Estadística

Gustavo Ortíz
Ana Camila Jiménez
Jorge Sánchez

Estadística descriptiva

```
[ ]  # importar tabla
     df = pd.read_csv('datos_2021.csv',na_values=' ')
```

1. Carga los datos usando tu lector de csv o con pandas.

Importamos el archivo csv usando pandas y generamos la tabla de datos

```
[ ]  df
```

	Estación SIMAJ	Fecha	Hora	O3	NO	NO2	NOX	SO2	CO	PM10	...	TMPI	TMP	RH	WS	WD	PP	RS	PBA	UV	UVI
0	Aguilas	01/01/21	0	0.004	NaN	NaN	NaN	0.0016	1.528	NaN	...	23.0	12.2	62.8	1.12	168.51	0.0	NaN	NaN	NaN	NaN
1	Aguilas	01/01/21	1	0.003	NaN	NaN	NaN	0.0000	0.000	NaN	...	23.0	12.4	61.3	2.62	65.69	0.0	NaN	NaN	NaN	NaN
2	Aguilas	01/01/21	2	0.000	NaN	NaN	NaN	0.0029	1.683	NaN	...	23.0	11.9	63.3	1.02	174.88	0.0	NaN	NaN	NaN	NaN
3	Aguilas	01/01/21	3	0.000	NaN	NaN	NaN	0.0021	1.387	NaN	...	23.0	11.4	66.2	1.05	314.18	0.0	NaN	NaN	NaN	NaN
4	Aguilas	01/01/21	4	0.002	NaN	NaN	NaN	0.0025	1.207	NaN	...	23.0	10.9	68.2	1.46	274.03	0.0	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8755	Aguilas	31/12/21	19	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	22.0	NaN	NaN	NaN	NaN	0.0	NaN	NaN	NaN	NaN
8756	Aguilas	31/12/21	20	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	22.2	NaN	NaN	NaN	NaN	0.0	NaN	NaN	NaN	NaN
8757	Aguilas	31/12/21	21	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	22.2	NaN	NaN	NaN	NaN	0.0	NaN	NaN	NaN	NaN
8758	Aguilas	31/12/21	22	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	22.2	NaN	NaN	NaN	NaN	0.0	NaN	NaN	NaN	NaN
8759	Aguilas	31/12/21	23	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	22.2	NaN	NaN	NaN	NaN	0.0	NaN	NaN	NaN	NaN

