Trabajo Práctico Integrador

Big Data - Codo a Codo 4.0

Constanza Vazquez

Análisis exploratorio

```
# IMPORTAR LIBRERIAS
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```
# CARGAR DATOS EN EL DATAFRAME
df = pd.read_csv('/work/exams.csv')
df
```

	id object	gender object	race/ethnicity obj...	parental level of ...	lunch object	employed object	test preparation ...	math score float6
	53-9893429 0.2% 10-1068446 0.2% 998 others 99.6%	male 51.9% female 48.1%	group C 32% group D 26.3% 3 others 41.7%	some college 22.5% associate's ... 20.4% 4 others 57.1%	standard 65.3% free/reduced 34.7%	yes 51.3% no 48.7%	none 66.4% completed 33.6%	13.0 - 100.0
0	10-5894942	male	group A	high school	standard	yes	completed	6
1	41-1676468	female	group D	some high school	free/reduced	no	none	4
2	64-6396924	male	group E	some college	free/reduced	no	none	5
3	35-2426788	male	group B	high school	standard	yes	none	7
4	60-9387304	male	group E	associate's degree	standard	yes	completed	7
5	67-3666190	female	group D	high school	standard	yes	none	6
6	27-7702214	female	group A	bachelor's degree	standard	yes	none	6
7	46-2257650	male	group E	some college	standard	yes	completed	9
8	40-1499649	male	group D	high school	standard	no	none	6

9	67-7378468	male	group C	some college	free/reduced	no	none	4

```python
# Las primeras 5 filas
df.head()
```

	id object	gender object	race/ethnicity obj...	parental level of ...	lunch object	employed object	test preparation ...	math score float64
0	10-5894942	male	group A	high school	standard	yes	completed	67.0
1	41-1676468	female	group D	some high school	free/reduced	no	none	40.0
2	64-6396924	male	group E	some college	free/reduced	no	none	59.0
3	35-2426788	male	group B	high school	standard	yes	none	77.0
4	60-9387304	male	group E	associate's degree	standard	yes	completed	78.0

```python
# Las últimas 5 filas
df.tail()
```

	id object	gender object	race/ethnicity obj...	parental level of ...	lunch object	employed object	test preparation ...	math score float64
1013	82-7312119	male	group E	associate's degree	standard	yes	none	74.0
1014	45-3445439	male	group E	some college	free/reduced	no	none	78.0
1015	02-3651562	male	group A	some college	standard	no	completed	78.0
1016	05-5203587	female	group B	some college	standard	yes	none	75.0
1017	13-3347050	male	group D	some college	standard	no	completed	70.0

```python
# Resumen estadístico
df.describe()
```

	math score float64	physics score flo...	chemistry score f...	algebra_score flo...
count	1011.0	1011.0	1011.0	1011.0
mean	66.48071216617211	69.06330365974283	67.7893175074184	67.77843719090009
std	15.326879704379337	14.694107007851635	15.55985328614052	14.450679861041094
min	13.0	27.0	23.0	22.0
25%	56.0	60.0	58.0	59.0
50%	67.0	70.0	68.0	68.0
75%	77.0	79.0	79.0	78.0
max	100.0	100.0	100.0	100.0

```python
# REVISAR TIPOS DE DATOS
df.dtypes
```

```
id                              object
gender                          object
race/ethnicity                  object
parental level of education     object
lunch                           object
employed                        object
test preparation course         object
math score                     float64
physics score                  float64
chemistry score                float64
algebra_score                  float64
dtype: object
```

```python
# ELIMINAR DUPLICADOS
print(f'Original: {df.id.count()} filas')
duplicate_rows_df = df[df.duplicated()]
print(f'Cantidad de filas duplicadas: {duplicate_rows_df.id.count()}')

df = df.drop_duplicates()
```

```
Original: 1018 filas
Cantidad de filas duplicadas: 18
```

```python
# Filas despues de eliminar los duplicados
print(f'Final: {df.id.count()} filas')
```

```
Final: 1000 filas
```

```python
# ELIMINAR COLUMNAS IRRELEVANTES
print(df.columns)
#df = df.drop(['id'], axis=1)
```

```
Index(['id', 'gender', 'race/ethnicity', 'parental level of education',
       'lunch', 'employed', 'test preparation course', 'math score',
       'physics score', 'chemistry score', 'algebra_score'],
      dtype='object')
```

```python
# RENOMBRAR LAS COLUMNAS
df = df.rename(columns= {
    "gender":"Gender",
    "race/ethnicity":"Ethnicity",
    "parental level of education": "Parental level of education",
    "lunch":"Lunch",
    "employed":"Employed",
    "test preparation course":"Test preparation course",
    "math score":"Math score",
    "physics score":"Physics score",
    "chemistry score":"Chemistry score",
    "algebra_score":"Algebra score"
})
df.columns
```

```
Index(['id', 'Gender', 'Ethnicity', 'Parental level of education', 'Lunch',
       'Employed', 'Test preparation course', 'Math score', 'Physics score',
       'Chemistry score', 'Algebra score'],
      dtype='object')
```

