Sistemas de Información y Telemedicina. [*]

Marta Girones Sanguesa
Silvia Marset Gomis
Ignacio Amat Hernández
Sofía Gutiérrez Santamaría

November 25, 2019

1 Preámbulo

```
# names of variables
labels = ['age', 'leptin', 'bmi', 'adiponectin', 'glucose',
        'resistin', 'insulin', 'MCP1', 'HOMA']

# loads data
data = np.loadtxt (open (r'../../data.csv', 'rb'), delimiter = ',')

# rewrites data as all the rows of data w/out nan cells
data = data [~np.isnan (data).any (axis=1)]

# separates parameters into matrix x
x    = np.array ([list (data [x][:-1])    for x in range (len (data))])

# and class (1, 2) into vector y
y    = np.array ([int  (data [x][ -1])    for x in range (len (data))])

# removes outliers
data_no = data [(np.abs (stats.zscore (data)) < 3).all (axis = 1)]
#   ↑ = No Outliers

x_no = np.array ([list (data_no [x][:-1]) for x in range (len (data_no))])
y_no = np.array ([int  (data_no [x][ -1]) for x in range (len (data_no))])
```

Listing 1: Importaciones iniciales y preparacion de datos.

[*] Grado en Ingeniería Biomédica, Escuela Técnica Superior de Ingenieros Industriales, Valencia, España.

2 Histogramas

En este apartado dibujamos los histogramas comparativos.

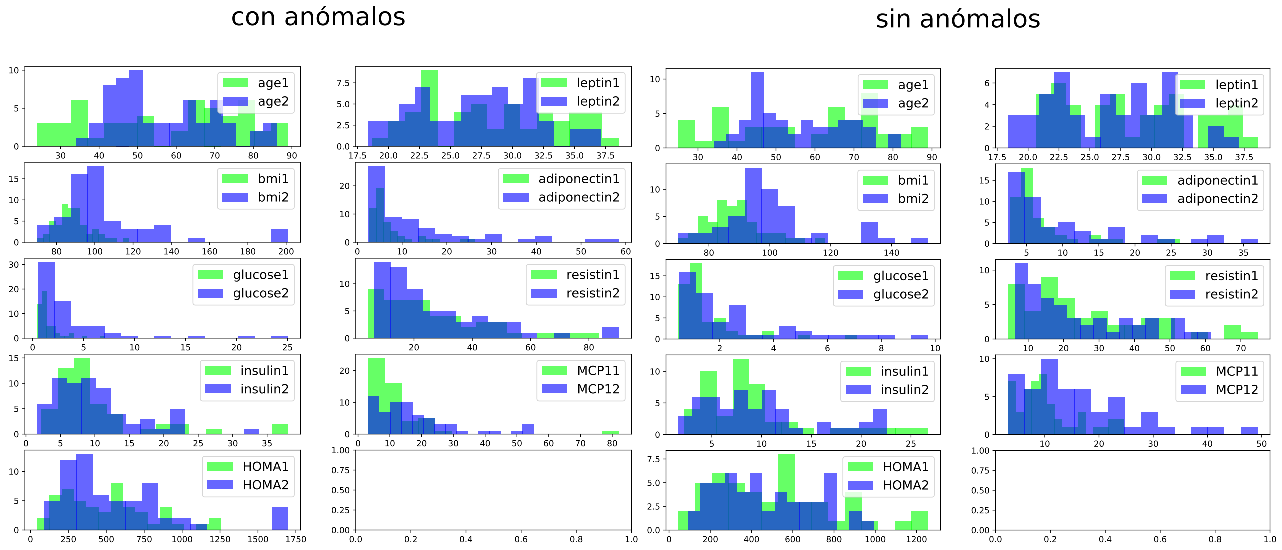

Fig. 1: Histogramas para datos con y sin anomalias.

```
import matplotlib as mpl
import matplotlib.pyplot as plt

#  load preprocessed data, x and y are raw, x_no and y_no contain no outliers
from preprocessing import x, y, x_no, y_no, labels

#  colours for the histograms
fc = [(), (0, 1, 0, 0.6), (0, 0, 1, 0.6)]
#         (R, G, B,  α )← transparency

fig, ax = plt.subplots (nrows = 5, ncols = 2, figsize = (13, 10))
ax = ax.flatten ()

#  draws each of the histograms, two for each variable
for i in range (0, 9):
    for j in [1, 2]:
        ax[i].hist (x [y == j, i], bins = 15, fc = fc [j], label = labels [i] + str (j))
        ax[i].legend (loc = 1, prop={'size': 15})

fig.suptitle ('con anómalos', fontsize = 30)
fig.savefig  ('../images/hist.pdf', bbox_inches = 'tight', pad_inches = 0)
```

Listing 2: Código generador de los histogramas con datos anómalos.

