunto de datos de ataques al corazón

Preguntas:

1. ¿Cuáles son las variables que crees que están más correlacionadas y por qué?
2. ¿Crees que la edad influye en esta correlación?, es decir, ¿Crees que al aumentar la edad esta correlación cambie?

### Paso 2 - Construir las variables de análisis

Siguiendo el análisis de la **Práctica — Estadística Descriptiva** podemos construir las variables de análisis derivadas de los ejes de datos y su naturaleza.

```
# Edad - Enteros
x1 = heart_attack["Age"]

# Hombre - Binaria
x2 = heart_attack["Gender"] == 1

# Mujer - Binaria
x3 = heart_attack["Gender"] == 0

# Latidos del corazon - Enteros
x4 = heart_attack["Heart rate"]

# Presion sistolica de la sangre - Enteros
x5 = heart_attack["Systolic blood pressure"]

# Presion diastolica de la sangre - Enteros
x6 = heart_attack["Diastolic blood pressure"]

# Azucar en la sangre - Decimales
x7 = heart_attack["Blood sugar"]

# Enzima cardiaca en el musculo daniado - Decimales
x8 = heart_attack["CK-MB"]

# Troponina - Decimales
x9 = heart_attack["Troponin"]

# Resultado positivo - Binaria
x10 = heart_attack["Result"] == "positive"

# Resultado negativo - Binaria
x11 = heart_attack["Result"] == "negative"
```

**Código** 35: Construcción de las variables de análisis según su naturaleza

Preguntas:

1. ¿Crees que la presión sistólica y diastólica estén correlacionadas?
2. ¿Crees que la edad influye en que cambie esta correlación?
3. ¿Crees que hay otras variables que se correlacionan?

### Paso 3 - Correlaciones

Primero vamos a obtener las correlaciones entre algunas variables, para ver si hay una influencia positiva, negativa o nula.

```
x1.corr(x5)
```

**Código** 36: Correlación entre dos variables

Esta correlación es casi nula, por lo que inspeccionaremos otras dos variables

```
x5.corr(x6)
```

**Código** 37: Correlación entre otras dos variables

En esta última correlación se muestra una mayor influencia positiva, por lo que vamos a visualizar esta correlación, importando primero las librerías de visualización

```
import matplotlib.pyplot as pyplot
import seaborn
```

**Código** 38: Importar las librerías de visualización de datos

```
seaborn.regplot(x=x5, y=x6, line_kws={"color": "red"})
pyplot.show()
```

**Código** 39: Gráfica de correlación entre dos variables continuas

Preguntas:

1. ¿Cómo se determina que una correlación en nula?
2. ¿Qué significa que la correlación sea positiva y cercana a 1?
3. ¿Qué significa que la correlación sea negativa y cercana a -1?
4. ¿Que la correlación sea de 0.5 es bueno o malo?

**Paso 4 - Regresión lineal simple**

Ahora construiremos un modelo de regresión lineal simple para encontrar los valores de $\beta_0$ y $\beta_1$, dado que $y$ es la presión diastólica ($x_6$) y $x$ es la presión sistólica ($x_5$).

```
from scipy.stats import linregress

reg = linregress(x5, x6)

reg.rvalue ** 2
```

**Código** 40: Modelo de regresión lineal simple para dos variables continuas

El valor de $R^2$ indica qué tan fuerte fue la correlación respecto a los datos, entre más cercana a 1 fue mejor y a 0 peor. Esto se puede ver muchas veces como la capacidad de predicción o su precisión. Ahora que el modelo está ajustado, podemos recuperar los coeficientes de regresión $\beta_0, \beta_1$.

```
b0 = reg.intercept
b1 = reg.slope

b0, b1
```

**Código** 41: Extracción de los coeficientes de regresión

Con estos valores de $\beta_0, \beta_1$ podemos visualizar el espacio de predicción (la recta de predicción).

```
import numpy

xp = numpy.linspace(0, 225)
yp = b0 + b1 * xp

pyplot.scatter(x5, x6)
pyplot.plot(xp, yp, "r--")
pyplot.show()
```

**Código** 42: Espacio de predicción de la regresión lineal simple (recta)

Además, ahora con $\beta_0, \beta_1$ seremos capaces de hacer predicciones sobre cualquier valor de $x$, por ejemplo, para cuando $x = 180$, es decir, obtener la presión diastólica dado que la presión sistólica es de 180.

```
b0 + b1 * 180
```

**Código** 43: Predicción de un valor puntual (regresión lineal simple)

Preguntas:

1. ¿Cuál es el valor de $R^2$ en la regresión y que concluyes de este valor?
2. ¿Cómo interpretas los valores de $\beta_0, \beta_1$?