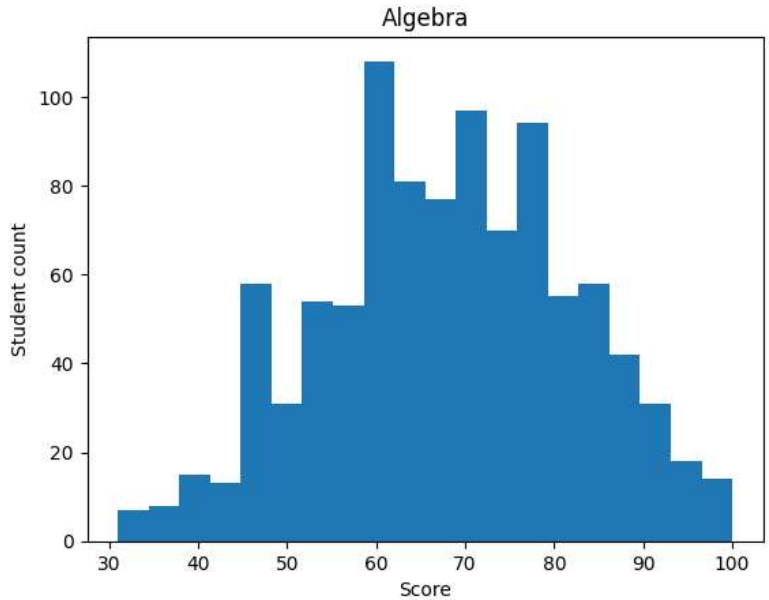
```python
# ELIMINAR VALORES PERDIDOS O NULOS

# Encontrar los valores nulos
print(df.isnull().sum())

# Eliminar los valores nulos
df = df.dropna()
print()
```

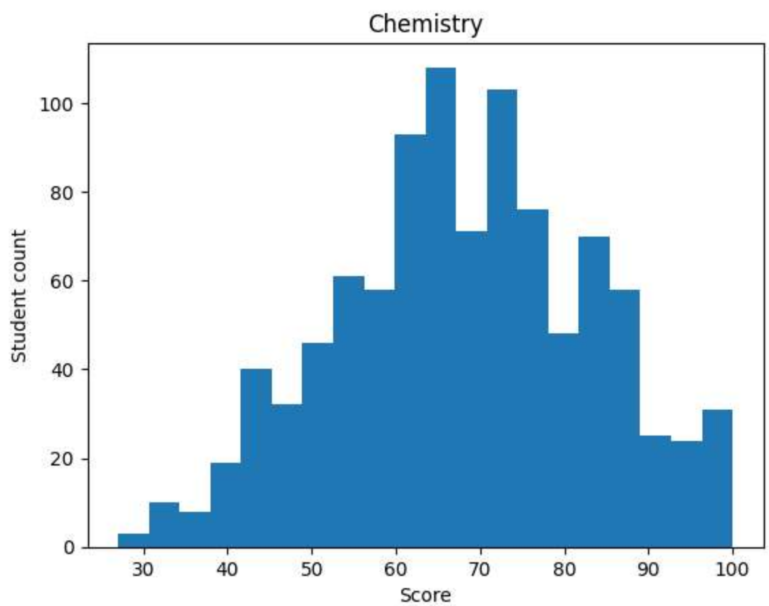
```
# Despues de eliminar los nulos
print(df.isnull().sum())
```

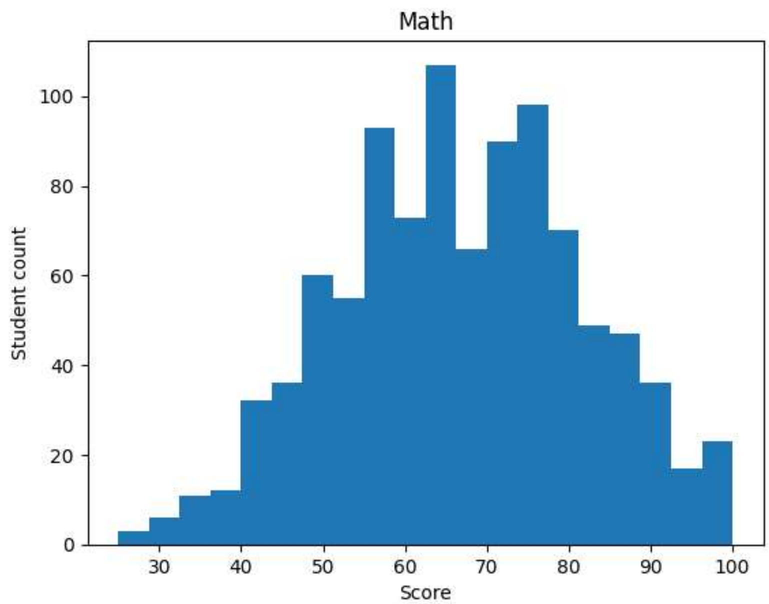
```
id                            0
Gender                        0
Ethnicity                     0
Parental level of education   0
Lunch                         0
Employed                      0
Test preparation course       0
Math score                    7
Physics score                 7
Chemistry score               7
Algebra score                 7
dtype: int64

id                            0
Gender                        0
Ethnicity                     0
Parental level of education   0
Lunch                         0
Employed                      0
Test preparation course       0
Math score                    0
Physics score                 0
Chemistry score               0
Algebra score                 0
dtype: int64
```

```
print(f'Antes: {df.Lunch.count()} filas\n')
Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3 - Q1
print(IQR)
df = df[~((df < (Q1 - 1.5 * IQR)) | (df > (Q3 + 1.5 * IQR))).any(axis=1)]
print(f'\Despues: {df.Lunch.count()} filas')
```

```
Antes: 993 filas

Math score         21.0
Physics score      19.0
Chemistry score    21.0
Algebra score      19.0
dtype: float64
\Despues: 984 filas
/tmp/ipykernel_73/3252133947.py:6: FutureWarning: Automatic reindexing on DataFrame vs Series comparisons is deprecated and will raise ValueError in a future ver
  df = df[~((df < (Q1 - 1.5 * IQR)) | (df > (Q3 + 1.5 * IQR))).any(axis=1)]
/tmp/ipykernel_73/3252133947.py:6: FutureWarning: Automatic reindexing on DataFrame vs Series comparisons is deprecated and will raise ValueError in a future ver
  df = df[~((df < (Q1 - 1.5 * IQR)) | (df > (Q3 + 1.5 * IQR))).any(axis=1)]
```