8760 rows × 21 columns

2. Verifica la cantidad de datos que tienes, las variables que contiene cada vector de datos e identifica el tipo de variables.

```
print(df.shape)
print(df.columns)
obj_columns = df.select_dtypes(include=np.object).columns.tolist()
df[obj_columns] = df[obj_columns].astype('string')
print(df.info())
```

```
(8760, 21)
Index(['Estación SIMAJ', 'Fecha', 'Hora', 'O3', 'NO', 'NO2', 'NOX', 'SO2',
       'CO', 'PM10', 'PM2.5', 'TMPI', 'TMP', 'RH', 'WS', 'WD', 'PP', 'RS',
       'PBA', 'UV', 'UVI'],
      dtype='object')
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8760 entries, 0 to 8759
Data columns (total 21 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Estación SIMAJ  8760 non-null   string
 1   Fecha           8760 non-null   string
 2   Hora            8760 non-null   int64
 3   O3              5568 non-null   float64
 4   NO              3216 non-null   float64
 5   NO2             3216 non-null   float64
 6   NOX             3216 non-null   float64
 7   SO2             2232 non-null   float64
 8   CO              4296 non-null   float64
 9   PM10            2088 non-null   float64
 10  PM2.5           561 non-null    float64
 11  TMPI            7794 non-null   float64
 12  TMP             7416 non-null   float64
 13  RH              7440 non-null   float64
 14  WS              7440 non-null   float64
 15  WD              7416 non-null   string
 16  PP              7896 non-null   float64
 17  RS              0 non-null      float64
 18  PBA             0 non-null      float64
 19  UV              0 non-null      float64
 20  UVI             0 non-null      float64
dtypes: float64(17), int64(1), string(3)
memory usage: 1.4 MB
None
```

```
[ ]  print(df.describe())
```

```
              Hora           O3           NO          NO2          NOX  \
count  8760.000000  5568.000000  3216.000000  3216.000000  3216.00000
mean     11.500000     0.028869     0.013967     0.021505     0.03547
std       6.922582     0.021738     0.023529     0.016476     0.03568
min       0.000000     0.000000     0.000000     0.000000     0.00000
25%       5.750000     0.011000     0.002000     0.011000     0.01400
50%      11.500000     0.026000     0.006000     0.019000     0.02600
75%      17.250000     0.042000     0.015000     0.030000     0.04600
max      23.000000     0.139000     0.268000     0.135000     0.34900

              SO2           CO         PM10        PM2.5         TMPI  \
count  2232.000000  4296.000000  2088.000000   561.000000  7794.000000
mean      0.001104     0.685496    45.375383    46.036542    23.199666
std       0.000984     0.432117    35.664791    21.425526     4.278118
min       0.000000     0.000000     0.000000    11.500000     0.000000
25%       0.000500     0.432000    22.500000    30.200000    21.600000
50%       0.000800     0.585000    38.400000    42.500000    22.200000
75%       0.001400     0.830250    59.525000    56.500000    22.800000
max       0.010100     3.896000   454.600000   180.900000    44.900000

              TMP           RH           WS           PP   RS  PBA   UV  UVI
count  7416.000000  7440.000000  7440.000000  7896.000000  0.0  0.0  0.0  0.0
mean     23.795577    47.760914     3.817481     0.010992  NaN  NaN  NaN  NaN
std       5.395954    23.517940     2.311717     0.126035  NaN  NaN  NaN  NaN
min       0.000000     0.000000     0.000000     0.000000  NaN  NaN  NaN  NaN
25%      20.300000    29.600000     2.090000     0.000000  NaN  NaN  NaN  NaN
50%      23.200000    45.100000     3.370000     0.000000  NaN  NaN  NaN  NaN
75%      27.725000    67.000000     5.110000     0.000000  NaN  NaN  NaN  NaN
max      37.600000    95.200000    13.580000     2.845000  NaN  NaN  NaN  NaN
```

```
[ ]  #Cantidad de informacion ausente
     print(df.isna().sum()/len(df)*100)
```

```
Estación SIMAJ      0.000000
Fecha               0.000000
Hora                0.000000
O3                 36.438356
NO                 63.287671
NO2                63.287671
NOX                63.287671
SO2                74.520548
CO                 50.958904
PM10               76.164384
PM2.5              93.595890
TMPI               11.027397
TMP                15.342466
RH                 15.068493
WS                 15.068493
WD                 15.342466
PP                  9.863014
RS                100.000000
PBA               100.000000
UV                100.000000
UVI               100.000000
dtype: float64
```

```
[ ]  df_nuevo = df.loc[:,['Hora','O3','CO','TMP']]
     df_nuevo
```

	Hora	O3	CO	TMP
0	0	0.004	1.528	12.2
1	1	0.003	0.000	12.4
2	2	0.000	1.683	11.9
3	3	0.000	1.387	11.4
4	4	0.002	1.207	10.9
...	...	...	...	...
8755	19	NaN	NaN	NaN
8756	20	NaN	NaN	NaN
8757	21	NaN	NaN	NaN
8758	22	NaN	NaN	NaN
8759	23	NaN	NaN	NaN