3. ¿Por qué es importante construir el espacio de predicción manualmente?

4. ¿Cuál sería la presión diastólica para una presión sistólica de 120?

5. ¿Cómo podrías obtener la presión sistólica para una presión diastólica de 90?

**Paso 5 - Regresión lineal múltiple**

Ahora construiremos un modelo de regresión lineal múltiple para encontrar los valores de $\beta_0, \beta_1$ y $\beta_2$, dado que $y$ es la presión diastólica ($x_6$), $x_1$ es la presión sistólica ($x_5$) y $x_2$ es la edad ($x_1$). Ahora buscamos una influencia de la edad sobre la predicción en la presión diastólica.

Primero, visualizaremos el comportamiento de las tres variables involucradas, donde la edad será el color asociado a la correlación anterior.

```
figure = pyplot.scatter(x5, x6, c=x1)
pyplot.colorbar(figure, label="Edad")
pyplot.show()
```

**Código** 44: Visualización de dos variables continuas coloreadas por una variable continua

Observamos que la edad se comporta de tal manera que se colorean en morado los valores más altos de la presión diastólica y en amarillo los más bajos, aunque el comportamiento no es completamente claro, deducimos que al aumentar la edad la presión diastólica decae.

Ahora podemos construir un modelo que agrupe múltiples covariables continuas para predecir la variable de respuesta continua.

```
X = numpy.array([x5, x1]).T
Y = x6

X.shape, Y.shape
```

**Código** 45: Contrucción de la matriz de covariables y el vector de respuesta

Para el entrenamiento, debemos aleatorizar y partir los datos para poder hacer evaluaciones del desempeño real de la regresión.

```
from sklearn.model_selection import train_test_split

X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
    train_size=0.8)

X_train.shape, X_test.shape, Y_train.shape, Y_test.shape
```

**Código** 46: Aleatorización y partición de los datos de aprendizaje

Ahora podemos aplicar un modelo de regresión múltiple con los datos partidos para el entrenamiento, y después evaluar su desempeño con los datos de prueba.

```
from sklearn.linear_model import LinearRegression

reg = LinearRegression()

reg.fit(X_train, Y_train)

reg.score(X_test, Y_test)
```

**Código** 47: Modelo de regresión lineal múltiple

Una vez ajustado el modelo podemos hacer predicciones.

```
reg.predict([
    [150, 10],
    [150, 20],
    [150, 30],
    [150, 40],
    [150, 60],
    [150, 80],
])
```

**Código** 48: Predicción sobre el modelo de regresión lineal múltiple

Y podemos conocer quiénes son $\beta_0, \beta_1, \beta_2$.

```
b0 = reg.intercept_
b1 = reg.coef_[0]
b2 = reg.coef_[1]

b0, b1, b2
```

**Código** 49: Extracción de los coeficientes de regresión (regresión lineal múltiple)

Con los coeficientes de regresión, podemos construir el espacio de predicción para ver que tanto se ajustan los datos

```
figure = pyplot.figure(figsize=(10, 7))
axis = figure.add_subplot(111, projection="3d")

# Regiones para el plano de prediccion en las covariables
x5_range = numpy.linspace(x5.min(), x5.max(), 20)
x1_range = numpy.linspace(x1.min(), x1.max(), 20)

# Mezcla de las regiones
x5_grid, x1_grid = numpy.meshgrid(x5_range, x1_range)

# Calculo del vector de respuesta
y_pred = b0 + b1 * x5_grid + b2 * x1_grid

# Dibujar el plano
```

```
axis.plot_surface(x5_grid, x1_grid, y_pred, alpha=0.5, color
    ="red", label="Plano de prediccion")

# Dibujar los puntos reales
axis.scatter(x5, x1, x6, color="blue", label="Datos reales")

# Etiquetas y titulo
axis.set_xlabel("Presion Sistolica")
axis.set_ylabel("Edad")
axis.set_zlabel("Presion Diastolica")
axis.set_title("Espacio de prediccion")
pyplot.legend()
pyplot.show()
```

**Código** 50: Espacio de predicción (regresión lineal múltiple)

Preguntas:

1. ¿Observas como decae la presión diastólica conforme aumenta la edad?
2. ¿Por qué es importante construir la matriz de covariables?
3. ¿Por qué es importante aleatorizar y partir en datos de entrenamiento y pruebas?
4. ¿Qué significa el puntuaje de evualuación que devuelven las pruebas?
5. ¿Qué notas en la predicción de la presión diastólica a diferentes edades para una misma presión sistólica de 150?
6. ¿Cómo interpretas los coeficientes de regresión $\beta_0, \beta_1, \beta_2$?
7. ¿Crees que el plano de predicción se ajusta a los datos reales?

