**Visualización de datos**

**Equipo número 5**

**Grupo: 03, Lunes**


- **1595894 GONZALEZ CAMPOS EDSON ALI**
- **1941521 LOPEZ DOMINGUEZ FRANCISCO EVERARDO**
- **1663288 HERRERA RIVERA JENNIFER JACQUELINE**
- **1981779 SEGOVIA GONZÁLEZ ANAKAREN**


**Segunda base de datos**

- **Se tradujeron algunos nombres para mejor entendimiento.**
- **Se cambiaron el nombre de algunas columnas para mejor manejo de la información.**

In [156]:

```
import pandas as pd
import numpy as np
import json

data_frame = pd.read_csv("StudentsPerformance.csv",header =0,sep= ",")

data_frame = data_frame.rename(columns = {'gender':'Género'})
data_frame = data_frame.rename(columns = {'race/ethnicity':'RazaEtnia'})
data_frame = data_frame.rename(columns = {'parental level of education':'Nivel de educaci
ón de los padres'})
data_frame = data_frame.rename(columns = {'lunch':'Comida o refrigerios'})
data_frame = data_frame.rename(columns = {'test preparation course':'Curso de preparació
n para exámenes'})
data_frame = data_frame.rename(columns = {'math score':'mathscore'})
data_frame = data_frame.rename(columns = {'writing score':'writingscore'})

print(data_frame)
```

```
     Género RazaEtnia Nivel de educación de los padres Comida o refrigerios  \
0    female   group B                bachelor's degree             standard
1    female   group C                     some college             standard
2    female   group B                  master's degree             standard
3      male   group A               associate's degree         free/reduced
4      male   group C                     some college             standard
..      ...       ...                              ...                  ...
995  female   group E                  master's degree             standard
996    male   group C                      high school         free/reduced
997  female   group C                      high school         free/reduced
998  female   group D                     some college             standard
999  female   group D                     some college         free/reduced

    Curso de preparación para exámenes  mathscore  reading score  writingscore
0                                 none         72             72            74
1                            completed         69             90            88
2                                 none         90             95            93
3                                 none         47             57            44
4                                 none         76             78            75
..                                 ...        ...            ...           ...
995                          completed         88             99            95
996                               none         62             55            55
997                          completed         59             71            65
998                          completed         68             78            77
999                               none         77             86            86

[1000 rows x 8 columns]
```

## Funciones de estadísticas

## Funciones de estadísticas

**Aplicamos las funciones para sacar las estadísticas que se nos pide.**

- **Suma**

In [19]:

```
data_frame['mathscore'].sum()
```

Out[19]:

```
66089
```

- **Promedio**

In [21]:

```
data_frame['writingscore'].mean()
```

Out[21]:

```
68.054
```

- **Suma acumulada de la columna fila por fila**

In [23]:

```
data_frame['reading score'].cumsum()
```

Out[23]:

```
0          72
1         162
2         257
3         314
4         392
        ...
995     68879
996     68934
997     69005
998     69083
999     69169
Name: reading score, Length: 1000, dtype: int64
```

- **Resumen estadístico de la columna**

In [24]:

```
data_frame['Género'].describe()
```

Out[24]:

```
count        1000
unique          2
top        female
freq          518
Name: Género, dtype: object
```

- **Cuantos elementos no nulos hay en la columna**

In [26]:

```
data_frame['Curso de preparación para exámenes'].count()
```

Out[26]:

```
1000
```

- **Minimo y máximo de una columna**

In [27]:

```
data_frame['mathscore'].max()
```

Out[27]:

```
100
```

In [28]:

```
data_frame['mathscore'].min()
```

Out[28]:

```
0
```

- **Mediana, varianza y desviación estandar**

In [29]:

```
data_frame['writingscore'].median()
```

Out[29]:

```
69.0
```

In [30]:

```
data_frame['writingscore'].var()
```

Out[30]:

```
230.90799199199168
```

In [31]:

```
data_frame['writingscore'].std()
```

Out[31]:

```
15.195657010869642
```

- **Valor de asimetría en los datos**

In [32]:

```
data_frame['reading score'].skew()
```

Out[32]:

```
-0.25910451810923063
```

- **Característica de forma de su distribución de frecuencias/probabilidad**

In [33]:

```
data_frame['reading score'].kurt()
```

Out[33]:

```
-0.0682654585647704
```

- **Correlación de los datos**

In [158]:

```
data_frame.corr()
```

Out[158]:

| | mathscore | reading score | writingscore |
|---|---|---|---|
| **mathscore** | 1.000000 | 0.817580 | 0.802642 |
| **reading score** | 0.817580 | 1.000000 | 0.954598 |
| **writingscore** | 0.802642 | 0.954598 | 1.000000 |

- **Covarianza de los datos**

In [157]:

```
data_frame.cov()
```

Out[157]:

| | mathscore | reading score | writingscore |
|---|---|---|---|
| **mathscore** | 229.918998 | 180.998958 | 184.939133 |
| **reading score** | 180.998958 | 213.165605 | 211.786661 |
| **writingscore** | 184.939133 | 211.786661 | 230.907992 |

