In [1]:
```
import pandas as pd
```

In [2]:
```
df=pd.read_csv(r"C:\Users\Saiko\OneDrive\Desktop\Data Analytics\Car Price Pre
```

Dataset Description:

The Car Price Dataset contains 10,000 records with 10 attributes detailing used cars and their resale prices. It includes brand, model, year (2000–2023), engine size (1.0L–5.0L), fuel type, transmission, mileage, doors, owner count, and price ($2,000$–18,301). Newer cars, luxury brands, and lower mileage vehicles generally have higher prices. Automatic, diesel, and hybrid cars also tend to be more valuable. The dataset is ideal for price prediction models and market analysis, revealing trends such as depreciation patterns and the rising popularity of hybrid and electric vehicles due to environmental concerns.

Columns in dataset:

- Brand (object): The car manufacturer (e.g., Kia, Chevrolet, Mercedes, Audi, etc.).
- Model (object): The specific model of the car.
- Year (int64): The manufacturing year of the car (range: 2000 to 2023).
- Engine_Size (float64): The size of the engine in liters (range: 1.0L to 5.0L).
- Fuel_Type (object): The type of fuel used (e.g., Diesel, Hybrid, Electric).
- Transmission (object): The type of transmission (e.g., Manual, Automatic, Semi-Automatic).
- Mileage (int64): The total distance the car has traveled, in kilometers (range: 25 to 299,947).
- Doors (int64): The number of doors (range: 2 to 5).
- Owner_Count (int64): The number of previous owners (range: 1 to 5).
- Price (int64): The selling price of the car in USD (range: $2,000 to$18,301)

In [3]:
```
df
```

Out[3]:

	Brand	Model	Year	Engine_Size	Fuel_Type	Transmission	Mileage	Doors	Owne
0	Kia	Rio	2020	4.2	Diesel	Manual	289944	3	
1	Chevrolet	Malibu	2012	2.0	Hybrid	Automatic	5356	2	
2	Mercedes	GLA	2020	4.2	Diesel	Automatic	231440	4	
3	Audi	Q5	2023	2.0	Electric	Manual	160971	2	
4	Volkswagen	Golf	2003	2.6	Hybrid	Semi-Automatic	286618	3	
...	...	...	...	...	...	...	...	...	
9995	Kia	Optima	2004	3.7	Diesel	Semi-Automatic	5794	2	
9996	Chevrolet	Impala	2002	1.4	Electric	Automatic	168000	2	
9997	BMW	3 Series	2010	3.0	Petrol	Automatic	86664	5	
9998	Ford	Explorer	2002	1.4	Hybrid	Automatic	225772	4	
9999	Volkswagen	Tiguan	2001	2.1	Diesel	Manual	157882	3	

10000 rows × 10 columns

In [4]:
```
df.isnull().sum()
```

Out[4]:
```
Brand           0
Model           0
Year            0
Engine_Size     0
Fuel_Type       0
Transmission    0
Mileage         0
Doors           0
Owner_Count     0
Price           0
dtype: int64
```

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Brand         10000 non-null  object 
 1   Model         10000 non-null  object 
 2   Year          10000 non-null  int64  
 3   Engine_Size   10000 non-null  float64
 4   Fuel_Type     10000 non-null  object 
 5   Transmission  10000 non-null  object 
 6   Mileage       10000 non-null  int64  
 7   Doors         10000 non-null  int64  
 8   Owner_Count   10000 non-null  int64  
 9   Price         10000 non-null  int64  
dtypes: float64(1), int64(5), object(4)
memory usage: 781.4+ KB
```

In [6]:
```
from sklearn.linear_model import LinearRegression
from sklearn import linear_model
```

In [7]:
```
brand=pd.get_dummies(df['Brand'],prefix="Brand")
```

In [8]:
```
model=pd.get_dummies(df['Model'],prefix="Model")
```

In [9]:
```
fuel_type=pd.get_dummies(df['Fuel_Type'],prefix="Fuel_Type")
```

In [10]:
```
transmission=pd.get_dummies(df['Transmission'],prefix="Transmission")
```

In [11]:
```
print(brand)
```

```
      Brand_Audi  Brand_BMW  Brand_Chevrolet  Brand_Ford  Brand_Honda  \
0          False      False            False       False        False
1          False      False             True       False        False
2          False      False            False       False        False
3           True      False            False       False        False
4          False      False            False       False        False
...          ...        ...              ...         ...          ...
9995       False      False            False       False        False
9996       False      False             True       False        False
9997       False       True            False       False        False
9998       False      False            False        True        False
9999       False      False            False       False        False

      Brand_Hyundai  Brand_Kia  Brand_Mercedes  Brand_Toyota  Brand_Volkswa
gen
0             False       True           False         False             Fa
lse
1             False      False           False         False             Fa
lse
2             False      False            True         False             Fa
lse
3             False      False           False         False             Fa
lse
4             False      False           False         False              T
rue
...             ...        ...             ...           ...
...
9995          False       True           False         False             Fa
lse
9996          False      False           False         False             Fa
lse
9997          False      False           False         False             Fa
lse
9998          False      False           False         False             Fa
lse
9999          False      False           False         False              T
rue

[10000 rows x 10 columns]
```

In [12]:
```
print(model)
```

```
      Model_3 Series  Model_5 Series  Model_A3  Model_A4  Model_Accord  \
0              False           False     False     False         False
1              False           False     False     False         False
2              False           False     False     False         False
3              False           False     False     False         False
4              False           False     False     False         False
...              ...             ...       ...       ...           ...
9995           False           False     False     False         False
9996           False           False     False     False         False
9997            True           False     False     False         False
9998           False           False     False     False         False
9999           False           False     False     False         False

      Model_C-Class  Model_CR-V  Model_Camry  Model_Civic  Model_Corolla
...  \
0             False       False        False        False          False
...
1             False       False        False        False          False
...
2             False       False        False        False          False
...
3             False       False        False        False          False
...
4             False       False        False        False          False
...
...             ...         ...          ...          ...            ...
...
9995          False       False        False        False          False
...
9996          False       False        False        False          False
...
9997          False       False        False        False          False
...
9998          False       False        False        False          False
...
9999          False       False        False        False          False
...