## Conclusiones

En la **Práctica — Regresión Lineal** hemos generado un modelo de regresión lineal simple y un modelo de regresión lineal múltiple para el conjunto de datos sobre ataques al corazón.

Con esto hemos desarrollado las habilidades fundamentales para poder predecir una variable de respuesta continua que se comporte linealmente sobre covariables continuas. Por ejemplo, para determinar el monto de una venta dado el número de productos que se compran, predecir el sueldo de un empleado dado el número de horas que trabaja o determinar el costo de un departamento dada la superficie de construcción y el número de baños.

# Práctica — Regresión Logística

En esta práctica desarrollaremos las habilidades necesarias para analizar la relación entre una variable de respuesta binaria o probabilística y una o más covariables continuas.

## Teoría

La regresión logística consiste en predecir el valor de una variable de respuesta binaria $y$ a partir de las covariables continuas $x_1, x_2, \ldots, x_k$. En el modelo simple, solo se tiene una única covariable $x$ y se genera un modelo $x, y$, pero en el caso general tendremos:

$$\begin{aligned} y &= \phi(\beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \ldots + \beta_k \cdot x_k) \\ &= \frac{e^{\beta_0+\beta_1 \cdot x_1+\beta_2 \cdot x_2+\ldots+\beta_k \cdot x_k}}{1 + e^{\beta_0+\beta_1 \cdot x_1+\beta_2 \cdot x_2+\ldots+\beta_k \cdot x_k}} \end{aligned} \tag{2}$$

Donde $\beta_0, \beta_1, \beta_k, \ldots, \beta_k$, son llamados los coeficientes de regresión que tenemos encontrar para que se ajusten mejor a nuestros datos.

En el modelo de regresión logística simple, se busca que $x$ tiene un efecto de activación sobre $y$, es decir, que para cierto valor de $x$ es muy probable que $y$ valga 1 (o 0), en el modelo general, buscamos que la combinación de covariables activen a $y$ y no guarden realación entre ellas (problema de multicolinealidad).

Si logramos encontrar los valores de $\beta_0, \beta_1, \beta_k, \ldots, \beta_k$ mediante algún algoritmo de optimización, entonces, podremos predecir para cada valor de $x_1, x_2, \ldots, x_k$, cuál debería ser el valor de respuesta de $y$.

## Planteamiento del Problema

En esta práctica ajustaremos un modelo de regresión logística simple y múltiple para explicar una variable binaria a partir de otra u otras variables continuas.

## Desarrollo

El desarrollo consiste en adquirir el conjunto de datos desde el repositorio, construir cada una de las variables de análisis, generar los modelos de regresión y visualizar su comportamiento.

### Paso 1 - Aquisición del conjunto de datos

Primero obtendrémos los datos de ataques al corazón directamente desde el repositorio y mostraremos las primeras 5 muestras.

```
1 import pandas
2
```

```
url = "https://github.com/dragonnomada/ipn-cic-pycien-abril-2025/raw/refs/heads/main/datasets/practicas/heart_attack.csv"

heart_attack = pandas.read_csv(url)

heart_attack.head()
```

**Código** 51: Adquisición del conjunto de datos de ataques al corazón

Preguntas:

1. ¿Cuáles crees que son las variables que activan una respuesta positiva de ataque al corazón?

2. ¿Crees que el aumento de edad influye en esto?, y ¿Crees que el aumento de azúcar causa esto?

### Paso 2 - Construir las variables de análisis

Siguiendo el análisis de la **Práctica — Estadística Descriptiva** podemos construir las variables de análisis derivadas de los ejes de datos y su naturaleza.

```
# Edad - Enteros
x1 = heart_attack["Age"]

# Hombre - Binaria
x2 = heart_attack["Gender"] == 1

# Mujer - Binaria
x3 = heart_attack["Gender"] == 0

# Latidos del corazon - Enteros
x4 = heart_attack["Heart rate"]

# Presion sistolica de la sangre - Enteros
x5 = heart_attack["Systolic blood pressure"]

# Presion diastolica de la sangre - Enteros
x6 = heart_attack["Diastolic blood pressure"]

# Azucar en la sangre - Decimales
x7 = heart_attack["Blood sugar"]

# Enzima cardiaca en el musculo daniado - Decimales
x8 = heart_attack["CK-MB"]

# Troponina - Decimales
x9 = heart_attack["Troponin"]

# Resultado positivo - Binaria
```

```
x10 = heart_attack["Result"] == "positive"

# Resultado negativo - Binaria
x11 = heart_attack["Result"] == "negative"
```

**Código** 52: Construcción de las variables de análisis según su naturaleza

Preguntas:

1. ¿Crees que la edad ($x_1$) influye en un resultado positivo ($x_{10}$)?
2. ¿Crees que la azúcar en la sangre ($x_8$) influye en un resultado positivo ($x_{10}$)?
3. ¿Crees que la enzima del músculo cardiaco ($x_9$) influye en un resultado positivo ($x_{10}$)?
4. ¿Crees que hay otras variables que activan el ataque cardiaco?

