

Práctica 2 (25% nota final)

1. **Descripción del dataset.** ¿Por qué es importante y qué pregunta/problema pretende responder?

El conjunto de datos "Heart Attack Analysis & Prediction dataset" es una colección de registros médicos que se recopilaron para analizar y predecir la probabilidad de que un paciente sufra un ataque al corazón basado en una serie de parámetros y medidas de salud. Contiene información sobre factores de riesgo cardiovasculares, incluyendo la edad, el sexo, el tipo de dolor en el pecho, la presión arterial en reposo, el colesterol, los niveles de glucosa, los resultados de electrocardiogramas en reposo, la frecuencia cardíaca máxima alcanzada, el ejercicio inducido a la angina, el ST depresión inducida por el ejercicio, el número de vasos principales coloreados por la fluoroscopia y el talio en la prueba de esfuerzo.

Este dataset es importante porque la enfermedad del corazón es la principal causa de muerte en todo el mundo. Con una mejor comprensión de cómo estos factores están relacionados con la probabilidad de un ataque al corazón, los médicos y los científicos pueden desarrollar mejores estrategias para la prevención y el tratamiento de las enfermedades del corazón.

El problema que intenta responder es: ¿Cómo se pueden predecir los ataques al corazón basados en factores de riesgo y características de salud? En otras palabras, ¿qué combinaciones de factores son más indicativas de un riesgo elevado de ataque al corazón? Y, por tanto ¿qué pacientes deben ser monitoreados más de cerca o deben recibir tratamiento preventivo?

2. **Integración y selección** de los datos de interés a analizar. Puede ser el resultado de adicionar diferentes datasets o una subselección útil de los datos originales, en base al objetivo que se quiera conseguir.

En esta etapa de la integración y selección de los datos, necesitamos identificar y seleccionar los datos que serán útiles para nuestro análisis. Dado que estamos trabajando con un solo dataset, no necesitamos agregar datos adicionales.

Para determinar qué datos serán útiles para nuestro análisis, necesitamos considerar qué variables pueden estar más fuertemente asociadas con el riesgo de un ataque al corazón. Por ejemplo, es posible que estemos particularmente

interesados en variables como la edad, el sexo, el colesterol, el tipo de dolor en el pecho y la frecuencia cardíaca máxima alcanzada, ya que estudios previos han demostrado que estos factores pueden estar asociados con un mayor riesgo de enfermedad cardíaca.

Sin embargo, también es importante tener en cuenta que diferentes variables pueden interactuar entre sí de formas complejas para influir en el riesgo de un ataque al corazón. Por lo tanto, en lugar de simplemente seleccionar un subconjunto de variables de interés, podemos optar por incluir todas las variables disponibles en nuestro análisis inicial para tener una imagen completa de cómo los diferentes factores de riesgo pueden estar relacionados con el riesgo de un ataque al corazón.

Por lo tanto, en este caso, seleccionaremos todas las variables disponibles en el dataset para nuestro análisis. Una vez que hayamos hecho esto, podemos realizar un análisis exploratorio de los datos para identificar cualquier patrón o tendencia interesante que pueda surgir.

3. Limpieza de los datos.

A continuación, cargaremos el dataset para poder hacer el análisis exploratorio de los datos y poder responder a las preguntas propuestas en el enunciado.

```
# Importar la biblioteca necesaria
import pandas as pd

# Carga el conjunto de datos
data = pd.read_csv('/kaggle/input/heart-attack-analysis-prediction-dataset/heart.csv')

# Muestra las primeras filas del conjunto de datos
data.head()
```

| | age | sex | cp | trtbps | chol | fbs | restecg | thalachh | exng | oldpeak | slp | caa | thall | output |
|---|-----|-----|----|--------|------|-----|---------|----------|------|---------|-----|-----|-------|--------|
| 0 | 63 | 1 | 3 | 145 | 233 | 1 | 0 | 150 | 0 | 2.3 | 0 | 0 | 1 | 1 |
| 1 | 37 | 1 | 2 | 130 | 250 | 0 | 1 | 187 | 0 | 3.5 | 0 | 0 | 2 | 1 |
| 2 | 41 | 0 | 1 | 130 | 204 | 0 | 0 | 172 | 0 | 1.4 | 2 | 0 | 2 | 1 |
| 3 | 56 | 1 | 1 | 120 | 236 | 0 | 1 | 178 | 0 | 0.8 | 2 | 0 | 2 | 1 |
| 4 | 57 | 0 | 0 | 120 | 354 | 0 | 1 | 163 | 1 | 0.6 | 2 | 0 | 2 | 1 |

3.1. ¿Los datos contienen ceros o elementos vacíos? Gestiona cada uno de estos casos.