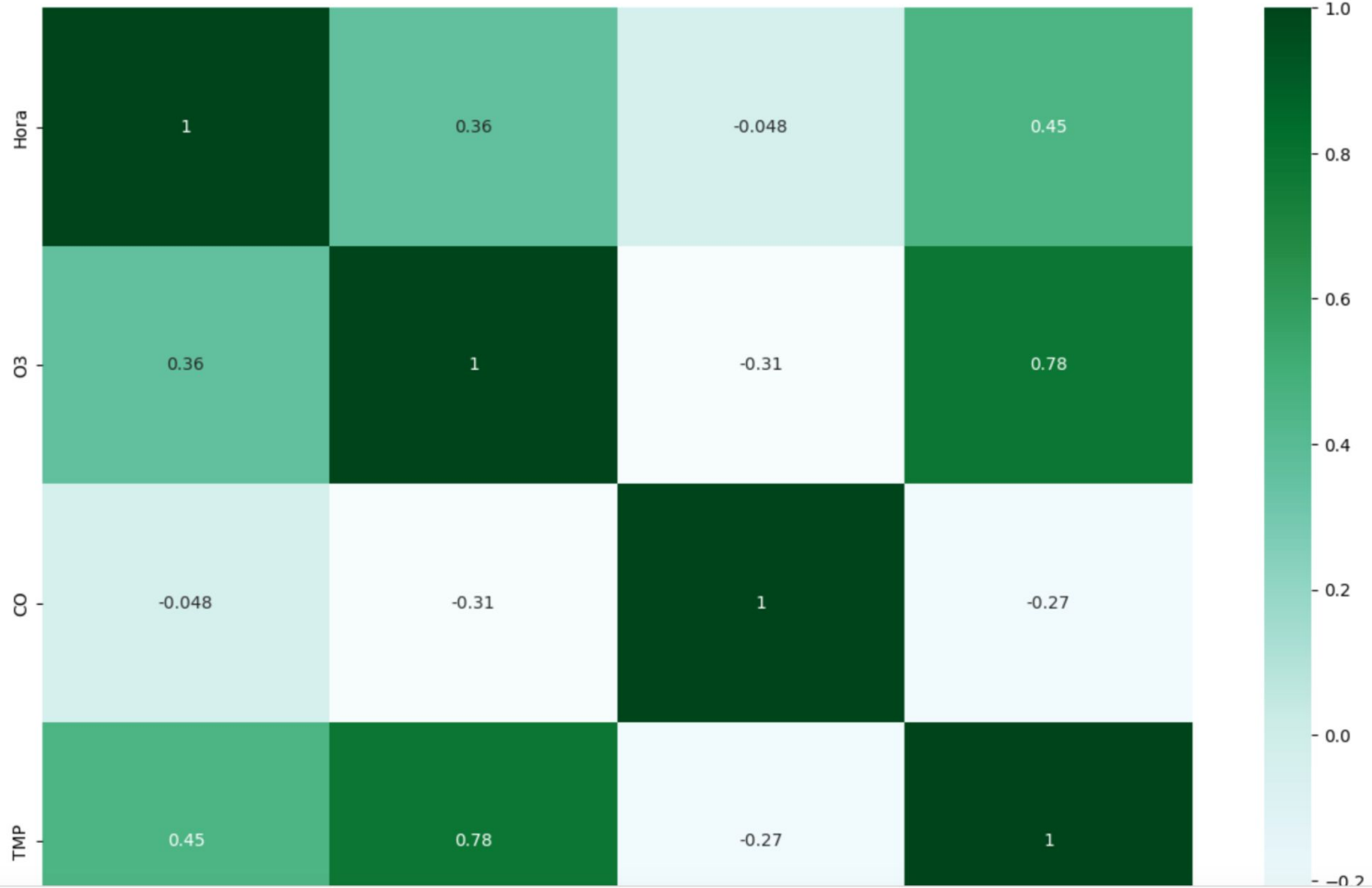
8760 rows × 4 columns

Boxplots y mapas de calor

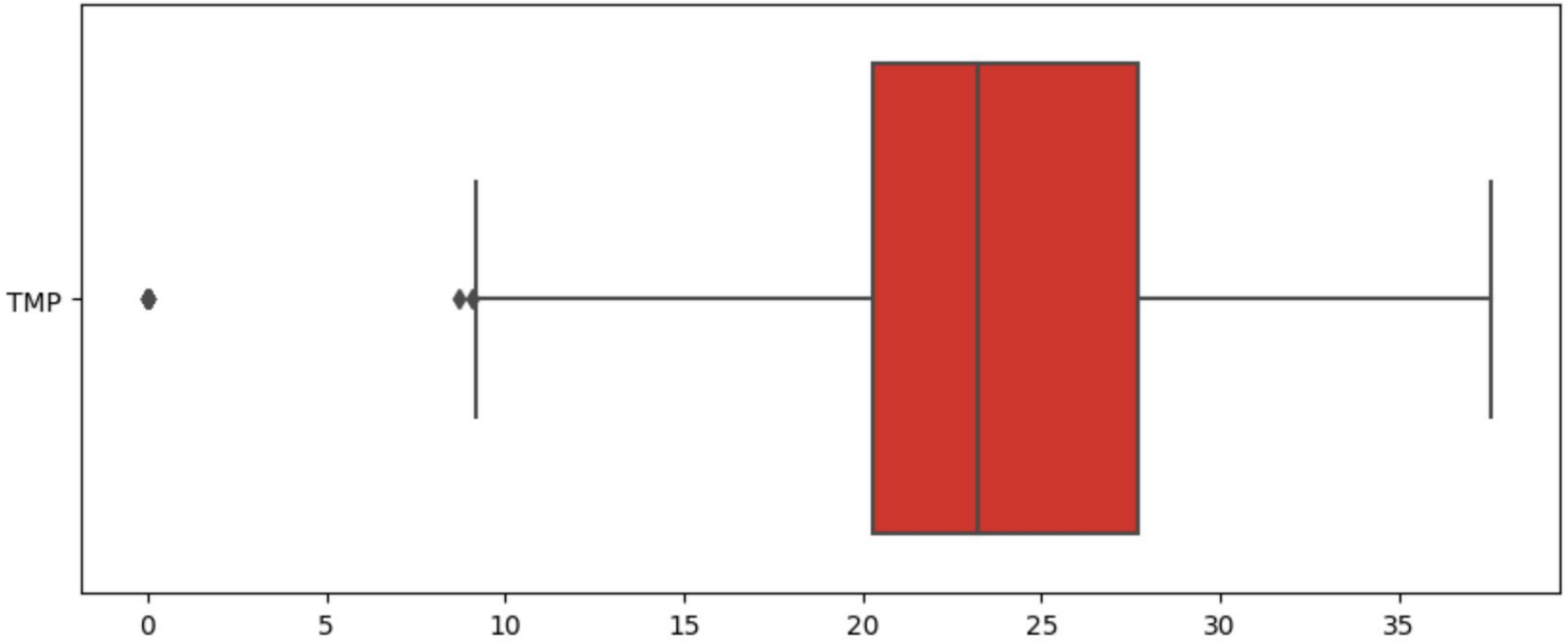
```
[ ]  df_nuevo.corr()
```

	Hora	O3	CO	TMP
Hora	1.000000	0.362315	-0.047729	0.453854
O3	0.362315	1.000000	-0.309503	0.779743
