Análisis de bases de datos

Taller 4: Análisis multidimensional de criminalidad

Laura Nicole Bermudez Santa - labermudezs@unal.edu.co
David Sebastián Mendoza Cruz - damendozac@unal.edu.co
David Alejandro Alquichire Rincón - dalquichire@unal.edu.co
Laura Nicole Bermudez Santa - labermudezs@unal.edu.co
Juan David Bernal - jubernalv@unal.edu.co

Considere los datos del archivo Excel adjunto; en él se muestran los datos de criminalidad, fondos policiales y educación de la población en las ciudades pequeñas de los Estados Unidos.

Las variables **(X1, X2, X3, X4, X5, X6, X7)** representan la siguiente información:

- **X1:** reporte total de criminalidad por millón de residentes.
- **X2:** tasa de crímenes violentos por 100.000 residentes.
- **X3:** fondos anuales policiales en dólares por habitante.
- **X4:** porcentaje de personas de 25 años o más con bachillerato.
- **X5:** porcentaje de la población de 16 a 19 años sin bachillerato.
- **X6:** porcentaje de la población entre 18 a 24 años que realiza estudios universitarios.
- **X7:** porcentaje de la población de 25 o más años con por lo menos 4 años de estudios universitarios.

In [1]:
```
# Importamos algunas librerias necesarias
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb

# Leemos el archivo
criminalidad = pd.read_csv('Datos Criminalidad.csv')
criminalidad.head(5)
```

Out[1]:

	X1	X2	X3	X4	X5	X6	X7
0	478	184	40	74	11	31	20
1	494	213	32	72	11	43	18
2	643	347	57	70	18	16	16
3	341	565	31	71	11	25	19
4	773	327	67	72	9	29	24

In [2]:
```
criminalidad.info() # Información basica de las columnas que conforman el dataset
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 7 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   X1      50 non-null     int64
 1   X2      50 non-null     int64
 2   X3      50 non-null     int64
 3   X4      50 non-null     int64
 4   X5      50 non-null     int64
 5   X6      50 non-null     int64
 6   X7      50 non-null     int64
dtypes: int64(7)
memory usage: 2.9 KB
```

A. Presente un análisis estadístico básico por variable indicando sus opiniones sobre cada uno de los valores calculados. En este literal debe efectuar el cálculo de todas las medidas vistas en clase de centralización y dispersión, debe incluir un análisis intercuartílico.

In [3]:
```
criminalidad.describe() # Estadisticas basicas de las columnas que conforman el dat
```

Out[3]:

	X1	X2	X3	X4	X5	X6	X7
count	50.000000	50.000000	50.000000	50.000000	50.000000	50.000000	50.000000
mean	717.960000	616.180000	37.760000	58.800000	15.400000	29.900000	13.820000
std	293.938766	573.739175	13.820364	9.965246	6.023762	14.801062	5.157479
min	341.000000	29.000000	16.000000	42.000000	4.000000	7.000000	8.000000
25%	497.000000	230.750000	30.000000	49.000000	11.000000	21.250000	11.000000
50%	654.500000	454.000000	34.500000	59.000000	14.000000	25.000000	12.000000
75%	820.500000	822.500000	42.250000	67.000000	19.000000	34.250000	15.750000
max	1740.000000	3545.000000	86.000000	81.000000	34.000000	81.000000	36.000000

La tabla anterior presenta un análisis estadístico básico por cada variable contemplando los 50 registros del dataset. Se expone en su orden la cantidad de datos no nulos evaluados, la

media aritmética, la desviación estandar, el valor mínimo, primer y tercer cuartil, mediana y por ultimo el valor máximo. Algunas observaciones que podemos deducir del análisis estadístico basico son:

- El reporte total de criminalidad por millón de residentes (X1) tiene una media de 717.96, con un rango desde 341 hasta 1740. Lo cual nos indica que existe una variabilidad significativa en los niveles de criminalidad entre los reportes realizados.
- La tasa de crímenes violentos por 100,000 residentes (X2) tiene una media de 616.18, con una desviación estándar alta de 573.74, lo cual sugiere una gran dispersión en los datos, variando desde 29 hasta 3545.
- El porcentaje de personas de 25 años o más con bachillerato (X4) tiene una media de 58.8%, con un rango de 42% a 81%.
- El porcentaje de la población de 16 a 19 años sin bachillerato (X5) tiene una media de 15.4%, con un rango de 4% a 34%.
- El porcentaje de la población entre 18 a 24 años que realiza estudios universitarios (X6) tiene una media de 29.9%, con un rango de 7% a 81%.
- El porcentaje de la población de 25 o más años con al menos 4 años de estudios universitarios (X7) tiene una media de 13.82%, con un rango de 8% a 36%.

B. Presente una matriz de "calor" para las correlaciones (de Person) entre pares de variables. Indique cuales son los cuatro pares de variables con mayor correlación.

