**Informe Estadístico #1**

**Análisis de datos Resultados Sabaneta Saber 11**

**Estadística Inferencial**

**Keyner José Rodriguez Chaverra**

**Andrés Fabián Leal Archila**

**Universidad UNAB**

**Bucaramanga**

**Santander  
  
2025**

**Introducción**

Para este estudio se utilizó Google Colab para correr fragmentos de código Python, para poder analizar un conjunto de datos extraído de un documento el cual se cargo a la plataforma. En este caso, se trabajó con los resultados de los ICFES Sabaneta 11, explorando sus diferentes variables y aplicando técnicas estadísticas para su correcta interpretación

El objetivo principal fue, además aprender más sobre el lenguaje de Python y la manipulación de datos, reconociendo variables numéricas, categóricas y sus subtipos. Con base en estas clasificaciones, se generaron diferentes tipos de gráficas según la naturaleza de cada variable, siguiendo pautas específicas para su representación. En la toma de decisiones fundamentadas en estadísticas a partir de datos reales.

La importancia de la estadística inferencial radica en que en función a métricas reales se pueden tomar decisiones fundamentadas para x o y motivo. Por ejemplo, en el ámbito educativo, analizar los resultados de pruebas como el ICFES puede ayudar a identificar debilidades en el currículo, asignar recursos de manera más eficiente o implementar estrategias pedagógicas específicas. En la vida cotidiana, este tipo de análisis es clave para predecir tendencias como el comportamiento del mercado, la evolución del clima o la propagación de enfermedades.

Asimismo, en este contexto el uso de la programación es una herramienta para optimizar y automatizar procesos, reduciendo el tiempo de trabajo y minimizando errores humanos. Herramientas como lo pueden ser Python o cualquier otro lenguaje de programación facilitan la carga, limpieza, visualización, interpretación y análisis de manera más integra de grandes volúmenes de datos, lo que se resume en conclusiones rápidas y precisas para la toma de decisiones.

**Metodología**

Siguiendo las indicaciones del docente, el proceso inició con el acceso a un repositorio de GitHub que contenía la guía y los recursos necesarios para el análisis. Después de una breve explicación introductoria, se procedió a subir a la plataforma Google Colab un documento con información correspondiente a los resultados de las pruebas ICFES.

Este archivo se integró en un cuaderno de trabajo proveniente del repositorio, el cual incluía una plantilla previamente diseñada por el profesor. A partir de esta base, se personalizó el código y se adaptaron los procedimientos para realizar la exploración, clasificación y representación gráfica de las variables presentes en el conjunto de datos.

El desarrollo contempló la identificación del tipo de cada variable (numérica o categórica), la selección de las representaciones gráficas más adecuadas para cada caso y la interpretación de los resultados. Este proceso no solo permitió cumplir con los requerimientos planteados, sino que también sentó las bases para la elaboración del presente informe estadístico, consolidando los conocimientos en análisis de datos y programación en Python.

**Resultados**

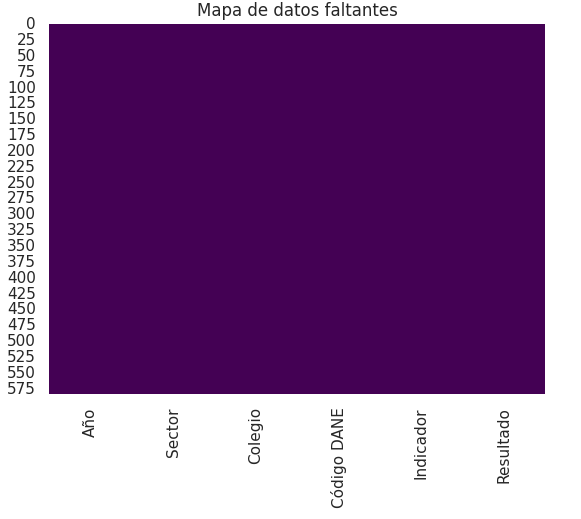
Se presentan los resultados obtenidos de los datos de Resultados ICFES Sabaneta 11. Para cada variable se generaron las visualizaciones correspondientes según su naturaleza (categórica/numérica), siguientes las pautas vistas y aplicando als funciones de Python en La herramienta Colab.