### Paso 3 - Activaciones

Primero vamos a obtener las activaciones entre algunas variables, para ver si hay un comportamiento de activación entre la variable continua y la variable binaria. Entonces, vamos a visualizar esta activación, importando primero las librerías de visualización

```
import matplotlib.pyplot as pyplot
import seaborn
```

**Código** 53: Importar las librerías de visualización de datos

```
seaborn.regplot(x=x1, y=x10, logistic=True, line_kws={"color
    ": "red"})
pyplot.show()
```

**Código** 54: Gráfica de activación entre una variable continua y una variable binaria

Preguntas:

1. ¿Cómo interpretas la activación entre la edad ($x_1$) y el resultado positivo ($x_{10}$)?
2. ¿A partir de qué edad hay más del 70% de probabilidad de sufrir un atque cardiaco?
3. ¿Cuál es la probabilidad de que alguien de 40 años sufra un ataque al corazón?

**Paso 4 - Regresión logística simple**

Ahora construiremos un modelo de regresión logística simple para encontrar los valores de $\beta_0$ y $\beta_1$, dado que $y$ es la respuesta positiva ($x_{10}$) y $x$ es la edad ($x_1$). Debemos agrupar la única variable como si fuera el modelo de regresión múltiple.

```
import numpy

X = numpy.array([x1]).T
Y = x10

X.shape, Y.shape
```

**Código** 55: Modelo de regresión logítica simple para una variable continua y una variable binaria

Para el entrenamiento, debemos aleatorizar y partir los datos para poder hacer evaluaciones del desempeño real de la regresión.

```
from sklearn.model_selection import train_test_split

X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
    train_size=0.8)

X_train.shape, X_test.shape, Y_train.shape, Y_test.shape
```

**Código** 56: Aleatorización y partición de los datos de aprendizaje

Ahora podemos aplicar un modelo de regresión logística simple con los datos partidos para el entrenamiento, y después evaluar su desempeño con los datos de prueba. La puntuación final indica la precisión alcanzada en la predicción.

```
from sklearn.linear_model import LogisticRegression

reg = LogisticRegression()

reg.fit(X_train, Y_train)

reg.score(X_test, Y_test)
```

**Código** 57: Ajuste del modelo de regresión logística simple

Ahora que el modelo está ajustado, podemos obtener los coeficientes de regresión $\beta_0, \beta_1$.

```
b0 = reg.intercept_
b1 = reg.coef_[0]

b0, b1
```

**Código** 58: Extracción de los coeficientes de regresión

Con estos valores de $\beta_0, \beta_1$ podemos visualizar el espacio de predicción (la curva de predicción de probabilidad).

```
from numpy import exp

xp = numpy.linspace(x1.min(), x1.max())
yp = numpy.exp(b0 + b1 * xp) / (1 + exp(b0 + b1 * xp))

pyplot.scatter(x1, x10)
pyplot.plot(xp, yp, "r--")
pyplot.show()
```

**Código** 59: Espacio de predicción de la regresión logística simple (curva de activación)

Además, ahora con $\beta_0, \beta_1$ seremos capaces de hacer predicciones sobre cualquier valor de $x$, por ejemplo, para la edad es $x = 60$, es decir, obtener la probabilidad de tener un ataque al corazón dado que la edad es de 60.

```
b0 + b1 * 60
```

**Código** 60: Predicción de un valor puntual (regresión logística simple)

Preguntas:

1. ¿Cuál es la precisión alcanzada por la regresión logística?
2. ¿Cómo interpretas los valores de $\beta_0, \beta_1$?
3. ¿Por qué es importante construir el espacio de predicción manualmente?
4. ¿Cuál sería la probabilidad de sufrir una atque al corazón para una edad de 65?
5. ¿Cómo cambia la probabilidad entre una edad de 65 y 70 años?