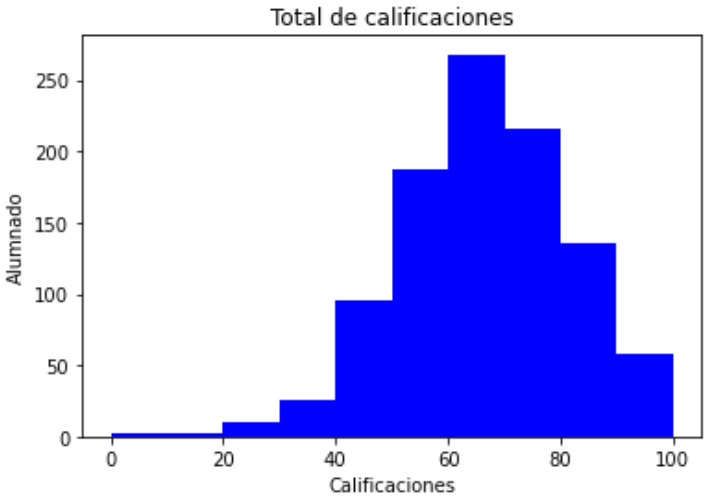
## Gráficos

## - Histograma

In [50]:

```
import matplotlib.pyplot as plt
%matplotlib inline

plt.hist(data_frame['mathscore'], color= 'blue')
plt.xlabel('Calificaciones')
plt.ylabel('Alumnado')
plt.title ("Total de calificaciones")
plt.show()
```




## Conclusión:

**Se puede concluir que hay más alumnos con calificaciones entre 80 y 60, en la materia de matemáticas.**

## Grafica de barras y datos categoricos:

**El gráfico de abajo nos representa un plot de los datos numericos**

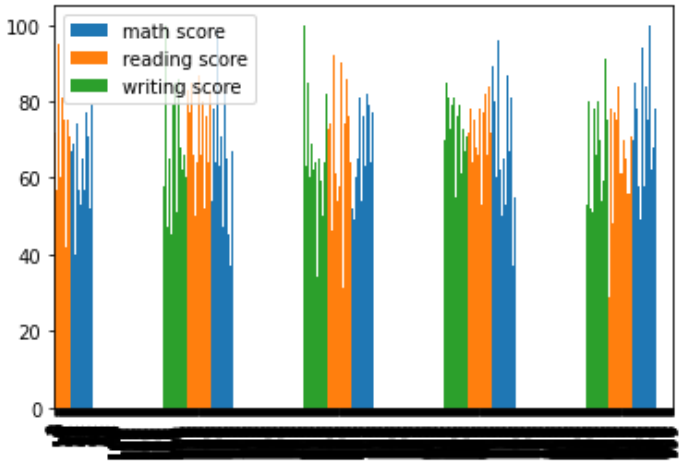
El grafico de abajo nos representa un plot de los datos numericos.

In [47]:

```
data_frame.plot.bar()
```

Out[47]:

<AxesSubplot:>




## Conclusión:

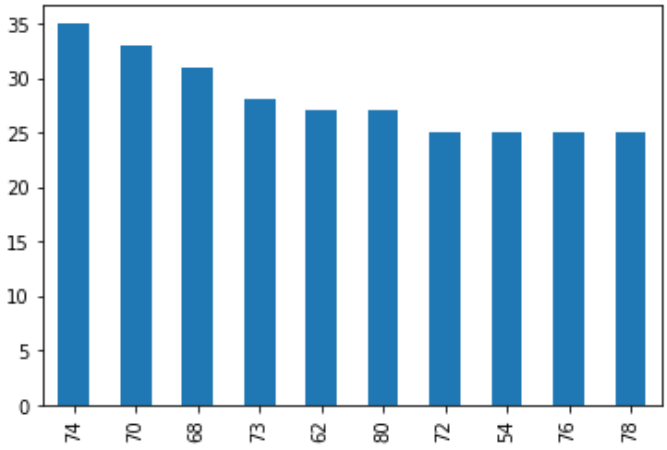
**Se puede comparar en que asignatura es mejor cada raza/etnia.**

In [59]:

```
data_frame['writingscore'].value_counts().head(10).plot.bar()
```

Out[59]:

<AxesSubplot:>




## Conclusión:

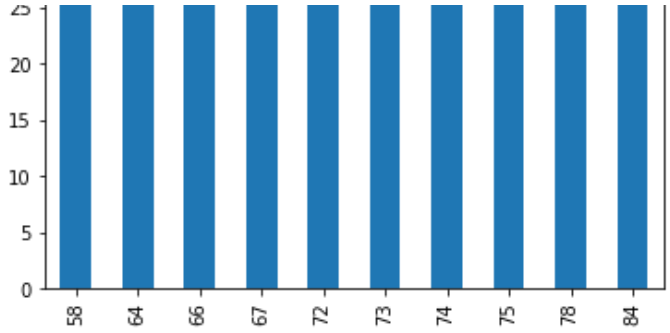
**Esta función nos da la información de cuales son las calificaciones que más se repiten en la asignatura de escritura.**

In [63]:

```
data_frame['reading score'].value_counts().head(10).sort_index().plot.bar()
```

Out[63]:

<AxesSubplot:>




## Conclusión:

**Se puede ver una media en las calificaciones en la asignatura de lectura.**

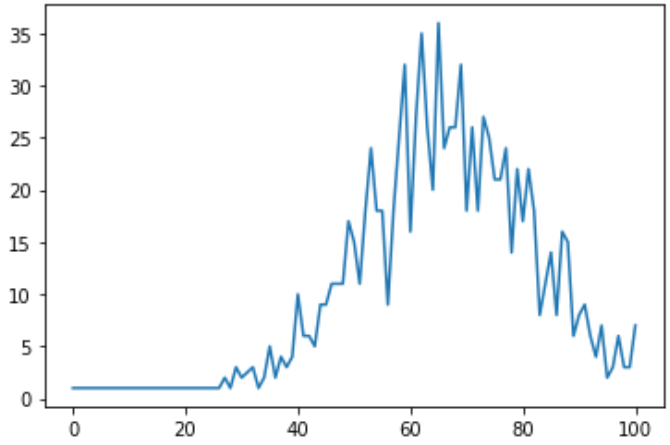
## - Lineas

In [64]:

```
data_frame['mathscore'].value_counts().sort_index().plot.line()
```

Out[64]:

