Predicción de mortandad de pacientes con fallo al corazón

Integrantes del equipo:

Carlos Sánchez Mejorada Raynal A01702188

María de los Angeles Arista Huerta A01369984

Ariann Fernando Arriaga Alcántara A01703556

Base de datos de pacientes que han tenido una falla cardiaca, compilando datos del nivel de suero de creatina y Fracción de eyección

Importación de librerias y declaración del archivo en modo lectura como una variable

In [4]:
```
import numpy as np
import seaborn as sb
import pandas as pd
import csv
from matplotlib import pyplot as plt
data=pd.read_csv('heart_failure_clinical_records_dataset.csv')
```

Base de datos:

In [5]:
```
data
```

Out[5]:

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelet
0	75.0	0	582	0	20	1	265000.0
1	55.0	0	7861	0	38	0	263358.0
2	65.0	0	146	0	20	0	162000.0
3	50.0	1	111	0	20	0	210000.0
4	65.0	1	160	1	20	0	327000.0
...	...	...	...	...	...	...	
294	62.0	0	61	1	38	1	155000.0
295	55.0	0	1820	0	38	0	270000.0
296	45.0	0	2060	1	60	0	742000.0
297	45.0	0	2413	0	38	0	140000.0
298	50.0	0	196	0	45	0	395000.0

299 rows × 13 columns

In [6]:
```
type (data)
```

Out[6]:
```
pandas.core.frame.DataFrame
```

Encabezado de bases de datos

In [7]:
```
data.head()
```

Out[7]:

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets
0	75.0	0	582	0	20	1	265000.00
1	55.0	0	7861	0	38	0	263358.03
2	65.0	0	146	0	20	0	162000.00
3	50.0	1	111	0	20	0	210000.00
4	65.0	1	160	1	20	0	327000.00

Análisis de datos generales:

-Numeros de lineas

-Dialecto

-Campos de información recabada

In [8]:
```
csvarchivo=open('heart_failure_clinical_records_dataset.csv')
entrada=csv.DictReader(csvarchivo)
listadeval=list(entrada)
print('Líneas:',entrada.line_num)
print('Dialecto:',entrada.dialect)
print('Campos:',entrada.fieldnames)
```

```
Líneas: 300
Dialecto: excel
Campos: ['age', 'anaemia', 'creatinine_phosphokinase', 'diabetes', 'ejection_fraction',
'high_blood_pressure', 'platelets', 'serum_creatinine', 'serum_sodium', 'sex', 'smokin
g', 'time', 'DEATH_EVENT']
```

Análisis de datos columna por columna

In [9]:
```
type(data.age)
```

Out[9]:
```
pandas.core.series.Series
```

Edades de pacientes

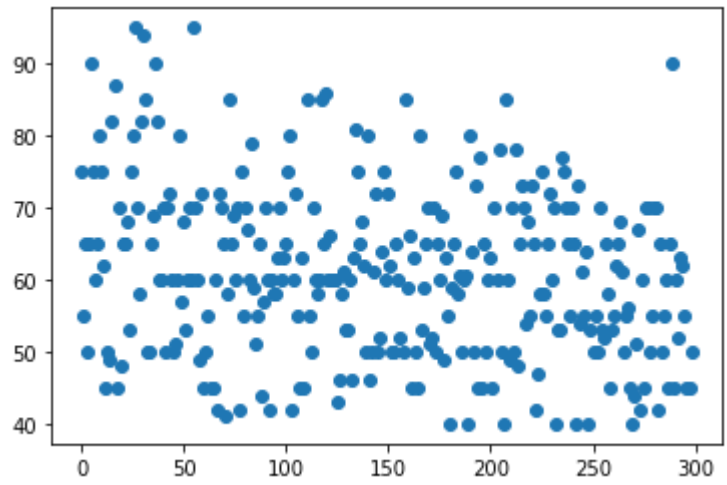
In [10]:
```
data.age
```

Out[10]:
```
0      75.0
1      55.0
2      65.0
3      50.0
4      65.0
       ...
294    62.0
295    55.0
296    45.0
297    45.0
```

```
298    50.0
Name: age, Length: 299, dtype: float64
```

Representación gráfica de edades.

In [11]:
```
plt.plot(data.age,'o')
plt.show()
```



Tipos de datos

In [12]:
```
pd.unique(data['age'])
```

Out[12]:
```
array([75.   , 55.   , 65.   , 50.   , 90.   , 60.   , 80.   , 62.   ,
       45.   , 49.   , 82.   , 87.   , 70.   , 48.   , 68.   , 53.   ,
       95.   , 58.   , 94.   , 85.   , 69.   , 72.   , 51.   , 57.   ,
       42.   , 41.   , 67.   , 79.   , 59.   , 44.   , 63.   , 86.   ,
       66.   , 43.   , 46.   , 61.   , 81.   , 52.   , 64.   , 40.   ,
       60.667, 73.   , 77.   , 78.   , 54.   , 47.   , 56.   ])
```

Medidas de dispersión de los datos

In [13]:
```
data['age'].describe()
```

Out[13]:
```
count    299.000000
mean      60.833893
std       11.894809
min       40.000000
25%       51.000000
50%       60.000000
75%       70.000000
max       95.000000
Name: age, dtype: float64
```

Promedio de los datos

In [14]:
```
data['age'].mean()
```

Out[14]:
```
60.83389297658862
```

Mediana de los datos

In [15]:
```
data['age'].median()
```

Out[15]:
```
60.0
```

Desviación estandar de los datos

In [16]:
```
data['age'].std()
```

Out[16]:
```
11.894809074044478
```

Rango de los datos

In [17]:
```
Ran=data['age'].max()-data['age'].min()
print(Ran)
```

```
55.0
```

Conclusión de datos estadísticos descriptivos.

El promedio de edades son 61 años

El rango de las edades de los pacienttes son 55

La mediana de los datos son 60

La desviación estandar de las edades es equivalente a 11.894

In [18]:
```
type(data.anaemia)
```

Out[18]:
```
pandas.core.series.Series
```

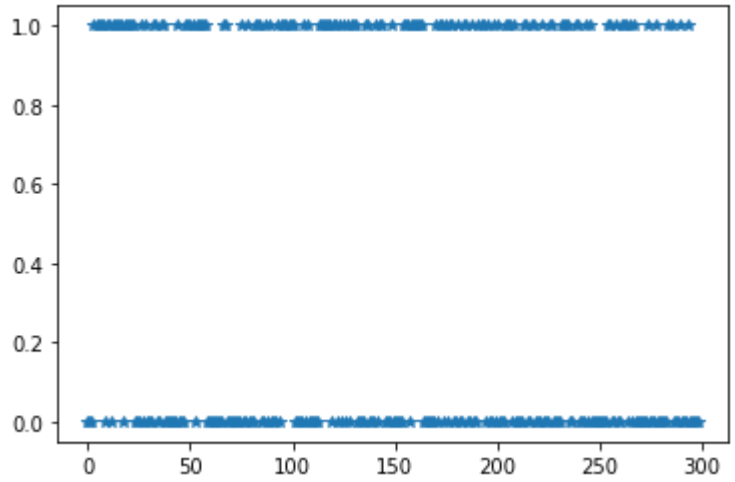
Anemia en pacientes

In [19]:
```
data.anaemia
```

Out[19]:
```
0      0
1      0
2      0
3      1
4      1
      ..
294    0
295    0
296    0
297    0
298    0
Name: anaemia, Length: 299, dtype: int64
```

Representación gráfica de los datos

In [20]:
```
plt.plot(data.anaemia,'*')
plt.show()
```



Tipos de datos

In [21]:
```
pd.unique(data['anaemia'])
```

Out[21]:
```
array([0, 1], dtype=int64)
```

Medidas de dispersión de los datos

In [22]:
```
data['anaemia'].describe()
```

Out[22]:
```
count    299.000000
mean       0.431438
std        0.496107
min        0.000000
25%        0.000000
50%        0.000000
75%        1.000000
max        1.000000
Name: anaemia, dtype: float64
```

Promedio de los datos

In [23]:
```
data['anaemia'].mean()
```

Out[23]:
```
0.431438127090301
```

Mediana de los datos

In [24]:
```
data['anaemia'].median()
```

Out[24]:
```
0.0
```

Desviación estandar de los datos

In [25]:
```
data['anaemia'].std()
```

Out[25]:
```
0.49610726813307915
```

Rango de los datos

In [26]:
```
Ran2=data['anaemia'].max()-data['anaemia'].min()
print(Ran2)
```

```
1
```

Conclusión de datos estadísticos descriptivos.

La minoria de las pacientes tienen anemia esto se debe a que el valor de la media se aproxima más a 0 que a 1.

In [27]:
```
type(data.creatinine_phosphokinase)
```

Out[27]:
```
pandas.core.series.Series
```

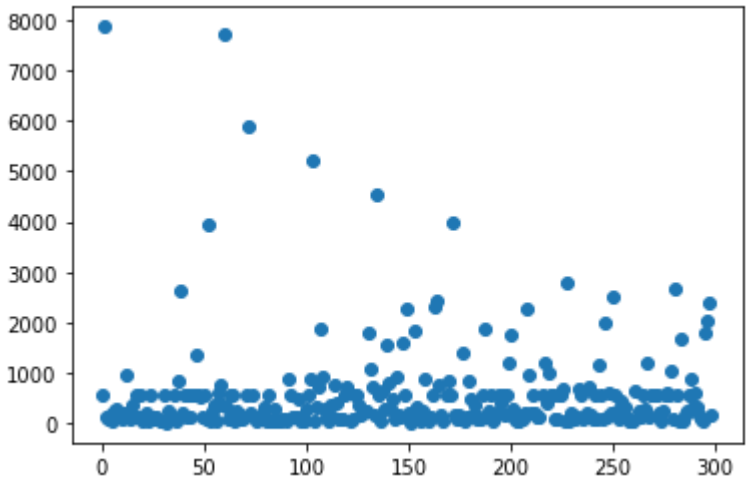
Creatina de los Pacientes

In [28]:
```
data.creatinine_phosphokinase
```

Out[28]:
```
0       582
1      7861
2       146
3       111
4       160
       ...
294      61
295    1820
296    2060
297    2413
298     196
Name: creatinine_phosphokinase, Length: 299, dtype: int64
```

Representación gráfica de la creatina.

In [29]:
```
plt.plot(data.creatinine_phosphokinase,'o')
plt.show()
```



Tipos de datos

In [30]:
```
pd.unique(data['creatinine_phosphokinase'])
```

Out[30]:
```
array([ 582, 7861,  146,  111,  160,   47,  246,  315,  157,  123,   81,
        231,  981,  168,   80,  379,  149,  125,   52,  128,  220,   63,
        148,  112,  122,   60,   70,   23,  249,  159,   94,  855, 2656,
        235,  124,  571,  127,  588, 1380,  553,  129,  577,   91, 3964,
         69,  260,  371,   75,  607,  789,  364, 7702,  318,  109,   68,
        250,  110,  161,  113, 5882,  224,   92,  102,  203,  336,   76,
         55,  280,   78,   84,  115,   66,  897,  154,  144,  133,  514,
         59,  156,   61,  305,  898, 5209,   53,  328,  748, 1876,  936,
        292,  369,  143,  754,  400,   96,  737,  358,  200,  248,  270,
       1808, 1082,  719,  193, 4540,  646,  281, 1548,  805,  291,  482,
```

```
  943,  185,  132, 1610, 2261,  233,   30, 1846,  335,   58,  910,
   72,  130, 2334, 2442,  776,  196,  835, 3966,  171,  198,   95,
 1419,  478,  176,  395,   99,  145,  104, 1896,  151,  244,   62,
  121,  418,  167, 1211, 1767,  308,   97,   64,  101,  212, 2281,
  972,  131,  135, 1202,  427, 1021,  118,   86,  675,   57, 2794,
   56,  211,  166,   93,  707,  119,  232,  720,  180,   90, 1185,
 2017,  624,  207, 2522,  572,  245,   88,  446,  191,  326,  655,
  258,  298, 1199,  213,  257,  618, 1051, 2695, 1688,   54,  170,
  253,  892,  337,  615,  320,  190,  103, 1820, 2060, 2413],
dtype=int64)
```

Medidas de dispersión de los datos

In [31]:
```
data['creatinine_phosphokinase'].describe()
```

Out[31]:
```
count     299.000000
mean      581.839465
std       970.287881
min        23.000000
25%       116.500000
50%       250.000000
75%       582.000000
max      7861.000000
Name: creatinine_phosphokinase, dtype: float64
```

Promedio de los datos

In [32]:
```
data['creatinine_phosphokinase'].mean()
```

Out[32]:
```
581.8394648829432
```

Mediana de los datos

In [33]:
```
data['creatinine_phosphokinase'].median()
```

Out[33]:
```
250.0
```

Desviación estandar de los datos

In [34]:
```
data['creatinine_phosphokinase'].std()
```

Out[34]:
```
970.2878807124363
```

Rango de los datos

In [35]:
```
Ran3=data['creatinine_phosphokinase'].max()-data['creatinine_phosphokinase'].min()
print(Ran3)
```

```
7838
```

Conclusión de datos estadísticos descriptivos.

