Visualización

1. Dependencias

```
In [104…  import matplotlib.pyplot as plt
          import numpy as np
          import pandas as pd
          import plotly.express as px
          import pandoc as pdc
          import seaborn as sns
```

2. Importar data

```
In [130…  #Se define la ruta en la que guardamos la data que usaremos
          ruta = "D:/Users/Camilo 323/Desktop/Universidad/Python Repositorio/Corredores_Davivier
```

```
In [131…  students = pd.read_csv(ruta+'StudentsPerformance.csv')
```

```
In [133…  #Cambiamos el nombre de la columna para evitar problemas con el "/"
          students.rename(columns = {'race/ethnicity':'ethnicity'},inplace=True)
```

```
In [134…  students.isna().sum()
          #No hay Na's
```

```
Out[134]:  gender                         0
           ethnicity                      0
           parental level of education    0
           lunch                          0
           test preparation course        0
           math score                     0
           reading score                  0
           writing score                  0
           dtype: int64
```

```
In [135…  students.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 8 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   gender                       1000 non-null   object
 1   ethnicity                    1000 non-null   object
 2   parental level of education  1000 non-null   object
 3   lunch                        1000 non-null   object
 4   test preparation course      1000 non-null   object
 5   math score                   1000 non-null   int64
 6   reading score                1000 non-null   int64
 7   writing score                1000 non-null   int64
dtypes: int64(3), object(5)
memory usage: 62.6+ KB
```

```
In [136…   students.loc[students["parental level of education"] == "associate's degree", "parenta
           students.loc[students["parental level of education"] == "bachelor's degree", "parental
           students.loc[students["parental level of education"] == "high school", "parental level
           students.loc[students["parental level of education"] == "master's degree", "parental l
           students.loc[students["parental level of education"] == "some college", "parental leve
           students.loc[students["parental level of education"] == "some high school", "parental
```

```
In [137…   students.head()
```

Out[137]:

	gender	ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	5	standard	none	72	72	74
1	female	group C	3	standard	completed	69	90	88
2	female	group B	6	standard	none	90	95	93
3	male	group A	4	free/reduced	none	47	57	44
4	male	group C	3	standard	none	76	78	75

3. Estadística descrptiva y visualización

```
In [139…   students['mean score']= students[['math score','reading score','writing score']].mean(
```

```
In [120…   #Tabla de estadística descriptiva
           #Unicamente para variables cuantitativas
           students.describe()
```

Out[120]:

	math score	reading score	writing score	mean score
count	1000.00000	1000.000000	1000.000000	1000.000000
mean	66.08900	69.169000	68.054000	67.770667
std	15.16308	14.600192	15.195657	14.257326
min	0.00000	17.000000	10.000000	9.000000
25%	57.00000	59.000000	57.750000	58.333333
50%	66.00000	70.000000	69.000000	68.333333
75%	77.00000	79.000000	79.000000	77.666667
max	100.00000	100.000000	100.000000	100.000000

```
In [55]:   students.columns
```

```
Out[55]:   Index(['gender', 'ethnicity', 'parental level of education', 'lunch',
                  'test preparation course', 'math score', 'reading score',
                  'writing score'],
                 dtype='object')
```

```
In [60]:   #Porcentaje de estudiantes por etnicidad
           plt.figure(figsize = (6,6))
```

```
plt.pie(students.ethnicity.value_counts().values, labels = students.ethnicity.value_co
plt.title('Porcentaje de estudiantes por etnicidad')
```

Out[60]:   Text(0.5, 1.0, 'Porcentaje de estudiantes por etnicidad')