<AxesSubplot:>




## Conclusión:

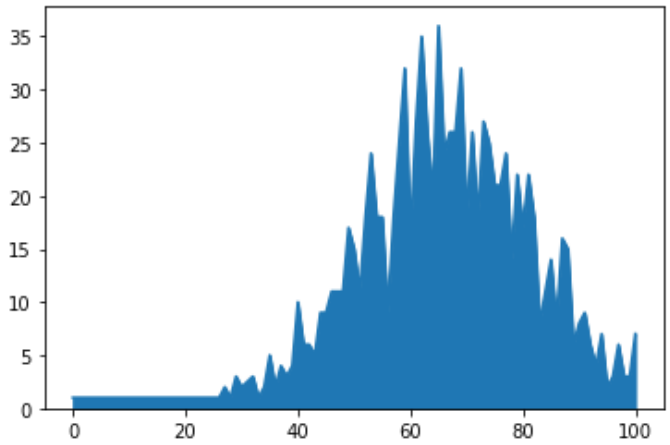
**Nos muestra con un gráfico de lineas cuantos alumnos tienen una cierta calificación, también nos muestra que hay más entre 60 y 70, pero con una mayor presición que el histograma.**

In [65]:

```
data_frame['mathscore'].value_counts().sort_index().plot.area()
```

Out[65]:

<AxesSubplot:>



## Conclusión:

**Nos muestra la solidez de los datos.**

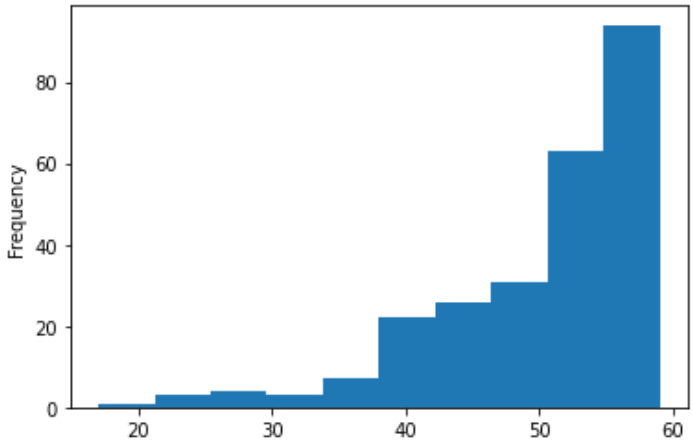
## Histogramas con datos en intervalos

In [67]:

```
data_frame[data_frame['reading score'] < 60]['reading score'].plot.hist()
```

Out[67]:

```
<AxesSubplot:ylabel='Frequency'>
```




## Conclusión:

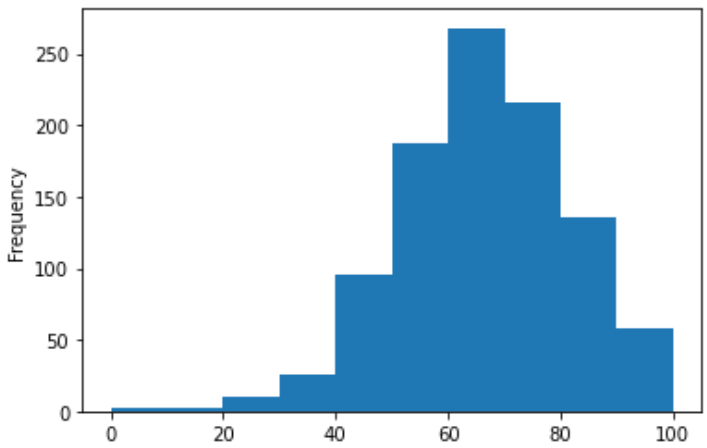
**Aquí nos muestra un histograma con las calificaciones menores e iguales a 60 y, la cantidad de alumnos con dichas calificaiones.**

In [68]:

```
data_frame['mathscore'].plot.hist()
```

Out[68]:

```
<AxesSubplot:ylabel='Frequency'>
```




## Conclusión:

**Nos muestra los datos, pero sin poder poner los parámetros que representa.**

## Gráficas bi-variantes

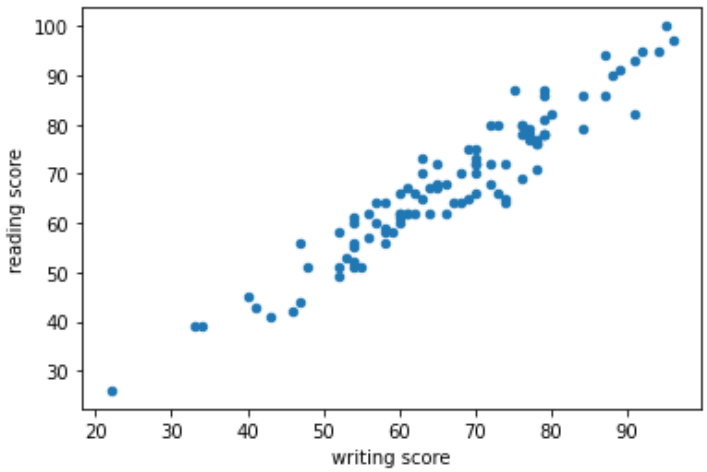
## - Scatter plot

In [73]:

```
data_frame[data_frame['writingscore'] < 100].sample(100).plot.scatter(x='writingscore',
y='reading score')
```