3 Kernel Density

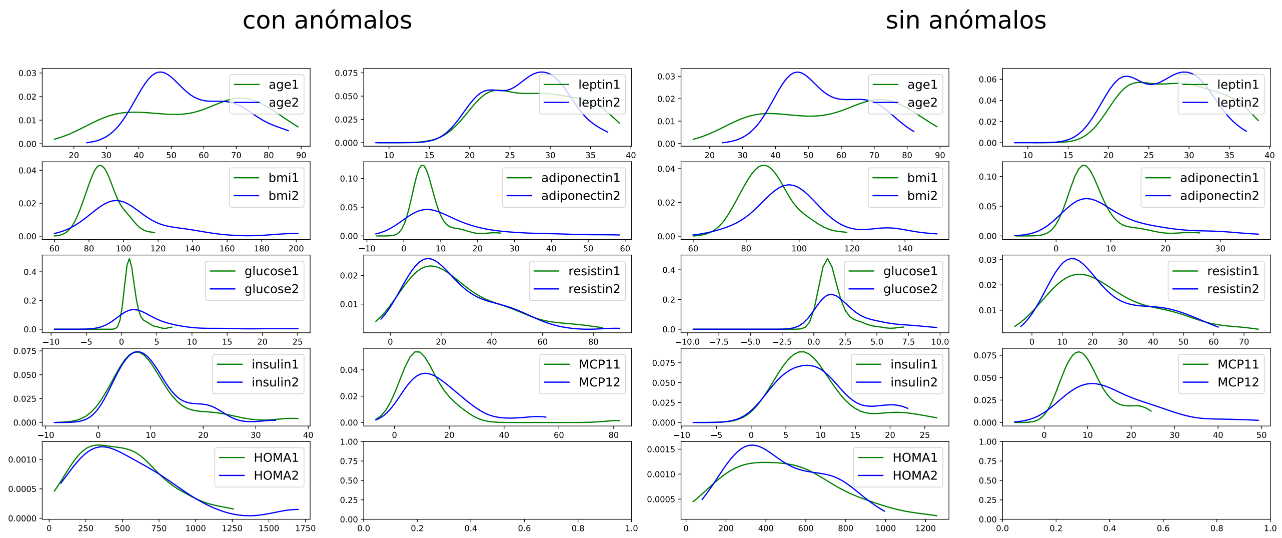

Fig. 2: Kernel Density para datos con y sin anomalias.

```
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np
from scipy.stats import gaussian_kde

#  load preprocessed data, x and y are raw, x_no and y_no contain no outliers
from preprocessing import x, y, x_no, y_no, labels

#  colours
fc = [’’, ’green’, ’blue’]

fig, ax = plt.subplots (nrows = 5, ncols = 2, figsize = (13, 10))
ax = ax.flatten ()

#  same loop in principle as before
for i in range (0, 9):
    for j in [1, 2]:
        kde = gaussian_kde (x_ := x [y == j, i])
        xs = np.linspace(np.min (x_) - 10, np.max (x_), num=len (x_))
        ax[i].plot (xs, kde(xs), c = fc[j], label = labels [i] + str (j))
        ax[i].legend (loc = 1, prop={’size’: 15})

fig.suptitle (’con anómalos’, fontsize = 30)
fig.savefig  (’../images/kden.pdf’, bbox_inches = ’tight’, pad_inches = 0)
```

Listing 3: Código generador de los kernel density plots con datos anómalos.