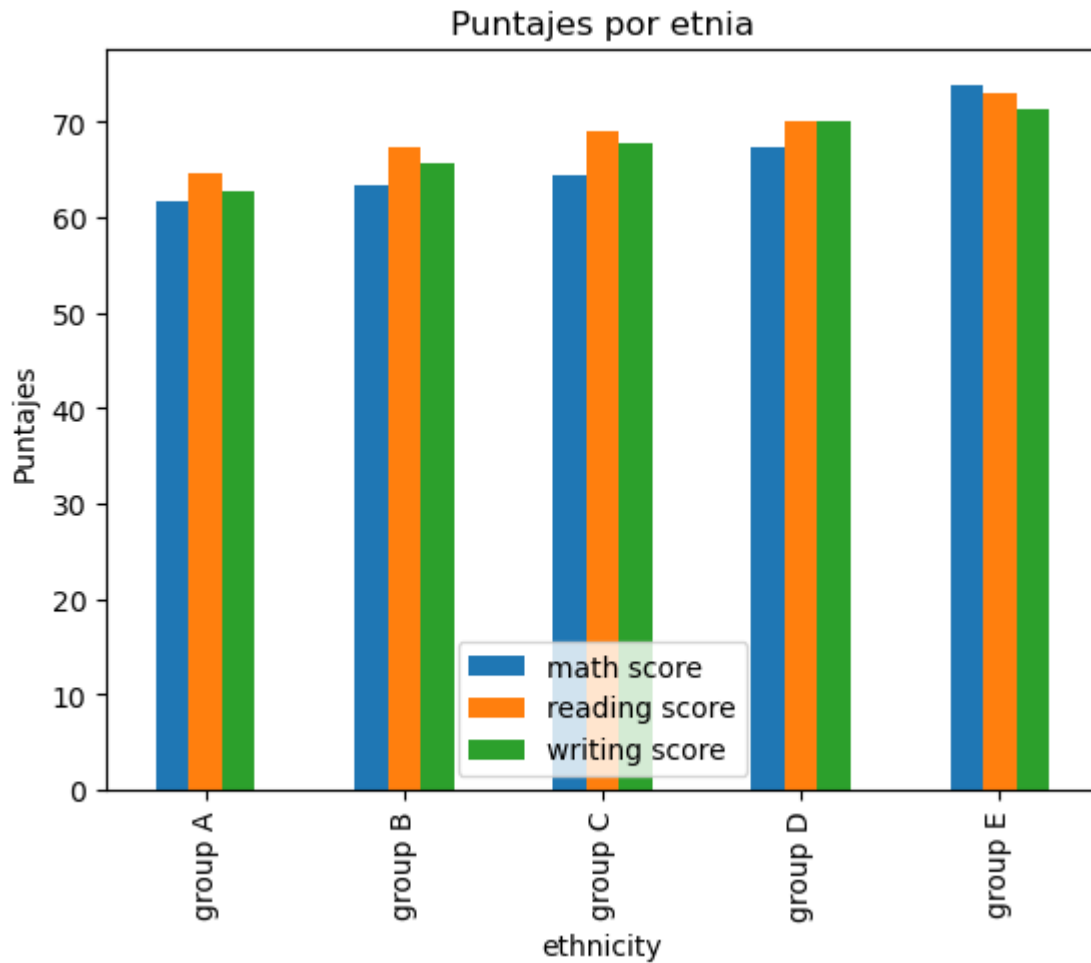## Porcentaje de estudiantes por etnicidad
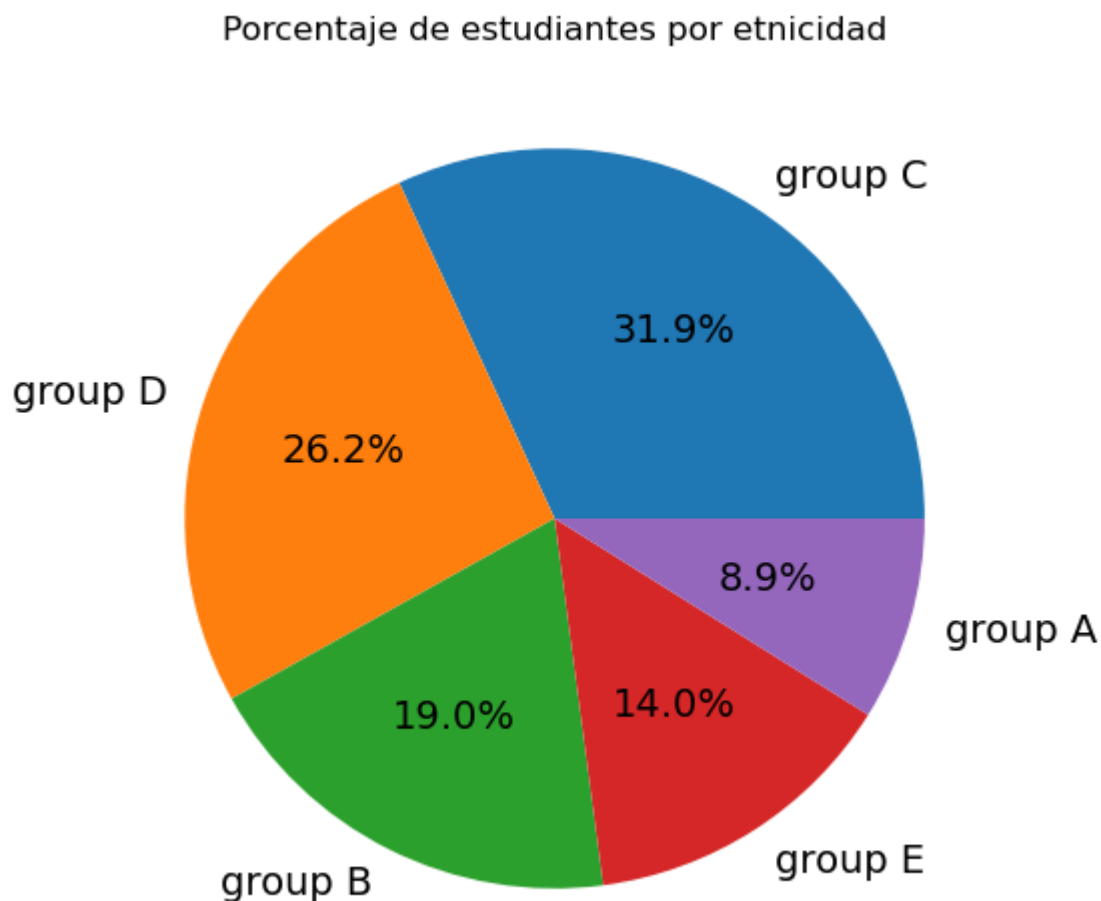

```