El promedio de la cantidad de creatinina de los pacientes es de 581.82

El rango de la cantidad de creatinina de los pacientes es de 7838

La mediana de los datos es 250

La desviación estandar de la cantidad de creatinina de los pacientes es de 970.287

In [36]:
```
type(data.diabetes)
```

Out[36]:
```
pandas.core.series.Series
```

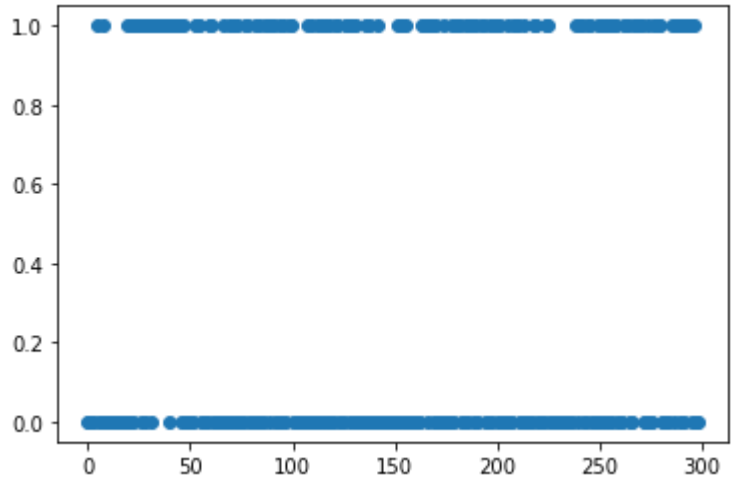
Diabetes en pacientes

In [37]:
```
data.diabetes
```

Out[37]:
```
0      0
1      0
2      0
3      0
4      1
      ..
294    1
295    0
296    1
297    0
298    0
Name: diabetes, Length: 299, dtype: int64
```

Representación gráfica de diabetes.

In [38]:
```
plt.plot(data.diabetes,'o')
plt.show()
```



Tipos de datos

In [39]:
```
pd.unique(data['diabetes'])
```

Out[39]:
```
array([0, 1], dtype=int64)
```

Medidas de dispersión de los datos

In [40]:
```
data['diabetes'].describe()
```

Out[40]:
```
count    299.000000
mean       0.418060
std        0.494067
min        0.000000
25%        0.000000
50%        0.000000
75%        1.000000
max        1.000000
Name: diabetes, dtype: float64
```

Promedio de los datos

In [41]:
```
data['diabetes'].mean()
```

Out[41]:
```
0.4180602006688963
```

Mediana de los datos

In [42]:
```
data['diabetes'].median()
```

Out[42]:
```
0.0
```

Desviación estandar de los datos

In [43]:
```
data['diabetes'].std()
```

Out[43]:
```
0.49406706510360887
```

Rango de los datos

In [44]:
```
Ran=data['diabetes'].max()-data['diabetes'].min()
print(Ran)
```

```
1
```

Conclusión de datos estadísticos descriptivos.

La minoria de las pacientes tienen diabetes esto se debe a que el valor de la media se aproxima más a 0 que a 1.

In [45]:
```
type(data.ejection_fraction)
```

Out[45]:
```
pandas.core.series.Series
```

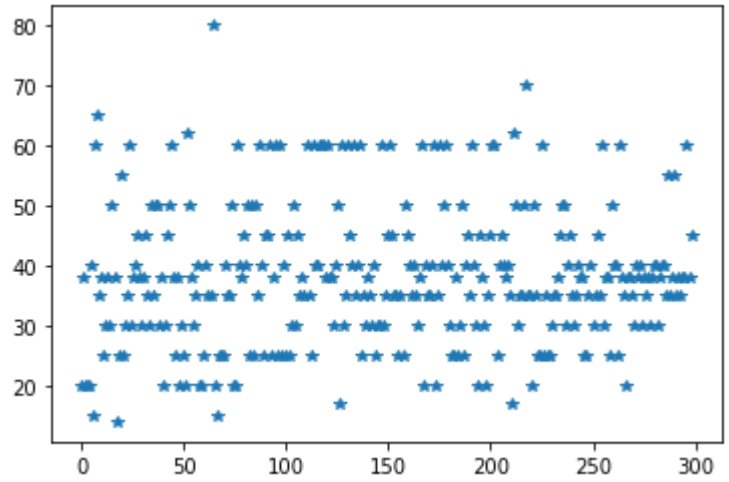
Fracción de eyección en pacientes

In [46]:
```
data.ejection_fraction
```

Out[46]:
```
0      20
1      38
2      20
3      20
4      20
       ..
294    38
295    38
296    60
297    38
298    45
Name: ejection_fraction, Length: 299, dtype: int64
```

Representación gráfica de los datos

In [47]:
```
plt.plot(data.ejection_fraction,'*')
plt.show()
```



Tipos de datos

In [48]:
```
pd.unique(data['ejection_fraction'])
```

Out[48]:
```
array([20, 38, 40, 15, 60, 65, 35, 25, 30, 50, 14, 55, 45, 62, 80, 17, 70],
      dtype=int64)
```

Medidas de dispersión de los datos

In [49]:
```
data['ejection_fraction'].describe()
```

Out[49]:
```
count    299.000000
mean      38.083612
std       11.834841
min       14.000000
25%       30.000000
50%       38.000000
75%       45.000000
max       80.000000
Name: ejection_fraction, dtype: float64
```

Promedio de los datos

In [50]:
```
data['ejection_fraction'].mean()
```

Out[50]:
```
38.08361204013378
```

Mediana de los datos

In [51]:
```
data['ejection_fraction'].median()
```

Out[51]:
```
38.0
```

Desviación estandar de los datos

In [52]:
```
data['ejection_fraction'].std()
```

Out[52]:
```
11.834840741039173
```

Rango de los datos

In [53]:
```
Ran2=data['ejection_fraction'].max()-data['ejection_fraction'].min()
print(Ran2)
```

66

Conclusión de datos estadísticos descriptivos.

El promedio de la cantidad de fracción de eyección de los pacientes es de 38.08

El rango de la cantidad de fracción de eyección de los pacientes es de 66

La mediana de los datos es 38

La desviación estandar de la cantidad de fracción de eyección de los pacientes es de 11.834

In [54]:
```
type(data.high_blood_pressure)
```

Out[54]:
```
pandas.core.series.Series
```

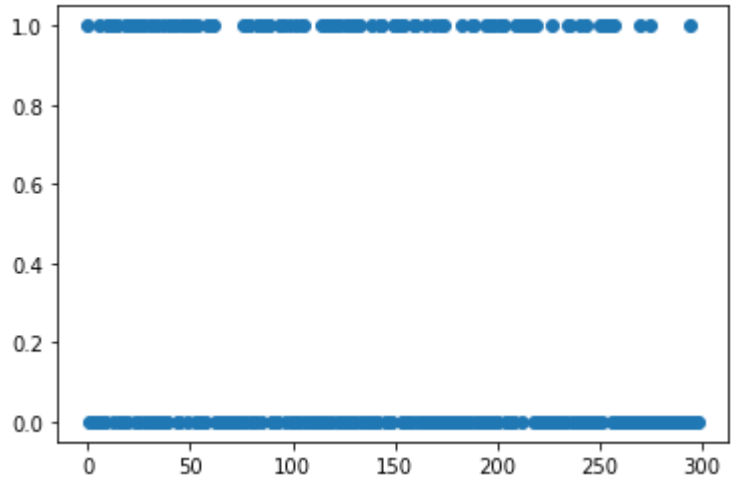
Alta presión de los Pacientes

In [55]:
```
data.high_blood_pressure
```

Out[55]:
```
0      1
1      0
2      0
3      0
4      0
      ..
294    1
295    0
296    0
297    0
298    0
Name: high_blood_pressure, Length: 299, dtype: int64
```

Representación gráfica de la alta presión en los pacientes.

In [56]:
```
plt.plot(data.high_blood_pressure,'o')
plt.show()
```



Tipos de datos

In [57]:
```
pd.unique(data['high_blood_pressure'])
```

Out[57]:
```
array([1, 0], dtype=int64)
```

Medidas de dispersión de los datos

In [58]:
```
data['high_blood_pressure'].describe()
```

Out[58]:
```
count    299.000000
mean       0.351171
std        0.478136
min        0.000000
25%        0.000000
50%        0.000000
75%        1.000000
max        1.000000
Name: high_blood_pressure, dtype: float64
```

Promedio de los datos

In [59]:
```
data['high_blood_pressure'].mean()
```

Out[59]:
```
0.3511705685618729
```

Mediana de los datos

In [60]:
```
data['high_blood_pressure'].median()
```

Out[60]:
```
0.0
```

Desviación estandar de los datos

In [61]:
```
data['high_blood_pressure'].std()
```

Out[61]:
```
0.4781363790627452
```

Rango de los datos

In [62]:
```
Ran3=data['high_blood_pressure'].max()-data['high_blood_pressure'].min()
print(Ran3)
```

```
1
```

Conclusión de datos estadísticos descriptivos.

La minoria de las pacientes tienen hipertensión esto se debe a que el valor de la media se aproxima más a 0 que a 1.

In [63]:
```
type(data.platelets)
```

Out[63]:
```
pandas.core.series.Series
```

Platelets de pacientes

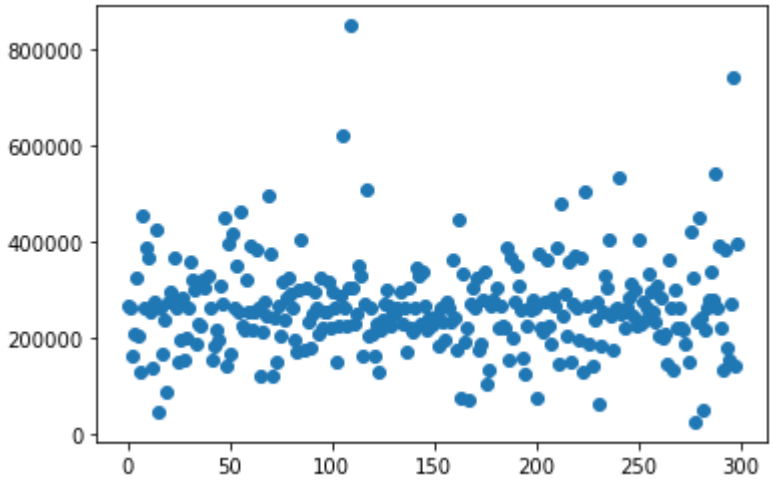
In [64]:
```
data.platelets
```

Out[64]:
```
0      265000.00
1      263358.03
2      162000.00
3      210000.00
4      327000.00
         ...
294    155000.00
295    270000.00
```

```
296    742000.00
297    140000.00
298    395000.00
Name: platelets, Length: 299, dtype: float64
```

Representación gráfica de plaquetas.

In [65]:
```
plt.plot(data.platelets,'o')
plt.show()
```



Tipos de datos

In [66]:
```
pd.unique(data['platelets'])
```

Out[66]:
```
array([265000.  , 263358.03, 162000.  , 210000.  , 327000.  , 204000.  ,
       127000.  , 454000.  , 388000.  , 368000.  , 253000.  , 136000.  ,
       276000.  , 427000.  ,  47000.  , 262000.  , 166000.  , 237000.  ,
        87000.  , 297000.  , 289000.  , 149000.  , 196000.  , 284000.  ,
       153000.  , 200000.  , 360000.  , 319000.  , 302000.  , 188000.  ,
       228000.  , 226000.  , 321000.  , 305000.  , 329000.  , 185000.  ,
       218000.  , 194000.  , 310000.  , 271000.  , 451000.  , 140000.  ,
       395000.  , 418000.  , 351000.  , 255000.  , 461000.  , 223000.  ,
       216000.  , 254000.  , 390000.  , 385000.  , 119000.  , 213000.  ,
       274000.  , 244000.  , 497000.  , 374000.  , 122000.  , 243000.  ,
       266000.  , 317000.  , 283000.  , 324000.  , 293000.  , 172000.  ,
       406000.  , 173000.  , 304000.  , 235000.  , 181000.  , 249000.  ,
       219000.  , 318000.  , 221000.  , 298000.  , 286000.  , 621000.  ,
       263000.  , 850000.  , 306000.  , 252000.  , 328000.  , 164000.  ,
       507000.  , 203000.  , 217000.  , 300000.  , 267000.  , 227000.  ,
       250000.  , 295000.  , 231000.  , 211000.  , 348000.  , 229000.  ,
       338000.  , 242000.  , 225000.  , 184000.  , 277000.  , 362000.  ,
       174000.  , 448000.  ,  75000.  , 334000.  , 192000.  , 220000.  ,
        70000.  , 270000.  , 325000.  , 176000.  , 189000.  , 281000.  ,
       337000.  , 105000.  , 132000.  , 279000.  , 303000.  , 224000.  ,
       389000.  , 365000.  , 201000.  , 275000.  , 350000.  , 309000.  ,
       260000.  , 160000.  , 126000.  , 259000.  ,  73000.  , 377000.  ,
       212000.  , 186000.  , 268000.  , 147000.  , 481000.  , 290000.  ,
       358000.  , 151000.  , 371000.  , 130000.  , 504000.  , 141000.  ,
        62000.  , 330000.  , 248000.  , 257000.  , 533000.  , 264000.  ,
       282000.  , 314000.  , 246000.  , 301000.  , 404000.  , 236000.  ,
       294000.  , 233000.  , 308000.  , 198000.  , 208000.  , 133000.  ,
       222000.  , 215000.  , 150000.  , 422000.  ,  25100.  , 232000.  ,
       241000.  ,  51000.  , 336000.  , 543000.  , 382000.  , 179000.  ,
       155000.  , 742000.  ])
```

Medidas de dispersión de los datos

In [67]:
```
data['platelets'].describe()
```

Out[67]:
```
count       299.000000
mean     263358.029264
std       97804.236869
min       25100.000000
25%      212500.000000
50%      262000.000000
75%      303500.000000
max      850000.000000
Name: platelets, dtype: float64
```

Promedio de los datos

In [68]:
```
data['platelets'].mean()
```

Out[68]:
```
263358.02926421416
```

Mediana de los datos

In [69]:
```
data['platelets'].median()
```

Out[69]:
```
262000.0
```

Desviación estandar de los datos

In [70]:
```
data['platelets'].std()
```

Out[70]:
```
97804.23686859828
```

Rango de los datos

In [71]:
```
Ran=data['platelets'].max()-data['platelets'].min()
print(Ran)
```

```
824900.0
```

Conclusión de datos estadísticos descriptivos.