### Paso 5 - Regresión logística múltiple

Ahora construiremos un modelo de regresión logística múltiple para encontrar los valores de $\beta_0, \beta_1$ y $\beta_2$, dado que $y$ es la probabilidad de sufrir un ataque al corazón ($x_{10}$), $x_1$ es la edad ($x_1$), $x_2$ es la azúcar en la sangre ($x_8$) y $x_3$ es la enzima del músculo dañado ($x_9$). Ahora buscamos una influencia de la edad, la azúcar en la sangre y la enzima *CK-MB* sobre la predicción en el ataque al corazón.

Primero, visualizaremos el comportamiento de las tres variables involucradas.

```
seaborn.regplot(x=x1, y=x10, logistic=True, line_kws={"color
    ": "red"})
pyplot.show()
```

**Código** 61: Regresión logística de la edad al ataque de corazón

```
seaborn.regplot(x=x8, y=x10, logistic=True, line_kws={"color
    ": "red"})
pyplot.show()
```

**Código** 62: Regresión logística de la azúcar en la sangre al ataque de corazón

```
seaborn.regplot(x=x9, y=x10, logistic=True, line_kws={"color
    ": "red"})
pyplot.show()
```

**Código** 63: Regresión logística de la enzima *CK-MB* al ataque de corazón

Observamos que la edad se influye lentamente al ataque al corazón, mientras que subir la azúcar en la sangre el valor de la enzima aceleran la predicción a 1 de sufrir un ataque al corazón.

Ahora podemos construir un modelo que agrupe múltiples covariables continuas para predecir la variable de respuesta binaria.

```
X = numpy.array([x1, x8, x9]).T
Y = x10

X.shape, Y.shape
```

**Código** 64: Contrucción de la matriz de covariables y el vector de respuesta

Para el entrenamiento, debemos aleatorizar y partir los datos para poder hacer evaluaciones del desempeño real de la regresión.

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
    train_size=0.8)

X_train.shape, X_test.shape, Y_train.shape, Y_test.shape
```

**Código** 65: Aleatorización y partición de los datos de aprendizaje

Ahora podemos aplicar un modelo de regresión múltiple con los datos partidos para el entrenamiento, y después evaluar su desempeño con los datos de prueba.

```
reg = LogisticRegression()

reg.fit(X_train, Y_train)

reg.score(X_test, Y_test)
```

**Código** 66: Modelo de regresión logística múltiple

Y podemos conocer quiénes son $\beta_0, \beta_1, \beta_2, \beta_3$.

```
b0 = reg.intercept_
b1 = reg.coef_[0][0]
b2 = reg.coef_[0][1]
```

```
b3 = reg.coef_[0][2]

b0, b1, b2, b3
```

**Código** 67: Extracción de los coeficientes de regresión (regresión logística múltiple)

Una vez ajustado el modelo podemos hacer predicciones. Antes, debemos inspeccionar el espacio donde viven las covariables usadas (edad, azúcar en la sangre y enzima *CK-MB*).

```
pyplot.boxplot(x1)
pyplot.show()
```

**Código** 68: Espacio de los valores de la edad

```
pyplot.boxplot(x8)
pyplot.show()
```

**Código** 69: Espacio de los valores de la azúcar en la sangre

```
pyplot.boxplot(x9)
pyplot.show()
```

**Código** 70: Espacio de los valores de la enzima *CK-MB*

Ahora podemos calular algunas probabilidades.

```
xb1 = b0 + b1 * 60 + b2 * 2 + b3 * 0.006
yp1 = exp(xb1) / (1 + exp(xb1))

yp1
```

**Código** 71: Predicción de ataque al corazón a los 60 años con nivel 2 de azúcar

```
xb2 = b0 + b1 * 60 + b2 * 3 + b3 * 0.006
yp2 = exp(xb2) / (1 + exp(xb2))

yp2
```

**Código** 72: Predicción de ataque al corazón a los 60 años con nivel 3 de azúcar

```
xb3 = b0 + b1 * 60 + b2 * 10 + b3 * 0.006
yp3 = exp(xb3) / (1 + exp(xb3))

