Inteligencia de Negocios

“Clase Práctica 2”

Elaborado por: Diego García

Docente: Arlen Jeanette Lopez

Fecha de entrega: 18/09/2024

```
df = pd.read_csv(url)
df_original = pd.read_csv(url)
print(df.head())
print('\n')
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\ellie\Downloads\clpa2 - Diego Garcia> & C:/Users/ellie/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/ellie/Downloads/clpa2 - Diego Garcia/data.py"
   VIN (1-10)    County       City State  Postal Code  ...  Legislative District DOL Vehicle ID                Vehicle Location                                Electric Utility 2020 Census Tract
0  JTMAB3FV3P    Kitsap    Seabeck    WA      98380.0  ...                  35.0      240684006  POINT (-122.8728334 47.5798304)                              PUGET SOUND ENERGY INC      5.303509e+10
1  1N4AZ1CP6J    Kitsap  Bremerton    WA      98312.0  ...                  35.0      474183811  POINT (-122.6961203 47.5759584)                              PUGET SOUND ENERGY INC      5.303508e+10
2  5YJ3E1EA4L      King    Seattle    WA      98101.0  ...                  43.0      113120017  POINT (-122.3340795 47.6099315)  CITY OF SEATTLE - (WA)|CITY OF TACOMA - (WA)      5.303301e+10
3  1N4AZ0CP8E      King    Seattle    WA      98125.0  ...                  46.0      108188713    POINT (-122.304356 47.715668)  CITY OF SEATTLE - (WA)|CITY OF TACOMA - (WA)      5.303300e+10
4  1G1FX6S00H  Thurston       Yelm    WA      98597.0  ...                  20.0      176448940  POINT (-122.5715761 46.9095798)                              PUGET SOUND ENERGY INC      5.306701e+10

[5 rows x 17 columns]
```

Comenzamos por utilizar df.head() para tener una idea de si está trayendo los datos en nuestro dataframe.

```
print(df.info())
print('\n')
print(df.describe())
print('\n')
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205439 entries, 0 to 205438
Data columns (total 17 columns):
 #   Column                                             Non-Null Count   Dtype
---  ------                                             --------------   -----
 0   VIN (1-10)                                         205439 non-null  object
 1   County                                             205436 non-null  object
 2   City                                               205436 non-null  object
 3   State                                              205439 non-null  object
 4   Postal Code                                        205436 non-null  float64
 5   Model Year                                         205439 non-null  int64
 6   Make                                               205439 non-null  object
 7   Model                                              205438 non-null  object
 8   Electric Vehicle Type                              205439 non-null  object
 9   Clean Alternative Fuel Vehicle (CAFV) Eligibility  205439 non-null  object
 10  Electric Range                                     205431 non-null  float64
 11  Base MSRP                                          205431 non-null  float64
 12  Legislative District                               204997 non-null  float64
 13  DOL Vehicle ID                                     205439 non-null  int64
 14  Vehicle Location                                   205431 non-null  object
 15  Electric Utility                                   205436 non-null  object
 16  2020 Census Tract                                  205436 non-null  float64
dtypes: float64(5), int64(2), object(10)
memory usage: 26.6+ MB
None


        Postal Code     Model Year  Electric Range      Base MSRP  Legislative District  DOL Vehicle ID  2020 Census Tract
count  205436.000000  205439.000000  205431.000000  205431.000000         204997.000000    2.054390e+05       2.054360e+05
mean    98177.971870    2020.960363      52.164342     922.670532             28.970848    2.277156e+08       5.297704e+10
std      2419.037479       2.989059      88.075859    7761.753602             14.910052    7.205737e+07       1.588435e+09
min      1731.000000    1997.000000       0.000000       0.000000              1.000000    4.469000e+03       1.001020e+09
25%     98052.000000    2019.000000       0.000000       0.000000             17.000000    1.935324e+08       5.303301e+10
50%     98125.000000    2022.000000       0.000000       0.000000             33.000000    2.382368e+08       5.303303e+10
75%     98372.000000    2023.000000      48.000000       0.000000             42.000000    2.618718e+08       5.305307e+10
max     99577.000000    2025.000000     337.000000  845000.000000             49.000000    4.792548e+08       5.602100e+10
```

Después, utilizamos df.info() y df.describe() para poder entender mejor los datos con los que estamos trabajando (sus tipos de datos, cuantas columnas hay, etc).

```
19  importantes = ['Make', 'Model Year', 'Electric Range', 'Electric Vehicle Type']
20  print(df[importantes].head())
21  print('\n')
22
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

        Make  Model Year  Electric Range                 Electric Vehicle Type
0     TOYOTA        2023            42.0  Plug-in Hybrid Electric Vehicle (PHEV)
1     NISSAN        2018           151.0           Battery Electric Vehicle (BEV)
2      TESLA        2020           266.0           Battery Electric Vehicle (BEV)
3     NISSAN        2014            84.0           Battery Electric Vehicle (BEV)
4  CHEVROLET        2017           238.0           Battery Electric Vehicle (BEV)
```

Ahora, tomaré los datos más relevantes del data frame el cual serán transformados y utilizados para nuestro beneficio desde el punto de vista de una empresa.

```
22
23  missing_values = df.isnull().sum()
24  print(missing_values[missing_values > 0])
25  print('\n')
26  df['Electric Range'] = df['Electric Range'].fillna(df['Electric Range'].median(), inplace=True)
```

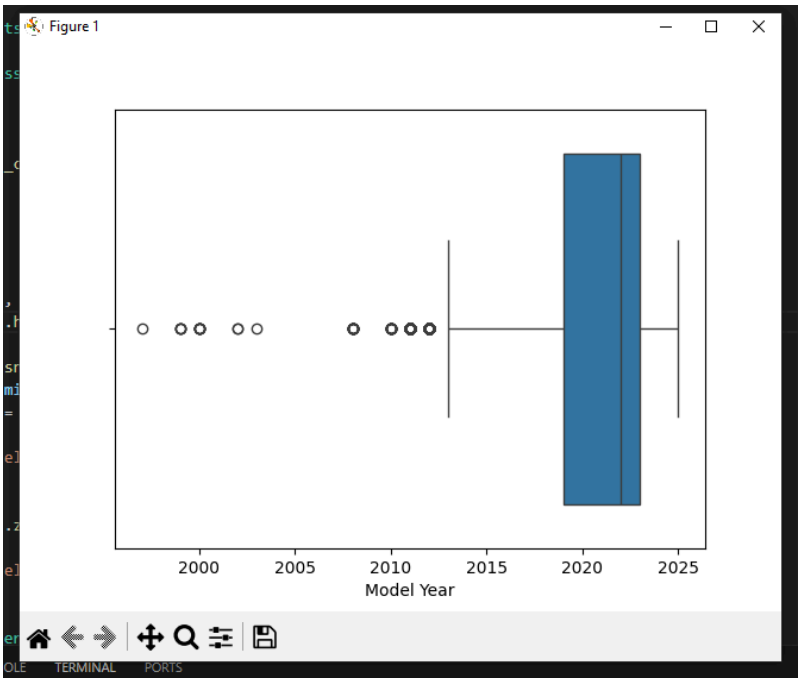
```
County                   3
City                     3
Postal Code              3
Model                    1
Electric Range           8
Base MSRP                8
Legislative District   442
Vehicle Location         8
Electric Utility         3
2020 Census Tract        3
dtype: int64
```

Una vez tomado los datos más importantes, los pasaremos mediante df.isnull().sum(), el cual mostrará cuántos datos son null en cada una de las columnas.

```
df['Electric Range'] = df['Electric Range'].fillna(df['Electric Range'].median(), inplace=True)
```

Sabiendo que ‘Electric Range’ tiene datos nulos utilizaremos df.median() para imputar estos datos y poder tener una compilación de datos más completa.

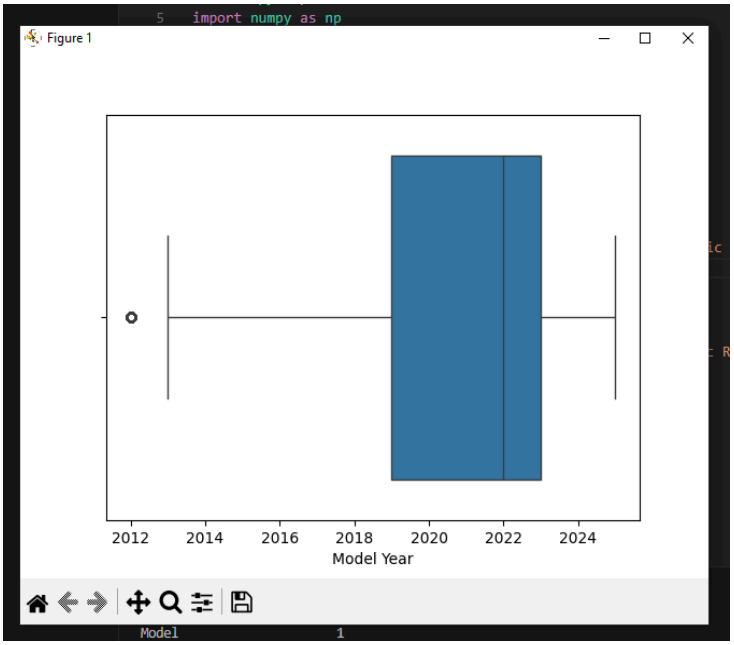
```
sns.boxplot(x=df['Model Year'])
plt.show()
```



Para detectar Outliers en nuestros valores más importantes utilizaremos un boxplot que nos muestre si existen o no. Como se puede observar en la imagen nuestra columna contiene outliers.

```
df = df[(np.abs(stats.zscore(df['Model Year'])) < 3)]

sns.boxplot(x=df['Model Year'])
plt.show()
```



Para deshacernos de estos, usaremos status.zscore eliminando así todos los datos que están fuera de 3 desviaciones estándar.

```python
scaler = StandardScaler()
df['Model Year_scaled'] = scaler.fit_transform(df[['Model Year']])

min_max_scaler = MinMaxScaler()
df['Model Year_normalized'] = min_max_scaler.fit_transform(df[['Model Year']])

print(df[['Model Year', 'Model Year_scaled', 'Model Year_normalized']].head())
print('\n')
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

	Model Year	Model Year_scaled	Model Year_normalized
0	2023	0.683962	0.846154
1	2018	-1.025075	0.461538
2	2020	-0.341460	0.615385
3	2014	-2.392304	0.153846
4	2017	-1.366882	0.384615

Aplicamos normalización y estandarización a nuestros valores importantes.

```python
Range_cut = pd.qcut(df['Model Year'], q=3, labels=['Old Car', 'Avg Car', 'New Car'])
df['Range_cat'] = Range_cut

category_mapping = {
    'Low Mileage': 0,
    'Avg Mileage': 1,
    'High Mileage': 2
}

print(df[['Model Year', 'Range_cat']].sample(8))
print('\n')
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

	Model Year	Range_cat
104215	2024	New Car
109905	2021	Old Car
31372	2023	Avg Car
158046	2023	Avg Car
165002	2019	Old Car
90684	2017	Old Car
128277	2020	Old Car
52463	2023	Avg Car

Y por último, creamos una nueva columna Categórica de Label Encoded, La cual nos muestra cuales carros eléctricos son antiguos, intermedios y más recientes.

```
57  print("Antes del preprocesamiento:")
58  print(df_original.describe())
59  print('\n')
60  print("Después del preprocesamiento:")
61  print(df.describe())
62
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

	Postal Code	Model Year	Electric Range	Base MSRP	Legislative District	DOL Vehicle ID	2020 Census Tract
count	205436.000000	205439.000000	205431.000000	205431.000000	204997.000000	2.054390e+05	2.054360e+05
mean	98177.971870	2020.960363	52.164342	922.670532	28.970848	2.277156e+08	5.297704e+10
std	2419.037479	2.989059	88.075859	7761.753602	14.910052	7.205737e+07	1.588435e+09
min	1731.000000	1997.000000	0.000000	0.000000	1.000000	4.469000e+03	1.001020e+09
25%	98052.000000	2019.000000	0.000000	0.000000	17.000000	1.935324e+08	5.303301e+10
50%	98125.000000	2022.000000	0.000000	0.000000	33.000000	2.382368e+08	5.303303e+10
75%	98372.000000	2023.000000	48.000000	0.000000	42.000000	2.618718e+08	5.305307e+10
max	99577.000000	2025.000000	337.000000	845000.000000	49.000000	4.792548e+08	5.602100e+10

Después del preprocesamiento:

	Postal Code	Model Year	Base MSRP	Legislative District	DOL Vehicle ID	2020 Census Tract	Model Year_scaled	Model Year_normalized
count	204668.000000	204671.000000	204663.000000	204229.000000	2.046710e+05	2.046680e+05	2.046710e+05	204671.000000
mean	98177.458675	2020.998984	900.916360	28.971782	2.278601e+08	5.297680e+10	-2.177757e-14	0.692230
std	2423.475073	2.925631	7608.005091	14.911623	7.200882e+07	1.591408e+09	1.000002e+00	0.225049
min	1731.000000	2012.000000	0.000000	1.000000	4.469000e+03	1.001020e+09	-3.075919e+00	0.000000
25%	98052.000000	2019.000000	0.000000	17.000000	1.936629e+08	5.303301e+10	-6.832674e-01	0.538462
50%	98125.000000	2022.000000	0.000000	33.000000	2.383740e+08	5.303303e+10	3.421548e-01	0.769231
75%	98372.000000	2023.000000	0.000000	42.000000	2.618904e+08	5.305307e+10	6.839622e-01	0.846154
max	99577.000000	2025.000000	845000.000000	49.000000	4.792548e+08	5.602100e+10	1.367577e+00	1.000000

PS C:\Users\ellie\Downloads\clpa2 - Diego Garcia>

Se puede notar que nuestra base de datos comenzó con datos nulos en nuestras columnas importantes y finalizó teniendo columnas adicionales, además de Estandarizar y Normalizar nuestros valores importantes y categorizandolos.