TALENTO





Universidad Tecnológica de Bolívar

www.utb.edu.co/falento-

www.utb.edu.co/talento-tecl





Data Wrangling end EDA with PANDAS

Ejecutor técnico: Jorge Luis Villalba Acevedo

Actividad: Exploración y Limpieza de Datos de Exportaciones Agrícolas enColombia

Estimado CAMPISTAS,

Integrantes de grupo:

- Emmy Luz Oyola Díaz
- Carlos José Romero Escorcia
- Juan Carlos Gutiérrez Ortega
- Oscar Ricardo Sierra Alean

Contexto: Desafíos en la Educación en Colombia

En Colombia, la educación enfrenta retos significativos, como la falta de acceso y la deserción escolar. A menudo, los estudiantes no logran alcanzar un rendimiento académico óptimo debido a factores como la asistencia irregular y la desigualdad en las oportunidades educativas. Este conjunto de datos simulado representa a estudiantes colombianos, incluyendo información sobre su rendimiento, asistencia ycaracterísticas demográficas. A través de este análisis, se busca resaltar cómo estas variables pueden afectar el desempeño educativo.

Descripción del Conjunto de Datos

- **student_id**: Identificación única para cada estudiante (algunos registrospueden estar duplicados).
- age: Edad del estudiante, variando entre 6 y 17 años.
- score: Calificación promedio en exámenes, con algunas calificacionesfaltantes.
- **attendance**: Porcentaje de asistencia a clases, también con valores faltantes.
- **gender**: Género del estudiante, "Masculino" o "Femenino".
- **grade**: Grado escolar, ya sea "Primaria" o "Secundaria".
- **region**: Región de Colombia a la que pertenece el estudiante, incluyendo "Caribe", "Pacífico", "Andino" y "Amazonía".



Preguntas sobre Manejo de Datos Faltantes

1. ¿Cuántos valores faltantes hay en todo el DataFrame?

R/ En la base de datos se encuentran 22 datos faltantes, 11 para la columna score y 11 para la columna attendance.

```
1 df.isna().sum().sum()
```

2. ¿Qué porcentaje de valores faltantes hay en la columna 'score'?

R/ La variable Score tiene un 10% de valores faltantes, esto significa que el 10% de los estudiantes no cuentan con una calificación promedio de exámenes en el DataFrame

```
1 tot = df.shape[0]
2 NScore = df['score'].isna().sum()
3 (NScore/tot)*100
```

3. Rellena los valores faltantes de 'score' con la media.

| | student_id | age | score | attendance | gender | grade | region |
|---|------------|-----|-------|------------|-----------|------------|----------|
| 0 | 1 | 6 | 75.9 | 93.050075 | Masculino | Primaria | Andino |
| 1 | 2 | 10 | 71.3 | 29.684641 | Masculino | Secundaria | Amazonía |
| 2 | 3 | 6 | NaN | 65.321825 | Femenino | Primaria | Amazonía |
| 3 | 4 | 14 | 72.7 | 90.107048 | Masculino | Secundaria | Caribe |
| 4 | 5 | 15 | 77.3 | NaN | Femenino | Secundaria | Andino |

La variable Score tiene 11 datos faltantes y se reemplazaron con la media de ésta: 73.05





| | student_id | age | score | attendance | gender | grade | region |
|---|------------|-----|-----------|------------|-----------|------------|----------|
| 9 | 1 | 6 | 75.900000 | 93.050075 | Masculino | Primaria | Andino |
| 1 | 2 | 10 | 71.300000 | 29.684641 | Masculino | Secundaria | Amazonía |
| 2 | 3 | 6 | 73.050505 | 65.321825 | Femenino | Primaria | Amazonía |
| 3 | 4 | 14 | 72.700000 | 90.107048 | Masculino | Secundaria | Caribe |
| 4 | 5 | 15 | 77.300000 | NaN | Femenino | Secundaria | Andino |

```
1 df['score'] = df['score'].fillna(df['score'].mean())
2 df
```



4. Elimina las filas con valores faltantes en 'attendance'.

| | student_id | age | score | attendance | gender | grade | region |
|---|------------|-----|-------|------------|-----------|------------|----------|
| 0 | 1 | 6 | 75.9 | 93.050075 | Masculino | Primaria | Andino |
| 1 | 2 | 10 | 71.3 | 29.684641 | Masculino | Secundaria | Amazonía |
| 2 | 3 | 6 | NaN | 65.321825 | Femenino | Primaria | Amazonía |
| 3 | 4 | 14 | 72.7 | 90.107048 | Masculino | Secundaria | Caribe |
| 4 | 5 | 15 | 77.3 | NaN | Femenino | Secundaria | Andino |

La variable Attendance tiene 11 datos faltantes, los cuales fueron eliminados

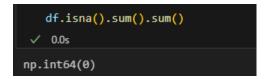
| | student_id | age | score | attendance | gender | grade | region |
|---|------------|-----|-------|------------|-----------|------------|----------|
| 0 | 1 | 6 | 75.9 | 93.050075 | Masculino | Primaria | Andino |
| 1 | 2 | 10 | 71.3 | 29.684641 | Masculino | Secundaria | Amazonía |
| 2 | 3 | 6 | NaN | 65.321825 | Femenino | Primaria | Amazonía |
| 3 | 4 | 14 | 72.7 | 90.107048 | Masculino | Secundaria | Caribe |
| 5 | 6 | 9 | 72.3 | 43.033322 | Femenino | Primaria | Andino |





5. ¿Cuántos valores faltantes hay en cada columna?

R/ No hay valores faltantes dado que estos fueron reemplazados para la columna score y eliminados para la columna attendance.

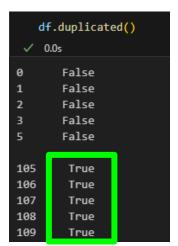


Preguntas sobre Registros Duplicados

6. ¿Cuántos registros duplicados hay en el DataFrame?

R/ El DataFrame tiene 10 registros duplicados.

7. Elimina los registros duplicados.





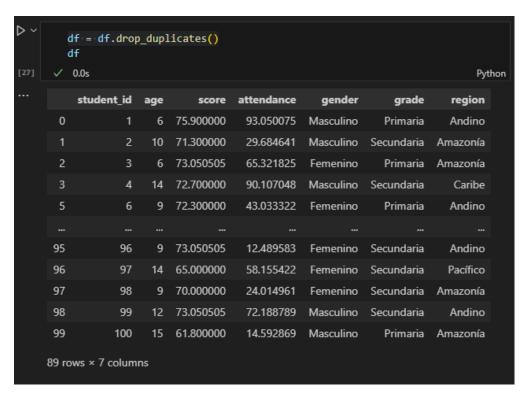


De los 10 registros duplicados en el DataFrame en esta imagen se observan 5 (105 - 109) de los 10.

Los registros fueron eliminados con el código:

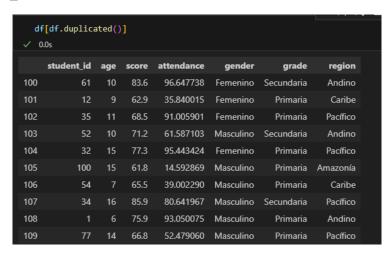
df = df.drop duplicates()

En la siguiente imagen se puede observar que los registros (105 - 109) fueron eliminados.



8. Muestra los registros duplicados.

En la siguiente imagen se muestran los registros duplicados teniendo como referencia la variable students_id.







9. ¿Cuántos estudiantes únicos hay en el DataFrame después de eliminar duplicados?

R/ Tras eliminar los registros duplicados podemos observar que quedan 89 registros en el DataFrame

Preguntas sobre Transformación de Datos

10. Transforma la columna 'attendance' a un rango de 0 a 1.

| | student_id | age | score | attendance | gender | grade | region |
|---|------------|-----|-----------|------------|-----------|------------|----------|
| 0 | 1 | 6 | 75.900000 | 93.050075 | Masculino | Primaria | Andino |
| 1 | 2 | 10 | 71.300000 | 29.684641 | Masculino | Secundaria | Amazonía |
| 2 | 3 | 6 | 73.050505 | 65.321825 | Femenino | Primaria | Amazonía |
| 3 | 4 | 14 | 72.700000 | 90.107048 | Masculino | Secundaria | Caribe |
| 5 | 6 | 9 | 72.300000 | 43.033322 | Femenino | Primaria | Andino |

Tras normalizar la variable attendance con el código:

```
df['attendance'] = df['attendance'].apply(lambda x: x/100)
df
```

Se puede observar el DataFrame actualizado

| | Se puede observar er batarrame actualizado | | | | | | | | | |
|---|--|-----|-----------|------------|-----------|------------|----------|--|--|--|
| | student_id | age | score | attendance | gender | grade | region | | | |
| 0 | 1 | 6 | 75.900000 | 0.930501 | Masculino | Primaria | Andino | | | |
| 1 | 2 | 10 | 71.300000 | 0.296846 | Masculino | Secundaria | Amazonía | | | |
| 2 | 3 | 6 | 73.050505 | 0.653218 | Femenino | Primaria | Amazonía | | | |
| 3 | 4 | 14 | 72.700000 | 0.901070 | Masculino | Secundaria | Caribe | | | |
| 5 | 6 | 9 | 72.300000 | 0.430333 | Femenino | Primaria | Andino | | | |
| | | | | | | | | | | |





11. Crea una nueva columna que indique si el estudiante aprobó (score >= 60).

Para crear la nueva columna primero se definió una función que clasificara los estudiantes aprobados y no aprobados. Luego se aplicó esta función en la columna score para crear la nueva variable.

```
# Función para clasificar los estudiantes aprobados

def classify_approved(score):
    if score >60:
        return 'Sí'
    else:
        return 'No'

# Aplicar la función a la columna 'score' para crear la nueva columna
df['approved'] = df['score'].apply(classify_approved)
```

| | student_id | age | score | attendance | gender | grade | region | aprobado |
|---|------------|-----|-----------|------------|-----------|------------|----------|----------|
| 0 | 1 | 6 | 75.900000 | 0.930501 | Masculino | Primaria | Andino | si |
| 1 | 2 | 10 | 71.300000 | 0.296846 | Masculino | Secundaria | Amazonía | si |
| 2 | 3 | 6 | 73.050505 | 0.653218 | Femenino | Primaria | Amazonía | si |
| 3 | 4 | 14 | 72.700000 | 0.901070 | Masculino | Secundaria | Caribe | si |
| 5 | 6 | 9 | 72.300000 | 0.430333 | Femenino | Primaria | Andino | si |
| 3 | 4 | 14 | 72.700000 | 0.901070 | Masculino | Secundaria | Caribe | |

12. Crea un DataFrame que contenga solo estudiantes de secundaria.

La siguiente imagen muestra a los estudiantes de secundaria.



| | student_id | age | score | attendance | gender | grade | region | aprobado |
|----|------------|-----|-----------|------------|-----------|------------|----------|----------|
| 1 | 2 | 10 | 71.300000 | 0.296846 | Masculino | Secundaria | Amazonía | si |
| 3 | 4 | 14 | 72.700000 | 0.901070 | Masculino | Secundaria | Caribe | si |
| 8 | 9 | 6 | 77.600000 | 0.279807 | Femenino | Secundaria | Pacífico | si |
| 12 | 13 | 14 | 73.050505 | 0.842007 | Masculino | Secundaria | Caribe | si |
| 16 | 17 | 7 | 91.000000 | 0.002378 | Masculino | Secundaria | Amazonía | si |





13. Convierte la columna 'grade' a un tipo categórico.

```
# Convertir la columna 'grade' a categórico
df['grade'] = df['grade'].astype('category')
df['grade'].dtypes
```



14. ¿Cuál es la calificación promedio por grado? Usando el código:

```
promedio_grado = df.groupby('grade')['score'].mean()
```

Se encontró que el promedio para cada grado fue:







www.utb.edu.co/talento-tech