yp3
```

**Código** 73: Predicción de ataque al corazón a los 60 años con nivel 10 de azúcar

Preguntas:

1. ¿A partir de qué nivel de azúcar en la sangre es casi 100% probable que se dé un ataque al corazón?

2. ¿Cuál de las 3 covariables influye más?

3. ¿Por qué se filtrar los datos en las cajas estadísticas?

4. ¿Cómo cambia la probabilidad de tener un ataque al corazón en una persona de 60 años conforme aumenta su nivel de azúcar en la sangre?

## Conclusiones

En la **Práctica — Regresión Logística** hemos generado los modelos de regresión logística simple y regresión logística múltiple para el conjunto de datos sobre ataques al corazón.

Con esto hemos desarrollado las habilidades fundamentales para poder predecir una variable de respuesta binaria que se comporte en modo de activación sobre covariables continuas. Por ejemplo, para determinar si un cliente cometerá fraude, si un cliente comprará un producto o si un empleado renunciará.

# Práctica — Redes Neuronales

En esta práctica desarrollaremos las habilidades necesarias para analizar un conjunto de datos usando redes neuronales.

## Teoría

Las redes neuronales se componen de neuronas artificiales que tienen asociada una función de activación y pesos para regular las entradas para producir las salidas.

Según el tipo de salida (variable de respuesta) requerida, se puede usar una u otra función de activación, por ejemplo:

- **Identidad** - No modifica las entradas, produce una salida entre $(-\infty, \infty)$, ideal para variables de respuesta continuas, positivas y negativas.
- **ReLU** - Regula las entradas negativas, produce una salida entre $(0, \infty)$, ideal para variables de respuesta continuas y positivas.
- **Sigmoid** - Regula las entradas, produce una salida entre $(0, 1)$, ideal para una única variable de respuesta binaria.
- **Softmax** - Regula las entradas, produce una salida entre $(0, 1)^d$, ideal para múltiples variables de respuesta binarias ($d$-variables de respuesta).
- **Tanh** - Regula las entradas, produce una salida entre $(-1, 1)$, ideal para variables de respuesta de sentimiento positivo $(+1)$, negativo $(-1)$ y neutro $(0)$.

Para las capas ocultas se recomienda usar la función de activación $ReLU$.

Para optimizar los pesos hay diferentes optimizadores como *SGD* (*Stochastic Gradient Descend*) y *Adam* (*Adaptative Gradient with Momentum*) entre los más populares.

Para medir la pérdida, se usará la función de pérdida adecuada para la capa de salida, es decir, la variable de respuesta:

- **Identidad** - usar $MSE$.
- **ReLU** - usar $MSE$.
- **Sigmoid** - usar $BinaryCrossEntropy$.
- **Softmax** - usar $CategoricalCrossEntropy$.
- **Tanh** - $MSE$.

La red neuronal se puede construir como una secuencia de capas desde una capa de entrada, opcionalmente las capas ocultas intermedias (capas densas) y una capa de salida (capa de densa). Las entradas y salidas deben coincidir en el número de nodos y forma. Se recomienda tener más nodos que entradas en las capas ocultas.

Al realizar el entrenamiento se debe considerar un tamaño de lote (*Batch Size*) y el número de iteraciones o épocas de entrenamiento (*Epochs*).

## Planteamiento del Problema

En esta práctica ajustaremos una red neuronal para explicar una variable binaria a partir de otra u otras variables continuas.

## Desarrollo

El desarrollo consiste en adquirir el conjunto de datos desde el repositorio, construir cada una de las variables de análisis, generar una red neuronal y finalmente visualizar su comportamiento.

### Paso 1 - Aquisición del conjunto de datos

Primero obtendrémos los datos de ataques al corazón directamente desde el repositorio y mostraremos las primeras 5 muestras.

```
import pandas

url = "https://github.com/dragonnomada/ipn-cic-pycien-abril-2025/raw/refs/heads/main/datasets/practicas/heart_attack.csv"

heart_attack = pandas.read_csv(url)

heart_attack.head()
```

**Código** 74: Adquisición del conjunto de datos de ataques al corazón

Preguntas:

1. ¿Cuáles crees que debería ser la variable de respuesta?
2. ¿Crees que es mejor predecir un valor continuo o uno binario?
3. ¿Crees que la red neuronal sea mejor en problemas de clasificación o en problemas de regresión?

### Paso 2 - Construir las variables de análisis

Siguiendo el análisis de la **Práctica — Estadística Descriptiva** podemos construir las variables de análisis derivadas de los ejes de datos y su naturaleza.

```
# Edad - Enteros
x1 = heart_attack["Age"]

# Hombre - Binaria
x2 = heart_attack["Gender"] == 1

# Mujer - Binaria
x3 = heart_attack["Gender"] == 0

# Latidos del corazon - Enteros
x4 = heart_attack["Heart rate"]

# Presion sistolica de la sangre - Enteros
x5 = heart_attack["Systolic blood pressure"]

# Presion diastolica de la sangre - Enteros
x6 = heart_attack["Diastolic blood pressure"]

# Azucar en la sangre - Decimales
x7 = heart_attack["Blood sugar"]

# Enzima cardiaca en el musculo daniado - Decimales
x8 = heart_attack["CK-MB"]

# Troponina - Decimales
x9 = heart_attack["Troponin"]

# Resultado positivo - Binaria
x10 = heart_attack["Result"] == "positive"

# Resultado negativo - Binaria
x11 = heart_attack["Result"] == "negative"
```

**Código** 75: Construcción de las variables de análisis según su naturaleza

Preguntas:

1. ¿Crees que todas las variables pueden explicar si habrá un ataque al corazón?
2. ¿Cuál crees que es un buen porcentaje de aprendizaje en una red neuronal?

### Paso 3 - Modelo de la Red Neuronal

Ahora construiremos un modelo de red neuronal para explicar $y$ que es la respuesta positiva a un ataque al corazón ($x_{10}$) a partir de todas las demás variables.