Out[73]:

```
<AxesSubplot:xlabel='writing score', ylabel='reading score'>
```




## Conclusión:

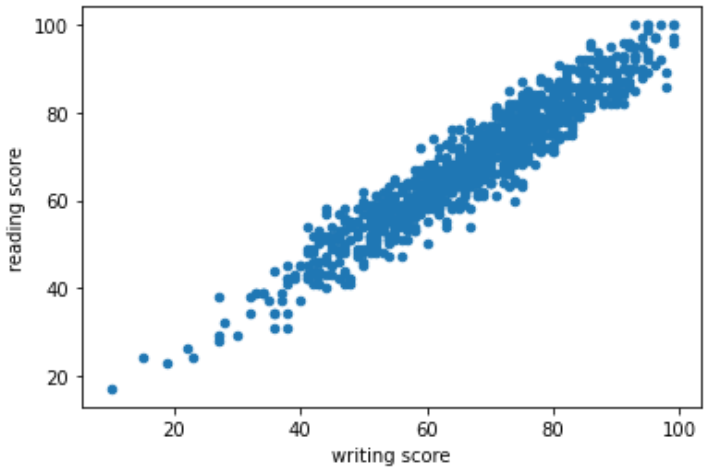
**Podemos concluir una correlación positiva-débil entre las calificaciones de las asignaturas de lectura y escrituta, ya que va hacía arriba a la derecha y es débil porque no es una linea recta marcada y, agregamos que relativamente les va bien. Haciendo referencia que solo se tomaron 100 datos.**

In [71]:

```
data_frame[data_frame['writingscore'] < 100].plot.scatter(x='writingscore', y='reading sc
ore')
```

Out[71]:

```
<AxesSubplot:xlabel='writing score', ylabel='reading score'>
```




## Conclusión:

**Aquí se puede ver la relación con todos los datos, lo que nos confirma que es una correlación débil, ya que no forma una linea recta marcada.**

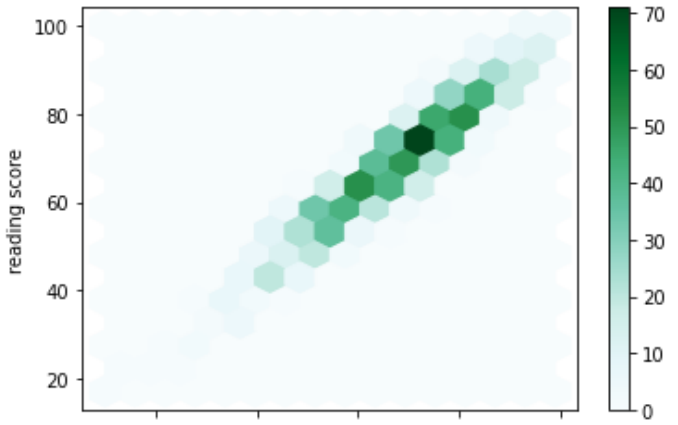
In [74]:

```
data_frame[data_frame['writingscore'] < 100].plot.hexbin(x='writingscore', y='reading sco
re', gridsize=15)
```

Out[74]:

Out[74]:

```
<AxesSubplot:xlabel='writing score', ylabel='reading score'>
```




## Conclusión:

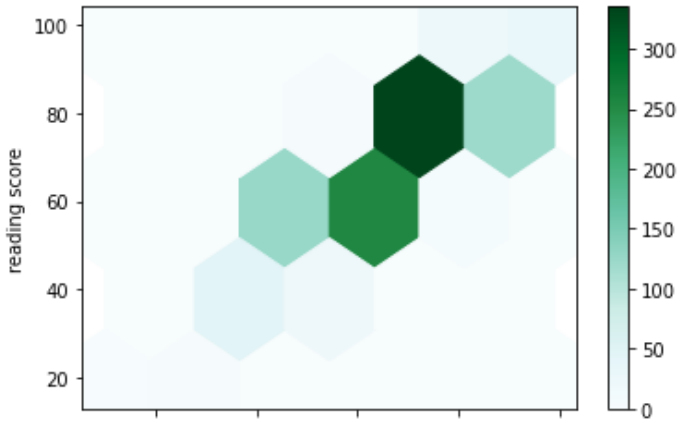
**Nos muestra donde hay una mayor concentración de datos.**

In [76]:

```
data_frame[data_frame['writingscore'] < 100].plot.hexbin(x='writingscore', y='reading sco
re', gridsize=5)
```

Out[76]:

```
<AxesSubplot:xlabel='writing score', ylabel='reading score'>
```




## Conclución:

**Aquí nos muestra un rango más grande en donde están concentrados los datos.**

In [82]:

```
wine_counts = pd.read_csv("StudentsPerformance.csv",index_col=0)
wine_counts.head()
```

Out[82]:

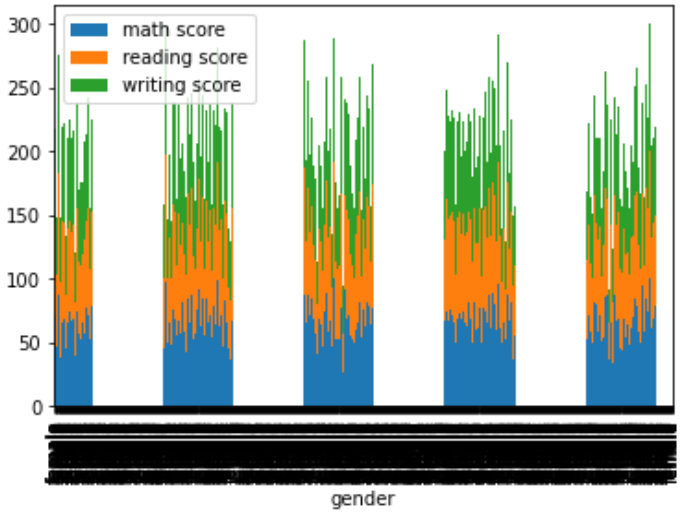
| gender | race/ethnicity | parental level of education | lunch | test preparation course | math score | reading score | writing score |
|---|---|---|---|---|---|---|---|
| female | group B | bachelor's degree | standard | none | 72 | 72 | 74 |
| female | group C | some college | standard | completed | 69 | 90 | 88 |
| female | group B | master's degree | standard | none | 90 | 95 | 93 |
| male | group A | associate's degree | free/reduced | none | 47 | 57 | 44 |
| male | group C | some college | standard | none | 76 | 78 | 75 |