**1. Análisis general del dataset**

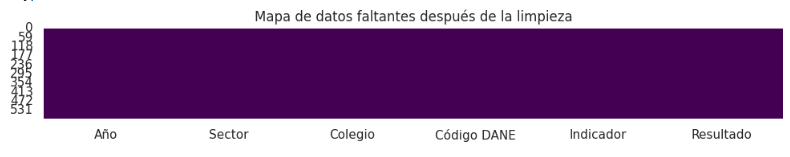
Se verificó que el conjunto de datos estaba compuesto por **583 registros** y 6 columnas:

* Año
* Sector
* Colegio
* Código DANE
* Indicador
* Resultado

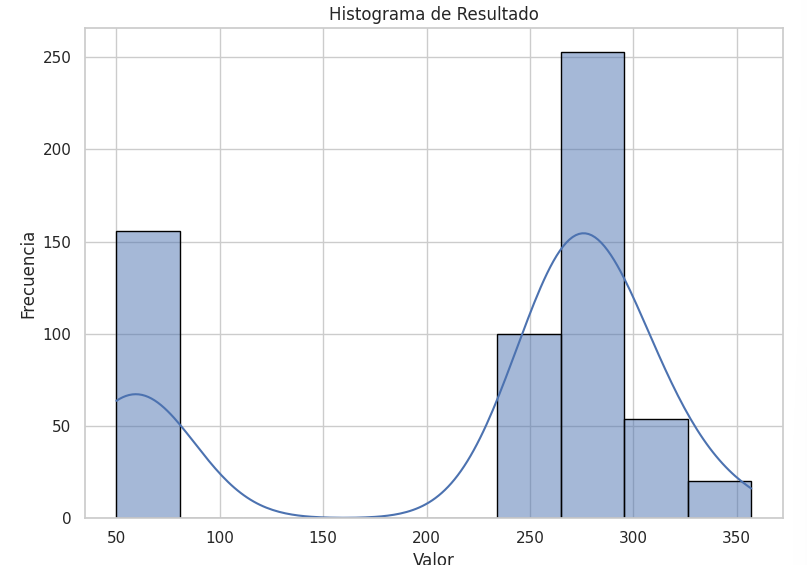
No se identificaron valores faltantes, lo que permitió trabajar con la base de datos completa.



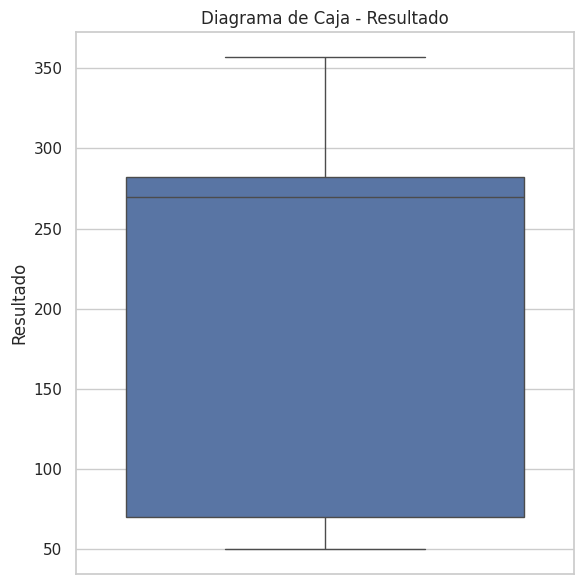
Se realizo de igual manera una pequeña limpieza con el objetivo de verificar si realmente faltaban datos. En este caso simplemente se simplifico el rango eliminando algunas columnas.



Luego de esto se realizó un análisis de la variable resultado, la única variable numérica de la que realmente podría realizarse un histograma y en función a su valor y frecuencia, y un diagrama de caja para revisar su dispersión y donde se ubicaban la mayoría de estos datos.