      Model_Optima  Model_Passat  Model_Q5  Model_RAV4  Model_Rio  \
0            False         False     False       False       True
1            False         False     False       False      False
2            False         False     False       False      False
3            False         False      True       False      False
4            False         False     False       False      False
...            ...           ...       ...         ...        ...
9995          True         False     False       False      False
9996         False         False     False       False      False
9997         False         False     False       False      False
9998         False         False     False       False      False
9999         False         False     False       False      False

      Model_Sonata  Model_Sportage  Model_Tiguan  Model_Tucson  Model_X5
0            False           False         False         False     False
1            False           False         False         False     False
2            False           False         False         False     False
3            False           False         False         False     False
4            False           False         False         False     False
```

```
...          ...          ...          ...          ...          ...
9995       False        False        False        False        False
9996       False        False        False        False        False
9997       False        False        False        False        False
9998       False        False        False        False        False
9999       False        False         True        False        False

[10000 rows x 30 columns]
```

In [13]:

```
print(transmission)
```

```
      Transmission_Automatic  Transmission_Manual  Transmission_Semi-Automa
tic
0                      False                 True                        Fa
lse
1                       True                False                        Fa
lse
2                       True                False                        Fa
lse
3                      False                 True                        Fa
lse
4                      False                False                         T
rue
...                      ...                  ...                          
...
9995                   False                False                         T
rue
9996                    True                False                        Fa
lse
9997                    True                False                        Fa
lse
9998                    True                False                        Fa
lse
9999                   False                 True                        Fa
lse

[10000 rows x 3 columns]
```

In [14]:
```
print(fuel_type)
```

```
      Fuel_Type_Diesel  Fuel_Type_Electric  Fuel_Type_Hybrid  Fuel_Type_Pet
rol
0                 True               False             False              Fa
lse
1                False               False              True              Fa
lse
2                 True               False             False              Fa
lse
3                False                True             False              Fa
lse
4                False               False              True              Fa
lse
...                ...                 ...               ...
...
9995              True               False             False              Fa
lse
9996             False                True             False              Fa
lse
9997             False               False             False               T
rue
9998             False               False              True              Fa
lse
9999              True               False             False              Fa
lse

[10000 rows x 4 columns]
```

In [15]:
```
df.drop(["Brand","Model","Fuel_Type","Transmission"],axis=1,inplace=True)
```

In [16]:
```
df=pd.concat([df,brand,model,fuel_type,transmission],axis=1)
```

In [17]:
```
reg=linear_model.LinearRegression()
reg.fit(df.drop('Price',axis='columns'),df.Price)
```

Out[17]:
```
LinearRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [18]:
```
reg.coef_
```

Out[18]:
```
array([ 2.98601356e+02,  9.92739311e+02, -1.98902437e-02, -5.50438898e-01,
        3.58753313e-02, -3.00390404e+00, -9.13203152e-01, -1.55918923e+00,
       -1.38937007e-01,  3.95555544e+00, -1.86013447e+00,  4.31507789e+00,
        2.03848302e+00, -1.10822340e+00, -1.72552506e+00,  2.86758378e-01,
       -6.77909298e+00,  1.13687704e+00, -3.18040004e+00, -9.45409685e-01,
       -6.93412569e-01, -1.24997283e+00, -4.99892135e+00,  6.15093795e+00,
       -2.03858422e+00, -3.47440863e+00, -2.29589683e+00, -3.14473413e+00,
        3.99551403e-01, -1.85058540e+00,  1.31209699e+00,  6.20630422e+00,
       -8.45653658e-01,  1.79503760e+00, -2.09492702e-01,  1.22018606e+00,
       -1.77553804e-02, -9.60381043e-01,  5.92928217e+00,  1.26855132e+00,
       -1.08576609e+00,  1.82634052e+00, -8.62116017e-01,  1.52152846e+00,
        5.57913145e+00, -7.44851446e+02,  1.24307000e+03,  2.45486884e+02,
       -7.43705441e+02,  9.94064951e+02, -4.96166920e+02, -4.97898031e+02])
```

In [19]:
```
reg.intercept_
```

Out[19]:
```
-591822.953814718
```

In [20]:
```
df
```

Out[20]:

	Year	Engine_Size	Mileage	Doors	Owner_Count	Price	Brand_Audi	Brand_BMW	Brai
0	2020	4.2	289944	3	5	8501	False	False	
1	2012	2.0	5356	2	3	12092	False	False	
2	2020	4.2	231440	4	2	11171	False	False	
3	2023	2.0	160971	2	1	11780	True	False	
4	2003	2.6	286618	3	3	2867	False	False	
...	...	...	...	...	...	...	...	...	
9995	2004	3.7	5794	2	4	8884	False	False	
9996	2002	1.4	168000	2	1	6240	False	False	
9997	2010	3.0	86664	5	1	9866	False	True	
9998	2002	1.4	225772	4	1	4084	False	False	
9999	2001	2.1	157882	3	3	3342	False	False	

10000 rows × 53 columns

In [21]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

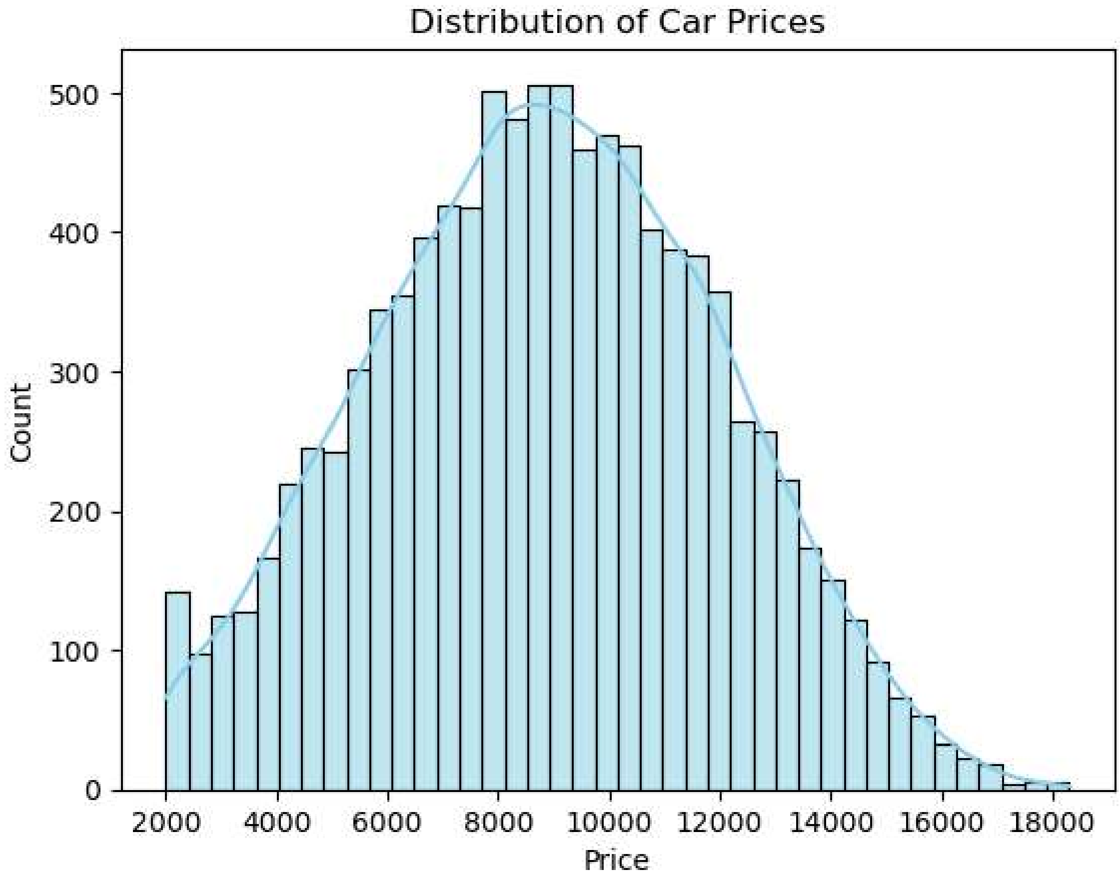
In [22]:
```
plt.figure(figsize=(15, 10))
```