El dataset "Heart Attack Analysis & Prediction" no contiene valores nulos. Esto significa que no necesitas preocuparte por la imputación o la eliminación de valores faltantes, lo cual es un buen indicio de la calidad del dataset.

```
# Verificar si hay elementos vacíos en el conjunto de datos
data.isnull().sum()
```

```
age      0
sex      0
cp       0
trtbps   0
chol     0
fbs      0
restecg  0
thalachh 0
exng     0
oldpeak  0
slp      0
caa      0
thall    0
output   0
dtype: int64
```

Sin embargo, encontramos varios ceros en varias columnas.

```
# Verificar si hay ceros en el conjunto de datos
(data == 0).sum()
```

```
age      0
sex      96
cp       143
trtbps    0
chol     0
fbs      258
restecg   147
thalachh  0
exng     204
oldpeak   99
slp       21
caa      175
thall     2
output   138
dtype: int64
```

Algunos de estos ceros son significativos y representan categorías o condiciones. Veamos cada uno de los casos detectados:

- *sex' es una variable binaria que representan la categoría a la que pertenece el sujeto (0 = mujer, 1 = hombre).*
- *'cp' corresponde al tipo de dolor de pecho y representa una categoría específica de dolor (0 = 'typical angina', 1 = 'atypical angina', 2 = 'non-anginal pain', 3 = 'asymptomatic')*

- *'fbs'* corresponde al nivel de azúcar en sangre (en ayunas) y nuevamente es una variable binaria que indica si el nivel de azúcar se encuentra por encima o por debajo de 120 mg/dl (0 = false, 1 = true).
- *'rest_ecg'* corresponde a los resultados electrocardiográficos en reposo y representa si se observan anomalías en los gráficos o no (0 = normal, 1 = anomalías onda ST-T, 2 = indicios de hipertrofia ventricular).
- *'exng'* corresponde a angina inducida por el ejercicio y representa la existencia de angina tras el proceso (0 = No, 1 = Si).
- *'old_peak'* indica el nivel de cambios en el ST por el ejercicio en relación con el reposo y parece un valor que oscila entre 0 y 6.2 (en este caso).
- *'slp'* representa la categoría específica de la pendiente del segmento ST en el ejercicio (0 = sin pendiente, 1 = plana, 2 = descendiente).
- *'caa'* corresponde al número de vasos principales coloreados por la fluoroscopia y representa si se coloreo o no algún vaso principal oscilando los valores de esta variable entre 0 y 3.
- *'output'* es el target o diagnostico de enfermedad cardíaca y nuevamente es una variable binaria que indica si con un estrechamiento del 50% del diámetro, existe mayor o menor probabilidad de enfermedad del corazón (0 = Menor, 1 = Mayor).

Respecto a la variable *'thall'* podría considerarse que existen datos faltantes puesto que en la descripción del dataset se indica que debe tener un rango entre 1~3 aunque también definen la existencia del valor 0 como valor nulo (null). Para este caso, podríamos considerar imputar estos dos valores ausentes con la moda de la columna ya que son datos categóricos.

3.2. Identifica y gestiona los valores extremos.

Los valores extremos o atípicos son datos que se diferencian significativamente del resto de las observaciones. Pueden ser causados por variabilidad en los datos o errores de medición. En cualquier caso, los valores extremos pueden alterar significativamente los resultados de un análisis de datos y deben ser correctamente gestionados.

A continuación, se muestra el método del rango intercuartil (IQR) utilizado para detectar los valores atípicos.

```
# Calcular el IQR para cada columna
Q1 = data.quantile(0.25)
Q3 = data.quantile(0.75)
IQR = Q3 - Q1

# Definir los límites superior e inferior para los valores extremos
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Identificar los valores extremos
outliers = data[(data < lower_bound) | (data > upper_bound)].count()
print(outliers)
```

```
age      0
sex      0
cp       0
trtbps   9
chol     5
fbs     45
restecg  0
thalachh 1
exng     0
oldpeak  5
slp      0
caa     25
thall    2
output   0
dtype: int64
```

Podemos ver que varias columnas que presentan valores atípicos a excepción de la variable 'fbs' ya que al ser valores binarios no son realmente outliers en el sentido convencional. Respecto al resto de variables, la mejor opción sería reemplazar los valores atípicos por la mediana del campo tal y como se muestra a continuación.

```
# Importar la biblioteca necesaria
import numpy as np

# Calcular la mediana para cada columna
medians = data.median()

# Reemplazar los outliers por la mediana en cada columna
for column in ['trtbps', 'chol', 'thalachh', 'oldpeak', 'caa', 'thall']:
    data[column] = np.where((data[column] < lower_bound[column]) |
                           (data[column] > upper_bound[column]),
                           medians[column],
                           data[column])
```

4. Análisis de los datos.

- 4.1. Selección de los grupos de datos que se quieren analizar/comparar (p. ej., si se van a comparar grupos de datos, ¿cuáles son estos grupos y qué tipo de análisis se van a aplicar?).

