

A la hora de desarrollar un modelo predictivo con machine learning es de vital importancia visualizar nuestros datos para conocer su comportamiento y distribución. Esta primera observación de datos posibilita aprender más sobre ellos siendo la forma más rápida y útil de conocer qué técnicas son las más adecuadas en pre y pos procesamiento.

FASE: Análisis de Datos. Visualización.

David Víctor Gómez Ramírez

*Técnico Superior en Desarrollo de Aplicaciones
Multiplataforma especialidad en BIGDATA*

PREÁMBULO	2
IMPORTAR LIBRERÍAS NECESARIAS	2
CARGAR EL CONJUNTO DE DATOS	2
VISUALIZACIÓN UNIVARIABLE	3
HISTOGRAMAS	3
DENSIDAD	4
BOXPLOTS	5
ESTUDIO VISUALIZACIÓN "POR_CP"	7
ESTUDIO VISUALIZACIÓN "DISTANCIA"	8
ESTUDIO VISUALIZACIÓN "RITMO"	9
ESTUDIO VISUALIZACIÓN "FREC_CARDIACA"	10
ESTUDIO VISUALIZACIÓN "CADENCIA"	11
ESTUDIO VISUALIZACIÓN "TSC"	12
ESTUDIO VISUALIZACIÓN "FP"	13
ESTUDIO VISUALIZACIÓN "LSS"	14
ESTUDIO VISUALIZACIÓN "OSC_VERTICAL"	15
ESTUDIO VISUALIZACIÓN "L_ZANCADA"	16
ESTUDIO VISUALIZACIÓN "RFP"	17
ESTUDIO VISUALIZACIÓN "RLSS"	18
ESTUDIO VISUALIZACIÓN "ROV"	19
ESTUDIO VISUALIZACIÓN "RE"	20
ESTUDIO VISUALIZACIÓN "AIRE"	21
ESTUDIO VISUALIZACIÓN "PENDIENTE"	22
CORRELACIÓN ENTRE LAS CARACTERÍSTICAS	ERROR! BOOKMARK NOT DEFINED.
<i>Coeficiente de Pearson</i>	<i>Error! Bookmark not defined.</i>
<i>Coeficiente de Spearman</i>	<i>Error! Bookmark not defined.</i>
SESGO (SKEW)	ERROR! BOOKMARK NOT DEFINED.
GRÁFICAS DE DISTRIBUCIÓN Y DENSIDAD	ERROR! BOOKMARK NOT DEFINED.
<i>Cox-Box-Densidad</i>	<i>Error! Bookmark not defined.</i>

PREÁMBULO

Lo primero que debemos realizar a la hora de trabajar con machine learning es visualizar nuestros datos para conocer su comportamiento y distribución. Esta primera observación de datos posibilita aprender más sobre ellos siendo la forma más rápida y útil de conocer qué técnicas son las más adecuadas en pre y pos procesamiento. En este sentido en esta tercera sección trabajaremos:

- Cómo crear gráficos para entender cada atributo de manera independiente.
- Cómo crear gráficos para entender las relaciones entre los diferentes atributos.

Los gráficos de las relaciones entre los atributos pueden darnos una idea de los atributos que pueden ser redundantes, los métodos de remuestreo que pueden ser necesarios y, en última instancia, la dificultad de un problema de predicción. Para ello, la fase de visualización puede dividirse en las siguientes partes:

- **Visualización univariable:** cuando queremos visualizar un atributo de manera independiente a los demás.
- **Visualización multivariable:** cuando queremos visualizar la interacción entre los diferentes atributos de nuestro conjunto de datos.

Importar librerías necesarias

```
# *****
# ***** LIBRERÍAS A IMPORTAR *****
# *****
import pandas as pd
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
```

Cargar el conjunto de datos

Para esta práctica vamos a cargar el conjunto de datos de nuestro proyecto "SEGMENTOS_csv.csv" para hacer observaciones con las funciones que nos permitan hacer diferentes tipos de visualizaciones. Además y, conocedores que en "DESNIVEL" hay un registro vacío, procederemos a su resolución para un mejor tratamiento.

```
# *****
# ***** CARGAMOS NUESTRO DATAFRAME *****
# *****
# Cargamos los datos contenidos en "SEGMENTOS_csv.csv"
data = pd.read_csv('SEGMENTOS_csv.csv')
# Completamos los datos nulos con la media de cada uno
data['DESNIVEL'] = data['DESNIVEL'].fillna(data['DESNIVEL'].median())
```

	POR_CP	DISTANCIA	RETRO	FREC_CARDIACA	CADENCIA	TSC	FP	LSS	%
0	85,863	500	519	156	146	160	66	12.0	
1	84,816	1000	547	138	159	342	66	11.8	
2	84,293	1000	537	137	161	346	66	12.1	
3	84,816	1000	540	142	162	347	66	12.2	
4	84,816	1000	539	146	163	349	66	12.1	
5	84,816	1000	542	150	163	346	66	12.2	
6	84,816	1000	537	152	165	339	66	12.4	
7	85,863	600	528	155	165	341	66	12.1	
8	85,340	550	501	120	158	340	65	11.7	
9	84,816	5570	538	145	163	345	66	11.2	
10	84,816	1000	545	134	160	345	66	11.9	
11	84,816	2000	548	144	162	346	66	12.2	
12	84,816	2000	539	151	164	342	66	12.1	
13	84,816	5000	541	136	161	348	66	12.0	
14	84,816	3000	530	146	164	343	66	12.3	

	OSC_VERTICAL	L_TARONJA	RFP	RLSS	POV	RE	AIRE	PERCENTE	%
0	5.80	0.077	39.024	140.342	7.128	0.985	1	-0.5	
1	5480.00	0.089	48.740	143.992	8.858	0.925	0	0.4	
2	5470.00	0.093	48.500	147.560	7.927	0.948	0	0.0	
3	5420.00	0.085	48.740	148.780	7.970	0.937	0	0.1	
4	5290.00	0.082	48.123	147.590	7.770	0.939	0	0.0	
5	5280.00	0.079	48.123	150.000	7.691	0.933	0	0.1	
6	5180.00	0.077	48.123	151.219	7.617	0.942	1	-0.2	
7	5070.00	0.088	39.624	147.160	7.347	0.948	1	-0.2	
8	5410.00	0.070	38.877	147.082	8.074	0.888	1	0.0	
9	5380.00	0.083	48.740	148.780	7.862	0.939	0	0.1	
10	5490.00	0.093	48.740	145.121	7.942	0.935	0	0.2	
11	5360.00	0.081	48.740	148.780	7.882	0.937	0	0.1	
12	5210.00	0.078	48.123	150.000	7.661	0.938	1	0.0	
13	5460.00	0.089	48.740	146.342	7.813	0.936	0	0.2	
14	5200.00	0.077	48.123	150.000	7.504	0.937	0	0.0	

	ALTITUD	DESNIVEL	SEX	PORCENTE	%
0	15	3.8	2	355	
1	18	7.0	38	344	
2	18	4.0	5	951	
3	19	6.0	6	940	
4	20	5.0	0	525	
5	22	7.0	0	542	
6	22	2.0	0	527	
7	27	5.0	6	517	
8	15	7.0	7	102	
9	18	37.0	71	1030	
10	27	11.0	10	1083	
11	29	11.0	18	1089	
12	29	9.0	18	1074	
13	27	12.0	20	1420	
14	28	13.0	20	1227	

Visualización Univariable

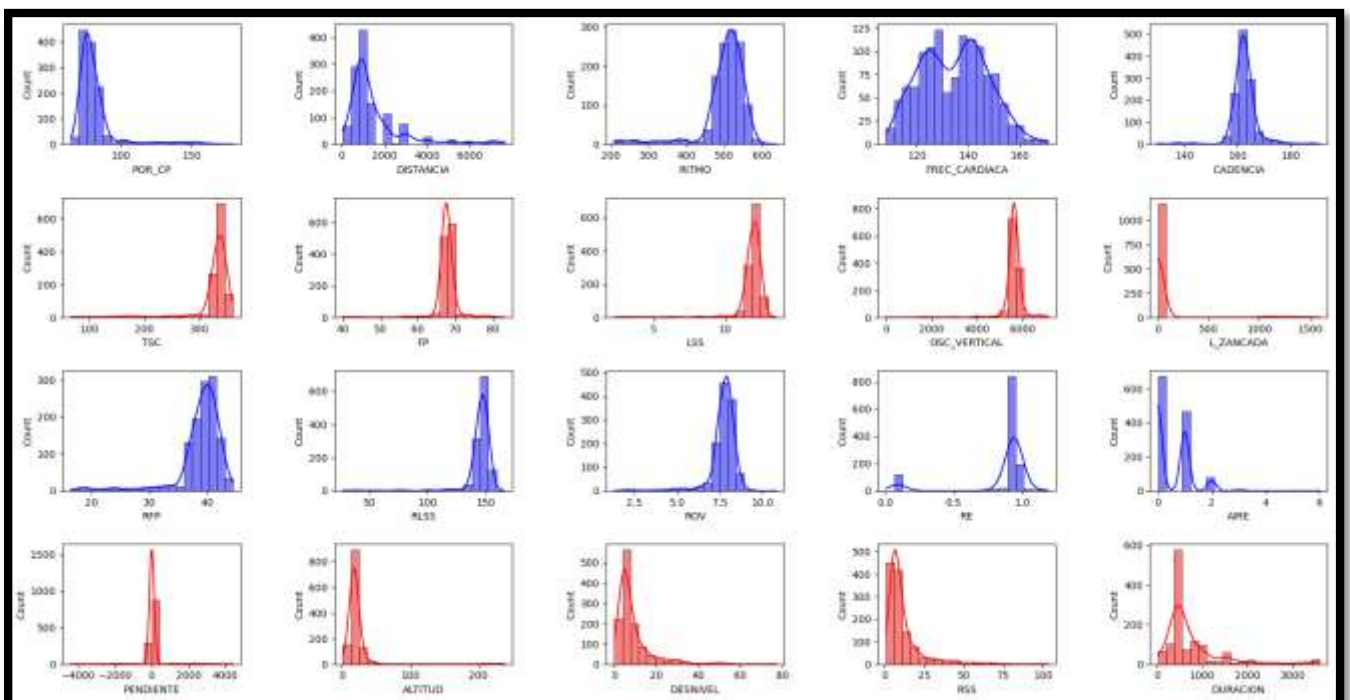
Como se ha comentado anteriormente, las gráficas univariable nos permiten visualizar los atributos individuales sin interacciones; las cuales, el objetivo principal de las mismas es aprender algo sobre la distribución, la tendencia y la propagación de cada atributo.

A continuación se describen las más relevantes.

Histogramas

A partir de la forma de los contenedores, puede tener una idea rápida de si un atributo es gaussiano, sesgado o incluso tiene una distribución exponencial. También puede ayudarlo a ver posibles valores atípicos, por lo que tanto Matplotlib como Seaborn pueden ser potentes librerías de visualización de datos.

```
# *****
# ***** VISUALIZACIONES *****
# *****
# HISTOGRAMAS O DISTRIBUCIÓN CON DENSIDAD
f, axes = plt.subplots(4,5, figsize=(11.7,8.27))
sns.histplot(data["POR_CP"], ax=axes[0,0], kde=True, bins=20, color="Blue", fill=True)
sns.histplot(data["DISTANCIA"], ax=axes[0,1], kde=True, bins=20, color="Blue", fill=True)
sns.histplot(data["RITMO"], ax=axes[0,2], kde=True, bins=20, color="Blue", fill=True)
sns.histplot(data["FREC_CARDIACA"], ax=axes[0,3], kde=True, bins=20, color="Blue", fill=True)
sns.histplot(data["CADENCIA"], ax=axes[0,4], kde=True, bins=20, color="Blue", fill=True)
sns.histplot(data["FP"], ax=axes[1,0], kde=True, bins=20, color="Red", fill=True)
sns.histplot(data["LSS"], ax=axes[1,1], kde=True, bins=20, color="Red", fill=True)
sns.histplot(data["OSC_VERTICAL"], ax=axes[1,2], kde=True, bins=20, color="Red", fill=True)
sns.histplot(data["L_ZANCADA"], ax=axes[1,3], kde=True, bins=20, color="Red", fill=True)
sns.histplot(data["RFP"], ax=axes[2,0], kde=True, bins=20, color="Blue", fill=True)
sns.histplot(data["RLSS"], ax=axes[2,1], kde=True, bins=20, color="Blue", fill=True)
sns.histplot(data["ROV"], ax=axes[2,2], kde=True, bins=20, color="Blue", fill=True)
sns.histplot(data["RE"], ax=axes[2,3], kde=True, bins=20, color="Blue", fill=True)
sns.histplot(data["AIRE"], ax=axes[2,4], kde=True, bins=20, color="Blue", fill=True)
sns.histplot(data["PENDIENTE"], ax=axes[3,0], kde=True, bins=20, color="Red", fill=True)
sns.histplot(data["ALTITUD"], ax=axes[3,1], kde=True, bins=20, color="Red", fill=True)
sns.histplot(data["DESNIVEL"], ax=axes[3,2], kde=True, bins=20, color="Red", fill=True)
sns.histplot(data["RSS"], ax=axes[3,3], kde=True, bins=20, color="Red", fill=True)
sns.histplot(data["DURACION"], ax=axes[3,4], kde=True, bins=20, color="Red", fill=True)
plt.tight_layout()
plt.show()
```



Siempre, a priori, y visualmente hablando, podemos observar que la mayoría de los atributos suelen tener una distribución casi gaussiana o normal (algunas simples, otras con doble campana FREC_CARDIACA y REC y otra triple campana AIRE)y aparentemente alguna exponencial L_ZANCADA. También podemos observar la existencia de sesgo en casi todas las distribuciones siendo este menor o casi inexistente en FREC_CARDIACA y CADENCIA. Esto es interesante porque muchas técnicas de aprendizaje automático suponen una distribución univariada gaussiana en las variables de entrada.

Densidad

Las gráficas se ven como un histograma abstracto con una curva suave dibujada a través de la parte superior de cada contenedor, al igual que su ojo intentó hacer con los histogramas. Podemos ver que la distribución de cada atributo es más clara que los histogramas

```
# GRAFICOS DE DENSIDAD
f, axes = plt.subplots(4,5, figsize=(11.7,8.27))
sns.kdeplot(data["POR_CP"], ax=axes[0,0], shade=True, color="Blue", fill=True,
bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.kdeplot(data["DISTANCIA"], ax=axes[0,1], shade=True, color="Blue", fill=True,
bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.kdeplot(data["RITMO"], ax=axes[0,2], shade=True, color="Blue", fill=True,
bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.kdeplot(data["FREC_CARDIACA"], ax=axes[0,3], shade=True, color="Blue", fill=True,
bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.kdeplot(data["CADENCIA"], ax=axes[0,4], shade=True, color="Blue", fill=True,
bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.kdeplot(data["TSC"], ax=axes[1,0], shade=True, color="Red", fill=True,
bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.kdeplot(data["FP"], ax=axes[1,1], shade=True, color="Red", fill=True,
bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.kdeplot(data["LSS"], ax=axes[1,2], shade=True, color="Red", fill=True,
bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.kdeplot(data["OSC_VERTICAL"], ax=axes[1,3], shade=True, color="Red", fill=True,
bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.kdeplot(data["L_ZANCADA"], ax=axes[1,4], shade=True, color="Red", fill=True,
bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.kdeplot(data["RFP"], ax=axes[2,0], shade=True, color="Red", fill=True,
bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.kdeplot(data["RLSS"], ax=axes[2,1], shade=True, color="Blue", fill=True,
bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.kdeplot(data["ROV"], ax=axes[2,2], shade=True, color="Blue", fill=True,
bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.kdeplot(data["RE"], ax=axes[2,3], shade=True, color="Blue", fill=True,
bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.kdeplot(data["AIRE"], ax=axes[2,4], shade=True, color="Blue", fill=True,
bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.kdeplot(data["PENDIENTE"], ax=axes[3,0], shade=True, color="Red", fill=True,
bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

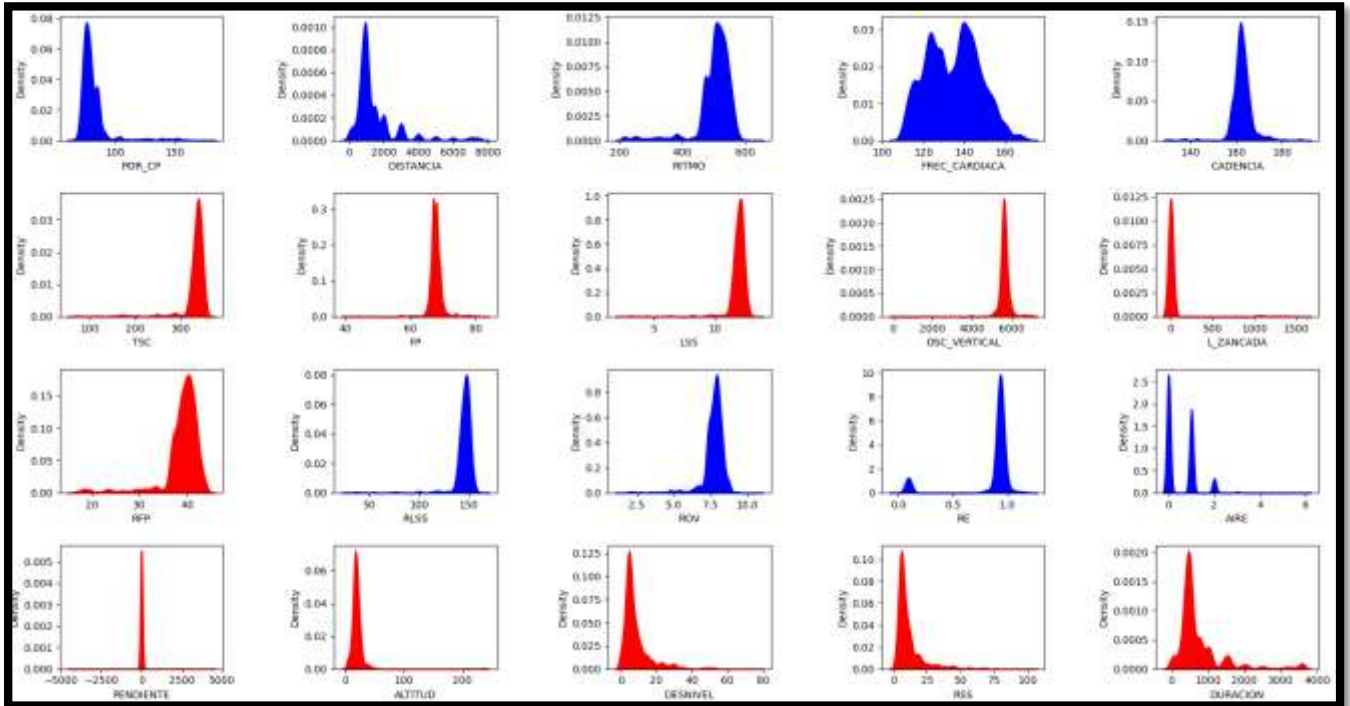
sns.kdeplot(data["ALTITUD"], ax=axes[3,1], shade=True, color="Red", fill=True,
bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.kdeplot(data["DESNIVEL"], ax=axes[3,2], shade=True, color="Red", fill=True,
bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.kdeplot(data["RSS"], ax=axes[3,3], shade=True, color="Red", fill=True,
bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)
```

```
sns.kdeplot(data["DURACION"], ax = axes [3,4], shade = True, color = "Red", fill = True,
            bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

plt.tight_layout()
plt.show()
```



Boxplots

Podemos ver que la extensión de los atributos es bastante diferente. Algunos como la FREC_CARDIACA, L_ZANCADA y RFP parecen bastante sesgados hacia valores más pequeños.

```
# GRAFICOS BOXPLOTS
f, axes = plt.subplots(4,5, figsize = (11.7,8.27))
sns.boxplot(x = data["POR_CP"], ax = axes [0,0], orient = "h", color = "lightblue", saturation= 1,
            width = 0.7, dodge = True, fliersize = 3, linewidth = 2)

sns.boxplot(x = data["DISTANCIA"], ax = axes [0,1], orient = "h", color = "lightblue", saturation= 1,
            width = 0.7, dodge = True, fliersize = 3, linewidth = 2)

sns.boxplot(x = data["RITMO"], ax = axes [0,2], orient = "h", color = "lightblue", saturation= 1,
            width = 0.7, dodge = True, fliersize = 3, linewidth = 2)

sns.boxplot(x = data["FREC_CARDIACA"], ax = axes [0,3], orient = "h", color = "lightblue", saturation= 1,
            width = 0.7, dodge = True, fliersize = 3, linewidth = 2)

sns.boxplot(x = data["CADENCIA"], ax = axes [0,4], orient = "h", color = "lightblue", saturation= 1,
            width = 0.7, dodge = True, fliersize = 3, linewidth = 2)

sns.boxplot(x = data["TSC"], ax = axes [1,0], orient = "h", color = "lightgreen",saturation= 1,
            width = 0.7, dodge = True, fliersize = 3, linewidth = 2)

sns.boxplot(x = data["FP"], ax = axes [1,1], orient = "h", color = "lightgreen",saturation= 1,
            width = 0.7, dodge = True, fliersize = 3, linewidth = 2)

sns.boxplot(x = data["LSS"], ax = axes [1,2], orient = "h", color = "lightgreen",saturation= 1,
            width = 0.7, dodge = True, fliersize = 3, linewidth = 2)

sns.boxplot(x = data["OSC_VERTICAL"], ax = axes [1,3], orient = "h", color = "lightgreen",saturation= 1,
            width = 0.7, dodge = True, fliersize = 3, linewidth = 2)

sns.boxplot(x = data["L_ZANCADA"], ax = axes [1,4], orient = "h", color = "lightgreen",saturation= 1,
            width = 0.7, dodge = True, fliersize = 3, linewidth = 2)

sns.boxplot(x = data["RFP"], ax = axes [2,0], orient = "h", color = "lightblue",saturation= 1,
            width = 0.7, dodge = True, fliersize = 3, linewidth = 2)
```



```

sns.boxplot(x = data["RLSS"], ax = axes [2,1], orient = "h", color = "lightblue", saturation = 1,
width = 0.7, dodge = True, fliersize = 3, linewidth = 2)

sns.boxplot(x = data["ROV"], ax = axes [2,2], orient = "h", color = "lightblue", saturation = 1,
width = 0.7, dodge = True, fliersize = 3, linewidth = 2)

sns.boxplot(x = data["RE"], ax = axes [2,3], orient = "h", color = "lightblue", saturation = 1,
width = 0.7, dodge = True, fliersize = 3, linewidth = 2)

sns.boxplot(x = data["AIRE"], ax = axes [2,4], orient = "h", color = "lightblue", saturation = 1,
width = 0.7, dodge = True, fliersize = 3, linewidth = 2)

sns.boxplot(x = data["PENDIENTE"], ax = axes [3,0], orient = "h", color = "lightgreen", saturation = 1,
width = 0.7, dodge = True, fliersize = 3, linewidth = 2)

sns.boxplot(x = data["ALTITUD"], ax = axes [3,1], orient = "h", color = "lightgreen", saturation = 1,
width = 0.7, dodge = True, fliersize = 3, linewidth = 2)

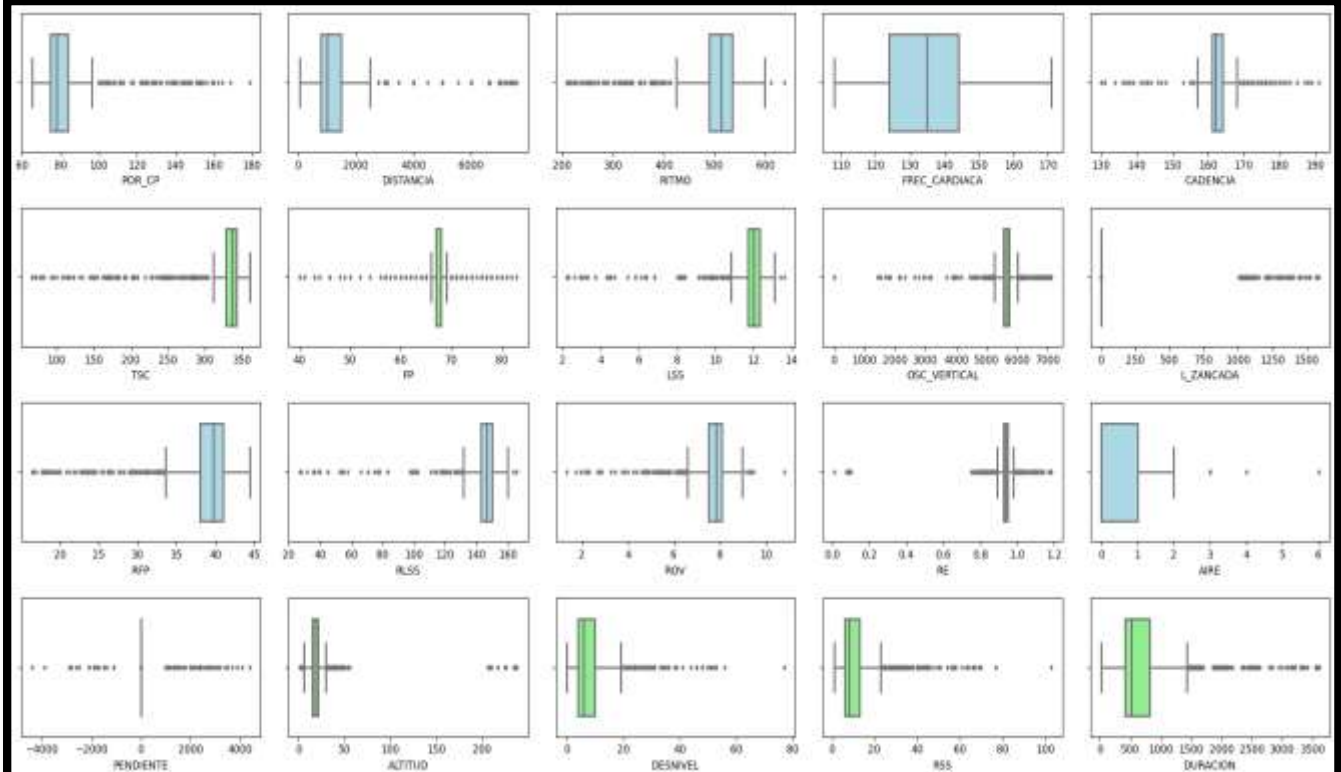
sns.boxplot(x = data["DESNIVEL"], ax = axes [3,2], orient = "h", color = "lightgreen", saturation = 1,
width = 0.7, dodge = True, fliersize = 3, linewidth = 2)

sns.boxplot(x = data["RSS"], ax = axes [3,3], orient = "h", color = "lightgreen", saturation = 1,
width = 0.7, dodge = True, fliersize = 3, linewidth = 2)

sns.boxplot(x = data["DURACION"], ax = axes [3,4], orient = "h", color = "lightgreen", saturation = 1,
width = 0.7, dodge = True, fliersize = 3, linewidth = 2)

plt.tight_layout()
plt.show()

```



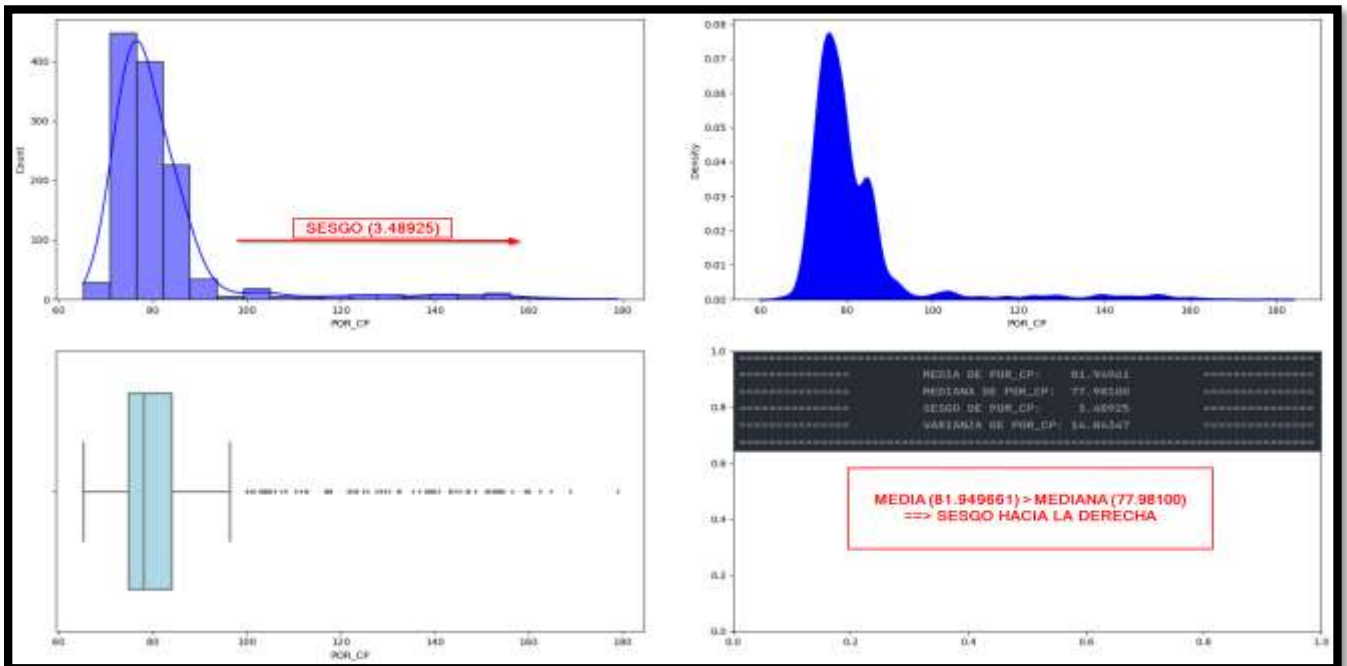
Estudio Visualización "POR_CP"

```
# Estudio Visualización "POR_CP"
# Métricas POR_CP
print("
=====")
print(f"
===== MEDIANA DE POR_CP:
{data['POR_CP'].median():.5f}
=====")
print(f"
===== MEDIA DE POR_CP:
{data['POR_CP'].mean():.5f}
=====")
print(f"
===== SESGO DE POR_CP:
{data['POR_CP'].skew():.5f}
=====")
print("
=====")
f, axes = plt.subplots(2,2, figsize=(11.7,8.27))

sns.histplot(data["POR_CP"], ax = axes [0,0], kde = True, bins = 20, color="Blue",
fill=True)

sns.kdeplot(data["POR_CP"], ax = axes [0,1], shade = True, color = "Blue", fill =
True,
bw_adjust = .5, clip_on = False, alpha=1, linewidth=1.5)

sns.boxplot(x = data["POR_CP"], ax = axes [1,0], orient = "h", color = "lightblue",
saturation = 1,
width = 0.7, dodge = True, fliersize = 3, linewidth = 2)
plt.tight_layout()
plt.show()
```



Estudio Visualización "DISTANCIA"

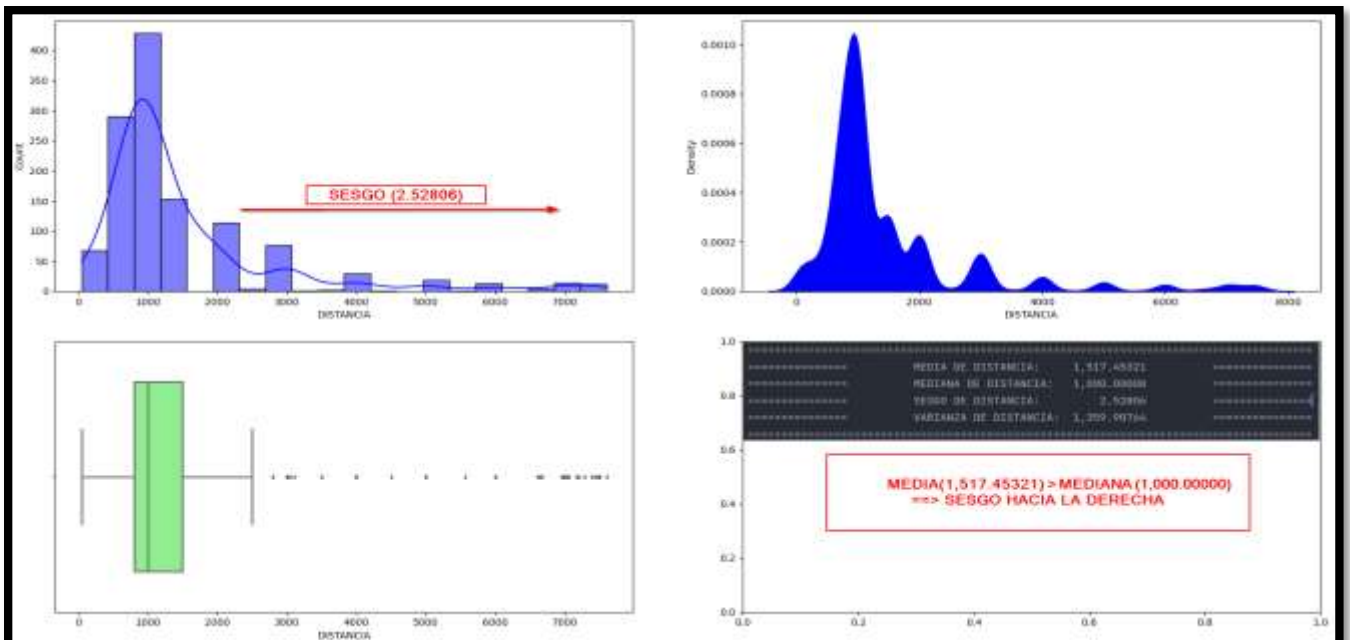
```
# Estudio Visualización "DISTANCIA"
# Métricas DISTANCIA
print("
=====")
print(f"
=====          MEDIA DE DISTANCIA:
{data['DISTANCIA'].mean():.5f}          =====")
print(f"
=====          MEDIANA DE DISTANCIA:
{data['DISTANCIA'].median():.5f}        =====")
print(f"
=====          SESGO DE DISTANCIA:
{data['DISTANCIA'].skew():.5f}          =====")
print(f"
=====          VARIANZA DE DISTANCIA:
{data['DISTANCIA'].std():.5f}          =====")
print("
=====")

# Visualización Histograma, Densidad y Boxplot de DISTANCIA
f, axes = plt.subplots(2,2, figsize =(11.7,8.27))

sns.histplot(data["DISTANCIA"],      ax = axes [0,0], kde = True, bins = 20, color="Blue",
fill=True)

sns.kdeplot(data["DISTANCIA"],      ax = axes [0,1], shade = True, color = "Blue", fill =
True,
          bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.boxplot(x = data["DISTANCIA"],  ax = axes [1,0], orient = "h", color = "lightgreen",
saturation = 1,
          width = 0.7, dodge = True, fliersize = 3, linewidth =
2)
plt.tight_layout()
plt.show()
```



Estudio Visualización "RITMO"

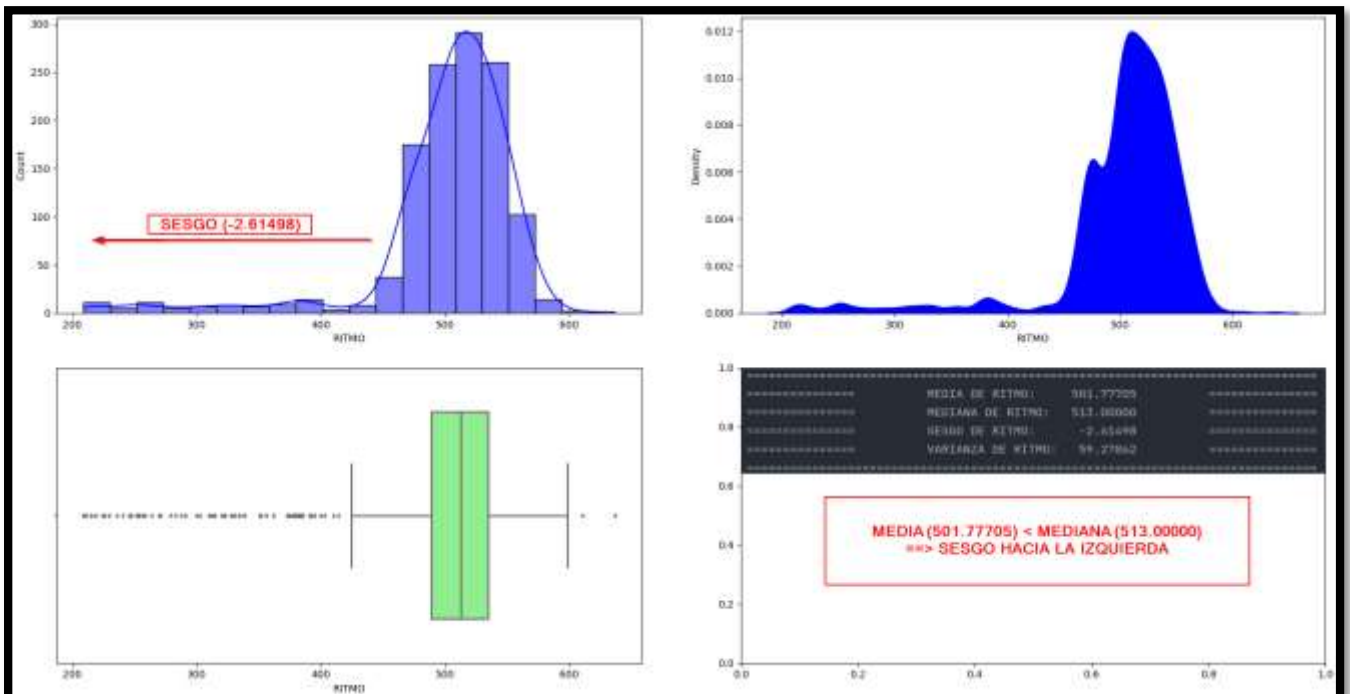
```
# Estudio Visualización "RITMO"
# Métricas RITMO
print("
=====")
print(f"
=====          MEDIA DE RITMO:
{data['RITMO'].mean():,.5f}          =====")
print(f"
=====          MEDIANA DE RITMO:
{data['RITMO'].median():,.5f}          =====")
print(f"
=====          SESGO DE RITMO:
{data['RITMO'].skew():,.5f}          =====")
print(f"
=====          VARIANZA DE RITMO:
{data['RITMO'].std():,.5f}          =====")
print("
=====")

# Visualización Histograma, Densidad y Boxplot de RITMO
f, axes = plt.subplots(2,2, figsize =(11.7,8.27))

sns.histplot(data["RITMO"],      ax = axes [0,0], kde = True, bins = 20, color="Blue",
fill=True)

sns.kdeplot(data["RITMO"],      ax = axes [0,1], shade = True, color = "Blue", fill = True,
bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.boxplot(x = data["RITMO"],  ax = axes [1,0], orient = "h", color = "lightgreen",
saturation = 1,
width = 0.7, dodge = True, fliersize = 3, linewidth =
2)
plt.tight_layout()
plt.show()
```



Estudio Visualización "FREC_CARDIACA"

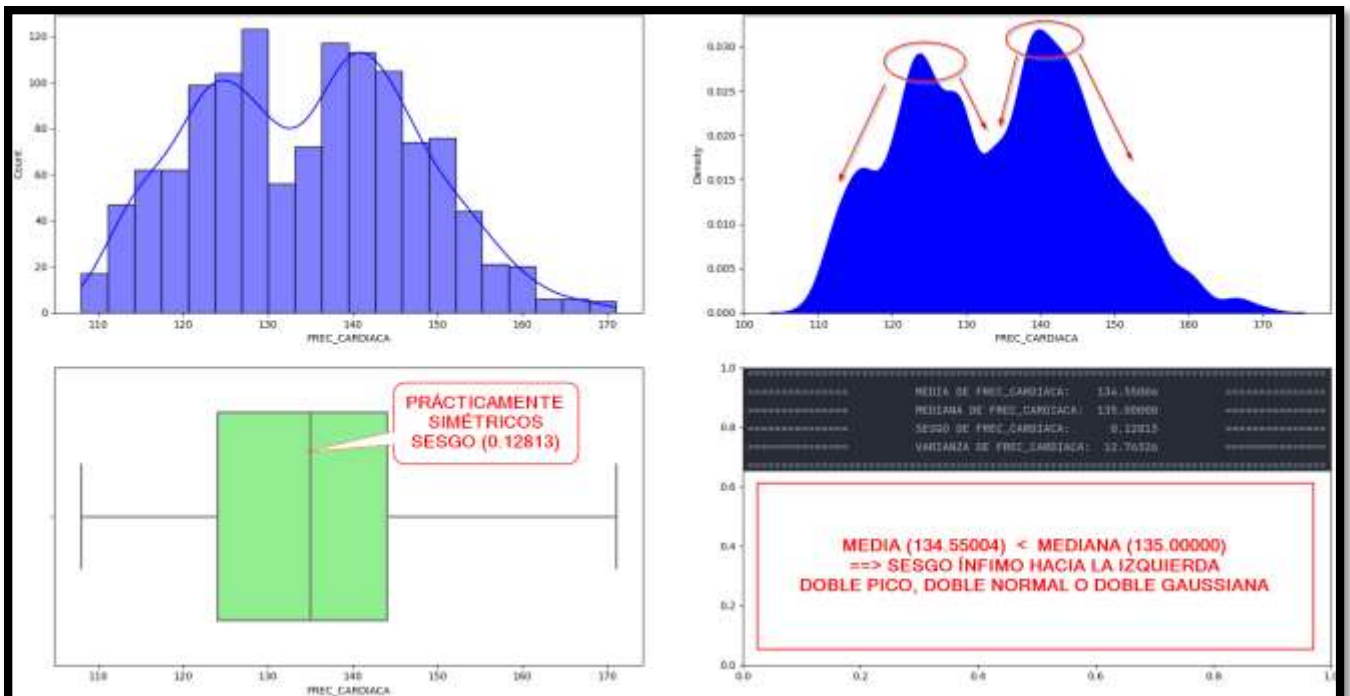
```
# Estudio Visualización "FREC_CARDIACA"
# Métricas FREC_CARDIACA
print("
=====")
print(f"
{data['FREC_CARDIACA'].mean():,.5f}          MEDIA DE FREC_CARDIACA:
=====")
print(f"
{data['FREC_CARDIACA'].median():,.5f}        MEDIANA DE FREC_CARDIACA:
=====")
print(f"
{data['FREC_CARDIACA'].skew():,.5f}          SESGO DE FREC_CARDIACA:
=====")
print(f"
{data['FREC_CARDIACA'].std():,.5f}          VARIANZA DE FREC_CARDIACA:
=====")
print("
=====")

# Visualización Histograma, Densidad y Boxplot de FREC_CARDIACA
f, axes = plt.subplots(2,2, figsize =(11.7,8.27))

sns.histplot(data["FREC_CARDIACA"],          ax = axes [0,0], kde = True, bins = 20, color="Blue",
fill=True)

sns.kdeplot(data["FREC_CARDIACA"],          ax = axes [0,1], shade = True, color = "Blue", fill =
True,
bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.boxplot(x = data["FREC_CARDIACA"],      ax = axes [1,0], orient = "h", color = "lightgreen",
saturation = 1,
width = 0.7, dodge = True, fliersize = 3, linewidth =
2)
plt.tight_layout()
plt.show()
```



Estudio Visualización "CADENCIA"

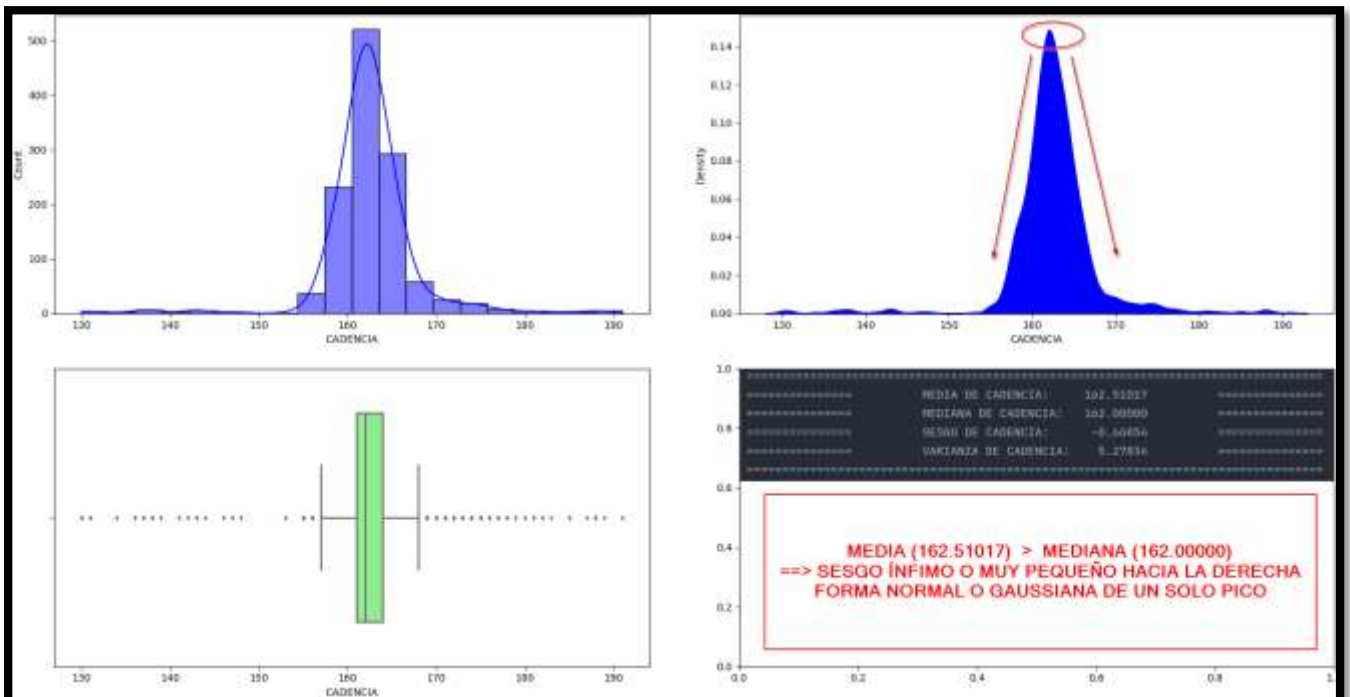
```
# Estudio Visualización "CADENCIA"
# Métricas CADENCIA
print("
=====")
print(f"
=====          MEDIA DE CADENCIA:
{data['CADENCIA'].mean():.5f}          =====")
print(f"
=====          MEDIANA DE CADENCIA:
{data['CADENCIA'].median():.5f}          =====")
print(f"
=====          SESGO DE CADENCIA:
{data['CADENCIA'].skew():.5f}          =====")
print(f"
=====          VARIANZA DE CADENCIA:
{data['CADENCIA'].std():.5f}          =====")
print("
=====")

# Visualización Histograma, Densidad y Boxplot de CADENCIA
f, axes = plt.subplots(2,2, figsize=(11.7,8.27))

sns.histplot(data["CADENCIA"],      ax = axes [0,0], kde = True, bins = 20, color="Blue",
fill=True)

sns.kdeplot(data["CADENCIA"],      ax = axes [0,1], shade = True, color = "Blue", fill = True,
bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.boxplot(x = data["CADENCIA"],  ax = axes [1,0], orient = "h", color = "lightgreen",
saturation = 1,
width = 0.7, dodge = True, fliersize = 3, linewidth =
2)
plt.tight_layout()
plt.show()
```



Estudio Visualización "TSC"

```
# Estudio Visualización "TSC"
# Métricas TSC
print("
=====")
print(f"                =====          MEDIA DE TSC:      {data['TSC'].mean():,.5f}
=====")
print(f"                =====          MEDIANA DE TSC:     {data['TSC'].median():,.5f}
=====")
print(f"                =====          SESGO DE TSC:        {data['TSC'].skew():,.5f}
=====")
print(f"                =====          VARIANZA DE TSC:      {data['TSC'].std():,.5f}
=====")
print("
=====")

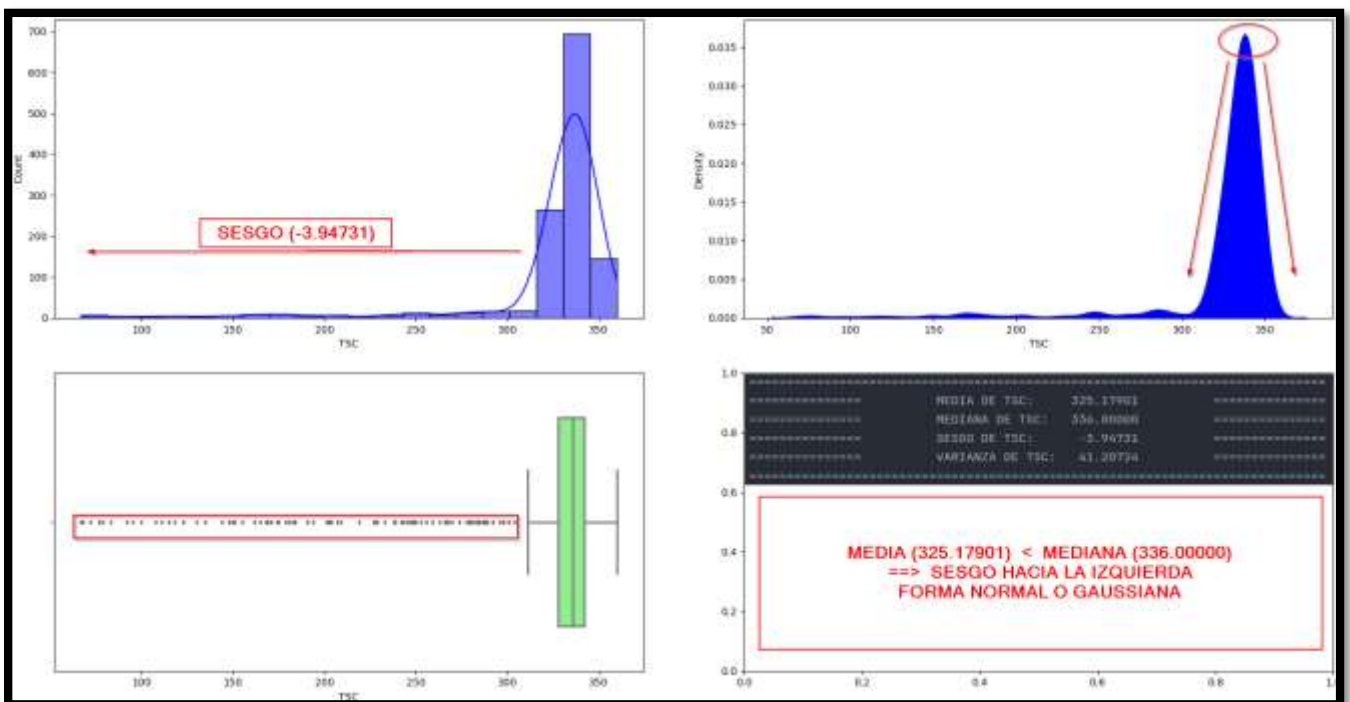
# Visualización Histograma, Densidad y Boxplot de TSC
f, axes = plt.subplots(2,2, figsize =(11.7,8.27))

sns.histplot(data["TSC"],      ax = axes [0,0], kde = True, bins = 20, color="Blue", fill=True)

sns.kdeplot(data["TSC"],      ax = axes [0,1], shade = True, color = "Blue", fill = True,
bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.boxplot(x = data["TSC"],  ax = axes [1,0], orient = "h", color = "lightgreen", saturation
= 1,
                                width = 0.7, dodge = True, fliersize = 3, linewidth = 2)

plt.tight_layout()
plt.show()
```



Estudio Visualización "FP"

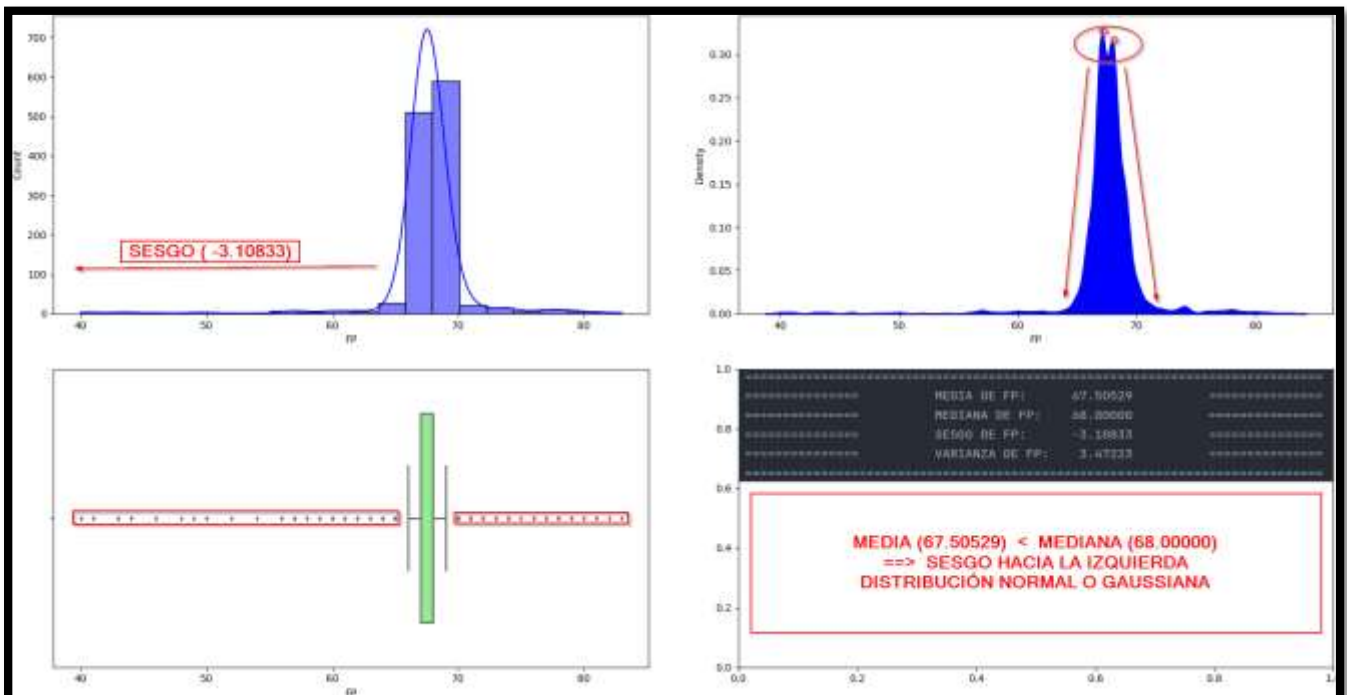
```
# Estudio Visualización "FP"
# Métricas FP
print("
=====")
print(f"                =====          MEDIA DE FP:      {data['FP'].mean():,.5f}
=====")
print(f"                =====          MEDIANA DE FP:     {data['FP'].median():,.5f}
=====")
print(f"                =====          SESGO DE FP:       {data['FP'].skew():,.5f}
=====")
print(f"                =====          VARIANZA DE FP:     {data['FP'].std():,.5f}
=====")
print("
=====")

# Visualización Histograma, Densidad y Boxplot de FP
f, axes = plt.subplots(2,2, figsize=(11.7,8.27))

sns.histplot(data["FP"],      ax = axes [0,0], kde = True, bins = 20, color="Blue", fill=True)

sns.kdeplot(data["FP"],      ax = axes [0,1], shade = True, color = "Blue", fill = True,
            bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.boxplot(x = data["FP"],  ax = axes [1,0], orient = "h", color = "lightgreen", saturation
= 1,
            width = 0.7, dodge = True, fliersize = 3, linewidth =
2)
plt.tight_layout()
plt.show()
```



Estudio Visualización "LSS"

```
# Estudio Visualización "LSS"
# Métricas LSS
print("
=====")
print(f"                =====          MEDIA DE LSS:      {data['LSS'].mean():,.5f}
=====")
print(f"                =====          MEDIANA DE LSS:     {data['LSS'].median():,.5f}
=====")
print(f"                =====          SESGO DE LSS:       {data['LSS'].skew():,.5f}
=====")
print(f"                =====          VARIANZA DE LSS:     {data['LSS'].std():,.5f}
=====")
print("
=====")

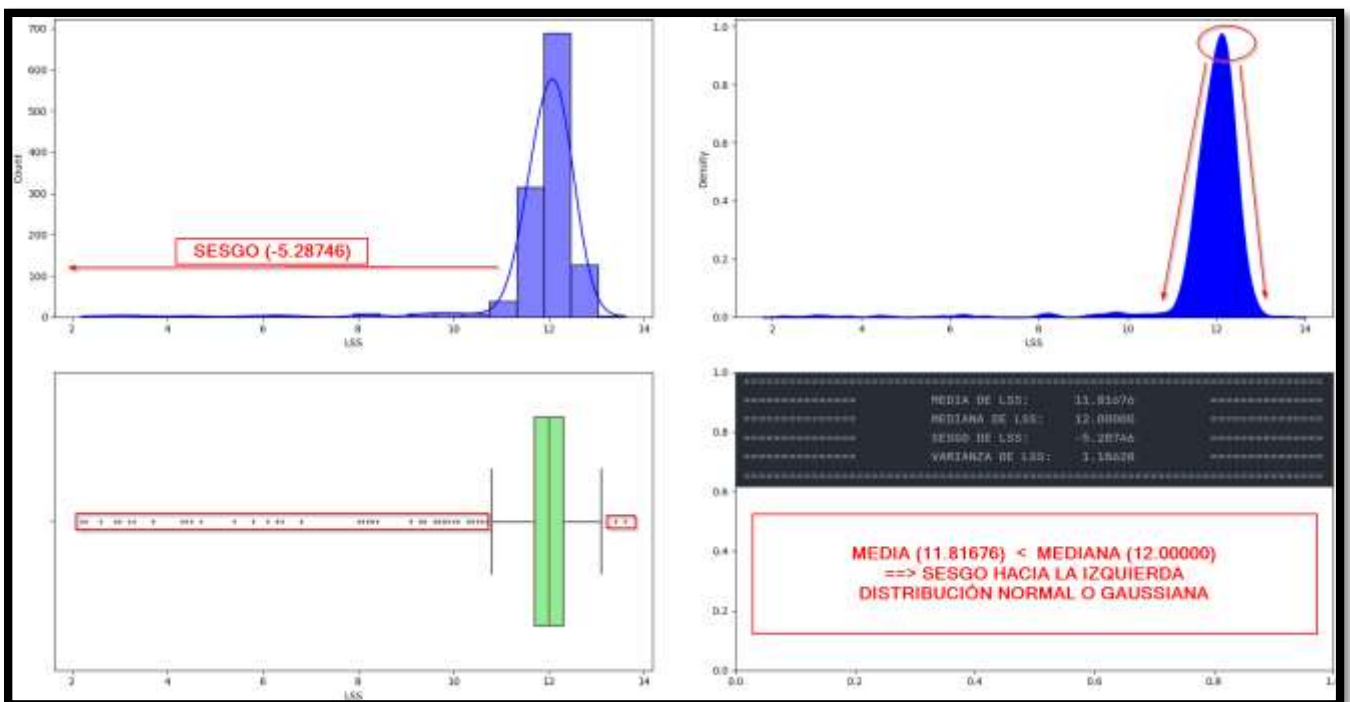
# Visualización Histograma, Densidad y Boxplot de LSS
f, axes = plt.subplots(2,2, figsize=(11.7,8.27))

sns.histplot(data["LSS"],      ax = axes [0,0], kde = True, bins = 20, color="Blue", fill=True)

sns.kdeplot(data["LSS"],      ax = axes [0,1], shade = True, color = "Blue", fill = True,
bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.boxplot(x = data["LSS"],  ax = axes [1,0], orient = "h", color = "lightgreen", saturation
= 1,
                                width = 0.7, dodge = True, fliersize = 3, linewidth = 2)

plt.tight_layout()
plt.show()
```



Estudio Visualización "OSC_VERTICAL"

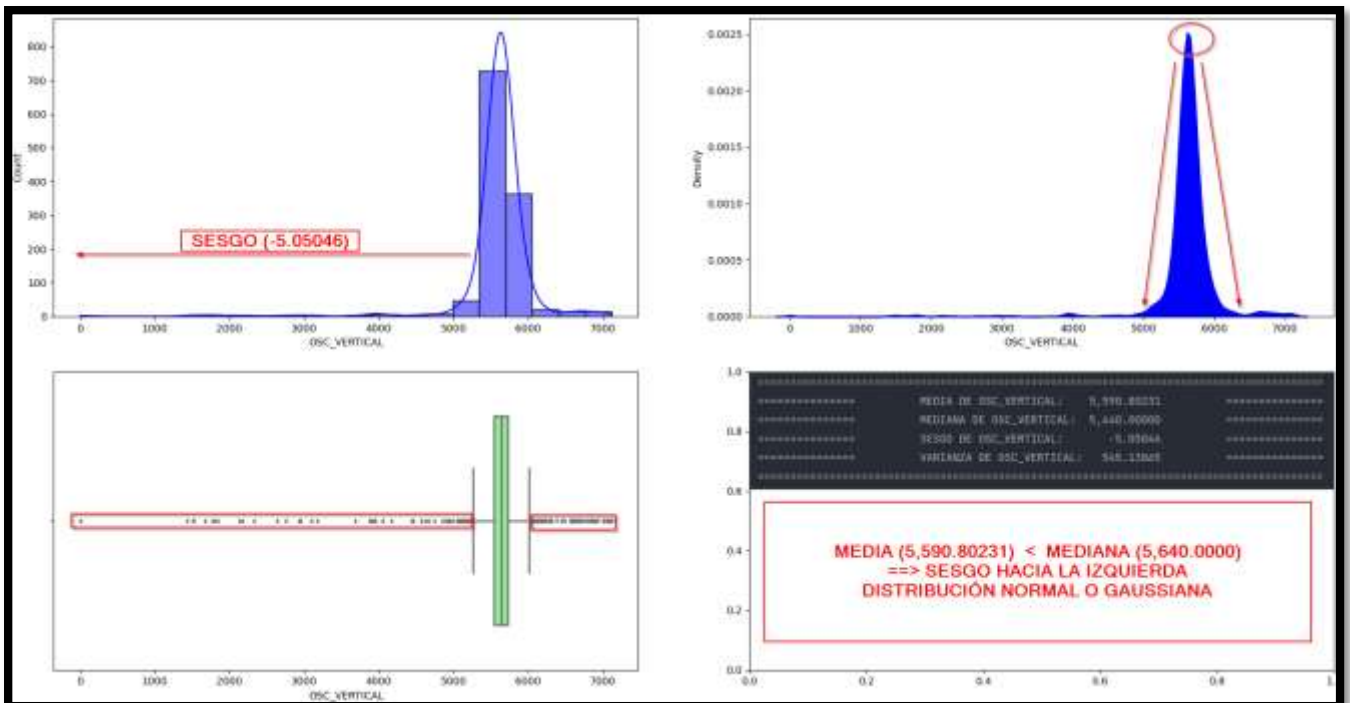
```
# Estudio Visualización "OSC_VERTICAL"
# Métricas OSC_VERTICAL
print("
=====")
print(f"
=====
{data['OSC_VERTICAL'].mean():.5f}
=====
print(f"
=====
{data['OSC_VERTICAL'].median():.5f}
=====
print(f"
=====
{data['OSC_VERTICAL'].skew():.5f}
=====
print(f"
=====
{data['OSC_VERTICAL'].std():.5f}
=====
print("
=====")

# Visualización Histograma, Densidad y Boxplot de OSC_VERTICAL
f, axes = plt.subplots(2,2, figsize=(11.7,8.27))

sns.histplot(data["OSC_VERTICAL"], ax = axes [0,0], kde = True, bins = 20, color="Blue",
fill=True)

sns.kdeplot(data["OSC_VERTICAL"], ax = axes [0,1], shade = True, color = "Blue", fill =
True,
bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.boxplot(x = data["OSC_VERTICAL"], ax = axes [1,0], orient = "h", color = "lightgreen",
saturation = 1,
width = 0.7, dodge = True, fliersize = 3, linewidth =
2)
plt.tight_layout()
plt.show()
```



Estudio Visualización "L_ZANCADA"

```
# Estudio Visualización "L_ZANCADA"
# Métricas L_ZANCADA
print("=====")
print(f"===== MEDIA DE L_ZANCADA: {data['L_ZANCADA'].mean():.5f} =====")
print(f"===== MEDIANA DE L_ZANCADA: {data['L_ZANCADA'].median():.5f} =====")
print(f"===== SESGO DE L_ZANCADA: {data['L_ZANCADA'].skew():.5f} =====")
print(f"===== VARIANZA DE L_ZANCADA: {data['L_ZANCADA'].std():.5f} =====")
print("=====")

# Visualización Histograma, Densidad y Boxplot de L_ZANCADA
f, axes = plt.subplots(2,2, figsize=(11.7,8.27))

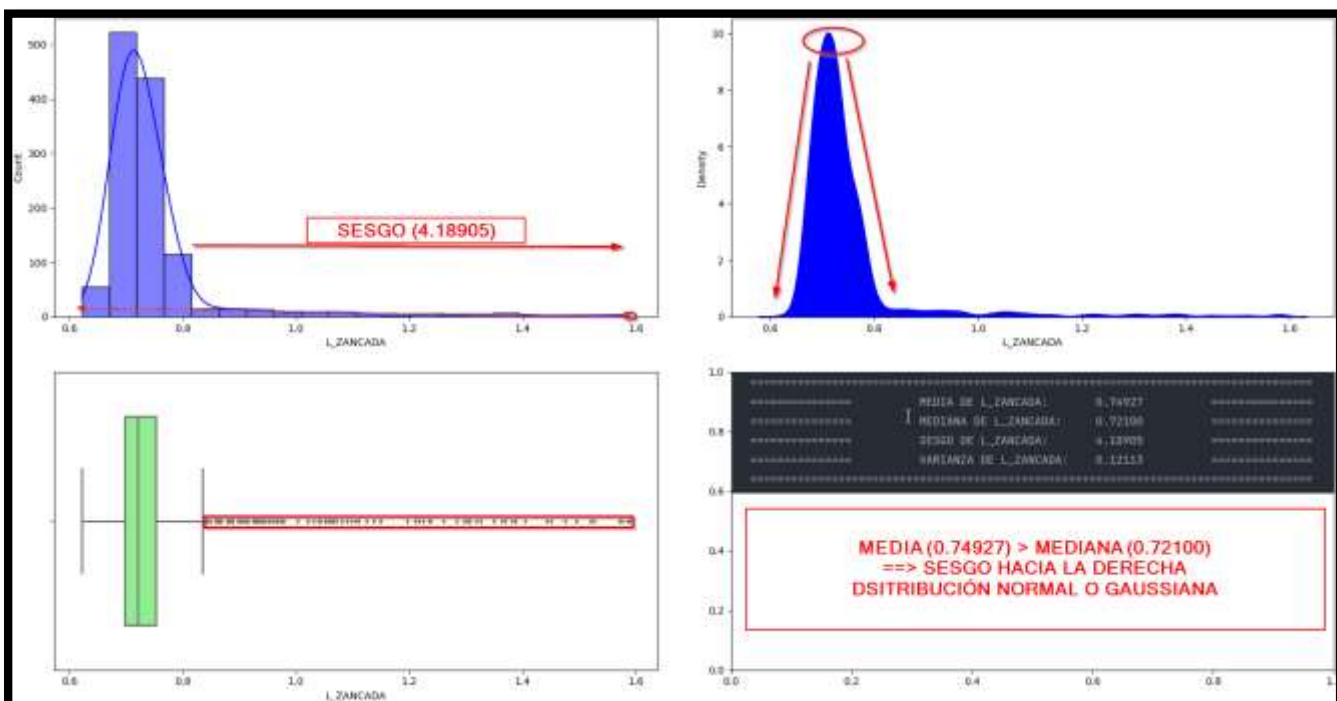
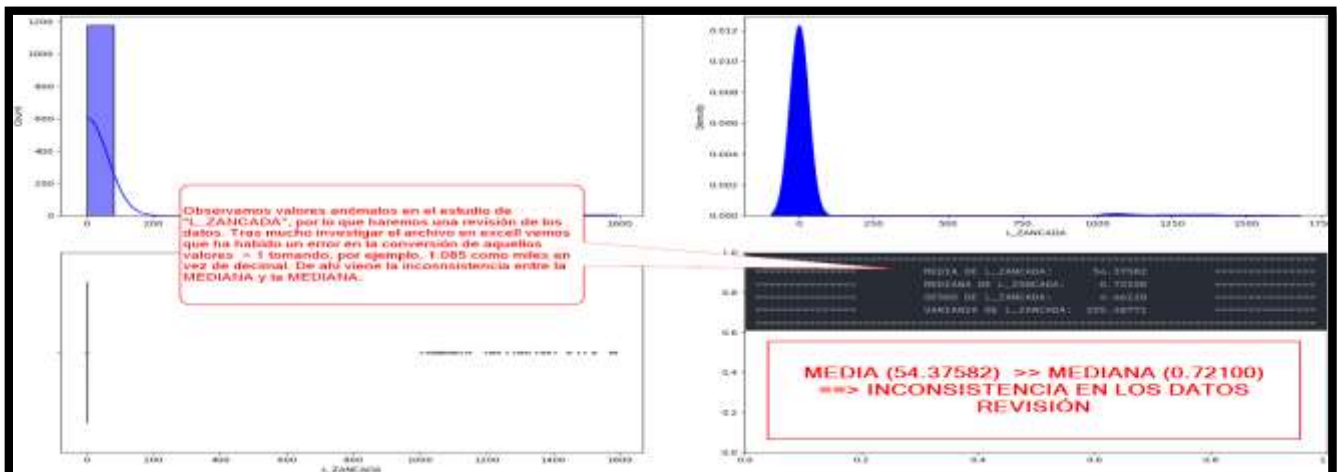
sns.histplot(data["L_ZANCADA"], ax = axes [0,0], kde = True, bins = 20, color="Blue", fill=True)

sns.kdeplot(data["L_ZANCADA"], ax = axes [0,1], shade = True, color = "Blue", fill = True,
            bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.boxplot(x = data["L_ZANCADA"], ax = axes [1,0], orient = "h", color = "lightgreen", saturation = 1,
            width = 0.7, dodge = True, fliersize = 3, linewidth = 2)

plt.tight_layout()
plt.show()
```

!!!! OJO-ERROR!!!!



Estudio Visualización "RFP"

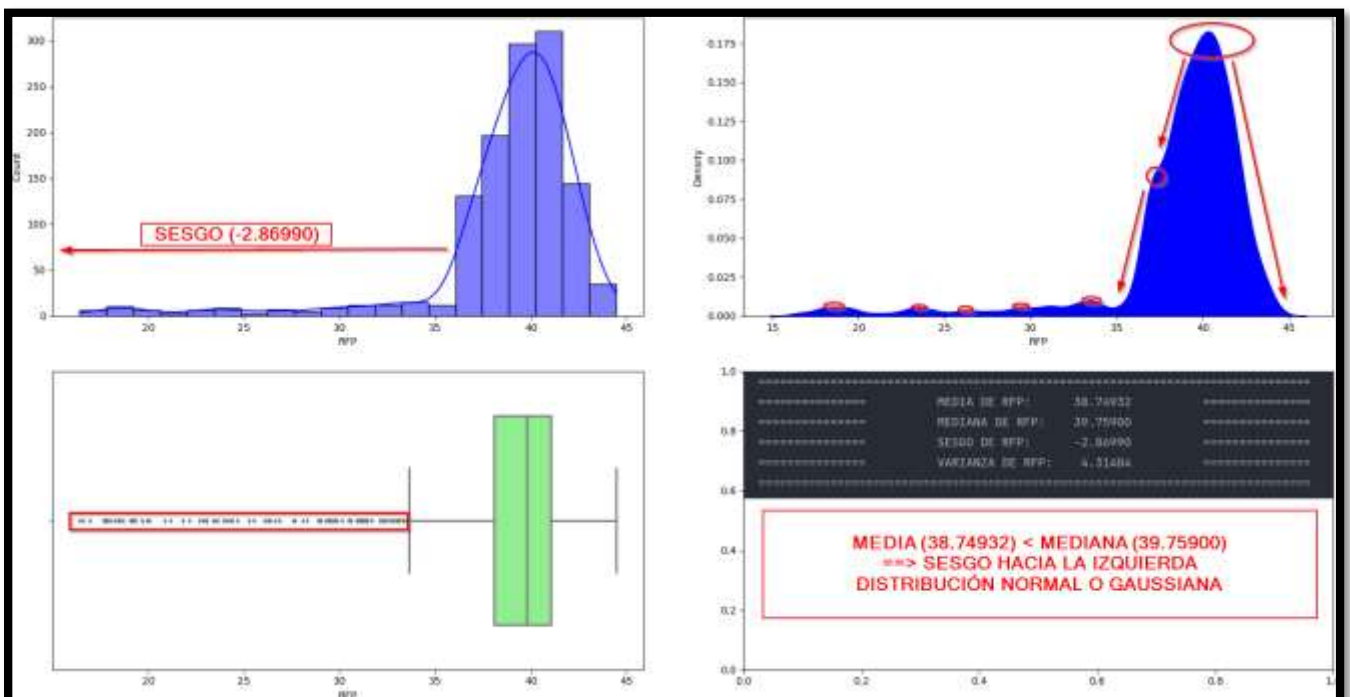
```
# Estudio Visualización "RFP"
# Métricas RFP
print("
=====")
print(f"                =====          MEDIA DE RFP:      {data['RFP'].mean():.5f}
=====")
print(f"                =====          MEDIANA DE RFP:     {data['RFP'].median():.5f}
=====")
print(f"                =====          SESGO DE RFP:        {data['RFP'].skew():.5f}
=====")
print(f"                =====          VARIANZA DE RFP:      {data['RFP'].std():.5f}
=====")
print("
=====")

# Visualización Histograma, Densidad y Boxplot de RFP
f, axes = plt.subplots(2,2, figsize=(11.7,8.27))

sns.histplot(data["RFP"],      ax = axes [0,0], kde = True, bins = 20, color="Blue", fill=True)

sns.kdeplot(data["RFP"],      ax = axes [0,1], shade = True, color = "Blue", fill = True,
            bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.boxplot(x = data["RFP"],   ax = axes [1,0], orient = "h", color = "lightgreen", saturation
            = 1,
            width = 0.7, dodge = True, fliersize = 3, linewidth =
            2)
plt.tight_layout()
plt.show()
```



Estudio Visualización "RLSS"

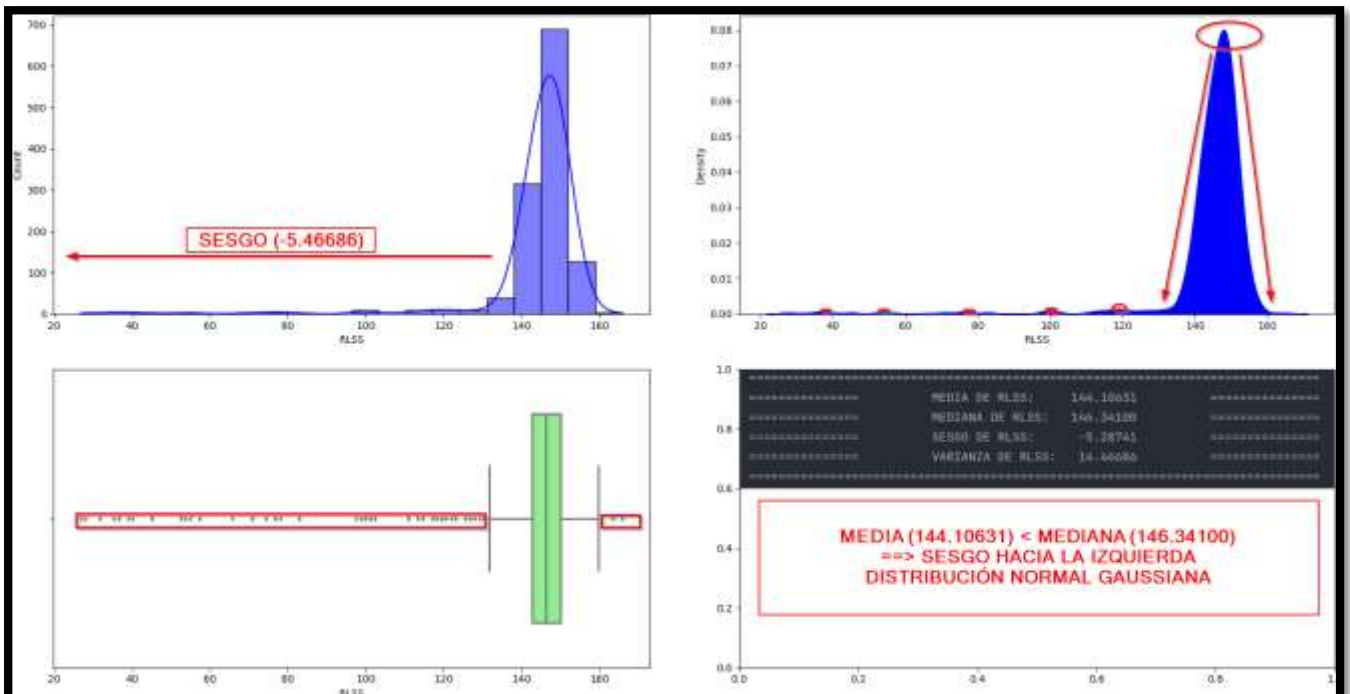
```
# Estudio Visualización "RLSS"
# Métricas RLSS
print("
=====")
print(f"                =====          MEDIA DE RLSS:      {data['RLSS'].mean():,.5f}
=====")
print(f"                =====          MEDIANA DE RLSS:
{data['RLSS'].median():,.5f}          =====")
print(f"                =====          SESGO DE RLSS:      {data['RLSS'].skew():,.5f}
=====")
print(f"                =====          VARIANZA DE RLSS:    {data['RLSS'].std():,.5f}
=====")
print("
=====")

# Visualización Histograma, Densidad y Boxplot de RLSS
f, axes = plt.subplots(2,2, figsize=(11.7,8.27))

sns.histplot(data["RLSS"],      ax = axes [0,0], kde = True, bins = 20, color="Blue",
fill=True)

sns.kdeplot(data["RLSS"],      ax = axes [0,1], shade = True, color = "Blue", fill = True,
bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.boxplot(x = data["RLSS"],  ax = axes [1,0], orient = "h", color = "lightgreen",
saturation = 1,
width = 0.7, dodge = True, fliersize = 3, linewidth =
2)
plt.tight_layout()
plt.show()
```



Estudio Visualización "ROV"

```
# Estudio Visualización "ROV"
# Métricas ROV
print("
=====")
print(f"                =====          MEDIA DE ROV:          {data['ROV'].mean():,.5f}
=====")
print(f"                =====          MEDIANA DE ROV:
{data['ROV'].median():,.5f}          =====")
print(f"                =====          SESGO DE ROV:          {data['ROV'].skew():,.5f}
=====")
print(f"                =====          VARIANZA DE ROV:          {data['ROV'].std():,.5f}
=====")
print("
=====")

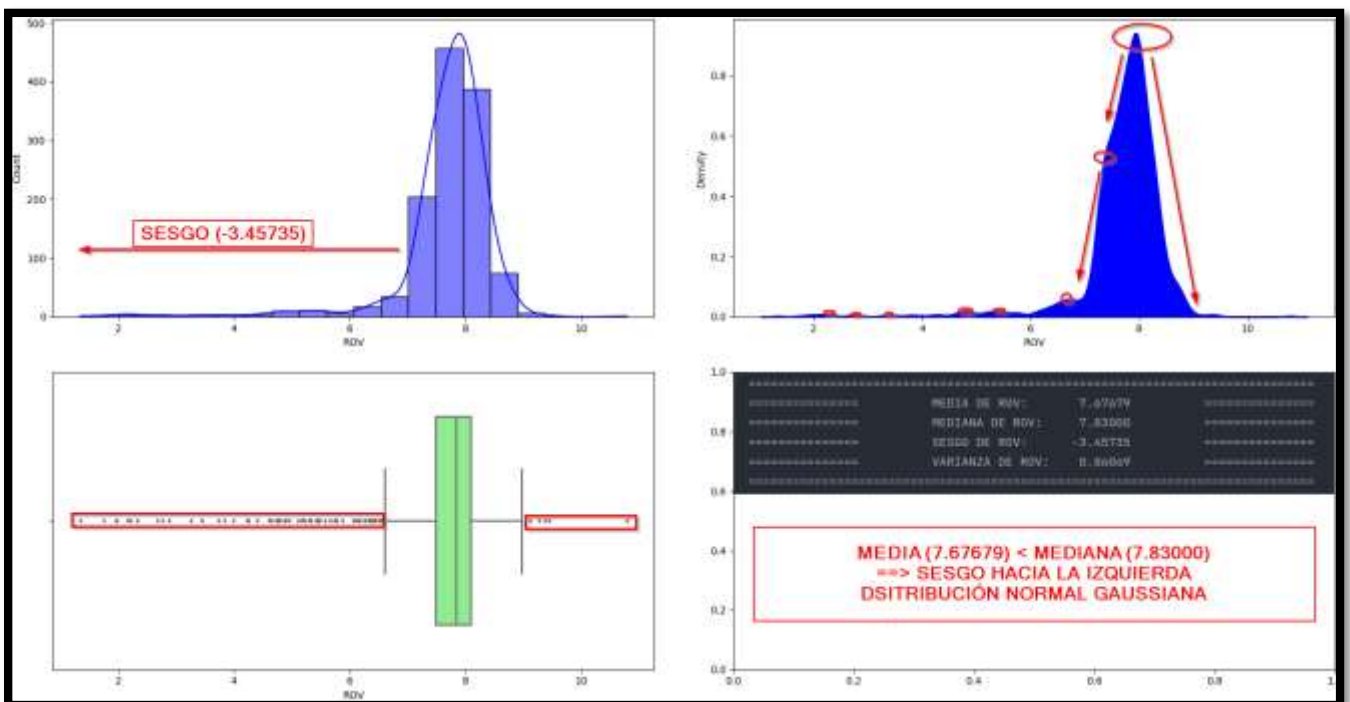
# Visualización Histograma, Densidad y Boxplot de ROV
f, axes = plt.subplots(2,2, figsize =(11.7,8.27))

sns.histplot(data["ROV"],      ax = axes [0,0], kde = True, bins = 20, color="Blue", fill=True)

sns.kdeplot(data["ROV"],      ax = axes [0,1], shade = True, color = "Blue", fill = True,
bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.boxplot(x = data["ROV"],  ax = axes [1,0], orient = "h", color = "lightgreen", saturation
= 1,
                                width = 0.7, dodge = True, fliersize = 3, linewidth = 2)

plt.tight_layout()
plt.show()
```



Estudio Visualización "RE"

```
# Estudio Visualización "RE"
# Métricas RE
print("
=====")
print(f"                =====          MEDIA DE RE:      {data['RE'].mean():,.5f}
=====")
print(f"                =====          MEDIANA DE RE:     {data['RE'].median():,.5f}
=====")
print(f"                =====          SESGO DE RE:       {data['RE'].skew():,.5f}
=====")
print(f"                =====          VARIANZA DE RE:    {data['RE'].std():,.5f}
=====")
print("
=====")

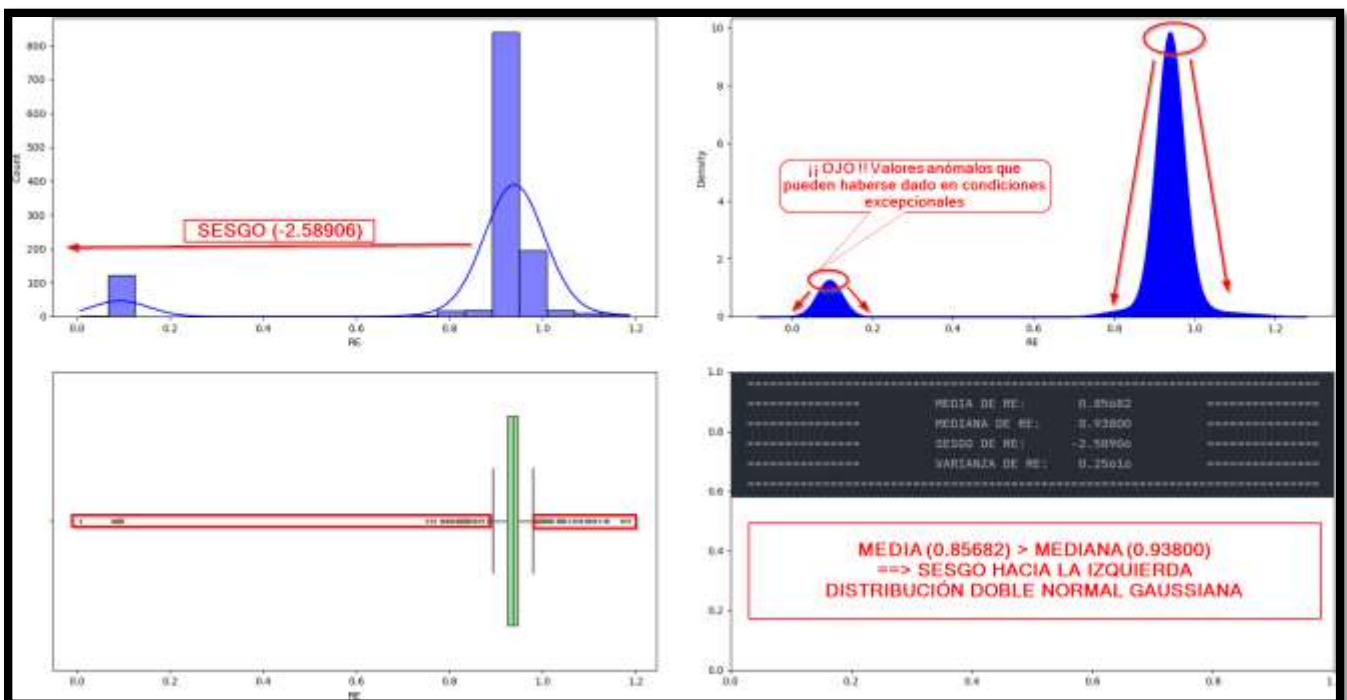
# Visualización Histograma, Densidad y Boxplot de RE
f, axes = plt.subplots(2,2, figsize=(11.7,8.27))

sns.histplot(data["RE"],          ax = axes [0,0], kde = True, bins = 20, color="Blue", fill=True)

sns.kdeplot(data["RE"],          ax = axes [0,1], shade = True, color = "Blue", fill = True,
            bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.boxplot(x = data["RE"],      ax = axes [1,0], orient = "h", color = "lightgreen", saturation =
1,
            width = 0.7, dodge = True, fliersize = 3, linewidth = 2)

plt.tight_layout()
plt.show()
```



Estudio Visualización "AIRE"

```
# Estudio Visualización "AIRE"
# Métricas AIRE
print("
=====")
print(f"                =====          MEDIA DE AIRE:      {data['AIRE'].mean():,.5f}
=====")
print(f"                =====          MEDIANA DE AIRE:
{data['AIRE'].median():,.5f}          =====")
print(f"                =====          SESGO DE AIRE:      {data['AIRE'].skew():,.5f}
=====")
print(f"                =====          VARIANZA DE AIRE:    {data['AIRE'].std():,.5f}
=====")
print("
=====")

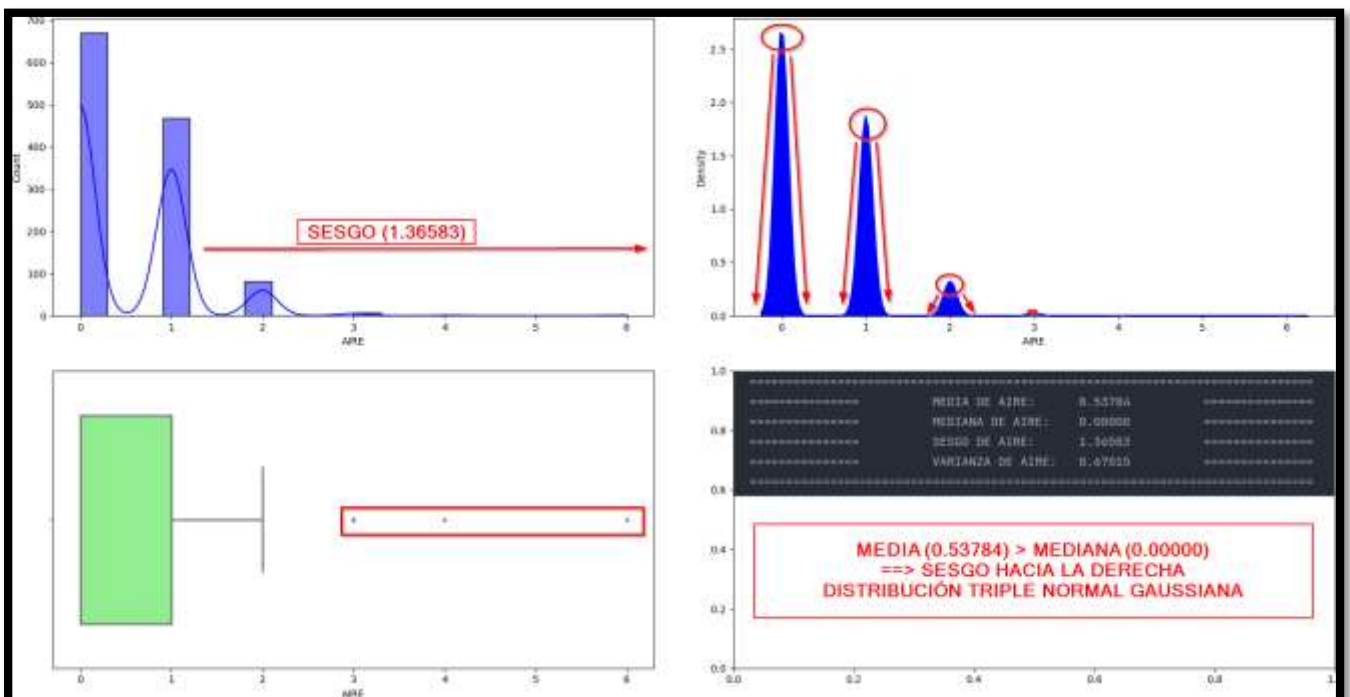
# Visualización Histograma, Densidad y Boxplot de AIRE
f, axes = plt.subplots(2,2, figsize=(11.7,8.27))

sns.histplot(data["AIRE"],      ax = axes [0,0], kde = True, bins = 20, color="Blue",
fill=True)

sns.kdeplot(data["AIRE"],      ax = axes [0,1], shade = True, color = "Blue", fill = True,
bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.boxplot(x = data["AIRE"],  ax = axes [1,0], orient = "h", color = "lightgreen", saturation
= 1,
                                width = 0.7, dodge = True, fliersize = 3, linewidth = 2)

plt.tight_layout()
plt.show()
```



Estudio Visualización "PENDIENTE"

```
# Estudio Visualización "PENDIENTE"
# Métricas PENDIENTE
print("=====")
print(f"=====                MEDIA DE PENDIENTE:      {data['PENDIENTE'].mean():.5f}
=====")
print(f"=====                MEDIANA DE PENDIENTE:      {data['PENDIENTE'].median():.5f}
=====")
print(f"=====                SESGO DE PENDIENTE:         {data['PENDIENTE'].skew():.5f}
=====")
print(f"=====                VARIANZA DE PENDIENTE:      {data['PENDIENTE'].std():.5f}
=====")
print("=====")

# Visualización Histograma, Densidad y Boxplot de PENDIENTE
f, axes = plt.subplots(2,2, figsize=(11.7,8.27))

sns.histplot(data["PENDIENTE"], ax = axes [0,0], kde = True, bins = 20, color="Blue", fill=True)

sns.kdeplot(data["PENDIENTE"], ax = axes [0,1], shade = True, color = "Blue", fill = True,
bw_adjust=.5, clip_on=False, alpha=1, linewidth=1.5)

sns.boxplot(x = data["PENDIENTE"], ax = axes [1,0], orient = "h", color = "lightgreen", saturation = 1,
width = 0.7, dodge = True, fliersize = 3, linewidth = 2)

plt.tight_layout()
plt.show()
```

!!!! OJO-ERROR!!!!