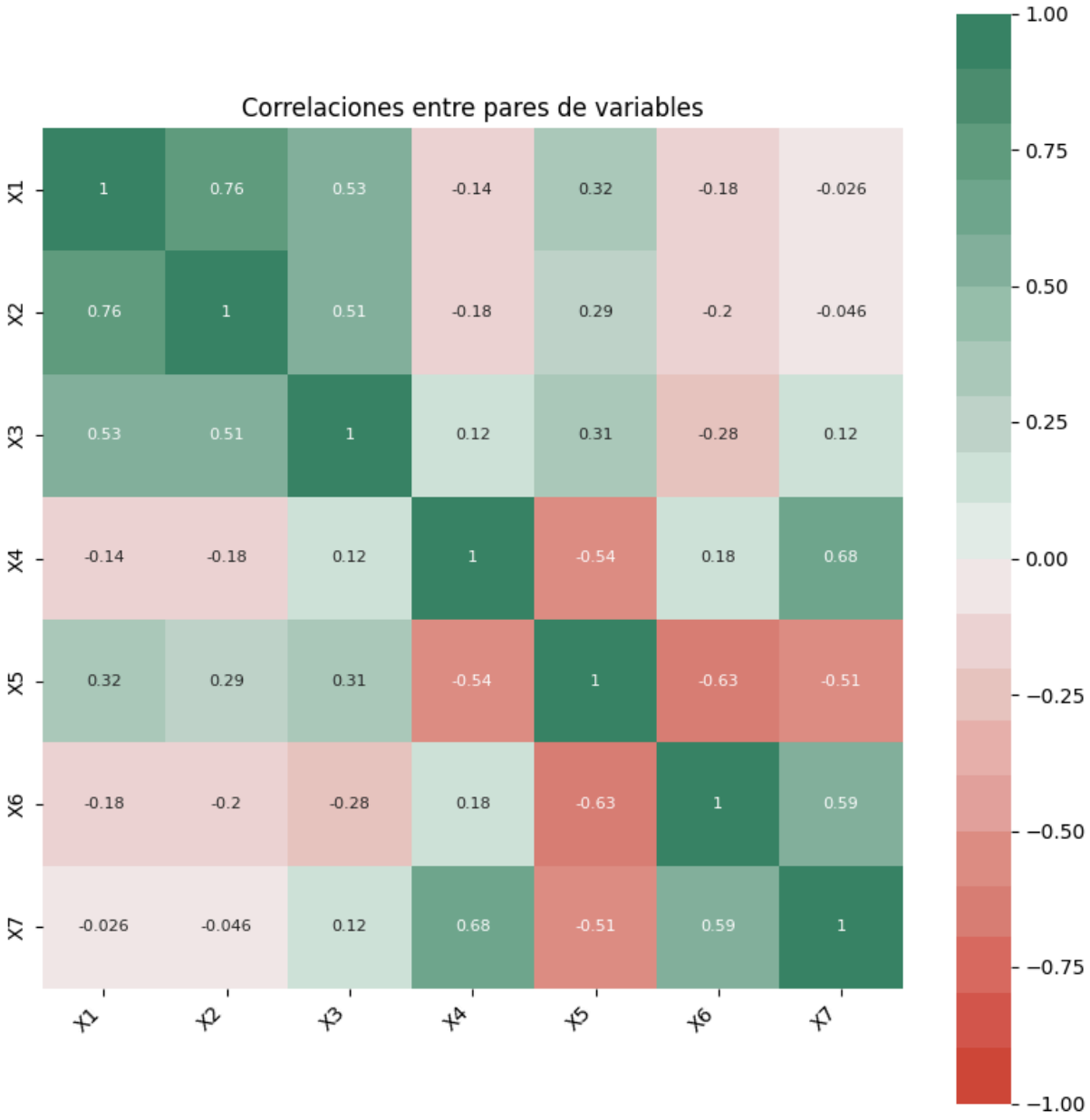
In [4]:
```
def Tabular_corr_matrix(corr_mat):
    '''
    Esta función presenta en forma tabular
    la correlación entre pares de columnas eliminando los
    pares iguales
    '''
    corr_mat = corr_mat.stack().reset_index()
    corr_mat.columns = ['variable_1','variable_2','correlación']
    corr_mat = corr_mat.loc[corr_mat['variable_1'] != corr_mat['variable_2'], :]
    corr_mat = corr_mat.sort_values('correlación', ascending=False)

    return(corr_mat)

corr_matrix = criminalidad.select_dtypes(include=['number']).corr(method='pearson')
Tabular_corr_matrix(corr_matrix).head(8)
```

Out[4]:

	variable_1	variable_2	correlación
1	X1	X2	0.756505
7	X2	X1	0.756505
45	X7	X4	0.681072
27	X4	X7	0.681072
47	X7	X6	0.591663
41	X6	X7	0.591663
2	X1	X3	0.533198
14	X3	X1	0.533198

In [5]:

```
import seaborn as sns
fig, ax = plt.subplots(nrows=1, ncols=1, figsize=(8, 8))

sns.heatmap(
    corr_matrix,
    annot=True,
    cbar=True,
    annot_kws={"size": 8},
    vmin=-1,
    vmax=1,
    center=0,
    cmap=sns.diverging_palette(15, 150, n=20),
    square=True,
    ax=ax
)

ax.set_xticklabels(
    ax.get_xticklabels(),
    rotation=45,
    horizontalalignment='right',
)

ax.tick_params(labelsize=10)
ax.set_title("Correlaciones entre pares de variables")
plt.tight_layout()
plt.show()
```



In [6]:
```
tab_corr = Tabular_corr_matrix(corr_matrix)

selected_pairs = set()
final_pairs = []

for _, row in tab_corr.iterrows():
    var1, var2 = row.variable_1, row.variable_2

    # Evitar agregar el mismo par en orden inverso
    if (var1, var2) not in selected_pairs and (var2, var1) not in selected_pairs:
        selected_pairs.add((var1, var2))
        final_pairs.append((var1, var2, row.correlación))

    if len(final_pairs) == 4:
        break
print(selected_pairs)
```

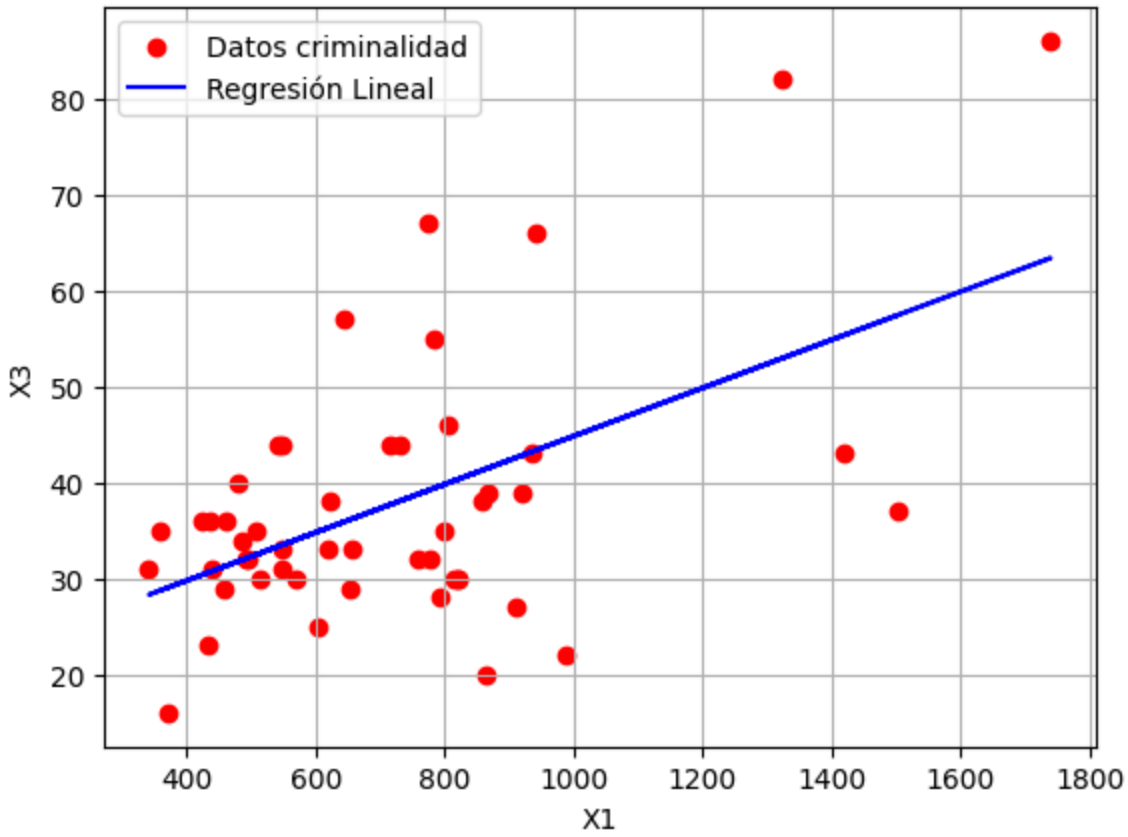
```
{('X1', 'X2'), ('X7', 'X4'), ('X7', 'X6'), ('X1', 'X3')}
```