Para determinar los grupos de datos que queremos analizar o comparar, debemos referirnos al objetivo principal de nuestro análisis, que en este caso es la predicción de ataques cardíacos.

La variable objetivo en este análisis es 'output', que indica si un paciente tiene más o menos probabilidades de tener un ataque cardíaco por lo que, algunas de las variables o grupos de datos que podríamos comparar podrían ser:

1. *Sexo y ataque al corazón ('sex' y 'output'):* Podríamos estar interesados en comparar la incidencia de ataques cardíacos entre hombres y mujeres para determinar si el sexo es un factor de riesgo significativo.
2. *Edad y ataque al corazón ('age' y 'output'):* Otro análisis podría ser examinar la relación entre la edad de los pacientes y la incidencia de ataques cardíacos.
3. *Dolor en el pecho y ataque al corazón ('cp' y 'output'):* También podríamos analizar la relación entre los diferentes tipos de dolor en el pecho y la incidencia de ataques cardíacos.
4. *Presión arterial en reposo y ataque al corazón ('trtbps' y 'output'):* Podríamos comparar la presión arterial en reposo en pacientes que tuvieron ataques cardíacos frente a aquellos que no los tuvieron.

- 4.2. Comprobación de la normalidad y homogeneidad de la varianza.

La normalidad es una propiedad importante de los datos que asumen muchos métodos estadísticos. Para comprobar la normalidad de los datos, podemos utilizar pruebas estadísticas, como la prueba de Shapiro-Wilk, o técnicas de visualización, como los histogramas y los gráficos Q-Q.

La prueba de Shapiro-Wilk es una prueba estadística que podemos utilizar para evaluar si una muestra de datos se ajusta a una distribución normal.

```
from scipy.stats import shapiro

# Realizar la prueba de Shapiro-Wilk en cada columna
for column in data.columns:
    _, p_value = shapiro(data[column])
    if p_value < 0.05:
        print(f"{column} no sigue una distribución normal")
    else:
        print(f"{column} sigue una distribución normal")
```

```
age no sigue una distribución normal
sex no sigue una distribución normal
cp no sigue una distribución normal
trtbps no sigue una distribución normal
chol sigue una distribución normal
fbs no sigue una distribución normal
restecg no sigue una distribución normal
thalachh no sigue una distribución normal
exng no sigue una distribución normal
oldpeak no sigue una distribución normal
slp no sigue una distribución normal
caa no sigue una distribución normal
thall no sigue una distribución normal
output no sigue una distribución normal
```

Según la prueba de Shapiro-Wilk, parece que la mayoría de las variables no siguen una distribución normal lo cual es bastante común en los datos del mundo real y hay muchas razones por las que podría suceder como por ejemplo que los datos estén sesgados o que la variable no esté en realidad distribuida normalmente en la población.

Además de las pruebas estadísticas, los gráficos de histogramas y los gráficos Q-Q también son útiles para verificar la normalidad visualmente.

```
variables_to_check = ['age', 'sex', 'cp', 'trtbps', 'chol', 'fbs', 'restecg',
                      'thalachh', 'exng', 'oldpeak', 'slp', 'caa', 'thall']

for var in variables_to_check:
    # Histograma
    plt.figure(figsize=(10, 4))
    plt.subplot(1, 2, 1)
    plt.hist(data[var], bins='auto', alpha=0.7, rwidth=0.85)
    plt.title(f'Histograma de "{var}"')

    # Gráfico Q-Q
    plt.subplot(1, 2, 2)
    stats.probplot(data[var], dist="norm", plot=plt)
    plt.title(f'Gráfico Q-Q de "{var}"')
    plt.tight_layout()
    plt.show()
```


Por otro lado, la homogeneidad de la varianza se refiere a la suposición de que las varianzas son iguales en los diferentes grupos que estamos comparando. La prueba de Levene es una prueba estadística que podemos utilizar para evaluar la homogeneidad de las varianzas.

```
from scipy.stats import levene

# Comprobando la homogeneidad de la varianza para las variables seleccionadas
variables_to_check = [('sex', 'output'), ('age', 'output'),
                      ('cp', 'output'), ('trtbps', 'output')]

for var_pair in variables_to_check:
    _, p_value = levene(data[var_pair[0]], data[var_pair[1]])
    if p_value < 0.05:
        print(f"Las varianzas entre {var_pair[0]} y {var_pair[1]} no son homogéneas")
    else:
        print(f"Las varianzas entre {var_pair[0]} y {var_pair[1]} son homogéneas")
```

Las varianzas entre sex y output no son homogéneas
 Las varianzas entre age y output no son homogéneas
 Las varianzas entre cp y output no son homogéneas
 Las varianzas entre trtbps y output no son homogéneas

- 4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos.
 En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc. Aplicar al menos tres métodos de análisis diferentes.

La Prueba de Chi-Cuadrado es útil para verificar la independencia de dos variables categóricas. En nuestro caso, podríamos usarla para verificar si la salida 'output' (si un paciente tiene o no una enfermedad cardíaca) es independiente de variables como el sexo o el tipo de dolor en el pecho ('cp').

```
from scipy.stats import chi2_contingency

# Prueba de Chi-Cuadrado
crosstab = pd.crosstab(data['sex'], data['output'])
_, p_value, _, _ = chi2_contingency(crosstab)
print(f"Chi-Cuadrado p-value: {p_value}")
```

Chi-Cuadrado p-value: 1.8767776216941503e-06

Como podemos observar, el valor del Chi-Cuadrado es muy pequeño lo que indica que existe una dependencia significativa entre el sexo del paciente y la presencia de enfermedad cardíaca por lo que, podríamos afirmar que el género tiene un impacto en el riesgo de enfermedad cardíaca.

Con la Prueba T de Student podemos comparar la media de dos grupos y ver si son significativamente diferentes. Por ejemplo, podríamos comparar las edades medias de los pacientes que sufrieron un ataque al corazón con los que no lo hicieron.

```
from scipy.stats import ttest_ind

# Prueba T de Student
group1 = data[data['output'] == 1]['age']
group2 = data[data['output'] == 0]['age']
_, p_value = ttest_ind(group1, group2)
print(f"T-Test p-value: {p_value}")
```

T-Test p-value: 7.524801303442373e-05

El resultado del p-value nos indica que las edades medias de los pacientes con o sin enfermedades cardíacas son significativamente diferentes por lo que la edad puede no ser un factor indicativo del riesgo de la enfermedad.