Out[22]:
```
<Figure size 1500x1000 with 0 Axes>
```

```
<Figure size 1500x1000 with 0 Axes>
```

In [23]:
```
sns.histplot(df['Price'], kde=True, color='skyblue')
plt.title('Distribution of Car Prices')
```

Out[23]:
```
Text(0.5, 1.0, 'Distribution of Car Prices')
```



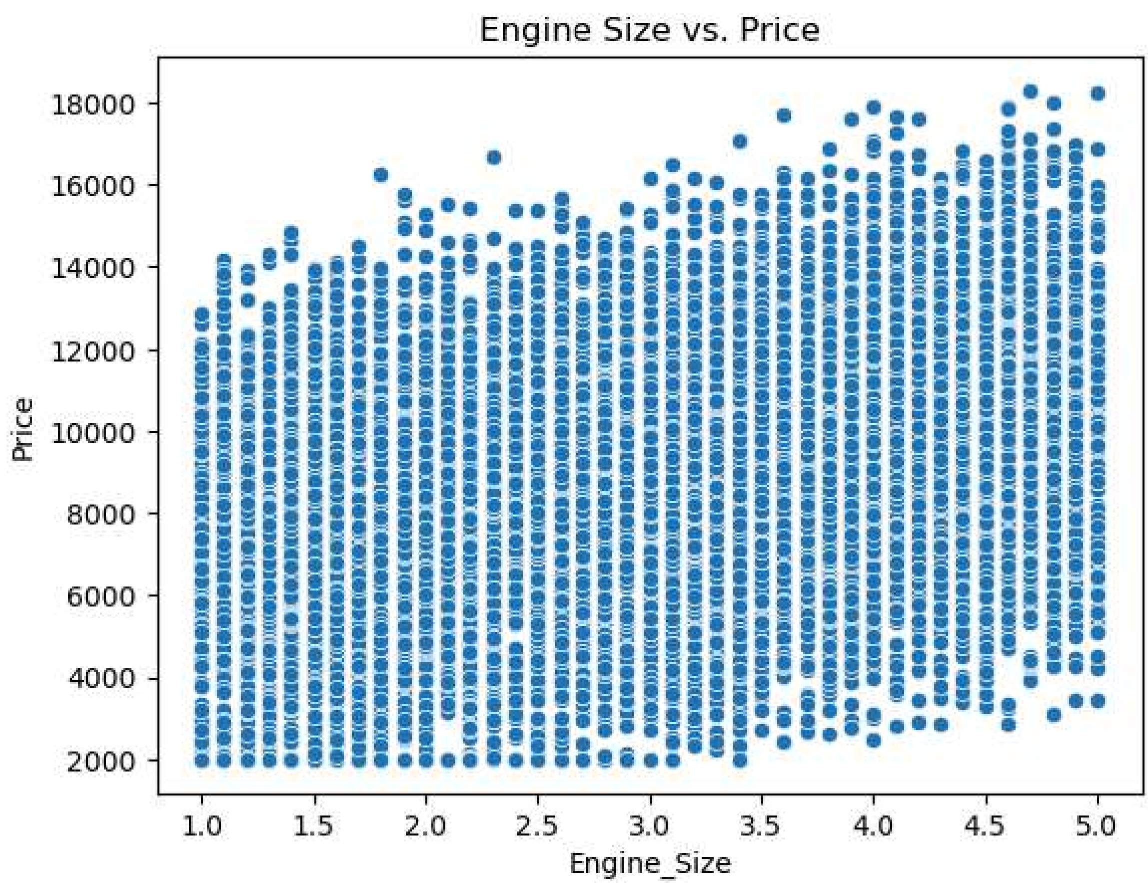
In [24]:

```
sns.scatterplot(x='Engine_Size', y='Price', data=df, palette='viridis')
plt.title('Engine Size vs. Price')
```

```
C:\Users\Saiko\AppData\Local\Temp\ipykernel_17756\28750575.py:1: UserWarnin
g: Ignoring `palette` because no `hue` variable has been assigned.
  sns.scatterplot(x='Engine_Size', y='Price', data=df, palette='viridis')
```

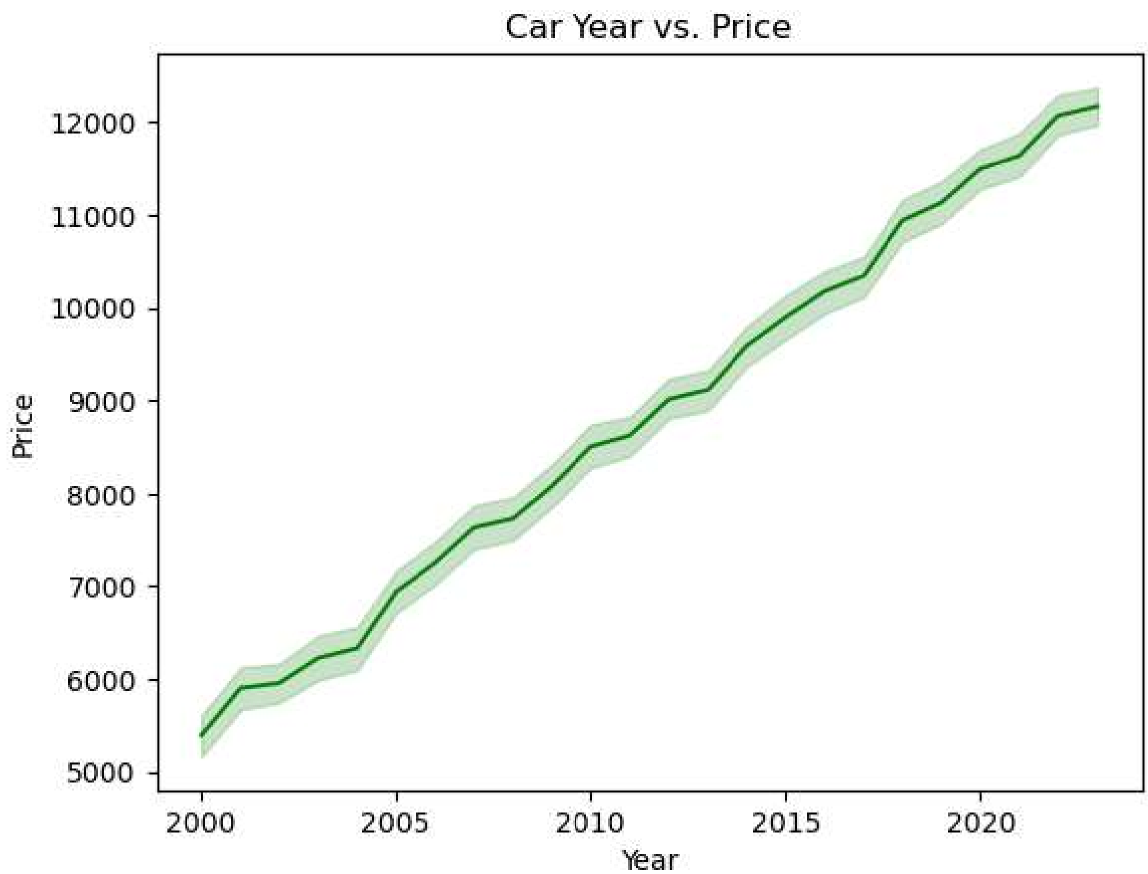
Out[24]:

```
Text(0.5, 1.0, 'Engine Size vs. Price')
```