CO	-0.047729	-0.309503	1.000000	-0.267204
TMP	0.453854	0.779743	-0.267204	1.000000

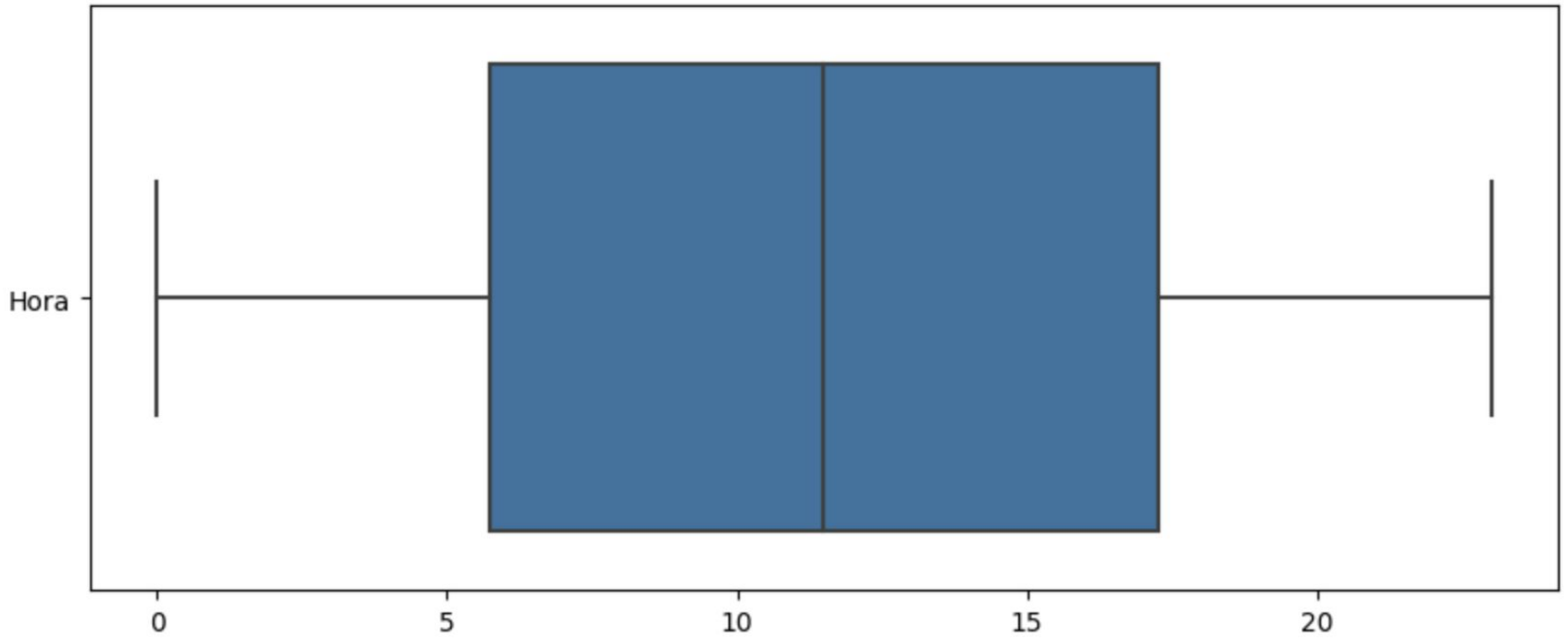
```
[ ]  fig = plt.figure(figsize=(15,10))
     sns.heatmap(df_nuevo.corr(),annot=True, cmap = 'BuGn')
     plt.show()
```



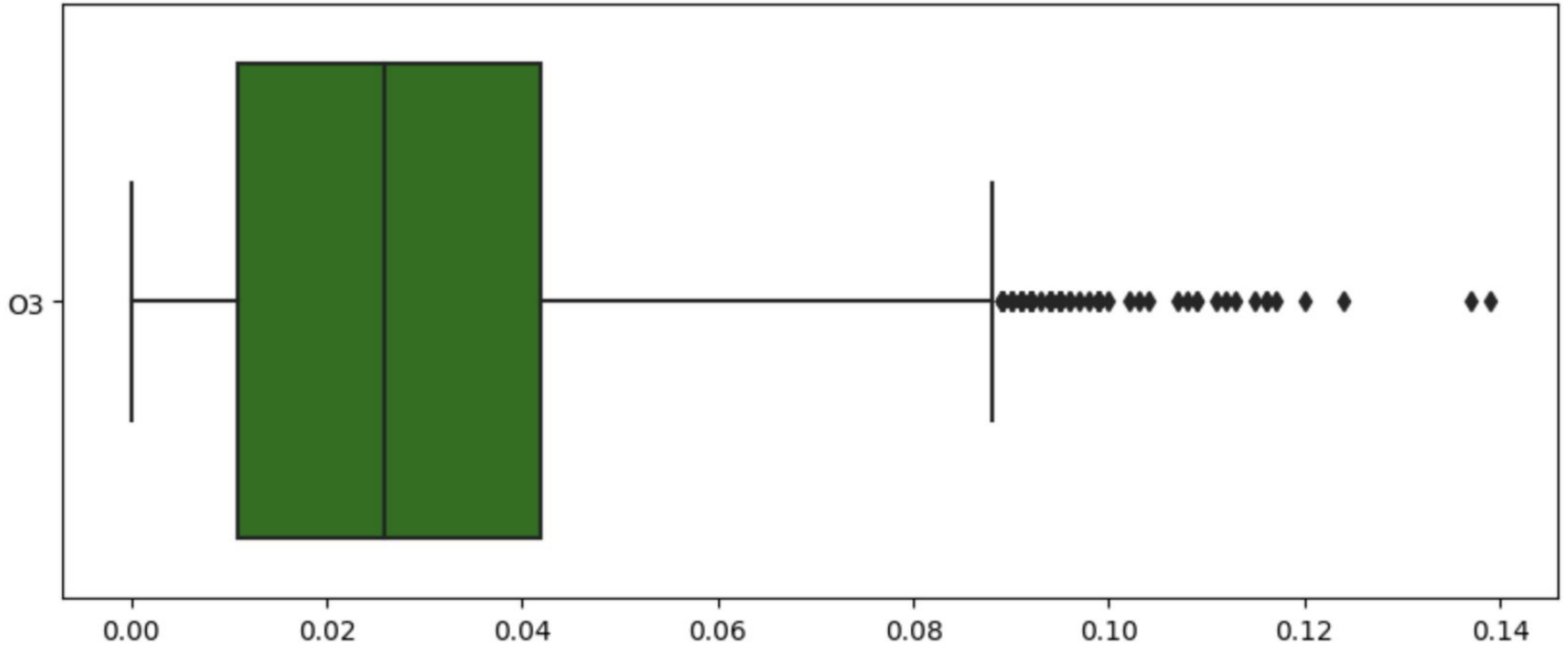
```
[ ]  fig = plt.figure(figsize=(10,4))
     sns.boxplot(data=df_nuevo[['TMP']], orient="h",color='r')
     plt.show()
```