4 Boxplot

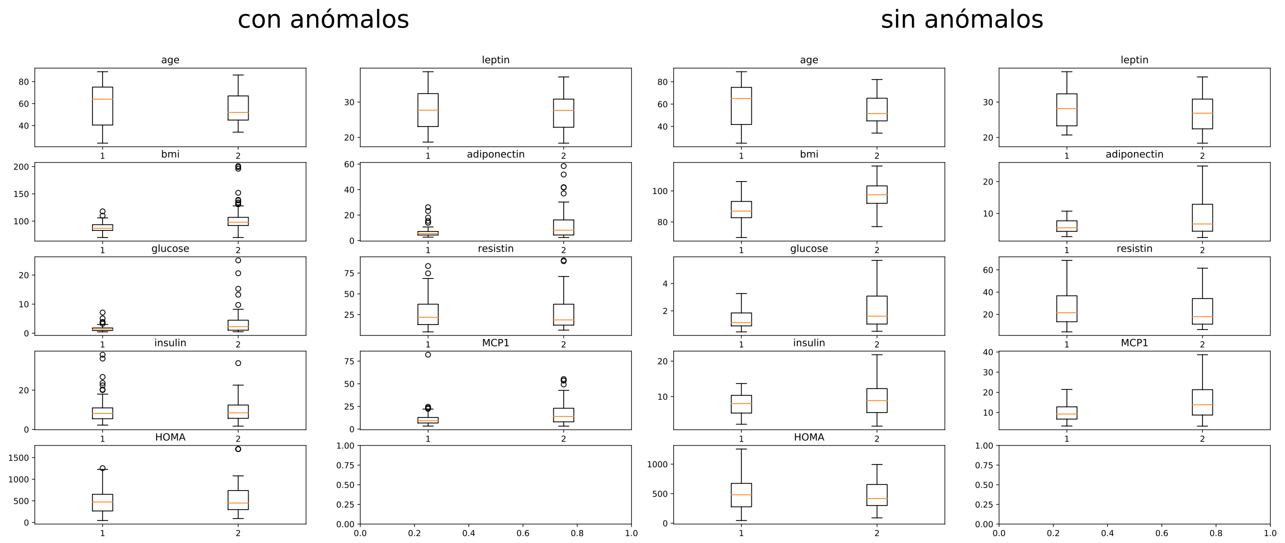

Fig. 3: Boxplots para datos con y sin anomalias.

```
import matplotlib as mpl
import matplotlib.pyplot as plt

# load preprocessed data, x and y are raw, x_no and y_no contain no outliers
from preprocessing import x, y, x_no, y_no, labels

fig, ax = plt.subplots (nrows = 5, ncols = 2, figsize = (13, 10))
ax = ax.flatten ()

for i in range (0, 9):
    ax[i].boxplot ([x [y == 1, i], x [y == 2, i]])
    ax[i].title.set_text (labels [i])

fig.suptitle ('con anómalos', fontsize = 30)
fig.savefig  ('../images/boxp.pdf', bbox_inches = 'tight', pad_inches = 0)
```

Listing 4: Código generador de los boxplots con datos anómalos.