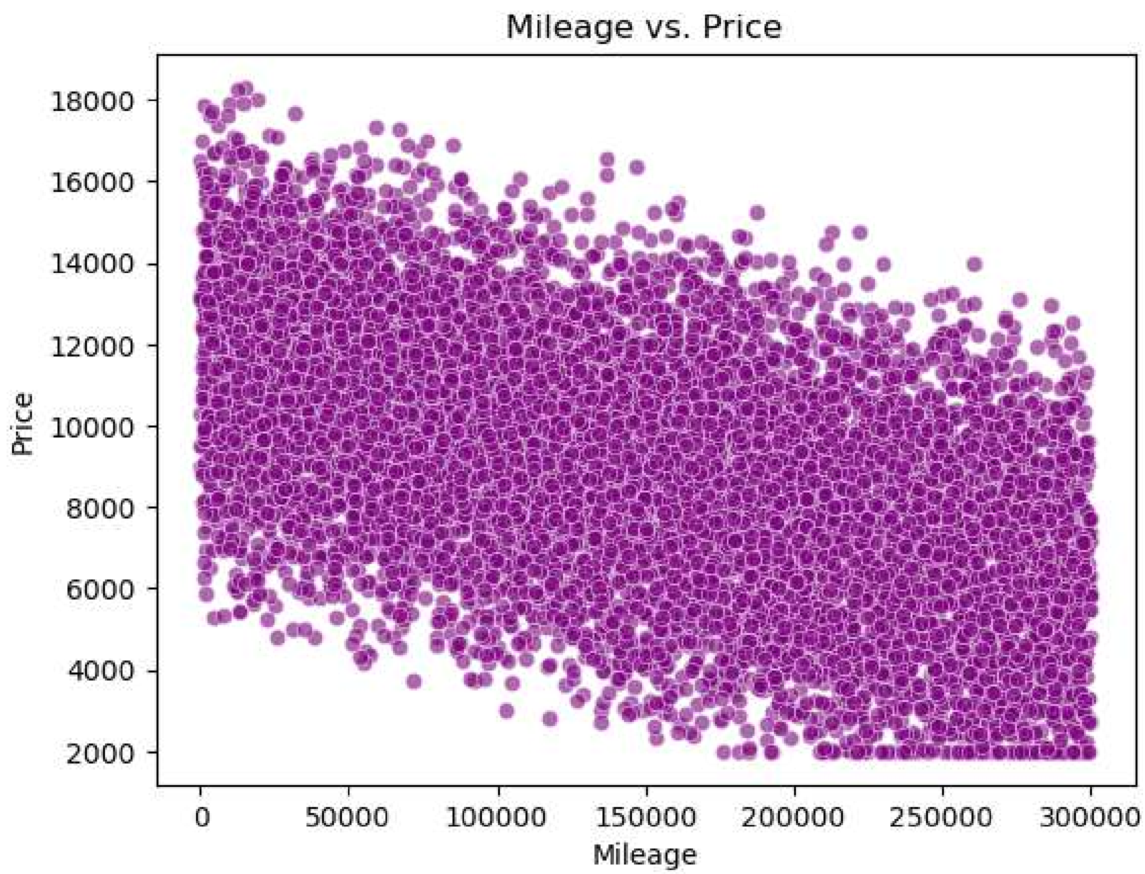
In [25]:
```
sns.lineplot(x='Year', y='Price', data=df, color='green')
plt.title('Car Year vs. Price')
```

Out[25]: Text(0.5, 1.0, 'Car Year vs. Price')



In [26]:
```
sns.scatterplot(x='Mileage', y='Price', data=df, color='purple', alpha=0.6)
plt.title('Mileage vs. Price')
```

Out[26]: Text(0.5, 1.0, 'Mileage vs. Price')