El promedio de la cantidad de plaquetas de los pacientes es de 263358.0292

El rango de la cantidad de plaquetas de los pacientes es de 824900

La mediana de los datos es 262000

La desviación estandar de la cantidad de plaquetas de los pacientes es de 97804.236

In [72]:
```
type(data.serum_creatinine)
```

Out[72]:
```
pandas.core.series.Series
```

Suero de creatina en pacientes

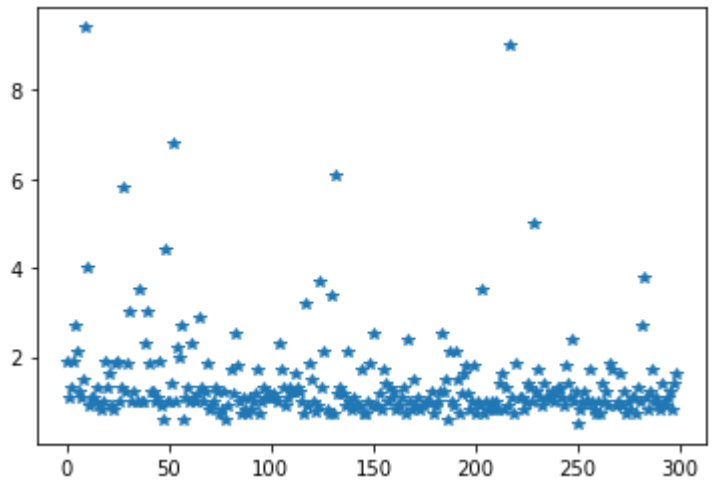
In [73]:
```
data.serum_creatinine
```

Out[73]:
```
0      1.9
1      1.1
2      1.3
3      1.9
4      2.7
```

```
      ...
294    1.1
295    1.2
296    0.8
297    1.4
298    1.6
Name: serum_creatinine, Length: 299, dtype: float64
```

Representación gráfica de los datos

In [74]:
```
plt.plot(data.serum_creatinine,'*')
plt.show()
```



Tipos de datos

In [75]:
```
pd.unique(data['serum_creatinine'])
```

Out[75]:
```
array([1.9 , 1.1 , 1.3 , 2.7 , 2.1 , 1.2 , 1.5 , 9.4 , 4.  , 0.9 , 1.  ,
       0.8 , 1.6 , 1.83, 5.8 , 3.  , 3.5 , 2.3 , 0.6 , 4.4 , 1.4 , 6.8 ,
       2.2 , 2.  , 1.18, 2.9 , 0.7 , 1.7 , 2.5 , 1.8 , 3.2 , 0.75, 3.7 ,
       3.4 , 6.1 , 2.4 , 9.  , 5.  , 0.5 , 3.8 ])
```

Medidas de dispersión de los datos

In [76]:
```
data['serum_creatinine'].describe()
```

Out[76]:
```
count    299.00000
mean       1.39388
std        1.03451
min        0.50000
25%        0.90000
50%        1.10000
75%        1.40000
max        9.40000
Name: serum_creatinine, dtype: float64
```

Promedio de los datos

In [77]:
```
data['serum_creatinine'].mean()
```

Out[77]:
```
1.393879598662207
```

Mediana de los datos

In [78]:
```
data['serum_creatinine'].median()
```

Out[78]:
```
1.1
```

Desviación estandar de los datos

In [79]:
```
data['serum_creatinine'].std()
```

Out[79]:
```
1.034510064089853
```

Rango de los datos

In [80]:
```
Ran2=data['serum_creatinine'].max()-data['serum_creatinine'].min()
print(Ran2)
```

```
8.9
```

Conclusión de datos estadísticos descriptivos.

El promedio de la cantidad de suero de creatina de los pacientes es de 1.3938

El rango de la cantidad de suero de creatina de los pacientes es de 8.9

La mediana de los datos es 1.1

La desviación estandar de la cantidad de suero de creatina de los pacientes es de 1.0345

In [81]:
```
type(data.serum_sodium)
```

Out[81]:
```
pandas.core.series.Series
```

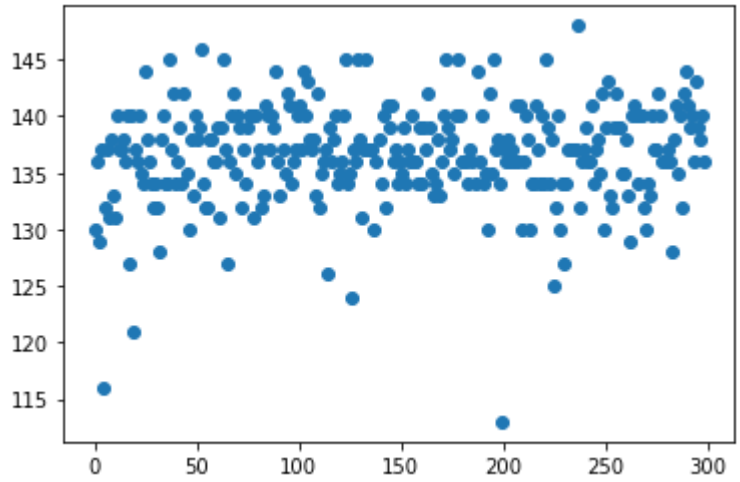
Suero de sodio de los Pacientes

In [82]:
```
data.serum_sodium
```

Out[82]:
```
0      130
1      136
2      129
3      137
4      116
      ...
294    143
295    139
296    138
297    140
298    136
Name: serum_sodium, Length: 299, dtype: int64
```

Representación gráfica del sodio en pacientes.

In [83]:
```
plt.plot(data.serum_sodium,'o')
plt.show()
```



Tipos de datos

In [84]:
```
pd.unique(data['serum_sodium'])
```

Out[84]:
```
array([130, 136, 129, 137, 116, 132, 131, 138, 133, 140, 127, 121, 135,
       134, 144, 128, 145, 142, 139, 146, 141, 143, 126, 124, 113, 125,
       148], dtype=int64)
```

Medidas de dispersión de los datos

In [85]:
```
data['serum_sodium'].describe()
```

Out[85]:
```
count    299.000000
mean     136.625418
std        4.412477
min      113.000000
25%      134.000000
50%      137.000000
75%      140.000000
max      148.000000
Name: serum_sodium, dtype: float64
```

Promedio de los datos

In [86]:
```
data['serum_sodium'].mean()
```

Out[86]:
```
136.62541806020067
```

Mediana de los datos

In [87]:
```
data['serum_sodium'].median()
```

Out[87]:
```
137.0
```

Desviación estandar de los datos

In [88]:
```
data['serum_sodium'].std()
```

Out[88]:
```
4.412477283909233
```

Rango de los datos

In [89]:
```
Ran3=data['serum_sodium'].max()-data['serum_sodium'].min()
print(Ran3)
```

```
35
```

Conclusión de datos estadísticos descriptivos.

El promedio de la cantidad de suero de sodio de los pacientes es de 136.625

El rango de la cantidad de suero de sodio de los pacientes es de 35

La mediana de los datos es 137

La desviación estandar de la cantidad de suero de sodio de los pacientes es de 4.4124

In [90]:
```
type(data.sex)
```

Out[90]:
```
pandas.core.series.Series
```

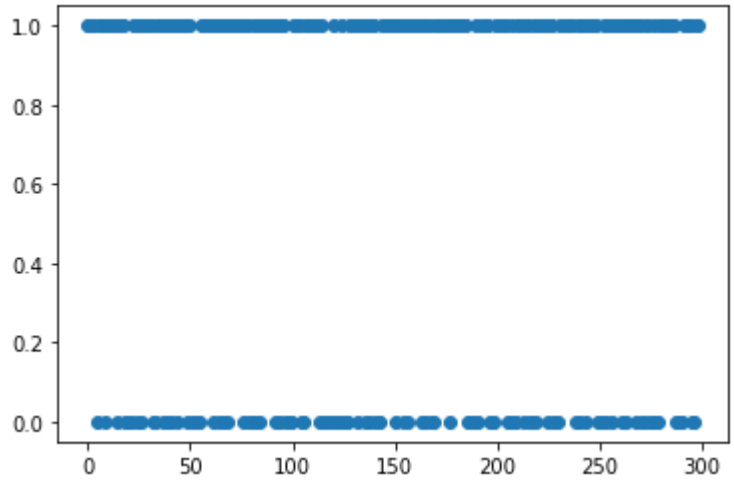
Sexo de pacientes

In [91]:
```
data.sex
```

Out[91]:
```
0      1
1      1
2      1
3      1
4      0
      ..
294    1
295    0
296    0
297    1
298    1
Name: sex, Length: 299, dtype: int64
```

Representación gráfica de sexos de pacientes.

In [92]:
```
plt.plot(data.sex,'o')
plt.show()
```



Tipos de datos

In [93]:
```
pd.unique(data['sex'])
```

Out[93]:
```
array([1, 0], dtype=int64)
```

Medidas de dispersión de los datos

In [94]:
```
data['sex'].describe()
```

Out[94]:
```
count    299.000000
mean       0.648829
std        0.478136
min        0.000000
25%        0.000000
50%        1.000000
75%        1.000000
max        1.000000
Name: sex, dtype: float64
```

Promedio de los datos

In [95]:
```
data['sex'].mean()
```

Out[95]:
```
0.6488294314381271
```

Mediana de los datos

In [96]:
```
data['sex'].median()
```

Out[96]:
```
1.0
```

Desviación estandar de los datos

In [97]:
```
data['sex'].std()
```

Out[97]:
```
0.47813637906274487
```

Rango de los datos

In [98]:
```
Ran=data['sex'].max()-data['sex'].min()
print(Ran)
```

```
1
```

Conclusión de datos estadísticos descriptivos.

La mayoría de las pacientes son hombres esto se debe a que el valor de la media se aproxima más a 0 que a 1.

In [99]:
```
type(data.smoking)
```

Out[99]:
```
pandas.core.series.Series
```

Pacientes que fuman

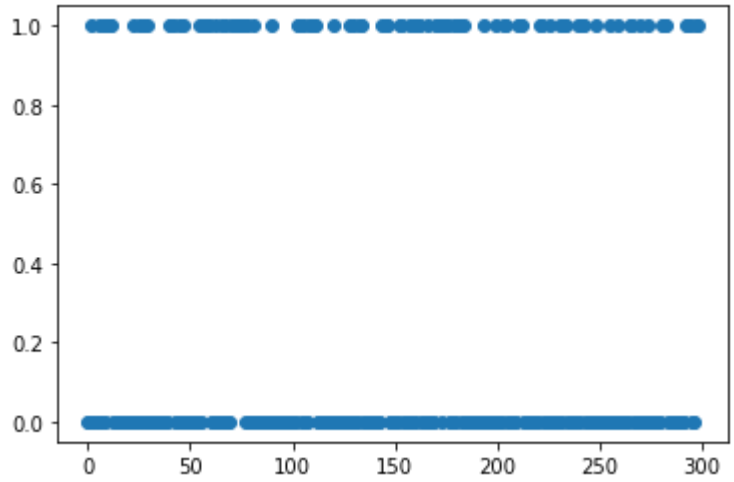
In [100…
```
data.smoking
```

Out[100…
```
0      0
1      0
2      1
3      0
4      0
      ..
294    1
295    0
```

```
296    0
297    1
298    1
Name: smoking, Length: 299, dtype: int64
```

Representación gráfica de los datos

In [101…
```
plt.plot(data.smoking,'o')
plt.show()
```



Tipos de datos

In [102…
```
pd.unique(data['smoking'])
```

Out[102…
```
array([0, 1], dtype=int64)
```

Medidas de dispersión de los datos

In [103…
```
data['smoking'].describe()
```

Out[103…
```
count    299.00000
mean       0.32107
std        0.46767
min        0.00000
25%        0.00000
50%        0.00000
75%        1.00000
max        1.00000
Name: smoking, dtype: float64
```

Promedio de los datos

In [104…
```
data['smoking'].mean()
```

Out[104…
```
0.3210702341137124
```

Mediana de los datos

In [105…
```
data['smoking'].median()
```

Out[105…
```
0.0
```

Desviación estandar de los datos

In [106…
```
data['smoking'].std()
```

Out[106…
```
0.4676704280567721
```

Rango de los datos

In [107…
```
Ran2=data['smoking'].max()-data['smoking'].min()
print(Ran2)
```

```
1
```

Conclusión de datos estadísticos descriptivos.

La minoría de los pacientes fuman esto se debe a que el valor de la media se aproxima más a 0 que a 1.