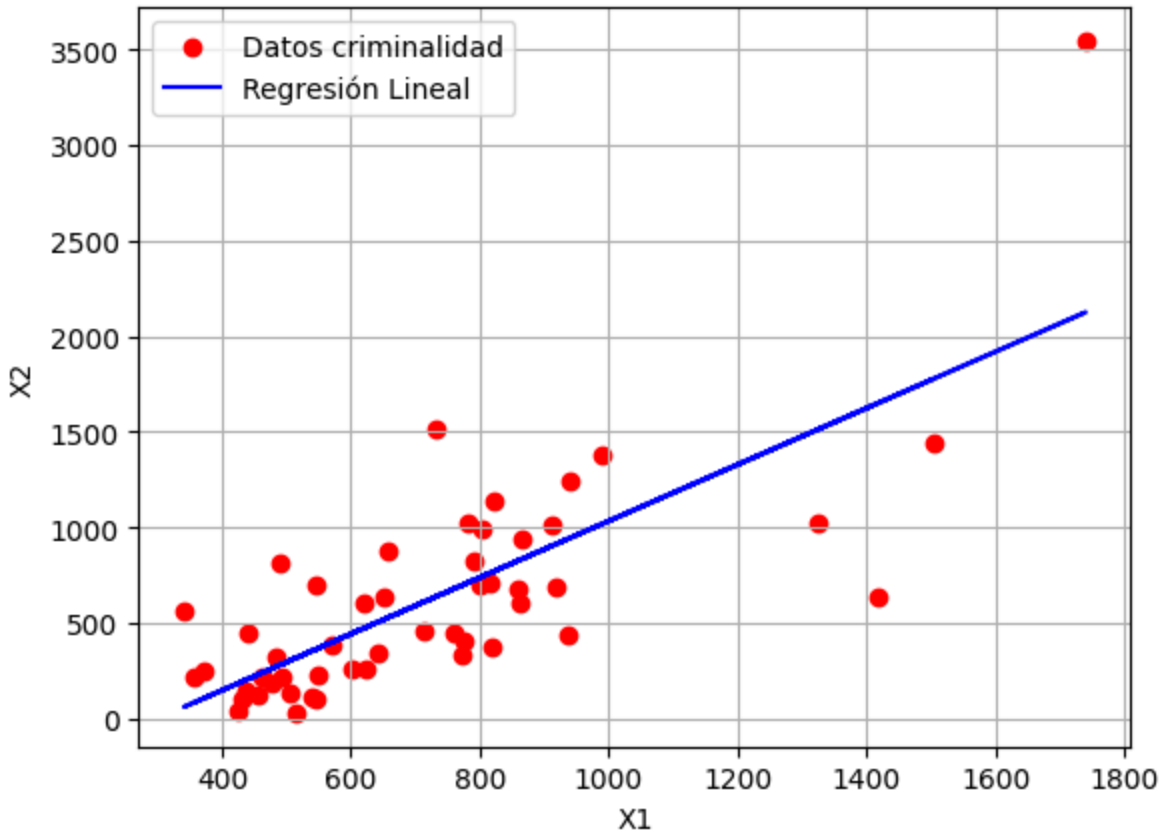
Vemos que las variables con mayor correlación son:

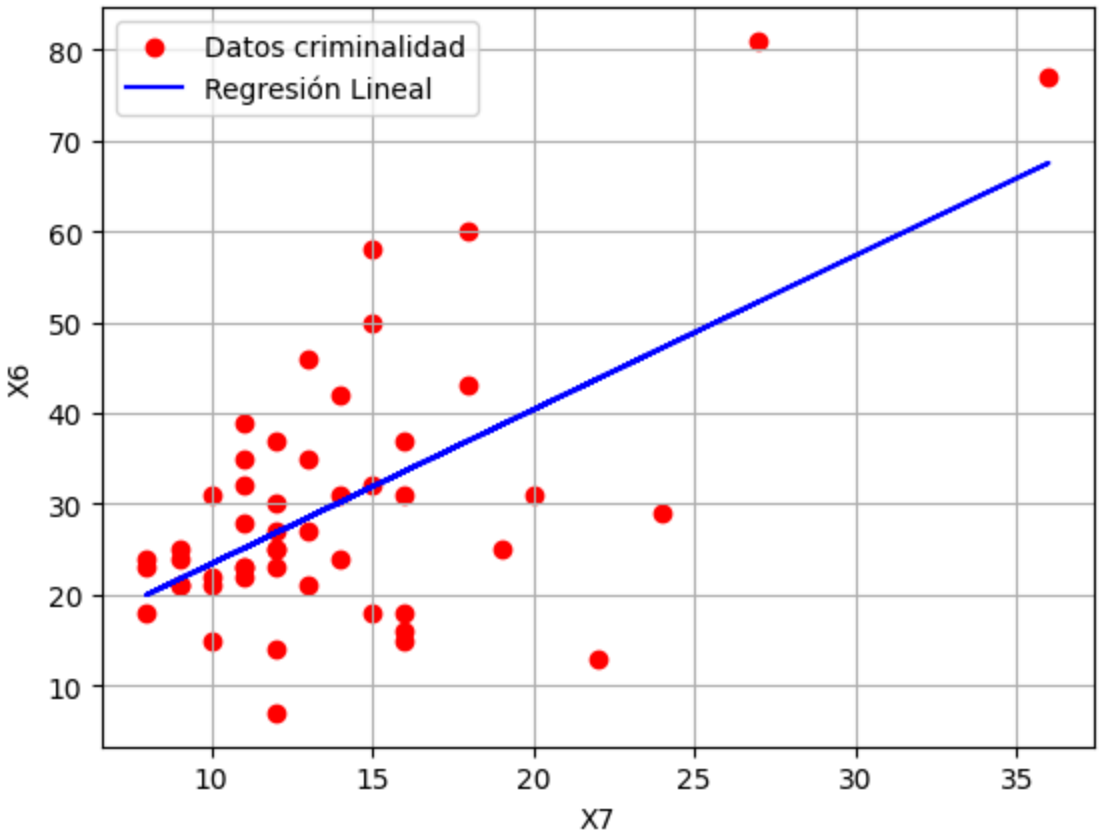
1. $(X1, X3)$
2. $(X1, X2)$
3. $(X7, X6)$
4. $(X7, X4)$

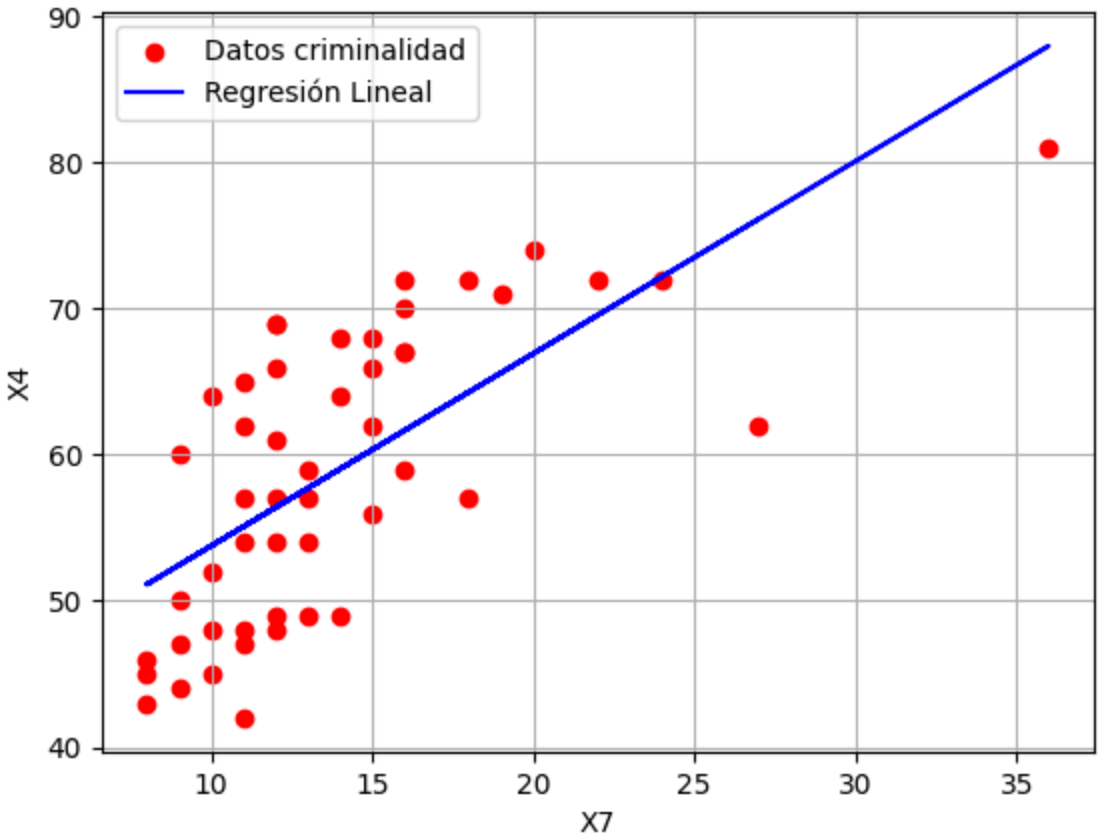
C. Plotee los cuatro pares de variables con mayor correlación, mostrando la recta de regresión lineal que mejor se ajusta a los datos.

In [11]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

def linear_regression(X, Y, xlabel, ylabel):
    a = np.inner(X, X)
    b = np.sum(X)
    c = np.inner(X, Y)
    d = np.sum(Y)
    N = len(X)
    Delta = a * N - b * b
    A = (N * c - b * d) / Delta
    B = (a * d - b * c) / Delta
    Y_pred = A * X + B

    plt.grid(True)
    plt.scatter(X, Y, color='red', label='Datos criminalidad')
    plt.plot(X, Y_pred, color='blue', label='Regresión Lineal')
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    plt.legend()
    plt.show()