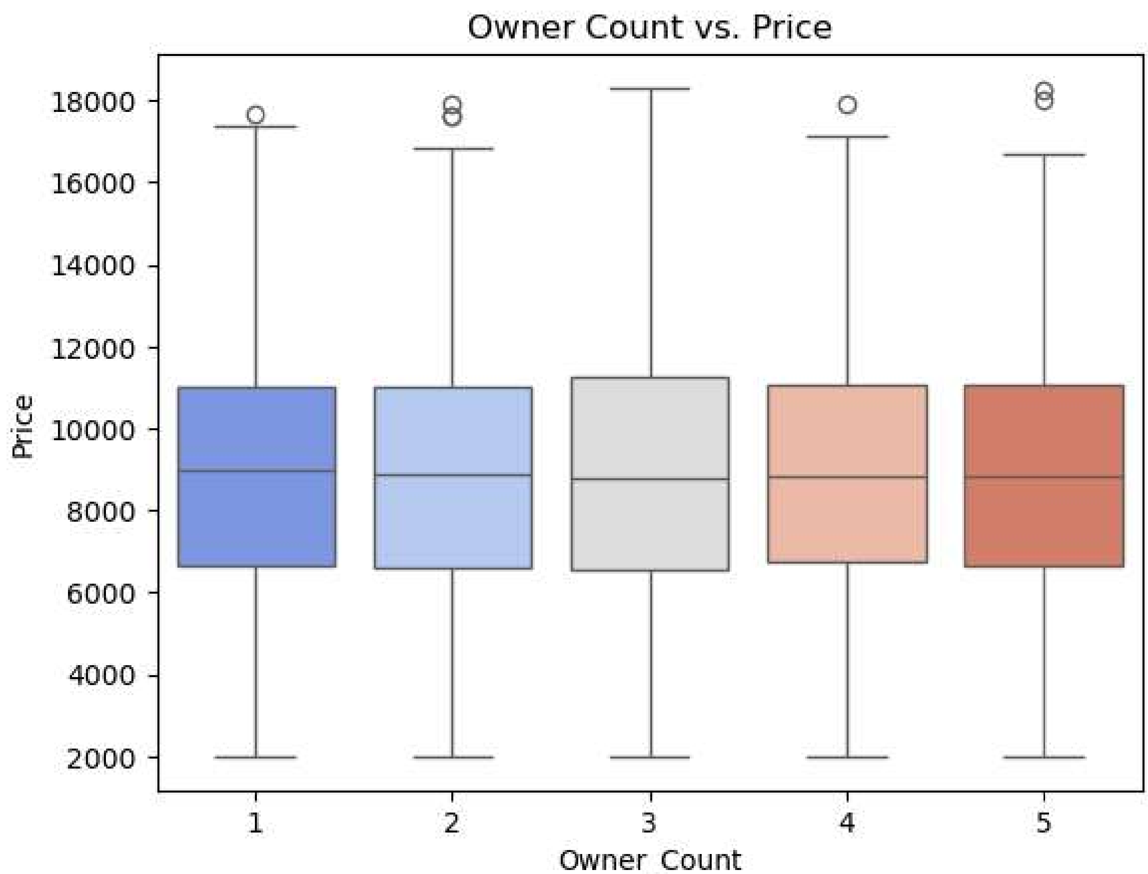


In [27]:

```
sns.boxplot(x='Owner_Count', y='Price', data=df, palette='coolwarm')
plt.title('Owner Count vs. Price')
```

```
C:\Users\Saiko\AppData\Local\Temp\ipykernel_17756\1912413548.py:1: FutureWa
rning:

Passing `palette` without assigning `hue` is deprecated and will be removed
in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the
same effect.

  sns.boxplot(x='Owner_Count', y='Price', data=df, palette='coolwarm')
```

Out[27]: Text(0.5, 1.0, 'Owner Count vs. Price')

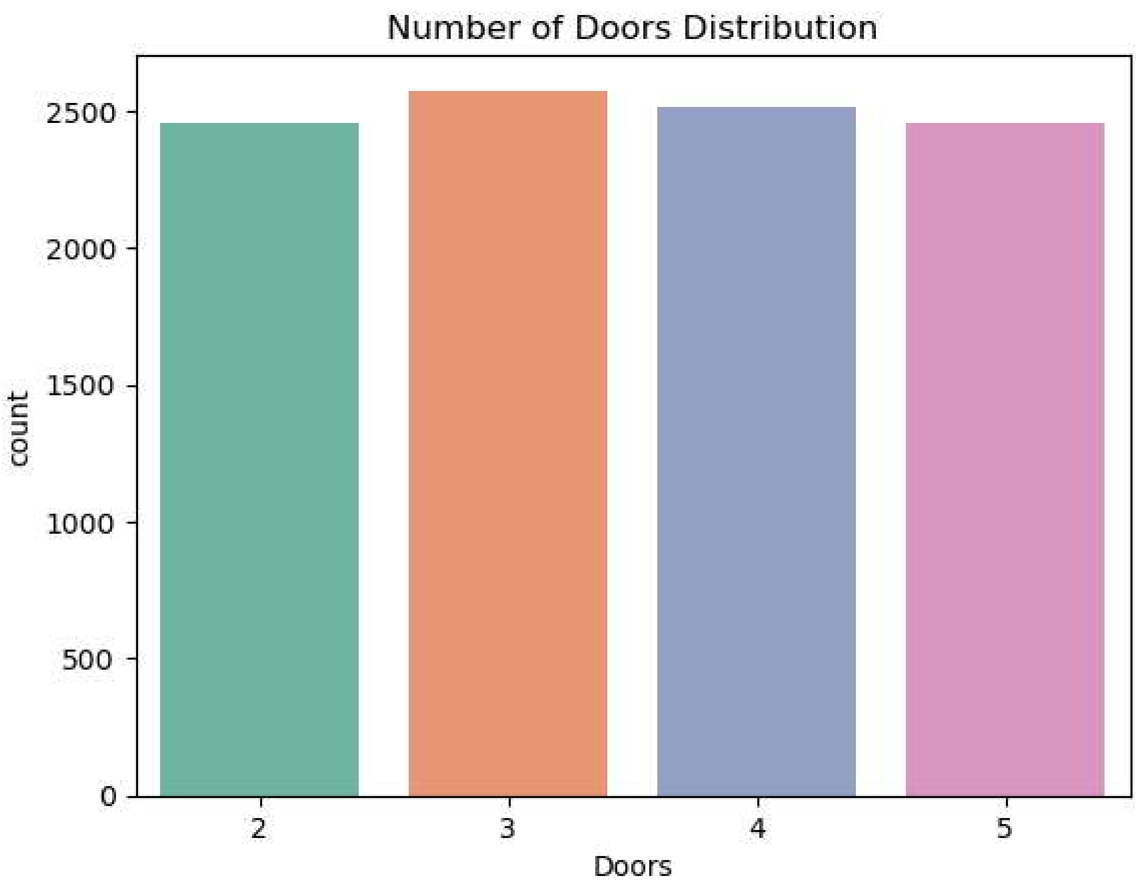


In [28]:

```
sns.countplot(x='Doors', data=df, palette='Set2')
plt.title('Number of Doors Distribution')
```

```
C:\Users\Saiko\AppData\Local\Temp\ipykernel_17756\283035502.py:1: FutureWar
ning:

Passing `palette` without assigning `hue` is deprecated and will be removed
in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the
same effect.

  sns.countplot(x='Doors', data=df, palette='Set2')
```