```
import numpy

X = numpy.array([x1, x2, x3, x4, x5, x6, x7, x8, x9]).T
Y = x10

X.shape, Y.shape
```

**Código** 76: Matriz de covariables y vector de respuesta

Para el entrenamiento, debemos aleatorizar y partir los datos para poder hacer evaluaciones del desempeño real de la regresión.

```
from sklearn.model_selection import train_test_split

X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
    train_size=0.8)

X_train.shape, X_test.shape, Y_train.shape, Y_test.shape
```

**Código** 77: Aleatorización y partición de los datos de aprendizaje

Ahora podemos generar el modelo de la red neuronal como la secuencia de la capa de entrada, las capas ocultas y la capa de salida.

```
from keras import Sequential
from keras.layers import Input, Dense
from keras.optimizers import Adam
from keras.losses import BinaryCrossentropy

model = Sequential([
    Input(shape=(9,)),
    Dense(100, activation="relu"),
    Dense(1, activation="sigmoid"),
])

model.compile(
    optimizer=Adam(),
    loss=BinaryCrossentropy(),
    metrics=["accuracy"]
)

model.summary()
```

**Código** 78: Ajuste del modelo de una red neuronal

Ahora entrenamos el modelo durante 20 épocas usando lotes de tamaño 20.

```
model.fit(X_train, Y_train, batch_size=20, epochs=20)
```

**Código** 79: Fase de entrenamiento de la red neuronal

Ahora podemos evaluar sobre los datos de prueba la precisión real de la red neuronal.

```
model.evaluate(X_test, Y_test)
```

**Código** 80: Evaluación de la presición de la red neuronal sobre los datos de prueba

Preguntas:

1. ¿Por qué la capa de salida tiene un único nodo de tipo sigmoid?
2. ¿Por qué la capa oculta tiene 100 nodos?
3. ¿Qué pasa si aumentamos la capa oculta a 200 nodos?
4. ¿Qué pasa si agregamos otra capa oculta de 20 nodos?
5. ¿Por qué se usa la función de pérdida *BinaryCrossEntropy*?
6. ¿Cómo varia la precisión alcanzada durante las épocas de entrenamiento?
7. ¿Cuál es la precisión real?

### Paso 4 - Modelo de la Red Neuronal simplificada

Ahora construiremos otra modelo de red neuronal para explicar $y$ que es la respuesta positiva a un ataque al corazón ($x_{10}$) a partir de menos variables.

```
import numpy

X = numpy.array([x1, x8, x9]).T
Y = x10

X.shape, Y.shape
```

**Código** 81: Matriz de covariables y vector de respuesta

Para el entrenamiento, debemos aleatorizar y partir los datos para poder hacer evaluaciones del desempeño real de la regresión.

```
from sklearn.model_selection import train_test_split

X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
    train_size=0.8)

X_train.shape, X_test.shape, Y_train.shape, Y_test.shape
```

**Código** 82: Aleatorización y partición de los datos de aprendizaje

Ahora podemos generar el modelo de la red neuronal como la secuencia de la capa de entrada, las capas ocultas y la capa de salida.

```
from keras import Sequential
from keras.layers import Input, Dense
from keras.optimizers import Adam
from keras.losses import BinaryCrossentropy

model = Sequential([
    Input(shape=(9,)),
    Dense(100, activation="relu"),
    Dense(1, activation="sigmoid"),
])

model.compile(
    optimizer=Adam(),
    loss=BinaryCrossentropy(),
    metrics=["accuracy"]
)

model.summary()
```

**Código** 83: Ajuste del modelo de una red neuronal

Ahora entrenamos el modelo durante 20 épocas usando lotes de tamaño 20.

```
model.fit(X_train, Y_train, batch_size=20, epochs=20)
```

**Código** 84: Fase de entrenamiento de la red neuronal

Ahora podemos evaluar sobre los datos de prueba la precisión real de la red neuronal.

