

Sistemas de Información y Telemedicina. *

Marta Girones Sanguesa

Silvia Marset Gomis

Ignacio Amat Hernández

Sofía Gutiérrez Santamaría

November 25, 2019

1 Preámbulo

```
1 # names of variables
2 labels = ['age', 'leptin', 'bmi', 'adiponectin', 'glucose',
3           'resistin', 'insulin', 'MCP1', 'HOMA']
4
5 # loads data
6 data = np.loadtxt (open (r'../../data.csv', 'rb'), delimiter = ',')
7
8 # rewrites data as all the rows of data w/out nan cells
9 data = data [~np.isnan (data).any (axis=1)]
10
11 # separates parameters into matrix x
12 x = np.array ([list (data [x][: -1]) for x in range (len (data))])
13
14 # and class (1, 2) into vector y
15 y = np.array ([int (data [x][ -1]) for x in range (len (data))])
16
17 # removes outliers
18 data_no = data [(np.abs (stats.zscore (data)) < 3).all (axis = 1)]
19 # ↑ = No Outliers
20
21 x_no = np.array ([list (data_no [x][: -1]) for x in range (len (data_no))])
22 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.

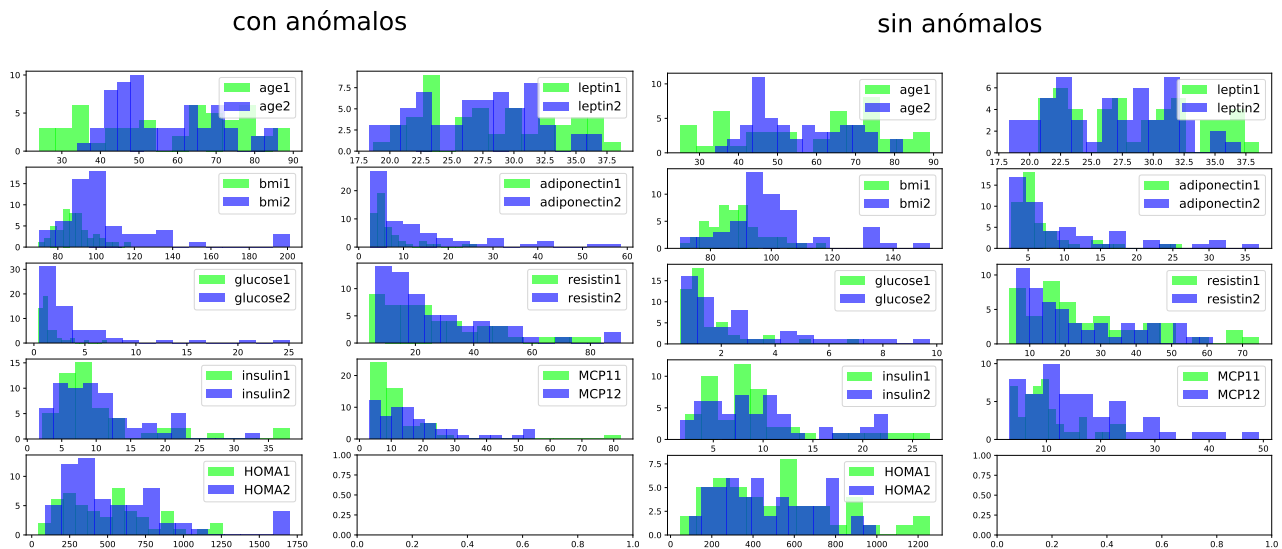


Fig. 1: Histogramas para datos con y sin anomalías.

```

1 import matplotlib as mpl
2 import matplotlib.pyplot as plt
3
4 # load preprocessed data, x and y are raw, x_no and y_no contain no outliers
5 from preprocessing import x, y, x_no, y_no, labels
6
7 # colours for the histograms
8 fc = [(), (0, 1, 0, 0.6), (0, 0, 1, 0.6)]
9 #      (R, G, B,  α )← transparency
10
11 fig, ax = plt.subplots (nrows = 5, ncols = 2, figsize = (13, 10))
12 ax = ax.flatten ()
13
14 # draws each of the histograms, two for each variable
15 for i in range (0, 9):
16     for j in [1, 2]:
17         ax[i].hist (x [y == j, i], bins = 15, fc = fc [j], label = labels [i] + str
18                     (j))
19         ax[i].legend (loc = 1, prop={'size': 15})
20
21 fig.suptitle ('con anomalías', fontsize = 30)
22 fig.savefig ('../images/hist.pdf', bbox_inches = 'tight', pad_inches = 0)

```

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

3 Kernel Density

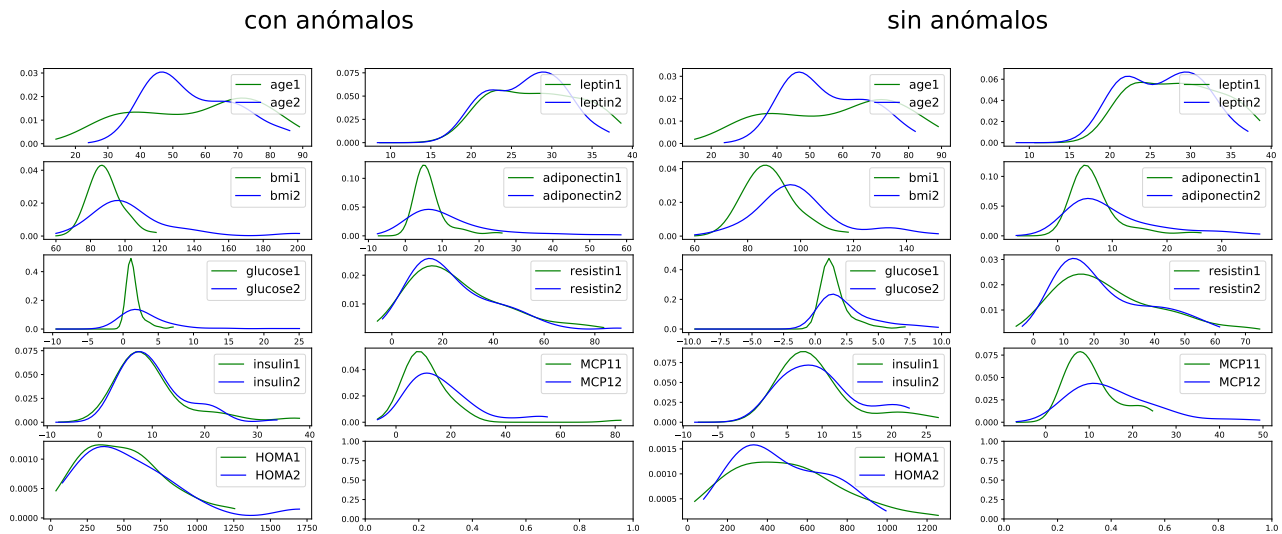


Fig. 2: Kernel Density para datos con y sin anomalias.

```

1  import matplotlib as mpl
2  import matplotlib.pyplot as plt
3  import numpy as np
4  from scipy.stats import gaussian_kde
5
6  # load preprocessed data, x and y are raw, x_no and y_no contain no outliers
7  from preprocessing import x, y, x_no, y_no, labels
8
9  # colours
10 fc = ['', 'green', 'blue']
11
12 fig, ax = plt.subplots (nrows = 5, ncols = 2, figsize = (13, 10))
13 ax = ax.flatten ()
14
15 # same loop in principle as before
16 for i in range (0, 9):
17     for j in [1, 2]:
18         kde = gaussian_kde (x_ := x [y == j, i])
19         xs = np.linspace(np.min (x_) - 10, np.max (x_), num=len (x_))
20         ax[i].plot (xs, kde(xs), c = fc[j], label = labels [i] + str (j))
21         ax[i].legend (loc = 1, prop={'size': 15})
22
23 fig.suptitle ('con anomalos', fontsize = 30)
24 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

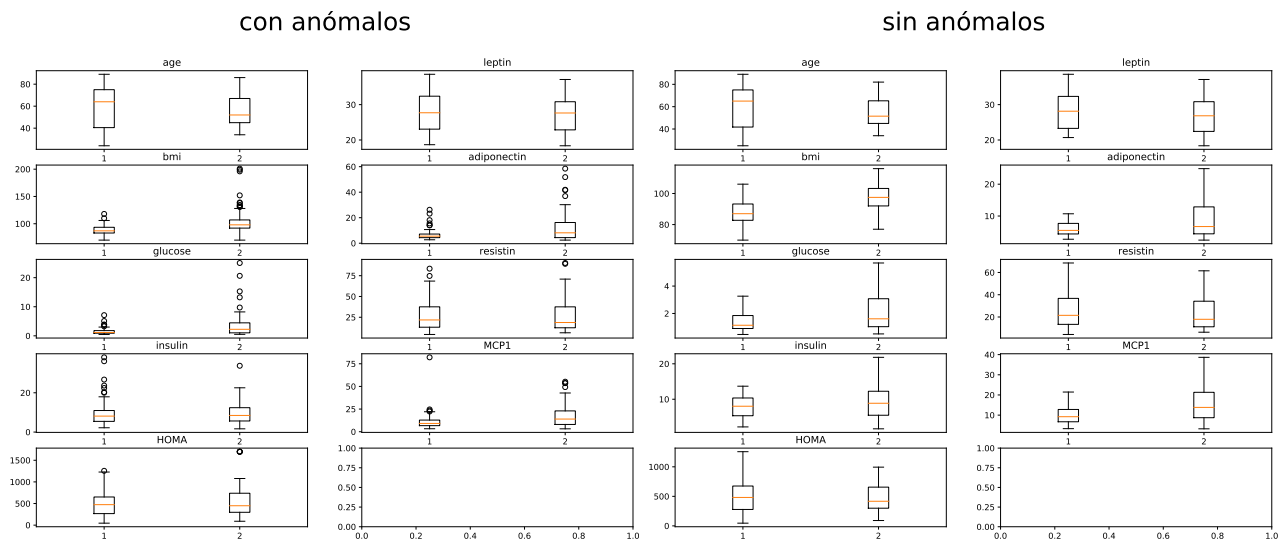


Fig. 3: Boxplots para datos con y sin anomalias.

```

1  import matplotlib as mpl
2  import matplotlib.pyplot as plt
3
4  # load preprocessed data, x and y are raw, x_no and y_no contain no outliers
5  from preprocessing import x, y, x_no, y_no, labels
6
7  fig, ax = plt.subplots (nrows = 5, ncols = 2, figsize = (13, 10))
8  ax = ax.flatten ()
9
10 for i in range (0, 9):
11     ax[i].boxplot ([x [y == 1, i], x [y == 2, i]])
12     ax[i].title.set_text (labels [i])
13
14 fig.suptitle ('con anomalos', fontsize = 30)
15 fig.savefig ('../images/boxp.pdf', bbox_inches = 'tight', pad_inches = 0)

```

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

5 QQplot

con anómalos

sin anómalos

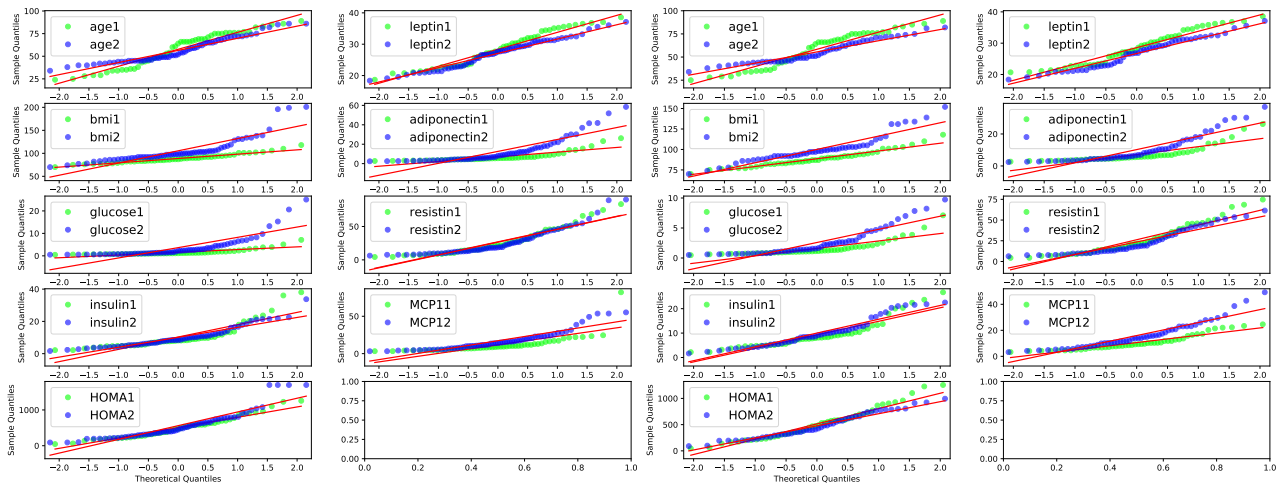


Fig. 4: QQplots para datos con y sin anomalías.

```

1 import matplotlib as mpl
2 import matplotlib.pyplot as plt
3
4 # load preprocessed data, x and y are raw, x_no and y_no contain no outliers
5 from preprocessing import x, y, x_no, y_no, labels
6 import statsmodels.api as sm
7
8 fc = [(0, (0, 1, 0, 0.6), (0, 0, 1, 0.6))]
9
10 fig, ax = plt.subplots (nrows = 5, ncols = 2, figsize = (13, 10))
11 ax = ax.flatten ()
12
13 for i in range (0, 9):
14     for j in [1, 2]:
15         sm.qqplot (x [y == j, i], ax = ax[i], c = fc[j],
16                     line = 's', label = labels [i] + str (j))
17         ax[i].legend (loc = 2, prop={'size': 15})
18
19 fig.suptitle ('con anómalos', fontsize = 30)
20 fig.savefig ('../images/qqp.pdf', bbox_inches = 'tight', pad_inches = 0)

```

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