In [108…
```
type(data.time)
```

Out[108…
```
pandas.core.series.Series
```

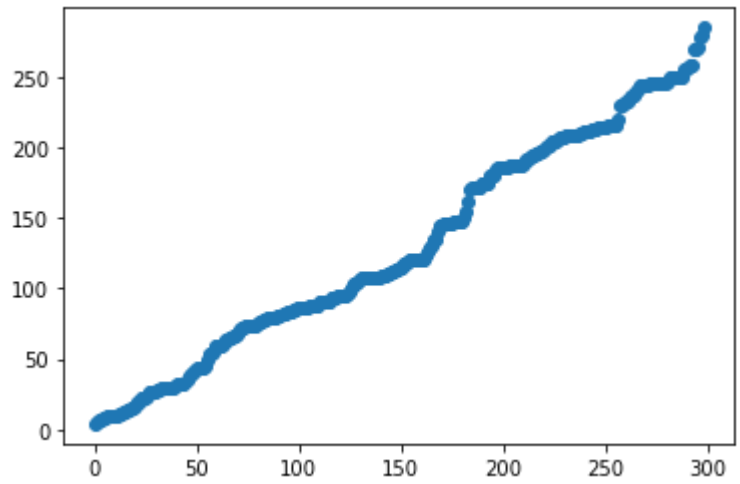
Tiempo de seguimiento de los pacientes

In [109…
```
data.time
```

Out[109…
```
0        4
1        6
2        7
3        7
4        8
      ...
294    270
295    271
296    278
297    280
298    285
Name: time, Length: 299, dtype: int64
```

Representación gráfica de el tiempo de seguimiento de los pacientes.

In [110…
```
plt.plot(data.time,'o')
plt.show()
```



Tipos de datos

In [111…
```
pd.unique(data['time'])
```

Out[111…
```
array([  4,   6,   7,   8,  10,  11,  12,  13,  14,  15,  16,  20,  22,
        23,  24,  26,  27,  28,  29,  30,  31,  32,  33,  35,  38,  40,
        41,  42,  43,  44,  45,  50,  54,  55,  59,  60,  61,  63,  64,
        65,  66,  67,  68,  71,  72,  73,  74,  75,  76,  77,  78,  79,
        80,  82,  83,  85,  86,  87,  88,  90,  91,  94,  95,  96,  97,
       100, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 115, 117,
       118, 119, 120, 121, 123, 126, 129, 130, 134, 135, 140, 145, 146,
       147, 148, 150, 154, 162, 170, 171, 172, 174, 175, 180, 185, 186,
       187, 188, 192, 193, 194, 195, 196, 197, 198, 200, 201, 205, 206,
       207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 220, 230, 231,
       233, 235, 237, 240, 241, 244, 245, 246, 247, 250, 256, 257, 258,
       270, 271, 278, 280, 285], dtype=int64)
```

Medidas de dispersión de los datos

In [112…
```
data['time'].describe()
```

Out[112…
```
count    299.000000
mean     130.260870
std       77.614208
min        4.000000
25%       73.000000
50%      115.000000
75%      203.000000
max      285.000000
Name: time, dtype: float64
```

Promedio de los datos

In [113…
```
data['time'].mean()
```

Out[113…
```
130.2608695652174
```

Mediana de los datos

In [114…
```
data['time'].median()
```

Out[114…
```
115.0
```

Desviación estandar de los datos

In [115…
```
data['time'].std()
```

Out[115…
```
77.61420795029342
```

Rango de los datos

In [116…
```
Ran3=data['time'].max()-data['time'].min()
print(Ran3)
```

```
281
```

Conclusión de datos estadísticos descriptivos.

El promedio de la cantidad de tiempo de seguimiento de los pacientes es de 130.260

El rango de la cantidad de tiempo de seguimiento de los pacientes es de 281

La mediana de los datos es 115

La desviación estandar de la cantidad de tiempo de seguimiento de los pacientes es de 77.6142

In [117…
```
type(data.DEATH_EVENT)
```

Out[117…
```
pandas.core.series.Series
```

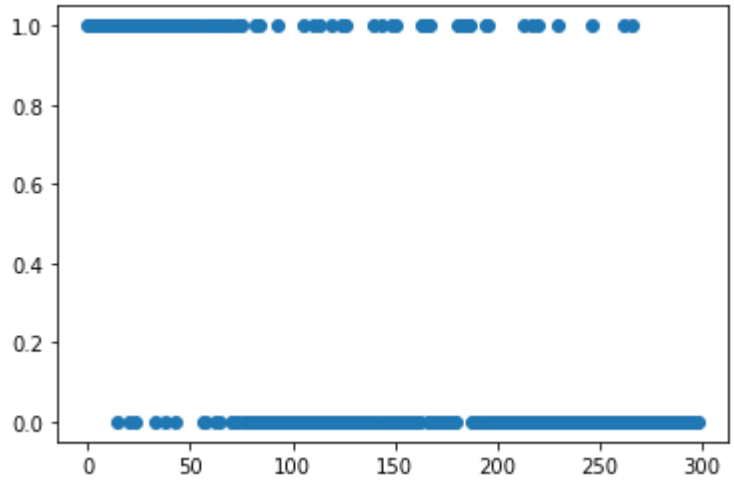
Muertes de los pacientes

In [118…
```
data.DEATH_EVENT
```

Out[118…
```
0      1
1      1
2      1
3      1
4      1
      ..
294    0
295    0
296    0
297    0
298    0
Name: DEATH_EVENT, Length: 299, dtype: int64
```

Representación gráfica de muerte de los pacientes.

In [119…
```
plt.plot(data.DEATH_EVENT,'o')
plt.show()
```



Tipos de datos

In [120…
```
pd.unique(data['DEATH_EVENT'])
```

Out[120…
```
array([1, 0], dtype=int64)
```

Medidas de dispersión de los datos

In [121…
```
data['DEATH_EVENT'].describe()
```

Out[121…
```
count    299.00000
mean       0.32107
std        0.46767
min        0.00000
25%        0.00000
50%        0.00000
75%        1.00000
max        1.00000
Name: DEATH_EVENT, dtype: float64
```

Promedio de los datos

In [122…
```
data['DEATH_EVENT'].mean()
```

Out[122…
```
0.3210702341137124
```

Mediana de los datos

In [123…
```
data['DEATH_EVENT'].median()
```

Out[123…
```
0.0
```

Desviación estandar de los datos

In [124…
```
data['DEATH_EVENT'].std()
```

Out[124…
```
0.4676704280567721
```

Rango de los datos

In [125…
```
Ran=data['DEATH_EVENT'].max()-data['DEATH_EVENT'].min()
print(Ran)
```

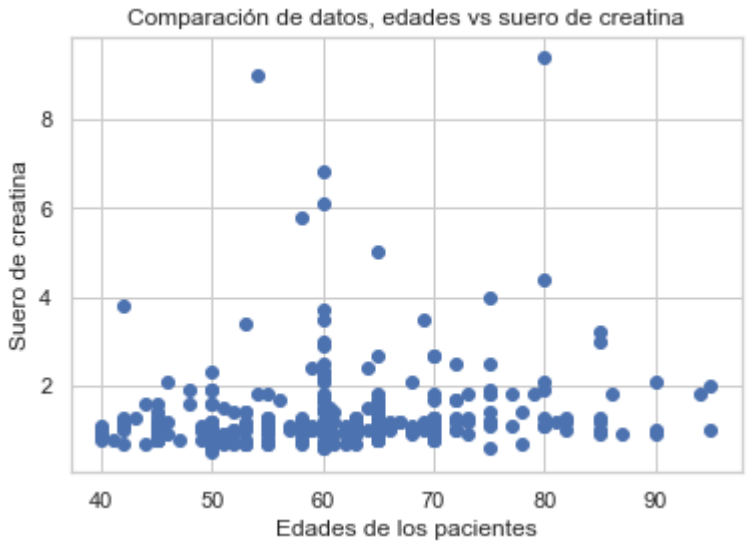
```
1
```

Conclusión de datos estadísticos descriptivos.

La minoría de los pacientes ham fallecido se debe a que el valor de la media se aproxima más a 0 que a 1.

Comparación de datos de edades contra suero de creatina

In [164…
```
plt.plot(data.age,data.serum_creatinine,'o')
plt.title("Comparación de datos, edades vs suero de creatina")
plt.xlabel("Edades de los pacientes")
plt.ylabel("Suero de creatina")
plt.show()
```

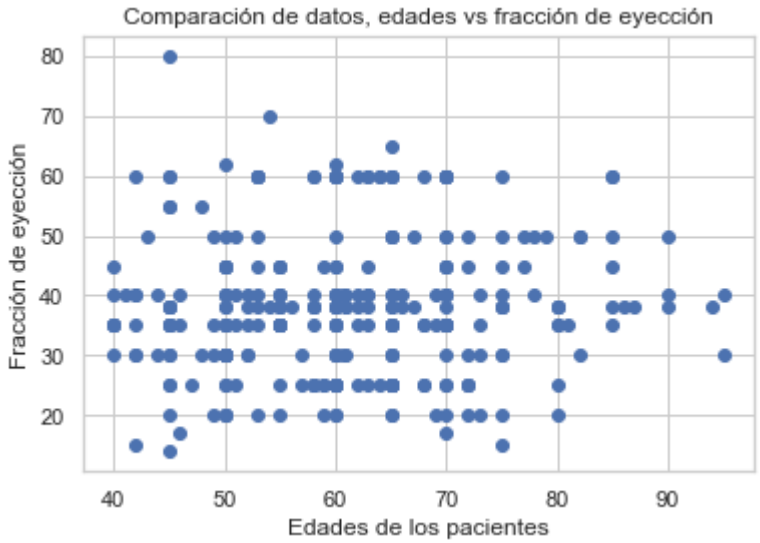


Conclusión de gráfica

Se puede inferir a partir de la gráfica que no hay una correlación directa mayor entre la edad de los pacientes y la cantidad de suero de creatina que producen sus cuerpos, esta relación es debil

Comparación de datos de edades contra la cantidad de fracción de eyección

In [165…
```
plt.plot(data.age,data.ejection_fraction,'o')
plt.title("Comparación de datos, edades vs fracción de eyección")
plt.xlabel("Edades de los pacientes")
plt.ylabel("Fracción de eyección")
plt.show()
```



Conclusión de gráfica

Se puede inferir a partir de la gráfica que no hay una correlación directa mayor entre la edad de los pacientes y la cantidad de fracción de eyección que producen sus cuerpos, esta relación es debil

Mapas de calor y boxplots de la base de datos

In [127…
```
import pandas as pd
import seaborn as sb
import numpy as np; np.random.seed(0)
import matplotlib.pyplot as plt
data=pd.read_csv('heart_failure_clinical_records_dataset.csv')
data.shape
```

Out[127… (299, 13)

In [128…
```
data.head
```

Out[128…
```
<bound method NDFrame.head of       age  anaemia  creatinine_phosphokinase  diabetes  ej
ection_fraction  \
0    75.0        0                       582         0                 20
1    55.0        0                      7861         0                 38
```

```
2    65.0        0                       146         0                 20
3    50.0        1                       111         0                 20
4    65.0        1                       160         1                 20
..    ...      ...                       ...       ...                ...
294  62.0        0                        61         1                 38
295  55.0        0                      1820         0                 38
296  45.0        0                      2060         1                 60
297  45.0        0                      2413         0                 38
298  50.0        0                       196         0                 45

     high_blood_pressure  platelets  serum_creatinine  serum_sodium  sex  \
0                      1  265000.00               1.9           130    1
1                      0  263358.03               1.1           136    1
2                      0  162000.00               1.3           129    1
3                      0  210000.00               1.9           137    1
4                      0  327000.00               2.7           116    0
..                   ...        ...               ...           ...  ...
294                    1  155000.00               1.1           143    1
295                    0  270000.00               1.2           139    0
296                    0  742000.00               0.8           138    0
297                    0  140000.00               1.4           140    1
298                    0  395000.00               1.6           136    1

     smoking  time  DEATH_EVENT
0          0     4            1
1          0     6            1
2          1     7            1
3          0     7            1
4          0     8            1
..       ...   ...          ...
294        1   270            0
295        0   271            0
296        0   278            0
297        1   280            0
298        1   285            0

[299 rows x 13 columns]>
```

Descripción estadística de los datos

In [129…
```
data.describe()
```

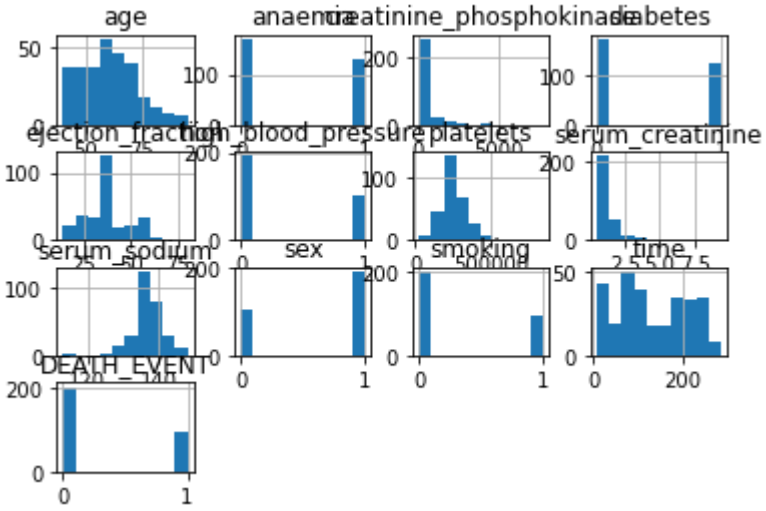
Out[129…

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressu
count	299.000000	299.000000	299.000000	299.000000	299.000000	299.000(
mean	60.833893	0.431438	581.839465	0.418060	38.083612	0.351
std	11.894809	0.496107	970.287881	0.494067	11.834841	0.478
min	40.000000	0.000000	23.000000	0.000000	14.000000	0.000(
25%	51.000000	0.000000	116.500000	0.000000	30.000000	0.000(
50%	60.000000	0.000000	250.000000	0.000000	38.000000	0.000(
75%	70.000000	1.000000	582.000000	1.000000	45.000000	1.000(
max	95.000000	1.000000	7861.000000	1.000000	80.000000	1.000(