In [79]:

```
wine_counts.plot.bar(stacked=True)
```

Out[79]:

<AxesSubplot:xlabel='gender'>




## Conclusión:

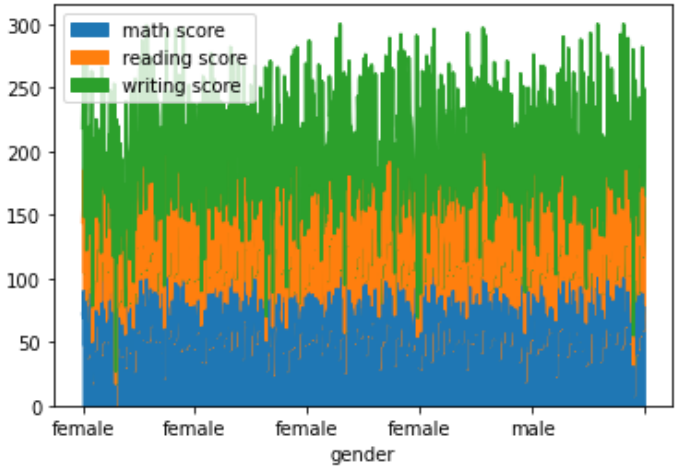
**Se muestra la relación entre las tres asignaturas, en los distintos grupos, se pueden ver cosas como que en el grupo c hay una mayor variabilidad en la escritura, en el grupo b, se muestra la mayor constancia en las personas con misma calificación.**

In [80]:

```
wine_counts.plot.area()
```

Out[80]:

<AxesSubplot:xlabel='gender'>




## Conclusión:

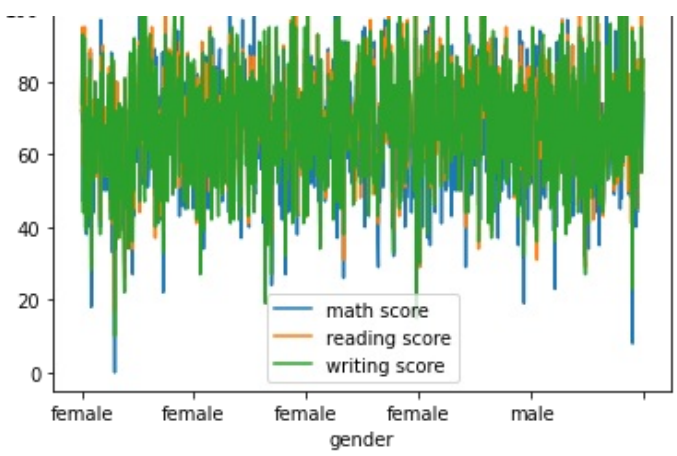
**Poco podemos concluir, ya que hay datos encima de otros.**

In [81]:

```
wine_counts.plot.line()
```

Out[81]:

<AxesSubplot:xlabel='gender'>




## Conclusión:

**Aunque los datos estén encima de otros, podemos ver como van variando las calificaciones de las diversas asignaturas.**

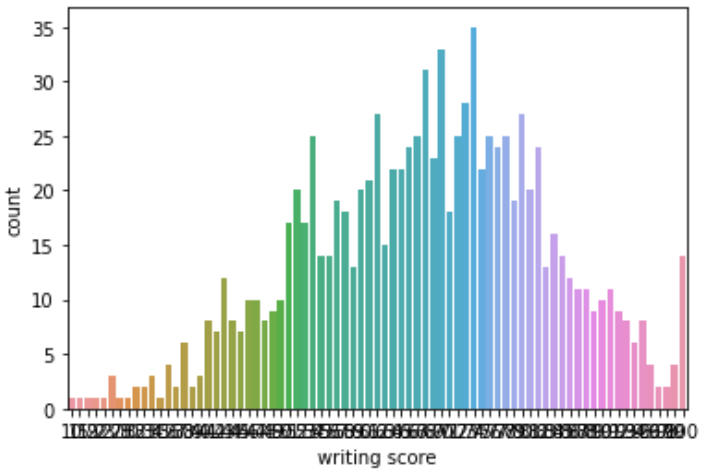
In [98]:

```
import seaborn as sns
sns.countplot(data_frame['writingscore'])
```

```
C:\Users\DESKTOP\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pa
ss the following variable as a keyword arg: x. From version 0.12, the only valid position
al argument will be `data`, and passing other arguments without an explicit keyword will
result in an error or misinterpretation.
  warnings.warn(
```

Out[98]:

```
<AxesSubplot:xlabel='writing score', ylabel='count'>
```




## Conclusión:

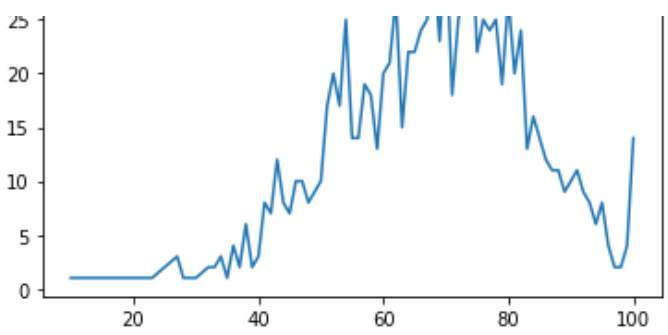
**Podemos observar de una mejor manera, aunque las etiquetas de abajo, que son las calificaciones, están empalmadas, podemos observar que el promedio si está entre 65 y 70.**

In [93]:

```
data_frame[data_frame['writingscore'] < 200]['writingscore'].value_counts().sort_index()
.plot.line()
```