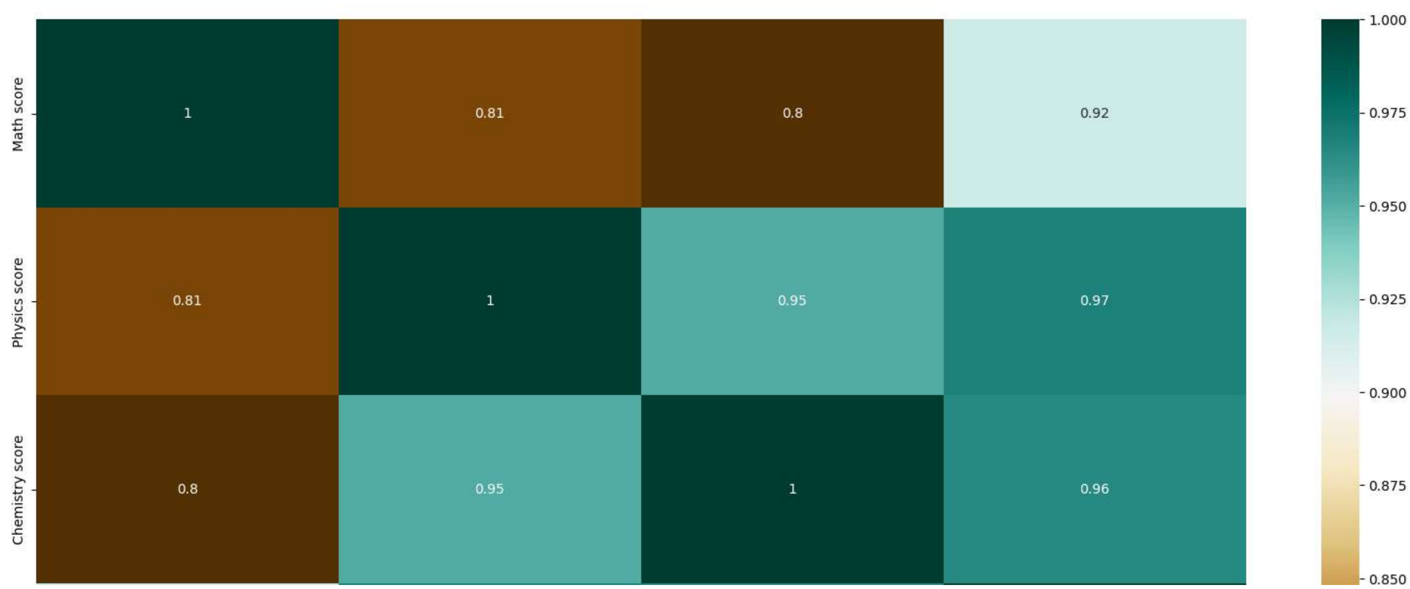

Algebra
Student count
Score
0
20
40
60
80
100
30
40
50
60
70
80
90
100

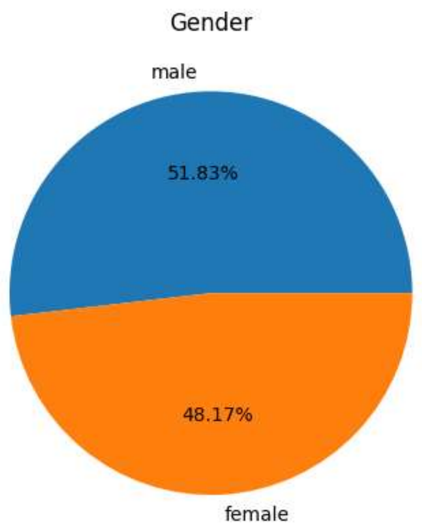

Chemistry
Student count
Score
0
20
40
60
80
100
30
40
50
60
70
80
90
100

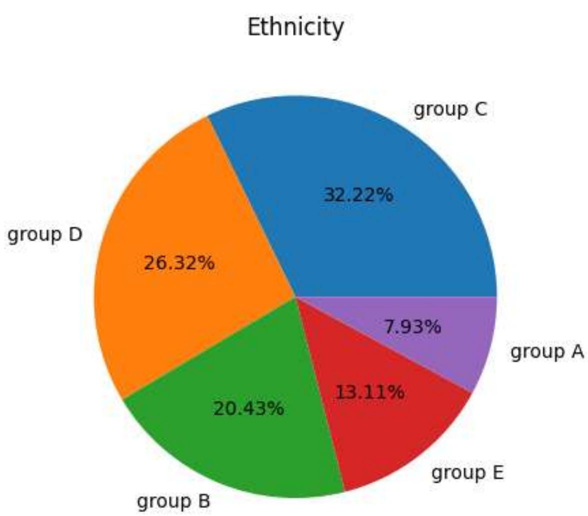


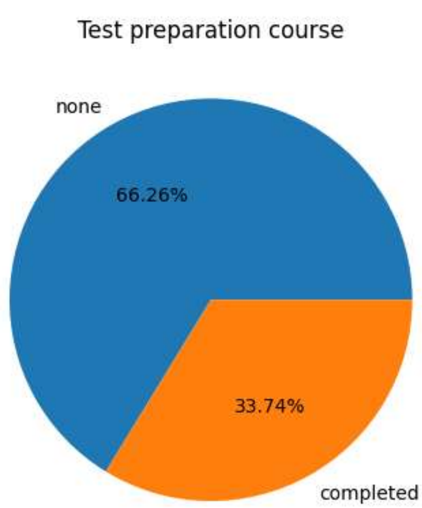

```
# Correlación entre los datos - Mapa de calor
c = df.corr()
print(c)
```

