

Estadísticas descriptivas

Christopher Evans A01742073 ITC

En este trabajo vamos a estar analizando un set de datos acerca de la diabetes, Esta data list incluye las variables de embarazos, glucosa, presión arterial, Grosor de piel, Insulina, IMC, Función de pedigrí de Diabetes, Edad y Resultado.

Dentro de este trabajo se va a estar detallando algunos puntos clave de nuestro data en general y se va a crear un data list nuevo con solo las variables de Grosor de piel, Insulina, IMC y Resultado y analizando las características que tiene el nuevo data set.

```
In [30]: #importa librerías
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df=pd.read_csv("diabetes.csv")
```

Descripción de Variables

Descripcion general de data set

```
In [9]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Pregnancies           768 non-null   int64
1   Glucose               768 non-null   int64
2   BloodPressure         768 non-null   int64
3   SkinThickness         768 non-null   int64
4   Insulin               768 non-null   int64
5   BMI                   768 non-null   float64
6   DiabetesPedigreeFunction 768 non-null   float64
7   Age                   768 non-null   int64
8   Outcome               768 non-null   int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

```
In [6]: df.describe()
```

Out[6]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Diabetes
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.347108
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.471405
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	0.000000

In [2]: `df.nunique()`

Out[2]:

Pregnancies	17
Glucose	136
BloodPressure	47
SkinThickness	51
Insulin	186
BMI	248
DiabetesPedigreeFunction	517
Age	52
Outcome	2

dtype: int64

Después de nuestro análisis general, este trabajo va a estar utilizando las siguientes variables como análisis

Skin Thickness Cuantitativa Continua

Insulin Cuantitativa Continua

BMI Cuantitativa Continua

Outcome Categorica Nominal

In [17]:

```
vars_interes = ['SkinThickness', 'Insulin', 'BMI', 'Outcome']
df_selected = df[vars_interes]
df_selected.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 4 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   SkinThickness    768 non-null    int64
1   Insulin          768 non-null    int64
2   BMI              768 non-null    float64
3   Outcome          768 non-null    int64
dtypes: float64(1), int64(3)
memory usage: 24.1 KB
```

In [26]: `df_selected.describe()`

Out[26]:

	SkinThickness	Insulin	BMI	Outcome
count	768.000000	768.000000	768.000000	768.000000
mean	20.536458	79.799479	31.992578	0.348958
std	15.952218	115.244002	7.884160	0.476951
min	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	27.300000	0.000000
50%	23.000000	30.500000	32.000000	0.000000
75%	32.000000	127.250000	36.600000	1.000000
max	99.000000	846.000000	67.100000	1.000000

In [24]:

```
for col in vars_interes:
    minimo = df_selected[col].min()
    maximo = df_selected[col].max()
    rango = maximo - minimo
    print(col, "Min:", minimo, "Max", maximo, "Rango", rango)
```

```
SkinThickness Min: 0 Max 99 Rango 99
Insulin Min: 0 Max 846 Rango 846
BMI Min: 0.0 Max 67.1 Rango 67.1
Outcome Min: 0 Max 1 Rango 1
```

In [25]:

```
# Estadísticas descriptivas
estadisticas = df_selected.describe()

# Desviación estándar
desviacion_std = df_selected.std()

# Mostrar resultados
print("Estadísticas Descriptivas:")
print(estadisticas)
print("Desviación estándar:")
print(desviacion_std)
```

Estadísticas Descriptivas:

	SkinThickness	Insulin	BMI	Outcome
count	768.000000	768.000000	768.000000	768.000000
mean	20.536458	79.799479	31.992578	0.348958
std	15.952218	115.244002	7.884160	0.476951
min	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	27.300000	0.000000
50%	23.000000	30.500000	32.000000	0.000000
75%	32.000000	127.250000	36.600000	1.000000
max	99.000000	846.000000	67.100000	1.000000

Desviación estándar:

SkinThickness	15.952218
Insulin	115.244002
BMI	7.884160
Outcome	0.476951

dtype: float64

Grosor del pliegue cutáneo: Se puede observar que la media y mediana son bastante cercanas, sugiriendo que la distribución de los datos es bastante simétrica. Dicho lo anterior, tenemos mínimos y máximos muy extremos, indicando que tenemos datos erróneos o sin tratar.

Nivel de insulina Podemos observar una desviación estándar grande que nos indica una dispersión en los datos significativa. Esto combinado con una media y mediana muy diferentes, nos indica una distribución con sesgo a la derecha.

Índice de Masa Corporal: La media y la mediana son muy similares indicando una distribución simétrica, vemos que el valor medio de los datos es 32 mostrando que la mayoría de individuos son personas con sobrepeso posiblemente indicando un prejuicio natural sobre nuestros otros datos, notamos un mínimo de 0 que no es posible en la escala de índice de masa corporal indicando error en nuestros datos.

Consultas

```
In [29]: # 1. Personas con BMI > 25
personas_bmi_mayor_25 = df[df['BMI'] > 25].shape[0]
print("Personas con BMI > 25:", personas_bmi_mayor_25)

# 2. Personas con SkinThickness > 20 y Outcome = 1
piel_y_positivo = df[(df['SkinThickness'] > 20) & (df['Outcome'] == 1)].shape[0]
print("Personas con SkinThickness > 20 y Outcome = 1:", piel_y_positivo)

# 3. Personas con BMI < 25 y Insulin > promedio
promedio_insulina = df['Insulin'].mean()
bajo_bmi_alta_insulina = df[(df['BMI'] < 25) & (df['Insulin'] > promedio_insulina)]
print("Personas con BMI < 25 e Insulina > promedio:", bajo_bmi_alta_insulina)
```

Personas con BMI > 25: 645

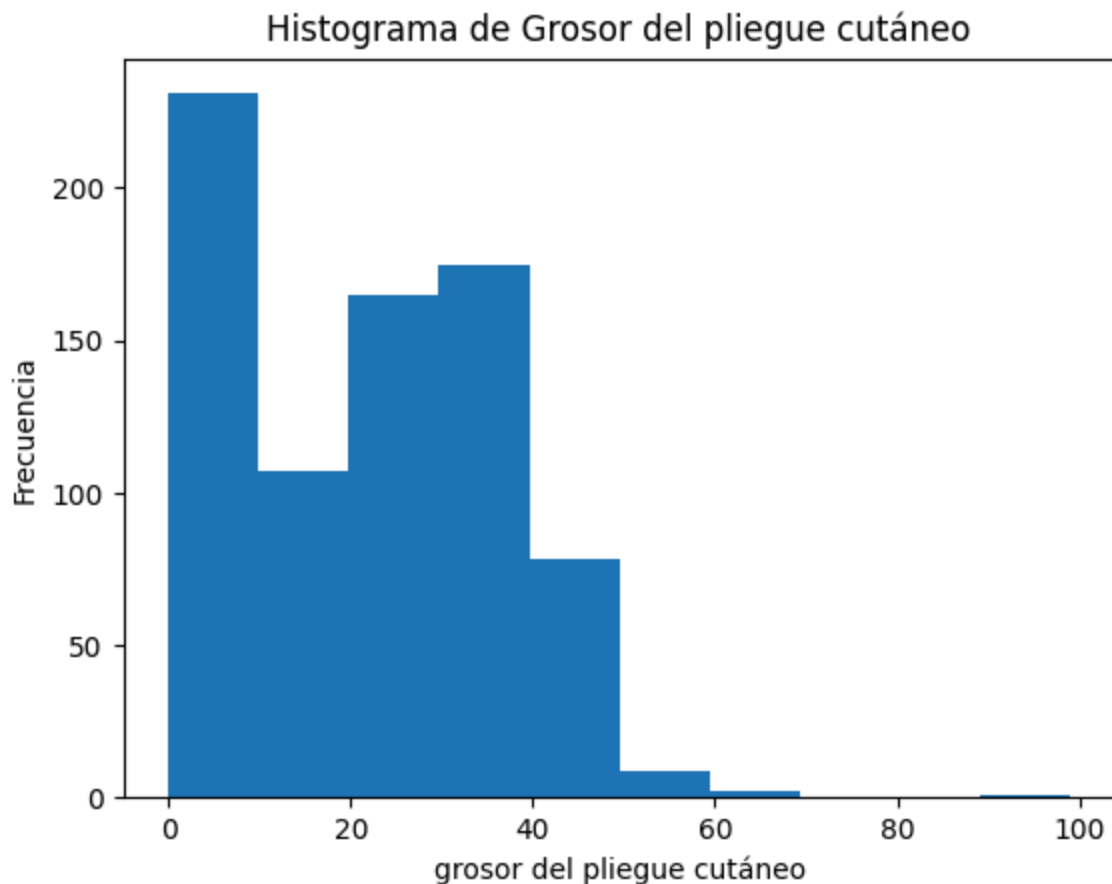
Personas con SkinThickness > 20 y Outcome = 1: 163

Personas con BMI < 25 e Insulina > promedio: 24

Visualización de Datos

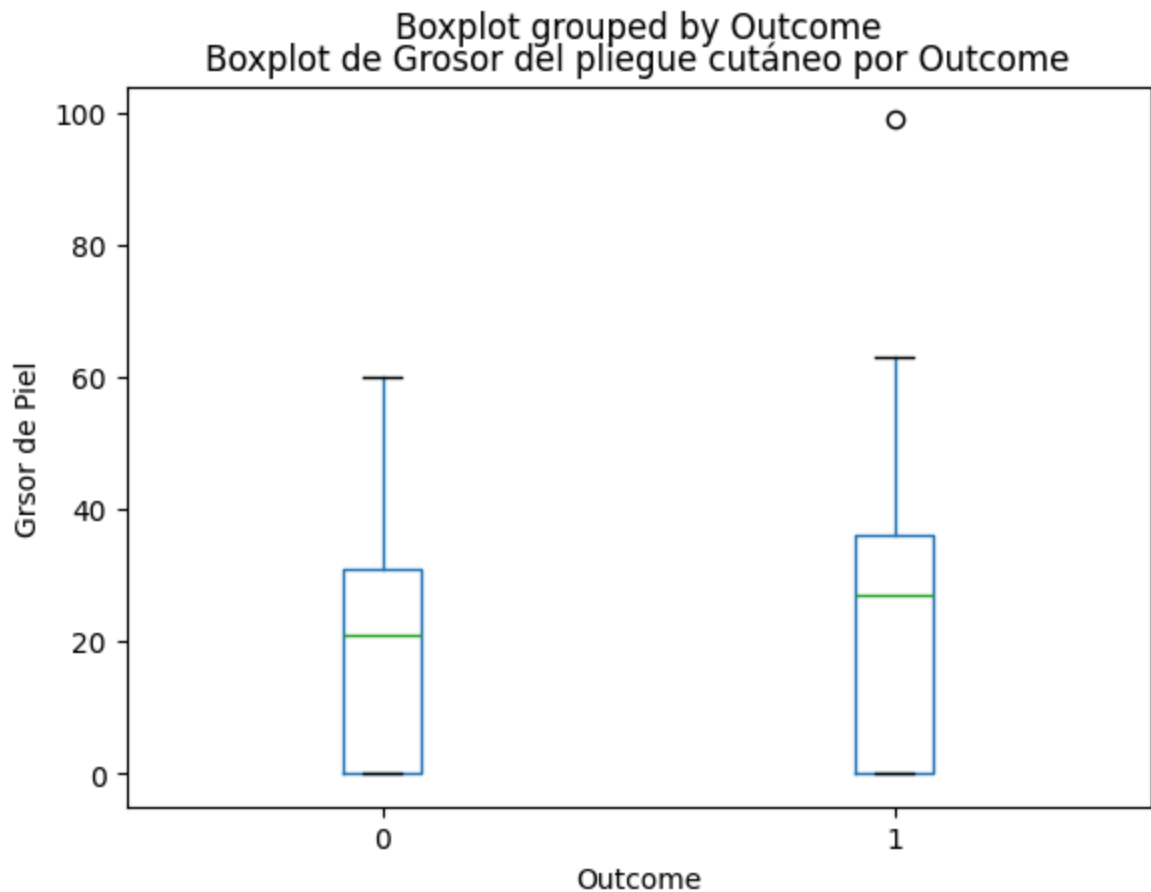
Grosor del pliegue cutáneo