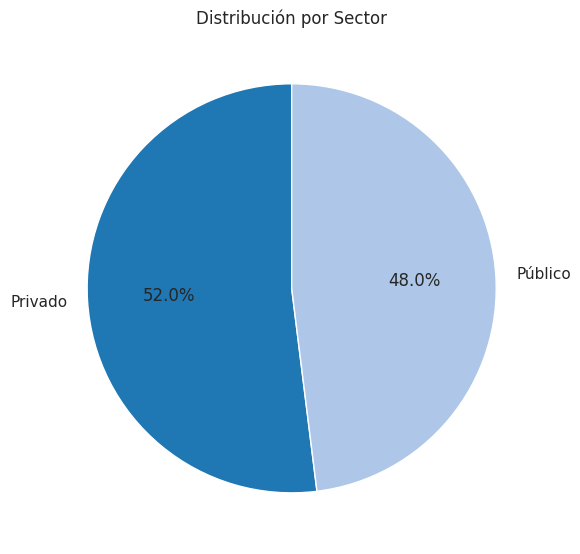
En el **histograma** se aprecia una clara concentración de puntajes en el rango de **250 a 350**, lo que indica que la mayoría de las instituciones obtuvo resultados por encima de 200, con casos aislados por debajo de 100. Esta distribución revela una **desigualdad en el desempeño**, marcada por la diferencia significativa entre los rangos bajos (50–70) y los altos (250–350).



En el **diagrama de caja**, se observa una **variabilidad moderada**, lo que sugiere que los puntajes de la mayoría de los colegios fueron relativamente similares. La ausencia de una cantidad significativa de valores atípicos confirma que no existen resultados extremadamente alejados del promedio, ni por debajo de 50 ni por encima de 350, lo que refuerza la idea de que los datos se mantienen dentro de un rango esperado.

**2. Distribución por Sector**

El dataset incluye instituciones tanto públicas como privadas. El análisis mostró que la participación entre ambos sectores es relativamente equilibrada, con una ligera predominancia del sector privado.

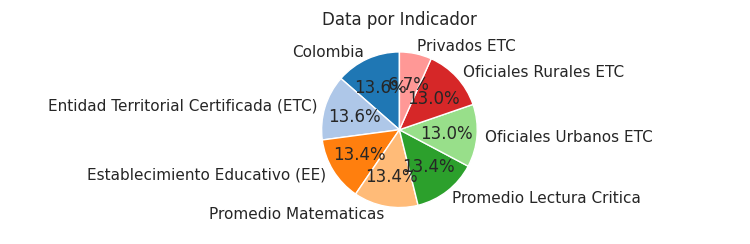


En la distribución por colegios presentados si bien la información es difícil de interpretar en la imagen, los datos dan a entender de donde viene la mayoría de los resultados (es decir, una distribución del porcentaje de aporte respecto número de pruebas). Se da a entender que esto varía de la cantidad de alumnos presentes, y en su mayoría, suelen tener un aporte del 5-7% de la participación en el estudio.