linear_regression(criminalidad['X1'], criminalidad['X3'], 'X1', 'X3')
linear_regression(criminalidad['X1'], criminalidad['X2'], 'X1', 'X2')
linear_regression(criminalidad['X7'], criminalidad['X6'], 'X7', 'X6')
linear_regression(criminalidad['X7'], criminalidad['X4'], 'X7', 'X4')
```


Datos criminalidad
Regresión Lineal
X3
X1
20
30
40
50
60
70
80
400
600
800
1000
1200
1400
1600
1800


Datos criminalidad
Regresión Lineal
X2
X1
0
500
1000
1500
2000
2500
3000
3500
400
600
800
1000
1200
1400
1600
1800





D. Haga un análisis multilineal tomando como variable dependiente los fondos anuales policiales por habitante, con las demás variables como variables predictorias. Indique la expresión que obtuvo.

Calcularemos los coeficientes de regresión $\beta_i; 0 \leq i \leq 6$ tales que permitan predecir las ventas mediante una igualdad de la forma:

$$\mathrm{X3}_{pred} = \beta_0 + \beta_1 \mathrm{X1} + \beta_2 \mathrm{X2} + \beta_3 \mathrm{X4} + \beta_4 \mathrm{X5} + \beta_5 \mathrm{X6} + \beta_6 \mathrm{X7}$$

In [8]:
```
X = pd.DataFrame(criminalidad[['X1','X2', 'X4', 'X5', 'X6', 'X7']]).values
Y = pd.DataFrame(criminalidad[['X3']]).values
#Generamos una columna de unos y la insertamos como primera columna de X
Unos = np.ones(len(criminalidad))
X = np.insert(X, 0, Unos, axis=1)
MPenrouse = np.linalg.pinv(np.matmul(X.transpose(),X)) # Cálculo de la seudo-invers
C = np.matmul(MPenrouse,X.transpose())
B = np.matmul(C,Y)
print(B)
```

```
[[-4.86463516]
 [ 0.01228691]
 [ 0.0057846 ]
 [ 0.27990054]
 [ 0.62677114]
 [-0.19446349]
 [ 0.71945018]]
```

Hemos obtenido la expresión

$$X3_{pred} = -4.864 + 0.012 \cdot X1 + 0.005 \cdot X2 + 0.279 \cdot X4 + 0.626 \cdot X5 - 0.194 \cdot X6 + 0.7$$

E. Indique que variables predictorias tiene mayor impacto sobre la variable independinte. ¿Tiene sentido lo obtenido? Explique.

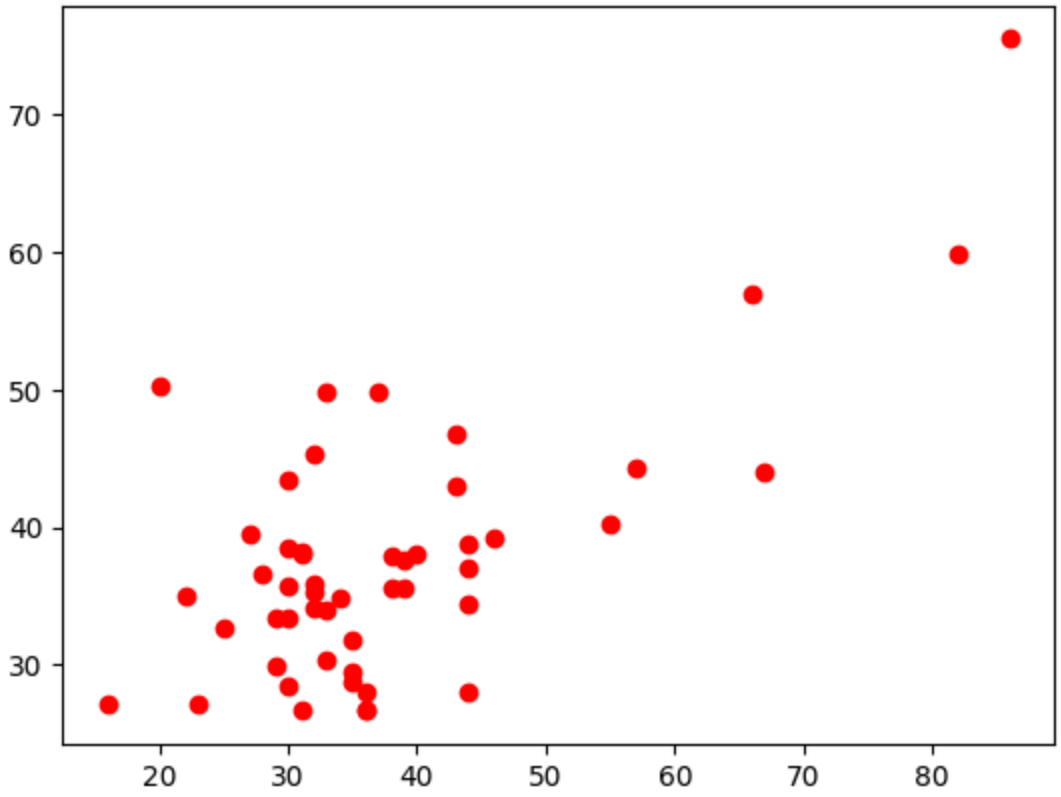
Solución:

Vemos de lo anterior que las variables tienen impacto en el siguiente orden de mayor a menor sobre la independiente:

1. **X7:** Porcentaje de la población de 25 o más años con por lo menos 4 años de estudios universitarios.
2. **X5:** Porcentaje de la población de 16 a 19 años sin bachillerato.
3. **X4:** Porcentaje de personas de 25 años o más con bachillerato.
4. **X6:** Porcentaje de la población entre 18 a 24 años que realiza estudios universitarios.
5. **X1:** Reporte total de criminalidad por millón de residentes.
6. **X2:** Tasa de crímenes violentos por 100.000 residentes.

In [9]:
```
from numpy import linalg as LA
Y_pred = np.matmul(X,B)
plt.scatter(Y,Y_pred, color='red')
plt.show()
P_int=np.matmul(np.transpose(Y),Y_pred)
Norm_Y=LA.norm(Y)
```

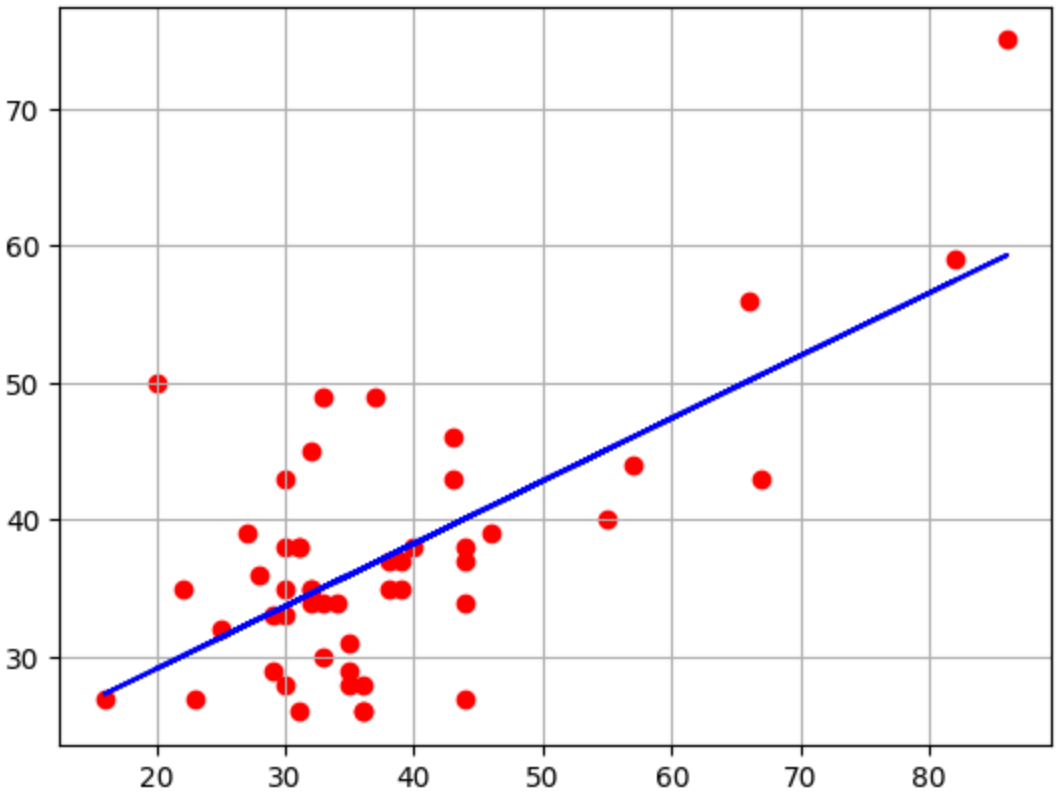
```
Norm_Yp=LA.norm(Y_pred)
c = P_int/(Norm_Y*Norm_Yp)
print("Coeficiente de correlación = ",c)
```