5 QQplot

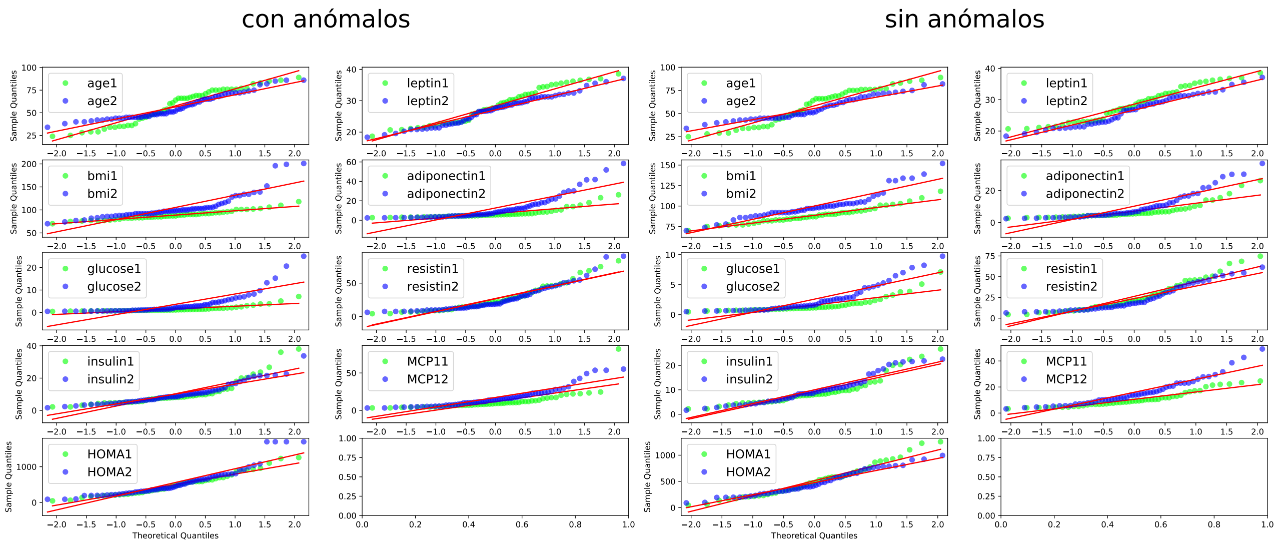

Fig. 4: QQplots para datos con y sin anomalias.

```
import matplotlib as mpl
import matplotlib.pyplot as plt

# load preprocessed data, x and y are raw, x_no and y_no contain no outliers
from preprocessing import x, y, x_no, y_no, labels
import statsmodels.api as sm

fc = [(), (0, 1, 0, 0.6), (0, 0, 1, 0.6)]

fig, ax = plt.subplots (nrows = 5, ncols = 2, figsize = (13, 10))
ax = ax.flatten ()

for i in range (0, 9):
    for j in [1, 2]:
        sm.qqplot (x [y == j, i], ax = ax[i], c = fc[j],
                line = ’s’, label = labels [i] + str (j))
        ax[i].legend (loc = 2, prop={’size’: 15})

fig.suptitle (’con anómalos’, fontsize = 30)
fig.savefig  (’../images/qqp.pdf’, bbox_inches = ’tight’, pad_inches = 0)
```

Listing 5: Código generador de los QQplots con datos anómalos.

6 Corrplot

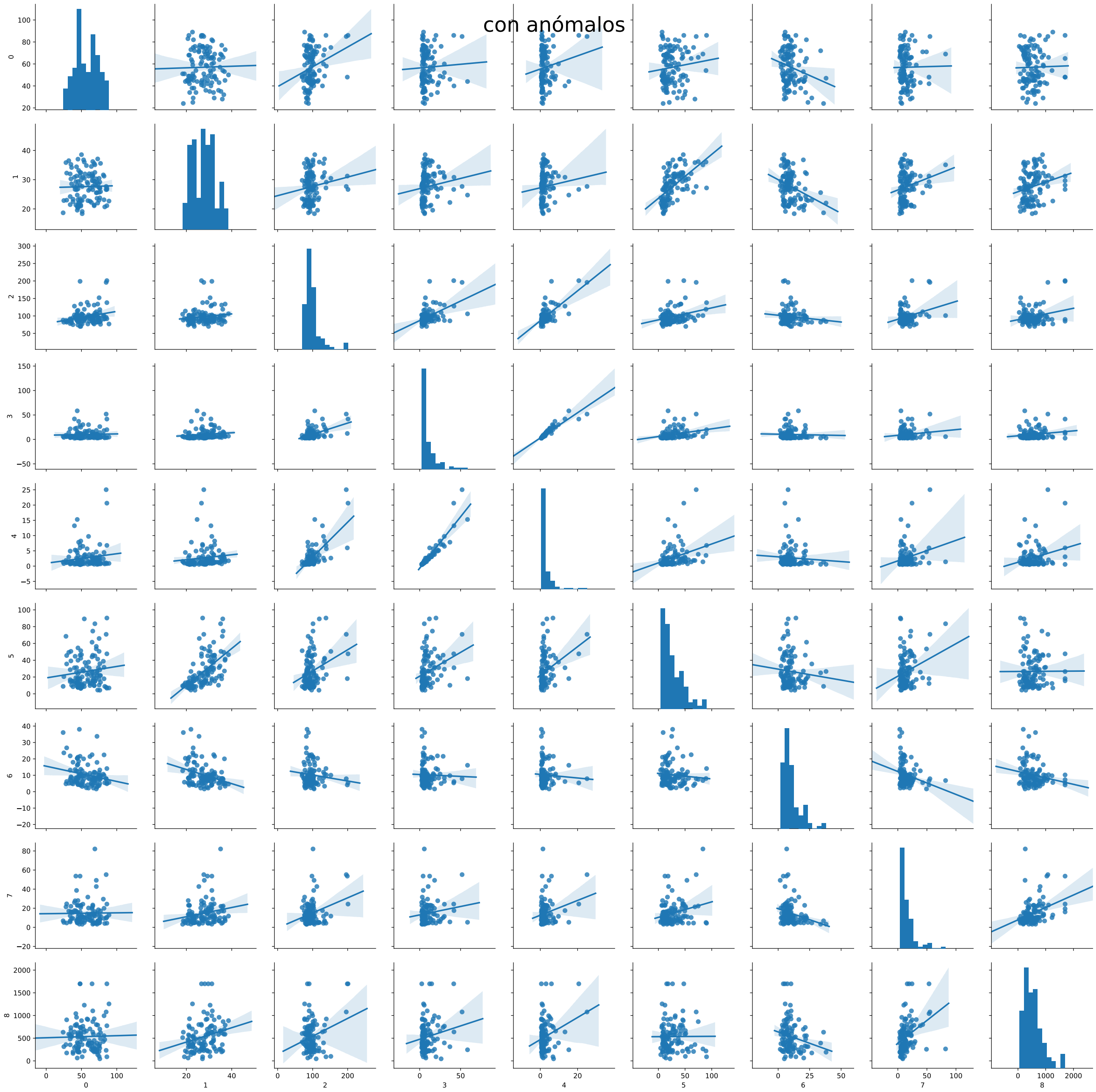

Fig. 5: Corrplot para datos con anomalias.

```
import seaborn as sns
import pandas as pd
from preprocessing import x, y, x_no, y_no, labels

dataframe = pd.DataFrame.from_records(x)
sns.pairplot(dataframe, kind = ’reg’)
plt.suptitle (’con anómalos’, fontsize = 30)
plt.savefig  (’../images/corrp.pdf’, bbox_inches = ’tight’, pad_inches = 0)
```

Listing 6: Código generador de los corrplots con datos anómalos.