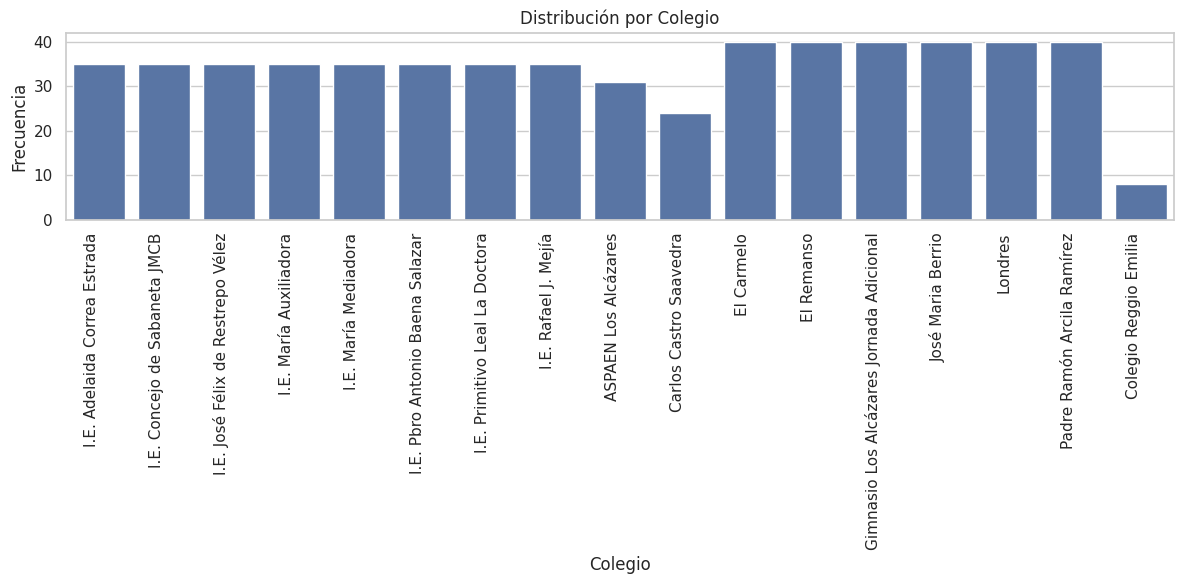


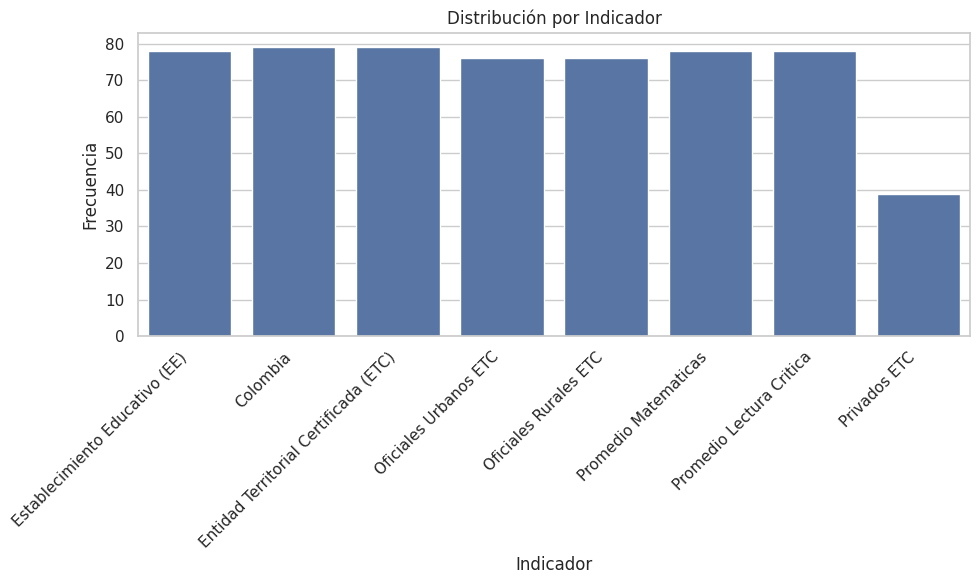
La gráfica de pastel “Datos por indicador” representa la proporción o participación relativa de cada categoría de un indicador específico dentro del total de datos analizados.

En este caso, cada sector del pastel corresponde a un indicador evaluado en el ICFES (por ejemplo: Lectura Crítica, Matemáticas, Ciencias Naturales, Inglés, Sociales y Ciudadanas, etc.), y su tamaño refleja qué porcentaje de registros o valores corresponde a cada uno respecto al conjunto total.



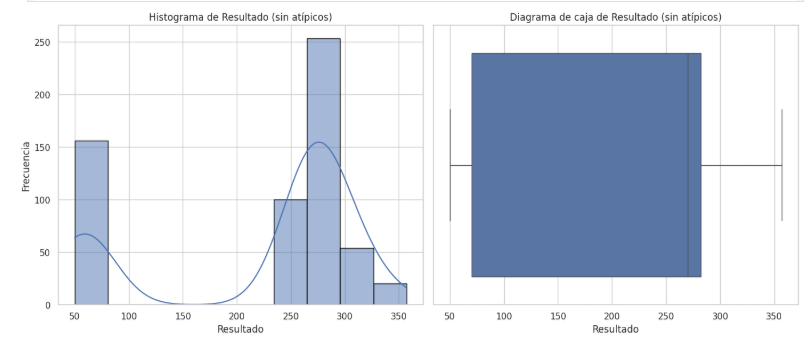
Tambien se presenta una representación de estos 2 estudios con gráfico de barras, para poder interpretar de mejor manera los valores según el colegio y la frecuencia en cada uno de estos. Se concluye que la mayoría de las instituciones tienen una media de 30/40 % de distribución, exceptuando 2 instituciones.





