**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 piden, es necesario cargar la base de datos usando alguna herramienta como Python o R. En nuestro caso, se usó la primera de estas.

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 empezar a realizar los cálculos, es bueno conocer si hay valores nulos en cada columna.

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

Como ya se tienen los datos cargados, ahora si es posible empezar 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 de 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() |

Ahora si podemos empezar a plantear el problema teniendo en cuenta 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 critica por el *i-esimo* colegio.

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

El primer paso consiste en obtener la media y la desviación 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 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) o en otras palabras el valor de 45 puntos no está dentro del intervalo de confianza.  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. 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 a 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? De este modo, 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 a 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 significación 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 significación 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**

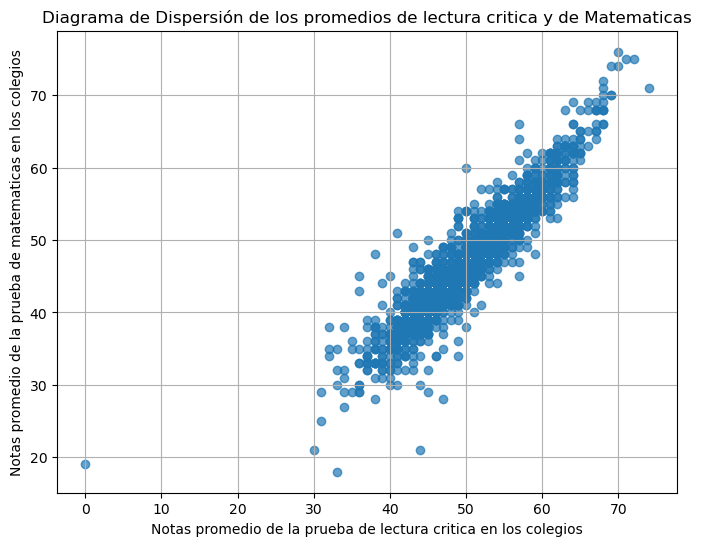
Inicialmente se obtienen los datos y se hace un diagrama de dispersión para visualizar gráficamente la relación entre estos. Para este caso se definieron estos de la siguiente manera:

* **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  # 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 proceder 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 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}') |

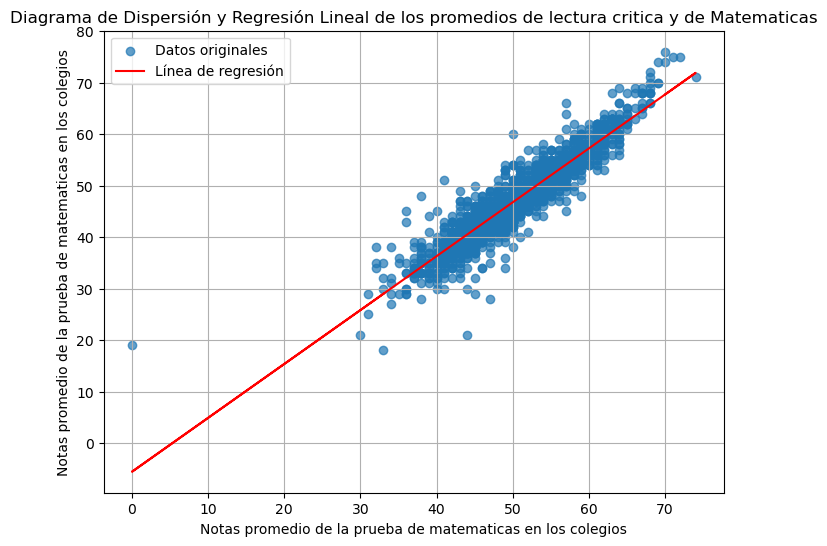
La salida del código anterior es la siguiente:

|  |
| --- |
| Pendiente (B1): 1.046945295837803  Intersepto (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:

|  |
| --- |
| # 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 matematicas 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() |



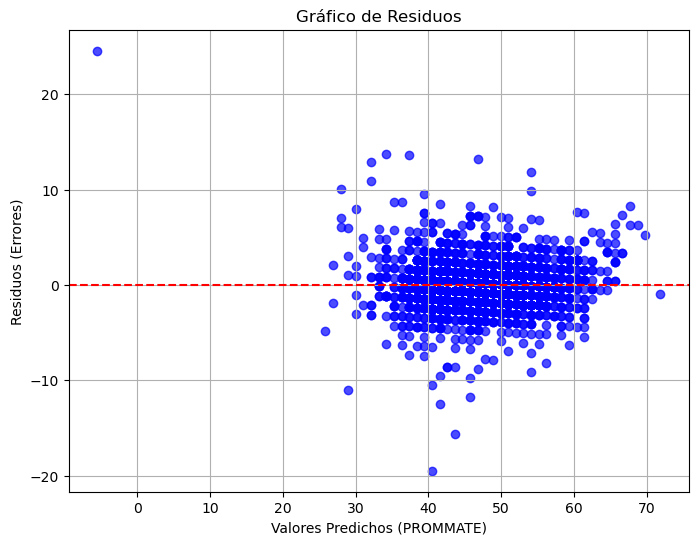
**Análisis del modelo**

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 grafica de **errores de predicción** (también conocidos como **residuos**) . La expresión asociada al error consiste entre la diferencia entre el valor observado () y el valor predicho (). Esto es:

En el siguiente código se realiza la gráfica en cuestión:

|  |
| --- |
| # 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:



Otras métricas que evalúan el rendimiento del modelo son:

* **Coeficiente de determinación ()**: Indica la proporción de la varianza en la variable dependiente que es explicada por las variables independientes.
* **Error cuadrático medio (MSE)**: Mide la media de los errores al cuadrado.
* **Raíz del error cuadrático medio (RMSE)**: Es la raíz cuadrada del MSE y tiene la misma unidad que la variable dependiente.
* **Error absoluto medio (MAE)**: Mide la media de los errores absolutos, es decir, sin tener en cuenta el signo.

El siguiente fragmento de código muestra cómo realizar estos cálculos:

|  |
| --- |
| # 1. Calculo de las métricas  r2 = r2\_score(y, y\_pred)  # Coeficiente de determinación  mse = mean\_squared\_error(y, y\_pred)  # Error cuadrático medio  rmse = np.sqrt(mse)  # Raíz del error cuadrático medio  mae = mean\_absolute\_error(y, y\_pred)  # Error absoluto medio  # 2. Despliegue de las metricas  print(f"R^2 (Coeficiente de determinación): {r2}")  print(f"Error cuadrático medio (MSE): {mse}")  print(f"Raíz del error cuadrático medio (RMSE): {rmse}")  print(f"Error absoluto medio (MAE): {mae}") |

La salida del fragmento anterior se muestra a continuación:

|  |
| --- |
| R^2 (Coeficiente de determinación): 0.868118631027635  Error cuadrático medio (MSE): 8.220144718705894  Raíz del error cuadrático medio (RMSE): 2.8670794754777718  Error absoluto medio (MAE): 2.0692130047500217 |

|  |
| --- |
| **Conclusión**:  Colocar la conclusión… |

**Notebook**

El notebook se encuentra en el siguiente link