Analysis of Toyota Corolla Car Specifications and Pricing

Objective:

To analyze the various factors that influence the pricing of Toyota Corolla cars. This includes exploring the relationships between car features (such as age, mileage, horsepower, and additional features) and their prices.

Id: A unique identifier for each car.

Model: The specific model of the car.

Price: The price of the car.

Age_08_04: The age of the car in months as of August 2004.

Mfg_Month: The manufacturing month.

Mfg_Year: The manufacturing year.

KM: The mileage of the car in kilometers.

Fuel_Type: The car's fuel type (e.g., Diesel, Petrol).

HP: Horsepower of the car.

Met_Color: Whether the car has metallic color (binary indicator).

Features: Various other features and specifications like central locking, powered windows, power steering, radio, mist lamps, sport model, backseat divider, metallic rim, radio cassette, and tow bar.

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib as plt
from sklearn.preprocessing import OneHotEncoder, StandardScaler,
PolynomialFeatures
from sklearn.impute import SimpleImputer
```

Data Cleaning and preparation

```
toyota_data=pd.read_csv(r"C:\Preet\ToyotaCorolla.csv",encoding='UTF-8-
SIG')

toyota_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1436 entries, 0 to 1435
Data columns (total 38 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Id                1436 non-null   int64
 1   Model             1436 non-null   object
 2   Price             1436 non-null   int64
 3   Age_08_04         1436 non-null   int64
 4   Mfg_Month         1436 non-null   int64
 5   Mfg_Year          1436 non-null   int64
 6   KM                1436 non-null   int64
 7   Fuel_Type         1436 non-null   object
 8   HP                1436 non-null   int64
 9   Met_Color         1436 non-null   int64
 10  Color             1436 non-null   object
 11  Automatic         1436 non-null   int64
 12  cc                1436 non-null   int64
 13  Doors             1436 non-null   int64
 14  Cylinders         1436 non-null   int64
 15  Gears             1436 non-null   int64
 16  Quarterly_Tax     1436 non-null   int64
 17  Weight            1436 non-null   int64
```

```
 18  Mfr_Guarantee     1436 non-null   int64
 19  BOVAG_Guarantee   1436 non-null   int64
 20  Guarantee_Period  1436 non-null   int64
 21  ABS               1436 non-null   int64
 22  Airbag_1          1436 non-null   int64
 23  Airbag_2          1436 non-null   int64
 24  Airco             1436 non-null   int64
 25  Automatic_airco   1436 non-null   int64
 26  Boardcomputer     1436 non-null   int64
 27  CD_Player         1436 non-null   int64
 28  Central_Lock      1436 non-null   int64
 29  Powered_Windows   1436 non-null   int64
 30  Power_Steering    1436 non-null   int64
 31  Radio             1436 non-null   int64
 32  Mistlamps         1436 non-null   int64
 33  Sport_Model       1436 non-null   int64
 34  Backseat_Divider  1436 non-null   int64
 35  Metallic_Rim      1436 non-null   int64
 36  Radio_cassette    1436 non-null   int64
 37  Tow_Bar           1436 non-null   int64
dtypes: int64(35), object(3)
memory usage: 426.4+ KB
```

```
toyota_data.head()
```

```
   Id                                           Model  Price
Age_08_04  \
0   1   TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors  13500
23
1   2   TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors  13750
23
2   3   TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors  13950
24
3   4   TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors  14950
26
4   5     TOYOTA Corolla 2.0 D4D HATCHB SOL 2/3-Doors  13750
30

   Mfg_Month  Mfg_Year     KM Fuel_Type  HP  Met_Color  ...
Central_Lock  \
0         10      2002  46986    Diesel  90          1  ...
1
1         10      2002  72937    Diesel  90          1  ...
1
2          9      2002  41711    Diesel  90          1  ...
0
3          7      2002  48000    Diesel  90          0  ...
0
4          3      2002  38500    Diesel  90          0  ...
1
```

```
   Powered_Windows  Power_Steering  Radio  Mistlamps  Sport_Model  \
0                1               1      0          0            0
1                0               1      0          0            0
2                0               1      0          0            0
3                0               1      0          0            0
4                1               1      0          1            0

   Backseat_Divider  Metallic_Rim  Radio_cassette  Tow_Bar
0                 1             0               0        0
1                 1             0               0        0
2                 1             0               0        0
3                 1             0               0        0
4                 1             0               0        0

[5 rows x 38 columns]
```

```
toyota_data.describe()
```

```
                Id         Price    Age_08_04    Mfg_Month
Mfg_Year  \
count  1436.000000   1436.000000  1436.000000  1436.000000
1436.000000
mean    721.555014  10730.824513    55.947075     5.548747
1999.625348
std     416.476890   3626.964585    18.599988     3.354085
1.540722
min       1.000000   4350.000000     1.000000     1.000000
1998.000000
25%     361.750000   8450.000000    44.000000     3.000000
1998.000000
50%     721.500000   9900.000000    61.000000     5.000000
1999.000000
75%    1081.250000  11950.000000    70.000000     8.000000
2001.000000
max    1442.000000  32500.000000    80.000000    12.000000
2004.000000

                  KM           HP    Met_Color    Automatic
cc  ...  \
count    1436.000000  1436.000000  1436.000000  1436.000000
1436.00000  ...
mean    68533.259749   101.502089     0.674791     0.055710
1576.85585  ...
std     37506.448872    14.981080     0.468616     0.229441
424.38677  ...
min         1.000000    69.000000     0.000000     0.000000
1300.00000  ...
25%     43000.000000    90.000000     0.000000     0.000000
1400.00000  ...
```

```
50%     63389.500000   110.000000     1.000000     0.000000
1600.00000  ...
75%     87020.750000   110.000000     1.000000     0.000000
1600.00000  ...
max    243000.000000   192.000000     1.000000     1.000000
16000.00000  ...

       Central_Lock  Powered_Windows  Power_Steering        Radio  \
count   1436.000000      1436.000000     1436.000000  1436.000000
mean       0.580084         0.561978        0.977716     0.146240
std        0.493717         0.496317        0.147657     0.353469
min        0.000000         0.000000        0.000000     0.000000
25%        0.000000         0.000000        1.000000     0.000000
50%        1.000000         1.000000        1.000000     0.000000
75%        1.000000         1.000000        1.000000     0.000000
max        1.000000         1.000000        1.000000     1.000000

         Mistlamps  Sport_Model  Backseat_Divider  Metallic_Rim  \
count  1436.000000  1436.000000       1436.000000   1436.000000
mean      0.256964     0.300139          0.770195      0.204735
std       0.437111     0.458478          0.420854      0.403649
min       0.000000     0.000000          0.000000      0.000000
25%       0.000000     0.000000          1.000000      0.000000
50%       0.000000     0.000000          1.000000      0.000000
75%       1.000000     1.000000          1.000000      0.000000
max       1.000000     1.000000          1.000000      1.000000

       Radio_cassette      Tow_Bar
count     1436.000000  1436.000000
mean         0.145543     0.277855
std          0.352770     0.448098
min          0.000000     0.000000
25%          0.000000     0.000000
50%          0.000000     0.000000
75%          0.000000     1.000000
max          1.000000     1.000000

[8 rows x 35 columns]
```

```
toyota_data.isnull().sum()
```

```
Id                    0
Model                 0
Price                 0
Age_08_04             0
Mfg_Month             0
Mfg_Year              0
KM                    0
Fuel_Type             0
HP                    0
```

```
Met_Color            0
Color                0
Automatic            0
cc                   0
Doors                0
Cylinders            0
Gears                0
Quarterly_Tax        0
Weight               0
Mfr_Guarantee        0
BOVAG_Guarantee      0
Guarantee_Period     0
ABS                  0
Airbag_1             0
Airbag_2             0
Airco                0
Automatic_airco      0
Boardcomputer        0
CD_Player            0
Central_Lock         0
Powered_Windows      0
Power_Steering       0
Radio                0
Mistlamps            0
Sport_Model          0
Backseat_Divider     0
Metallic_Rim         0
Radio_cassette       0
Tow_Bar              0
dtype: int64

 #Convert columns to appropriate data types if necessary
# Example: Convert 'Mfg_Month' and 'Mfg_Year' to string if needed for 
further processing
toyota_data['Mfg_Month'] = toyota_data['Mfg_Month'].astype(str)
toyota_data['Mfg_Year'] = toyota_data['Mfg_Year'].astype(str)

# If there are categorical variables, encode them
# For example, encoding 'Fuel_Type'
toyota_data['Fuel_Type'] = toyota_data['Fuel_Type'].astype('category')
toyota_data['Fuel_Type'] = toyota_data['Fuel_Type'].cat.codes

# Encoding 'Model' as it might have multiple categories
toyota_data['Model'] = toyota_data['Model'].astype('category')
toyota_data['Model_Code'] = toyota_data['Model'].cat.codes

# Check for duplicates and remove if any
print(f"Number of duplicate rows: {toyota_data.duplicated().sum()}")
toyota_data = toyota_data.drop_duplicates()
```

```
Number of duplicate rows: 0
```

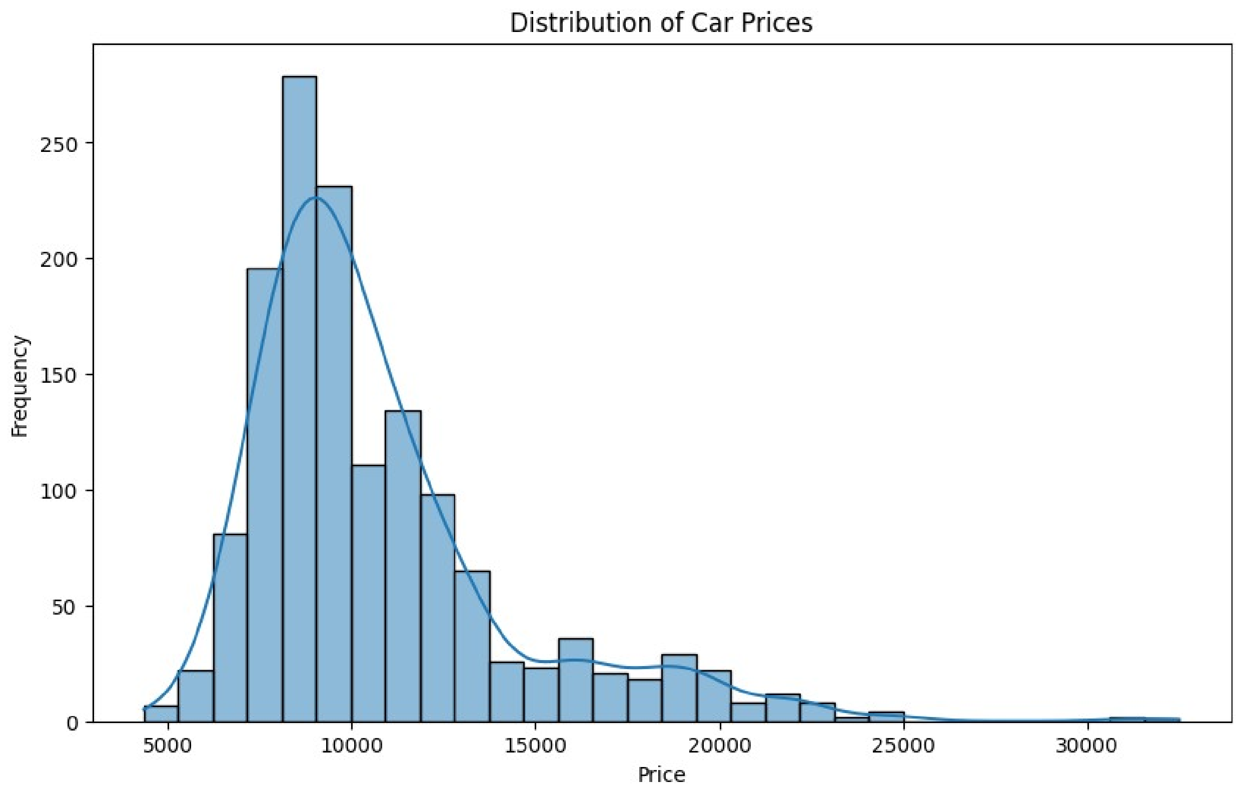
```python
# Basic statistics and data structure
print(toyota_data.describe(include='all'))
```

```
                 Id                                              Model  \
count   1436.000000                                               1436   
unique          NaN                                                372   
top             NaN  TOYOTA Corolla 1.6 16V HATCHB LINEA TERRA 2/3-...   
freq            NaN                                                107   
mean     721.555014                                                NaN   
std      416.476890                                                NaN   
min        1.000000                                                NaN   
25%      361.750000                                                NaN   
50%      721.500000                                                NaN   
75%     1081.250000                                                NaN   
max     1442.000000                                                NaN   


                Price    Age_08_04 Mfg_Month Mfg_Year             KM  \
count     1436.000000  1436.000000      1436     1436    1436.000000   
unique            NaN          NaN        12        7            NaN   
top               NaN          NaN         1     1999            NaN   
freq              NaN          NaN       207      441            NaN   
mean     10730.824513    55.947075       NaN      NaN   68533.259749   
std       3626.964585    18.599988       NaN      NaN   37506.448872   
min       4350.000000     1.000000       NaN      NaN       1.000000   
25%       8450.000000    44.000000       NaN      NaN   43000.000000   
50%       9900.000000    61.000000       NaN      NaN   63389.500000   
75%      11950.000000    70.000000       NaN      NaN   87020.750000   
max      32500.000000    80.000000       NaN      NaN  243000.000000   

          Fuel_Type           HP    Met_Color  ... Powered_Windows  \
count   1436.000000  1436.000000  1436.000000  ...     1436.000000   
unique          NaN          NaN          NaN  ...             NaN   
top             NaN          NaN          NaN  ...             NaN   
freq            NaN          NaN          NaN  ...             NaN   
mean       1.868384   101.502089     0.674791  ...        0.561978   
std        0.371572    14.981080     0.468616  ...        0.496317   
min        0.000000    69.000000     0.000000  ...        0.000000   
```

```
25%          2.000000    90.000000     0.000000  ...        0.000000
50%          2.000000   110.000000     1.000000  ...        1.000000
75%          2.000000   110.000000     1.000000  ...        1.000000
max          2.000000   192.000000     1.000000  ...        1.000000

        Power_Steering        Radio    Mistlamps  Sport_Model  \
count      1436.000000  1436.000000  1436.000000  1436.000000
unique             NaN          NaN          NaN          NaN
top                NaN          NaN          NaN          NaN
freq               NaN          NaN          NaN          NaN
mean          0.977716     0.146240     0.256964     0.300139
std           0.147657     0.353469     0.437111     0.458478
min           0.000000     0.000000     0.000000     0.000000
25%           1.000000     0.000000     0.000000     0.000000
50%           1.000000     0.000000     0.000000     0.000000
75%           1.000000     0.000000     1.000000     1.000000
max           1.000000     1.000000     1.000000     1.000000

        Backseat_Divider  Metallic_Rim  Radio_cassette      Tow_Bar  \
count        1436.000000   1436.000000     1436.000000  1436.000000
unique               NaN           NaN             NaN          NaN
top                  NaN           NaN             NaN          NaN
freq                 NaN           NaN             NaN          NaN
mean            0.770195      0.204735        0.145543     0.277855
std             0.420854      0.403649        0.352770     0.448098
min             0.000000      0.000000        0.000000     0.000000
25%             1.000000      0.000000        0.000000     0.000000
50%             1.000000      0.000000        0.000000     0.000000
75%             1.000000      0.000000        0.000000     1.000000
max             1.000000      1.000000        1.000000     1.000000

         Model_Code
count   1436.000000
unique          NaN
top             NaN
freq            NaN
mean     128.098886
std       97.908383
min        0.000000
25%       70.750000
50%       93.000000
75%      181.000000
max      371.000000

[11 rows x 39 columns]
```

Exploratory Data Analysis (EDA):

```
import seaborn as sns
import matplotlib.pyplot as plt
# Visualizing the distributions
plt.figure(figsize=(10, 6))
sns.histplot(toyota_data['Price'], bins=30, kde=True)
plt.title('Distribution of Car Prices')
plt.xlabel('Price')
plt.ylabel('Frequency')
plt.show()
```



The histogram you provided shows the distribution of car prices. Here are some insights:

Most Common Price Range:

The majority of cars are priced between $10,000 and $15,000, as indicated by the peak of the histogram.

Right-Skewed Distribution:

The distribution is right-skewed, meaning there are fewer cars with higher prices and more cars with lower prices.

Price Spread:

Car prices range from $0 to $30,000, with a significant drop in frequency as prices increase beyond $15,000.

Market Implications:

This distribution suggests that the market has a higher demand for more affordable cars, with fewer high-end vehicles.

```
# Exploring relationships between features and the target variable (Price)
plt.figure(figsize=(10, 6))
sns.scatterplot(data=toyota_data, x='Age_08_04', y='Price')
plt.title('Price vs. Age')
plt.xlabel('Age (months)')
plt.ylabel('Price')
plt.show()
```

The scatter plot you provided shows the relationship between car price and car age. Here are some insights:

Negative Correlation:

There is a clear negative correlation between car price and car age. As the age of the car increases, its price tends to decrease.

Depreciation Trend:

This trend indicates that cars lose value over time, which is a common phenomenon in the automotive market.

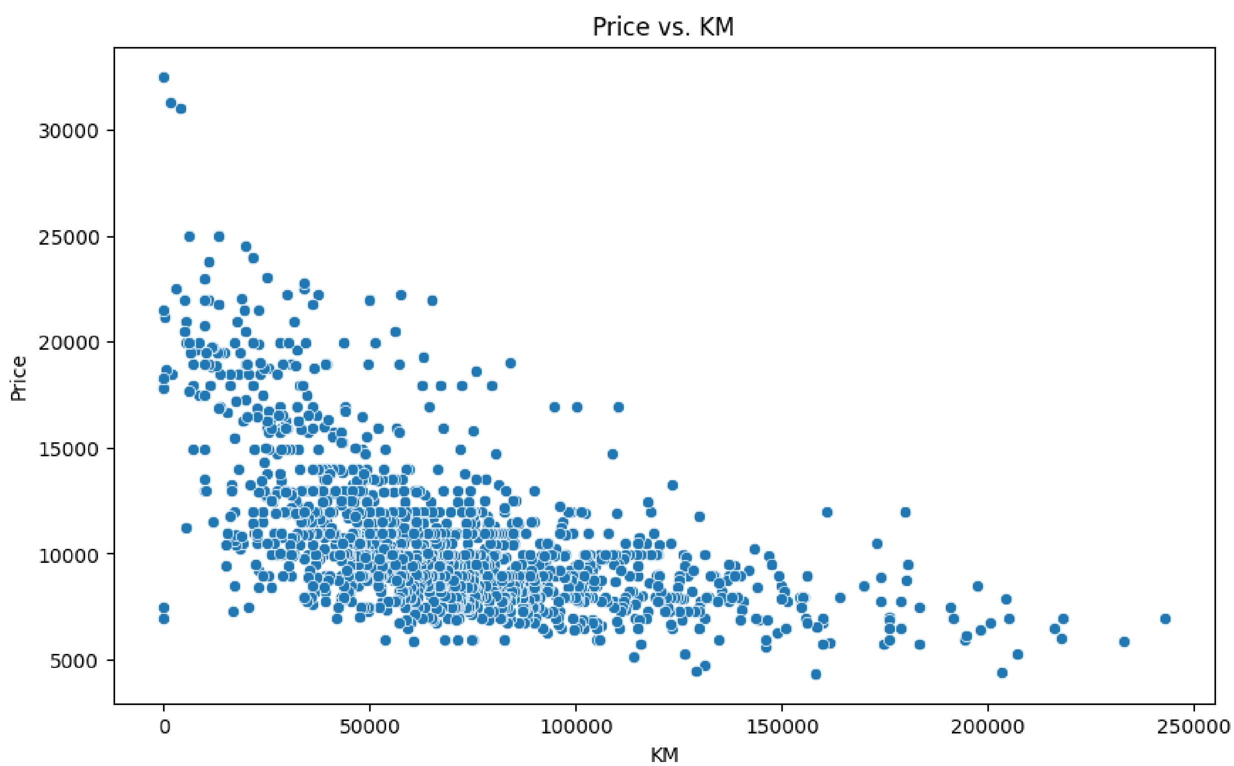
Price Range:

The prices range from approximately $0 to $35,000, with newer cars generally being more expensive.

Age Range:

The ages of the cars range from 0 to 80 months.

```
plt.figure(figsize=(10, 6))
sns.scatterplot(data=toyota_data, x='KM', y='Price')
plt.title('Price vs. KM')
plt.xlabel('KM')
plt.ylabel('Price')
plt.show()
```



The scatter plot you provided shows the relationship between car price and kilometers driven. Here are some insights:

Negative Correlation:

There is a clear negative correlation between car price and kilometers driven. As the kilometers increase, the price tends to decrease.

Depreciation with Usage:

This trend indicates that cars lose value as they accumulate more kilometers, which is typical in the automotive market.

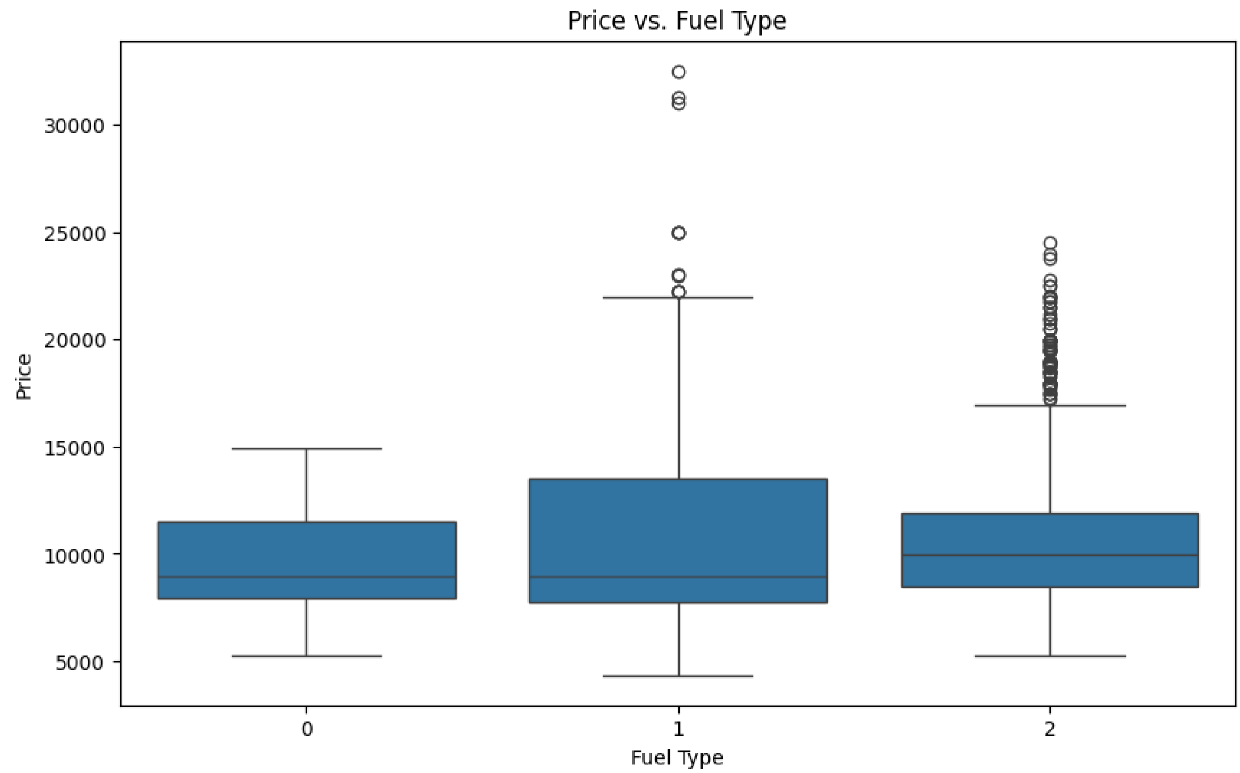
Price Range:

The prices range from approximately $0 to $30,000, with cars that have fewer kilometers generally being more expensive.

Kilometer Range:

The kilometers driven range from 0 to 250,000.

```python
# Box plot of Price vs Fuel Type
plt.figure(figsize=(10, 6))
sns.boxplot(data=toyota_data, x='Fuel_Type', y='Price')
plt.title('Price vs. Fuel Type')
plt.xlabel('Fuel Type')
plt.ylabel('Price')
plt.show()
```



The box plot you provided shows the relationship between car price and fuel type. Here are some insights:

CNG (0):

Cars running on CNG generally fall in the lower price bracket compared to petrol and diesel cars. The prices have a wide range but are mostly lower.

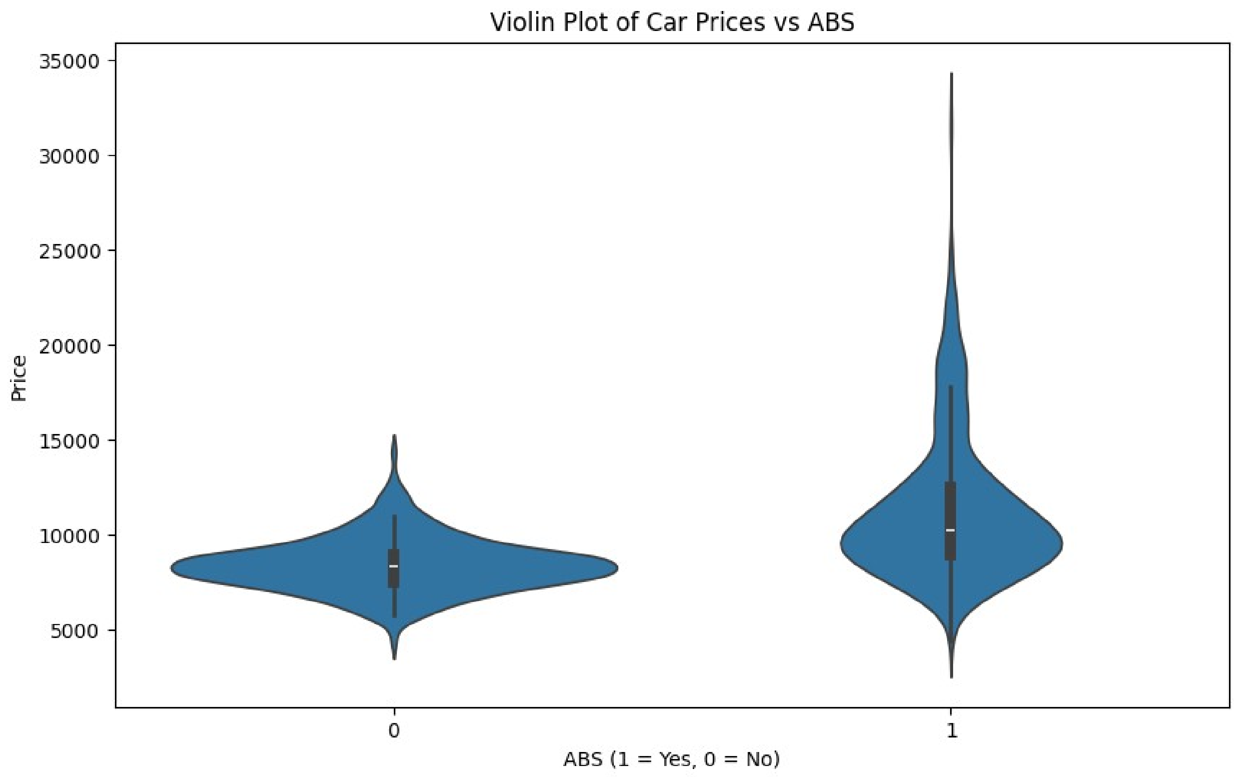
Petrol (1):

Petrol cars have a higher median price than CNG cars, with less variability in their prices.

Diesel (2):

Diesel cars show a similar median price to petrol cars but with more outliers, indicating some diesel cars are priced significantly higher than the average.

```python
# Violin plot of Price vs ABS
plt.figure(figsize=(10, 6))
sns.violinplot(data=toyota_data, x='ABS', y='Price')
plt.title('Violin Plot of Car Prices vs ABS')
plt.xlabel('ABS (1 = Yes, 0 = No)')
plt.ylabel('Price')
plt.show()
```



The violin plot you provided shows the relationship between car price and the presence of ABS (Anti-lock Braking System). Here are some insights:

Higher Prices with ABS:

Cars equipped with ABS (labeled as ‘1’) tend to have a higher price range compared to those without ABS (labeled as ‘0’).

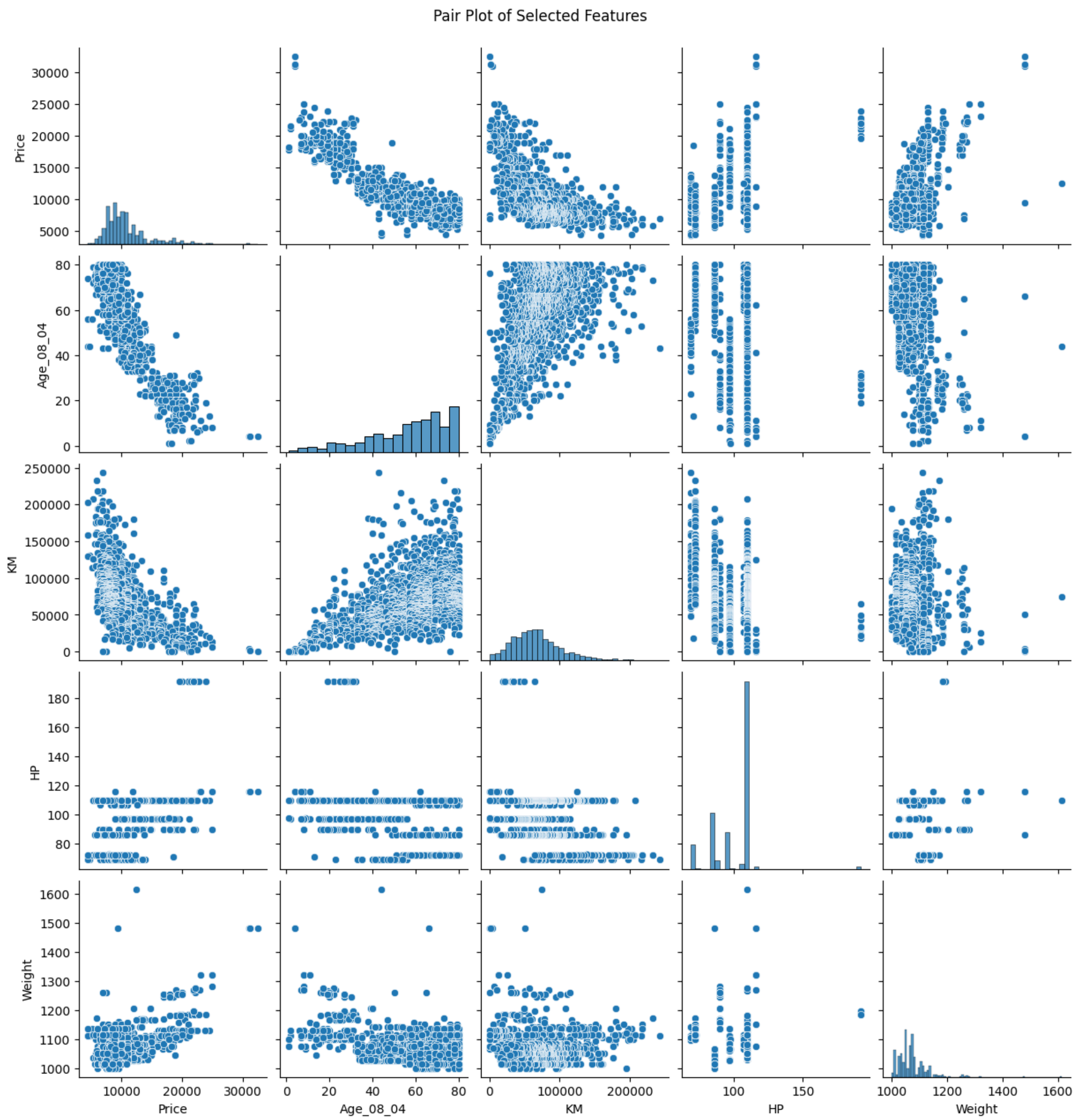
Wider Distribution:

The distribution for cars with ABS is wider and higher, indicating a broader range of prices and generally higher values.

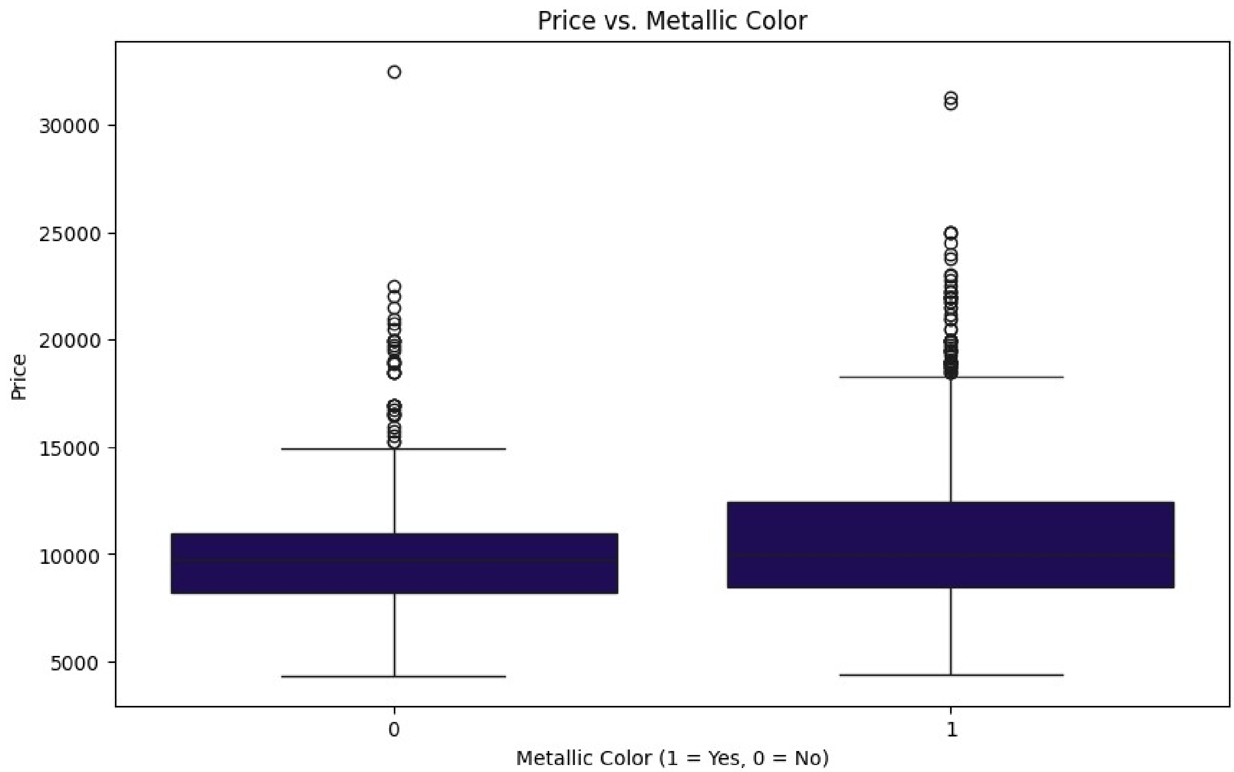
Price Influence:

The presence of ABS seems to positively influence car prices, suggesting that this feature is valued in the market.

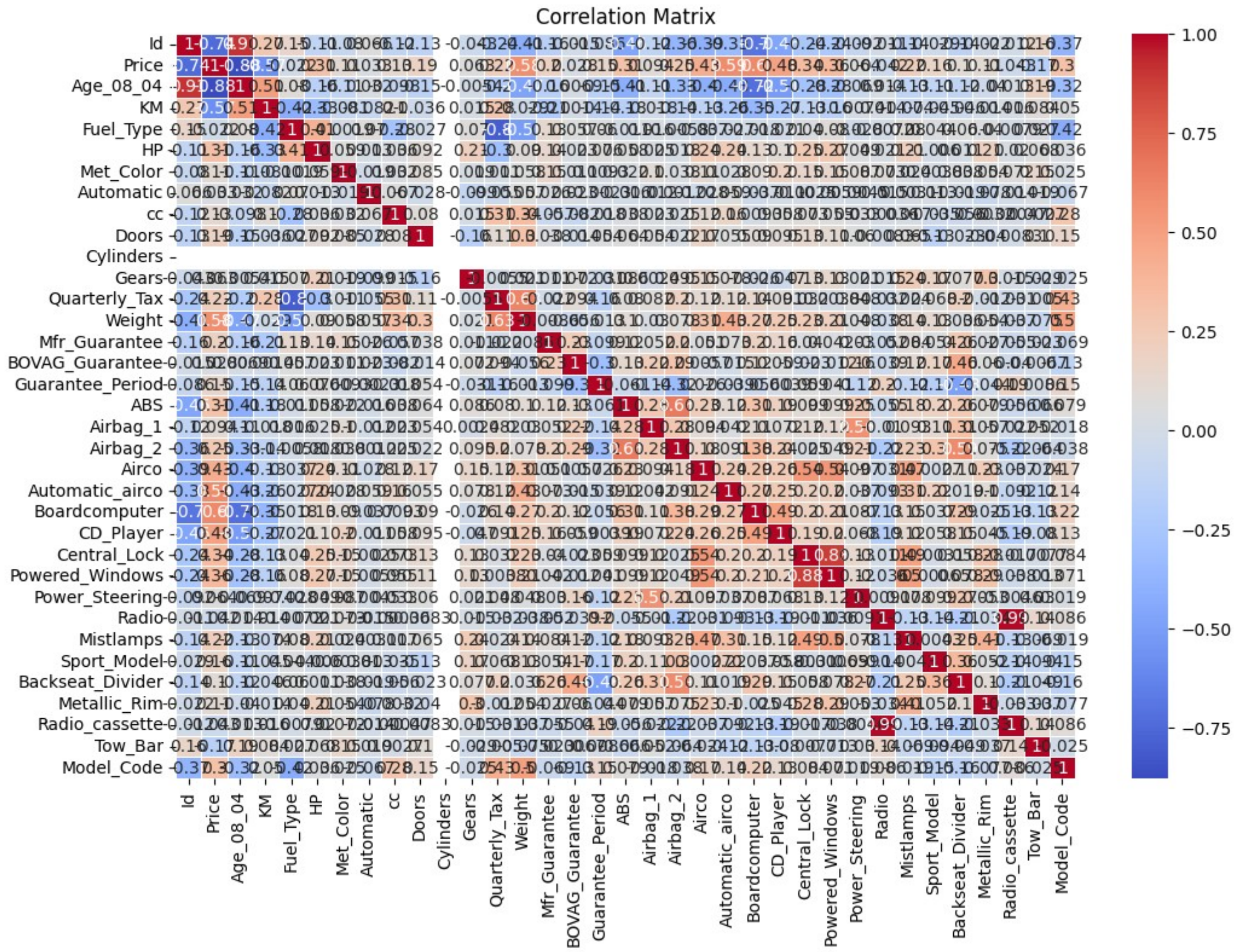
```
# Pair plot for a few features
sns.pairplot(toyota_data[['Price', 'Age_08_04', 'KM', 'HP',
'Weight']])
plt.suptitle('Pair Plot of Selected Features', y=1.02)
plt.show()
```




```
# Box plot of Price vs Metallic Color
plt.figure(figsize=(10, 6))
sns.boxplot(data=toyota_data, x='Met_Color',
y='Price',color='#180161')
plt.title('Price vs. Metallic Color')
plt.xlabel('Metallic Color (1 = Yes, 0 = No)')
plt.ylabel('Price')
plt.show()
```




```
# Select only the numeric columns
numeric_data = toyota_data.select_dtypes(include=[np.number])

# Correlation analysis
plt.figure(figsize=(12, 8))
correlation_matrix = numeric_data.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm',
linewidths=0.5)
plt.title('Correlation Matrix')
plt.show()
```



Positive Correlations (Red Cells):

When two variables have a positive correlation, an increase in one tends to be associated with an increase in the other. For example, if fuel efficiency (miles per gallon) increases, the car's weight might decrease.

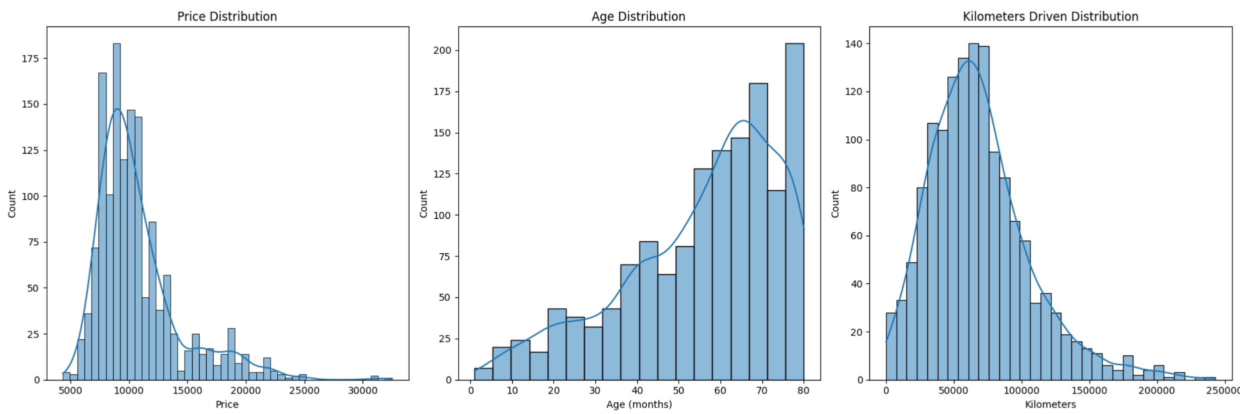
Negative Correlations (Blue Cells):

Conversely, negative correlations indicate that as one variable increases, the other tends to decrease. For instance, higher horsepower might be associated with lower fuel efficiency.

Strength of Correlation:

A coefficient close to 1 (or -1) indicates a strong positive (or negative) correlation. A coefficient around 0 suggests little to no correlation.

```
# Create subplots
fig, axes = plt.subplots(1, 3, figsize=(18, 6))
# Plot Price distribution
sns.histplot(toyota_data['Price'], kde=True, ax=axes[0])
axes[0].set_title('Price Distribution')
axes[0].set_xlabel('Price')
# Plot Age distribution using 'Age_08_04' column
sns.histplot(toyota_data['Age_08_04'], kde=True, ax=axes[1])
axes[1].set_title('Age Distribution')
axes[1].set_xlabel('Age (months)')
# Plot KM distribution
sns.histplot(toyota_data['KM'], kde=True, ax=axes[2])
axes[2].set_title('Kilometers Driven Distribution')
axes[2].set_xlabel('Kilometers')

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



Car Price Distribution:

The distribution of car prices can vary significantly based on factors like make, model, and market conditions. On average, passenger vehicles emit about 4.6 metric tons of carbon dioxide (CO_2) per year1. The most common price range for cars depends on the local market, but typically, there's a cluster of cars in the lower price range, with fewer in the higher price range. Keep in mind that this distribution can change over time due to economic factors, demand, and supply.

Car Age Distribution:

The average age of passenger cars varies by region and year. In the European Union, passenger cars are now on average 12 years old2. In the United States, the average age of cars has been increasing. As of recent data: Most vehicles on the road are model years 2015 to 2019, making up approximately 26% of all vehicles. About 23% of passenger cars are 20 years old or older. Model years 2010-2015 and 2005-2010 account for about 19% and 20%, respectively3.

Car Kilometer Distribution:

The number of kilometers driven by cars can vary widely based on usage patterns and individual driving habits. The average passenger vehicle emits about 400 grams of CO_2 per mile1. Cars with higher mileage tend to have more wear and tear, affecting their value and performance. Regular maintenance and proper care can extend a car's lifespan and maintain its value.

```
# Count plot for Automatic
plt.figure(figsize=(10, 6))
sns.countplot(data=toyota_data, x='Automatic')
plt.title('Count Plot of Transmission Type')
plt.xlabel('Automatic (1 = Yes, 0 = No)')
plt.ylabel('Count')
plt.show()
```