In [92]:    plt.figure(figsize = (6,6))
            pd.pivot_table(students, index = 'ethnicity', aggfunc = 'mean', values = ('math score'
            plt.title('Puntajes por etnia')
            plt.ylabel('Puntajes')
```
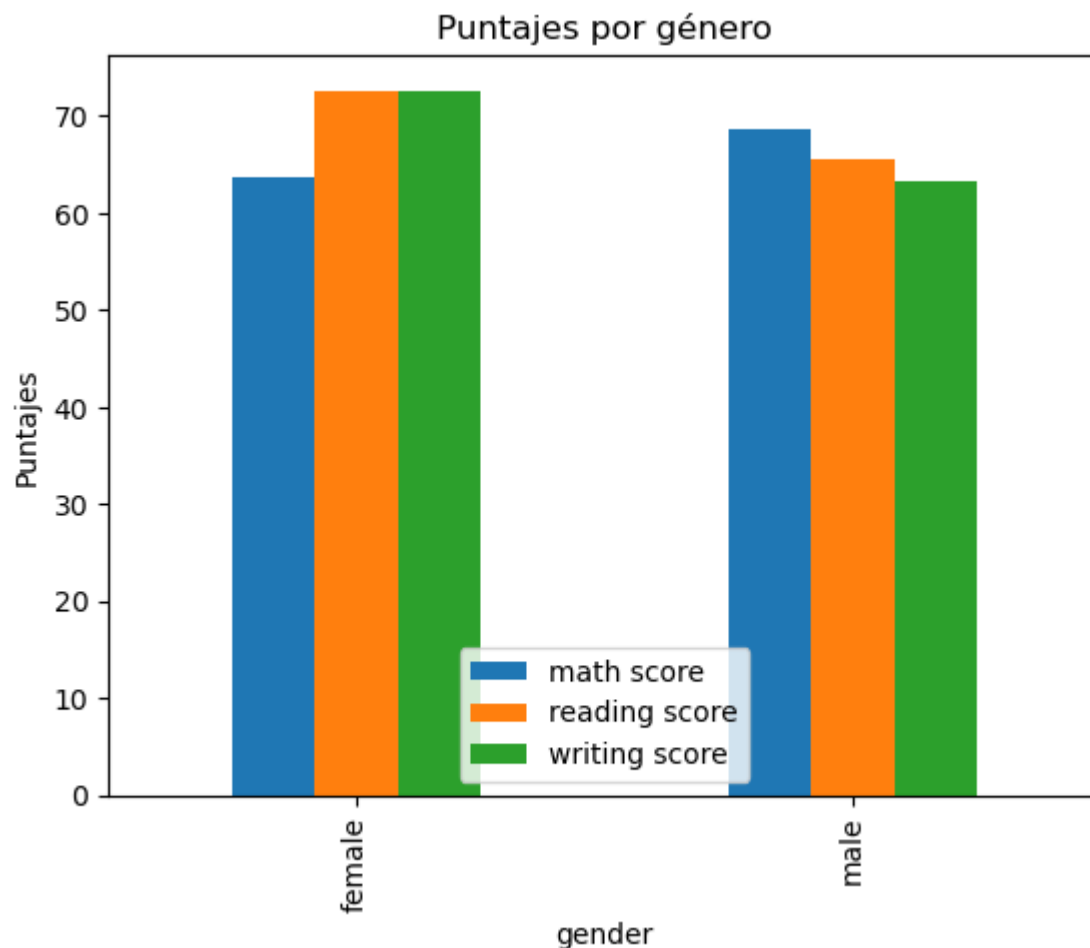