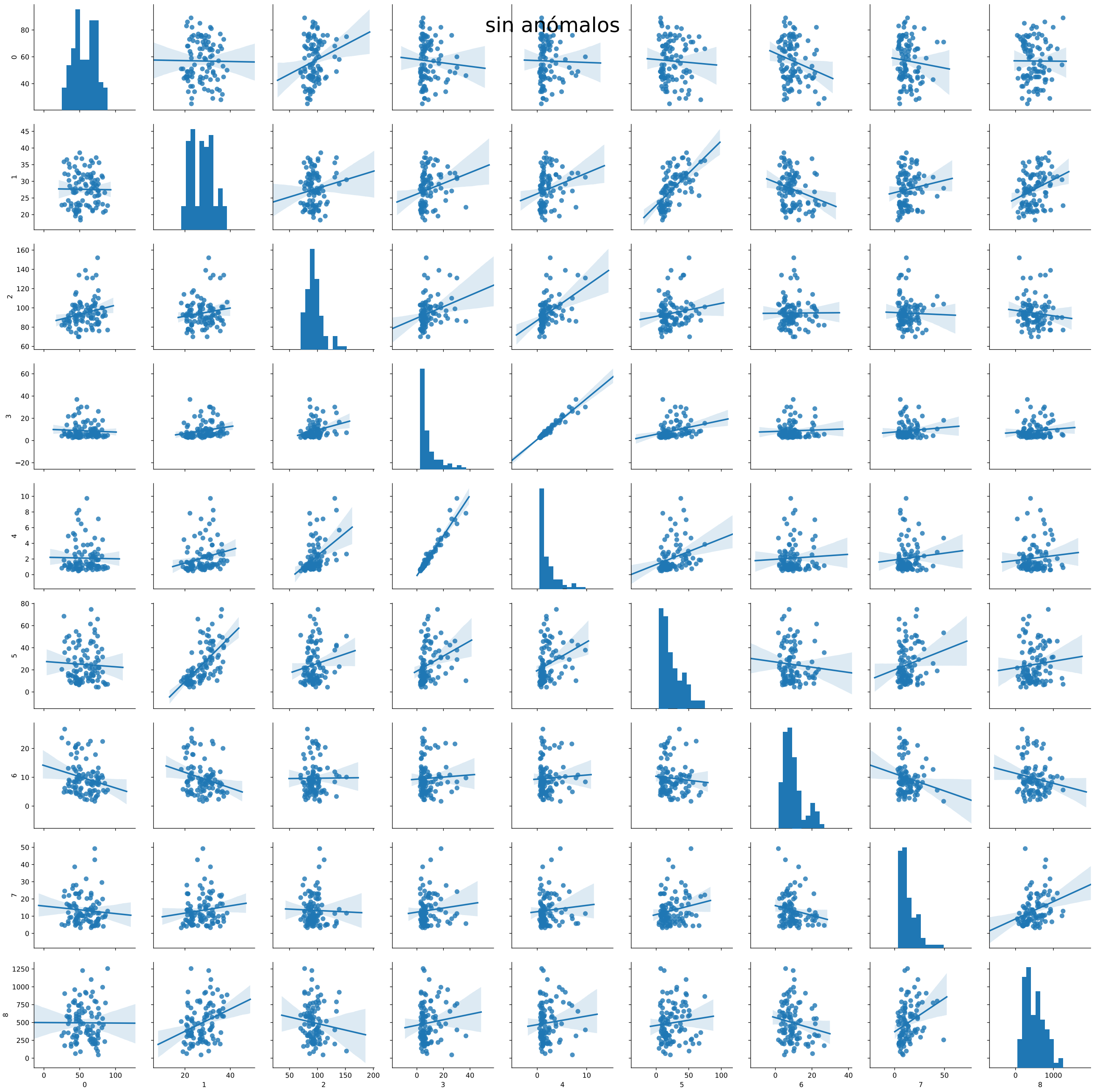

Fig. 6: Corrplot para datos sin anomalias.