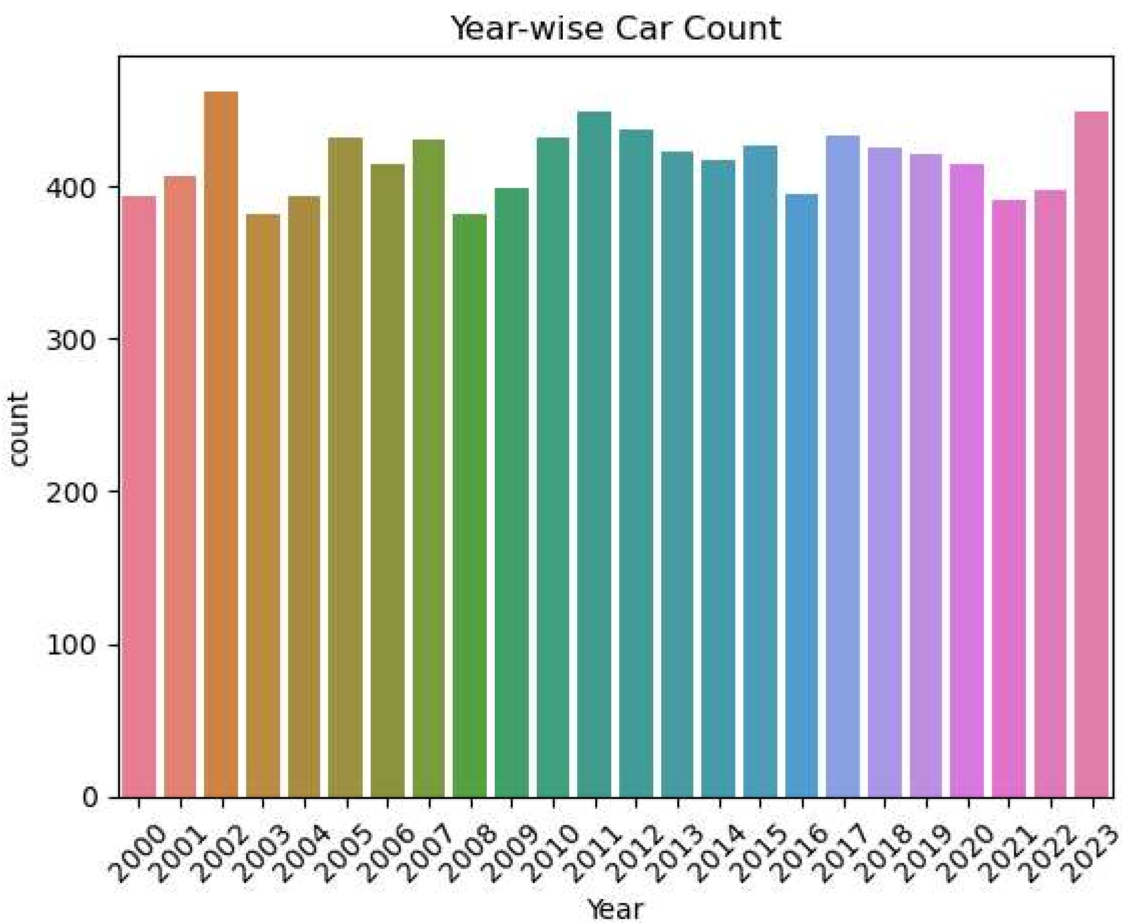
Out[28]: Text(0.5, 1.0, 'Number of Doors Distribution')



In [29]:
```
sns.countplot(x='Year', data=df, palette='husl')
plt.xticks(rotation=45)
plt.title('Year-wise Car Count')
```

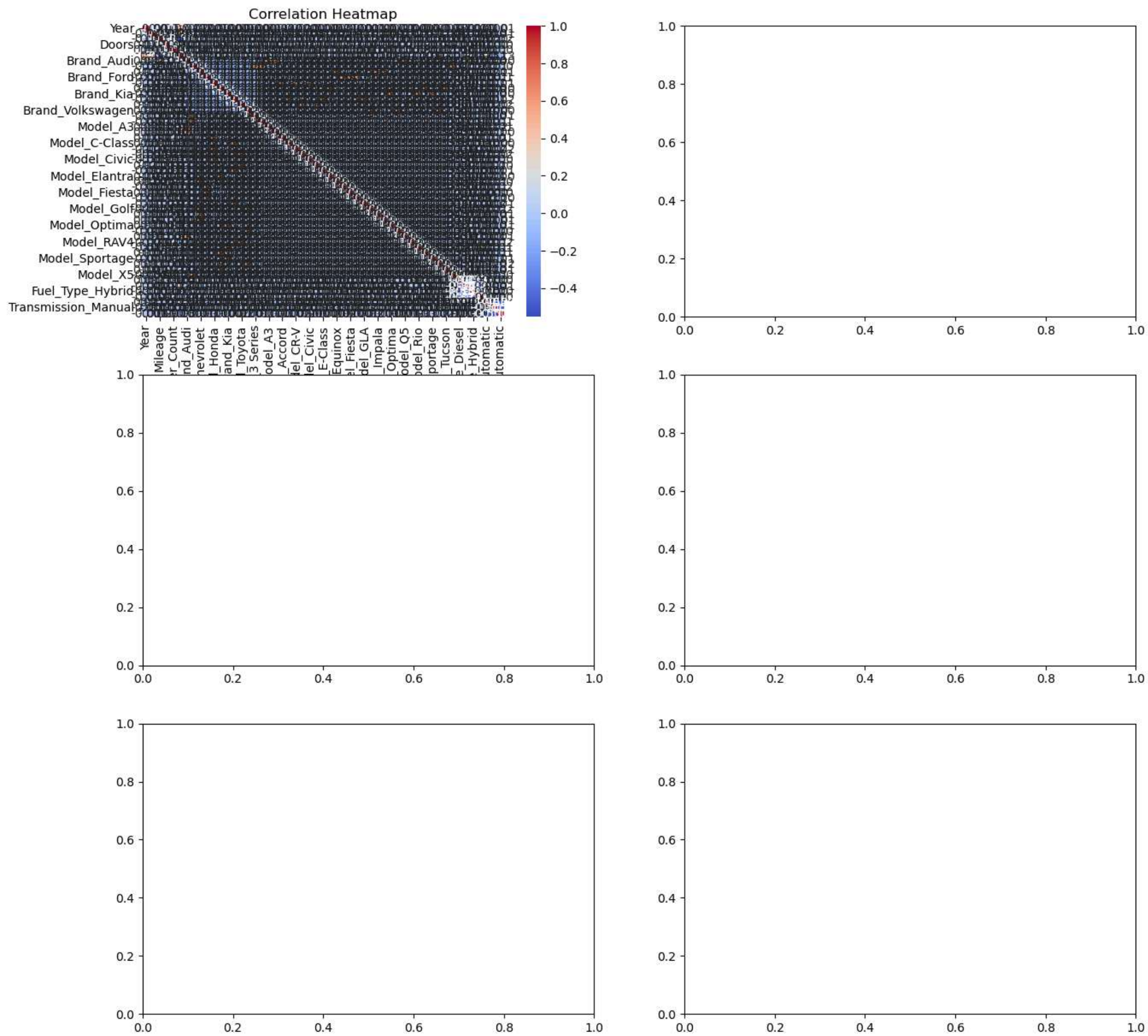
```
C:\Users\Saiko\AppData\Local\Temp\ipykernel_17756\4274427044.py:1: FutureWa
rning:

Passing `palette` without assigning `hue` is deprecated and will be removed
in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the
same effect.

  sns.countplot(x='Year', data=df, palette='husl')
```