```
Coeficiente de correlación =  [[0.96829506]]
```

F. Haga un análisis de regresión lineal simple entre la variable fondos anuales reales y los predichos por el modelo. Con base en este análisis evalúe el modelo.

In [10]:
```
Y_aux=np.array([int(x[0]) for x in Y])
Y_pred_aux=np.array([int(x[0]) for x in Y_pred])
linear_regression(Y_aux, Y_pred_aux)
```



Podemos observar varias cosas, en primer lugar antes de hacer la regresión hemos sacado el coeficiente de correlación entre la variable predicha y la real; podemos observar que este dio bastante cercano a 1, indicando así que el modelo se ajusta bien a los datos. Pero al observar los puntos graficados notaremos que para valores más grandes (mayores a 50), nuestra predicción tiende a estar más alejada del dato real, tendiendo a estar por debajo. Esto puede ser a la falta de más datos en ese rango, pues notamos que la mayoría está concentrado por debajo de 50, haciendo así más difícil la aproximación para valores altos.

Al hacer la regresión lineal entre los datos reales y los predichos podemos observar como la pendiente de la recta está visiblemente desviada de 1, por los datos mayores a 50 que hemos mencionado antes.

De lo anterior podemos concluir que aunque el modelo puede estar bien ajustado para valores pequeños, puede que para valores más grandes no sea fiable, y se requieran más observaciones en esas condiciones para refinar el modelo.

G. Plantee un modelo de regresión multilineal que permita explicar los índices de criminalidad con el grado de escolaridad de la población. Explique brevemente su modelo y los resultados que obtiene de él.

Primero, seleccionamos los datos y en este caso consideramos elsiguiente modelo que utiliza mínimos cuadrados para encontrar la mejor línea de ajuste. Se basa en la ecuación $\beta = (X^T X)^+ X^T Y$, donde X es la matriz de características y Y el vector de valores objetivo. En lugar de invertir $X^T X$, usamos su pseudo-inversa de Moore-Penrose. Así, el

modelo minimiza el error cuadrático entre las predicciones y los valores reales, ajustando los coeficientes de la regresión de manera óptima.

In [12]:
```
X = pd.DataFrame(criminalidad[['X4', 'X5', 'X6', 'X7']]).values # Indices de escola
Y = pd.DataFrame(criminalidad['X1']).values # Índices de Criminalidad