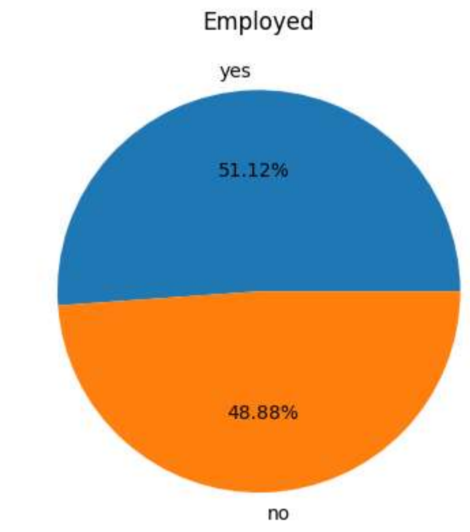
```
                 Math score  Physics score  Chemistry score  Algebra score
Math score         1.000000       0.812055         0.798312       0.916674
Physics score      0.812055       1.000000         0.951536       0.968358
Chemistry score    0.798312       0.951536         1.000000       0.964652
Algebra score      0.916674       0.968358         0.964652       1.000000
```

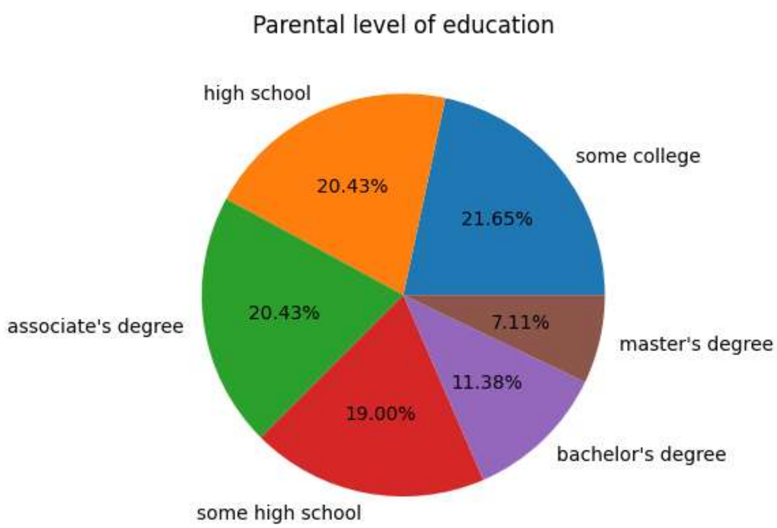


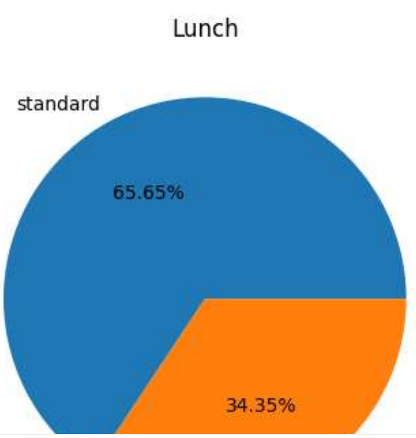

Gender
male
51.83%
48.17%
female


Ethnicity
group C
32.22%
group D
26.32%
7.93%
group A
13.11%
20.43%
group E
group B


Test preparation course
none
66.26%
33.74%
completed







Respondiendo preguntas

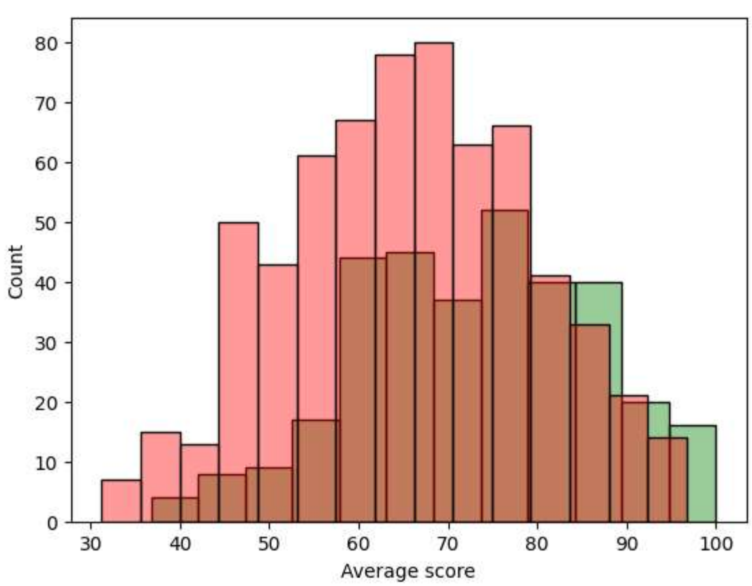
1. Hay alguna relación entre el promedio de notas obtenidas y el hecho de haber realizado el curso preparatorio?

```
df['Average score'] = df.mean(axis = 1)
df
```