```
In [53]: # Histograma para grosor del pliegue cutáneo
plt.hist(df_selected['SkinThickness'])
plt.title('Histograma de Grosor del pliegue cutáneo')
plt.xlabel('grosor del pliegue cutáneo')
plt.ylabel('Frecuencia')
plt.show()
```



Se observa una frecuencia muy alta en 0, indicando valores incompletos o erróneos.

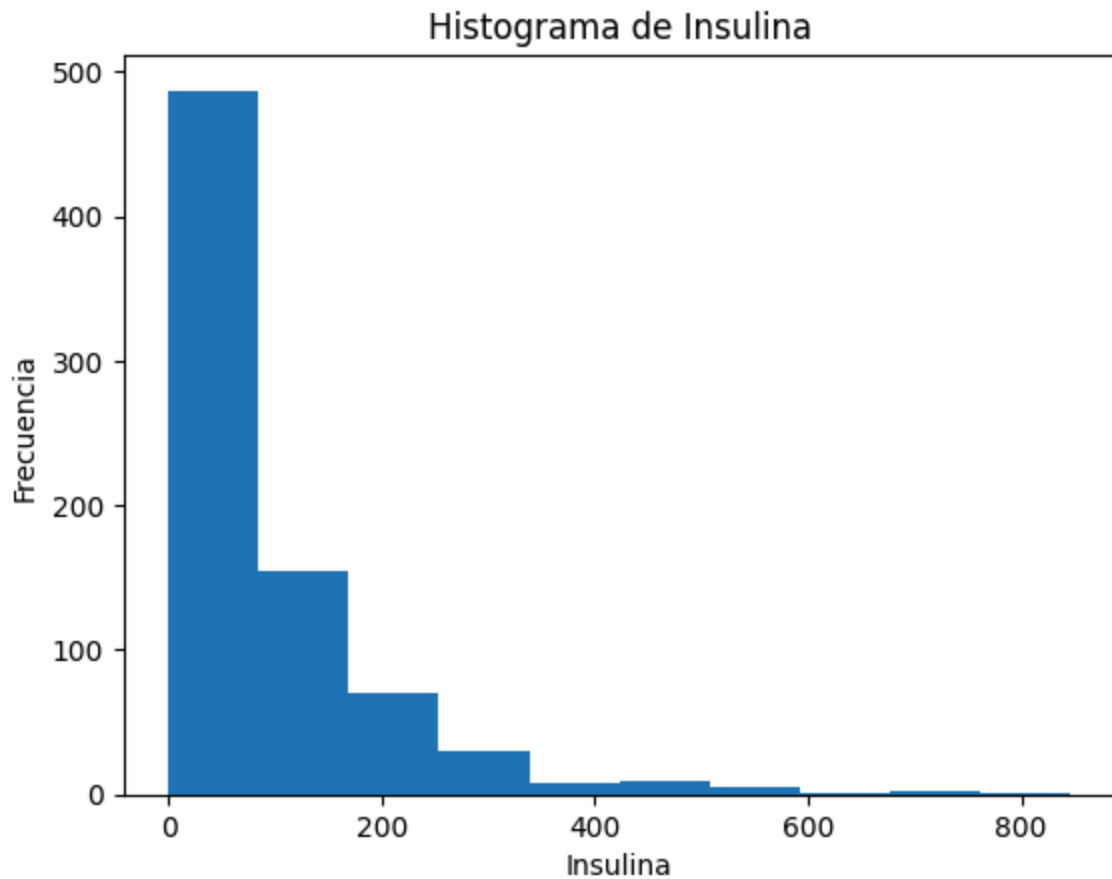
```
In [79]: # Boxplot de Skin Thickness por Outcome
df_selected.boxplot(column='SkinThickness', by='Outcome', grid=False)
plt.title('Boxplot de Grosor del pliegue cutáneo por Outcome')
plt.xlabel('Outcome')
plt.ylabel('Grosor de Piel')
plt.show()
```



Podemos observar una frecuencia muy alta de registros con un valor de 0, indicando valores faltantes, si solo se toman encuentra los valores válidos vemos como estos se concentran entre 20 y 40 mm con una mediana de 23 mm, en el boxplot se observa que las personas con diabetes tienden a tener un pliegue cutáneo ligeramente mayor que quienes no la tienen.

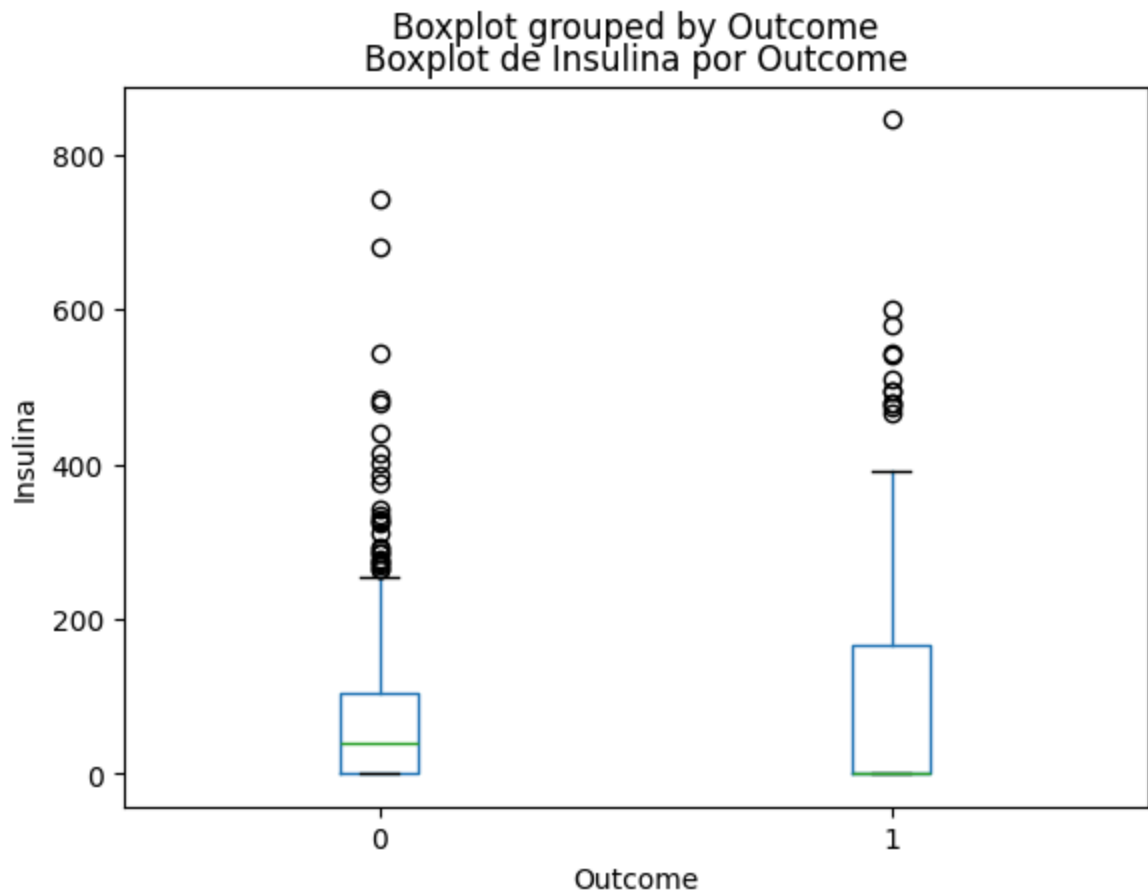
Niveles de Insulina