Out[29]: Text(0.5, 1.0, 'Year-wise Car Count')

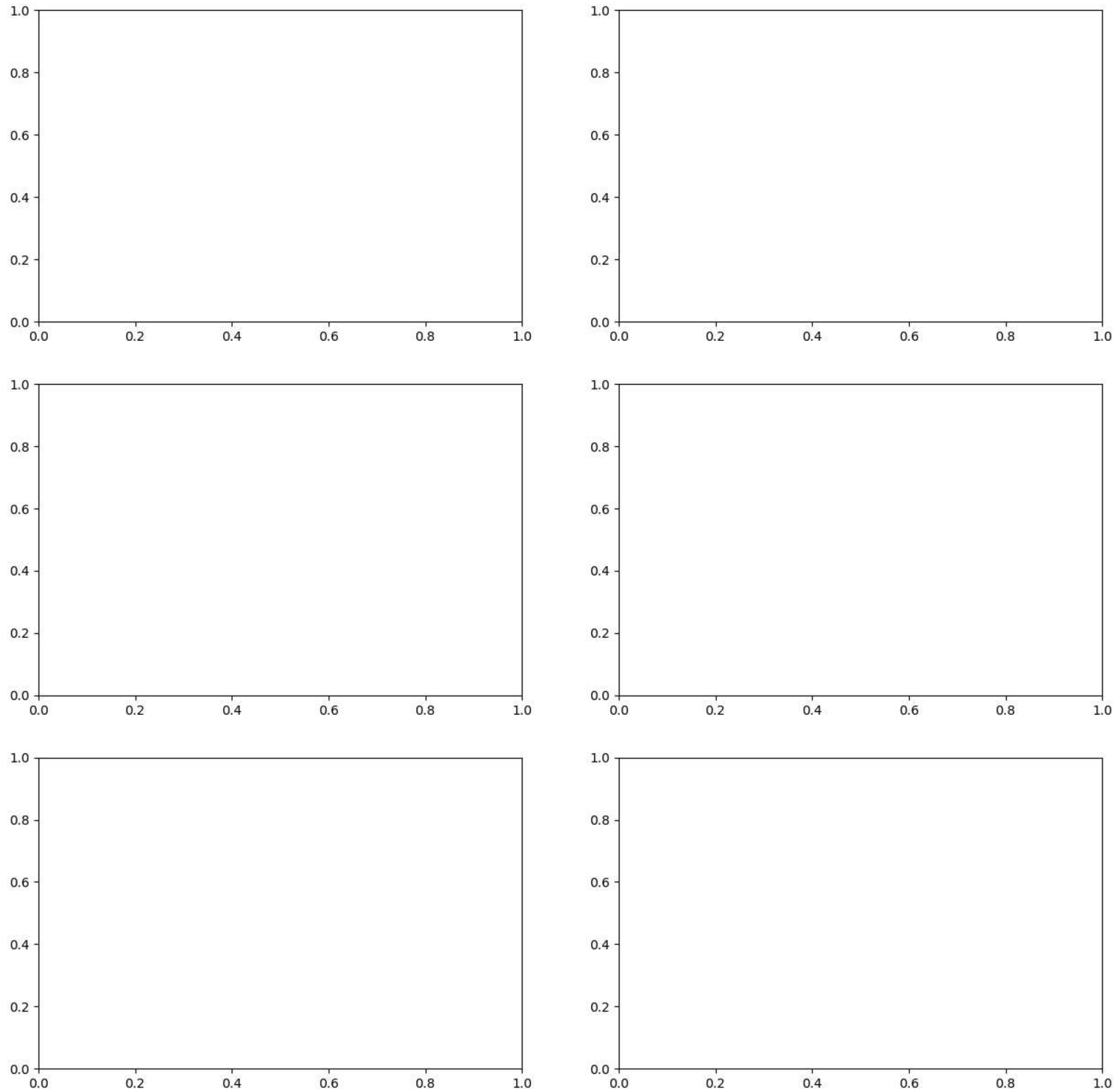


In [30]:
```
fig, axes = plt.subplots(3, 2, figsize=(15, 15))
axes = axes.flatten()
corr = df.corr(numeric_only=True)
sns.heatmap(corr, annot=True, fmt=".2f", cmap='coolwarm', ax=axes[0])
axes[0].set_title('Correlation Heatmap')
```

Out[30]: Text(0.5, 1.0, 'Correlation Heatmap')

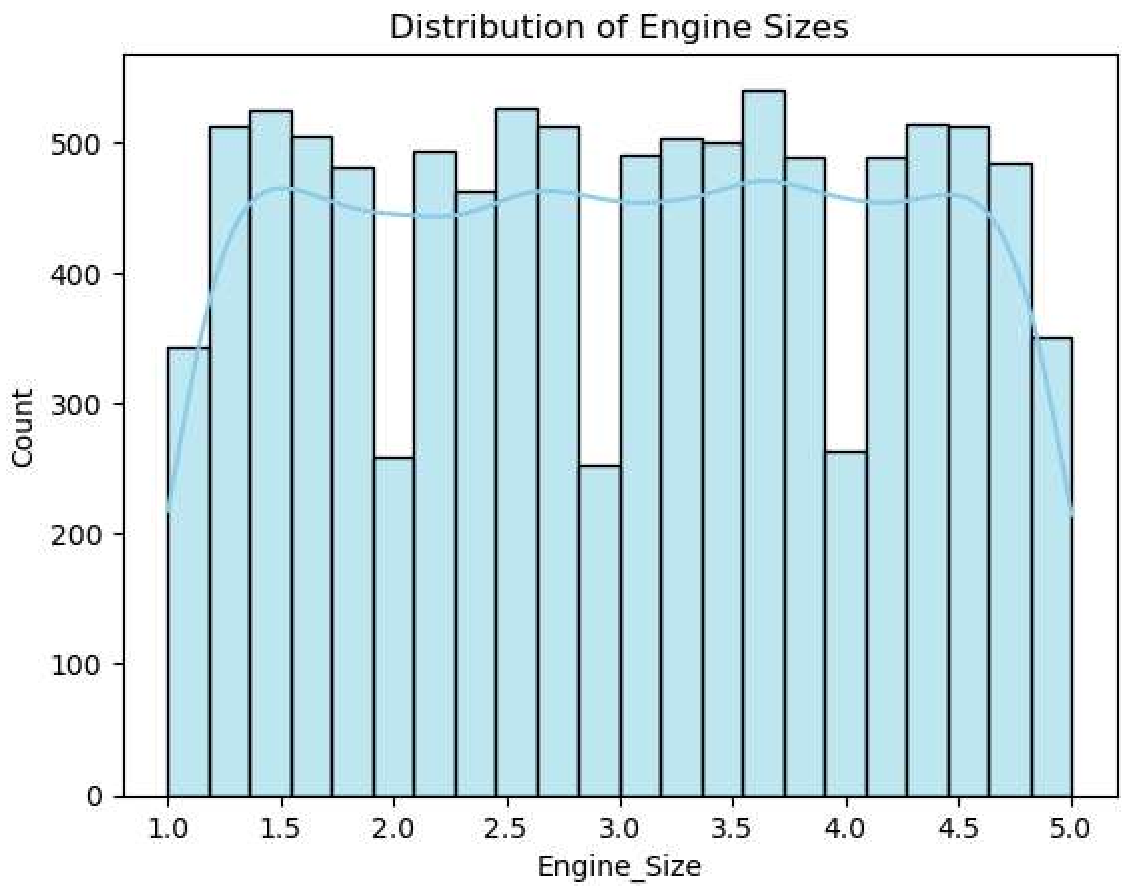


In [31]:
```
fig, axes = plt.subplots(3, 2, figsize=(15, 15))
axes = axes.flatten()
```