```
model.evaluate(X_test, Y_test)
```

**Código** 85: Evaluación de la presición de la red neuronal sobre los datos de prueba

Ahora que tenemos menos variables podemos graficar el espacio de predicción. Solo debemos mantener constantes algunos valores de los otros ejes, mientras variamos el eje de interés

```
import matplotlib.pyplot as pyplot
import seaborn
```

**Código** 86: Importar las librerías para la visualización de datos

```
x8p = numpy.linspace(x8.min(), x8.max())

yp = model.predict(numpy.array([60 * numpy.ones(x8p.shape),
    x8p, 0.006 * numpy.ones(x8p.shape)]).T)

pyplot.scatter(x8, x10)
pyplot.plot(x8p, yp, "r--")
```

```
pyplot.show()
```

**Código** 87: Gráfica de variación de la azúcar en la sangre

```
x1p = numpy.linspace(x1.min(), x1.max())

yp = model.predict(numpy.array([x1p, 2 * numpy.ones(x1p.
    shape), 0.006 * numpy.ones(x1p.shape)]).T)

pyplot.scatter(x1, x10)
pyplot.plot(x1p, yp, "r--")
pyplot.show()
```

**Código** 88: Gráfica de variación de la edad

```
x9p = numpy.linspace(x9.min(), x9.max())

yp = model.predict(numpy.array([60 * numpy.ones(x9p.shape),
    2 * numpy.ones(x9p.shape), x9p]).T)

pyplot.scatter(x9, x10)
pyplot.plot(x9p, yp, "r--")
pyplot.show()
```

**Código** 89: Gráfica de variación de la enzima *CK-MB*

En un modelo simplificado es más fácil hacer predicciones:

```
model.predict(numpy.array([
    [60, 2, 0.006],
    [60, 3, 0.006],
    [60, 10, 0.006],
]))
```

**Código** 90: Predicción de la red neuronal

Preguntas:

1. ¿Cuáles variables se usaron en el modelo simplificado?
2. ¿Qué precisión se alcanzó ahora?
3. ¿Mejoró o emperó la precisión real?
4. ¿Qué gráfica no parece muy acorde?
5. ¿A partir de qué nivel de azúcar en la sangre es casi probable tener un ataque al corazón?
6. ¿Cuál es la predicción más alta de un ataque al corazón para una persona de 60 años?

## Conclusiones

En la **Práctica — Redes Neuronales** hemos generado los modelos de red neuronal para el conjunto de datos sobre ataques al corazón.

Con esto hemos desarrollado las habilidades fundamentales para poder predecir una variable de respuesta binaria que se comporte en modo de activación sobre covariables continuas. Por ejemplo, para determinar si un producto será vendido hoy, para determinar si una persona desarrollará cáncer o para determinar si un empleado cometará fraude.

## Proyecto Final (Opcional)

Finalmente, puedes generar un proyecto final para tu portafolio profesional, similar a la práctica que hayas realizado, es decir, con la misma estructura de obtener un resultado paso a paso usando las técnicas aprendidas. Busca un conjunto de datos que se pueda analizar como el de la práctica.

Los puntos para completar el proyecto final son los siguientes:

1. Busca en internet una página que provea conjuntos de datos (*datasets*), por ejemplo, UCI Machine Learning Datasets o Kagggle.

2. Dentro de esa página busca un conjunto de datos que llame tu atención y lee cuidadósamente su documentación, por ejemplo, cuántas variables tiene, cuántos registros hay, etc.

3. Descarga el conjunto de datos en formato `.csv` o si no puedes busca otro. Luego, sube el conjunto de datos a tu libreta de Colab o Jupyter.

4. Carga los datos en la Libreta de Python, inspecciona los datos que contine y plantea un problema similar e incluso más corto que el de la práctica que resolviste, por ejemplo, si en la práctica se analizan 3 variables, tu solo analiza una o dos.

5. Desarrolla paso a paso la solución de tu problema planteado similar a la práctica.

6. Escribe tus conclusiones y documenta la libreta como se solicita en las instrucciones.

7. Envía tu práctica y proyecto final en formato PDF, descarga las libretas o crea un archivo de word con capturas de pantalla de tu código y resultados. Es preferible enviar la libreta convertida a PDF.

8. Si tienes alguna duda o no encuentras un repositorio, puedes utilizar alguno de los datasets vistos durante el curso, alguno del que dispongas o alguno del INEGI.

Intenta plantear algunas preguntas interesantes y responderlas con los resultados obtenidos. Por ejemplo, ¿Cuál es la probabilidad de ser hombre? o ¿Cuál es la probabilidad de ser vender un producto?, dependiendo de los datos que estés utilizando.

Mucha suerte y energía en tu práctica y proyecto final :)