Histograma de los campos

In [130…
```
data.drop([0,1]).hist()
```

Out[130…
```
array([[<AxesSubplot:title={'center':'age'}>,
        <AxesSubplot:title={'center':'anaemia'}>,
        <AxesSubplot:title={'center':'creatinine_phosphokinase'}>,
        <AxesSubplot:title={'center':'diabetes'}>],
       [<AxesSubplot:title={'center':'ejection_fraction'}>,
        <AxesSubplot:title={'center':'high_blood_pressure'}>,
        <AxesSubplot:title={'center':'platelets'}>,
        <AxesSubplot:title={'center':'serum_creatinine'}>],
       [<AxesSubplot:title={'center':'serum_sodium'}>,
        <AxesSubplot:title={'center':'sex'}>,
        <AxesSubplot:title={'center':'smoking'}>,
        <AxesSubplot:title={'center':'time'}>],
       [<AxesSubplot:title={'center':'DEATH_EVENT'}>, <AxesSubplot:>,
        <AxesSubplot:>, <AxesSubplot:>]], dtype=object)
```



Correlación de los datos por método de pearson

In [131…
```
data.corr(method='pearson')
```

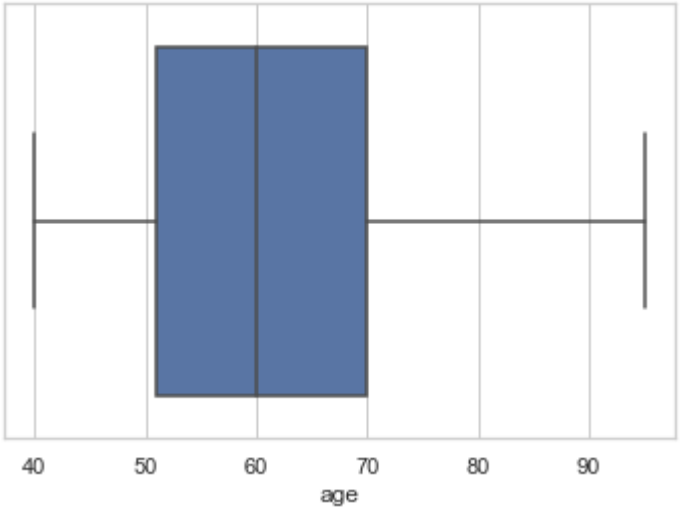
Out[131…

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	hi
age	1.000000	0.088006	-0.081584	-0.101012	0.060098	
anaemia	0.088006	1.000000	-0.190741	-0.012729	0.031557	
creatinine_phosphokinase	-0.081584	-0.190741	1.000000	-0.009639	-0.044080	
diabetes	-0.101012	-0.012729	-0.009639	1.000000	-0.004850	
ejection_fraction	0.060098	0.031557	-0.044080	-0.004850	1.000000	
high_blood_pressure	0.093289	0.038182	-0.070590	-0.012732	0.024445	
platelets	-0.052354	-0.043786	0.024463	0.092193	0.072177	
serum_creatinine	0.159187	0.052174	-0.016408	-0.046975	-0.011302	
serum_sodium	-0.045966	0.041882	0.059550	-0.089551	0.175902	
sex	0.065430	-0.094769	0.079791	-0.157730	-0.148386	
smoking	0.018668	-0.107290	0.002421	-0.147173	-0.067315	
time	-0.224068	-0.141414	-0.009346	0.033726	0.041729	
DEATH_EVENT	0.253729	0.066270	0.062728	-0.001943	-0.268603	

Gráfica de caja y bigotes

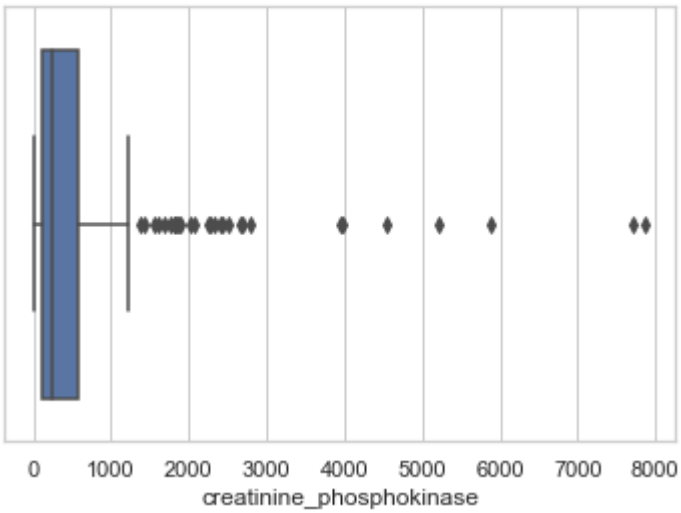
Edad

In [132…
```
sb.set_theme(style="whitegrid")
ax=sb.boxplot(x=data["age"])
```



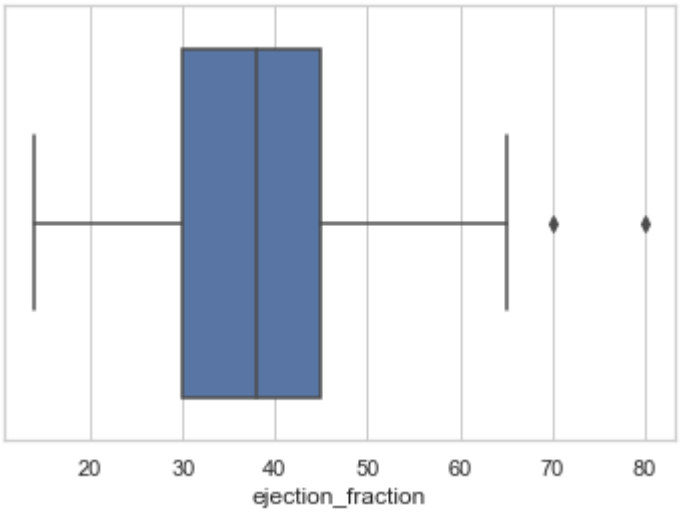
creatinine phosphokinase

In [133…
```
sb.set_theme(style="whitegrid")
ax=sb.boxplot(x=data["creatinine_phosphokinase"])
```



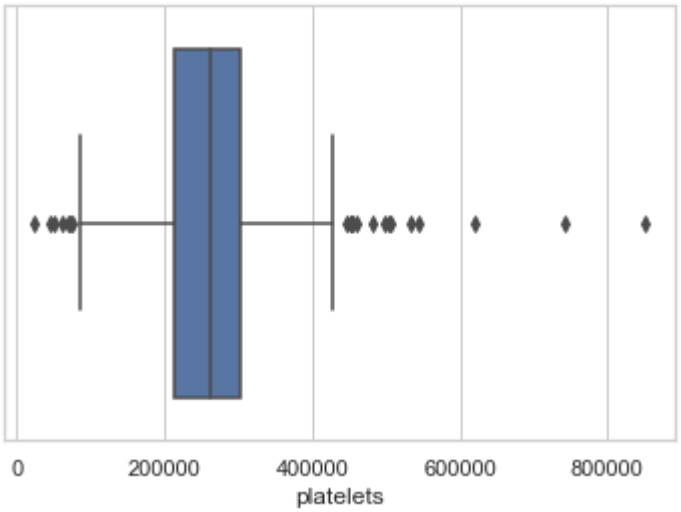
Fracción de eyección

In [134…
```
sb.set_theme(style="whitegrid")
ax=sb.boxplot(x=data["ejection_fraction"])
```



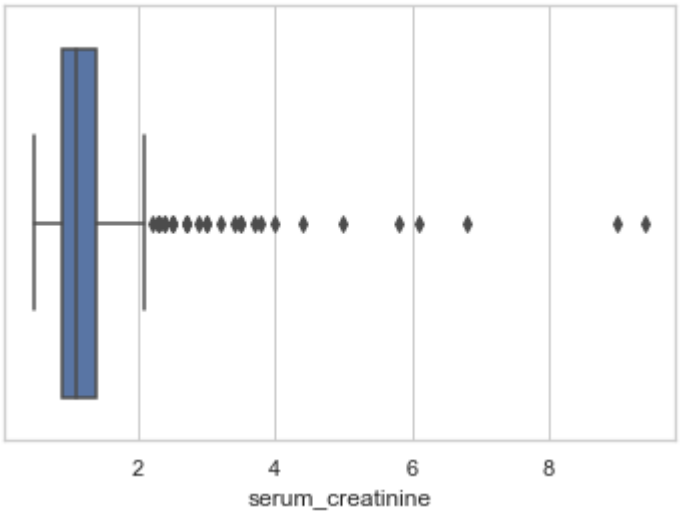
Plaquetas

In [135…
```
sb.set_theme(style="whitegrid")
ax=sb.boxplot(x=data["platelets"])
```



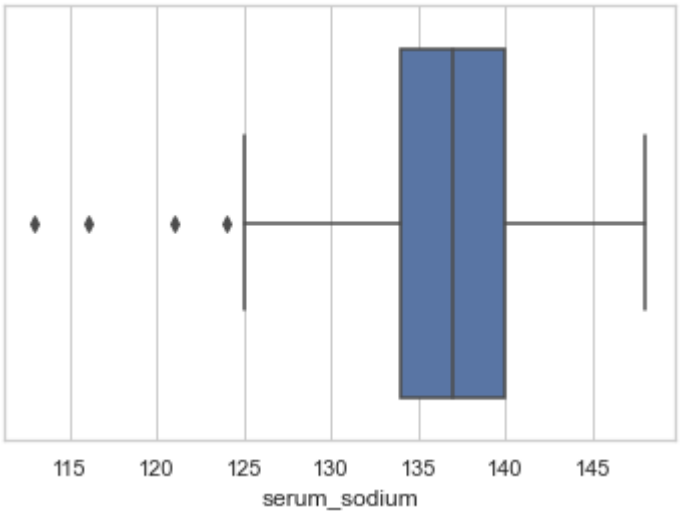
Suero de creatina

In [136…
```
sb.set_theme(style="whitegrid")
ax=sb.boxplot(x=data["serum_creatinine"])
```



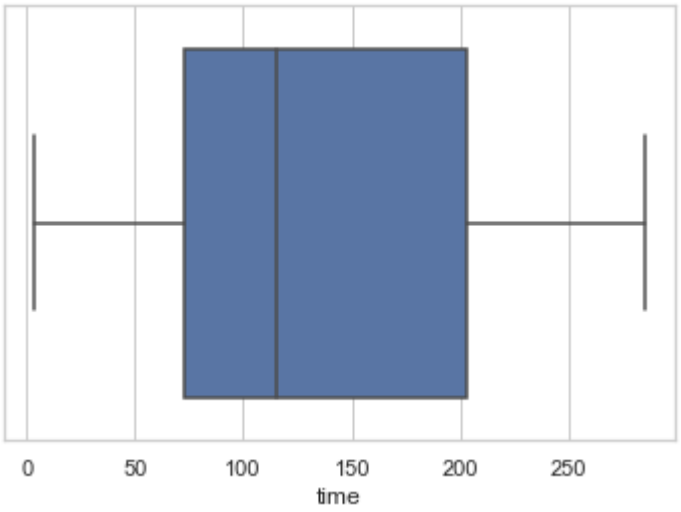
Suero de sodio

In [137…
```
sb.set_theme(style="whitegrid")
ax=sb.boxplot(x=data["serum_sodium"])
```



Tiempo de seguimiento del tratamiento

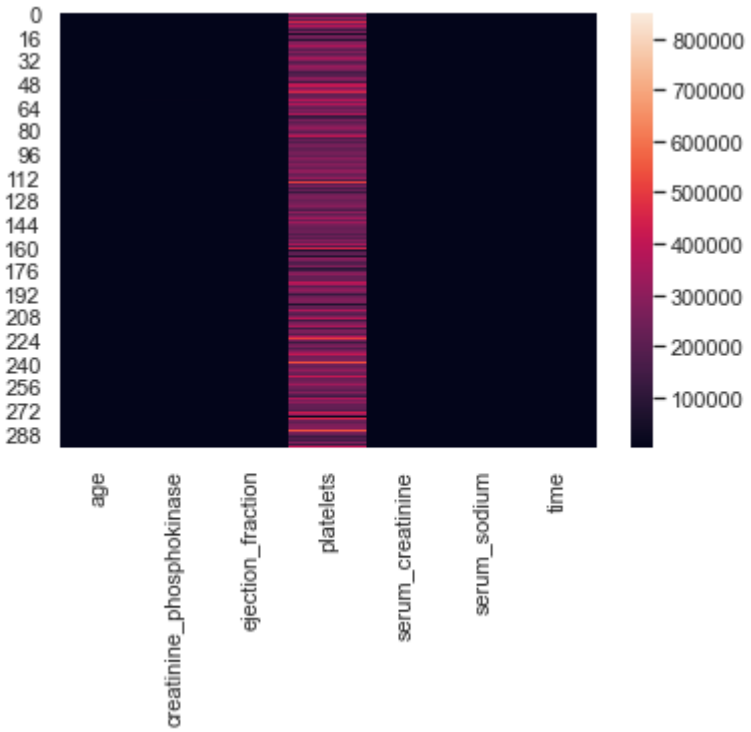
In [138…
```
sb.set_theme(style="whitegrid")
ax=sb.boxplot(x=data["time"])
```