Por último, el Análisis de correlación nos permite ver si existe una relación entre dos variables numéricas como, por ejemplo, si existe una correlación entre la edad y el colesterol.

```
from scipy.stats import pearsonr

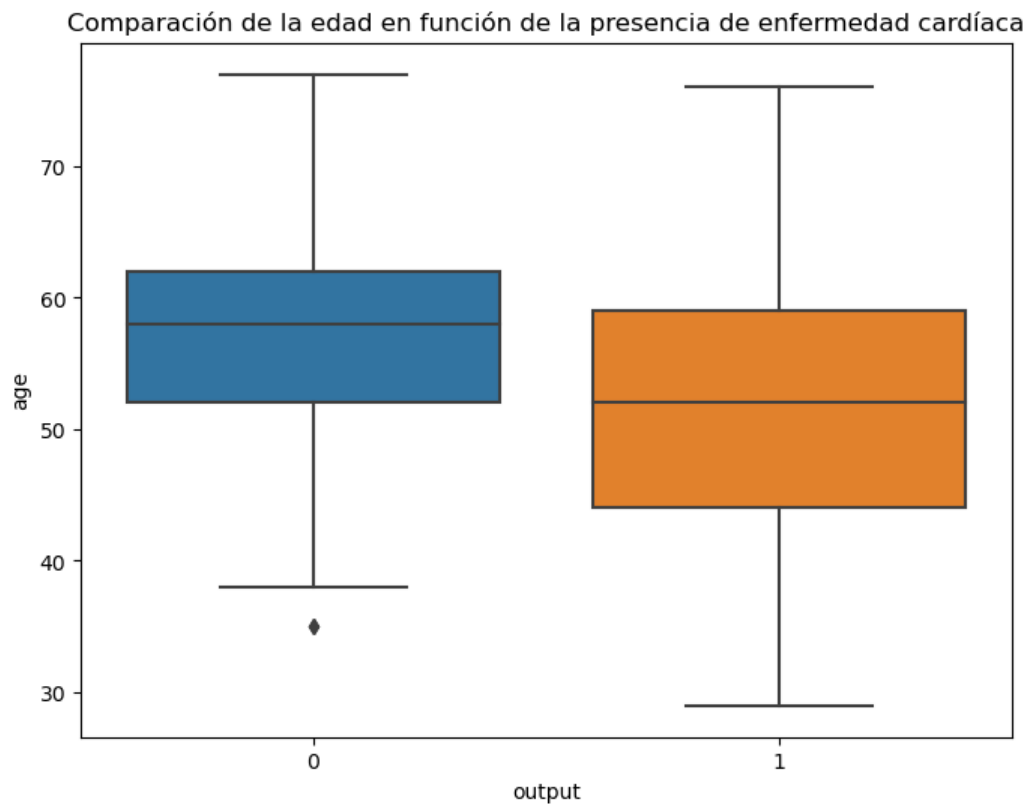
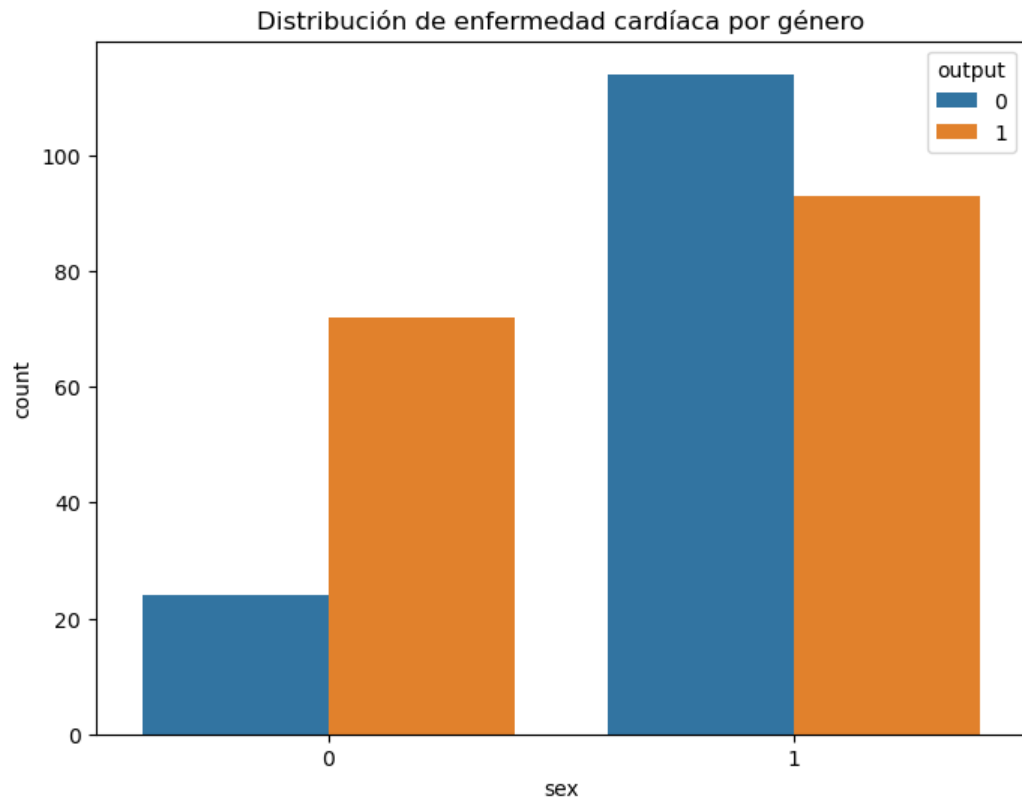
# Análisis de correlación
_, p_value = pearsonr(data['age'], data['chol'])
print(f"Pearson Correlation p-value: {p_value}")
```