```
In [50]: # Histograma para Insulina
plt.hist(df_selected['Insulin'])
plt.title('Histograma de Insulina')
plt.xlabel('Insulina')
plt.ylabel('Frecuencia')
plt.show()
```



Se observa una frecuencia muy alta en 0, indicando valores de data set con bastantes personas saludables.

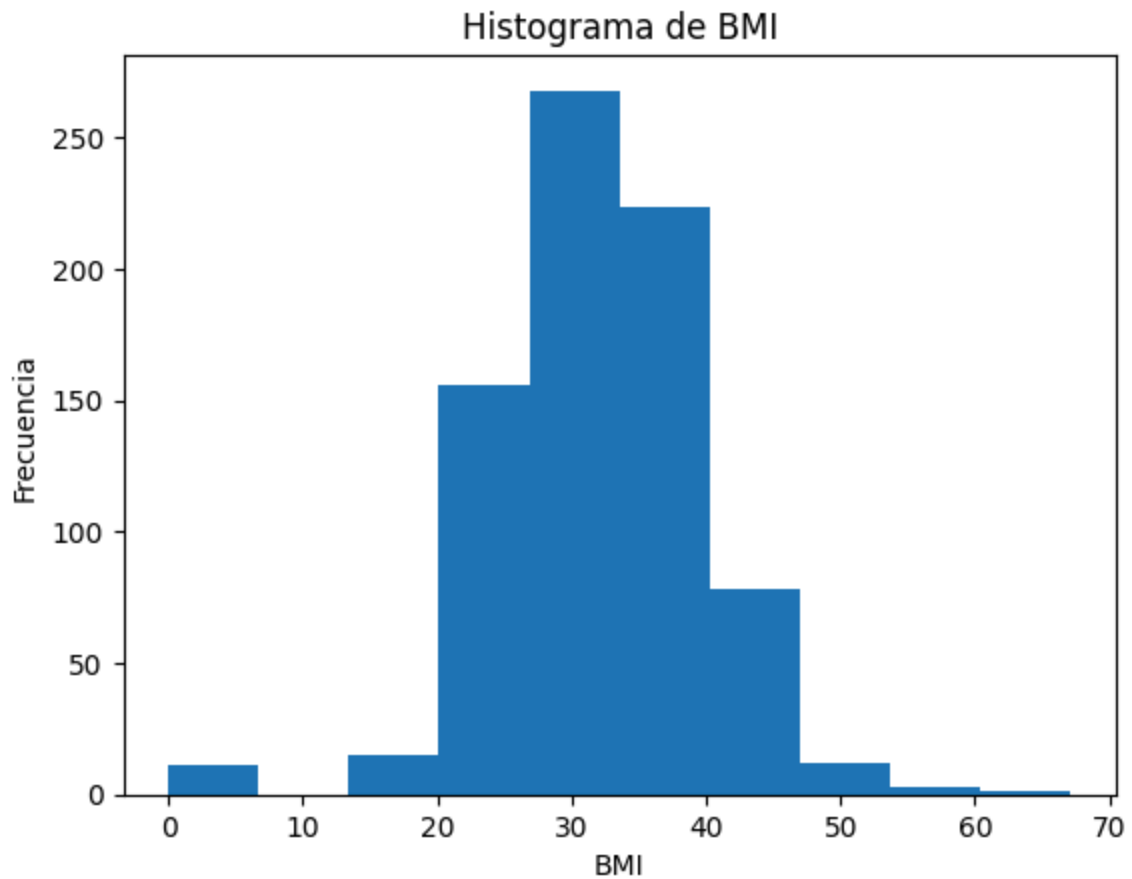
```
In [80]: df_selected.boxplot(column='Insulin', by='Outcome', grid=False)
plt.title('Boxplot de Insulina por Outcome')
plt.xlabel('Outcome')
plt.ylabel('Insulina')
plt.show()
```



La distribución de insulina muestra una alta concentración de valores bajos. La mayoría de los registros están por debajo de 200, con algunos valores altos que representan casos con mayor necesidad de insulina. Se observa que las personas con diabetes tienden a tener niveles de insulina más altos, sugiriendo una relación posible.

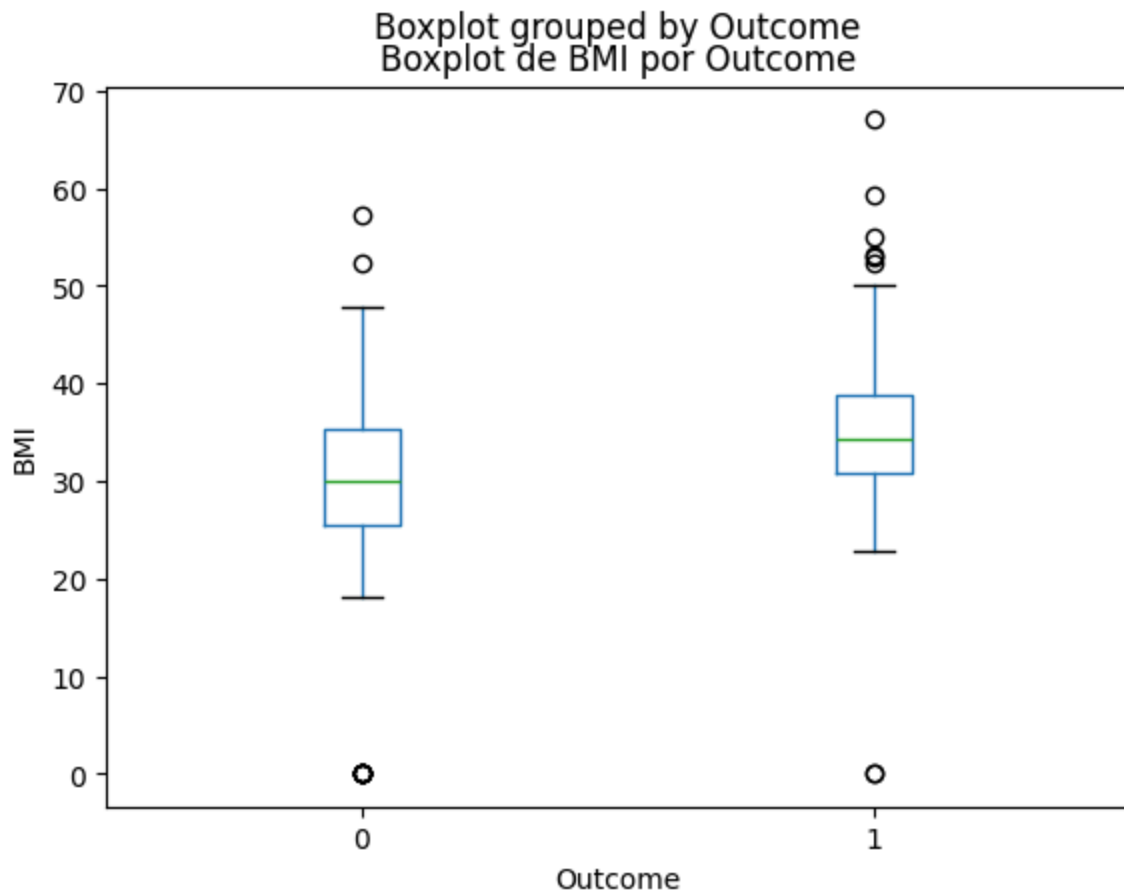
BMI