Mapas de calor

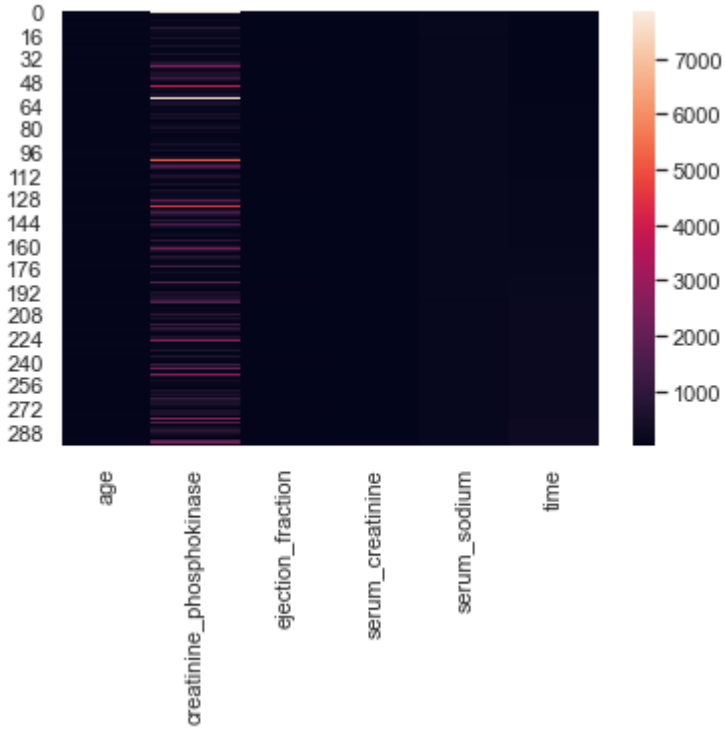
Con plaquetas

In [139…
```
Heart_health = pd.read_csv('heart_failure_clinical_records_dataset_new.csv')
ax=sb.heatmap(Heart_health)
```



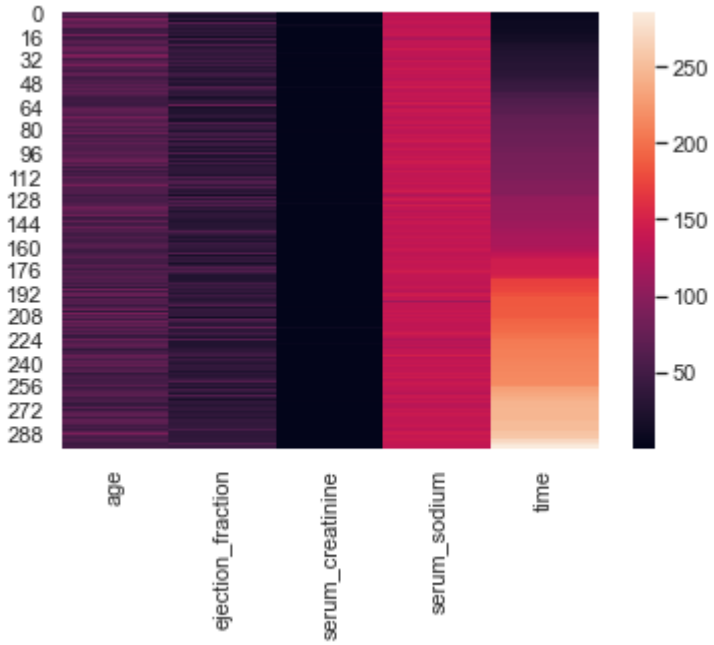
Sin plaquetas

In [140…
```
Heart_health2 = pd.read_csv('heart_failure_clinical_records_dataset_new_2 - copia.csv')
ax=sb.heatmap(Heart_health2)
```



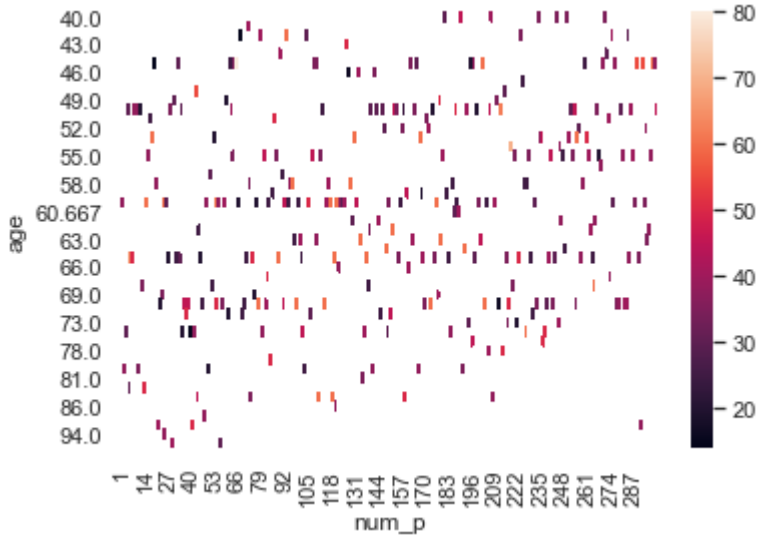
Sin creatina

In [141…
```
Heart_health3 = pd.read_csv('heart_failure_clinical_records_dataset_new_3.csv')
ax=sb.heatmap(Heart_health3)
```



Comparación de la edad de los pacientes con su número de fracción de eyección

In [142…
```
Heart_H=pd.read_csv('heart_failure_clinical_records_dataset_new_4.csv')
Heart_H= Heart_H.pivot('age','num_p','ejection_fraction')
ax=sb.heatmap(Heart_H)
```



Preguntas detonadoras de análisis

¿Hay alguna variable que no aporta información? Todas las variables booleanas su aportación es relativamente nula ya que tienen una clasificación bastante generalizada la cual se categoriza en dos secciones 1 u 0 y pues en nuestro analisis su valor informativo es realmente bajo. Estas siendo:anaemia, high_blood_pressure,sex,DEATH_EVENT Cabe aclarar que apesar de que si hay un pequeño aporte de información por parte de estas variable, su utilidad para un análisis de datos es casi nulo.

Si tuvieras que eliminar variables, ¿cuáles quitarías y por qué? Creatinine Phosphokinase: dispersión de datos y valores atípicos, poca correlación con los datos. Plaquetas: dispersión de datos y valores atípicos. Sexo: valores atípicos e irrelevantes para la investigación. Todas estas variables fueron seleccionadas para poder ser una opción para eliminar debido a que generan o tienen un alto impacto en la fiabilidad de nuestro análisis. Si bien es importante agregar que al tener cada una de estas variables tiene una alta dispersión y varios valores atípicos podemos decir que nuestro proceso es altamente afectado

¿Existen variables que tengan datos extraños? las variables anaemia, high_blood_pressure,sex,DEATH_EVENT,diabetes,smoking presentan valores booleanos, lo cual significa que no pueden ser usadas para nuestra evaluación de los datos

Si comparas las variables, ¿todas están en rangos similares? ¿Crees que esto afecte? No, si existe una variabilidad en los rangos de las variables y si consideramos que esto puede afectar nuestro proceso porque si bien dicho proceso tiene un rango fijo y al cada variable tener su propio rango esto genera cierta inestabiliedad en este mismo. Aunque si dividimos las variables por dos clasificaciones entonces si comparten rangos comunes, las dos clasificaciones son datos mayores a 2000 y los datos menores a 1000.

¿Puedes encontrar grupos qué se parezcan? ¿Qué grupos son estos? Si, existen dos tipos de grupos en como se clasifican los datos. Tenemos los datos que se clasifican entre 0 y 1 y los datos que se clasifican en función a como transcurre el proceso. Y dentre de estos datos están las clasificaciones previamente mencionadas donde sus datos superaban o no las 2000 unidades.

Patrones con K-means de la base de datos

Importar librerías para el desarrollo del entregable

In [143…
```
import pandas as pd
import seaborn as sb
import numpy as np; np.random.seed(0)
import matplotlib.pyplot as plt
import sklearn
from sklearn.cluster import KMeans
from mpl_toolkits.mplot3d import Axes3D
from sklearn.preprocessing import scale
import sklearn.metrics as sm
from sklearn import datasets
from sklearn.metrics import confusion_matrix, classification_report
```

In [144…
```
health_heart= pd.read_csv('heart_failure_clinical_records_dataset_new.csv')
X=scale(health_heart)
lista = []
for i in range(299):
     lista.append(i+1)
Y=pd.DataFrame(lista)
variable_names=health_heart.age
X[0:10,]
```

Out[144…
```
array([[ 1.19294523e+00,  1.65728387e-04, -1.53055953e+00,
         1.68164843e-02,  4.90056987e-01, -1.50403612e+00,
        -1.62950241e+00],
       [-4.91279276e-01,  7.51463953e+00, -7.07675018e-03,
         7.53566018e-09, -2.84552352e-01, -1.41976151e-01,
        -1.60369074e+00],
       [ 3.50832977e-01, -4.49938761e-01, -1.53055953e+00,
        -1.03807313e+00, -9.09000174e-02, -1.73104612e+00,
        -1.59078490e+00],
       [-9.12335403e-01, -4.86071002e-01, -1.53055953e+00,
        -5.46474088e-01,  4.90056987e-01,  8.50338444e-02,
        -1.59078490e+00],
       [ 3.50832977e-01, -4.35485864e-01, -1.53055953e+00,
         6.51798584e-01,  1.26466633e+00, -4.68217606e+00,
        -1.57787906e+00],
       [ 2.45611361e+00, -5.52141386e-01,  1.62199114e-01,
        -6.07923969e-01,  6.83709322e-01, -1.05001613e+00,
        -1.57787906e+00],
       [ 1.19294523e+00, -3.46703786e-01, -1.95374919e+00,
        -1.39653077e+00, -1.87726185e-01,  8.50338444e-02,
        -1.55206738e+00],
       [-7.02231493e-02, -2.75471654e-01,  1.85495776e+00,
         1.95248772e+00, -2.84552352e-01, -1.27702613e+00,
        -1.55206738e+00],
       [ 3.50832977e-01, -4.38582914e-01,  2.27814742e+00,
         7.53566018e-09,  1.02752318e-01,  3.12043840e-01,
        -1.55206738e+00],
       [ 1.61400136e+00, -4.73682805e-01, -2.60990546e-01,
         1.27653904e+00,  7.75201955e+00, -8.23006137e-01,
        -1.55206738e+00]])
```

Construcción del modelo

In [145…
```
clustering=KMeans(n_clusters=7,random_state=0)
clustering.fit(X)
```

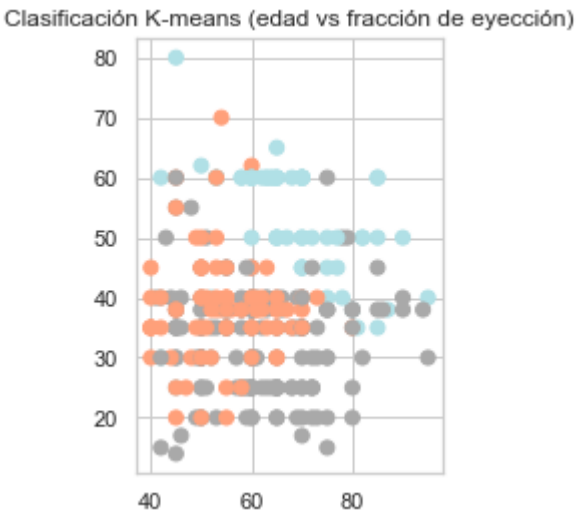
Out[145… KMeans(n_clusters=7, random_state=0)

Salida del modelo

In [146…
```
heart_df=pd.DataFrame(health_heart)
heart_df.columns=["age","creatinine_phosphokinase","ejection_fraction","platelets","ser
Y.columns=["age"]
```

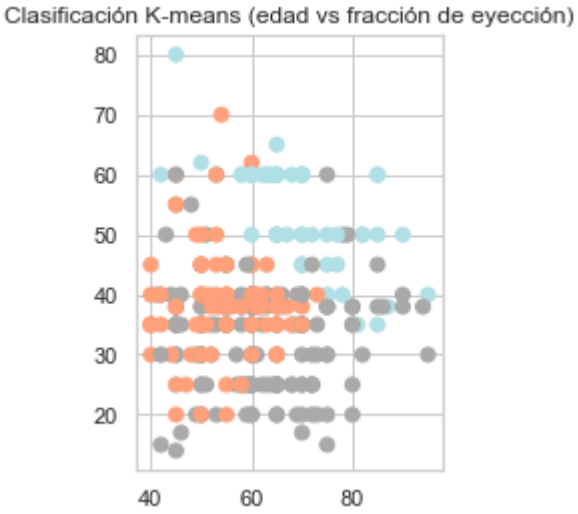
In [147…
```
color_theme=np.array(['darkgray','lightsalmon','powderblue','darkgray','lightsalmon','p
#plt.subplot(1,2,1)
#plt.scatter(x=heart_df.age,y=heart_df.ejection_fraction,c=color_theme[lista],s=50)
#plt.title("Clasificación actual")


plt.subplot(1,2,2)
plt.scatter(x=heart_df.age,y=heart_df.ejection_fraction,c=color_theme[clustering.labels
plt.title("Clasificación K-means (edad vs fracción de eyección)")
```

Out[147… Text(0.5, 1.0, 'Clasificación K-means (edad vs fracción de eyección)')