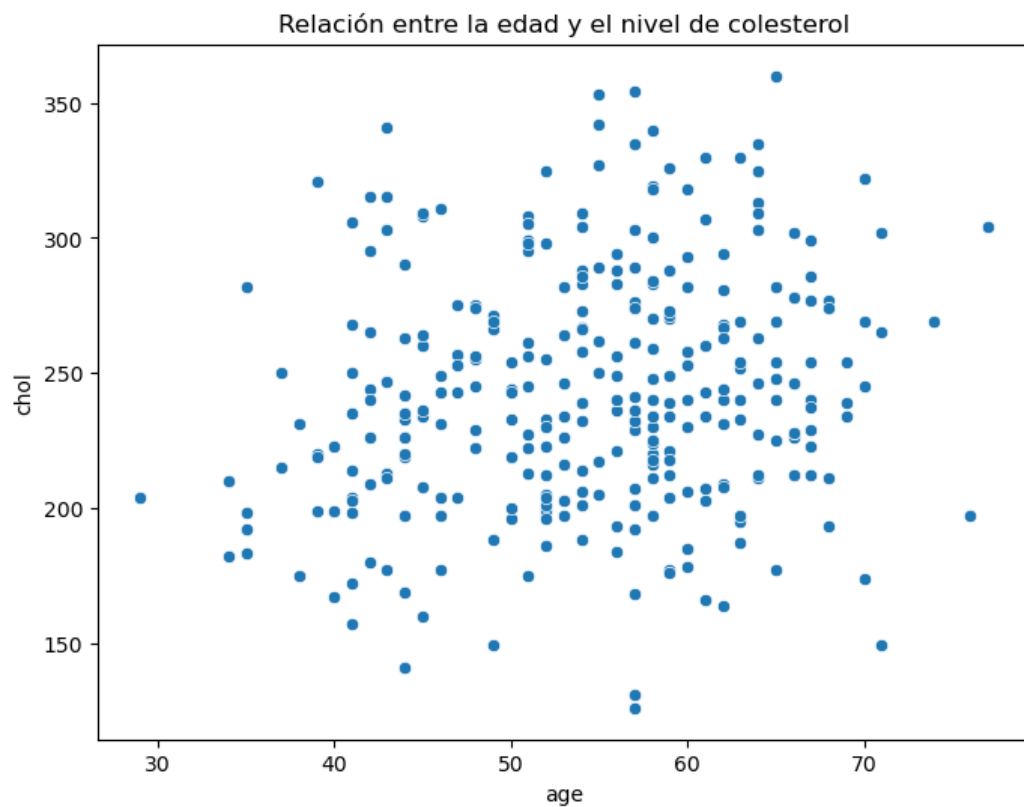
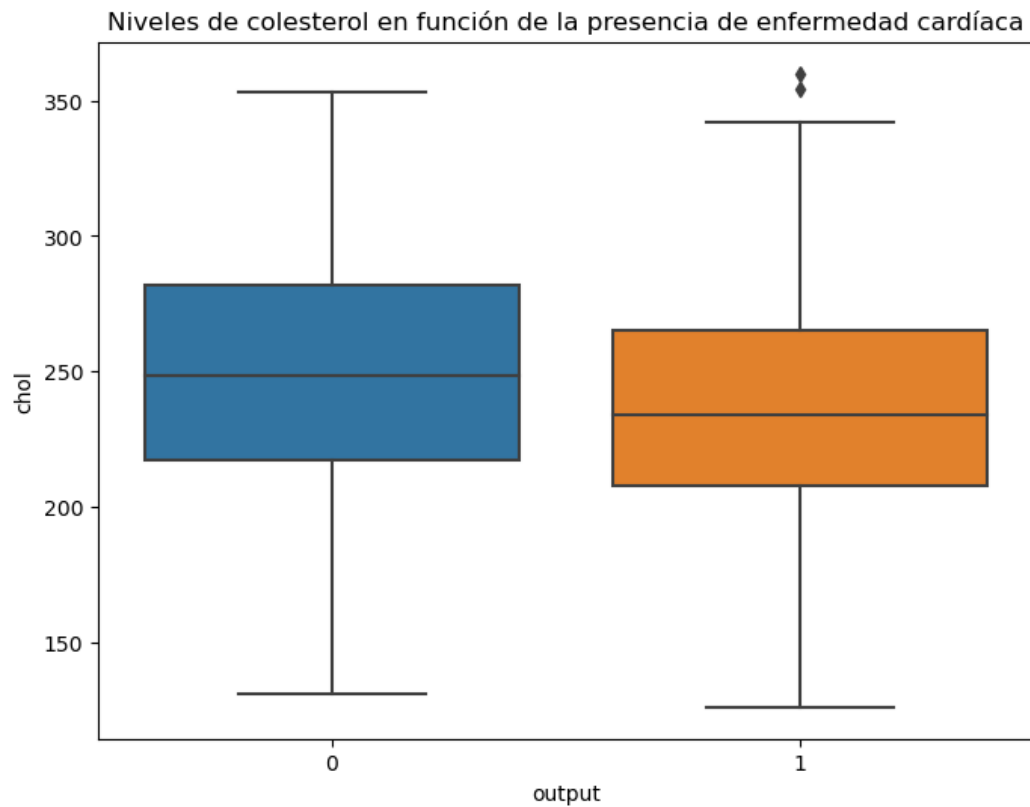
Pearson Correlation p-value: 0.0021890787419814673

El p-value en este caso nos está indicando que existe una correlación significativa entre la edad y el colesterol. Aunque la correlación no implica causalidad, podríamos afirmar que a medida que los pacientes envejecen, tienen niveles más altos de colesterol.

5. **Representación de los resultados** a partir de tablas y gráficas. Este apartado se puede responder a lo largo de la práctica, sin necesidad de concentrar todas las representaciones en este punto de la práctica.

A continuación, se muestran algunos de los resultados obtenidos para los grupos de estudio definidos en el apartado 4.1.





6. Resolución del problema. A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema?

Según los resultados obtenidos podemos llegar a las siguientes conclusiones:

- *El sexo (sex) de los individuos tiene un efecto significativo en la variable de salida (output), con un valor p muy por debajo del umbral de 0.05. Esto indica que la probabilidad de que la diferencia observada sea por casualidad es extremadamente baja.*
- *La edad (age) de los individuos también tiene un efecto significativo en la variable de salida. Al comparar las edades de aquellos con y sin enfermedad cardíaca, se puede observar que aquellos con enfermedad cardíaca tienden a ser más jóvenes en promedio.*
- *El nivel de colesterol (chol) también muestra una diferencia significativa entre aquellos con y sin enfermedad cardíaca. Sin embargo, en este caso, el nivel medio de colesterol es ligeramente mayor en aquellos sin enfermedad cardíaca.*
- *La correlación entre la edad y el nivel de colesterol es positiva pero débil. Esto sugiere que, aunque a medida que la gente envejece tiende a tener niveles de colesterol ligeramente más altos, este no es un fuerte predictor de la enfermedad cardíaca en este conjunto de datos.*

En resumen, los resultados obtenidos permiten dar una respuesta parcial al problema planteado. Podemos concluir que tanto el sexo como la edad parecen ser indicadores más fuertes de la presencia de enfermedad cardíaca que el nivel de colesterol. Sin embargo, es importante tener en cuenta que estas conclusiones se basan en este conjunto de datos específico y pueden no ser generalizables a la población en general.

Además, los resultados no deben interpretarse como que el nivel de colesterol no es importante para la salud cardíaca en general, sino que, en este conjunto de datos específico, la edad y el sexo parecen tener una mayor influencia. Los resultados de este estudio son el producto de análisis estadísticos y deben interpretarse como indicativos en lugar de definitivos.

Este estudio podrían utilizarse como punto de partida para futuras investigaciones que podrían explorar más a fondo estas relaciones en una muestra más grande y diversa.

7. **Código.** Hay que adjuntar el código, preferiblemente en R, con el que se ha realizado la limpieza, análisis y representación de los datos. Si lo preferís, también podéis trabajar en Python.

El código se ha ido proporcionando conforme se iban realizando las diferentes preguntas. A continuación, se adjunta el código de manera ordenada y continua.

```

1. # Importar la biblioteca necesaria
2. import pandas as pd
3.
4. # Carga el conjunto de datos
5. data = pd.read_csv('/kaggle/input/heart-attack-analysis-prediction-dataset/heart.csv')
6.
7. # Muestra las primeras filas del conjunto de datos
8. data.head()
9.
10. # Verificar si hay elementos vacíos en el conjunto de datos
11. data.isnull().sum()
12.
13. # Verificar si hay ceros en el conjunto de datos
14. (data == 0).sum()
15.
16. # Calcular el IQR para cada columna
17. Q1 = data.quantile(0.25)
18. Q3 = data.quantile(0.75)
19. IQR = Q3 - Q1
20.
21. # Definir los límites superior e inferior para los valores extremos
22. lower_bound = Q1 - 1.5 * IQR
23. upper_bound = Q3 + 1.5 * IQR
24.
25. # Identificar los valores extremos
26. outliers = data[(data < lower_bound) | (data > upper_bound)].count()
27. print(outliers)
28.
29. # Importar la biblioteca necesaria
30. import numpy as np
31.
32. # Calcular la mediana para cada columna
33. medians = data.median()
34.
35. # Reemplazar los outliers por la mediana en cada columna
36. for column in ['trtbps', 'chol', 'thalachh', 'oldpeak', 'caa', 'thall']:
37.     data[column] = np.where((data[column] < lower_bound[column]) |
38.                             (data[column] > upper_bound[column]),
39.                             medians[column],
40.                             data[column])
41.
42. from scipy.stats import Shapiro
43.
44. # Realizar la prueba de Shapiro-Wilk en cada columna
45. for column in data.columns:
46.     _, p_value = shapiro(data[column])
47.     if p_value < 0.05:
48.         print(f"{column} no sigue una distribución normal")
49.     else:
50.         print(f"{column} sigue una distribución normal")
51.
52.

```

```

53. import matplotlib.pyplot as plt
54. import scipy.stats as stats
55.
56. variables_to_check = ['age', 'sex', 'cp', 'trtbps', 'chol', 'fbs', 'restecg',
57.                        'thalachh', 'exng', 'oldpeak', 'slp', 'caa', 'thall']
58.
59. for var in variables_to_check:
60.     # Histograma
61.     plt.figure(figsize=(10, 4))
62.     plt.subplot(1, 2, 1)
63.     plt.hist(data[var], bins='auto', alpha=0.7, rwidth=0.85)
64.     plt.title(f'Histograma de "{var}"')
65.
66.     # Gráfico Q-Q
67.     plt.subplot(1, 2, 2)
68.     stats.probplot(data[var], dist="norm", plot=plt)
69.     plt.title(f'Gráfico Q-Q de "{var}"')
70.     plt.tight_layout()
71.     plt.show()
72.
73. from scipy.stats import Levene
74.
75. # Comprobando la homogeneidad de la varianza para las variables seleccionadas
76. variables_to_check = [('sex', 'output'), ('age', 'output'),
77.                        ('cp', 'output'), ('trtbps', 'output')]
78.
79. for var_pair in variables_to_check:
80.     _, p_value = levene(data[var_pair[0]], data[var_pair[1]])
81.     if p_value < 0.05:
82.         print(f"Las varianzas entre {var_pair[0]} y {var_pair[1]} no son homogéneas")
83.     else:
84.         print(f"Las varianzas entre {var_pair[0]} y {var_pair[1]} son homogéneas")
85.
86. from scipy.stats import chi2_contingency
87.
88. # Prueba de Chi-Cuadrado
89. crosstab = pd.crosstab(data['sex'], data['output'])
90. _, p_value, _, _ = chi2_contingency(crosstab)
91. print(f"Chi-Cuadrado p-value: {p_value}")
92.
93. from scipy.stats import ttest_ind
94.
95. # Prueba T de Student
96. group1 = data[data['output'] == 1]['age']
97. group2 = data[data['output'] == 0]['age']
98. _, p_value = ttest_ind(group1, group2)
99. print(f"T-Test p-value: {p_value}")
100.
101. from scipy.stats import pearsonr
102.
103. # Análisis de correlación
104. _, p_value = pearsonr(data['age'], data['chol'])
105. print(f"Pearson Correlation p-value: {p_value}")
106.
107. import seaborn as sns
108.
109. # Distribución de la enfermedad por género
110. plt.figure(figsize=(8, 6))
111. sns.countplot(data=data, x='sex', hue='output')
112. plt.title('Distribución de enfermedad cardíaca por género')
113. plt.show()
114.

```



```

115. # Comparar las edades entre los grupos con y sin enfermedad cardíaca
116. plt.figure(figsize=(8, 6))
117. sns.boxplot(data=data, x='output', y='age')
118. plt.title('Comparación de la edad en función de enfermedad cardíaca')
119. plt.show()
120.
121. # Visualizar la relación entre edad y nivel de colesterol
122. plt.figure(figsize=(8, 6))
123. sns.scatterplot(data=data, x='age', y='chol')
124. plt.title('Relación entre la edad y el nivel de colesterol')
125. plt.show()
126.
127. # Comparar los niveles de colesterol entre los grupos con y sin enfermedad cardíaca
128. plt.figure(figsize=(8, 6))
129. sns.boxplot(data=data, x='output', y='chol')
130. plt.title('Niveles de colesterol en función de la presencia de enfermedad cardíaca')
131. plt.show()

```

Repositorio: https://github.com/spulidos-UOC/M2.851_PRA2/

8. **Vídeo.** Realizar un breve vídeo explicativo de la práctica (**máximo 10 minutos**), donde ambos integrantes del equipo expliquen con sus propias palabras el desarrollo de la práctica, basándose en las preguntas del enunciado para justificar y explicar el código desarrollado. Este vídeo se deberá entregar a través de un enlace al Google Drive de la UOC (<https://drive.google.com/>...), junto con enlace al repositorio Git entregado.

Repositorio:

<https://drive.google.com/file/d/1V1EIFRrNozdgDz5GDIfopnNMIg0QIoPd/view?usp=sharing>

| Contribuciones | Firma |
|---|--------------------------------|
| <ul style="list-style-type: none"> Investigación previa Redacción de las respuestas Desarrollo del código Participación en el vídeo | <i>Salvador Pulido Sánchez</i> |