```
In [49]: # Histograma para BMI
plt.hist(df_selected['BMI'])
plt.title('Histograma de BMI')
plt.xlabel('BMI')
plt.ylabel('Frecuencia')
plt.show()
```

La distribución del BMI muestra que la mayoría de las personas se encuentran entre valores de 25 y 40, lo que indica un predominio de sobrepeso en nuestra muestra

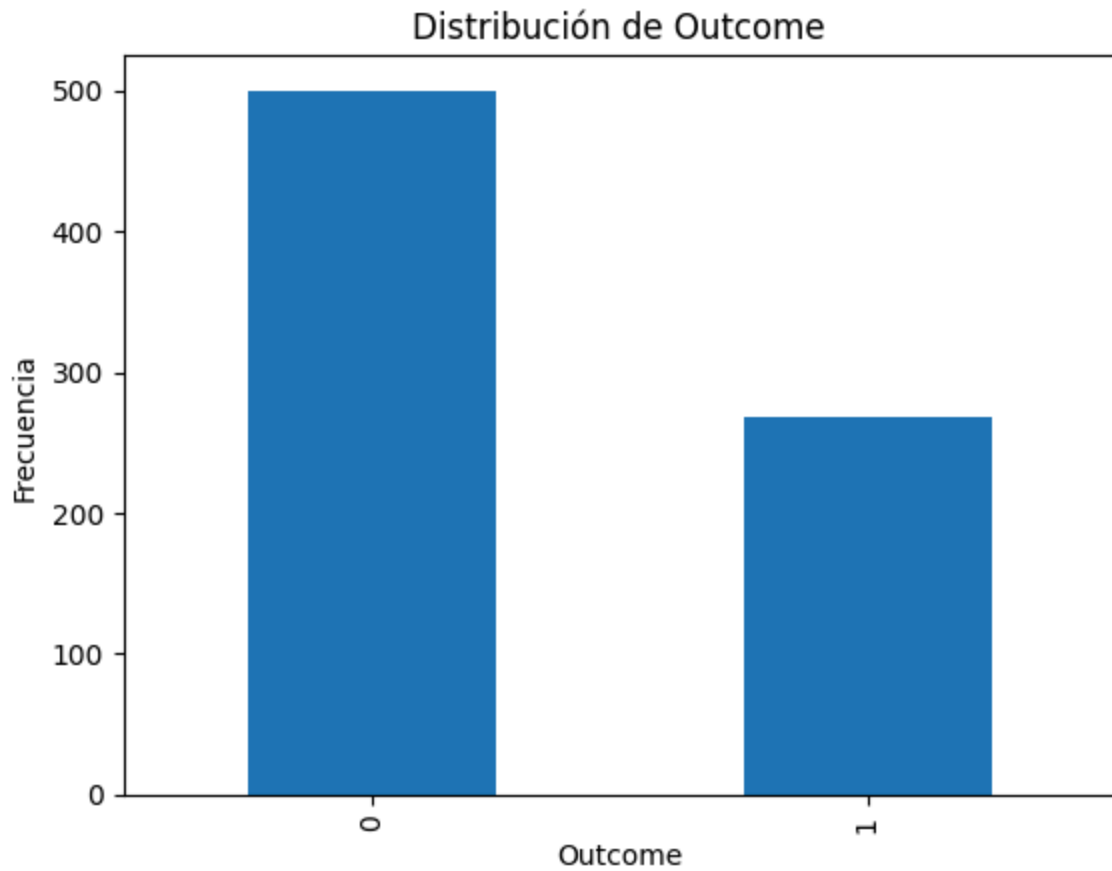
```
In [81]: df_selected.boxplot(column='BMI', by='Outcome', grid=False)
plt.title('Boxplot de BMI por Outcome')
plt.xlabel('Outcome')
plt.ylabel('BMI')
plt.show()
```



El box plot sugiere una posible relación entre un mayor BMI y el diagnóstico de diabetes.

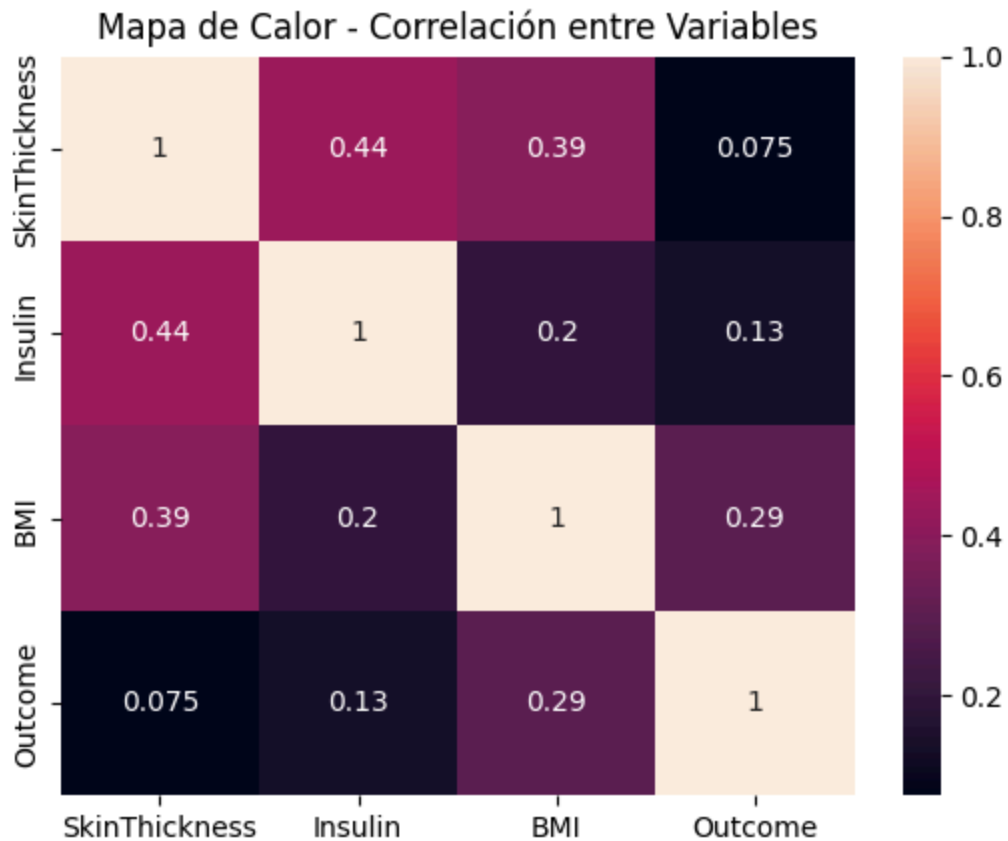
Outcome

```
In [62]: # Diagramas de barras
datos_outcome = df_selected['Outcome'].value_counts()
datos_outcome.plot(kind='bar')
plt.title('Distribución de Outcome')
plt.xlabel('Outcome')
plt.ylabel('Frecuencia')
plt.show()
```



En nuestra base de datos predominan personas sin diabetes con una relación aproximada de 2 personas saludables por cada persona enferma.

```
In [70]: # Mapa de calor de correlación
correlacion = df_selected.corr()
sns.heatmap(correlacion, annot=True)
plt.title('Mapa de Calor - Correlación entre Variables')
plt.show()
```



En nuestro mapa de calor vemos una correlación moderada entre el grosor de piel y la Insulina, y el grosor de piel y BMI indicando que estas variables aumentan juntas. Para nuestro outcome no tenemos ninguna variable individual con un nivel de correlación muy alto, indicando la importancia de usar un análisis de correlación múltiple o usar otras variables.

In []: