Carol Arrieta Moreno

A01275465

Momento de Retroalimentación: Módulo 1 Técnicas de procesamiento de datos para el análisis estadístico y para la construcción de modelos (Portafolio Análisis)

Primera entrega

```
import numpy as np
from pandas import DataFrame
import pandas as pd
import seaborn as sns
%matplotlib inline
```

```
#Conversion de CSV a DataFrame
train= pd.read_csv(r"precios_autos.csv")
train.head(10)
```

	symboling	CarName	fueltype	carbody	drivewheel	enginelocation	wheelbase	carlength	car
0	3	alfa-romero giulia	gas	convertible	rwd	front	88.6	168.8	
1	3	alfa-romero stelvio	gas	convertible	rwd	front	88.6	168.8	
2	1	alfa-romero Quadrifoglio	gas	hatchback	rwd	front	94.5	171.2	
3	2	audi 100 ls	gas	sedan	fwd	front	99.8	176.6	
4	2	audi 100ls	gas	sedan	4wd	front	99.4	176.6	
5	2	audi fox	gas	sedan	fwd	front	99.8	177.3	
6	1	audi 100ls	gas	sedan	fwd	front	105.8	192.7	
7	1	audi 5000	gas	wagon	fwd	front	105.8	192.7	
8	1	audi 4000	gas	sedan	fwd	front	105.8	192.7	
9	0	audi 5000s (diesel)	gas	hatchback	4wd	front	99.5	178.2	

10 rows × 21 columns

```
#Analisis de la Estadistica Descrpitiva de los datos
print(train.describe())
```

```
        symboling   wheelbase   carlength    carwidth   carheight  \
count  205.000000  205.000000  205.000000  205.000000  205.000000
mean     0.834146   98.756585  174.049268   65.907805   53.724878
std      1.245307    6.021776   12.337289    2.145204    2.443522
min     -2.000000   86.600000  141.100000   60.300000   47.800000
25%      0.000000   94.500000  166.300000   64.100000   52.000000
50%      1.000000   97.000000  173.200000   65.500000   54.100000
75%      2.000000  102.400000  183.100000   66.900000   55.500000
max      3.000000  120.900000  208.100000   72.300000   59.800000

        curbweight  enginesize      stroke  compressionratio  horsepower  \
count   205.000000  205.000000  205.000000        205.000000  205.000000
mean   2555.565854  126.907317    3.255415         10.142537  104.117073
std     520.680204   41.642693    0.313597          3.972040   39.544167
min    1488.000000   61.000000    2.070000          7.000000   48.000000
25%    2145.000000   97.000000    3.110000          8.600000   70.000000
50%    2414.000000  120.000000    3.290000          9.000000   95.000000
75%    2935.000000  141.000000    3.410000          9.400000  116.000000
max    4066.000000  326.000000    4.170000         23.000000  288.000000

           peakrpm     citympg  highwaympg         price
count   205.000000  205.000000  205.000000    205.000000
mean   5125.121951   25.219512   30.751220  13276.710571
std     476.985643    6.542142    6.886443   7988.852332
min    4150.000000   13.000000   16.000000   5118.000000
25%    4800.000000   19.000000   25.000000   7788.000000
```

```
     50%    5200.000000   24.000000   30.000000  10295.000000
     75%    5500.000000   30.000000   34.000000  16503.000000
     max    6600.000000   49.000000   54.000000  45400.000000
```

```
#Comando para ver el tipo de dato de cada columna en el dataframe y cantidad de valores no nulos
train.info()
```

```
     <class 'pandas.core.frame.DataFrame'>
     RangeIndex: 205 entries, 0 to 204
     Data columns (total 21 columns):
      #   Column            Non-Null Count  Dtype
     ---  ------            --------------  -----
      0   symboling         205 non-null    int64
      1   CarName           205 non-null    object
      2   fueltype          205 non-null    object
      3   carbody           205 non-null    object
      4   drivewheel        205 non-null    object
      5   enginelocation    205 non-null    object
      6   wheelbase         205 non-null    float64
      7   carlength         205 non-null    float64
      8   carwidth          205 non-null    float64
      9   carheight         205 non-null    float64
      10  curbweight        205 non-null    int64
      11  enginetype        205 non-null    object
      12  cylindernumber    205 non-null    object
      13  enginesize        205 non-null    int64
      14  stroke            205 non-null    float64
      15  compressionratio  205 non-null    float64
      16  horsepower        205 non-null    int64
      17  peakrpm           205 non-null    int64
      18  citympg           205 non-null    int64
      19  highwaympg        205 non-null    int64
      20  price             205 non-null    float64
     dtypes: float64(7), int64(7), object(7)
     memory usage: 33.8+ KB
```

```
#Transformacion Binaria a la columna fueltype
from sklearn import preprocessing
label = preprocessing.LabelEncoder()

train['fueltype']= label.fit_transform(train['fueltype'])
print(train['fueltype'].unique())
```

```
     [1 0]
```

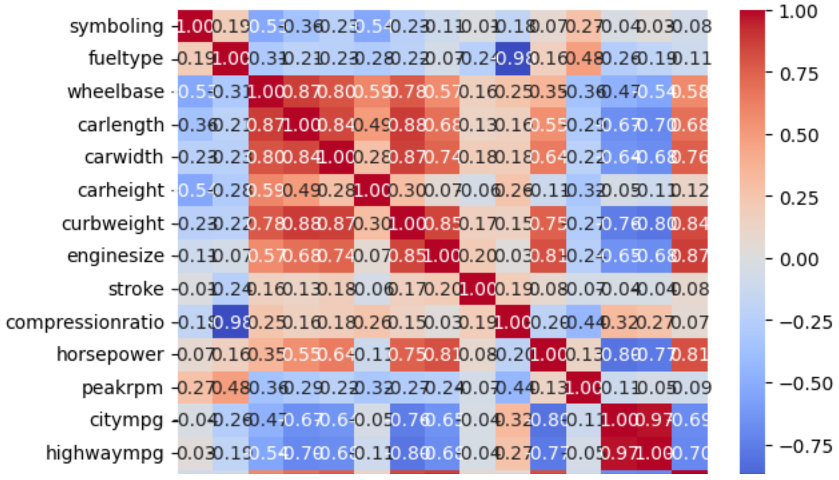
```
train.head()
```

	symboling	CarName	fueltype	carbody	drivewheel	enginelocation	wheelbase	carlength	car
0	3	alfa-romero giulia	1	convertible	rwd	front	88.6	168.8	
1	3	alfa-romero stelvio	1	convertible	rwd	front	88.6	168.8	
2	1	alfa-romero Quadrifoglio	1	hatchback	rwd	front	94.5	171.2	
3	2	audi 100 ls	1	sedan	fwd	front	99.8	176.6	
4	2	audi 100ls	1	sedan	4wd	front	99.4	176.6	

5 rows × 21 columns

```
# Mapa de correlación entre cada variable
sns.heatmap(train.corr(), annot = True, fmt = '.2f', cmap = 'coolwarm')
```

```
<ipython-input-20-7bfdaa43e6ed>:2: FutureWarning: The default value of numeric_only in DataFrame.co
  sns.heatmap(train.corr(), annot = True, fmt = '.2f', cmap = 'coolwarm')
<Axes: >
```



Limpieza de Datos: Train.csv

```
#Contamos los datos no nulos
train.count()
```

```
symboling           205
CarName             205
fueltype            205
carbody             205
drivewheel          205
enginelocation      205
wheelbase           205
carlength           205
carwidth            205
carheight           205
curbweight          205
enginetype          205
cylindernumber      205
enginesize          205
stroke              205
compressionratio    205
horsepower          205
peakrpm             205
citympg             205
highwaympg          205
price               205
dtype: int64
```

```
#Contamos los datos nulos de todas las columnas
train.isnull().sum()
```

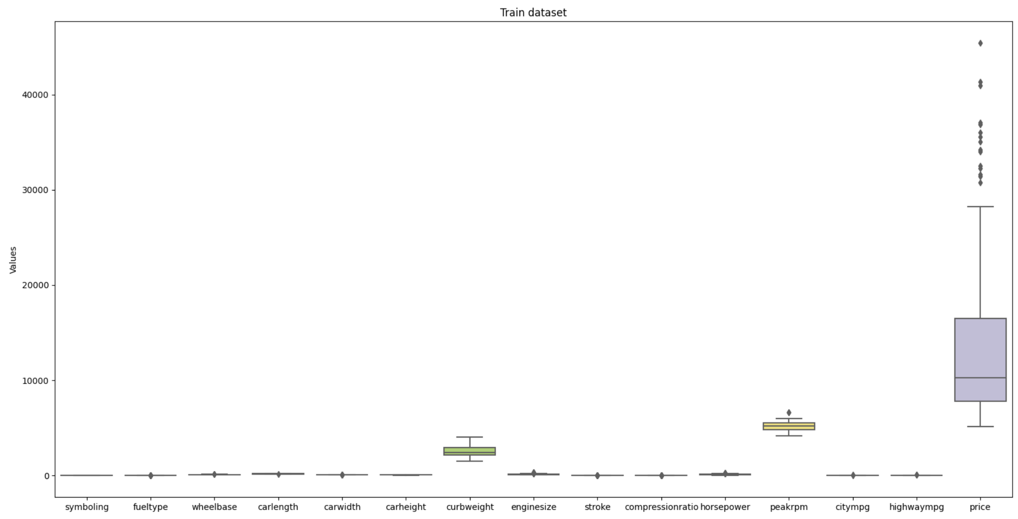
```
symboling           0
CarName             0
fueltype            0
carbody             0
drivewheel          0
enginelocation      0
wheelbase           0
carlength           0
carwidth            0
carheight           0
curbweight          0
enginetype          0
cylindernumber      0
enginesize          0
stroke              0
compressionratio    0
horsepower          0
peakrpm             0
citympg             0
highwaympg          0
price               0
dtype: int64
```

```
import matplotlib.pyplot as plt

# Create a boxplot using Seaborn
plt.figure(figsize=(20, 10))  # Optional: Set the figure size
sns.boxplot(data=train, palette="Set3")

# Optional: Add titles and labels
plt.title("Train dataset")
plt.ylabel("Values")


# Show the plot
plt.show()
```

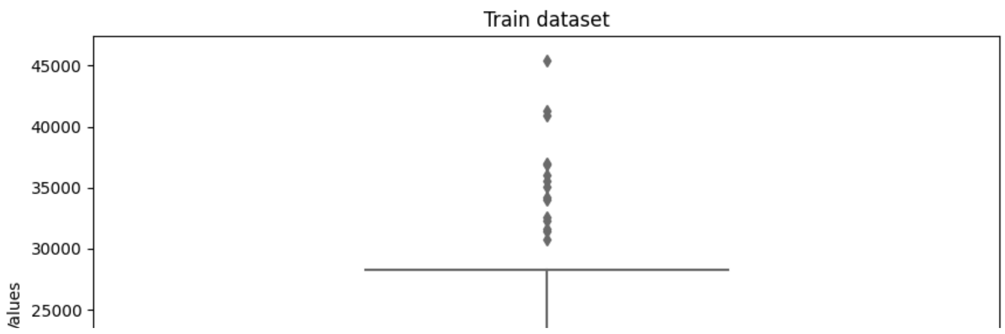


```
import seaborn as sns
import matplotlib.pyplot as plt


# Create a boxplot using Seaborn
plt.figure(figsize=(10, 6))  # Optional: Set the figure size
sns.boxplot(data=train.price, palette="Set3")

# Optional: Add titles and labels
plt.title("Train dataset")
plt.ylabel("Values")

# Show the plot
plt.show()
```

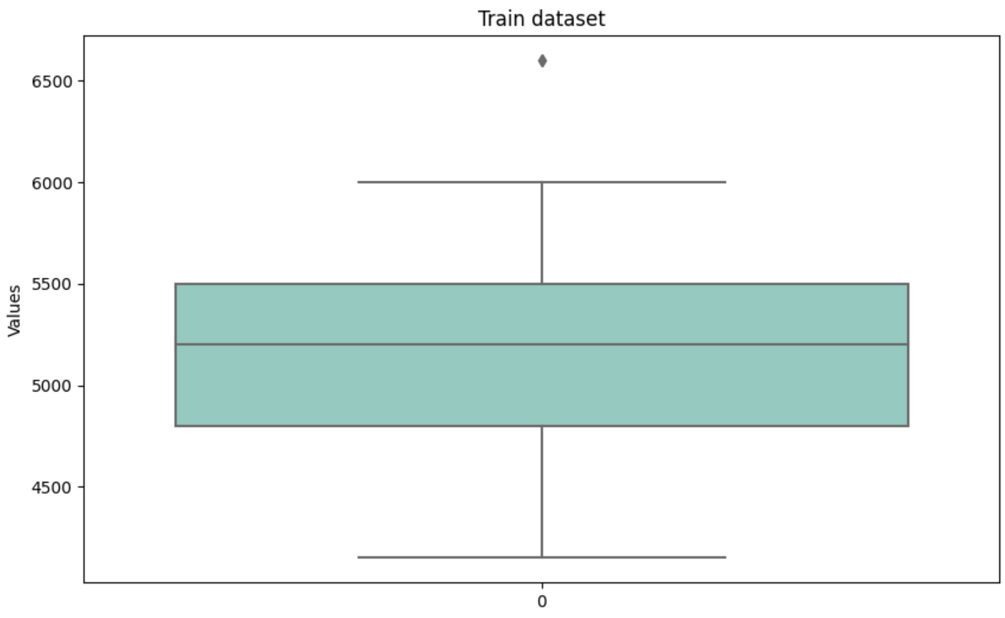


```
import seaborn as sns
import matplotlib.pyplot as plt


# Create a boxplot using Seaborn
plt.figure(figsize=(10, 6))  # Optional: Set the figure size
sns.boxplot(data=train.peakrpm, palette="Set3")

# Optional: Add titles and labels
plt.title("Train dataset")
plt.ylabel("Values")

# Show the plot
plt.show()
```

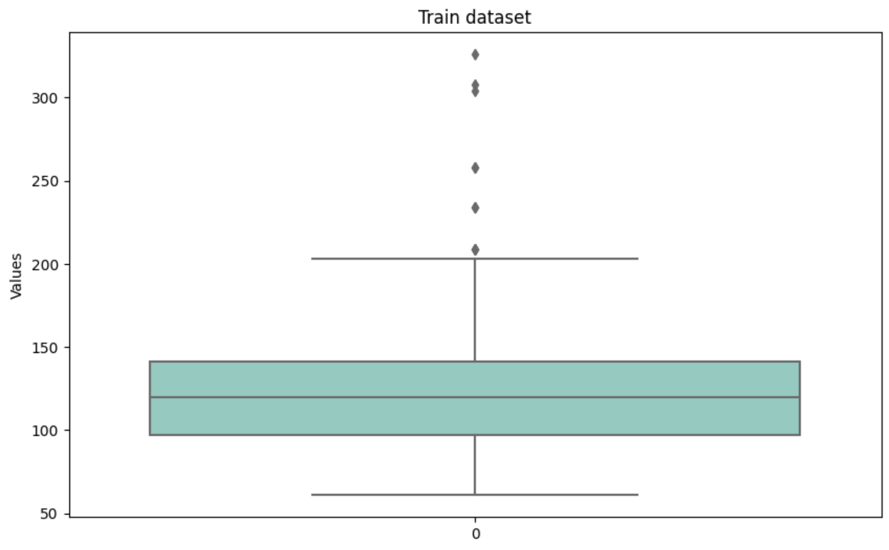


```
import seaborn as sns
import matplotlib.pyplot as plt


# Create a boxplot using Seaborn
plt.figure(figsize=(10, 6))  # Optional: Set the figure size
sns.boxplot(data=train.enginesize, palette="Set3")

# Optional: Add titles and labels
plt.title("Train dataset")
plt.ylabel("Values")

# Show the plot
plt.show()
```



```
train.describe()
```

	symboling	fueltype	wheelbase	carlength	carwidth	carheight	curbweight	enginesize
count	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000
mean	0.834146	0.902439	98.756585	174.049268	65.907805	53.724878	2555.565854	126.907317
std	1.245307	0.297446	6.021776	12.337289	2.145204	2.443522	520.680204	41.642693
min	-2.000000	0.000000	86.600000	141.100000	60.300000	47.800000	1488.000000	61.000000
25%	0.000000	1.000000	94.500000	166.300000	64.100000	52.000000	2145.000000	97.000000
50%	1.000000	1.000000	97.000000	173.200000	65.500000	54.100000	2414.000000	120.000000
75%	2.000000	1.000000	102.400000	183.100000	66.900000	55.500000	2935.000000	141.000000
max	3.000000	1.000000	120.900000	208.100000	72.300000	59.800000	4066.000000	326.000000

✓ 0s completed at 10:10 AM