ones_ = np.ones( len(criminalidad) )
X = np.insert(X, 0, ones_, axis=1) # Tener modelo b0 + b1 x + b2 y + b3 z + b4 w
```

In [13]:
```
M_Penrose = np.linalg.pinv(np.dot(X.T, X))
C = np.matmul(M_Penrose, X.T)
sol_least_square = np.matmul(C, Y)
```

In [14]:
```
sol_least_square
```

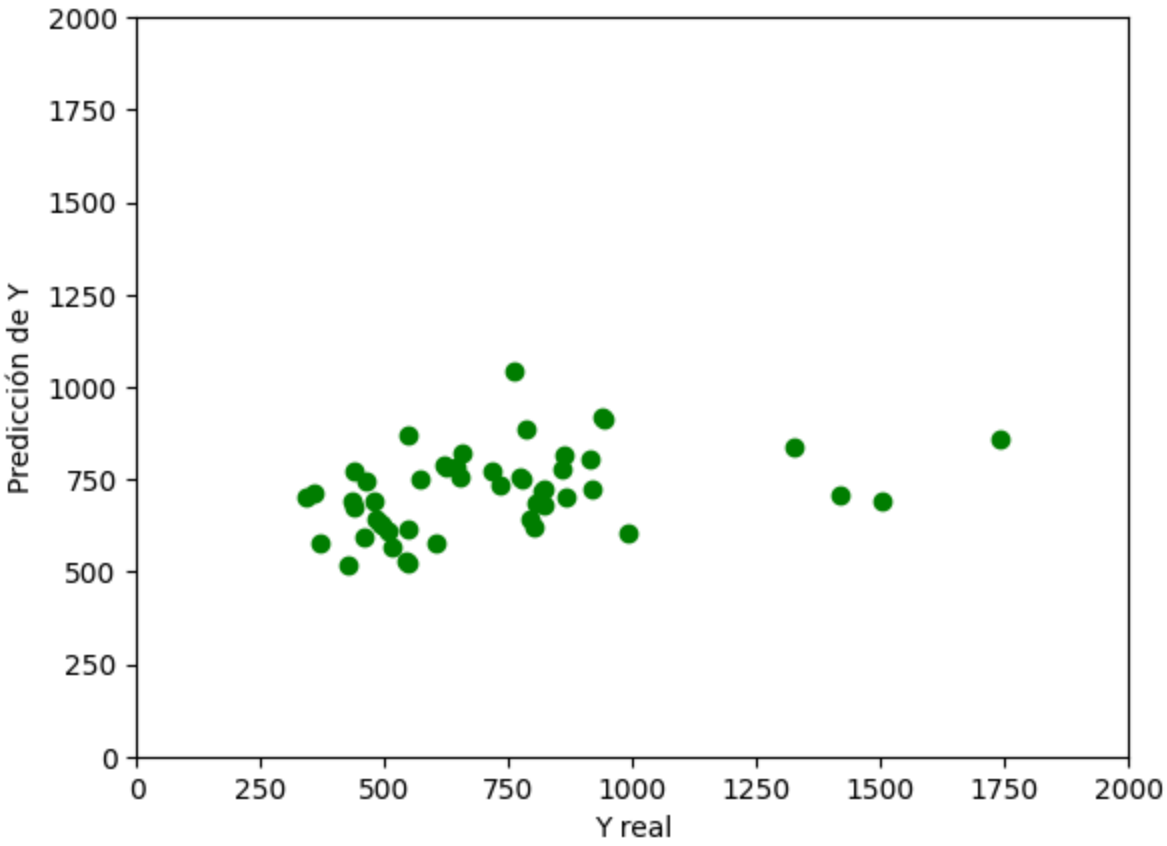
Out[14]:
```
array([[606.71367832],
       [ -5.10237042],
       [ 15.12460798],
       [ -2.98718625],
       [ 19.36784431]])
```

Por ende con los valores obtenidos tenemos el siguiente modelo de regresión multilineal:

$$y = 606.714 - 5.1024x_1 + 15.1246x_2 - 2.9871x_3 + 19.3678x_4$$

A continuación graficamos la correlación entre Y real y la predicción de Y realizada. Con su respectivo valor de correlación.

In [15]:
```
import matplotlib.pyplot as plt

Y_pred = np.matmul(X, sol_least_square)
plt.scatter(Y, Y_pred, c='green')
plt.xlabel('Y real')
plt.ylabel('Predicción de Y')
plt.xlim([0,2000])
plt.ylim([0,2000])
plt.show()
```



In [16]:
```
P_int = np.matmul(np.transpose(Y),Y_pred)
Norm_Y = np.linalg.norm(Y)
Norm_Yp = np.linalg.norm(Y_pred)
c = P_int/(Norm_Y*Norm_Yp)
print("Coeficiente de correlación = ",c)
```

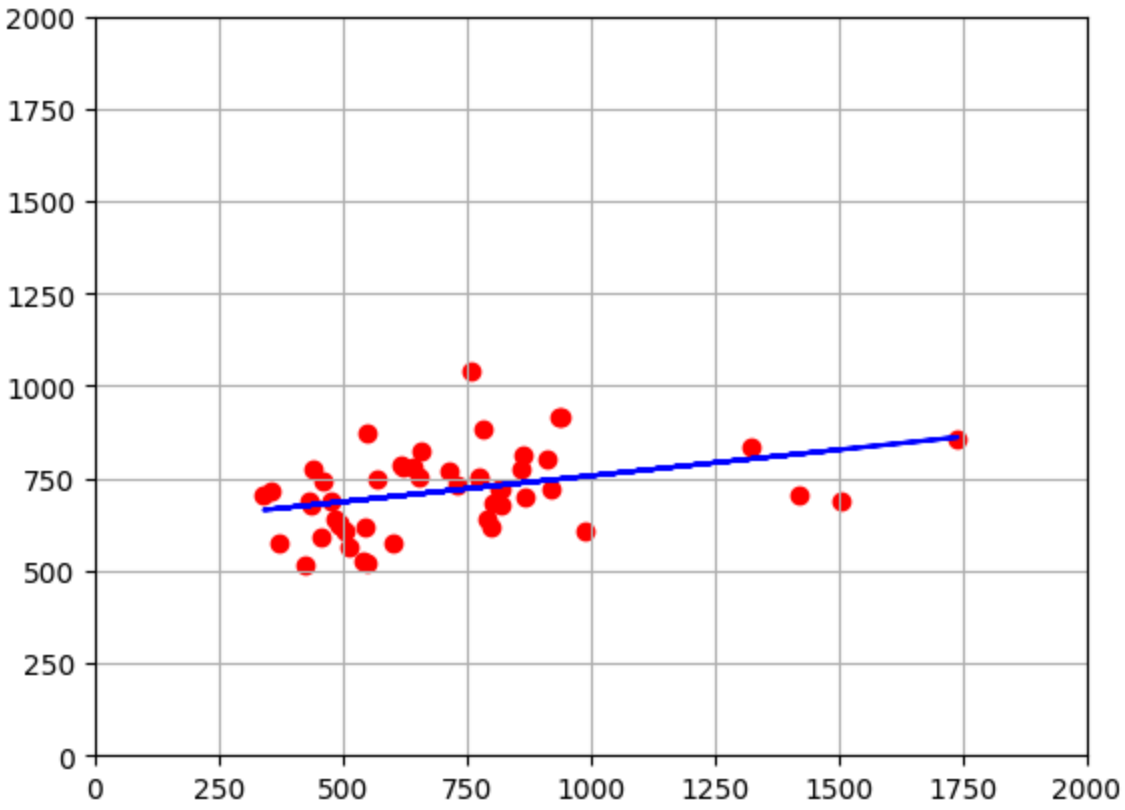
```
Coeficiente de correlación =  [[0.93743314]]
```

Como primeras impresiones, vemos en la gráfica que los datos entre Y real y Y predicha coinciden un poco, y estas forman ligeramente una relación lineal, salvo unos datos atípicos que vemos, que en Y real son mayores a 1250.

Como el coeficiente es 0.93, podemos confirmar que la relación lineal es buena.

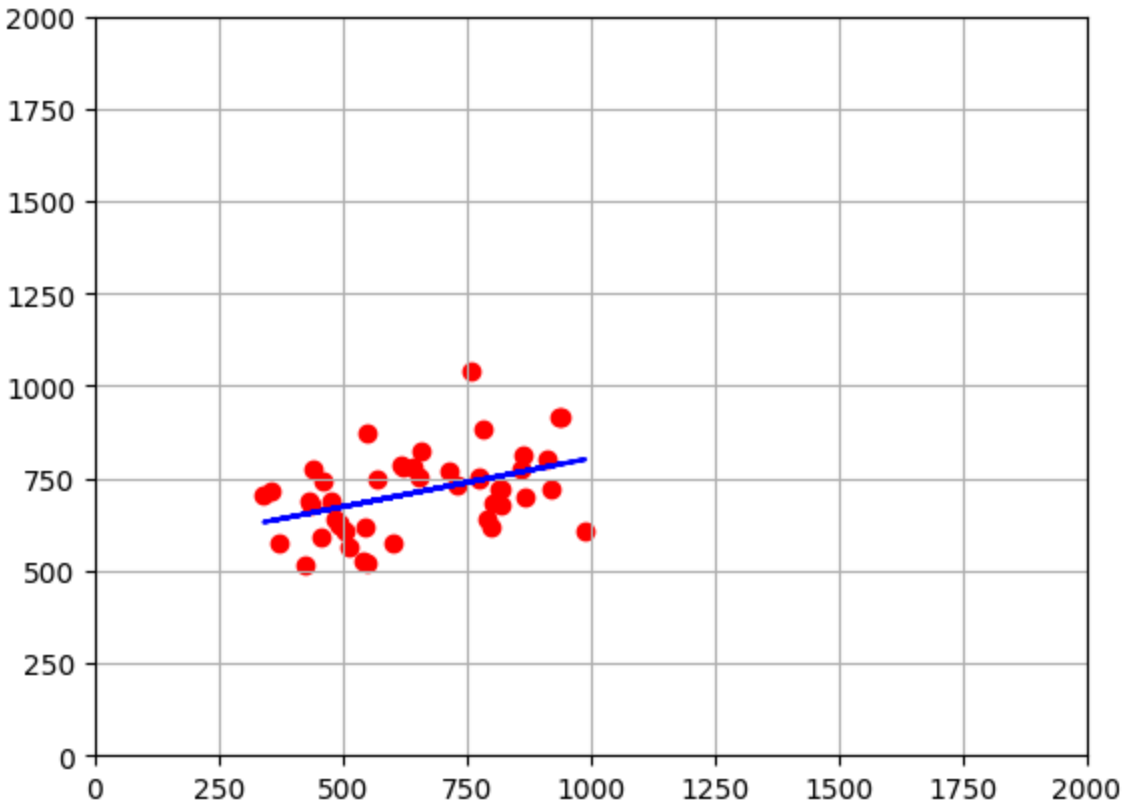
In [17]:
```
def linear_regression(X, Y):
    a = np.inner(X,X)
    b = np.sum(X)
    c = np.inner(X,Y)
    d = np.sum(Y)
    N = len(X)
    Delta = a*N-b*b
    A =(N*c-b*d)/Delta
    B =(a*d-b*c)/Delta
    Y_pred = A*X+B
    plt.grid(True)
    plt.scatter(X, Y,color='red')
    plt.plot(X, Y_pred, color='blue')
    plt.xlim([0,2000])
```