	id object	Gender object	Ethnicity object	Parental level of ...	Lunch object	Employed object	Test preparation...	Math score float6
	10-5894942 ...... 0.1% 41-1676468 ...... 0.1% 982 others 99.8%	male ...... 51.8% female 48.2%	group C ...... 32.2% group D ...... 26.3% 3 others 41.5%	some college .. 21.6% high school ..... 20.4% 4 others 57.9%	standard ...... 65.7% free/reduced . 34.3%	yes ...... 51.1% no 48.9%	none ...... 66.3% completed 33.7%	25.0 - 100.0
0	10-5894942	male	group A	high school	standard	yes	completed	67
1	41-1676468	female	group D	some high school	free/reduced	no	none	40
2	64-6396924	male	group E	some college	free/reduced	no	none	59
3	35-2426788	male	group B	high school	standard	yes	none	77
4	60-9387304	male	group E	associate's degree	standard	yes	completed	78
5	67-3666190	female	group D	high school	standard	yes	none	63
6	27-7702214	female	group A	bachelor's degree	standard	yes	none	62
7	46-2257650	male	group E	some college	standard	yes	completed	93
8	40-1499649	male	group D	high school	standard	no	none	63
9	67-7378468	male	group C	some college	free/reduced	no	none	47

```
si = df[df['Test preparation course'] == 'completed']
no = df[df['Test preparation course'] == 'none']
```

```
sns.histplot(si['Average score'], color = 'green', alpha=.4, fill = True)
sns.histplot(no['Average score'], color = 'red', alpha=.4, fill = True)
plt.show()
```




```
print('Realizaron el curso: ', si['Test preparation course'].count())
print('No realizaron el curso: ', no['Test preparation course'].count())
```

```
Realizaron el curso:  332
No realizaron el curso:  652
```

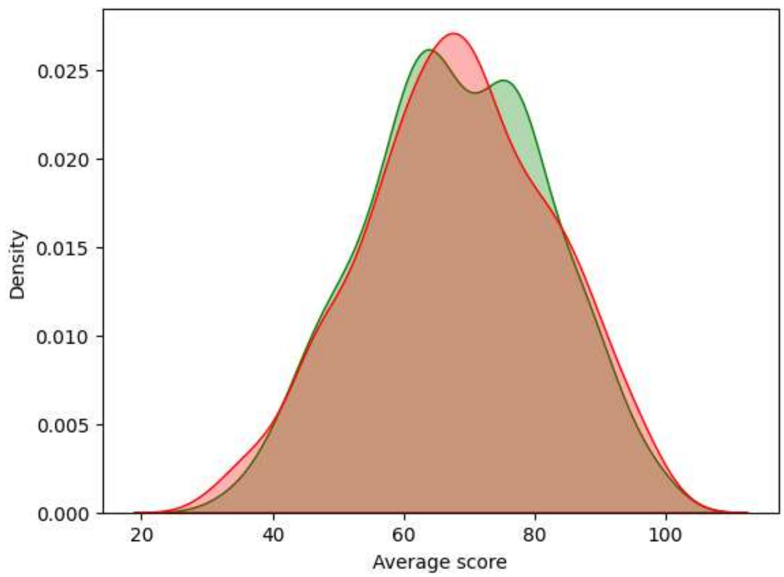
Conclusión: Si bien la cantidad de alumnos que no realizó el curso preparatorio casi duplica a la de quienes lo hay completado, esta diferencia no se ve reflejada siginificativamente en el promedio de notas.

Se recomienda auditar los contenidos del curso, a fines de lograr una mejora en el rendimiento académico y aumentar en interés del alumnado.

2. Hay alguna relación entre las notas obtenidas y el hecho de que este empleado o no el estudiante?

```
YesEmployed = df[df['Employed'] == 'yes'].copy()
NoEmployed = df[df['Employed'] == 'no'].copy()
sns.kdeplot(YesEmployed['Average score'], color = 'green',fill=True, alpha=0.3)
sns.kdeplot(NoEmployed['Average score'], color = 'red',fill=True, alpha=0.3)
```

```
<AxesSubplot: xlabel='Average score', ylabel='Density'>
```



