**Trabajo 2**

La base de datos con la que le corresponde trabajar se obtiene como una muestra aleatoria que corresponde a los resultados obtenidos por los jóvenes de todo el país en las pruebas Saber 11 del año 2022. La información aparece agregada por Colegio y solo abarca los colegios de Antioquia. Dicha base contiene las variables: NATURALEZA (Colegio Oficial o No-Oficial), JORNADA (Asociado a los diferentes tipos de jornada de estudio), PROMLECT (Promedio obtenido por el colegio en Lectura Crítica) y PROMMATE (Promedio obtenido por el colegio en Matemáticas).

**Pasos previos**

Antes de realizar los puntos que se pide, es necesario cargar la base de datos usando alguna herramienta como Python o R. En nuestro caso se usó la primera:

Antes de empezar, se importan las bibliotecas necesarias:

|  |
| --- |
| # Bibliotecas necesarias  import pandas as pd  import numpy as np  import scipy.stats as stats  from sklearn.linear\_model import LinearRegression  import matplotlib.pyplot as plt  from sklearn.metrics import mean\_squared\_error, mean\_absolute\_error, r2\_score |

Luego, se obtienen los datos del archivo para realizar el análisis:

|  |
| --- |
| # Leer archivo csv  raw\_data\_url = "https://raw.githubusercontent.com/repos-especializacion-UdeA/estadistica/refs/heads/main/trabajo2/DatosTrabajo2EAE20242.csv"  # Leer el archivo CSV  df = pd.read\_csv(raw\_data\_url) |

Antes de realizar los cálculos, es bueno conocer si hay valores nulos en cada columna.

|  |
| --- |
| df.isnull().sum() |

Como ya se tiene los datos cargados es posible empezar a realizar cada uno de los puntos del taller.

1. **Intervalo de confianza**: Construir en Python un IC para al 95% de confianza necesario para responder:

¿Puede afirmarse que el resultado medio obtenido por los colegios en Lectura es superior a 45 puntos? Justifique su respuesta.

Para construir una estimación del intervalo de confianza para una media poblacional desconocida necesitamos datos de una muestra aleatoria. En este caso, seleccionamos la columna relacionada (**PROMLECT**) con los datos que nos interesan:

|  |
| --- |
| # Datos del problema  data\_prom\_lectura = df.PROMLECT.copy() |

Procedemos a plantear el problema teniendo en cuenta que cada uno de estos datos es una VA (variable aleatoria) de modo que sea la VA relacionada con el promedio obtenido en la prueba de lectura por el *i-esimo* colegio.

Las VAs (con ) constituyen la muestra aleatoria (MA) del promedio obtenido en la prueba.

El primer paso consiste en obtener la media y la desviación estándar de la muestra:

En Python esto se hace así:

|  |
| --- |
| # Calculo de la media y la desviacion estandar muestral  n = data\_prom\_lectura.shape[0]  X = data\_prom\_lectura.mean()  S = data\_prom\_lectura.std()  # Despliegue de los resultados  print(f"n = {n}")  print(f"X = {X:.5f}")  print(f"S = {S:.5f}") |

El resultado mostrado en pantalla fue el siguiente:

|  |
| --- |
| n = 1807  X = 50.32153  S = 7.02803 |

De este modo tenemos que:

Como para la muestra aleatoria la distribución no es normal, el número de datos es grande () y la varianza () es desconocida, el intervalo de confianza esta dado por:

Donde de acuerdo al enunciado, el nivel de confianza es del 95% de modo que:

De modo que la expresión, con , para el queda:

Donde:

Con lo cual queda que el :

El código Python que hace lo anterior, se muestra a continuación:

|  |
| --- |
| NC = 95 # Nivel de confianza  alpha = 1 - NC/100  Z = stats.norm().ppf(1 - alpha/2)  print(f"Z = {Z:.5f}")  EBM = Z\*S/np.sqrt(n)  print(f"EBM = {EBM:.5f}")  IC\_left = X - EBM  IC\_right = X + EBM  print(f"({X:.5f} - {EBM:.5f},{X:.5f} + {EBM:.5f}) = ({IC\_left:.5f},{IC\_right:.5f})") |

El resultado a la salida se muestra a continuación:

|  |
| --- |
| Z = 1.95996  EBM = 0.32404  (50.32153 - 0.32404,50.32153 + 0.32404) = (49.99748,50.64557) |

|  |
| --- |
| **Conclusión**:  Como el intervalo en el cual se encuentra la media para la prueba de lectura está en el rango se puede afirmar con un nivel de confianza del 95% que el resultado promedio en la prueba de lectura es superior a 45 puntos. Esto se puede ver por qué el límite inferior del intervalo de confianza es 49.997 (mayor que 45).  Así, en resumen, dado que todo el intervalo está por encima de 45 puntos, podemos concluir que es muy poco probable que el promedio verdadero sea igual o menor a 45 puntos y, por lo tanto, podemos afirmar que el promedio es superior a 45 puntos con un alto nivel de confianza. |

1. ¿Se puede afirmar que el resultado medio obtenido por los colegios en Matemáticas es mayor en los Privados que en los Públicos? Justifique su respuesta mediante una prueba de hipótesis con una significancia del 4%.

De manera similar vamos obtener los datos de la muestra de la base de datos original:

|  |
| --- |
| # Informacion de la prueba de matematicas  # Instituciones publicas  notas\_mat\_publico = df[df['NATURALEZA'] == 'OFICIAL']['PROMMATE'].copy()  # Instituciones privadas  notas\_mat\_privado = df[df['NATURALEZA'] == 'NO OFICIAL']['PROMMATE'].copy() |

Tal y como se muestra en el fragmento de código anterior la muestra asociada al promedio obtenido por los colegios en matemáticas para las instituciones públicas se denominó **notas\_mat\_publico** y la asociada a las instituciones privadas se llamó **notas\_mat\_publico**.

El problema por analizar plantea la siguiente pregunta: ¿Se puede afirmar que el resultado medio obtenido por los colegios en Matemáticas es mayor en los Privados que en los Públicos?

Para resolverlo vamos a definir las siguientes variables aleatorias para el problema:

* : Promedio obtenido por el *i-esimo* colegio privado en la prueba de matemáticas.
* : Promedio obtenido por el *j-esimo* colegio público en la prueba de matemáticas.

De modo que, de acuerdo con la pregunta planteada, si es el resultado medio obtenido en matemáticas por los colegios privados y es la media obtenida en las pruebas de matemáticas de los colegios públicos lo que se nos pregunta es si lo cual llevado a la forma estándar para el planteamiento de hipótesis queda como: . De este modo tenemos el siguiente **planteamiento de hipótesis**:

* :
* :

El siguiente paso consiste en determinar la distribución de la prueba. Donde para el caso de cada tenemos:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Descripción | Promedio obtenido por el *i-esimo* colegio privado en la prueba de matemáticas | Promedio obtenido por el *j-esimo* colegio público en la prueba de matemáticas. |
| Numero de datos | Donde el código python que hace esto es:   |  | | --- | | nx = notas\_mat\_privado.shape[0]  print(f"nx = {nx}") | | Donde el código python que hace esto es:   |  | | --- | | ny = notas\_mat\_publico.shape[0]  print(f"ny = {ny}") | |
| Distribución | No normal con grande y varianza () desconocida. | No normal con grande y varianza () desconocida. |

Teniendo en cuenta la información de la tabla anterior, el estadístico de prueba a emplear para este caso es el siguiente, donde para este problema :

Ahora, procedamos a calcular los parámetros asociados a cada distribución antes de hacer el cálculo del estadístico de prueba:

|  |  |  |
| --- | --- | --- |
|  | Parámetros | Código python |
|  |  | A continuación, se adjunta el código python que realiza los cálculos requeridos:   |  | | --- | | # Colegio privado  nx = notas\_mat\_privado.shape[0]  X = notas\_mat\_privado.mean()  Sx = notas\_mat\_privado.std()  print(f"nx = {nx}")  print(f"X = {X:.5f}")  print(f"Sx = {Sx:.5f}") |   La salida de este código se muestra a continuación:   |  | | --- | | nx = 341  X = 52.75367  Sx = 9.59462 | |
|  |  | El código python que realiza los cálculos requeridos se muestra a continuación:   |  | | --- | | # Colegio publico  ny = notas\_mat\_publico.shape[0]  Y = notas\_mat\_publico.mean()  Sy = notas\_mat\_publico.std()  print(f"ny = {ny}")  print(f"Y = {Y:.5f}")  print(f"Sy = {Sy:.5f}") |   La salida de este código se muestra es:   |  | | --- | | ny = 1466  Y = 45.79127  Sy = 6.80895 | |

La siguiente tabla resume los resultados anteriores antes de sacar el estadístico:

|  |  |
| --- | --- |
|  |  |
|  |  |

Ahora ya es posible obtener el con :

|  |  |
| --- | --- |
| **Calculo manual** | **Código** |
|  | El código se muestra a continuación:   |  | | --- | | d0 = 0  Zc = (X - Y - d0) / ((Sx\*\*2 / nx + Sy\*\*2 / ny)\*\*0.5)  print(f"Zc = {Zc:.5f}") |   De modo que la salida es:   |  | | --- | | Zc = 12.67807 | |

Finalmente, el último paso es comparar el estadístico de prueba con el valor crítico determinado por el nivel de significancia que se requiere para la prueba, siendo este para el caso: . Esto es:

Usando python para calcular el valor P en este caso tenemos que:

|  |
| --- |
| VP = 1 - stats.norm.cdf(Zc)  print(f"VP = {VP:e}") |

Que da como salida:

|  |
| --- |
| VP = 0.000000e+00 |

|  |
| --- |
| **Conclusión**:  Como se rechaza la hipótesis nula aceptándose la hipótesis alternativa de modo que podemos decir con un nivel de significancia del 4% que el resultado medio en las pruebas de Matemáticas es mejor en los colegios privados que en los públicos, esto es . |

1. Analice si hay una relación lineal entre las variables “Promedio obtenido por el colegio en Lectura Crítica” y “Promedio obtenido por el colegio en Matemáticas”, mediante una regresión lineal con todos sus respectivos componentes.

**Obtención y análisis grafico de los datos**

Inicialmente se obtienen los datos y se hace un diagrama de dispersión para visualizar gráficamente la relación entre estos. Para este caso se definieron estos de la siguiente manera:

* **Variable independiente ()**: Nota promedio de la prueba de lectura crítica para cada colegio.
* **Variable dependiente ()**: Nota promedio de la prueba de matemáticas para cada colegio.

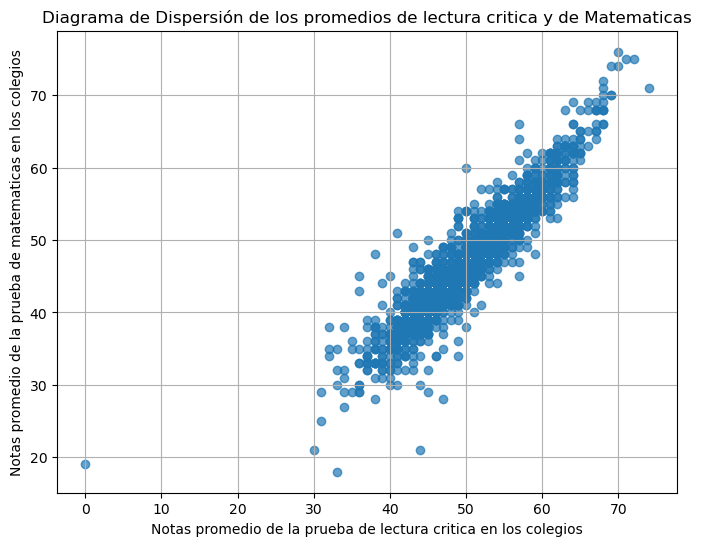
A continuación, se muestra el código python empleado:

|  |
| --- |
| # Preparar los datos (PROMLECT como variable independiente X y PROMMATE como variable dependiente y)  promedio\_notas = df.iloc[:][['PROMLECT','PROMMATE']].copy()  x = promedio\_notas.PROMLECT  y = promedio\_notas.PROMMATE |

Ahora obtengamos la gráfica de dispersión:

|  |
| --- |
| # Crear un diagrama de dispersión entre PROMLECT y PROMMATE  plt.figure(figsize=(8, 6))  plt.scatter(x, y, alpha=0.7)  # Añadir etiquetas y título  plt.title('Diagrama de Dispersión de los promedios de lectura critica y de Matematicas')  plt.xlabel('Notas promedio de la prueba de lectura critica en los colegios')  plt.ylabel('Notas promedio de la prueba de matematicas en los colegios')  # Mostrar el gráfico  plt.grid(True)  plt.show() |

La grafica de dispersión se muestra a continuación:



**Obtención del modelo**

Ahora vamos a obtener el modelo lineal simple, el cual es de la forma:

El objetivo es obtener el intercepto y la pendiente que definan la recta que mejor describe la relación entre los datos asociados a e y.

|  |
| --- |
| # 1. Preparar los datos  X = x.values.reshape(-1, 1)  # Variable independiente  y = y.values  # Variable dependiente  # 2. Crear el modelo de regresión lineal  modelo = LinearRegression()  # 3. Ajustar el modelo a los datos  modelo.fit(X, y)  # 4. Obtener el coeficiente y la intersección  B1 = modelo.coef\_[0]  # Coeficiente de la regresión  B0 = modelo.intercept\_  # Intersección (ordenada al origen)  print(f'Pendiente (B1): {B1}')  print(f'Intersepto (B0): {B0}') |

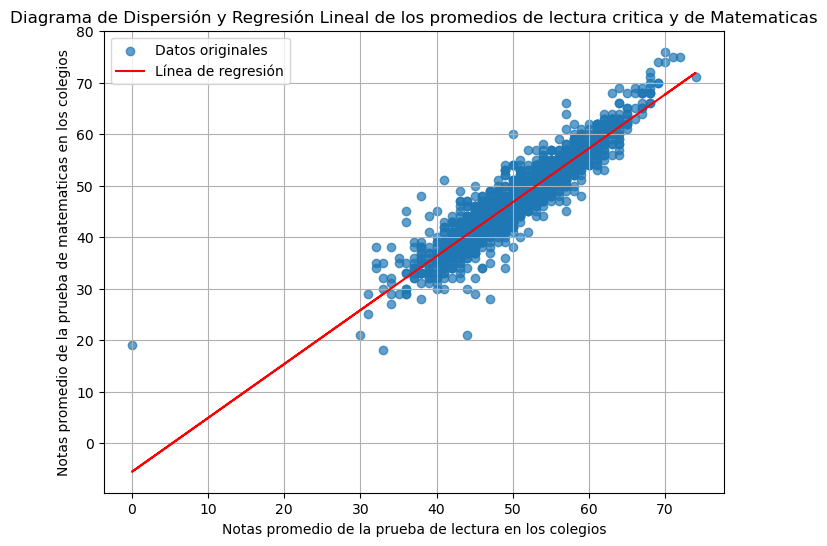
La salida del código anterior es la siguiente:

|  |
| --- |
| Pendiente (B1): 1.046945295837803  Intercepto (B0): -5.578739732057144 |

De modo que el modelo lineal queda de la siguiente manera:

El siguiente código compara los datos con la aproximación lineal previamente obtenida con la gráfica de dispersión:

|  |
| --- |
| # 1. Predecir los valores de y para graficar la línea de regresión  y\_pred = modelo.predict(X)  # 2. Graficar el diagrama de dispersión y la línea de regresión  plt.figure(figsize=(8, 6))  # 2.1. Diagrama de dispersión  plt.scatter(X, y, alpha=0.7, label='Datos originales')  # 2.2. Línea de regresión  plt.plot(X, y\_pred, color='red', label='Línea de regresión')  # 2.3. Añadir etiquetas y título  plt.title('Diagrama de Dispersión y Regresión Lineal de los promedios de lectura critica y de Matematicas')  plt.xlabel('Notas promedio de la prueba de lectura en los colegios')  plt.ylabel('Notas promedio de la prueba de matematicas en los colegios')  plt.legend()  plt.grid(True)  # 2.4. Mostrar el gráfico  plt.show() |



**Análisis del modelo**

Coeficiente de determinación

El código python que realiza la operación anterior se muestra a continuación:

|  |
| --- |
| # Calculo SCT, SCR y SCE  # SCE: Suma de cuadrados explicada  SCE = np.sum((Y\_pred - np.mean(Y)) \*\* 2)  # SCR: Suma de cuadrados residuales  SCR = np.sum((Y - Y\_pred) \*\* 2)  # SCT: Suma total de cuadrados  SCT = np.sum((Y - np.mean(Y)) \*\* 2)  # R2  R2 = 1 - (SCR / SCT)  # 2. Despliegue de las metricas  print(f"SCE: {SCE}")  print(f"SCR : {SCR}")  print(f"SCT: {SCT}")  print(f"R^2: {R2}") |

Para el caso, la salida se muestra a continuación:

|  |
| --- |
| SCE: 97776.22062943577  SCR : 14853.80150670155  SCT: 112630.02213613724  R^2: 0.868118631027635 |

**Coeficiente de correlación**

El valor del coeficiente de correlación se obtiene en el siguiente fragmento de código:

|  |
| --- |
| r = np.sqrt(R2)  print(f"Coeficiente de correlación (r): {r:.5f}") |

Donde la salida es la siguiente:

|  |
| --- |
| Coeficiente de correlación (r): 0.93173 |

**Supuestos del modelo**

Para mirar que tan bueno es el modelo hay diferentes métricas y gráficas. Una de las gráficas de mayor utilidad tiene que ver con la gráfica de **errores de predicción** (también conocidos como **residuos**) . La expresión asociada al error se entiende como la diferencia entre el valor observado () y el valor predicho (). Esto es:

En un análisis de regresión se empieza por hacer un supuesto acerca del modelo apropiado para la relación entre las variables dependientes e independientes. En el caso de la regresión lineal simple, se supone que el modelo de regresión es

Sin embargo, aun cuando se obtenga un valor grande para , la ecuación de regresión estimada no debe ser usada sino hasta que se realice un análisis para determinar si el modelo empleado es apropiado. Un paso importante para ver si el supuesto del modelo es adecuado, consiste en probar la significancia de la relación. Las pruebas de significancia en el análisis de regresión están basadas en los siguientes supuestos acerca del término del error :

* 1. Los valores de son independientes (no están correlacionados).
  2. El error es una variable aleatoria distribuida normalmente.

A continuación, vamos a proceder a la validación de cada uno de los supuestos:

**Aleatoriedad**

Para el caso lo que vamos a probar, se usará el estadístico de Durbin-Watson teniendo en cuenta lo siguiente:

|  |
| --- |
| **Hipótesis**:   * (Los errores son independientes entre sí, es decir no hay autocorrelación). * (Los errores no son independientes entre sí, o sea que estan correlacionados).   Dependiendo de los valores que toma el estadístico de Durbin-Watson en el rango [0,4] se llegan a las siguientes conclusiones:   * Si el valor está cercano a 2, los errores son independientes (no hay autocorrelación). * Si el valor está cercano a 0, hay autocorrelación positiva. * Si el valor está cercano a 4, hay autocorrelación negativa. |

Empleando python tenemos:

|  |
| --- |
| residuos = Y - Y\_pred  # Aplicar el test de Durbin-Watson  dw\_stat = durbin\_watson(residuos)  print(f'Estadístico de Durbin-Watson: {dw\_stat:.5f}') |

Siendo el siguiente el resultado:

|  |
| --- |
| Estadístico de Durbin-Watson: 0.73617 |

El valor del estadístico de Durbin-Watson para los residuos de este modelo es aproximadamente 0.73617 de modo que al ser este valore más cercanos a 0 los errores tienen **autocorrelación positiva**, de modo que estos **no son independientes**.

**Normalidad**

Para probar la normalidad se hará uso de la prueba de Shapiro-Wilk:

|  |
| --- |
| **Hipótesis**:   * (Los residuos **siguen** una distribución normal). * (Los errores **no siguen** una distribución normal).   **Regla de decisión**: Se rechaza si |

Empleando python tenemos:

|  |
| --- |
| # Realizar la prueba de Shapiro-Wilk para los residuos  alpha = 0.05  shapiro\_stat, shapiro\_p\_value = shapiro(residuos)  print(f'Estadístico de Shapiro-Wilk: {shapiro\_stat:.5f}')  print(f'Valor p: {shapiro\_p\_value}')  if shapiro\_p\_value < alpha:      print("Se rechaza H0 de modo que la distribución de los residuos no es normal")  else:      print("Se acepta H0 de modo que la distribución de los residuos es normal") |

Siendo el siguiente el resultado:

|  |
| --- |
| Estadístico de Shapiro-Wilk: 0.94516  Valor p: 2.172619690118276e-25  Se rechaza H0 de modo que la distribución de los residuos no es normal |

Como el valor p es mucho menor que , se rechaza la hipótesis nula de que los residuos se distribuyen normalmente. Esto significa que podemos decir con un nivel de significancia del 5% que los residuales **no siguen una distribución normal**.

**Varianza constante**

Para el caso lo que vamos a probar es lo siguiente:

|  |
| --- |
| **Hipótesis**:   * (La varianza **es constante**). * (La varianza **no** **es constante**).   **Regla de decisión**: Se rechaza si |

Para el caso se usó el estadístico de Breusch-Pagan:

|  |
| --- |
| alpha = 0.05  X\_with\_const = sm.add\_constant(X)  # Realizar el test de Breusch-Pagan  bp\_test = het\_breuschpagan(residuos, X\_with\_const)  # El test devuelve 4 valores: estadístico de Lagrange Multiplier, p-valor, estadístico f y p-valor f  bp\_stat, bp\_p\_value, \_, \_ = bp\_test  # Mostrar los resultados  print(f'Estadístico de Breusch-Pagan: {bp\_stat}')  print(f'Valor p: {bp\_p\_value}')  if bp\_p\_value < alpha:      print("Se rechaza H0 de modo que la varianza los residuos no es constante")  else:      print("Se acepta H0 de modo que la varianza los residuos es constante") |

El resultado del código anterior se muestra a continuación:

|  |
| --- |
| Estadístico de Breusch-Pagan: 44.34894437598986  Valor p: 2.7476012524878094e-11  Se rechaza H0 de modo que la varianza los residuos no es constante |

Como el valor p es muy pequeño (mucho menor que ) se rechaza la hipótesis nula de homocedasticidad; por lo tanto, los residuos **son** **heterostaticos**, es decir que su varianza no es constante a lo largo de los datos.

**Datos atípicos**

Para iniciar, la desviación estándar residual esta dada por:

Luego, se utiliza la siguiente fórmula para estandarizar los residuos:

Donde:

* : es el residuo estandarizado para la observación.
* : es el residuo ordinario para la observación (diferencia entre el valor observado y el valor predicho).
* : es la desviación estándar de los residuos.

Aquellos datos cuya serán candidatos a ser tomados como datos atípicos.

El código python que hace esto se muestra a continuación:

|  |
| --- |
| # Calcular la desviación estándar de los residuos  se = np.std(residuos, ddof=2)  # Estandarizar los residuos  residuos\_estandarizados = residuos / se  # umbral  umbral = 3  datos\_atipicos = promedio\_notas[np.abs(residuos\_estandarizados) > umbral].copy()  datos\_atipicos['d\_i'] =  residuos\_estandarizados[np.abs(residuos\_estandarizados) > umbral]  datos\_atipicos |

La siguiente tabla muestra los datos atípicos obtenidos según el criterio anterior:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 0 | 33 | 18 | -3.824234 |
| 1 | 0 | 19 | 8.567999 |
| 3 | 44 | 21 | -6.792998 |
| 7 | 47 | 28 | -5.447717 |
| 13 | 45 | 29 | -4.369206 |
| 19 | 44 | 30 | -3.655653 |
| 35 | 45 | 32 | -3.323424 |
| 71 | 49 | 34 | -4.086071 |
| 96 | 47 | 35 | -3.007560 |
| 122 | 49 | 36 | -3.388883 |
| 174 | 32 | 38 | 3.512603 |
| 231 | 50 | 38 | -3.056654 |
| 552 | 36 | 43 | 3.795738 |
| 639 | 39 | 44 | 3.049455 |
| 712 | 36 | 45 | 4.492925 |
| 713 | 40 | 45 | 3.033090 |
| 795 | 57 | 45 | -3.171208 |
| 960 | 38 | 48 | 4.808790 |
| 1047 | 43 | 49 | 3.332590 |
| 1226 | 41 | 51 | 4.759695 |
| 1691 | 50 | 60 | 4.612412 |
| 1765 | 57 | 64 | 3.452076 |
| 1777 | 57 | 66 | 4.149264 |

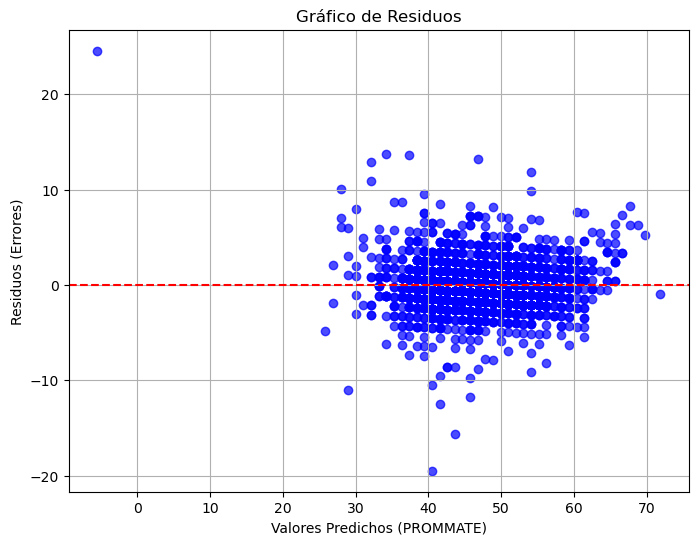
**Gráficos**

**Gráfico de residuos**

El siguiente fragmento de código en python muestra como la gráfica de los residuales:

|  |
| --- |
| # 1. Calcular los residuos (errores)  residuos = Y - Y\_pred  # 2. Graficar los residuos (errores) en función de los valores predichos  plt.figure(figsize=(8, 6))  plt.scatter(Y\_pred, residuos, alpha=0.7, color='blue')  # 2.1. Dibujar una línea horizontal en y=0 para visualizar los errores  plt.axhline(0, color='red', linestyle='--')  # 2.2. Añadir etiquetas y título  plt.title('Gráfico de Residuos')  plt.xlabel('Valores Predichos (PROMMATE)')  plt.ylabel('Residuos (Errores)')  plt.grid(True)  # 2.3. Mostrar el gráfico  plt.show() |

La grafica se muestra a continuación:

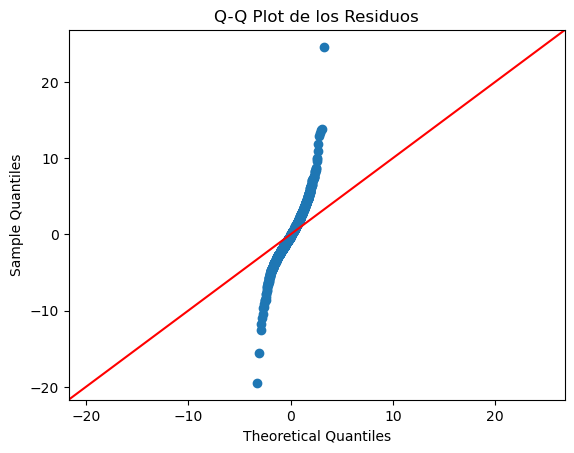


De la gráfica anterior, no se aprecia un patrón uniforme en los residuos por lo que estos no son homeostáticos violando uno de los supuestos de **varianza constante** para los residuos del modelo. Esto se comprobó previamente mediante el test de Breusch-Pagan.

**Q-Q plot**

El siguiente fragmento de código genera el Q-Q plot:

|  |
| --- |
| # Generar el Q-Q plot para visualizar la normalidad de los residuos  sm.qqplot(residuos, line ='45')  plt.title('Q-Q Plot de los Residuos')  plt.show() |



Del Q-Q plot, vemos que los residuos no se alinean uniformemente a lo largo de la diagonal por lo que se confirma los resultados del test se Shapiro-Wilk (realizado previamente) donde se mostro que los residuos no siguen una distribución normal.

|  |
| --- |
| **Conclusiones**:   1. Como se pudo ver con el valor del Coeficiente de Determinación (R2), al ser este de un 86.8 %, se puede afirmar que existe una relación alta positiva y fuerte (a mayor nota promedio en las pruebas de lectura, mayor nota promedio en las pruebas de matemáticas) entre la prueba de lectura con la de matemáticas lo que indica que la nota promedio de lectura es un bien predictor para la nota promedio de matemáticas. 2. Algunos supuestos claves para el modelo no se cumplen. Según los test anteriormente realizados los resultados arrojados fueron los siguientes:    * **No hay normalidad** en los errores.    * Los errores **no tienen varianza constante**.    * Los errores **están correlacionados** de manera positiva. 3. La violación de los supuestos puede llevar a inferencias poco fiables y predicciones no muy precisas. Para mejorar esto se podrían aplicar transformaciones sobre el modelo y eliminar los datos atípicos. 4. Existen varios datos atípicos que deben ser revisados teniendo en cuenta la opinión de un expertos para decidir qué hacer con estos pues estos están influyendo en el modelo de tal modo que supuestos claves no se cumplan. |

**Notebook**

El notebook se encuentra en el siguiente enlace: <https://github.com/repos-especializacion-UdeA/estadistica/blob/main/trabajo2/trabajo2.ipynb>