```
    plt.ylim([0,2000])
    plt.show()

Y_aux=np.array([int(x[0]) for x in Y])
Y_pred_aux=np.array([int(x[0]) for x in Y_pred])
linear_regression(Y_aux, Y_pred_aux)
```



Vemos que la regresión lineal es relativamente buena. Si eliminaramos los datos atípicos, la regresión sería un poco distinta.

In [19]:
```
new_Y_aux = Y_aux[ Y_aux < 1250]
new_Y_pred_aux = Y_pred_aux[ Y_aux < 1250]
linear_regression(new_Y_aux, new_Y_pred_aux)
```



Al eliminar los datos atípicos, la regresión comienza a ser un poco más inclinada. Por tanto, existe mayor correlación en este caso respecto a la variable real respecto a la variable predicha.

H. ¿Es posible detectar datos atípicos en la base de datos con las herramientas vistas en clase?

Según (Devore, 2012), la detección de datos atípicos en una muestra, se hace con ciertas cuentas, usando los cuartiles.

En clase, vimos la definición de cuartil, por ende, sí es posible detectar datos atípicos. Un dato q es atípico si cumple las siguientes desigualdades:

$$q < Q_1 - 1.5 \cdot (Q_3 - Q_1) \text{ o } q > Q_1 + 1.5 \cdot (Q_3 - Q_1)$$

Podemos verificar respecto a la base de datos:

In [21]:
```
Q_1 = criminalidad.quantile(0.25)
Q_2 = criminalidad.quantile(0.50)
Q_3 = criminalidad.quantile(0.75)
IQR = Q_3 - Q_1 # El rango intercuartílico
```

Podemos ir verificando por columna los datos atípicos.

In [24]:
```
criminalidad[criminalidad < Q_1 - 1.5 * IQR].head(9)
```

Out[24]:

	X1	X2	X3	X4	X5	X6	X7
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5	NaN	NaN	NaN	NaN	NaN	NaN	NaN
6	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7	NaN	NaN	NaN	NaN	NaN	NaN	NaN
8	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Por ejemplo, en este caso no hay datos atípicos menores. Pero la siguiente si muestra datos atípicos por encima de los datos 'comunes'.

In [25]:
```
criminalidad[criminalidad > Q_1 + 1.5 * IQR].head(9)
```

Out[25]:

	X1	X2	X3	X4	X5	X6	X7
0	NaN	NaN	NaN	NaN	NaN	NaN	20.0
1	NaN	NaN	NaN	NaN	NaN	43.0	NaN
2	NaN	NaN	57.0	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	NaN	NaN	19.0
4	NaN	NaN	67.0	NaN	NaN	NaN	24.0
5	NaN	NaN	NaN	NaN	NaN	NaN	NaN
6	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7	NaN	NaN	NaN	NaN	NaN	NaN	NaN
8	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Bibliografía

Devore, J. L. (2012). Probabilidad y Estadistica para Ingenieria y Ciencias. https://openlibrary.org/books/OL26233421M/Probabilidad_y_estad%C3%ADstica_para_ingenier9