Out[93]:

```
<AxesSubplot:>
```




## Conclusión:

**Podemos ver lo que antes hemos dicho sobre la media, y que gran parte de los datos está entre 60 y 80.**

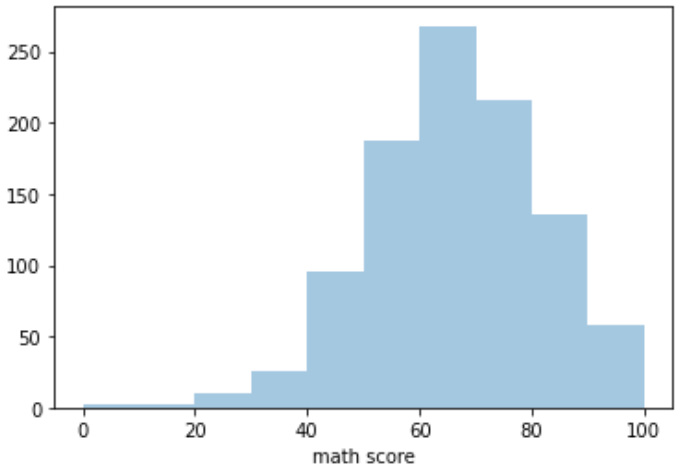
In [95]:

```
sns.distplot(data_frame['mathscore'], bins=10, kde=False)
```

```
C:\Users\DESKTOP\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning
: `distplot` is a deprecated function and will be removed in a future version. Please ada
pt your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[95]:

```
<AxesSubplot:xlabel='math score'>
```




## Conclusiones:

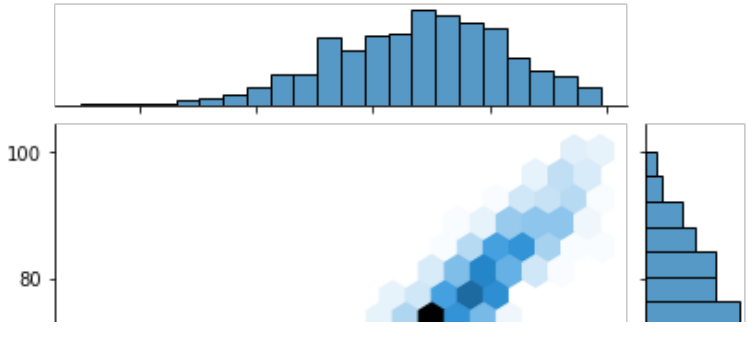
**Nos muestra lo mismo que el Histograma.**

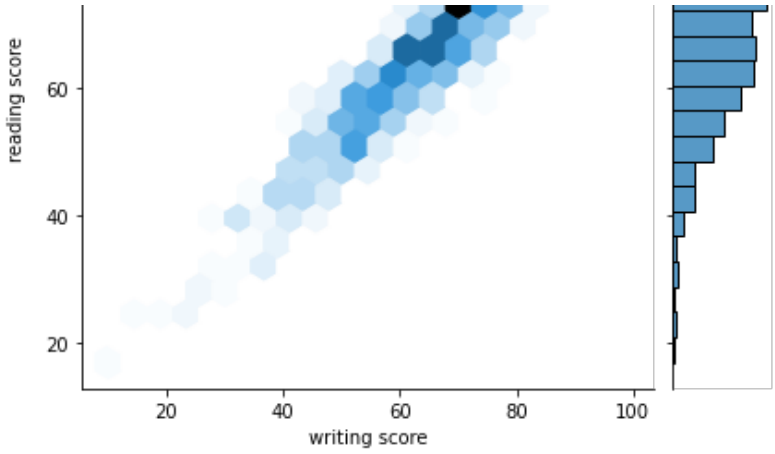
In [102]:

```
sns.jointplot(x='writingscore', y='reading score', data=data_frame[data_frame['writingsco
re'] < 100], kind='hex',
              gridsize=20)
```

Out[102]:

```
<seaborn.axisgrid.JointGrid at 0x29936900e50>
```






## Conclusiones:

**Nos representa donde se acumulan la mayor parte de los datos, en este caso, nuestra correlación sigue siendo la misma.**

## Box plot

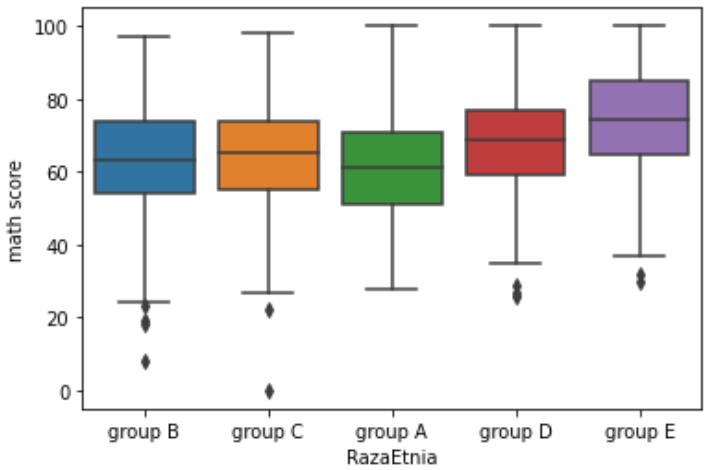
In [121]:

```
df = data_frame[data_frame.RazaEtnia.isin(data_frame.RazaEtnia.value_counts().head(5).index)]

sns.boxplot(
    x='RazaEtnia',
    y='mathscore',
    data=df
)
```

Out[121]:

```
<AxesSubplot:xlabel='RazaEtnia', ylabel='math score'>
```




## Conclusión:

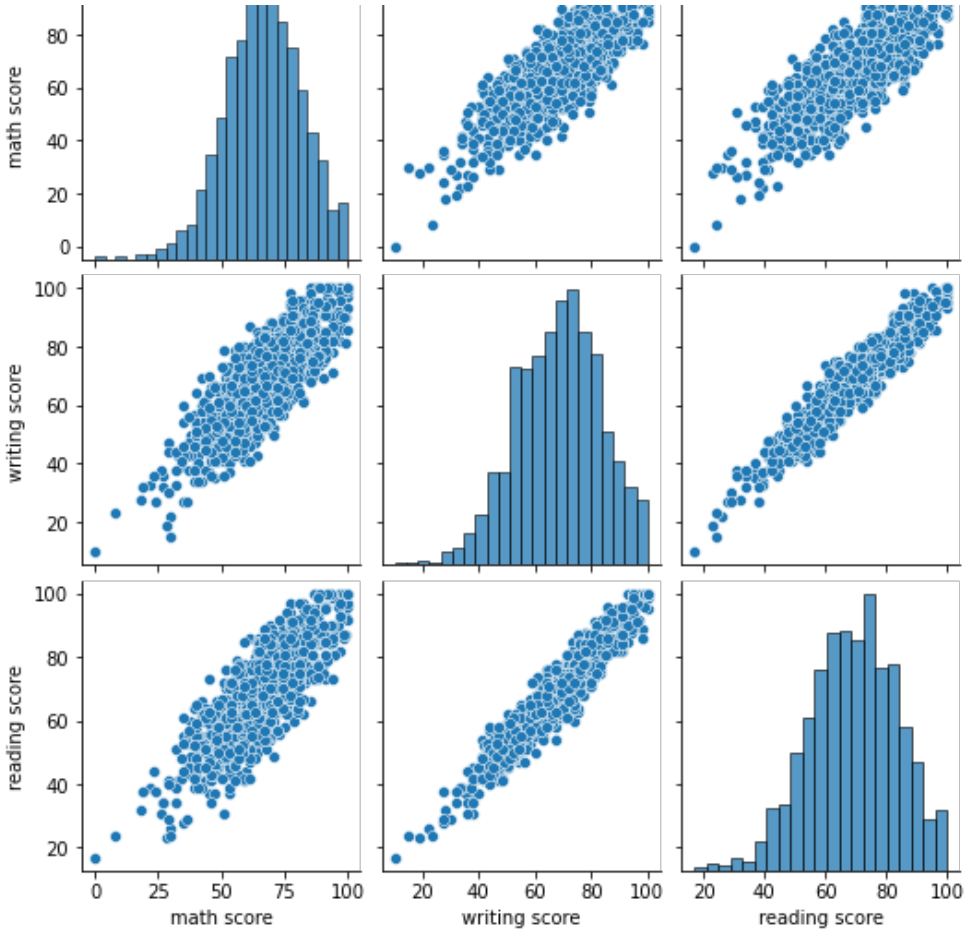
**Podemos ver donde está la mayor concentración de datos, en cuanto a calificaciones en la asignatura de matemáticas, hablamos, así como los puntos que están lejos.**

In [123]:

```
sns.pairplot(data_frame[['mathscore', 'writingscore', 'reading score']])
```

Out[123]:

```
<seaborn.axisgrid.PairGrid at 0x299382c6bb0>
```



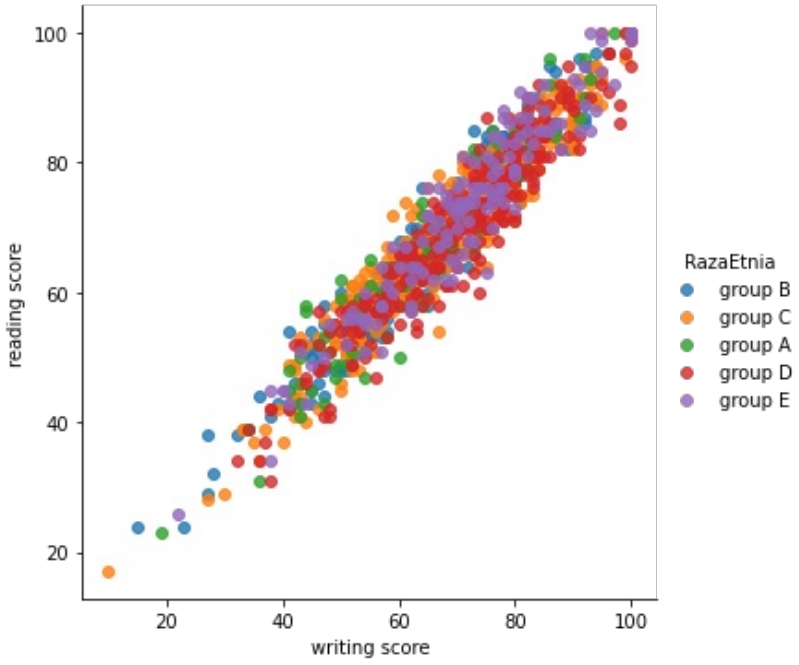

## Gráficas multivariantes

In [130]:

```
sns.lmplot(x='writingscore', y='reading score', hue='RazaEtnia',
data=data_frame.loc[data_frame['RazaEtnia'].isin(['group A', 'group B','group C','group D
','group E'])],
fit_reg=False)
```

Out[130]:

```
<seaborn.axisgrid.FacetGrid at 0x299389b9640>
```

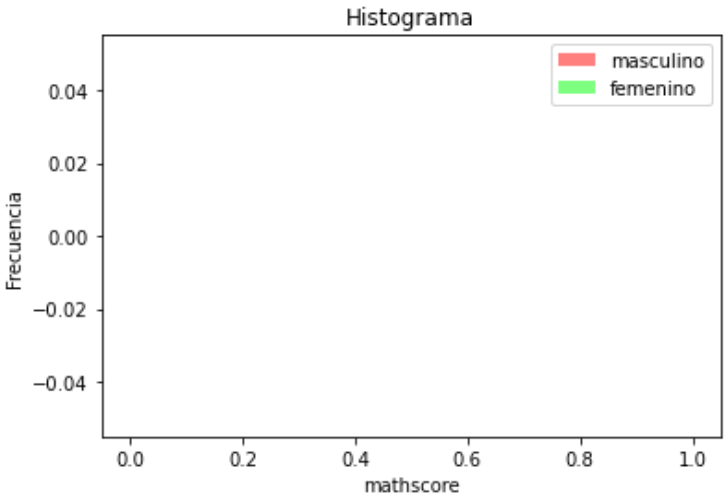



## Conclusión:

**Esta función nos permite observar que dato le pertenece a cada grupo etnico o raza.**

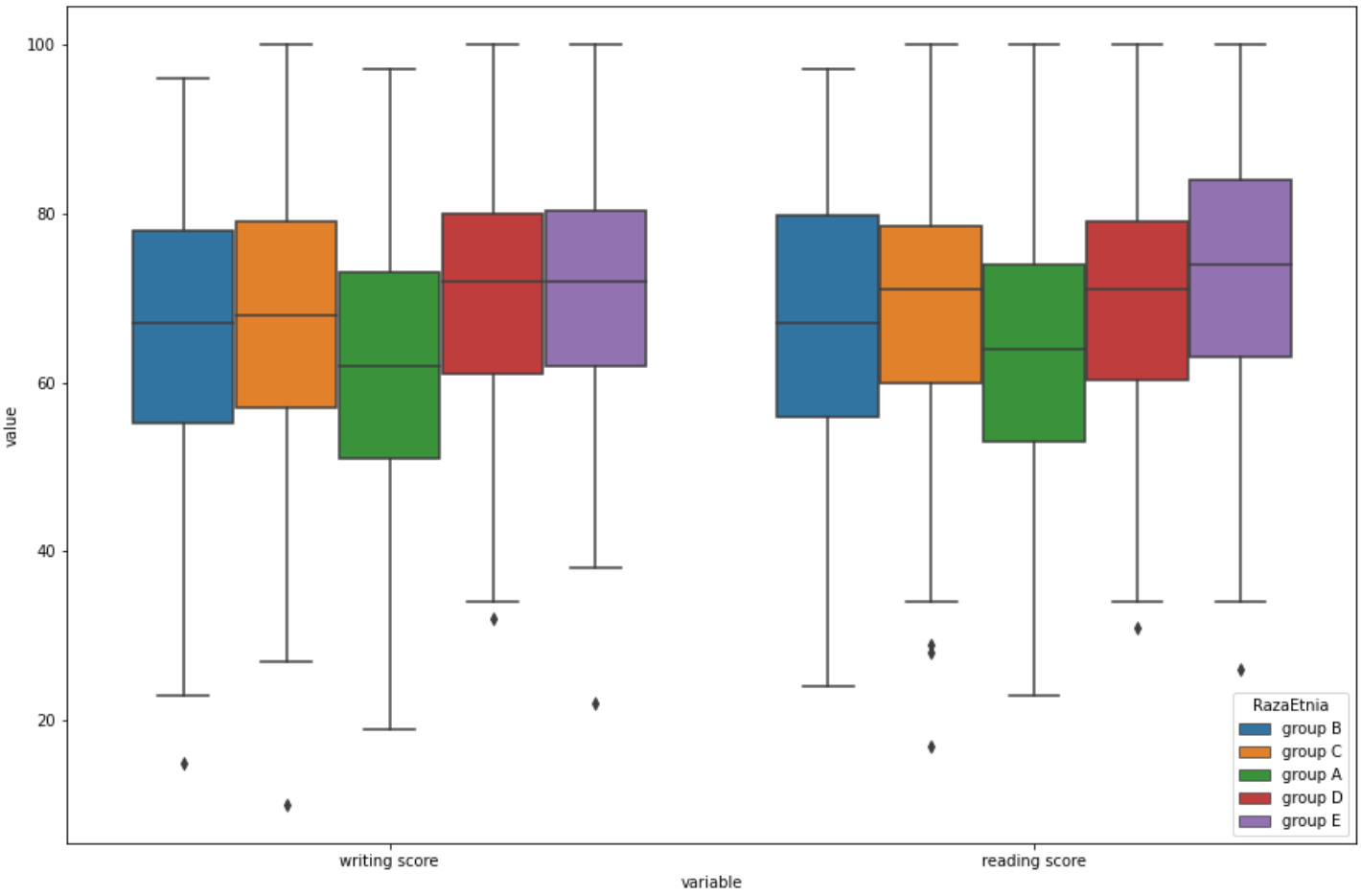
In [159]:

```
m = plt.hist(data_frame[data_frame["mathscore"] == "M"].mathscore,bins=30,fc = (1,0,0,0.
5),label = "masculino")
f = plt.hist(data_frame[data_frame["mathscore"] == "F"].mathscore,bins=30,fc = (0,1,0,0.
5),label = "femenino")
plt.legend()
plt.xlabel("mathscore")
plt.ylabel("frecuencia")
plt.title("Histograma")
plt.show()
```




In [142]:

```
mdata = pd.melt(df,id_vars = "RazaEtnia",value_vars = ['writing score', 'reading score']
)
plt.figure(figsize = (15,10))
sns.boxplot(x = "variable", y = "value", hue="RazaEtnia",data= mdata)
plt.show()
```



## Conclusión:

**Nos muestra la acumulación en dos variables a la vez.**

In [143]:

```
f,ax=plt.subplots(figsize = (18,18))
sns.heatmap(df.corr(),annot= True,linewidths=0.5,fmt = ".1f",ax=ax)
plt.xticks(rotation=90)
plt.yticks(rotation=0)
plt.title('Mapa de correlación')
plt.savefig('graph.png')
plt.show()
```