Y aquí se concluye que aquellos que tuvieron menor aporte al estudio fueron las instituciones privados ETC.

Retomando parte del estudio de la variable resultado, en cierto punto del código se preparó un histograma de estos datos pero sin datos atípicos. Se observaron estudios practicamente iguales a los presentes anteriormente, puesto que como se había mencionado, a pesar del sesgo que hay, no se reportaron datos extremadamente atípicos a la media, solamente disperson entre sí (refiriendonos a los puntajes cercanos a 50)

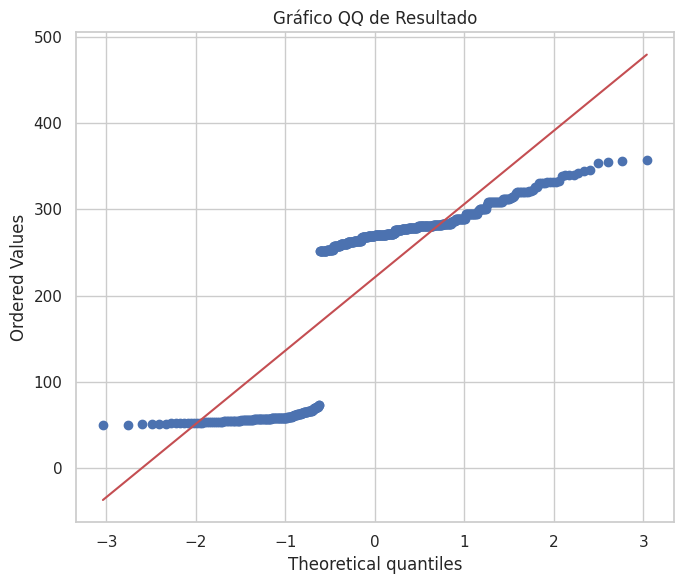


Para finalizar el reporte, se hizo un análisis de un test de normalidad visual y una gráfica Q-Q

La normalidad visual consiste en verificar si la distribución de los datos se asemeja a una normal usando gráficos como el histograma y el Q-Q plot. En este último, la cercanía de los puntos a la línea diagonal indica que los datos se ajustan bien a una distribución normal, mientras que desviaciones pronunciadas muestran que la distribución difiere de la normalidad.

En el histograma, la distribución de los puntajes no presenta la simetría característica de una distribución normal, sino que se observan concentraciones en rangos específicos, evidenciando cierto sesgo y una dispersión desigual. Esto sugiere que los datos no siguen una campana de Gauss perfecta, lo cual puede estar relacionado con diferencias marcadas en el rendimiento entre instituciones.

En el gráfico Q-Q, los puntos no se alinean de forma precisa sobre la diagonal de referencia. En lugar de seguir la recta, se observan desviaciones tanto en los extremos como en la zona central, lo que confirma que la variable *Resultado* no se ajusta completamente a la normalidad teórica. Este patrón indica la presencia de colas más pesadas o ligeras que las de una distribución normal, lo cual debe considerarse en el análisis estadístico posterior.



**Conclusiones**

El análisis de los resultados del ICFES Sabaneta 11 permitió comprender mejor cómo se distribuye el rendimiento académico entre las diferentes instituciones del municipio. La información analizada, sin valores faltantes, facilitó un estudio claro y completo de los datos.

Los histogramas y diagramas de caja evidenciaron que la mayoría de los puntajes se concentran entre 250 y 350, con muy pocos casos por debajo de 100. Esto refleja una diferencia marcada en el desempeño entre colegios, pero también una variabilidad moderada que indica que muchos obtuvieron resultados similares. Además, la ausencia de valores extremos sugiere que no hubo casos atípicos muy alejados de la media.

