INSTITUTO TECNOLOGICO DE LAS AMERICAS (ITLA)

ESTUDIANTE:
JUNIER SOTO GUERRA

MATRICULA:
2016-4396

TEMA:
ANÁLISIS DESCRIPTIVO DEL RENDIMIENTO ESTUDIANTIL EN STUDENTSPERFORMANCE

ASIGNATURA:
PROGRAMACION III

DOCENTE:
JOSE AQUINO

FECHA:
28 DE JULIO 2025

Introducción

El presente trabajo tiene como objetivo realizar un análisis descriptivo del conjunto de datos **StudentsPerformance**, el cual contiene información relevante sobre el rendimiento académico de estudiantes en tres áreas fundamentales: matemáticas, lectura y escritura. Además, incluye variables socio-demográficas que permiten contextualizar el desempeño escolar.

Este análisis se enmarca en la asignatura de Programación III, integrando técnicas de bases de datos, procesamiento de datos y análisis estadístico con herramientas de software ampliamente utilizadas como Git, MySQL, Python y R. A través de este estudio, se busca no solo aplicar los conocimientos técnicos adquiridos durante el curso, sino también obtener información significativa que contribuya a comprender mejor los factores que influyen en el rendimiento académico. Repositorio: https://github.com/Juniersg/Trab_Prog_3/blob/main/Trab_Prog_III.R

Justificación

Realizar un análisis descriptivo de los datos es un paso indispensable en cualquier proyecto de Ciencia de Datos, pues permite familiarizarse con la estructura, distribución y características fundamentales del conjunto de datos. Además, es útil para detectar inconsistencias, valores atípicos o faltantes que pueden distorsionar análisis posteriores.

En el contexto educativo, comprender cómo variables socioeconómicas, como el tipo de almuerzo que recibe un estudiante, o factores académicos, como la realización de un curso preparatorio, influyen en el rendimiento es vital para orientar políticas y estrategias educativas. La identificación de estas relaciones mediante métodos estadísticos contribuye a mejorar la equidad y eficacia del sistema educativo.

La utilización de herramientas de programación y bases de datos permite automatizar, replicar y validar el análisis, fortaleciendo su confiabilidad y facilitando su actualización con nuevos datos.

Dataset y Base de Datos

El dataset StudentsPerformance fue descargado desde Kaggle, reconocido repositorio de datasets abiertos, garantizando la disponibilidad y transparencia de los datos para la comunidad académica. Este conjunto contiene registros individuales de estudiantes, incluyendo variables categóricas como género, raza/etnia, tipo de almuerzo y curso de preparación, junto con las puntuaciones en tres áreas académicas fundamentales: matemáticas, lectura y escritura.

Para el manejo eficiente y estructurado de estos datos, se creó una base de datos relacional en MySQL, una de las tecnologías más populares y robustas para gestión de bases de datos. Los datos se almacenaron en la base Trabajo_Prog_III, en la tabla StudentsPerformance, que mantiene la integridad y consistencia de los registros, permitiendo consultas rápidas y seguras.

Metodología

Herramientas utilizadas

- **MySQL:** Se utilizó para almacenar y gestionar el dataset. Su lenguaje SQL permite extraer datos específicos mediante consultas, optimizando el rendimiento y permitiendo trabajar con grandes volúmenes de información.
- **Python (Jupyter Notebook):** Se empleó para la conexión a MySQL mediante la librería mysql.connector, procesamiento y análisis preliminar con pandas, y visualización gráfica con seaborn y matplotlib. Este entorno interactivo permite probar y ajustar rápidamente el análisis y visualizaciones.
- **R (RStudio):** Complementó el análisis con paquetes como DBI y RMySQL para conexión, dplyr para manipulación avanzada de datos y ggplot2 para la generación de gráficos profesionales y altamente personalizables. R destaca por su potencia en análisis estadístico.
- **Git y GitHub:** Se utilizó para controlar versiones del proyecto, facilitando la colaboración, organización y respaldo del código y documentación. La inclusión de .gitignore para proteger datos sensibles fue una buena práctica implementada.

Manejo seguro de credenciales

Para proteger la información sensible de acceso a la base de datos, se implementó un archivo .env en ambos entornos (Python y R) para almacenar variables de entorno con las credenciales necesarias. Esto previene la exposición accidental de datos confidenciales en el código fuente y facilita la configuración en diferentes máquinas o entornos.

Desarrollo y Resultados

Análisis con Python

Se estableció la conexión con MySQL usando la librería mysql.connector, extrayendo los datos completos para su análisis en un DataFrame de pandas. Se calcularon estadísticas descriptivas básicas y se realizó un análisis específico para determinar el promedio del puntaje en matemáticas agrupado por tipo de almuerzo.

importación de las librerías:

```python
# Importo las Librerías necesarias
import os
from dotenv import load_dotenv
import pandas as pd
import mysql.connector
from sqlalchemy import create_engine
import matplotlib.pyplot as plt
import seaborn as sns
```

✓ 0.0s

Carga de las variables de entorno y conexión a MySql

```python
# Cargar variables del archivo .env
load_dotenv()

host = os.getenv("MYSQL_HOST")
user = os.getenv("MYSQL_USER")
password = os.getenv("MYSQL_PASSWORD")
database = os.getenv("MYSQL_DB")


# Conexión a MySQL y creación de la base de datos
conn = mysql.connector.connect(
    host=host,
    user=user,
    password=password
)

cursor = conn.cursor()
cursor.execute("CREATE DATABASE IF NOT EXISTS Trabajo_Prog_III")
cursor.close()
conn.close()
```

Creación del DataFrame con Pandas e inserción de los datos en la Base de Datos Mysql

```python
# Subo el CSV a MySQL
df = pd.read_csv("StudentsPerformance.csv")

# Renombro columnas para que coincidan con estándares SQL
df.columns = [col.strip().lower().replace(" ", "_") for col in df.columns]

# Creo motor SQLAlchemy para insertar el dataframe a MySQL
engine = create_engine("mysql+mysqlconnector://root:Monitorhp1423.@localhost/Trabajo_Prog_III")

# Insertar datos en la tabla 'estudiantes'
df.to_sql("estudiantes", con=engine, if_exists="replace", index=False)
```

✓ 0.2s

Consulta a la Base de Datos y carga a un df de pandas, mostrando los primeros registros.

```python
# Leo los datos desde MySQL a Pandas
query = "SELECT * FROM estudiantes"
df = pd.read_sql(query, con=engine)
df.head()
```

✓ 0.0s

	gender	race/ethnicity	parental_level_of_education	lunch	test_preparation_course	math_score	reading_score	writing_score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

Visutalizacion de estatisticas descriptivas del DF

```python
# Veo Estadísticas básicas del dataframe
df.describe()
```

✓ 0.0s

	math_score	reading_score	writing_score
count	1000.00000	1000.000000	1000.000000
mean	66.08900	69.169000	68.054000
std	15.16308	14.600192	15.195657
min	0.00000	17.000000	10.000000
25%	57.00000	59.000000	57.750000
50%	66.00000	70.000000	69.000000
75%	77.00000	79.000000	79.000000
max	100.00000	100.000000	100.000000

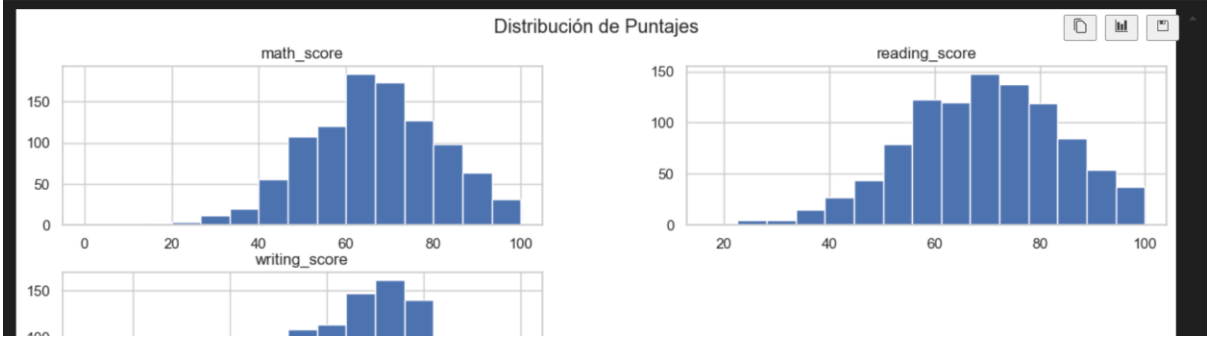
Los resultados mostraron que los estudiantes que reciben almuerzos estándar presentan puntajes promedio más altos en matemáticas que aquellos que reciben almuerzos gratuitos o reducidos, lo que podría estar reflejando factores socioeconómicos asociados.

La visualización mediante un gráfico de barras con seaborn permitió observar claramente estas diferencias, facilitando la comunicación de los resultados.

Visualizaciones:

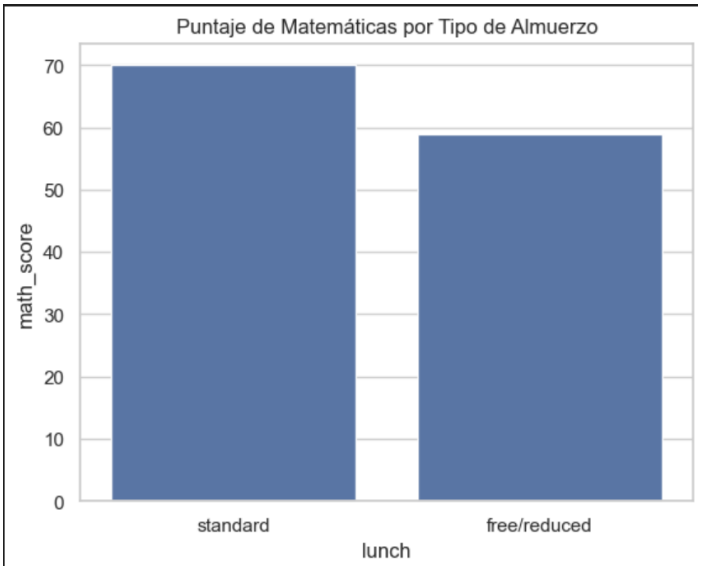
```python
# Visualizaciones
sns.set(style="whitegrid")

# Histogramas
df[["math_score", "reading_score", "writing_score"]].hist(bins=15, figsize=(15, 5))
plt.suptitle("Distribución de Puntajes")
plt.show()
```

✓ 0.4s Python



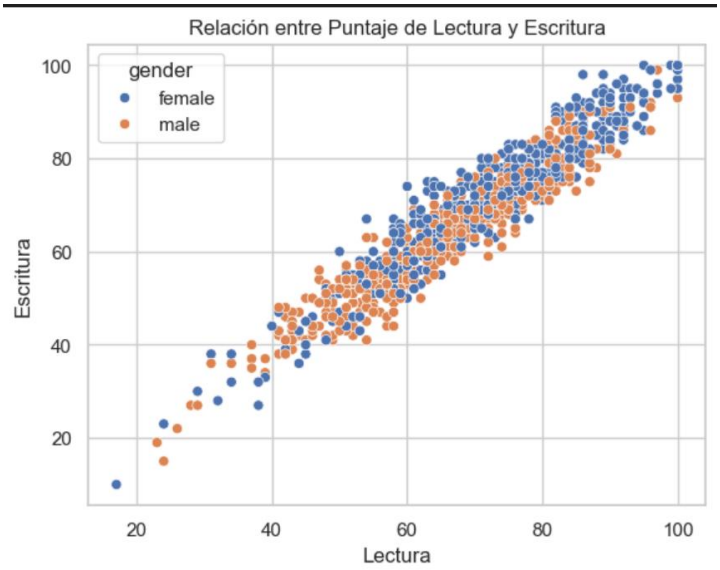

```python
# Promedio por tipo de almuerzo
sns.barplot(data=df, x="lunch", y="math_score", errorbar=None)
plt.title("Puntaje de Matemáticas por Tipo de Almuerzo")
plt.show()
```

✓ 0.0s



```python
# Correlación entre lectura y escritura
sns.scatterplot(data=df, x="reading_score", y="writing_score", hue="gender")
plt.title("Relación entre Puntaje de Lectura y Escritura")
plt.xlabel("Lectura")
plt.ylabel("Escritura")
plt.show()
```

✓ 0.1s



Análisis con R

En R, se replicó el proceso de conexión y consulta a la base de datos, utilizando DBI y RMySQL. Con dplyr se procesaron los datos para calcular promedios y generar agrupamientos similares a los realizados en Python.

Adicionalmente, se realizaron gráficos de caja para explorar la distribución y dispersión de los puntajes de escritura en función del curso de preparación que realizaron los estudiantes. Estos boxplots mostraron que los estudiantes que completaron el curso preparatorio tienden a tener puntajes más altos y menos dispersión negativa, lo que indica un efecto positivo de esta preparación.

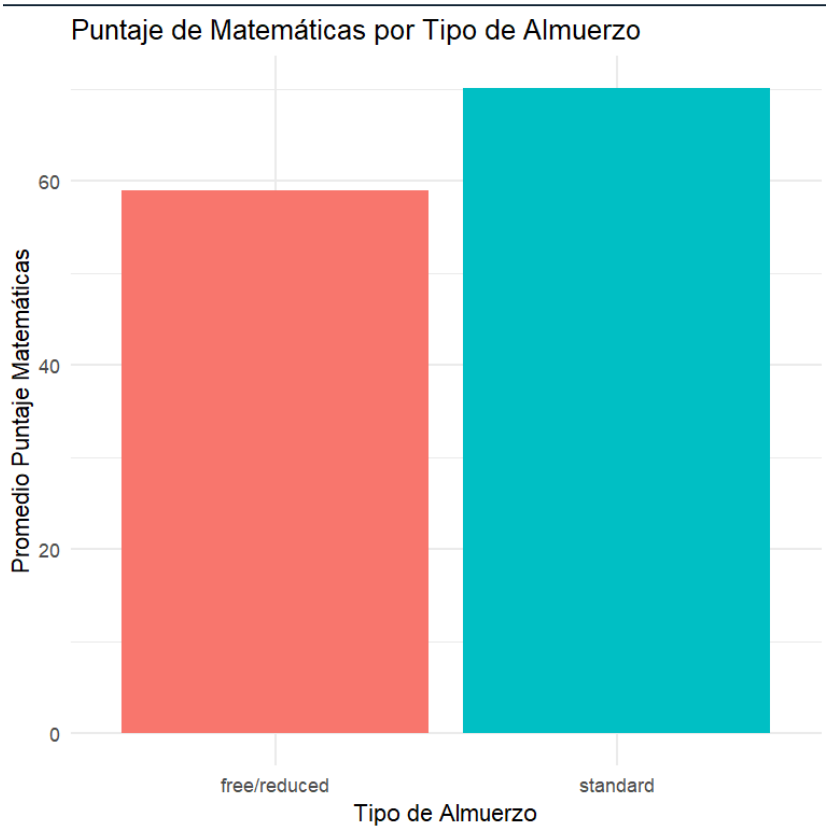
Los gráficos elaborados con ggplot2 se destacaron por su calidad visual y claridad, contribuyendo a un análisis más profundo.

Carga de Variables de entorno y conexión a MySql:

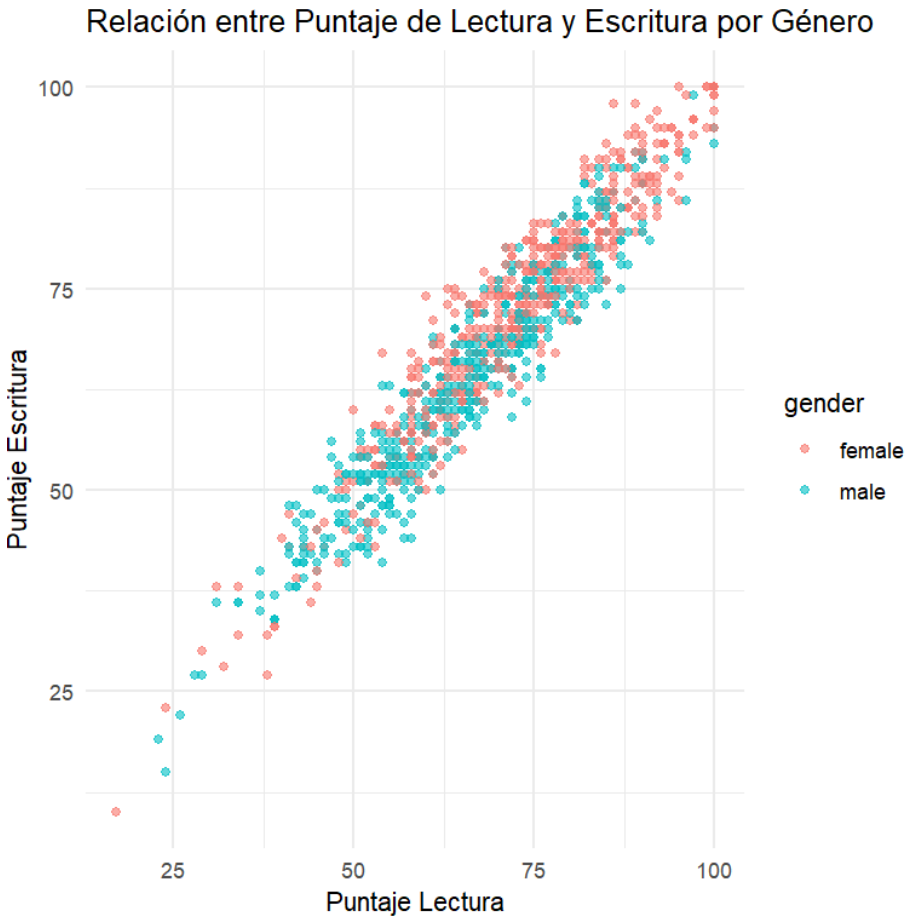
```r
# Cargar variables de entorno desde archivo .env
load_dot_env(file = ".env")

# Leer variables
host <- Sys.getenv("MYSQL_HOST")
user <- Sys.getenv("MYSQL_USER")
password <- Sys.getenv("MYSQL_PASSWORD")
dbname <- Sys.getenv("MYSQL_DB")

# Conexión a la base de datos MySQL
con <- dbConnect(
  RMySQL::MySQL(),
  host = host,
  user = user,
  password = password,
  dbname = dbname
)
```

Visualizaciones:

```r
ggplot(promedio_alimentos, aes(x = lunch, y = promedio_math, fill = lunch)) +
  geom_col(show.legend = FALSE) +
  labs(title = "Puntaje de Matemáticas por Tipo de Almuerzo",
       x = "Tipo de Almuerzo",
       y = "Promedio Puntaje Matemáticas") +
  theme_minimal()
```




```r
#  Scatterplot lectura vs escritura por género ---
ggplot(df, aes(x = reading_score, y = writing_score, color = gender)) +
  geom_point(alpha = 0.6) +
  labs(title = "Relación entre Puntaje de Lectura y Escritura por Género",
       x = "Puntaje Lectura",
       y = "Puntaje Escritura") +
  theme_minimal()
```



Control de versiones y colaboración con Git

Para garantizar la organización, trazabilidad y colaboración efectiva en el proyecto, se utilizó **Git** como sistema de control de versiones y **GitHub** como repositorio remoto.

Esto permitió registrar cada modificación en el código y documentos, facilitando la revisión y retroalimentación, además de mantener respaldos automáticos del trabajo realizado.

Se implementaron buenas prácticas como:

- Exclusión del archivo .env del repositorio mediante .gitignore para evitar exponer credenciales sensibles.
- Documentación clara en los commits para identificar fácilmente los cambios realizados.
- Sincronización constante con el repositorio remoto para mantener actualizado el proyecto.

Conclusiones

El proyecto integró exitosamente bases de datos, programación en Python y R, y análisis estadístico para realizar un estudio descriptivo sólido del dataset estudiantil.

Se evidenció la utilidad de combinar distintas herramientas para optimizar el flujo de trabajo y la presentación de resultados.

La metodología empleada garantiza la seguridad y reproducibilidad del análisis, siendo una base sólida para investigaciones futuras que puedan incluir análisis predictivos o prescriptivos.

Finalmente, los hallazgos reflejan la relevancia de variables socioeconómicas y académicas en el rendimiento estudiantil, aportando información valiosa para la comunidad educativa.