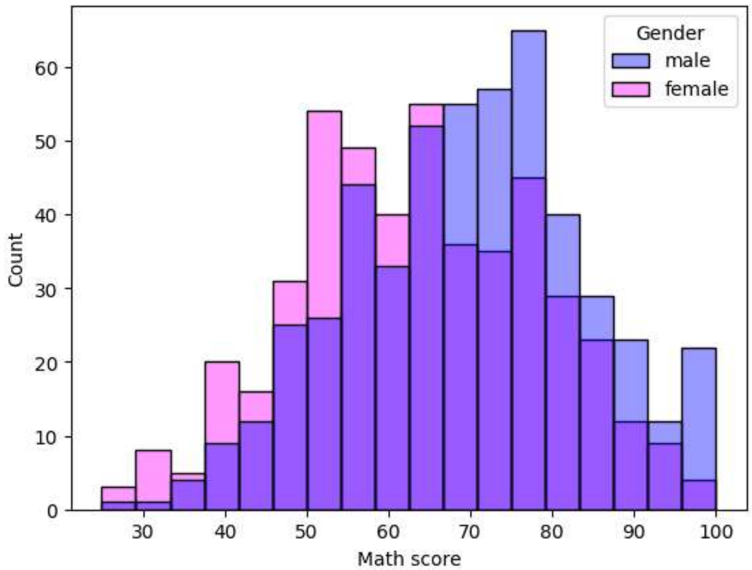

```
print('Empleado: ', YesEmployed['Employed'].count())
print('No empleado: ', NoEmployed['Employed'].count())
```

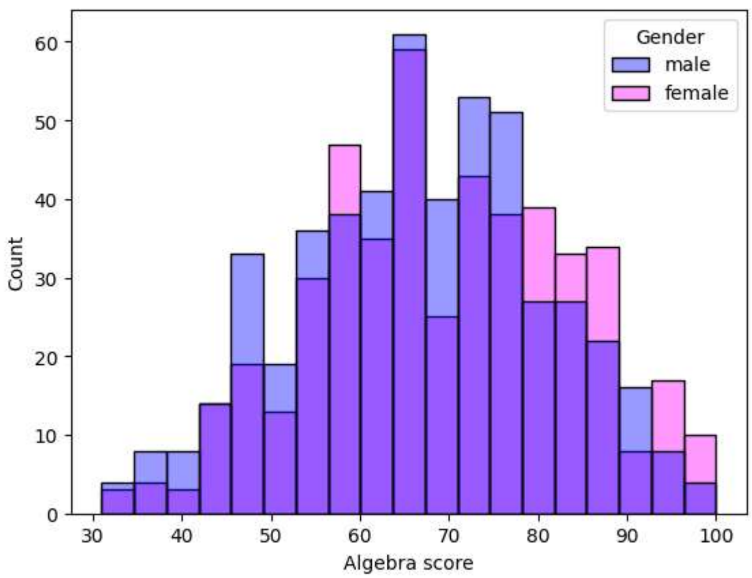
```
Empleado:  503
No empleado:  481
```

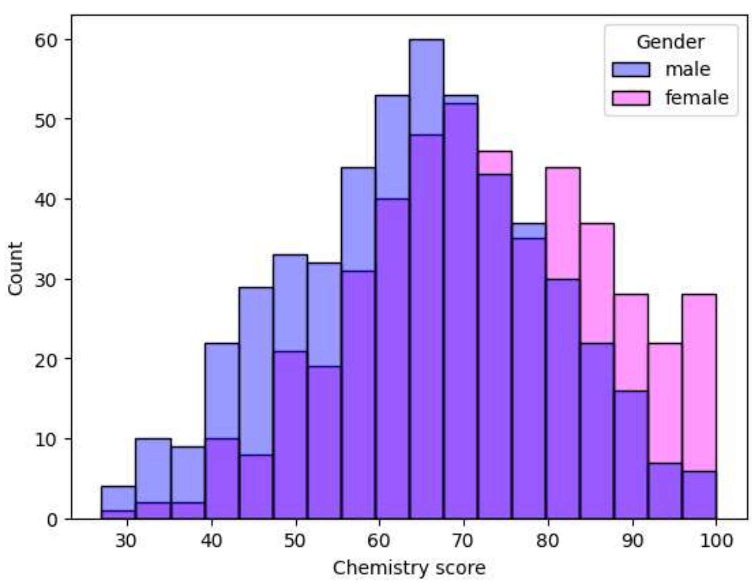
Conclusión: La cantidad de alumnos empleados y desempleados es casi la misma y sin embargo las notas obtenidas no difieren mucho unas de otras. Hay mas alumnos empleados con nota promedio de 80 que alumnos desempleados. Por otro lado, hay mas alumnos desempleados con nota promedio de 70.