Al observar la distribución por colegio y sector, se encontró que tanto instituciones públicas como privadas participan de forma importante, aunque con diferencias en la proporción de estudiantes evaluados. Las gráficas por indicador mostraron que la mayoría de los colegios se concentran en rangos intermedios de desempeño.

La revisión de la normalidad visual y el gráfico Q-Q confirmó que la variable *Resultado* no sigue completamente la línea esperada de una distribución normal, lo que indica que los puntajes presentan un sesgo y cierta concentración en rangos específicos.

En general, este trabajo permitió no solo aplicar herramientas estadísticas y de programación para analizar datos, sino también extraer información valiosa que puede ayudar a identificar patrones de rendimiento y posibles áreas de mejora en la educación local.

**Resumen del Código Utilizado**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from scipy.stats import shapiro, kstest, anderson, jarque\_bera, probplot

# Cargar archivo CSV

df = pd.read\_csv("ResultadosSabanetaSaber11\_estudio\_estadistica\_1.csv")

# Información básica

print("Información del dataset")

print(df.info())

print("Primeras filas del dataset")

print(df.head())

print("Resumen estadístico del dataset")

print(df.describe())

# Valores faltantes

print("\\nValores faltantes por columna:")

print(df.isnull().sum())

# Visualización de nulos

sns.heatmap(df.isnull(), cbar=False, cmap="viridis")

plt.title("Mapa de datos faltantes")

plt.show()

# Eliminar filas con datos faltantes y verificar limpieza

# Eliminar filas con datos faltantes

df\_filtrado = df.dropna()

# Mostrar el número de filas después de la limpieza

print(f"\\nNúmero de filas después de eliminar datos faltantes: {df\_filtrado.shape[0]}")

# Identificar y mostrar el número de datos faltantes por columna

faltantes = df\_filtrado.isnull().sum()

print("\\nDatos faltantes por columna después de la limpieza:")

print(faltantes)

# Visualizar el mapa de calor de datos faltantes (debería estar vacío si la limpieza fue exitosa)

plt.figure(figsize=(10, 2))

sns.heatmap(df\_filtrado.isnull(), cbar=False, cmap="viridis")

plt.title("Mapa de datos faltantes después de la limpieza")

plt.tight\_layout()

# plt.savefig("mapa\_datos\_faltantes.png") # Quitar comentario si se quiere guardar la figura

plt.show()

# Histograma para Resultado

plt.figure(figsize=(8, 6))

sns.histplot(df\_filtrado['Resultado'], bins=10, edgecolor='black', kde=True)

plt.xlabel('Valor')

plt.ylabel('Frecuencia')

plt.title('Histograma de Resultado')

plt.tight\_layout()

plt.show()

# Diagrama de caja para Resultado

plt.figure(figsize=(6, 6))

sns.boxplot(y=df\_filtrado['Resultado'])

plt.title('Diagrama de Caja - Resultado')

plt.tight\_layout()

plt.show()

# Countplot for Sector

plt.figure(figsize=(10, 6))

sns.countplot(data=df\_filtrado, x='Sector')

plt.title('Distribución por Sector')

plt.xlabel('Sector')

plt.ylabel('Frecuencia')

plt.xticks(rotation=45, ha='right') # Mejora la legibilidad de las etiquetas

plt.tight\_layout()

plt.show()

# Countplot for Colegio

plt.figure(figsize=(12, 6))

sns.countplot(data=df\_filtrado, x='Colegio')

plt.title('Distribución por Colegio')

plt.xlabel('Colegio')

plt.ylabel('Frecuencia')

plt.xticks(rotation=90, ha='right') # Rotate labels for better visibility

plt.tight\_layout()

plt.show()

# Countplot for Indicador

plt.figure(figsize=(10, 6))

sns.countplot(data=df\_filtrado, x='Indicador')

plt.title('Distribución por Indicador')

plt.xlabel('Indicador')

plt.ylabel('Frecuencia')

plt.xticks(rotation=45, ha='right') # Mejora la legibilidad de las etiquetas

plt.tight\_layout()

plt.show()

# Calcular Q1, Q3 e IQR para la variable 'Resultado'

Q1\_resultado = df\_filtrado['Resultado'].quantile(0.25)