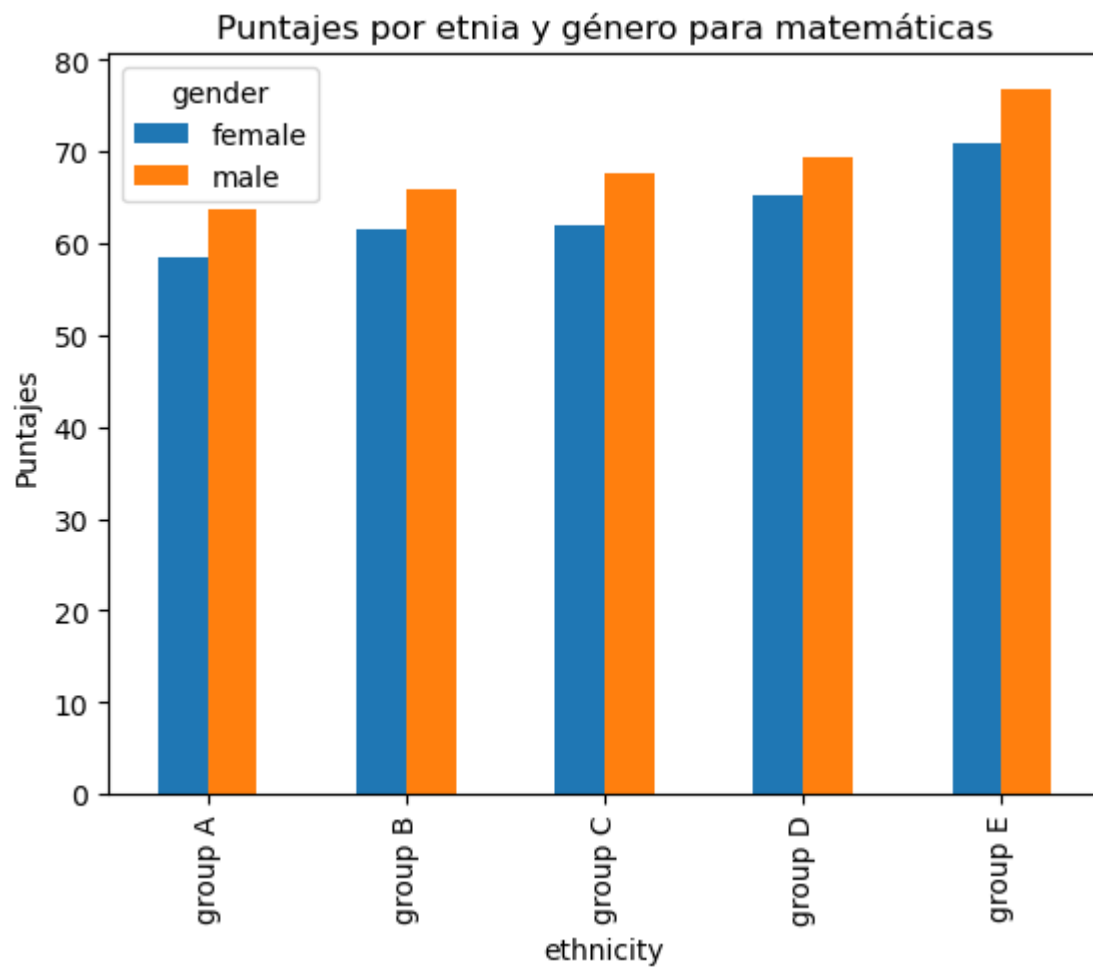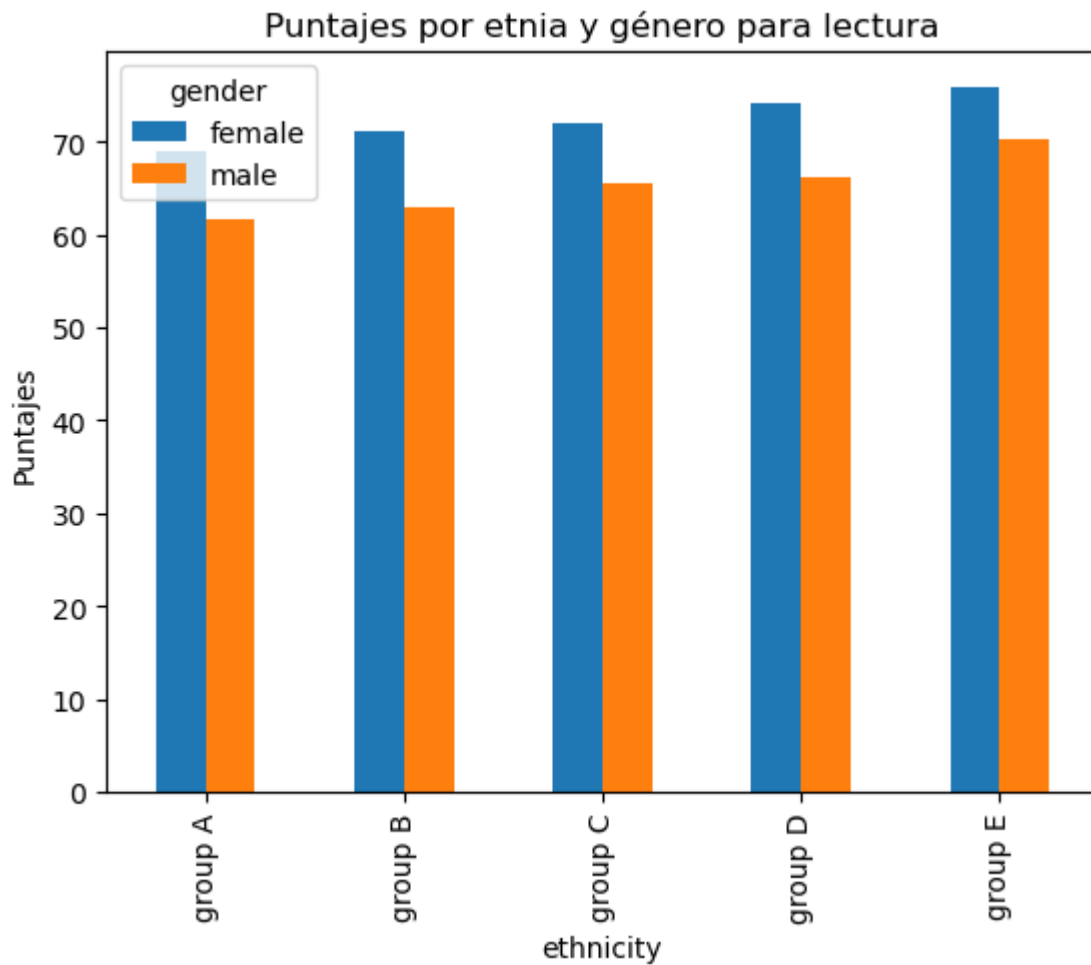
Out[92]:   Text(0, 0.5, 'Puntajes')

            <Figure size 600x600 with 0 Axes>

Puntajes por etnia


```
In [96]:  plt.figure(figsize = (6,6))
          pd.pivot_table(students, index = 'gender', aggfunc = 'mean', values = ('math score','r
          plt.title('Puntajes por género')
          plt.ylabel('Puntajes')
```

```
Out[96]:  Text(0, 0.5, 'Puntajes')

          <Figure size 600x600 with 0 Axes>
```

```python
plt.figure(figsize = (6,6))
pd.pivot_table(students, index = 'ethnicity', columns= 'gender',aggfunc = 'mean', valu
plt.title('Puntajes por etnia y género para matemáticas')
plt.ylabel('Puntajes')
```

Out[94]: Text(0, 0.5, 'Puntajes')

<Figure size 600x600 with 0 Axes>

Puntajes por etnia y género para matemáticas


```
In [85]:  pd.pivot_table(students, index = 'ethnicity', columns= 'gender',aggfunc = 'mean', valu
          plt.title('Puntajes por etnia y género para lectura')
          plt.ylabel('Puntajes')
```

Out[85]: Text(0, 0.5, 'Puntajes')

Puntajes por etnia y género para lectura

```
In [123…   plt.figure(figsize = (6,6))
           pd.pivot_table(students, index = 'ethnicity', columns= 'gender',aggfunc = 'mean', valu
           plt.title('Puntajes por etnicidad y género para escritura')
           plt.ylabel('Puntajes')
```

Out[123]:   Text(0, 0.5, 'Puntajes')