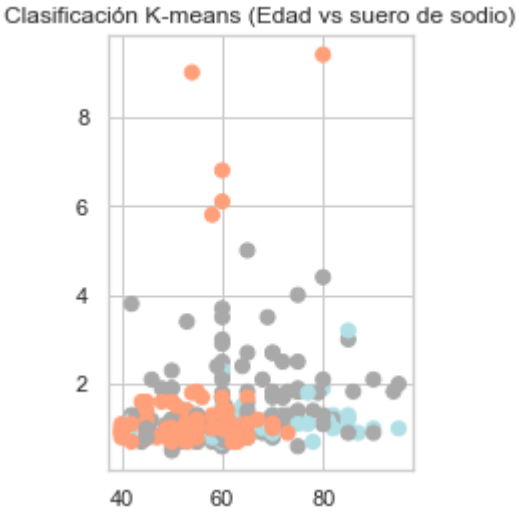
In [157…
```
relabel = np.choose (clustering.labels_,[2,1,1,1,1,1,1,1]).astype(np.int64)
plt.subplot(1,2,2)
plt.scatter(x=heart_df.age,y=heart_df.ejection_fraction,c=color_theme[clustering.labels
plt.title("Clasificación K-means (edad vs fracción de eyección)")
```

Out[157… Text(0.5, 1.0, 'Clasificación K-means (edad vs fracción de eyección)')



In [162…
```
plt.subplot(1,2,2)
plt.scatter(x=heart_df.age,y=heart_df.serum_creatinine,c=color_theme[clustering.labels_
plt.title("Clasificación K-means (Edad vs suero de sodio)")
```

Out[162… Text(0.5, 1.0, 'Clasificación K-means (Edad vs suero de sodio)')



In [161…
```
relabel2 = np.choose (clustering.labels_,[2,1,1,1,1,1,1]).astype(np.int64)
plt.subplot(1,2,2)
plt.scatter(x=heart_df.age,y=heart_df.serum_creatinine,c=color_theme[clustering.labels_
plt.title("Clasificación K-means (Edad vs suero de sodio)")
```

Out[161… Text(0.5, 1.0, 'Clasificación K-means (Edad vs suero de sodio)')

Evaluación de los datos

In [158…
```
print (classification_report(lista, relabel))
```

```
              precision    recall  f1-score   support

           1       0.00      1.00      0.01         1
           2       0.00      0.00      0.00         1
           3       0.00      0.00      0.00         1
           4       0.00      0.00      0.00         1
           5       0.00      0.00      0.00         1
           6       0.00      0.00      0.00         1
           7       0.00      0.00      0.00         1
           8       0.00      0.00      0.00         1
           9       0.00      0.00      0.00         1
          10       0.00      0.00      0.00         1
          11       0.00      0.00      0.00         1
          12       0.00      0.00      0.00         1
          13       0.00      0.00      0.00         1
          14       0.00      0.00      0.00         1
          15       0.00      0.00      0.00         1
          16       0.00      0.00      0.00         1
          17       0.00      0.00      0.00         1
          18       0.00      0.00      0.00         1
          19       0.00      0.00      0.00         1
          20       0.00      0.00      0.00         1
          21       0.00      0.00      0.00         1
          22       0.00      0.00      0.00         1
          23       0.00      0.00      0.00         1
          24       0.00      0.00      0.00         1
          25       0.00      0.00      0.00         1
          26       0.00      0.00      0.00         1
          27       0.00      0.00      0.00         1
          28       0.00      0.00      0.00         1
          29       0.00      0.00      0.00         1
          30       0.00      0.00      0.00         1
          31       0.00      0.00      0.00         1
          32       0.00      0.00      0.00         1
          33       0.00      0.00      0.00         1
          34       0.00      0.00      0.00         1
          35       0.00      0.00      0.00         1
          36       0.00      0.00      0.00         1
          37       0.00      0.00      0.00         1
          38       0.00      0.00      0.00         1
```

39	0.00	0.00	0.00	1
40	0.00	0.00	0.00	1
41	0.00	0.00	0.00	1
42	0.00	0.00	0.00	1
43	0.00	0.00	0.00	1
44	0.00	0.00	0.00	1
45	0.00	0.00	0.00	1
46	0.00	0.00	0.00	1
47	0.00	0.00	0.00	1
48	0.00	0.00	0.00	1
49	0.00	0.00	0.00	1
50	0.00	0.00	0.00	1
51	0.00	0.00	0.00	1
52	0.00	0.00	0.00	1
53	0.00	0.00	0.00	1
54	0.00	0.00	0.00	1
55	0.00	0.00	0.00	1
56	0.00	0.00	0.00	1
57	0.00	0.00	0.00	1
58	0.00	0.00	0.00	1
59	0.00	0.00	0.00	1
60	0.00	0.00	0.00	1
61	0.00	0.00	0.00	1
62	0.00	0.00	0.00	1
63	0.00	0.00	0.00	1
64	0.00	0.00	0.00	1
65	0.00	0.00	0.00	1
66	0.00	0.00	0.00	1
67	0.00	0.00	0.00	1
68	0.00	0.00	0.00	1
69	0.00	0.00	0.00	1
70	0.00	0.00	0.00	1
71	0.00	0.00	0.00	1
72	0.00	0.00	0.00	1
73	0.00	0.00	0.00	1
74	0.00	0.00	0.00	1
75	0.00	0.00	0.00	1
76	0.00	0.00	0.00	1
77	0.00	0.00	0.00	1
78	0.00	0.00	0.00	1
79	0.00	0.00	0.00	1
80	0.00	0.00	0.00	1
81	0.00	0.00	0.00	1
82	0.00	0.00	0.00	1
83	0.00	0.00	0.00	1
84	0.00	0.00	0.00	1
85	0.00	0.00	0.00	1
86	0.00	0.00	0.00	1
87	0.00	0.00	0.00	1
88	0.00	0.00	0.00	1
89	0.00	0.00	0.00	1
90	0.00	0.00	0.00	1
91	0.00	0.00	0.00	1
92	0.00	0.00	0.00	1
93	0.00	0.00	0.00	1
94	0.00	0.00	0.00	1
95	0.00	0.00	0.00	1
96	0.00	0.00	0.00	1
97	0.00	0.00	0.00	1
98	0.00	0.00	0.00	1
99	0.00	0.00	0.00	1
100	0.00	0.00	0.00	1
101	0.00	0.00	0.00	1
102	0.00	0.00	0.00	1
103	0.00	0.00	0.00	1

104	0.00	0.00	0.00	1
105	0.00	0.00	0.00	1
106	0.00	0.00	0.00	1
107	0.00	0.00	0.00	1
108	0.00	0.00	0.00	1
109	0.00	0.00	0.00	1
110	0.00	0.00	0.00	1
111	0.00	0.00	0.00	1
112	0.00	0.00	0.00	1
113	0.00	0.00	0.00	1
114	0.00	0.00	0.00	1
115	0.00	0.00	0.00	1
116	0.00	0.00	0.00	1
117	0.00	0.00	0.00	1
118	0.00	0.00	0.00	1
119	0.00	0.00	0.00	1
120	0.00	0.00	0.00	1
121	0.00	0.00	0.00	1
122	0.00	0.00	0.00	1
123	0.00	0.00	0.00	1
124	0.00	0.00	0.00	1
125	0.00	0.00	0.00	1
126	0.00	0.00	0.00	1
127	0.00	0.00	0.00	1
128	0.00	0.00	0.00	1
129	0.00	0.00	0.00	1
130	0.00	0.00	0.00	1
131	0.00	0.00	0.00	1
132	0.00	0.00	0.00	1
133	0.00	0.00	0.00	1
134	0.00	0.00	0.00	1
135	0.00	0.00	0.00	1
136	0.00	0.00	0.00	1
137	0.00	0.00	0.00	1
138	0.00	0.00	0.00	1
139	0.00	0.00	0.00	1
140	0.00	0.00	0.00	1
141	0.00	0.00	0.00	1
142	0.00	0.00	0.00	1
143	0.00	0.00	0.00	1
144	0.00	0.00	0.00	1
145	0.00	0.00	0.00	1
146	0.00	0.00	0.00	1
147	0.00	0.00	0.00	1
148	0.00	0.00	0.00	1
149	0.00	0.00	0.00	1
150	0.00	0.00	0.00	1
151	0.00	0.00	0.00	1
152	0.00	0.00	0.00	1
153	0.00	0.00	0.00	1
154	0.00	0.00	0.00	1
155	0.00	0.00	0.00	1
156	0.00	0.00	0.00	1
157	0.00	0.00	0.00	1
158	0.00	0.00	0.00	1
159	0.00	0.00	0.00	1
160	0.00	0.00	0.00	1
161	0.00	0.00	0.00	1
162	0.00	0.00	0.00	1
163	0.00	0.00	0.00	1
164	0.00	0.00	0.00	1
165	0.00	0.00	0.00	1
166	0.00	0.00	0.00	1
167	0.00	0.00	0.00	1
168	0.00	0.00	0.00	1

169	0.00	0.00	0.00	1
170	0.00	0.00	0.00	1
171	0.00	0.00	0.00	1
172	0.00	0.00	0.00	1
173	0.00	0.00	0.00	1
174	0.00	0.00	0.00	1
175	0.00	0.00	0.00	1
176	0.00	0.00	0.00	1
177	0.00	0.00	0.00	1
178	0.00	0.00	0.00	1
179	0.00	0.00	0.00	1
180	0.00	0.00	0.00	1
181	0.00	0.00	0.00	1
182	0.00	0.00	0.00	1
183	0.00	0.00	0.00	1
184	0.00	0.00	0.00	1
185	0.00	0.00	0.00	1
186	0.00	0.00	0.00	1
187	0.00	0.00	0.00	1
188	0.00	0.00	0.00	1
189	0.00	0.00	0.00	1
190	0.00	0.00	0.00	1
191	0.00	0.00	0.00	1
192	0.00	0.00	0.00	1
193	0.00	0.00	0.00	1
194	0.00	0.00	0.00	1
195	0.00	0.00	0.00	1
196	0.00	0.00	0.00	1
197	0.00	0.00	0.00	1
198	0.00	0.00	0.00	1
199	0.00	0.00	0.00	1
200	0.00	0.00	0.00	1
201	0.00	0.00	0.00	1
202	0.00	0.00	0.00	1
203	0.00	0.00	0.00	1
204	0.00	0.00	0.00	1
205	0.00	0.00	0.00	1
206	0.00	0.00	0.00	1
207	0.00	0.00	0.00	1
208	0.00	0.00	0.00	1
209	0.00	0.00	0.00	1
210	0.00	0.00	0.00	1
211	0.00	0.00	0.00	1
212	0.00	0.00	0.00	1
213	0.00	0.00	0.00	1
214	0.00	0.00	0.00	1
215	0.00	0.00	0.00	1
216	0.00	0.00	0.00	1
217	0.00	0.00	0.00	1
218	0.00	0.00	0.00	1
219	0.00	0.00	0.00	1
220	0.00	0.00	0.00	1
221	0.00	0.00	0.00	1
222	0.00	0.00	0.00	1
223	0.00	0.00	0.00	1
224	0.00	0.00	0.00	1
225	0.00	0.00	0.00	1
226	0.00	0.00	0.00	1
227	0.00	0.00	0.00	1
228	0.00	0.00	0.00	1
229	0.00	0.00	0.00	1
230	0.00	0.00	0.00	1
231	0.00	0.00	0.00	1
232	0.00	0.00	0.00	1
233	0.00	0.00	0.00	1

234	0.00	0.00	0.00	1
235	0.00	0.00	0.00	1
236	0.00	0.00	0.00	1
237	0.00	0.00	0.00	1
238	0.00	0.00	0.00	1
239	0.00	0.00	0.00	1
240	0.00	0.00	0.00	1
241	0.00	0.00	0.00	1
242	0.00	0.00	0.00	1
243	0.00	0.00	0.00	1
244	0.00	0.00	0.00	1
245	0.00	0.00	0.00	1
246	0.00	0.00	0.00	1
247	0.00	0.00	0.00	1
248	0.00	0.00	0.00	1
249	0.00	0.00	0.00	1
250	0.00	0.00	0.00	1
251	0.00	0.00	0.00	1
252	0.00	0.00	0.00	1
253	0.00	0.00	0.00	1
254	0.00	0.00	0.00	1
255	0.00	0.00	0.00	1
256	0.00	0.00	0.00	1
257	0.00	0.00	0.00	1
258	0.00	0.00	0.00	1
259	0.00	0.00	0.00	1
260	0.00	0.00	0.00	1
261	0.00	0.00	0.00	1
262	0.00	0.00	0.00	1
263	0.00	0.00	0.00	1
264	0.00	0.00	0.00	1
265	0.00	0.00	0.00	1
266	0.00	0.00	0.00	1
267	0.00	0.00	0.00	1
268	0.00	0.00	0.00	1
269	0.00	0.00	0.00	1
270	0.00	0.00	0.00	1
271	0.00	0.00	0.00	1
272	0.00	0.00	0.00	1
273	0.00	0.00	0.00	1
274	0.00	0.00	0.00	1
275	0.00	0.00	0.00	1
276	0.00	0.00	0.00	1
277	0.00	0.00	0.00	1
278	0.00	0.00	0.00	1
279	0.00	0.00	0.00	1
280	0.00	0.00	0.00	1
281	0.00	0.00	0.00	1
282	0.00	0.00	0.00	1
283	0.00	0.00	0.00	1
284	0.00	0.00	0.00	1
285	0.00	0.00	0.00	1
286	0.00	0.00	0.00	1
287	0.00	0.00	0.00	1
288	0.00	0.00	0.00	1
289	0.00	0.00	0.00	1
290	0.00	0.00	0.00	1
291	0.00	0.00	0.00	1
292	0.00	0.00	0.00	1
293	0.00	0.00	0.00	1
294	0.00	0.00	0.00	1
295	0.00	0.00	0.00	1
296	0.00	0.00	0.00	1
297	0.00	0.00	0.00	1
298	0.00	0.00	0.00	1

```
         299       0.00      0.00      0.00         1

    accuracy                           0.00       299
   macro avg       0.00      0.00      0.00       299
weighted avg       0.00      0.00      0.00       299
```