Q3\_resultado = df\_filtrado['Resultado'].quantile(0.75)

IQR\_resultado = Q3\_resultado - Q1\_resultado

# Definir los límites para considerar un dato como atípico en 'Resultado'

limite\_inferior\_resultado = Q1\_resultado - 1.5 \* IQR\_resultado

limite\_superior\_resultado = Q3\_resultado + 1.5 \* IQR\_resultado

# Filtrar el DataFrame para eliminar outliers en 'Resultado'

df\_sinatipicos = df\_filtrado[

(df\_filtrado['Resultado'] >= limite\_inferior\_resultado) & (df\_filtrado['Resultado'] <= limite\_superior\_resultado)

]

# Mostrar la cantidad de datos restantes

print(f"Número de datos restantes después de eliminar outliers en 'Resultado': {len(df\_sinatipicos)}")

# Histograma para Resultado sin atípicos

fig, axes = plt.subplots(1, 2, figsize=(14, 6))

sns.histplot(df\_sinatipicos['Resultado'], bins=10, edgecolor='black', kde=True, ax=axes[0])

axes[0].set\_xlabel('Resultado')

axes[0].set\_ylabel('Frecuencia')

axes[0].set\_title('Histograma de Resultado (sin atípicos)')

# Diagrama de caja para Resultado sin atípicos

sns.boxplot(x=df\_sinatipicos['Resultado'], ax=axes[1])

axes[1].set\_xlabel('Resultado')

axes[1].set\_title('Diagrama de caja de Resultado (sin atípicos)')

plt.tight\_layout()

plt.show()

# List of variable to evaluate

columnas\_a\_evaluar = ['Resultado']

# Dictionary to store results

resultados\_normalidad = {}

# Evaluate each variable

for col in columnas\_a\_evaluar:

if col in df\_sinatipicos.columns: # Use df\_sinatipicos after outlier removal

datos = df\_sinatipicos[col].dropna()

# Shapiro-Wilk

stat\_sw, p\_sw = shapiro(datos)

# Kolmogorov-Smirnov

media, desv = np.mean(datos), np.std(datos)

stat\_ks, p\_ks = kstest(datos, "norm", args=(media, desv))

# Anderson-Darling

resultado\_ad = anderson(datos)

stat\_ad = resultado\_ad.statistic

crit\_ad = resultado\_ad.critical\_values[2] # 5%

ad\_normal = stat\_ad < crit\_ad

# Jarque-Bera

stat\_jb, p\_jb = jarque\_bera(datos)

# Store results

resultados\_normalidad[col] = {

'Shapiro-Wilk': p\_sw,

'Kolmogorov-Smirnov': p\_ks,

'Anderson-Darling': f'Estadístico: {stat\_ad:.4f}, Crítico (5%): {crit\_ad:.4f}',

'Jarque-Bera': p\_jb,

'Normal': sum([

p\_sw > 0.05,

p\_ks > 0.05,

ad\_normal,

p\_jb > 0.05

]) >= 3

}

# Display results in a table

print("Resultados de los tests de normalidad (valores p):\\n")

for var, res in resultados\_normalidad.items():

print(f"Variable: {var}")

print(f" Shapiro-Wilk p = {res['Shapiro-Wilk']:.4f}")

print(f" Kolmogorov-Smirnov p = {res['Kolmogorov-Smirnov']:.4f}")

print(f" Anderson-Darling {res['Anderson-Darling']}")

print(f" Jarque-Bera p = {res['Jarque-Bera']:.4f}")

if res['Normal']:

print(" → Distribución normal (según mayoría de tests)\\n")

else:

print(" → NO es distribución normal (según mayoría de tests)\\n")

# QQ plots for each variable

fig, axes = plt.subplots(1, 1, figsize=(7, 6)) # Changed to 1 subplot

for i, col in enumerate(columnas\_a\_evaluar):

if col in df\_sinatipicos.columns:

probplot(df\_sinatipicos[col].dropna(), dist="norm", plot=axes) # Plot on the single axes

axes.set\_title(f"Gráfico QQ de {col}")

plt.tight\_layout()

plt.show()