            <Figure size 600x600 with 0 Axes>

Puntajes por etnicidad y género para escritura
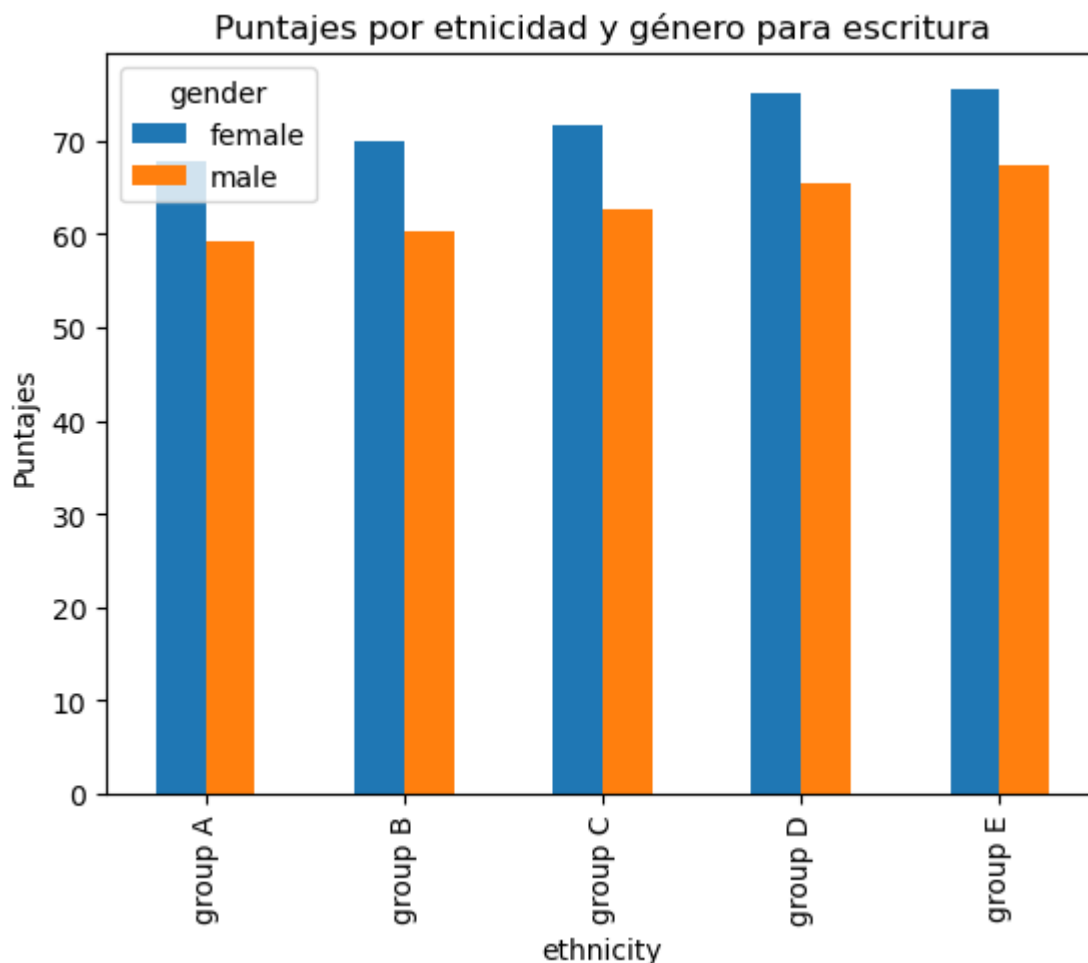
Según esta muestra, las mujeres tienen mejores puntajes en lectura y escritura que los hombres y los hombres mejores en lasmatemáticas que las mujeres. Estos resultados son consistentes para todos los grupos étnicos, además vale la pensa agregar que el grupo E es mejor que todos los otros en las 3 pruebas y el grupo A es el peor en todas las pruebas.
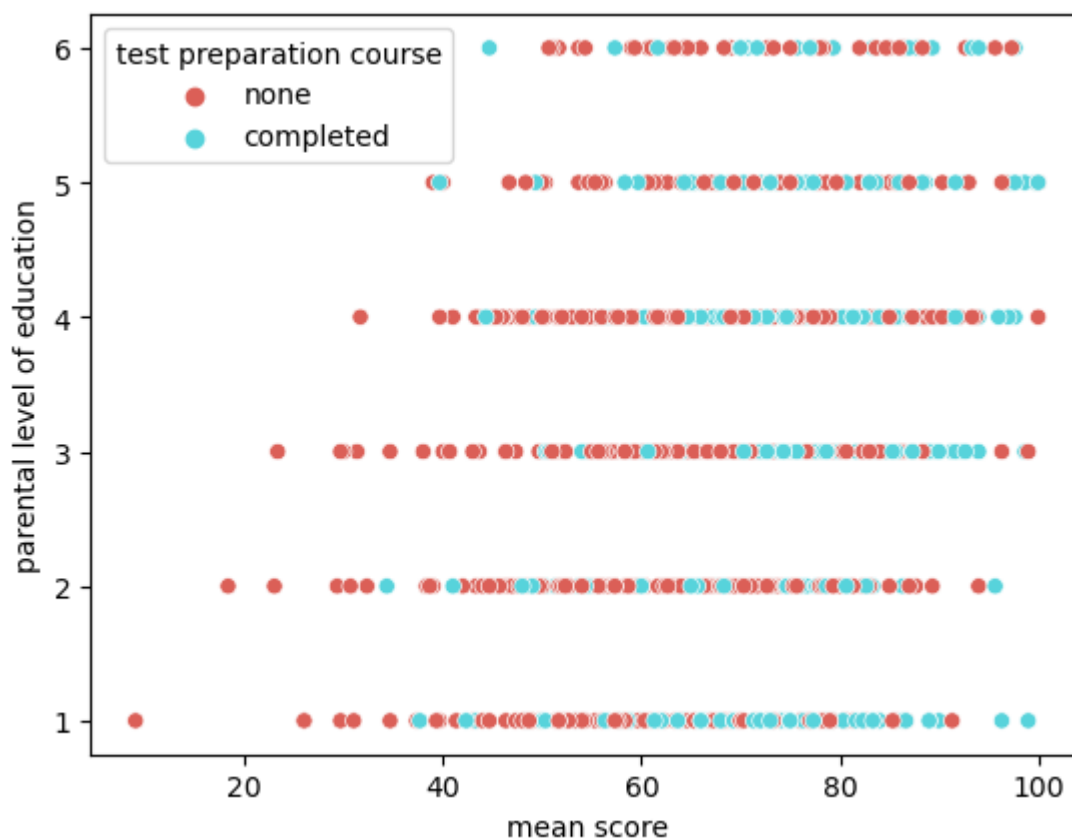
In [154…
```python
#Scatter plot
sns.scatterplot(students['mean score'],students['parental level of education'], hue =
```

```
C:\Users\Camilo 323\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarn
ing: Pass the following variables as keyword args: x, y. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an expl
icit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[154]:
```
<AxesSubplot:xlabel='mean score', ylabel='parental level of education'>
```

```
In [156…  #Scatter plot 2
          sns.scatterplot(students['mean score'],students['parental level of education'], hue =
```

C:\Users\Camilo 323\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarn
ing: Pass the following variables as keyword args: x, y. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an expl
icit keyword will result in an error or misinterpretation.
  warnings.warn(

Out[156]: <AxesSubplot:xlabel='mean score', ylabel='parental level of education'>