In [163…
```
print (classification_report(lista, relabel2))
```

```
              precision    recall  f1-score   support

           1       0.00      1.00      0.01         1
           2       0.00      0.00      0.00         1
           3       0.00      0.00      0.00         1
           4       0.00      0.00      0.00         1
           5       0.00      0.00      0.00         1
           6       0.00      0.00      0.00         1
           7       0.00      0.00      0.00         1
           8       0.00      0.00      0.00         1
           9       0.00      0.00      0.00         1
          10       0.00      0.00      0.00         1
          11       0.00      0.00      0.00         1
          12       0.00      0.00      0.00         1
          13       0.00      0.00      0.00         1
          14       0.00      0.00      0.00         1
          15       0.00      0.00      0.00         1
          16       0.00      0.00      0.00         1
          17       0.00      0.00      0.00         1
          18       0.00      0.00      0.00         1
          19       0.00      0.00      0.00         1
          20       0.00      0.00      0.00         1
          21       0.00      0.00      0.00         1
          22       0.00      0.00      0.00         1
          23       0.00      0.00      0.00         1
          24       0.00      0.00      0.00         1
          25       0.00      0.00      0.00         1
          26       0.00      0.00      0.00         1
          27       0.00      0.00      0.00         1
          28       0.00      0.00      0.00         1
          29       0.00      0.00      0.00         1
          30       0.00      0.00      0.00         1
          31       0.00      0.00      0.00         1
          32       0.00      0.00      0.00         1
          33       0.00      0.00      0.00         1
          34       0.00      0.00      0.00         1
          35       0.00      0.00      0.00         1
          36       0.00      0.00      0.00         1
          37       0.00      0.00      0.00         1
          38       0.00      0.00      0.00         1
          39       0.00      0.00      0.00         1
          40       0.00      0.00      0.00         1
          41       0.00      0.00      0.00         1
          42       0.00      0.00      0.00         1
          43       0.00      0.00      0.00         1
          44       0.00      0.00      0.00         1
          45       0.00      0.00      0.00         1
          46       0.00      0.00      0.00         1
          47       0.00      0.00      0.00         1
          48       0.00      0.00      0.00         1
          49       0.00      0.00      0.00         1
          50       0.00      0.00      0.00         1
          51       0.00      0.00      0.00         1
          52       0.00      0.00      0.00         1
          53       0.00      0.00      0.00         1
          54       0.00      0.00      0.00         1
```

55	0.00	0.00	0.00	1
56	0.00	0.00	0.00	1
57	0.00	0.00	0.00	1
58	0.00	0.00	0.00	1
59	0.00	0.00	0.00	1
60	0.00	0.00	0.00	1
61	0.00	0.00	0.00	1
62	0.00	0.00	0.00	1
63	0.00	0.00	0.00	1
64	0.00	0.00	0.00	1
65	0.00	0.00	0.00	1
66	0.00	0.00	0.00	1
67	0.00	0.00	0.00	1
68	0.00	0.00	0.00	1
69	0.00	0.00	0.00	1
70	0.00	0.00	0.00	1
71	0.00	0.00	0.00	1
72	0.00	0.00	0.00	1
73	0.00	0.00	0.00	1
74	0.00	0.00	0.00	1
75	0.00	0.00	0.00	1
76	0.00	0.00	0.00	1
77	0.00	0.00	0.00	1
78	0.00	0.00	0.00	1
79	0.00	0.00	0.00	1
80	0.00	0.00	0.00	1
81	0.00	0.00	0.00	1
82	0.00	0.00	0.00	1
83	0.00	0.00	0.00	1
84	0.00	0.00	0.00	1
85	0.00	0.00	0.00	1
86	0.00	0.00	0.00	1
87	0.00	0.00	0.00	1
88	0.00	0.00	0.00	1
89	0.00	0.00	0.00	1
90	0.00	0.00	0.00	1
91	0.00	0.00	0.00	1
92	0.00	0.00	0.00	1
93	0.00	0.00	0.00	1
94	0.00	0.00	0.00	1
95	0.00	0.00	0.00	1
96	0.00	0.00	0.00	1
97	0.00	0.00	0.00	1
98	0.00	0.00	0.00	1
99	0.00	0.00	0.00	1
100	0.00	0.00	0.00	1
101	0.00	0.00	0.00	1
102	0.00	0.00	0.00	1
103	0.00	0.00	0.00	1
104	0.00	0.00	0.00	1
105	0.00	0.00	0.00	1
106	0.00	0.00	0.00	1
107	0.00	0.00	0.00	1
108	0.00	0.00	0.00	1
109	0.00	0.00	0.00	1
110	0.00	0.00	0.00	1
111	0.00	0.00	0.00	1
112	0.00	0.00	0.00	1
113	0.00	0.00	0.00	1
114	0.00	0.00	0.00	1
115	0.00	0.00	0.00	1
116	0.00	0.00	0.00	1
117	0.00	0.00	0.00	1
118	0.00	0.00	0.00	1
119	0.00	0.00	0.00	1

120	0.00	0.00	0.00	1
121	0.00	0.00	0.00	1
122	0.00	0.00	0.00	1
123	0.00	0.00	0.00	1
124	0.00	0.00	0.00	1
125	0.00	0.00	0.00	1
126	0.00	0.00	0.00	1
127	0.00	0.00	0.00	1
128	0.00	0.00	0.00	1
129	0.00	0.00	0.00	1
130	0.00	0.00	0.00	1
131	0.00	0.00	0.00	1
132	0.00	0.00	0.00	1
133	0.00	0.00	0.00	1
134	0.00	0.00	0.00	1
135	0.00	0.00	0.00	1
136	0.00	0.00	0.00	1
137	0.00	0.00	0.00	1
138	0.00	0.00	0.00	1
139	0.00	0.00	0.00	1
140	0.00	0.00	0.00	1
141	0.00	0.00	0.00	1
142	0.00	0.00	0.00	1
143	0.00	0.00	0.00	1
144	0.00	0.00	0.00	1
145	0.00	0.00	0.00	1
146	0.00	0.00	0.00	1
147	0.00	0.00	0.00	1
148	0.00	0.00	0.00	1
149	0.00	0.00	0.00	1
150	0.00	0.00	0.00	1
151	0.00	0.00	0.00	1
152	0.00	0.00	0.00	1
153	0.00	0.00	0.00	1
154	0.00	0.00	0.00	1
155	0.00	0.00	0.00	1
156	0.00	0.00	0.00	1
157	0.00	0.00	0.00	1
158	0.00	0.00	0.00	1
159	0.00	0.00	0.00	1
160	0.00	0.00	0.00	1
161	0.00	0.00	0.00	1
162	0.00	0.00	0.00	1
163	0.00	0.00	0.00	1
164	0.00	0.00	0.00	1
165	0.00	0.00	0.00	1
166	0.00	0.00	0.00	1
167	0.00	0.00	0.00	1
168	0.00	0.00	0.00	1
169	0.00	0.00	0.00	1
170	0.00	0.00	0.00	1
171	0.00	0.00	0.00	1
172	0.00	0.00	0.00	1
173	0.00	0.00	0.00	1
174	0.00	0.00	0.00	1
175	0.00	0.00	0.00	1
176	0.00	0.00	0.00	1
177	0.00	0.00	0.00	1
178	0.00	0.00	0.00	1
179	0.00	0.00	0.00	1
180	0.00	0.00	0.00	1
181	0.00	0.00	0.00	1
182	0.00	0.00	0.00	1
183	0.00	0.00	0.00	1
184	0.00	0.00	0.00	1

185	0.00	0.00	0.00	1
186	0.00	0.00	0.00	1
187	0.00	0.00	0.00	1
188	0.00	0.00	0.00	1
189	0.00	0.00	0.00	1
190	0.00	0.00	0.00	1
191	0.00	0.00	0.00	1
192	0.00	0.00	0.00	1
193	0.00	0.00	0.00	1
194	0.00	0.00	0.00	1
195	0.00	0.00	0.00	1
196	0.00	0.00	0.00	1
197	0.00	0.00	0.00	1
198	0.00	0.00	0.00	1
199	0.00	0.00	0.00	1
200	0.00	0.00	0.00	1
201	0.00	0.00	0.00	1
202	0.00	0.00	0.00	1
203	0.00	0.00	0.00	1
204	0.00	0.00	0.00	1
205	0.00	0.00	0.00	1
206	0.00	0.00	0.00	1
207	0.00	0.00	0.00	1
208	0.00	0.00	0.00	1
209	0.00	0.00	0.00	1
210	0.00	0.00	0.00	1
211	0.00	0.00	0.00	1
212	0.00	0.00	0.00	1
213	0.00	0.00	0.00	1
214	0.00	0.00	0.00	1
215	0.00	0.00	0.00	1
216	0.00	0.00	0.00	1
217	0.00	0.00	0.00	1
218	0.00	0.00	0.00	1
219	0.00	0.00	0.00	1
220	0.00	0.00	0.00	1
221	0.00	0.00	0.00	1
222	0.00	0.00	0.00	1
223	0.00	0.00	0.00	1
224	0.00	0.00	0.00	1
225	0.00	0.00	0.00	1
226	0.00	0.00	0.00	1
227	0.00	0.00	0.00	1
228	0.00	0.00	0.00	1
229	0.00	0.00	0.00	1
230	0.00	0.00	0.00	1
231	0.00	0.00	0.00	1
232	0.00	0.00	0.00	1
233	0.00	0.00	0.00	1
234	0.00	0.00	0.00	1
235	0.00	0.00	0.00	1
236	0.00	0.00	0.00	1
237	0.00	0.00	0.00	1
238	0.00	0.00	0.00	1
239	0.00	0.00	0.00	1
240	0.00	0.00	0.00	1
241	0.00	0.00	0.00	1
242	0.00	0.00	0.00	1
243	0.00	0.00	0.00	1
244	0.00	0.00	0.00	1
245	0.00	0.00	0.00	1
246	0.00	0.00	0.00	1
247	0.00	0.00	0.00	1
248	0.00	0.00	0.00	1
249	0.00	0.00	0.00	1

250	0.00	0.00	0.00	1
251	0.00	0.00	0.00	1
252	0.00	0.00	0.00	1
253	0.00	0.00	0.00	1
254	0.00	0.00	0.00	1
255	0.00	0.00	0.00	1
256	0.00	0.00	0.00	1
257	0.00	0.00	0.00	1
258	0.00	0.00	0.00	1
259	0.00	0.00	0.00	1
260	0.00	0.00	0.00	1
261	0.00	0.00	0.00	1
262	0.00	0.00	0.00	1
263	0.00	0.00	0.00	1
264	0.00	0.00	0.00	1
265	0.00	0.00	0.00	1
266	0.00	0.00	0.00	1
267	0.00	0.00	0.00	1
268	0.00	0.00	0.00	1
269	0.00	0.00	0.00	1
270	0.00	0.00	0.00	1
271	0.00	0.00	0.00	1
272	0.00	0.00	0.00	1
273	0.00	0.00	0.00	1
274	0.00	0.00	0.00	1
275	0.00	0.00	0.00	1
276	0.00	0.00	0.00	1
277	0.00	0.00	0.00	1
278	0.00	0.00	0.00	1
279	0.00	0.00	0.00	1
280	0.00	0.00	0.00	1
281	0.00	0.00	0.00	1
282	0.00	0.00	0.00	1
283	0.00	0.00	0.00	1
284	0.00	0.00	0.00	1
285	0.00	0.00	0.00	1
286	0.00	0.00	0.00	1
287	0.00	0.00	0.00	1
288	0.00	0.00	0.00	1
289	0.00	0.00	0.00	1
290	0.00	0.00	0.00	1
291	0.00	0.00	0.00	1
292	0.00	0.00	0.00	1
293	0.00	0.00	0.00	1
294	0.00	0.00	0.00	1
295	0.00	0.00	0.00	1
296	0.00	0.00	0.00	1
297	0.00	0.00	0.00	1
298	0.00	0.00	0.00	1
299	0.00	0.00	0.00	1
accuracy			0.00	299
macro avg	0.00	0.00	0.00	299
weighted avg	0.00	0.00	0.00	299

Preguntas de análisis

1. ¿A qué se hace referencia al mencionar el "rango" de un análisis de datos?

2. ¿Los niveles de creatina infieren que los pacientes sean más propensos a las fallas cardiacas?

3. ¿La carencia de sodio es un factor que afecte el funcionamiento del corazón?
4. ¿Tener fallas en el corazón implica fallecer a causa de ello?
5. ¿Las personas mayores a 85 años son más proclives a presentar fallas cardíacas?

Respuestas:

1. Rango es el intervalo entre el valor máximo y el valor mínimo de nuestros datos obtenidos
2. Se puede inferir que la mayoría de los pacientes presentan niveles de creatina en un intervalo de 0-1000, así que si podría ser una constante que identifique las fallas cardíacas en pacientes, aunado a esto, normalmente cuando los niveles superaban estos rangos era cuando normalmente se presentaba los casos de muerte
3. Se puede reconocer como una variable importante ya que la mayoría de los pacientes poseen niveles de 135-140,además, normalmente cuando los niveles superaban estos rangos era cuando normalmente se presentaba los casos de muerte
4. No, tener fallas cardíacas no implica una muerte inminente, ya que de acuerdo a los datos analizados, la minoría de los pacientes fallecieron
5. No, 85 años o más no es un factor que determine la presencia de fallas en el corazón, puesto que nuestros datos demuestran mayor participación en personas de 50-70 años que de 85 años en adelante