INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

Analítica y visualización de datos

Flores Estrada Ituriel Enrique

Práctica 1

Quezada Espínola Paulina Yaëlle

México CDMX a 24 de octubre de 2025

INTRODUCCIÓN.

En esta práctica para la materia de Analítica y visualización de datos se documenta el análisis exploratorio, limpieza y transformación de datos, así como la ingeniería de características aplicado al conjunto de datos extraído de kaggle Cars4u.
Se presenta el proceso hecho en Python, sección por sección, donde se describirá de manera puntual con ayuda de imágenes y gráficos los resultados obtenidos en cada parte de código.
Finalmente se presentan los resultados mostrando un antes y un después de las etapas del preprocesamiento.

1. *ANÁLISIS EXPLORATORIO DE DATOS.*

En esta primer parte de la práctica haremos una vista previa de cómo está estructurado el dataset. Sin este paso, nuestros análisis básicamente serían a ciegas.

Primeramente, con .info() podemos observar que esta función nos ofrece toda la información básica del dataset; se conbvirtió a DataFrame, la dimensión, nombre de las columnas, valores no-nulos, tipos de dato de cada columna y la memoria usada.

DATOS GENERALES

[3] ✓ 0 s

```
#Información del dataset como número y nombre de columnas,
#conteo de valores no nulos y tipo de dato de la columna
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7253 entries, 0 to 7252
Data columns (total 14 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   S.No.              7253 non-null   int64  
 1   Name               7253 non-null   object 
 2   Location           7253 non-null   object 
 3   Year               7253 non-null   int64  
 4   Kilometers_Driven  7253 non-null   int64  
 5   Fuel_Type          7253 non-null   object 
 6   Transmission       7253 non-null   object 
 7   Owner_Type         7253 non-null   object 
 8   Mileage            7251 non-null   object 
 9   Engine             7207 non-null   object 
 10  Power              7207 non-null   object 
 11  Seats              7200 non-null   float64
 12  New_Price          1006 non-null   object 
 13  Price              6019 non-null   float64
dtypes: float64(2), int64(3), object(9)
memory usage: 793.4+ KB
```

Figura 1. Datos generales

Previamente comenté en el notebook, que podemos observar pocas columnas marcadas como numéricas (S.No., Year, Kilometers_Driven, Seats, Price) mientras que el resto de columnas son categóricas (Name, Location, Fuel_Type, Transmission, Owner_Type, Mileage, Engine, Power, New_Price), lo cual muestra que es necesario hacer varios procesos para tener "orden" en las dimensiones. Esto ya que al ojo humano, se puede deducir que por ejemplo Mileage (Millaje) es una unidad numérica pero está siendo contada por python como categórica debido al formato del registro.
Esto lo podemos corroborar al mostrar los primeros y últimos elementos del dataset, viendo los registros se puede notar que las dimensiones que deberían ser numéricas, no lo son para el programa debido a que está escrito el registro con el número y acompañado de unidades de medición como kmpl, CC, bhp y Lakh.
También obtuvimos los valores duplicados de cada columna, resltando que la columna S.No. es un identificador y no nos servirá posteriormente para analizar.

Después obtenemos el tamaño del dataset y la descripción estadística de las dimensiones numéricas, esto nos sirve para entender el comportamiento general de las variables antes de aplicar modelos o hacer análisis.
En este caso; se puede observar que existe una dispersión general en Kilometers_Driven y Price, muestran alta variabilidad con su desviación estándar y máximo siendo mayor que su media.

ANÁLISIS NUMÉRICO

[7] ✓ 0 s

```
#Dimensión del dataset, queremos saber la cantidad de datos con los que trabajaremos
print("Número de filas,columnas:",df.shape)
```

```
Número de filas,columnas: (7253, 14)
```

[8]

```
#Descripcion estadistica de las dimensiones numéricas
dim_num= df.select_dtypes(include= np.number).columns.tolist()
df[dim_num].describe()
```

	S.No.	Year	Kilometers_Driven	Seats	Price
count	7253.000000	7253.000000	7.253000e+03	7200.000000	6019.000000
mean	3626.000000	2013.365366	5.869906e+04	5.279722	9.479468
std	2093.905084	3.254421	8.442772e+04	0.811660	11.187917
min	0.000000	1996.000000	1.710000e+02	0.000000	0.440000
25%	1813.000000	2011.000000	3.400000e+04	5.000000	3.500000
50%	3626.000000	2014.000000	5.341600e+04	5.000000	5.640000
75%	5439.000000	2016.000000	7.300000e+04	5.000000	9.950000
max	7252.000000	2019.000000	6.500000e+06	10.000000	160.000000

Figura 2. Análisis numérico

Cabe mencionar que la moda y el rango se hicieron en bloques separados ya que no tenía conocimiento de cómo hacer los cálculos juntos.
Continuando con el reporte, se hizo un conteo de los valores faltantes de todas las columnas así como su porcentaje.

[12] ✓ 0 s

```
#Sacamos el porcentaje de registros nulos para observar la magnitud de cuántos faltan en total
print("Porcentaje de valores nulos de cada columna:")
(df.isnull().sum()/(len(df)))*100
```

```
Porcentaje de valores nulos de cada columna:
```

	0
S.No.	0.000000
Name	0.000000
Location	0.000000
Year	0.000000
Kilometers_Driven	0.000000
Fuel_Type	0.000000
Transmission	0.000000
Owner_Type	0.000000
Mileage	0.027575
Engine	0.634220
Power	0.634220
Seats	0.730732
New_Price	86.129877
Price	17.013650

Figura 3. Porcentaje de valores nulos para cada columna

En el notebook se hizo la observación de que la columna New_Price cuenta con el porcentaje más alto de valores nulos, llego casi a un 90%. Debido a esto y a que cuenta con un formato inconsistente (registros como “8.61 Lakh” o “21 Lakh”), ésta debe ser eliminada posteriormente para no afectar el proceso de análisis.
Y no se encontraron valores duplicados, lo cual inidica que al menos el identificador no tiene ningún error de registro.

Ahora, inicia la parte gráfica de la práctica donde se visualizará de manera precisa y eficiente lo que anteriormente nos decían los datos.

DISTRIBUCIÓN DE DIMENSIONES NUMÉRICAS.

ANÁLISIS GRÁFICO

```
[14]
✓ 4 s
#Graficamos la distribución de cada dimensión numérica
for col in dim_num:
  x= df[col].dropna().values
  if x.size == 0:
    continue
  plt.figure(figsize=(8, 4))
  plt.hist(x, bins=30, edgecolor="green", color="pink")
  plt.title(f"Distribución de {col}")
  plt.xlabel(col)
  plt.ylabel("Frecuencia")
  plt.tight_layout()
  plt.show()
```

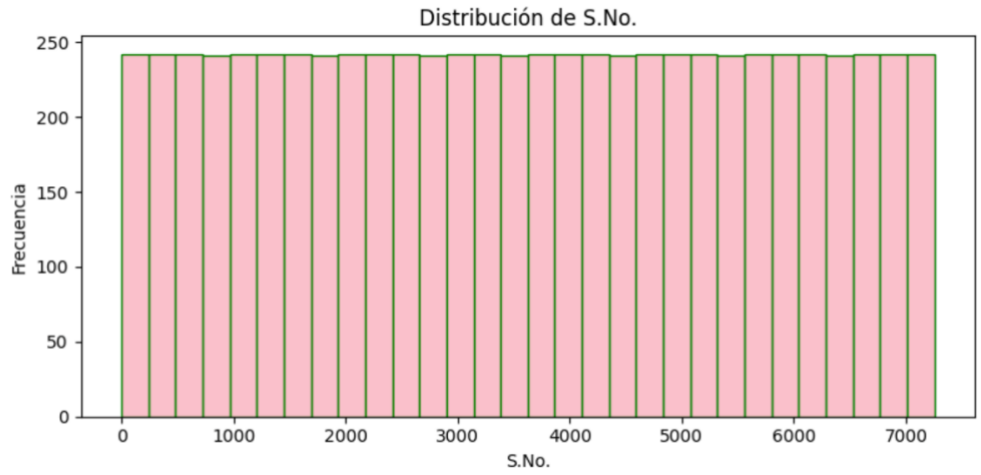


Figura 4. Análisis gráfico y distribución de S.No.

Mencionándolo por última vez, sabiendo que es por fines académicos, mantenemos como dimensión numérica a la columna S.No. para eliminarla posteriormente. Graficamente podemos observar su comportamiento constante.

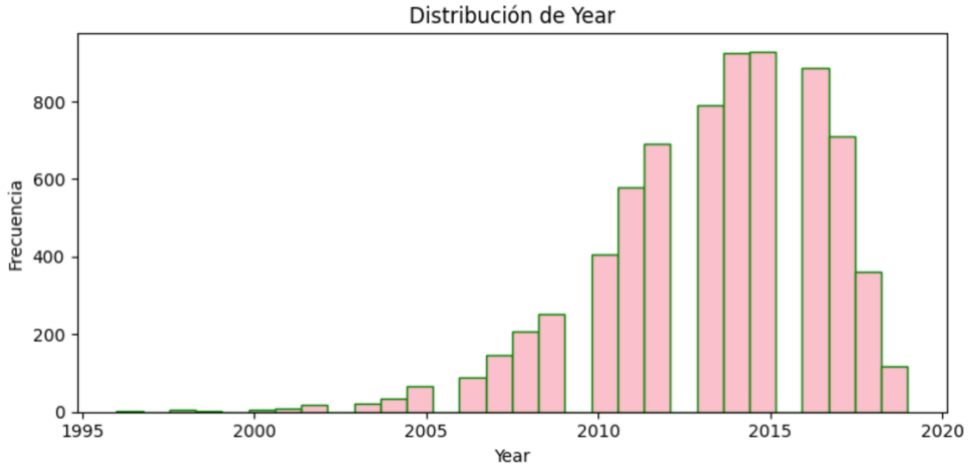


Figura 5. Distribución de Year

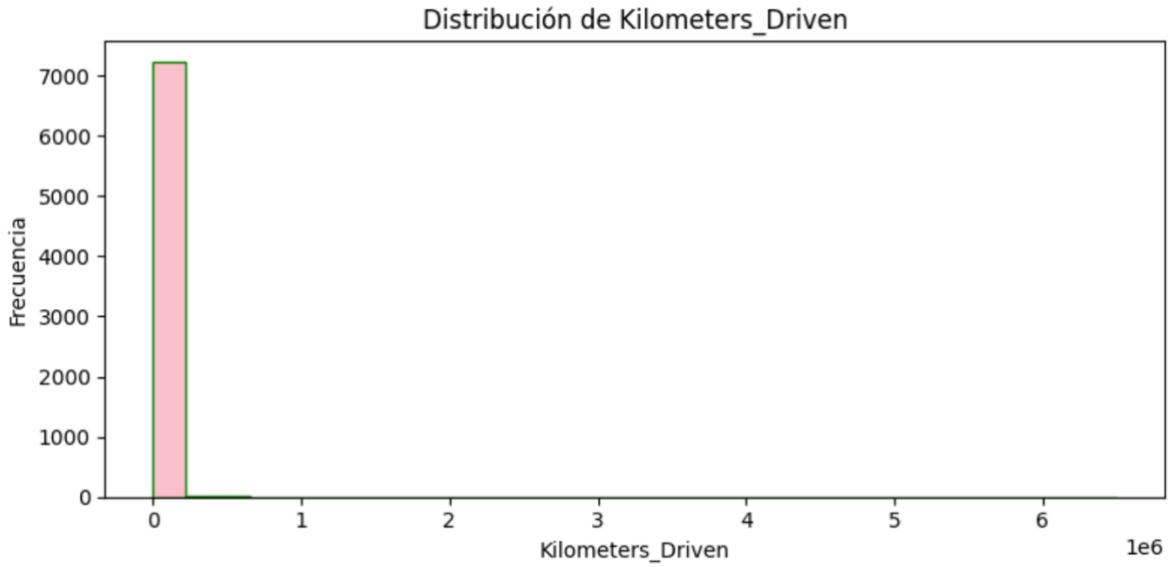


Figura 6. Distribución de Kilometers_Driven

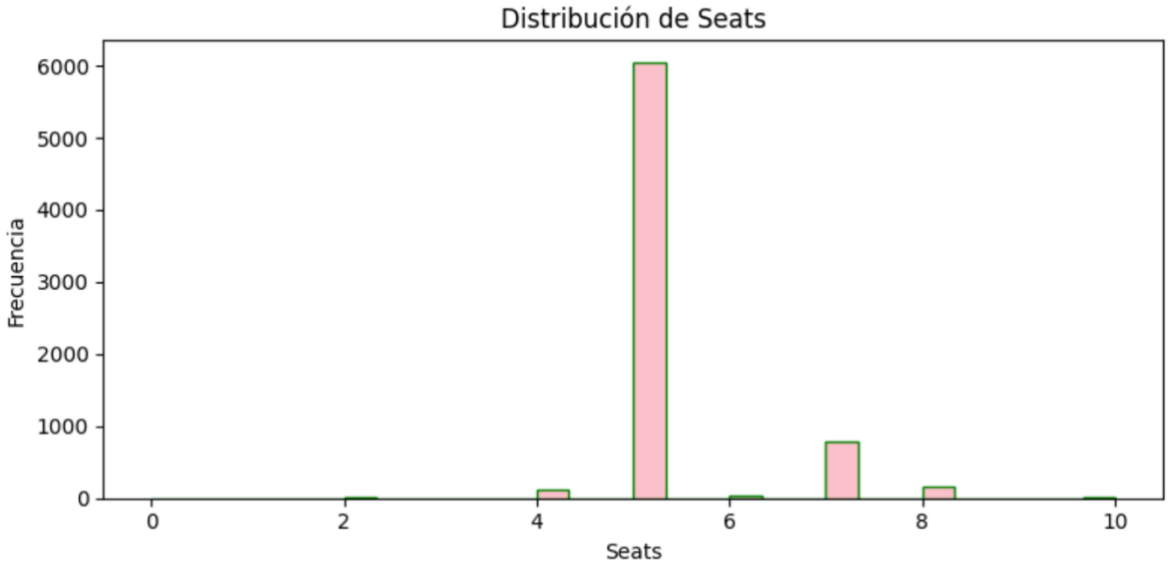


Figura 7. Distribución de Seats

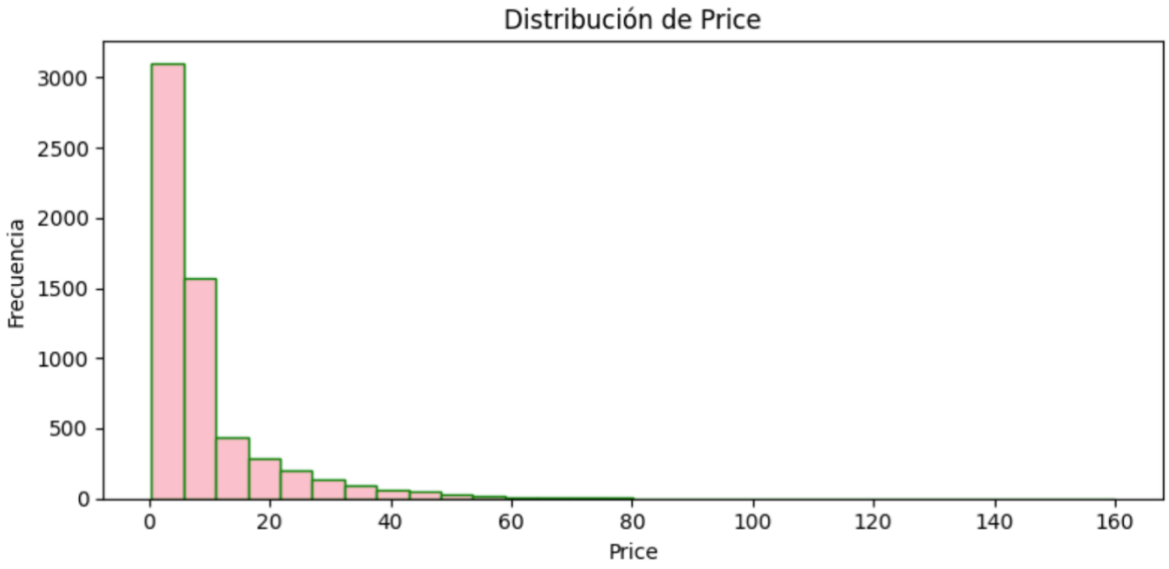


Figura 8. Distribución de Price

Aquí los puntos a destacar son que en Year los autos fueron diseñados cerca del 2012-2015 siendo la media 2013, Kilometers_Driven es un valor constante y en Seats nos dice que el número normal de asientos son 5.
Lo más destacable fue la distribución de Price la cual está fuertemente sesgada a la derecha y esto nos indica varias cosas; la mayor parte de los datos se concentra hacia los valores bajos (autos de bajo o medio costo) y que la cola de la distirbución se extienda hacia la derecha representa a pocos valores pero muy altos (autos más caros que el resto).

Pairplot: Year, Kilometers_Driven, Seats, Price

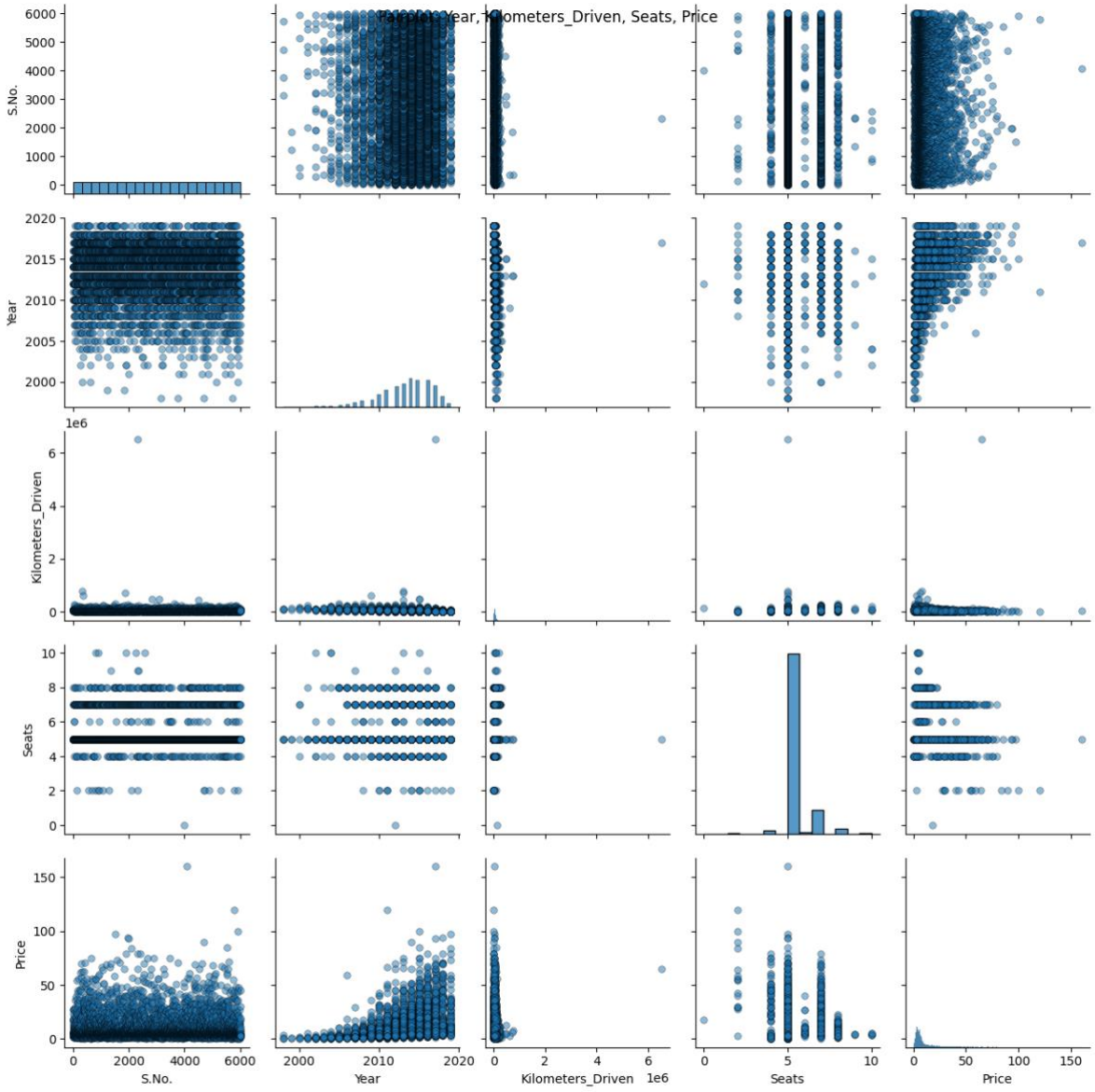


Figura 9. Análisis bivariado

En este pairplot se hizo un análisis bivariado, la relación de una columna con otra, donde se observa; la relación entre Year y Price es claramente positiva, esto quiere decir, que los autos más nuevos tienden a tener precios más altos, en cambio Kilometers_Driven muestra una tendencia inversa con el precio pues los autos más usados suelen valer menos.

El gráfico sugiere que las variables Year y Kilometers_Driven son las que más se relacionan con el precio del vehículo, mientras que Seats y S.No. no tienen una influencia visible.

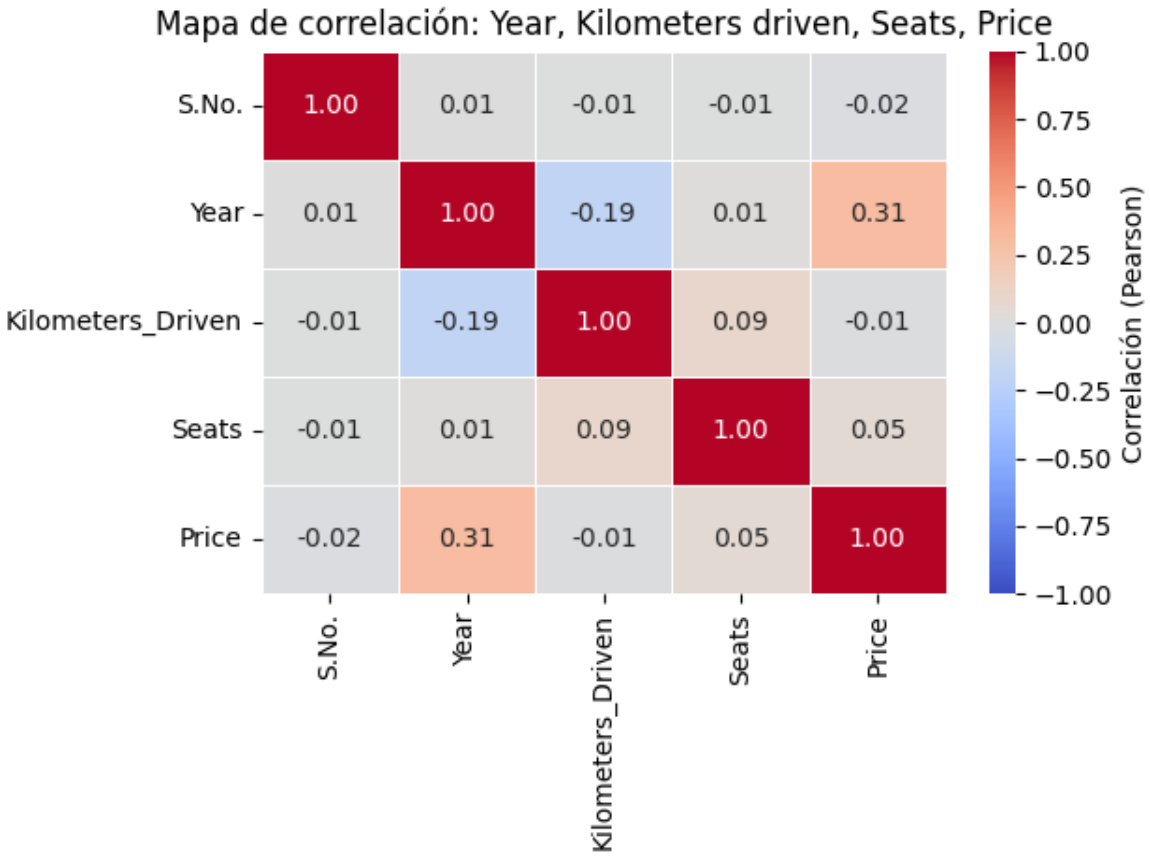


Figura 10. Mapa de calor para correlación

Esta matriz de correlación nos sirve para ver cuánta es la relación entre las variables, la más alta es la ya mencionada Price-Year aunque eso no quiere decir que tengan una correlación fuerte ya que apenas es del 30%. Las demás apenas tienen relación.

DISTRIBUCIÓN DE DIMENSIONES CATEGÓRICAS.

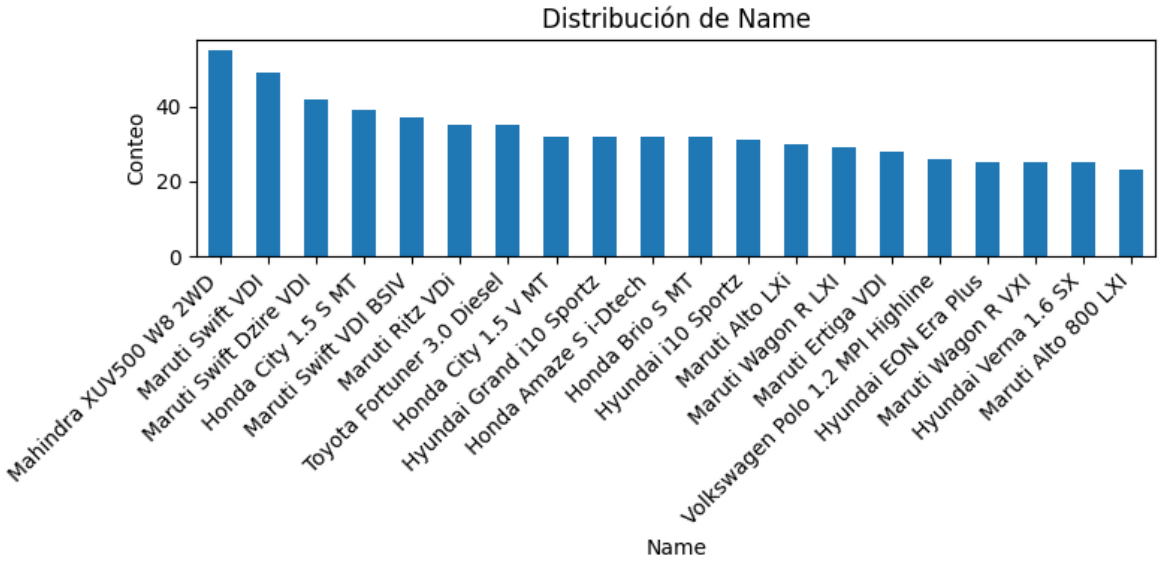


Figura 11. Distribución de Name

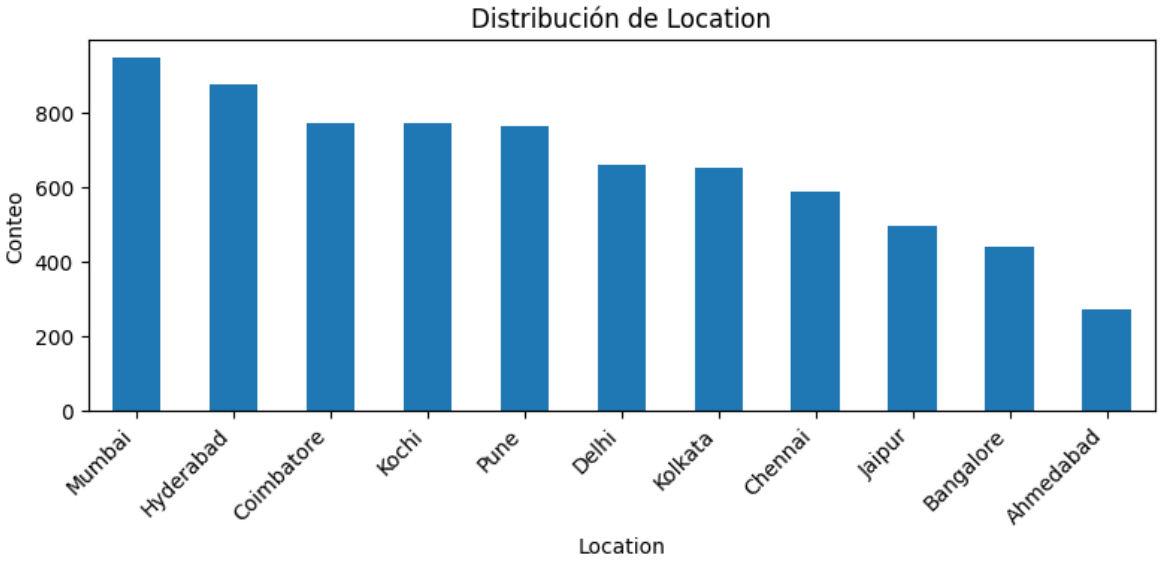


Figura 11. Distribución de Location

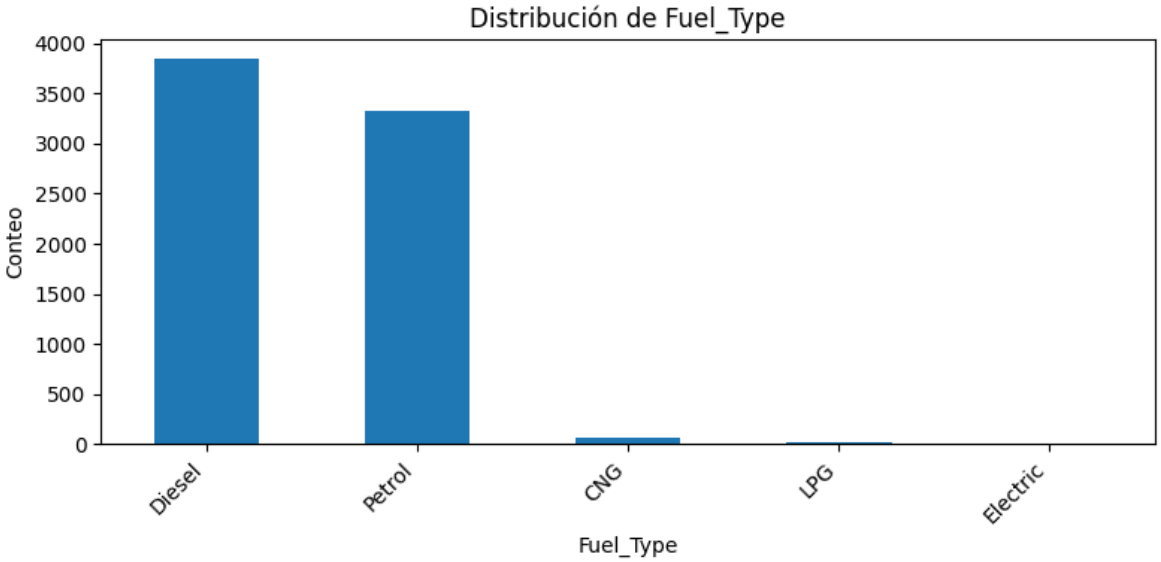


Figura 12. Distribución de Fuel_Type

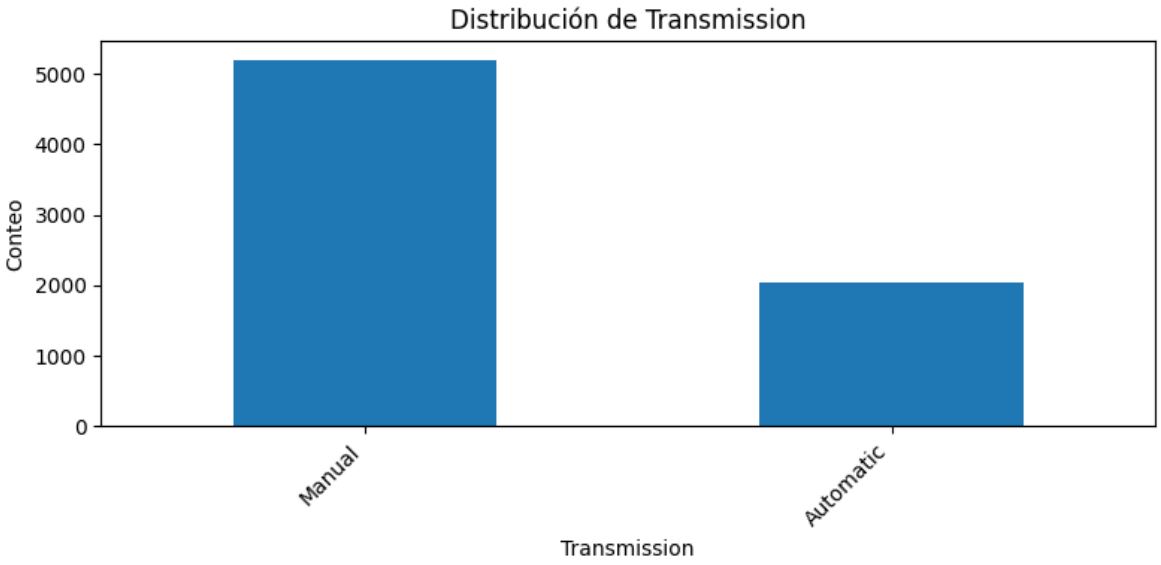


Figura 13. Distribución de Transmission

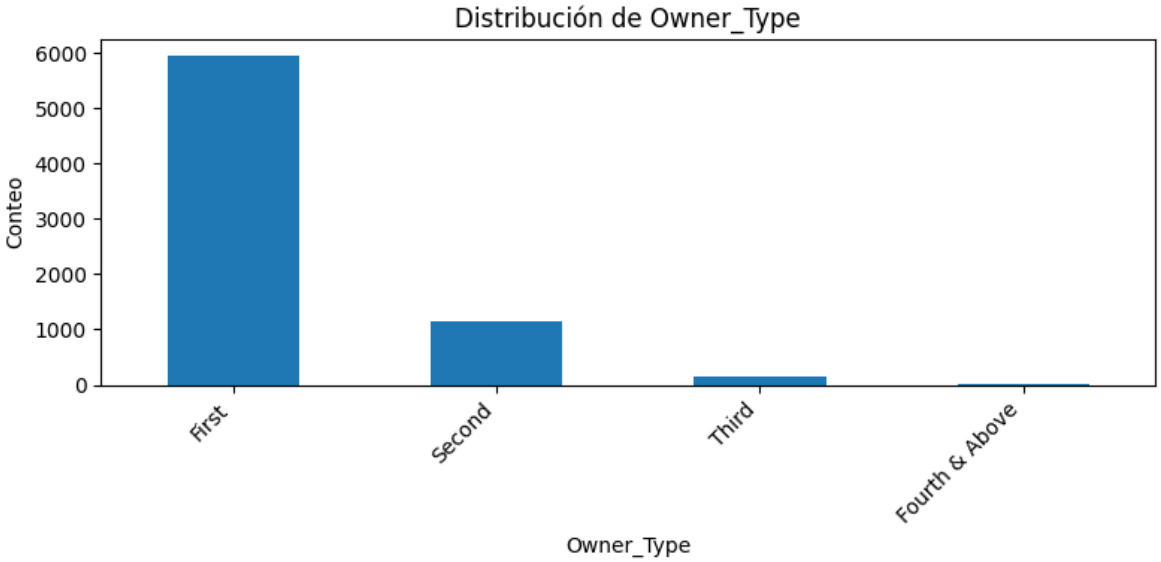


Figura 14. Distribución de Owner_Type

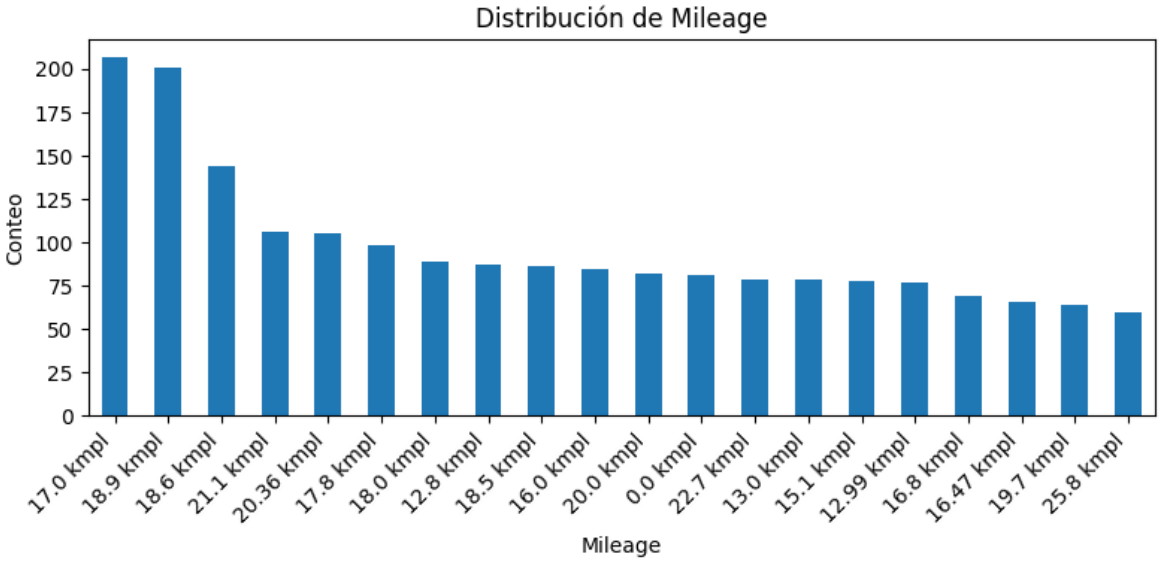


Figura 15. Distribución de Mileage

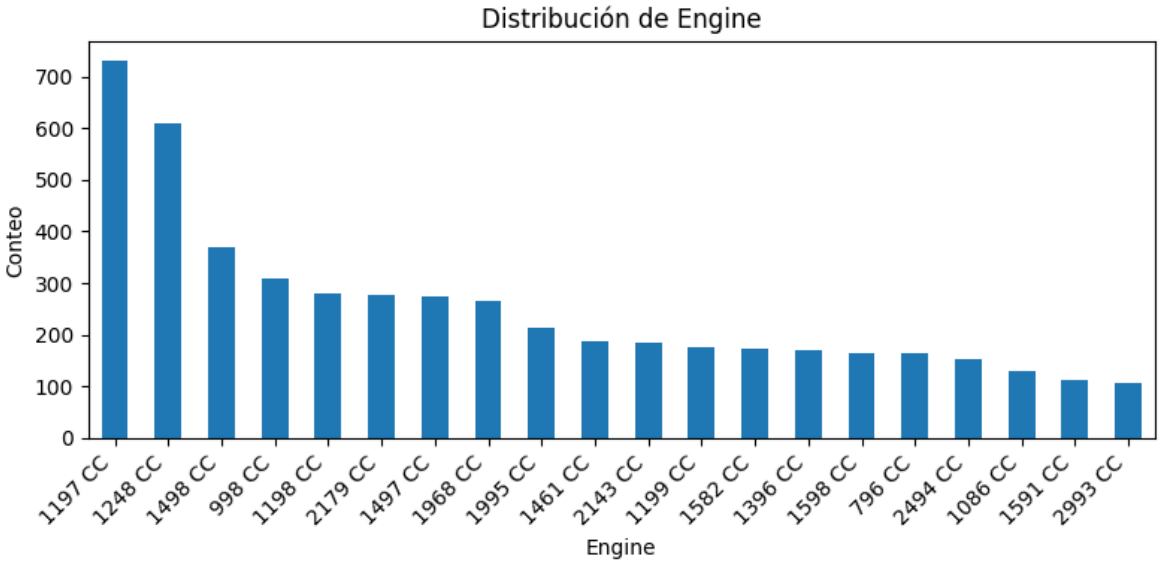


Figura 16. Distribución de Engine

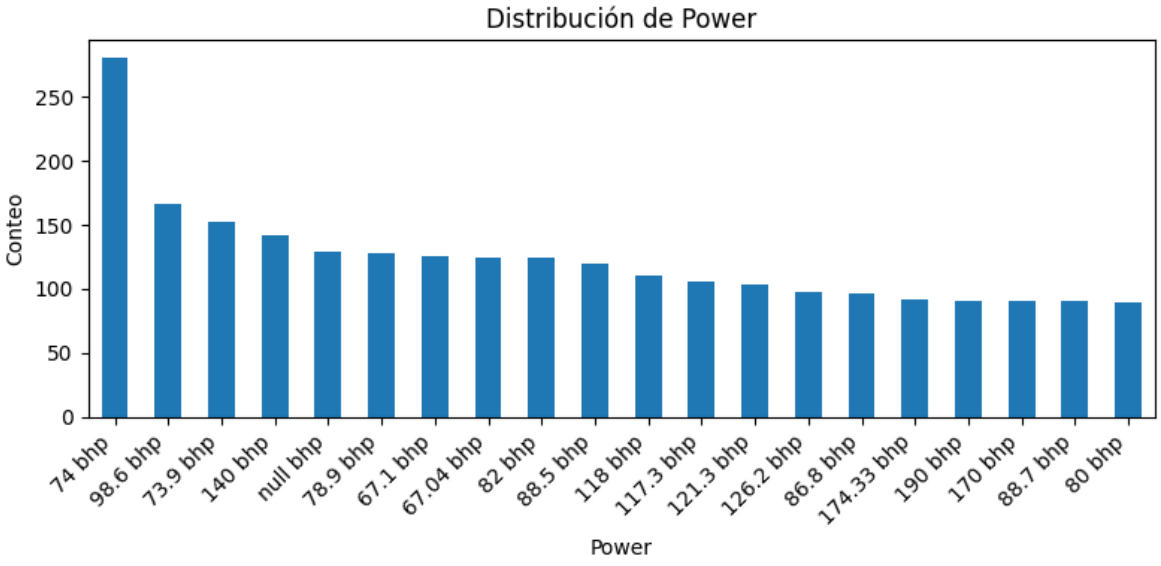


Figura 17. Distribución de Power

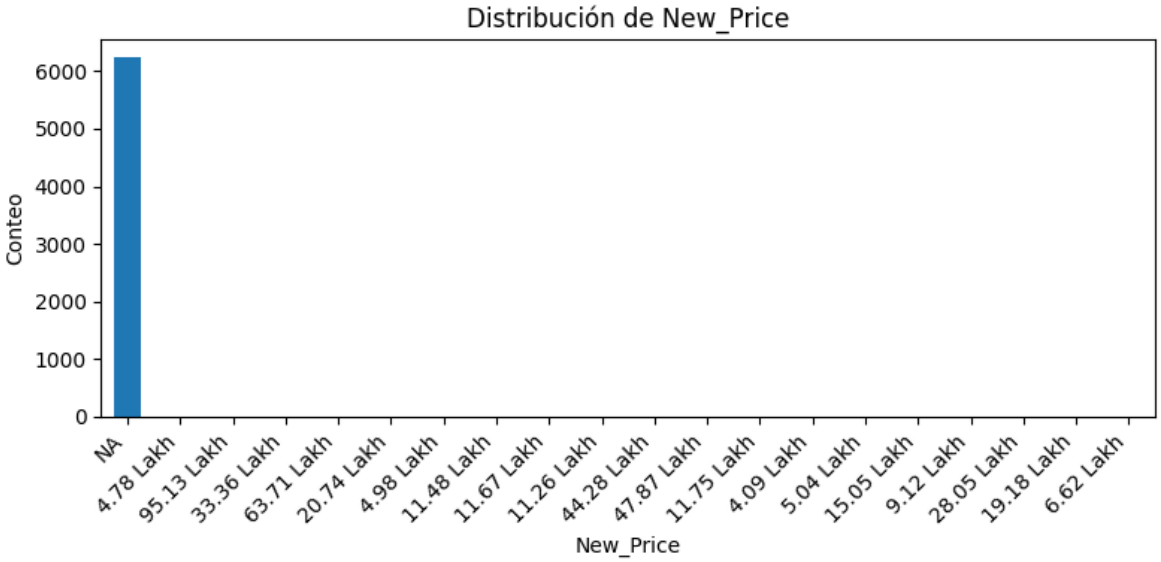


Figura 18. Distribución de New_Price

Las distribuciones de las dimensiones categóricas muestran cómo se distribuyen las principales características categóricas de los autos.

La mayoría de los carros pertenecen a modelos como Mahindra XUV500, Maruti Swift y Honda City,lo que indica su popularidad en el mercado.
En cuanto a la ubicación, ciudades como Mumbai,
Hyderabad y Coimbatore concentran la mayor cantidad de registros.
Respecto al tipo de combustible, predominan los autos Diésel y Gasolina, mientras que los de CNG, LPG o Eléctricosson muy pocos.
La mayoría de los vehículos tienen transmisión manual y son de primer propietario, lo que sugiere que el mercado está enfocado en autos usados de primera mano.

COMPARACIÓN DE CADA UNA DE LAS DIMENSIONES CATEGÓRICAS CONTRA LA DIMENSIÓN "PRECIO".

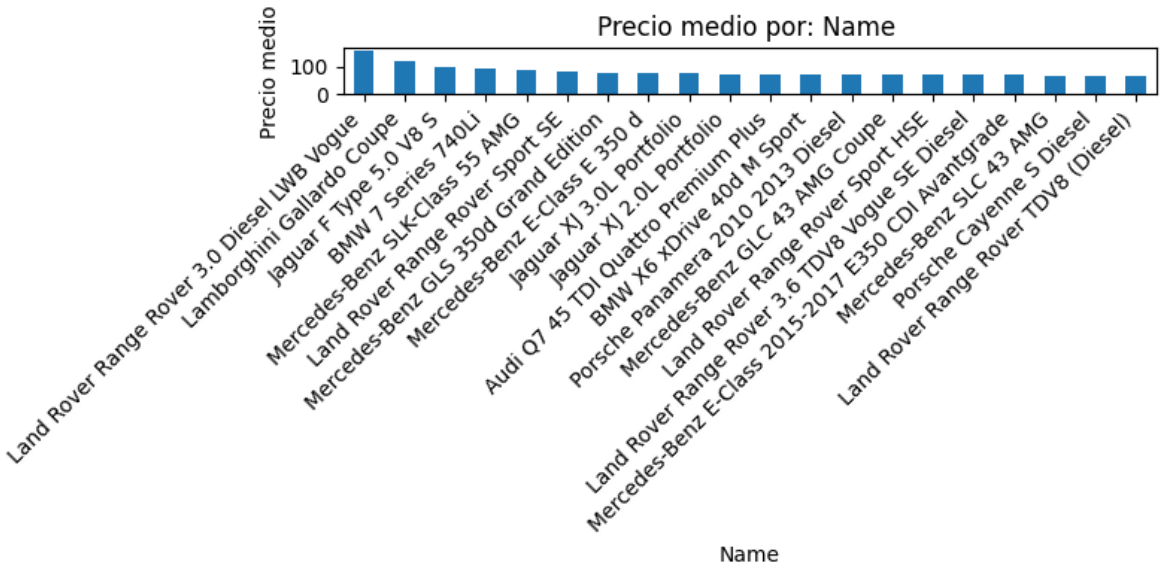


Figura 19. Precio medio por: Name

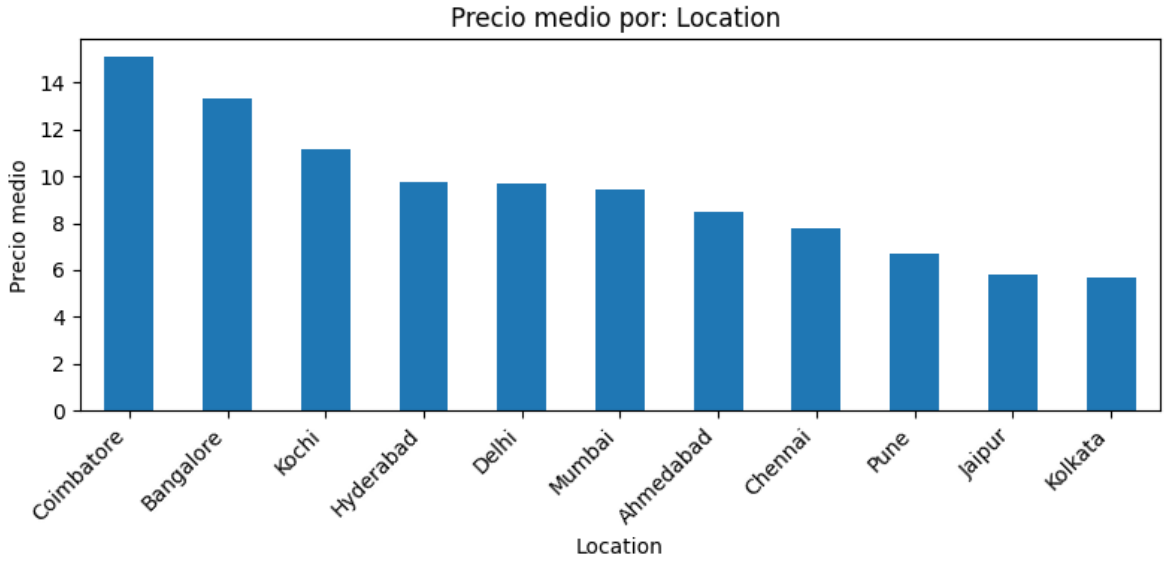


Figura 20. Precio medio por: Location

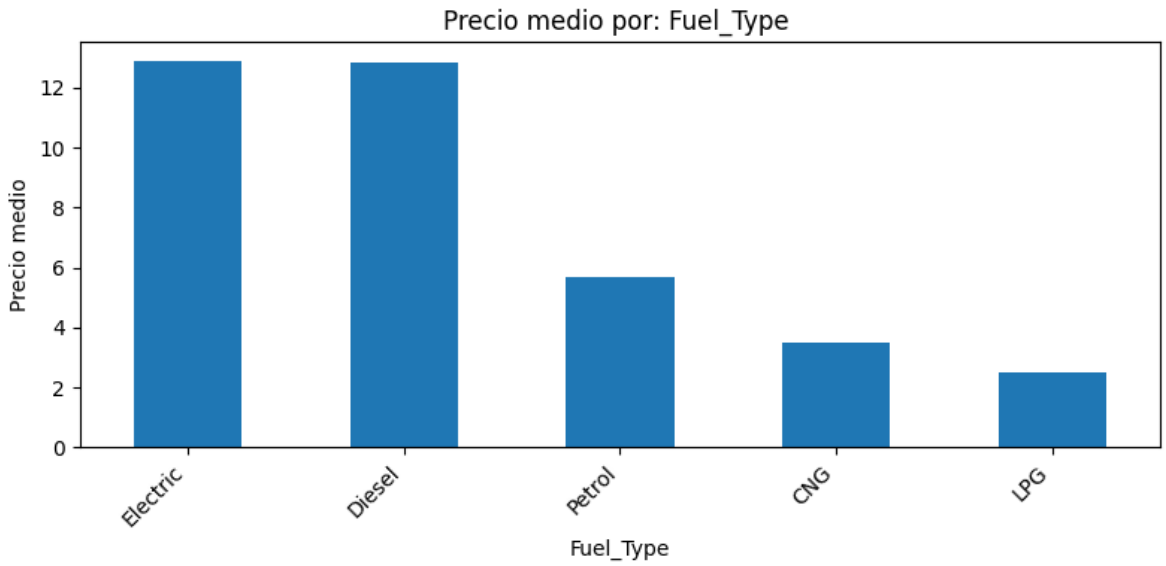


Figura 21. Precio medio por: Fuel_Type

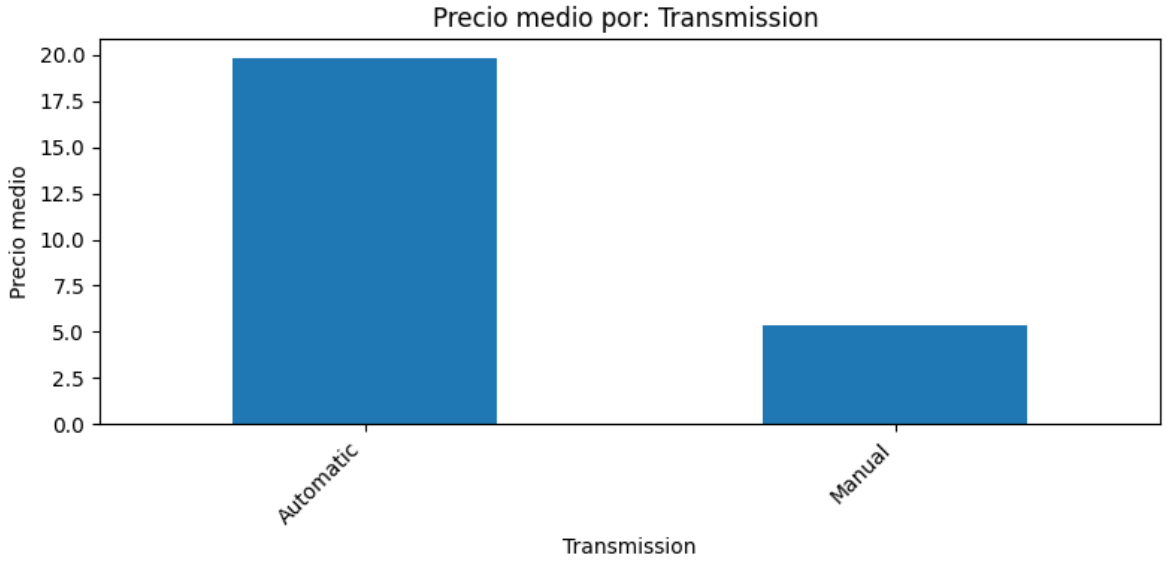


Figura 22. Precio medio por: Transmission

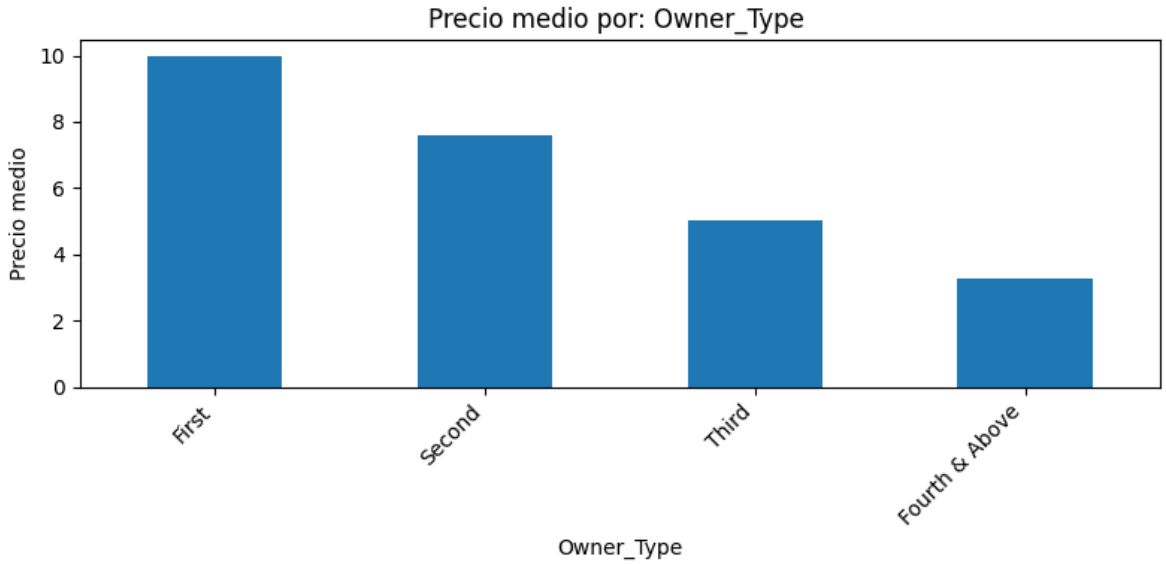


Figura 23. Precio medio por: Owner_Type

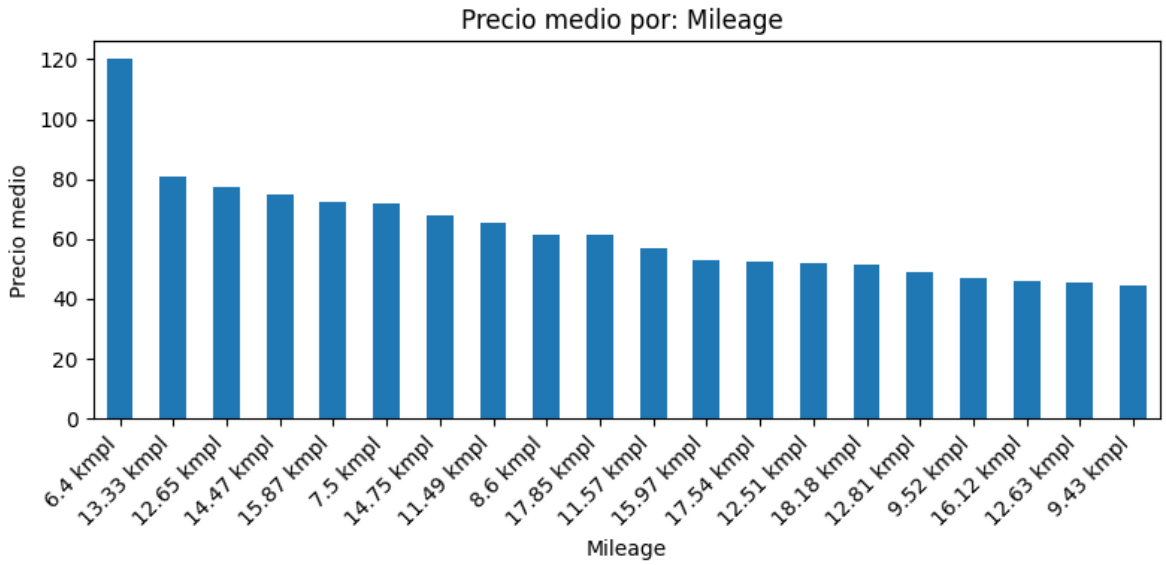


Figura 24. Precio medio por: Mileage

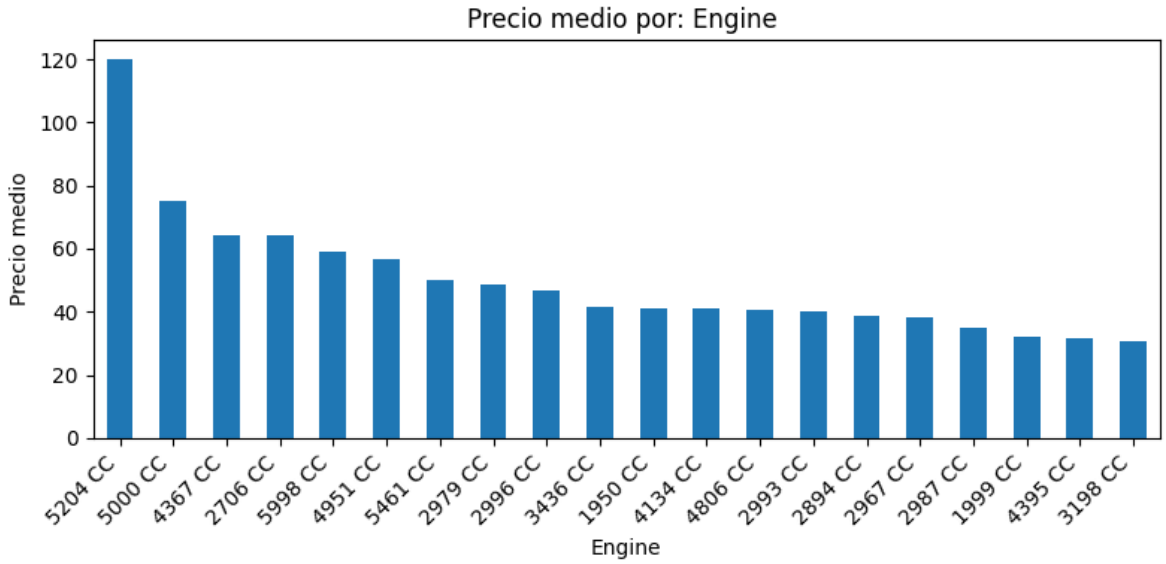


Figura 25. Precio medio por: Engine

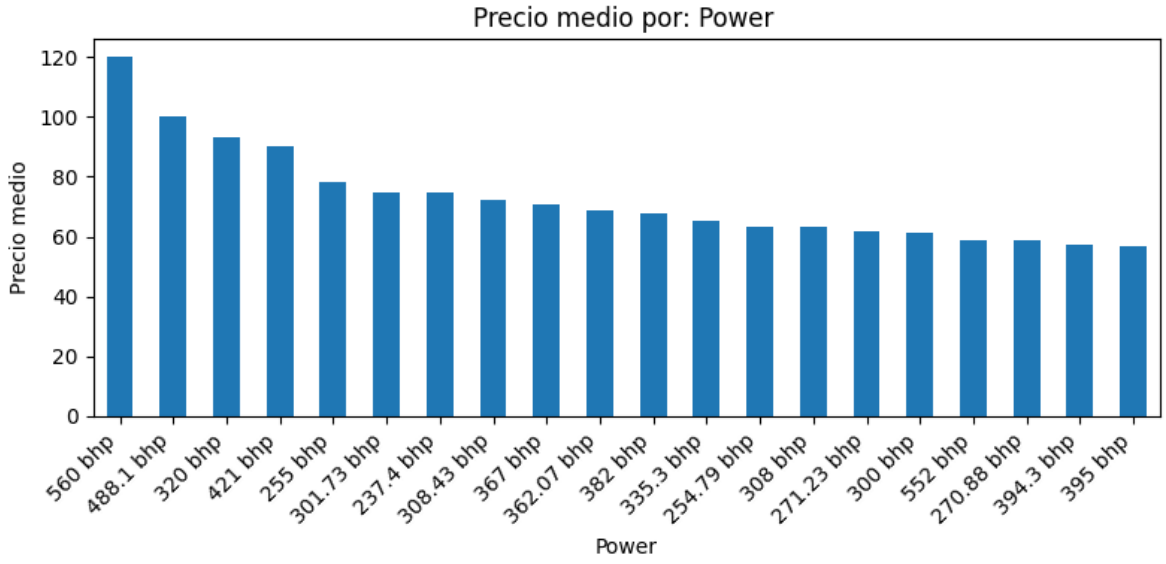


Figura 26. Precio medio por: Power

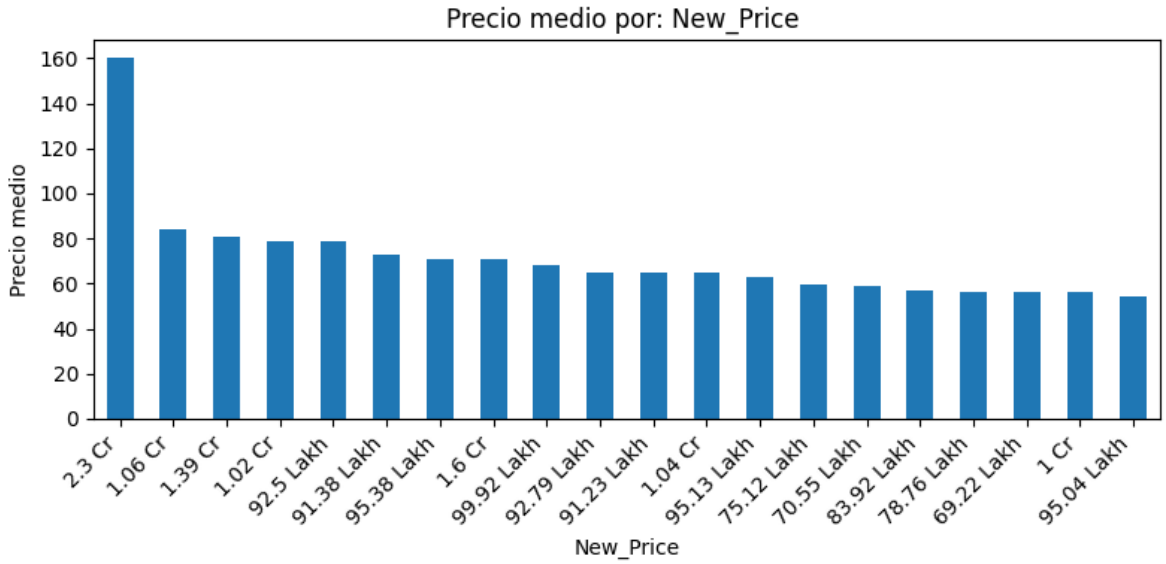


Figura 27. Precio medio por: New_Price

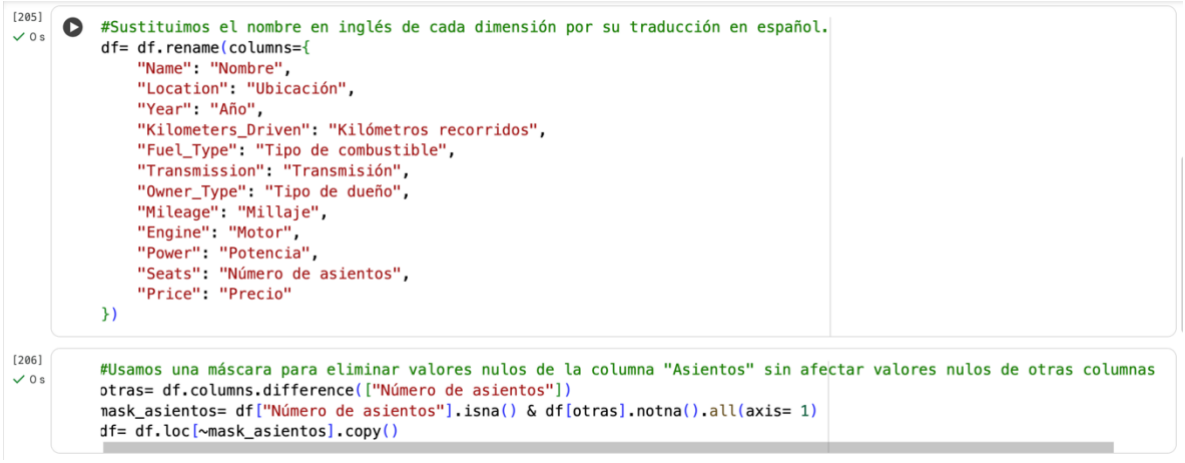
Con esta comparación de las dimensiones categóricas con la dimensión numérica Price, se puede observar que el precio medio de los autos varía bastante según las características categóricas. Por ejemplo:

- Las ciudades como Coimbatore y Bangalore tienen los autos más caros, lo que sugiere que cuentan con un mercado de autos de alta gama.
- Los autos eléctricos y de diésel tienen precios promedio mayores que los de gasolina CNG o LPG, probablemente por su tecnología o disponibilidad.
- Los autos con transmisión automática son significativamente más caros que los manuales.
- Los autos que tuvieron a 1 sólo dueño también mantienen precios altos, disminuyendo conforme al número de dueños.
- Finalmente, al comparar variables técnicas como Mileage, Engine y Power se nota una relación directa, motores más potentes y de mayor cilindrada tienden a tener precios promedio más elevados.

2. *LIMPIEZA DE DATOS.*

En esta sección hicimos la limpieza de datos, siendo factor clave para poder "moldear" nuestro dataset hacia nuestras necesidades (lo que nos funciona o no nos funciona). Con el EDA pudimos identificar cuáles son los puntos débiles de los datos y qué podríamos hacer para mejorarlos.

Primero eliminamos las columas innecesarias como el identificador S.No. y New_Price debido a su alto porcentaje de valores faltantes.
Después cambiamos el nombre de las demás columnas resultantes de inglés a español para mayor comodidad del analista.
Posterior a eso, hicimos uso de una máscara para eliminar los valores nulos de la columna "Asientos" sin afectar los valores nulos de las demás columnas en común, esto nos dejó con 2 registros menos.
Finalmente, sustituimos los valores nulos de la columna Millaje usando el valor medio de ésta misma; así como usamos la moda de la columna "Motor" para realizar imputación de los datos.



```
[205]
✓ 0 s
#Sustituimos el nombre en inglés de cada dimensión por su traducción en español.
df= df.rename(columns={
    "Name": "Nombre",
    "Location": "Ubicación",
    "Year": "Año",
    "Kilometers_Driven": "Kilómetros recorridos",
    "Fuel_Type": "Tipo de combustible",
    "Transmission": "Transmisión",
    "Owner_Type": "Tipo de dueño",
    "Mileage": "Millaje",
    "Engine": "Motor",
    "Power": "Potencia",
    "Seats": "Número de asientos",
    "Price": "Precio"
})
```

```
[206]
✓ 0 s
#Usamos una máscara para eliminar valores nulos de la columna "Asientos" sin afectar valores nulos de otras columnas
otras= df.columns.difference(["Número de asientos"])
mask_asientos= df["Número de asientos"].isna() & df[otras].notna().all(axis= 1)
df= df.loc[~mask_asientos].copy()
```

Figura 28. Parte de la limpieza de datos

3. *TRANSFORMACIÓN DE DATOS.*

La transformación de datos es una sección muy importante (quizá vital) para lo que es el análisis de datos, esto porque en el mundo real son demasiado comunes los errores humanos al momento de crear o hacer registros en un dataset.

Para esta parte de la práctica, primero se reemplazaron valores específicos donde el nombre de la marca/modelo del auto no está escrito de manera correcta.
Después para las dimensiones que deberían ser numéricas desde un principio (Millaje, Motor, Potencia), se hizo una limpieza de columnas eliminando cualquier carácter que no fuera numérico así como guiones, puntos, etc. Caso que es muy común en la vida cotidiana, ya que cada persona registra la información a su manera.
Para estas mismas columnas se les aplicó un redondeo de valores al entero más cercano con el objetivo de tener datos más consistentes y uniformes.
Posteriormente, se le realizó una multiplicación por cien a los valores de la columna Precio, esto porque el formato estaba incorrecto (ej. "1.90").
Finalmente, a las columnas Precio y Kilómetros recorridos se les hizo un cáculo de logaritmo natural con el propósito de usarlo para un análisis próximo.

[211] ✓ 0 s
```
#Columnas a redondear
cols_a_redondear = ["Millaje", "Motor", "Potencia"]

#Nos aseguramos de que son numéricas
df[cols_a_redondear] = df[cols_a_redondear].apply(pd.to_numeric, errors="coerce")

#Redondear al entero más cercano
df[cols_a_redondear] = df[cols_a_redondear].round(0).astype("Int64")  # usa tipo entero que acepta NaN
```

[212] ✓ 0 s
```
#Asegurarse de que "Precio" sea numérico
df["Precio"] = pd.to_numeric(df["Precio"], errors="coerce")

#Multiplicar por 1000
df["Precio"] = df["Precio"] * 1000
```

[213] ✓ 0 s
```
# Asegurarse de que sean numéricas
df["Precio"] = pd.to_numeric(df["Precio"], errors="coerce")
df["Kilómetros recorridos"] = pd.to_numeric(df["Kilómetros recorridos"], errors="coerce")

# Calcular el logaritmo natural (agregando pequeñas constantes para evitar log(0))
df["log_Precio"] = np.log(df["Precio"] + 1)
df["log_Kilómetros"] = np.log(df["Kilómetros recorridos"] + 1)
```

Figura 29. Parte de la transformación de datos

4. *INGENIERÍA DE CARACTERÍSTICAS.*

Personalmente, no había tenido oportunidad de haber trabajado una sección como lo es la ingeniería de características donde convertimos la información cruda del dataset en variables más útiles y consistentes, gracias a este proceso podemos obtener un conjunto de datos más estructurado e interpretativo para nuestro análisis, útil tanto como para la máquina como para el humano.

En este caso, primero calculamos la antigüedad de fabricación que tienen los carros usando la función datetime.now() quien hace referencia al año actual.
Y finalmente, se hizo la creación de columnas importantes como lo son Marca y Modelo a partir de la columna Nombre. Considero que esta es una partición importante ya que recuerdo por mis clases que entre más "chiquito" sea el dato es mejor, ahora podemos saber exactamente cuál es la marca del auto.

[215]
✓ 0 s

```python
# Crear columnas "Marca" y "Modelo" a partir de "Nombre"

# Asegurarse de que la columna "Nombre" sea de tipo texto
df["Nombre"] = df["Nombre"].astype(str)

# "Marca" = primera palabra de "Nombre"
df["Marca"] = df["Nombre"].str.split().str[0]

# "Modelo" = concatenación del resto de las palabras sin espacios
df["Modelo"] = df["Nombre"].str.split().apply(lambda x: "".join(x[1:]) if len(x) > 1 else "")

# Mostrar ejemplo
df[["Nombre", "Marca", "Modelo"]].head()
```

	Nombre	Marca	Modelo
0	Maruti Wagon R LXI CNG	Maruti	WagonRLXICNG
1	Hyundai Creta 1.6 CRDi SX Option	Hyundai	Creta1.6CRDiSXOption
2	Honda Jazz V	Honda	JazzV
3	Maruti Ertiga VDI	Maruti	ErtigaVDI
4	Audi A4 New 2.0 TDI Multitronic	Audi	A4New2.0TDIMultitronic

Figura 30. Parte de la ingeniería de características

5. *EDA POSTERIOR.*

En esta sección final se hará nuevamente un anaálisis exploratorio de los datos para poder observar el trabajo que se hizo en las secciones anteriores, con el fin de obtener un análisis sobre cómo es que recibimos los datos y cómo los vamos a entregar por así decirlo. Es la importancia que tienen los procesos ETL, ya que sin esto, nuestro trabajo como científicos de datos sería un dolor de cabeza y no podríamos hacer análisis o modelado de una forma eficiente con datos "sucios".

Ahora en este análisis numérico, lo que podemos destacar es que la columna Precio sigue teniendo una desviación alta reflejando así una gran variabilidad entre vehículos económicos y de gama alta. Mientras que las variables transformadas mediante logaritmo natural (log_Precio y log_Kilómetros) muestran una distribución más equilibrada, lo que facilita análisis posteriores que tengan que ver con estas dimensiones. Por último, la antigüedad promedio es de aproximadamente 11 años, siendo coherente con la media del año de fabricación anteriormente observada.

ANÁLISIS NUMÉRICO

[31] ✓ 0 s
```
#Nuevo tamaño de dimension de nuestro dataset
print("Número de filas,columnas:",df.shape)
```
```
Número de filas,columnas: (7247, 17)
```

[32] ✓ 0 s
```
#Descripcion estadistica del dataset
dim_num= df.select_dtypes(include= np.number).columns.tolist()
df[dim_num].describe()
```

	Año	Kilómetros recorridos	Millaje	Motor	Potencia	Número de asientos	Precio	log_Precio	log_Kilómetros	Antigüedad
count	7247.000000	7.247000e+03	7247.0	7247.0	7076.0	7200.000000	6013.000000	6013.000000	7247.000000	7247.000000
mean	2013.368428	5.868771e+04	18.154823	1613.989237	112.771764	5.279722	9485.925495	8.733961	10.760718	11.631572
std	3.252725	8.445748e+04	4.541985	594.256483	53.489695	0.811660	11191.465376	0.873711	0.716416	3.252725
min	1996.000000	1.710000e+02	0.0	72.0	34.0	0.000000	440.000000	6.089045	5.147494	6.000000
25%	2011.000000	3.400000e+04	15.0	1197.0	75.0	5.000000	3500.000000	8.160804	10.434145	9.000000
50%	2014.000000	5.339200e+04	18.0	1462.0	94.0	5.000000	5640.000000	8.637817	10.885435	11.000000
75%	2016.000000	7.300000e+04	21.0	1968.0	138.0	5.000000	9960.000000	9.206433	11.198228	14.000000
max	2019.000000	6.500000e+06	34.0	5998.0	616.0	10.000000	160000.000000	11.982935	15.687313	29.000000

Figura 31. Análisis numérico posterior, descripción estadística

Nuevamente queda aclarar que el cálculo de moda y rango se hicieron por separado de esta tabla, ya que no supe como juntarlas.

Para el porcentaje de valores nulos, podemos encontrar que ya no hay cantidades tan elevadas como lo fue con New_Price. Claramente sigue habiendo en las demás columnas pero estas cifras ya no son tan significativas.

[36]
✓ 0 s

```
print("Porcentaje de valores nulos de cada columna:")
(df.isnull().sum()/(len(df)))*100
```

```
Porcentaje de valores nulos de cada columna:
```

	0
Nombre	0.000000
Ubicación	0.000000
Año	0.000000
Kilómetros recorridos	0.000000
Tipo de combustible	0.000000
Transmisión	0.000000
Tipo de dueño	0.000000
Millaje	0.000000
Motor	0.000000
Potencia	2.359597
Número de asientos	0.648544
Precio	17.027736
log_Precio	17.027736
log_Kilómetros	0.000000
Antigüedad	0.000000
Marca	0.000000
Modelo	0.000000

Figura 32. Análisis numérico posterior, porcentaje de valores nulos

En este caso sólo se halló un valor duplicado.

ANÁLISIS GRÁFICO POSTERIOR.

En esta ocasión, sólo agregaré las distribuciones nuevas para una mejor visualización y después se hará la interpretación.

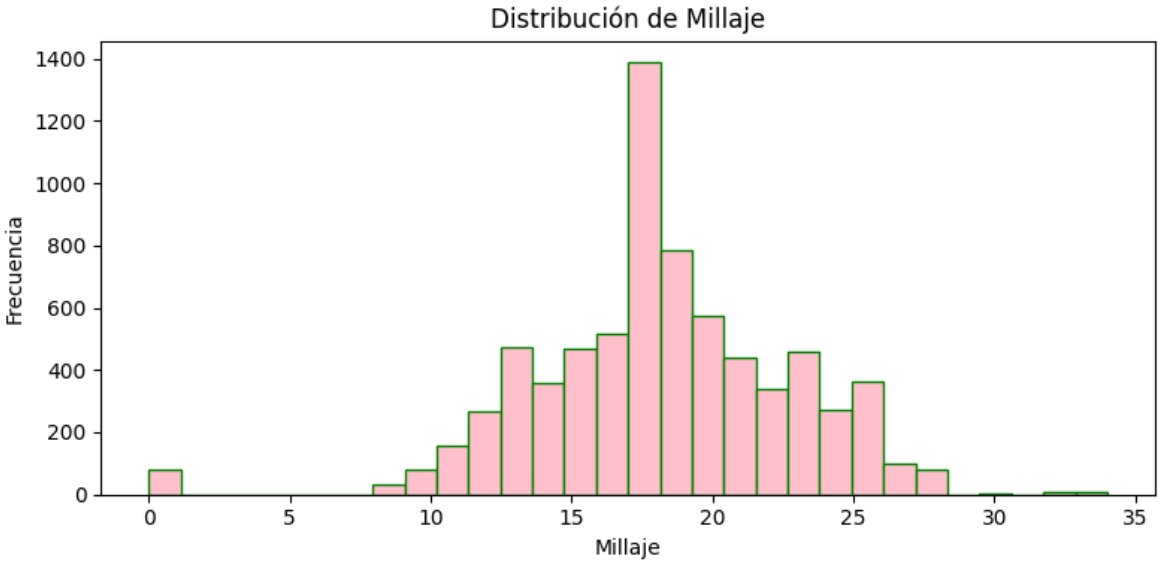


Figura 33. Distribución de Millaje

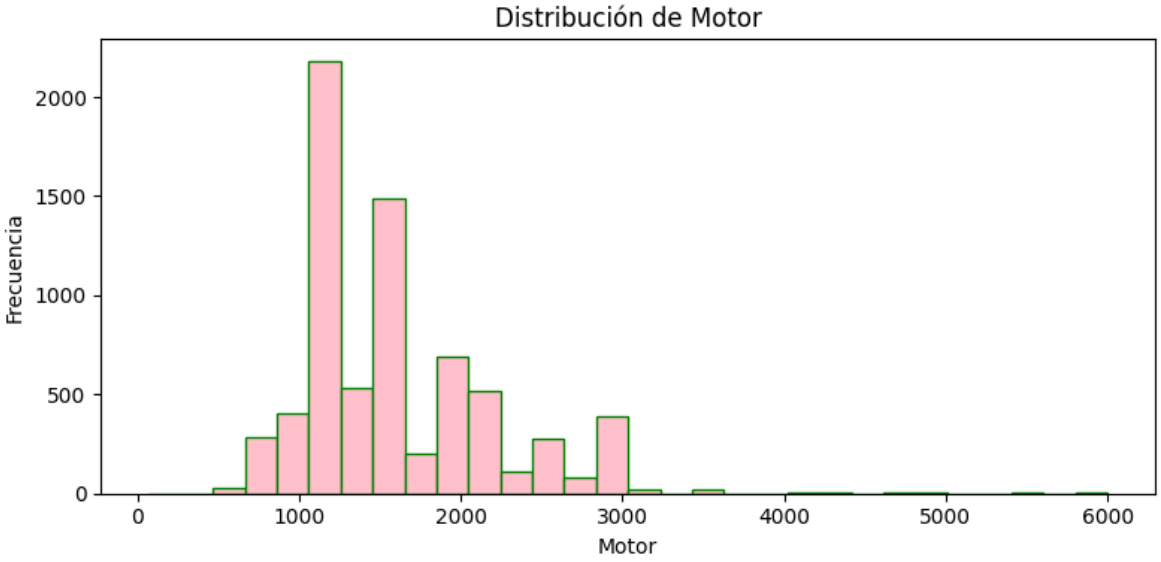


Figura 34. Distribución de Motor

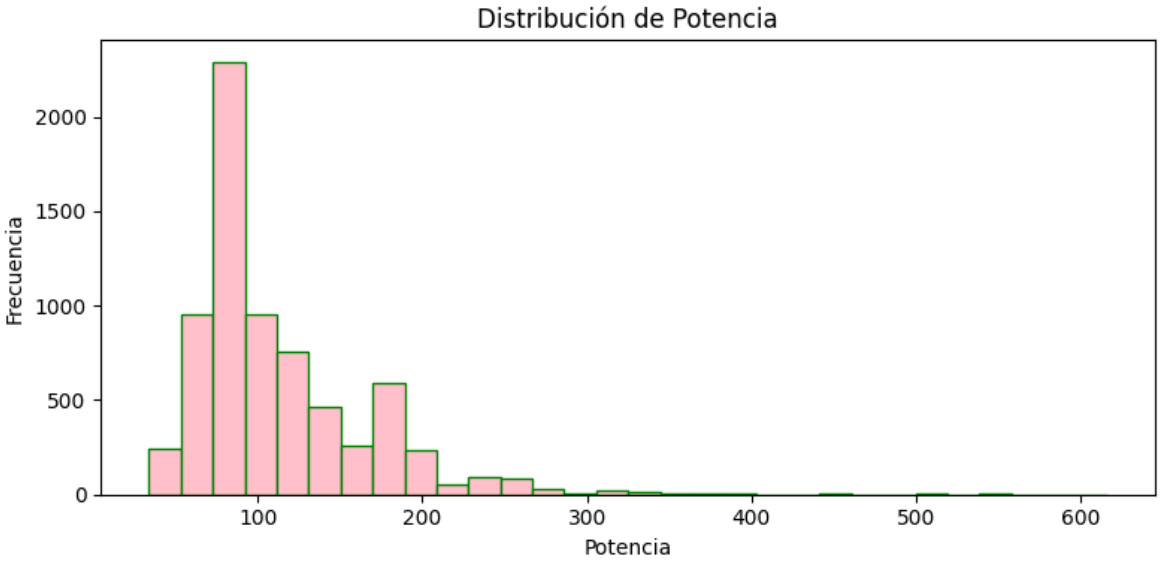


Figura 35. Distribución de Potencia

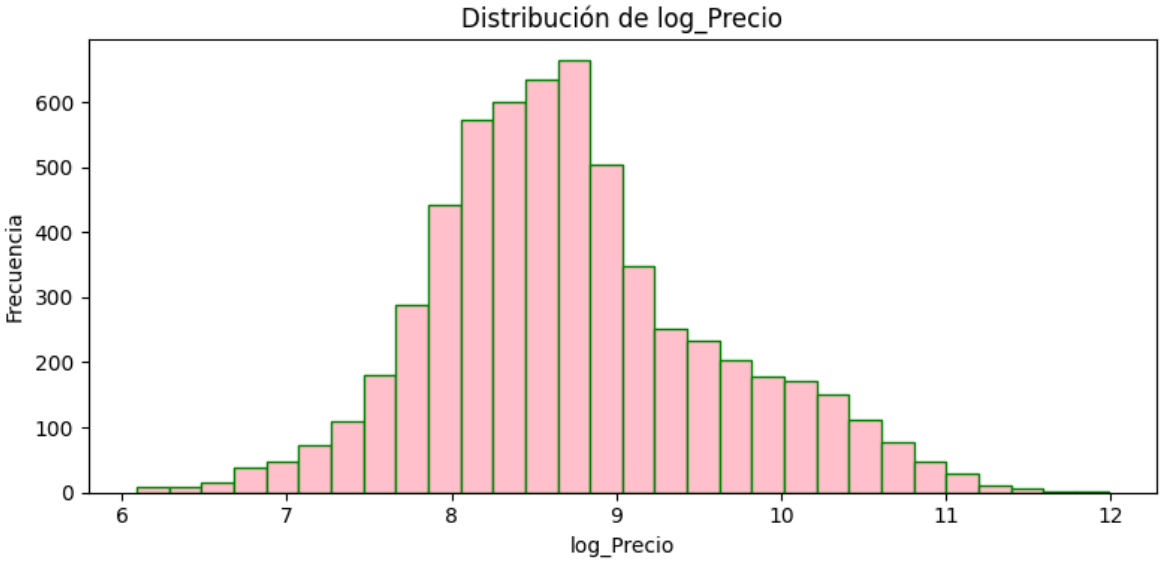


Figura 36. Distribución de log_Precio

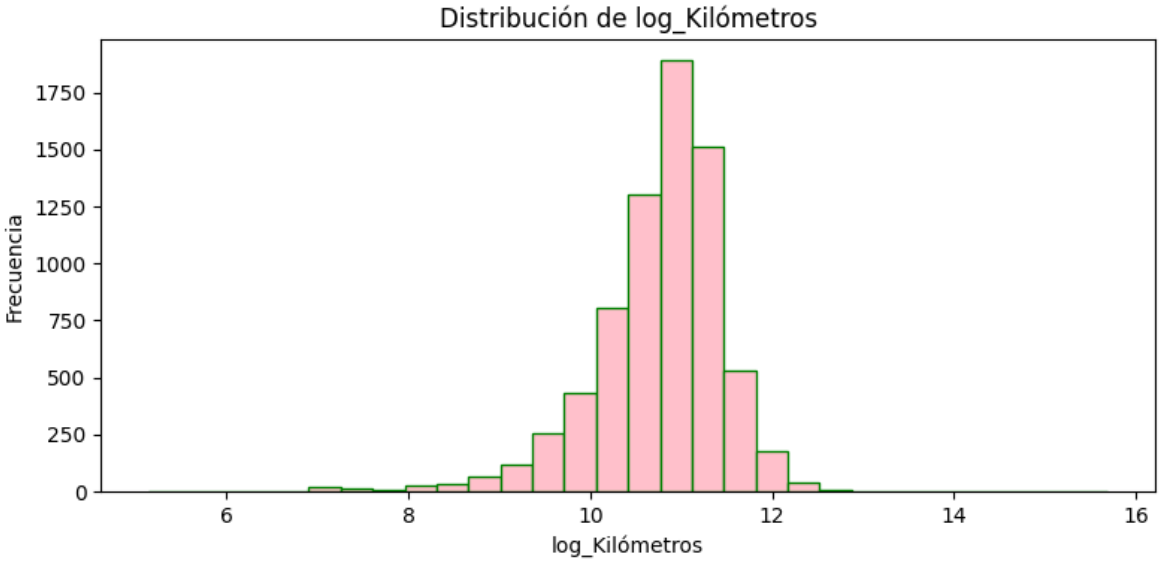


Figura 37. Distribución de log_Kilometros

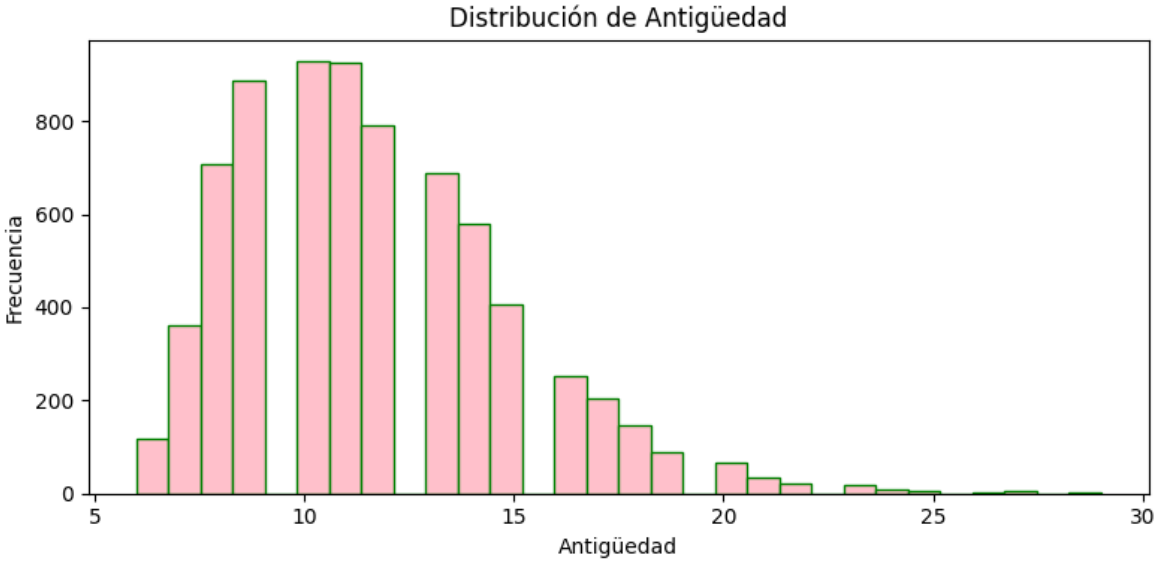


Figura 38. Distribución de log_Kilometros

Ésta sección gráfica es realmente interesante ya que se puede ver mejor el trabajo hecho anteriormente.
Empezando por las nuevas dimensiones numéricas (antes categóricas): Millaje, Motor y Potencia; si observamos las distribuciones que calculamos anteriormente y las comparamos con las nuevas distribuciones, se puede ver claramente que si cambia considerablemente.
La distribución Millaje presenta una forma aproximadamente normal, con una conccentración de valores entre 15 y 20km/L. Existen pocos outliers posiblemente debido a errores de registro o a modelos atípicos de alto rendimiento.
Las distribuciones de Motor y Potencia presentan un sesgo hacia la derecha, en ambos casos podría referir a la presencia de autos caros.

Pairplot: Análisis bivariado de dimensiones numéricas

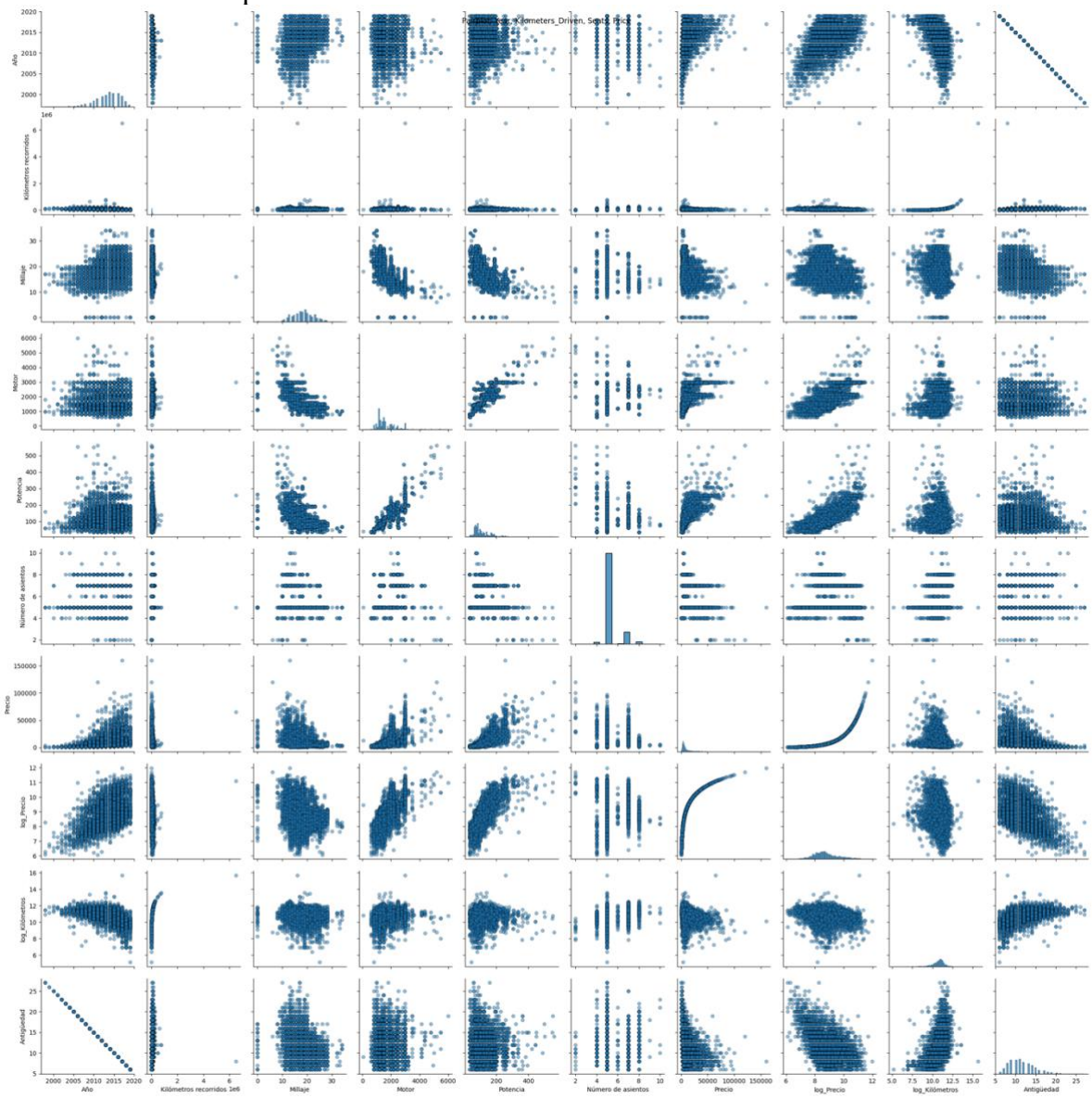


Figura 39. Pairplot: análisis bivariado de dimensiones numéricas

Además de las relaciones anteriormente mencionadas como Año-Precio (positiva) o Precio-Kilometros recorridos (negativa), podemos observar que:

> Las variables Motor y Potencia muestran una relación lineal fuerte, ya que los motores de mayor cilindrada suelen generar más caballos de fuerza. Además, ambas presentan una correlación directa con el Precio, lo que sugiere que el desempeño técnico del vehículo influye de manera importante en su valor.

El gráfico confirma que las variables más determinantes en la variación del precio son Año, Kilómetros recorridos, Motor y Potencia, mientras que el resto aporta información secundaria.

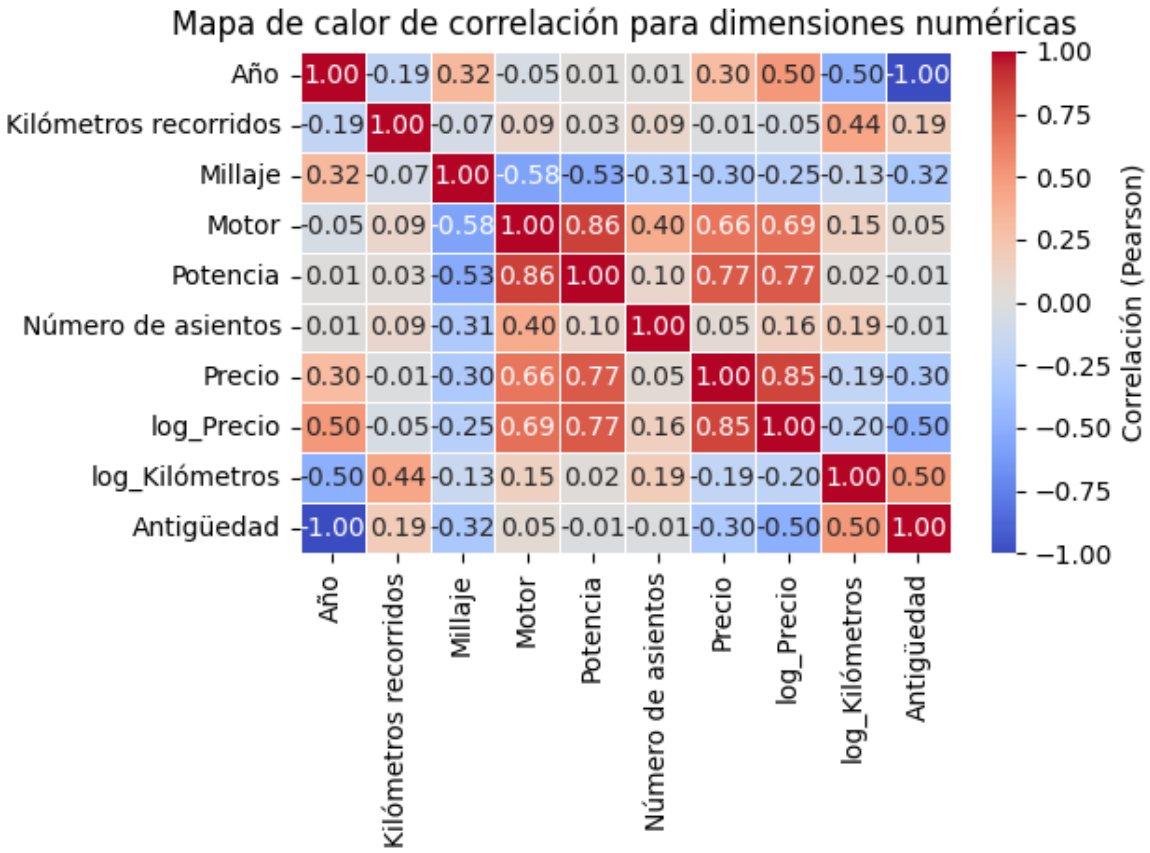


Figura 40. Mapa de calor de correlación para las dimensiones numéricas

En esta nueva matriz se puede observar mejores correlaciones y más coherentes donde las más fuertes son: Potencia-Motor, Precio-Potencia, Precio-Motor.

DISTRIBUCIÓN DE DIMENSIONES CATEGÓRICAS.

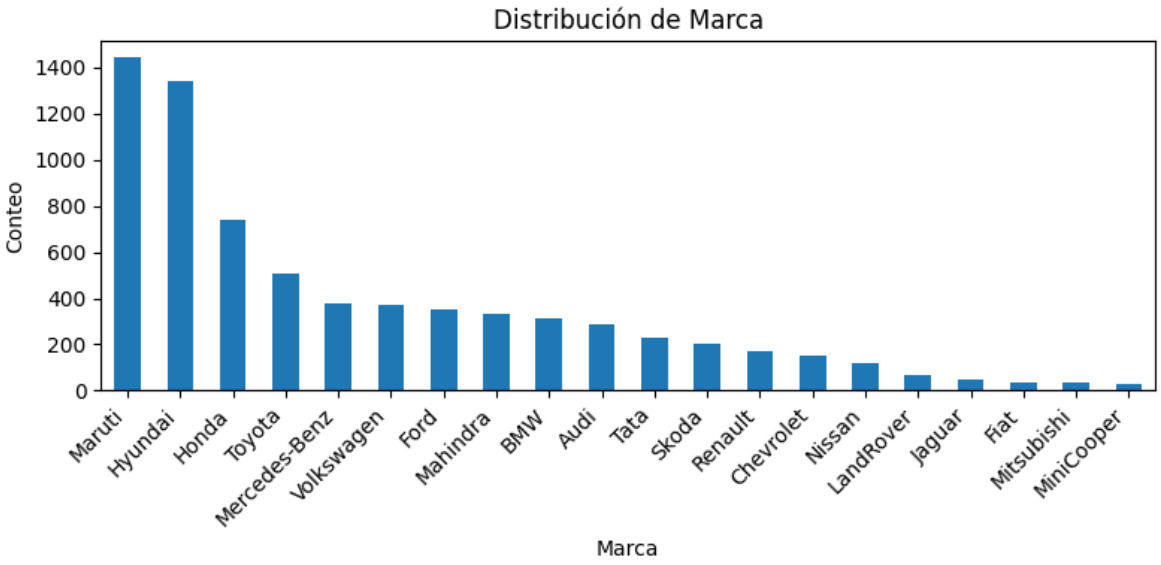


Figura 41. Distribución de Marca

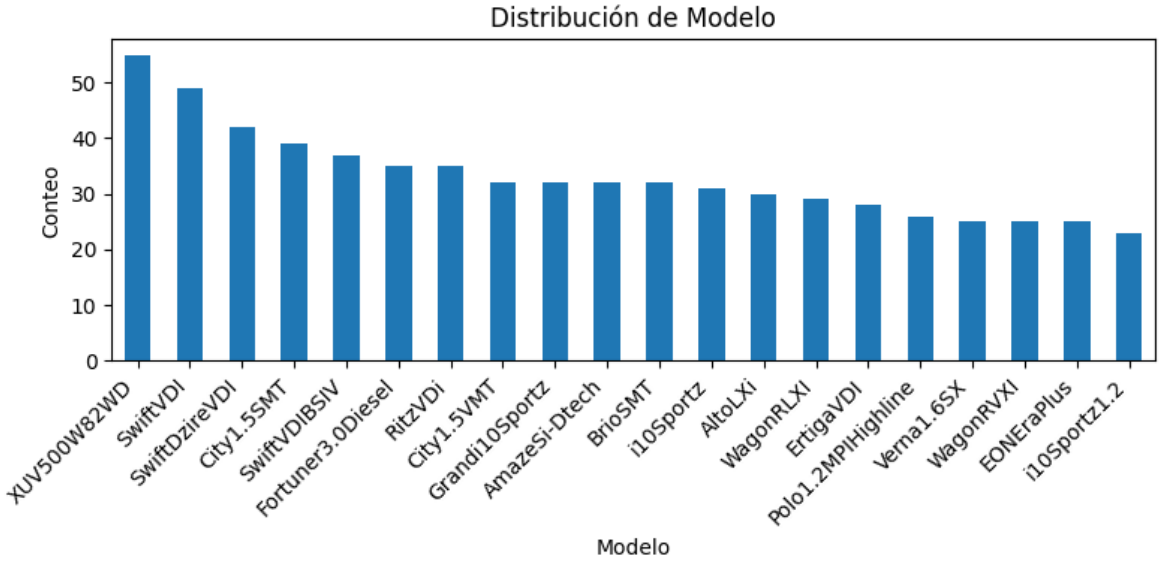


Figura 42. Distribución de Modelo

Para estas distribuciones sólo agregué las que se generaron para evitar ruido visual con las demás gráficas.
En ambas tristribuciones podemos ver un sesgo a la derecha, más marcado en Marca, lo que nuevamente puede ser referido a la presencia de autos de lujo o deportivos.

COMPARACIÓN DE CADA UNA DE LAS DIMENSIONES CATEGÓRICAS CONTRA LA DIMENSIÓN "PRECIO".

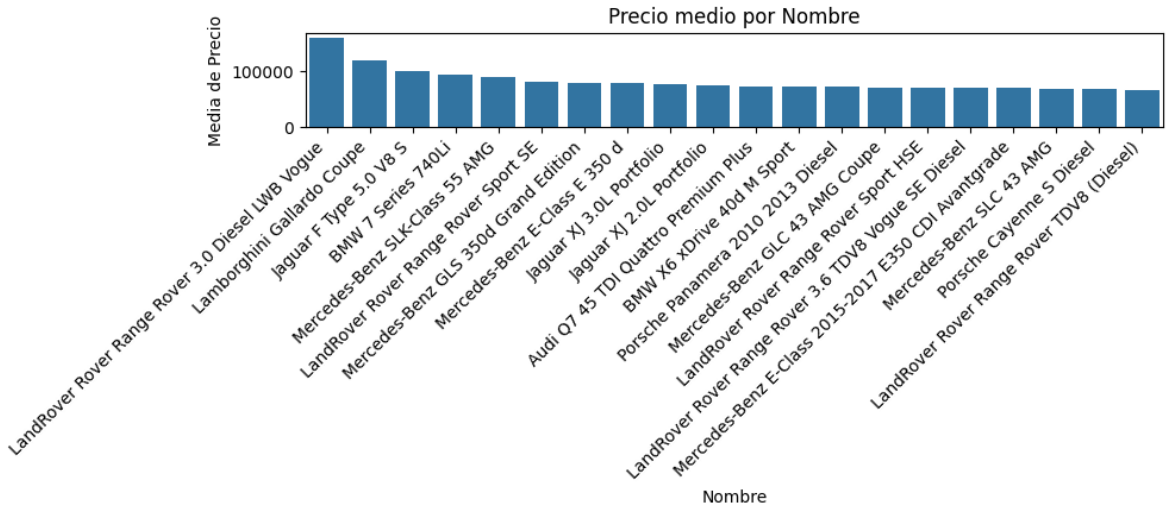


Figura 43. Precio medio por Nombre

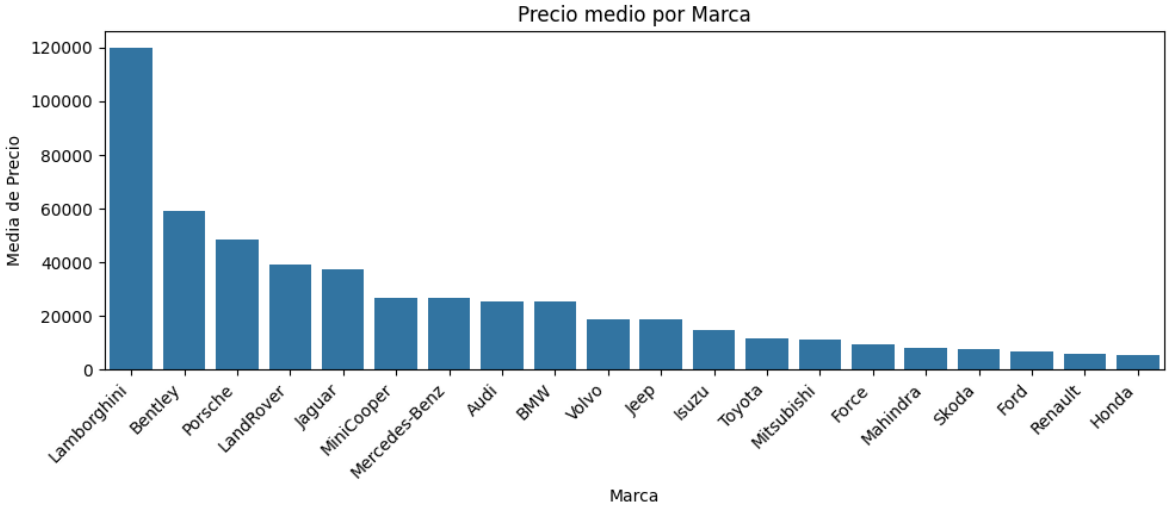


Figura 44. Precio medio por Marca

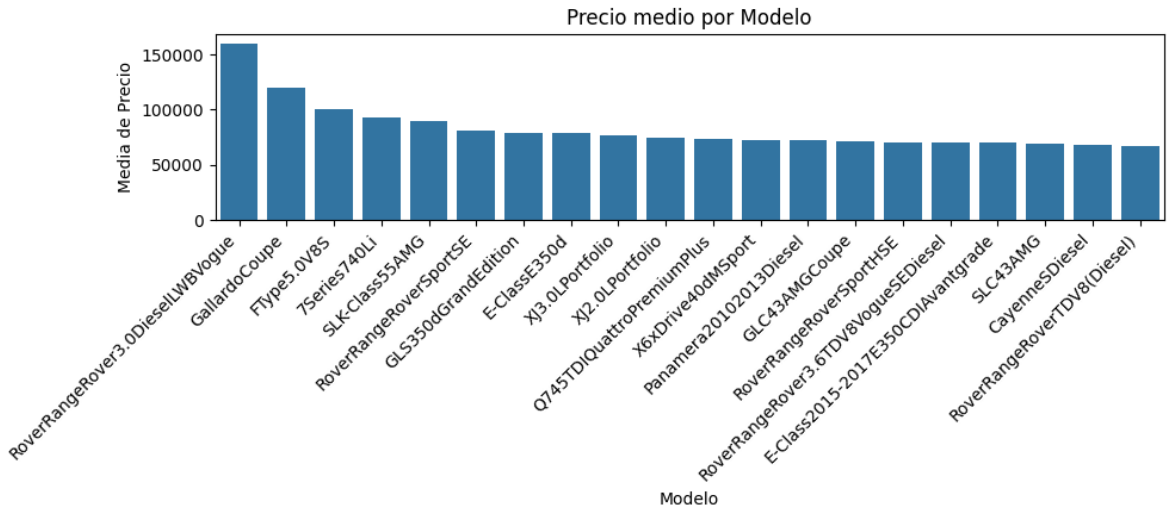


Figura 45. Precio medio por Modelo

En estas nuevas comparaciones se puede observar que la escala del precio ahora está multiplicada por mil, además de que ahora tenemos el precio medio por Marca y Modelo. En ambas podemos ver el sesgo marcado a la derecha por tratarse de autos caros.
Un ejemplo es la marca Lamborghini liderando el precio más alto sobre las otras marcas.

6. *ANÁLISIS FINAL Y CONCLUSIONES.*

Hablando primero en un contexto académico-personal, me gustaría mencionar que esta práctica fue de mucha ayuda para mi ya que nunca había hecho un reporte con todos los pasos que implica un preprocesamiento de datos y su documentación, siempre había sido de forma parcial (1 paso o el otro). Aunque fue un poco demandante, yo supongo que este trabajo puede ser la base de lo que nos podrían llegar a pedir para el TT o trabajo empresarial por lo que me satisface haber hecho una práctica bien lograda.

Ahora, al haber hecho paso a paso las instrucciones dadas y finalmente hacer comparaciones entre gráficas (antes y después), considero que los análisis más "técnicos" ya los he mencionado según mi razonamiento por lo que sólo cabe destacar que:

El análisis gráfico mediante la matriz de dispersión me permitió observar la estructura general de las relaciones entre las variables numéricas del conjunto de datos. En la diagonal se puede ver que la variable Price presenta una distribución fuertemente sesgada hacia la derecha, evidenciando que la mayoría de los vehículos se concentran en rangos de bajo costo mientras que sólo un grupo reducido alcaza precios altos. Este comportamiento justifica la aplicación de una transformación logarítmica, que contribuye a estabilizar la varianza y a representar de manera más equilibrada la información.
Y esto se puede notar claramente al ver la primer distribución de Price contra la distribución de log_Precio, donde se puede ver que ésta última tiene una estructura más "normalizada" y no tan sesgada a la derecha.

Es claro que el EDA posterior gana precisión frente al EDA inicial: las distribuciones ya no están "contaminadas" por etiquetas textuales, los resúmenes numéricos son comparables "por dimensión" en una sola tabla, y las relaciones Precio-(Potencia, Motor, Año, Antigüedad, Millaje) surgen con la dirección esperada y magnitudes razonables.

Finalmente, vuelvo a mencionar que es de vital importancia cada uno de los pasos que se realizaron en la práctica ya que cada uno aporta lo necesario para conseguir un análisis más consistente y limpio. No sé si caigo en un error pero yo pienso que este conjunto de procesos es lo que se conoce en la metodología CRISP-DM como el data understanding, te permiten conocer el "esqueleto" de tu dataset y con ello poder hacer trabajos como predicciones o modelados.