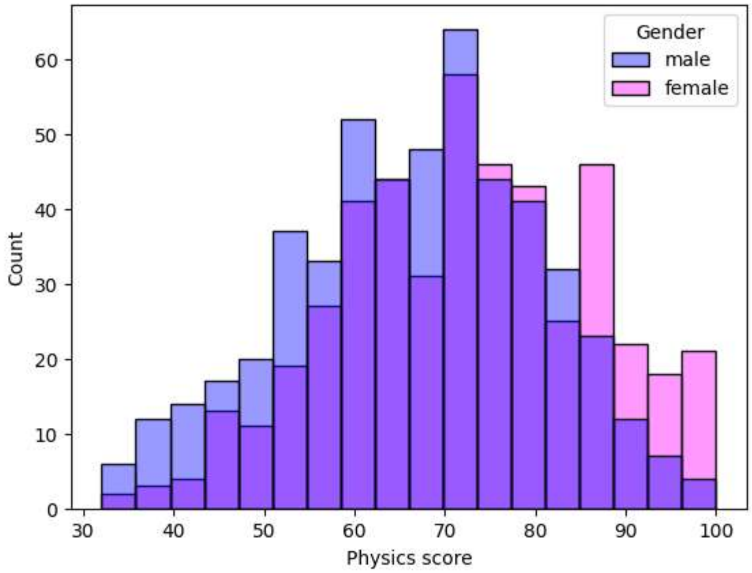
Hay alguna relación entre las notas obtenidas por los hombres y las mujeres?

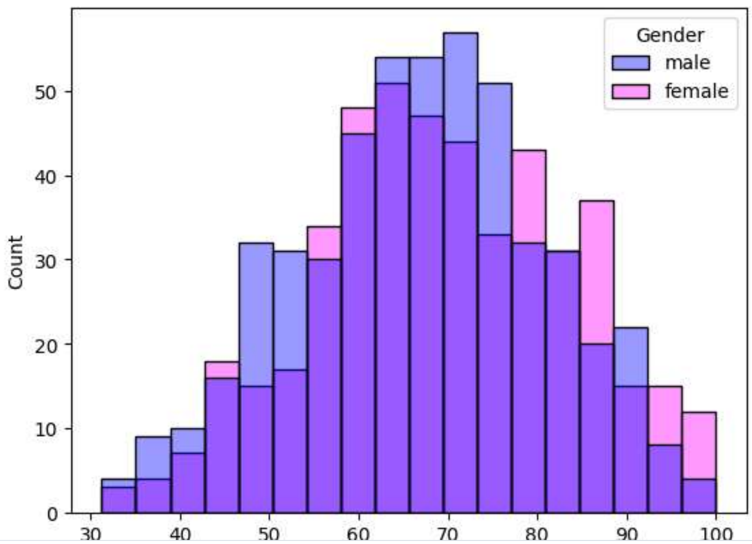
```
sns.histplot(data="exams.csv", x=df["Math score"], hue=df["Gender"], alpha=.4, palette={'fuchsia','blue'})
plt.show()

sns.histplot(data="exams.csv", x=df["Algebra score"], hue=df["Gender"], alpha=.4, palette={'fuchsia','blue'})
plt.show()

sns.histplot(data="exams.csv", x=df["Chemistry score"], hue=df["Gender"], alpha=.4, palette={'fuchsia','blue'})
plt.show()

sns.histplot(data="exams.csv", x=df["Physics score"], hue=df["Gender"], alpha=.4, palette={'fuchsia','blue'})
plt.show()

sns.histplot(data="exams.csv", x=df["Average score"], hue=df["Gender"], alpha=.4, palette={'fuchsia','blue'})
plt.show()
```


Gender
male
female
Count
Math score


Gender
male
female
Count
Algebra score


Gender
male
female
Count
Chemistry score





Conclusión: Podemos observar que en Matemática los hombres obtuvieron mejores notas que las mujeres, mientras que en el resto de las materias las mujeres fueron las que obtuvieron mejor puntaje.

En general, en el puntaje promedio podemos observar entre los 75 y 100 puntos que las mujeres obtuvieron puntaje mayor a los hombres. En cambio entre los 60 y 75 obtuvieron mayot puntaje los hombres.