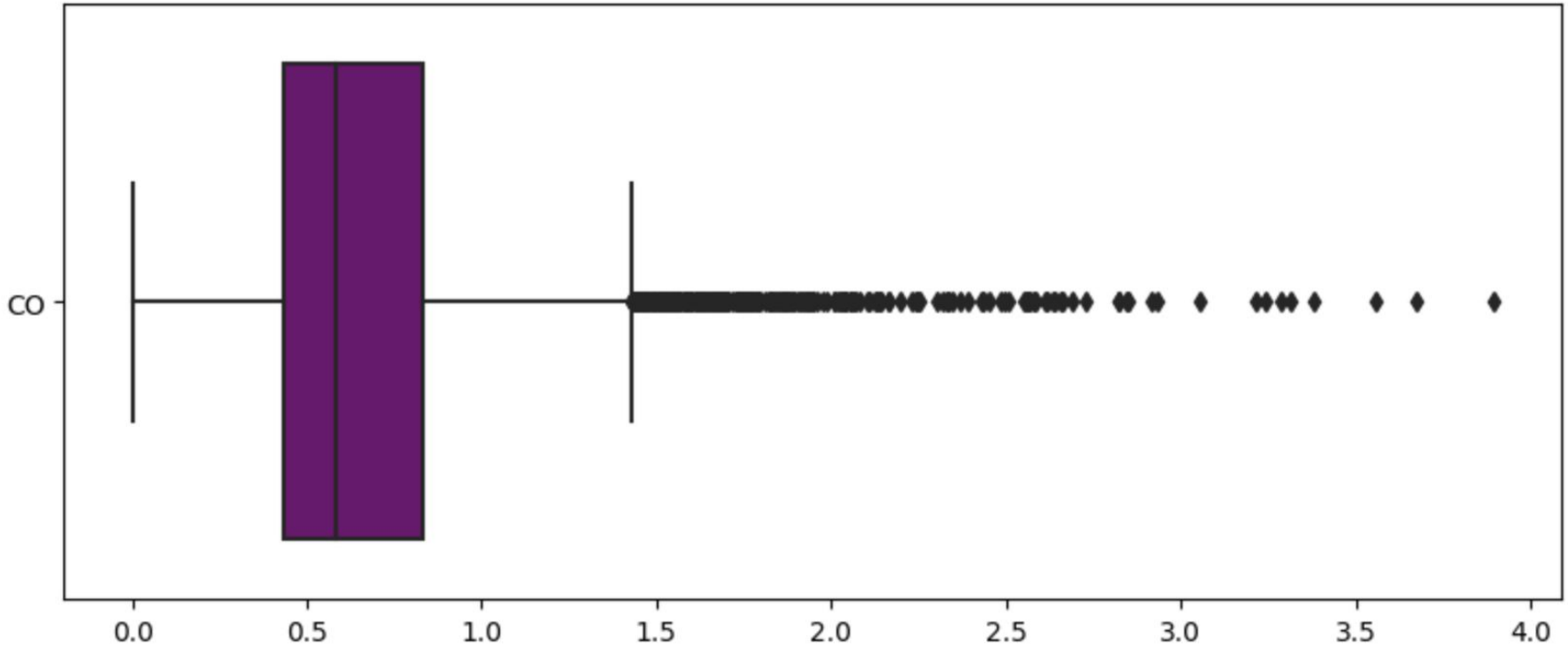
```
[ ]  fig = plt.figure(figsize=(10,4))
     sns.boxplot(data=df_nuevo[['Hora']], orient="h")
     plt.show()
```



```
[ ]  fig = plt.figure(figsize=(10,4))
     sns.boxplot(data=df_nuevo[['O3']], orient="h",color='g')
     plt.show()
```



```
[ ]  fig = plt.figure(figsize=(10,4))
     sns.boxplot(data=df_nuevo[['CO']], orient="h",color='purple')
     plt.show()
```



Conclusiones

Muchas gracias