In [32]:
```
sns.histplot(df['Engine_Size'], kde=True, color='skyblue')
plt.title('Distribution of Engine Sizes')
```

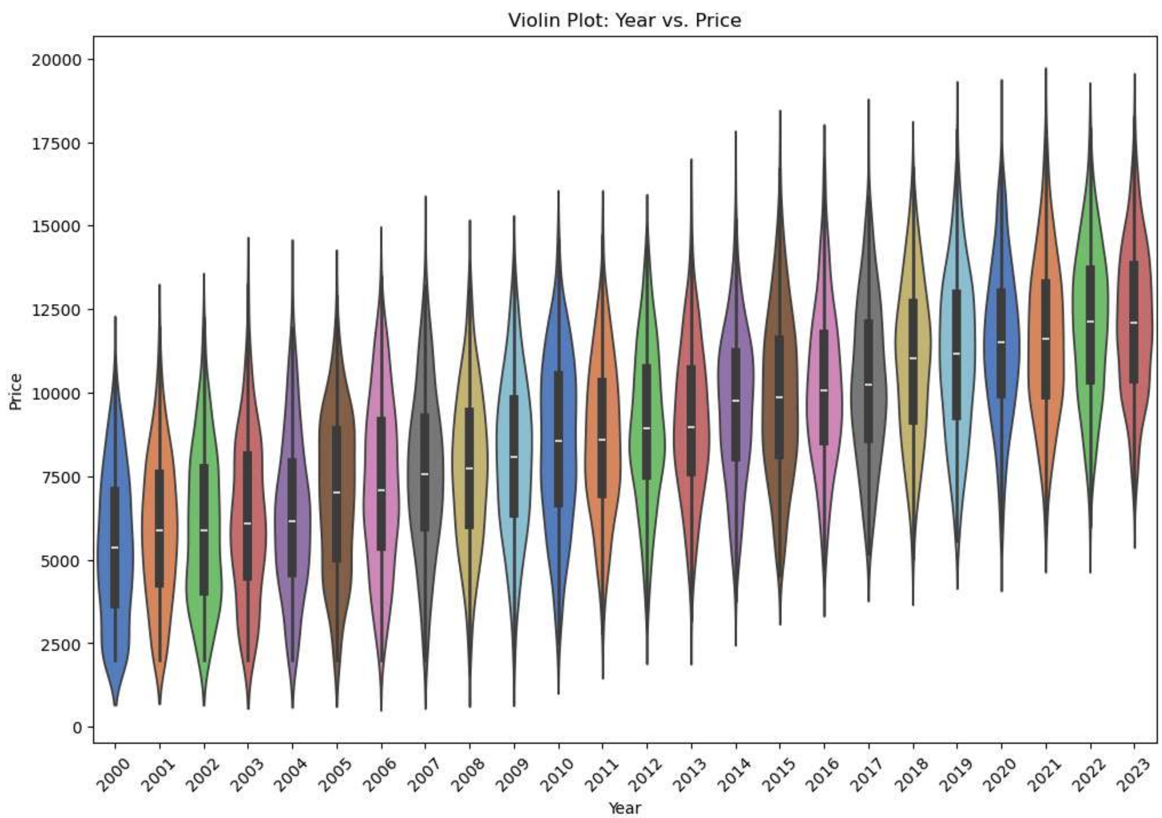
Out[32]:
```
Text(0.5, 1.0, 'Distribution of Engine Sizes')
```



In [33]:

```
plt.figure(figsize=(12, 8))
sns.violinplot(x='Year', y='Price', data=df, palette='muted')
plt.title('Violin Plot: Year vs. Price')
plt.xticks(rotation=45)
plt.show()
```

```
C:\Users\Saiko\AppData\Local\Temp\ipykernel_17756\3991876073.py:2: FutureWa
rning:

Passing `palette` without assigning `hue` is deprecated and will be removed
in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the
same effect.

  sns.violinplot(x='Year', y='Price', data=df, palette='muted')
```



DataSet Observation: Insights from the Car Price Dataset Visualizations

1. Price Trends by Year
    - The Price vs. Year scatter plot and violin plot reveal that newer cars (post-2015) have significantly higher prices, while older models tend to be cheaper.
    - The median price steadily increases for recent cars, reflecting their higher market value.
2. Mileage and Price Relationship
    - The Mileage vs. Price scatter plot highlights a clear trend:
    - Cars with higher mileage tend to have lower prices, reflecting wear and depreciation.
    - Lower-mileage cars retain their value better, especially newer models.
3. Engine Size Impact
    - The Engine Size vs. Price and Engine Size vs. Mileage scatter plots show that:

   - Cars with larger engines generally have higher prices, indicating more powerful and premium vehicles.
   - Larger engines tend to have lower mileage, suggesting they may be used less frequently or preserved for special purposes.
4. Doors and Pricing The Price Distribution by Number of Doors plot shows that:
   - 4-door cars dominate the market and have the widest price range, likely due to their popularity and versatility.
   - 2-door cars are generally priced lower, possibly because they are less practical for families.
   - 5-door cars exhibit moderate pricing, often associated with hatchbacks or compact SUVs.
5. Ownership Patterns The Owner Count vs. Price plot reveals that:
   - Cars with fewer previous owners generally have higher prices, as they are perceived to be better maintained.
   - Cars with 3 or more owners tend to have lower prices, likely due to increased wear and potential maintenance issues.
6. Fuel Type and Transmission Insights From the earlier plots (which included fuel and transmission types):
   - Hybrid and electric cars have higher prices, reflecting their growing demand and eco-friendliness.
   - Automatic cars generally have higher resale values compared to manual cars, indicating a consumer preference for convenience.
7. Yearly Car Trends The Year vs. Number of Doors and Year-wise Car Count plots show that:
   - The production of cars peaked between 2015 and 2020, indicating higher availability of newer cars in the dataset.

In [ ]: