In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import pymysql
```

In [2]:
```
# Load the dataset
file_path = "C:/Users/Komal Bhati/Desktop/New folder/used_cars.csv"
data = pd.read_csv(file_path)
```

In [4]:
```
import pymysql

# Define your MySQL configuration
db_config = {
    'host': 'localhost',
    'user': 'root',
    'password': '1881arpit',
    'database': 'used_cars_db'
}

# Create mileage column
data['mileage'] = (data['city_mileage'] + data['highway_mileage']) / 2

# Establish MySQL Connection
try:
    connection = pymysql.connect(**db_config)
    cursor = connection.cursor()

    # Create database if not exists
    cursor.execute("CREATE DATABASE IF NOT EXISTS used_cars_db")
    cursor.execute("USE used_cars_db")

    # Create table
    create_table_query = """
    CREATE TABLE IF NOT EXISTS car_details (
        id INT AUTO_INCREMENT PRIMARY KEY,
        brand VARCHAR(50),
        model VARCHAR(50),
        year INT,
        price FLOAT,
        mileage FLOAT
    )"""
    cursor.execute(create_table_query)

    # Insert data
    for _, row in data.iterrows():
        insert_query = """
        INSERT INTO car_details (brand, model, year, price, mileage)
        VALUES (%s, %s, %s, %s, %s)
        """
        cursor.execute(insert_query, (row['brand'], row['model'], row['year'],
                                       row['price'], row['mileage']))
    connection.commit()
    print("Data inserted successfully")

except pymysql.MySQLError as e:
    print(f"Error: {e}")

finally:
```

```
    if connection:
        cursor.close()
        connection.close()
```

```
Data inserted successfully
```

In [5]:
```
# 1. Basic Information
print('--- Dataset Info-')
data.info()

# 2. Checking for null values
print('\n--- Missing Values ---')
print(data.isnull().sum())
```

```
--- Dataset Info ---
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 52 entries, 0 to 51
Data columns (total 28 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   id                    52 non-null     int64  
 1   brand                 52 non-null     object 
 2   model                 52 non-null     object 
 3   year                  52 non-null     int64  
 4   miles                 52 non-null     int64  
 5   city_mileage          52 non-null     int64  
 6   highway_mileage       52 non-null     int64  
 7   horsepower            52 non-null     int64  
 8   torque                52 non-null     int64  
 9   engine_capacity_litre  52 non-null     float64
 10  fuel_capacity         52 non-null     float64
 11  num_cylinder          52 non-null     int64  
 12  num_seat              52 non-null     int64  
 13  num_owners            52 non-null     int64  
 14  price                 52 non-null     int64  
 15  link                  52 non-null     object 
 16  condition             1 non-null      float64
 17  type                  52 non-null     object 
 18  doors                 52 non-null     int64  
 19  wheel_drive           52 non-null     int64  
 20  engine_type           52 non-null     object 
 21  speed_levels          51 non-null     float64
 22  front_headroom        52 non-null     float64
 23  front_legroom         52 non-null     float64
 24  rear_headroom         52 non-null     float64
 25  rear_legroom          52 non-null     float64
 26  service_records       52 non-null     int64  
 27  mileage               52 non-null     float64
dtypes: float64(9), int64(14), object(5)
memory usage: 11.5+ KB

--- Missing Values ---
id                        0
brand                     0
model                     0
year                      0
miles                     0
city_mileage              0
highway_mileage           0
horsepower                0
torque                    0
engine_capacity_litre     0
fuel_capacity             0
num_cylinder              0
num_seat                  0
num_owners                0
price                     0
link                      0
condition                51
type                      0
doors                     0
wheel_drive               0
engine_type               0
speed_levels              1
```

```
front_headroom          0
front_legroom           0
rear_headroom           0
rear_legroom            0
service_records         0
mileage                 0
dtype: int64
```

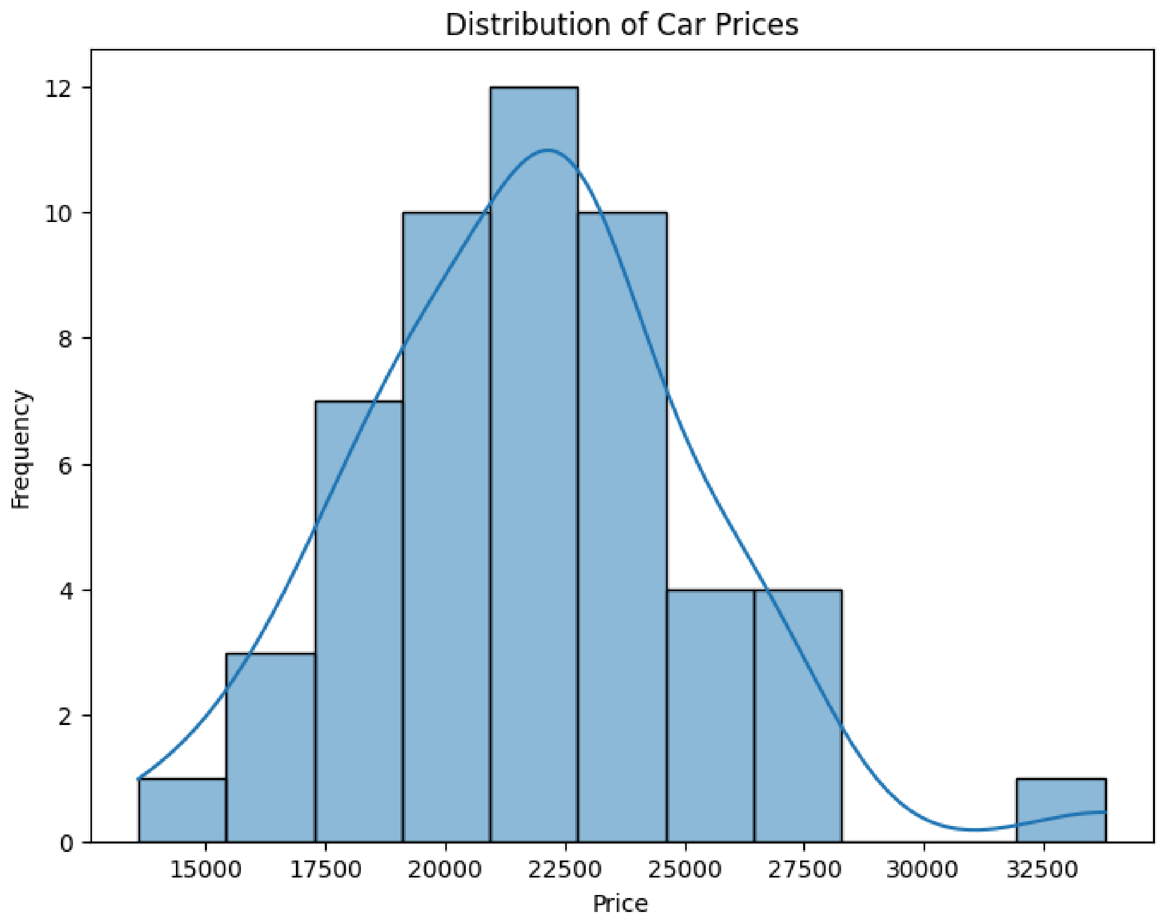
In [6]:
```
# 3. Descriptive Statistics
print('\n--- Descriptive Statistics ---')
print(data.describe())

# 4. Unique values in each column
print('\n--- Unique Values in Each Column ---')
print(data.nunique())
```

```
--- Descriptive Statistics ---
              id         year          miles  city_mileage  highway_mileage  \
count  52.000000    52.000000      52.000000     52.000000        52.000000   
mean   31.365385  2018.673077   33901.250000     29.038462        37.423077   
std    15.378912     1.396370   22700.646139      5.947356         4.136619   
min     3.000000  2014.000000    5000.000000     17.000000        24.000000   
25%    18.750000  2018.000000   16454.750000     26.500000        35.750000   
50%    31.500000  2019.000000   27448.500000     30.000000        38.000000   
75%    44.250000  2019.250000   42442.000000     30.000000        40.000000   
max    57.000000  2022.000000   97027.000000     55.000000        49.000000   

       horsepower      torque  engine_capacity_litre  fuel_capacity  \
count   52.000000   52.000000              52.000000      52.000000   
mean   176.865385  177.923077               1.832692      14.232692   
std     33.793507   47.256433               0.379743       2.603973   
min    143.000000   99.000000               1.400000       7.000000   
25%    152.000000  138.000000               1.500000      12.400000   
50%    159.500000  181.000000               2.000000      13.600000   
75%    192.000000  192.000000               2.000000      15.050000   
max    288.000000  294.000000               3.500000      19.000000   

       num_cylinder  ...  condition  doors  wheel_drive  speed_levels  \
count     52.000000  ...        1.0   52.0    52.000000     51.000000   
mean       4.038462  ...        4.0    4.0     2.153846      6.313725   
std        0.277350  ...        NaN    0.0     0.538138      0.761320   
min        4.000000  ...        4.0    4.0     2.000000      6.000000   
25%        4.000000  ...        4.0    4.0     2.000000      6.000000   
50%        4.000000  ...        4.0    4.0     2.000000      6.000000   
75%        4.000000  ...        4.0    4.0     2.000000      6.000000   
max        6.000000  ...        4.0    4.0     4.000000      9.000000   

       front_headroom  front_legroom  rear_headroom  rear_legroom  \
count       52.000000      52.000000      52.000000     52.000000   
mean        38.680769      42.467308      37.238462     37.632692   
std          0.919087       1.045198       0.444202      1.604249   
min         37.500000      41.100000      35.800000     33.200000   
25%         37.725000      42.200000      37.075000     37.175000   
50%         38.500000      42.300000      37.200000     37.400000   
75%         39.300000      42.400000      37.500000     38.300000   
max         40.400000      45.500000      38.000000     40.400000   

       service_records    mileage  
count        52.000000  52.000000  
mean          7.326923  33.230769  
std           4.714272   4.858247  
min           1.000000  20.500000  
25%           4.000000  30.875000  
50%           6.000000  34.000000  
75%          10.000000  35.000000  
max          26.000000  52.000000  

[8 rows x 23 columns]

--- Unique Values in Each Column ---
id                       52
brand                     6
model                    15
year                      8
miles                    50
city_mileage             15
```

```
highway_mileage          14
horsepower               20
torque                   17
engine_capacity_litre     7
fuel_capacity            13
num_cylinder              2
num_seat                  1
num_owners                3
price                    47
link                     51
condition                 1
type                      1
doors                     1
wheel_drive               2
engine_type               2
speed_levels              4
front_headroom           15
front_legroom            14
rear_headroom            11
rear_legroom             12
service_records          15
mileage                  22
dtype: int64
```

In [7]:

```
# 5. Distribution of target variable (if any)
if 'price' in data.columns:
    plt.figure(figsize=(8, 6))
    sns.histplot(data['price'], kde=True)
    plt.title('Distribution of Car Prices')
    plt.xlabel('Price')
    plt.ylabel('Frequency')
    plt.show()
```



In [8]:
```
import numpy as np

# Check for missing values
print("Missing values in each column:")
print(data.isnull().sum())

# Check for infinite values in numeric columns
numeric_data = data.select_dtypes(include=[np.number])
print("\nInfinite values in each numeric column:")
print(np.isinf(numeric_data).sum())
```

```
Missing values in each column:
id                       0
brand                    0
model                    0
year                     0
miles                    0
city_mileage             0
highway_mileage          0
horsepower               0
torque                   0
engine_capacity_litre    0
fuel_capacity            0
num_cylinder             0
num_seat                 0
num_owners               0
price                    0
link                     0
condition               51
type                     0
doors                    0
wheel_drive              0
engine_type              0
speed_levels             1
front_headroom           0
front_legroom            0
rear_headroom            0
rear_legroom             0
service_records          0
mileage                  0
dtype: int64

Infinite values in each numeric column:
id                      0
year                    0
miles                   0
city_mileage            0
highway_mileage         0
horsepower              0
torque                  0
engine_capacity_litre   0
fuel_capacity           0
num_cylinder            0
num_seat                0
num_owners              0
price                   0
condition               0
doors                   0
wheel_drive             0
speed_levels            0
front_headroom          0
front_legroom           0
rear_headroom           0
rear_legroom            0
service_records         0
mileage                 0
dtype: int64
```

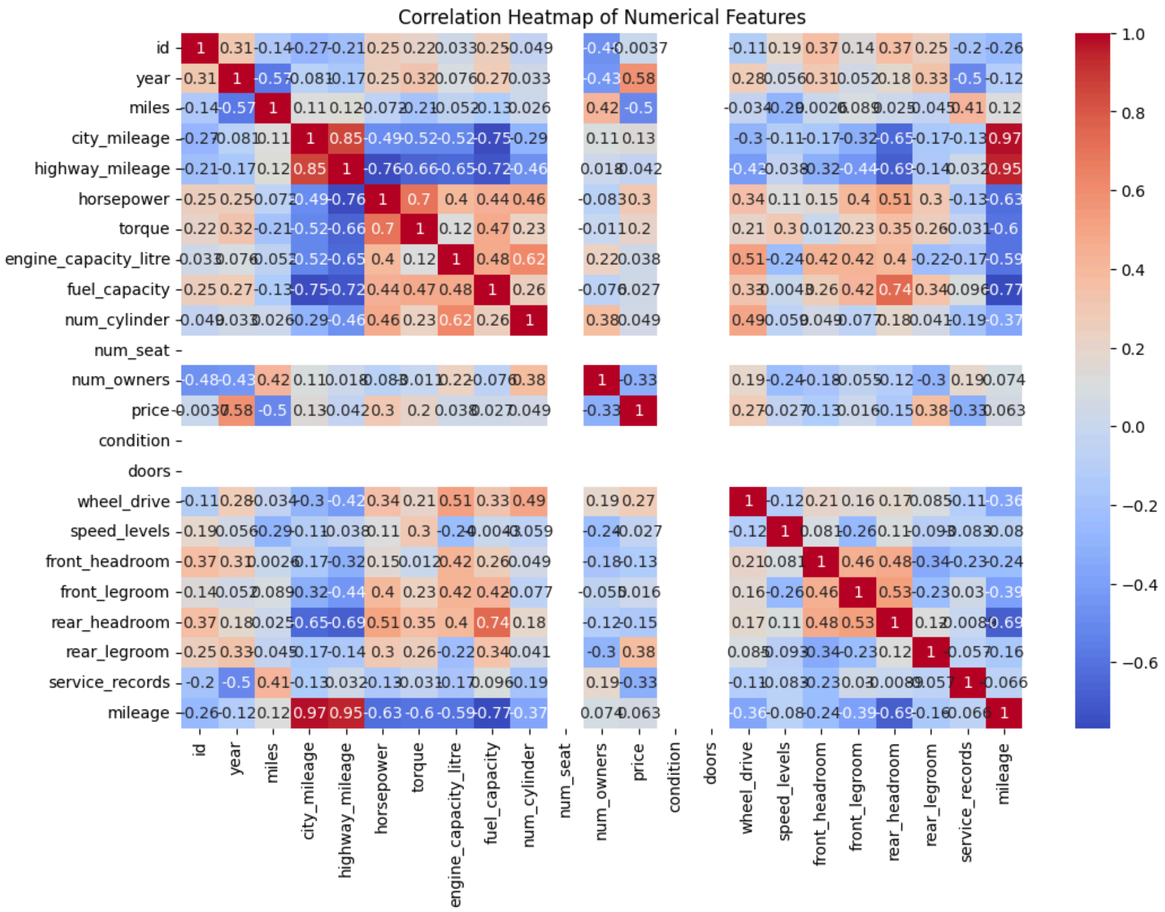
In [9]:
```
data_clean = data.replace([np.inf, -np.inf], np.nan).dropna()
```

In [10]:
```
# 6. Correlation Heatmap

corr_matrix = numeric_data.corr()
print(corr_matrix)

plt.figure(figsize=(12, 8))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap of Numerical Features')
plt.show()
```

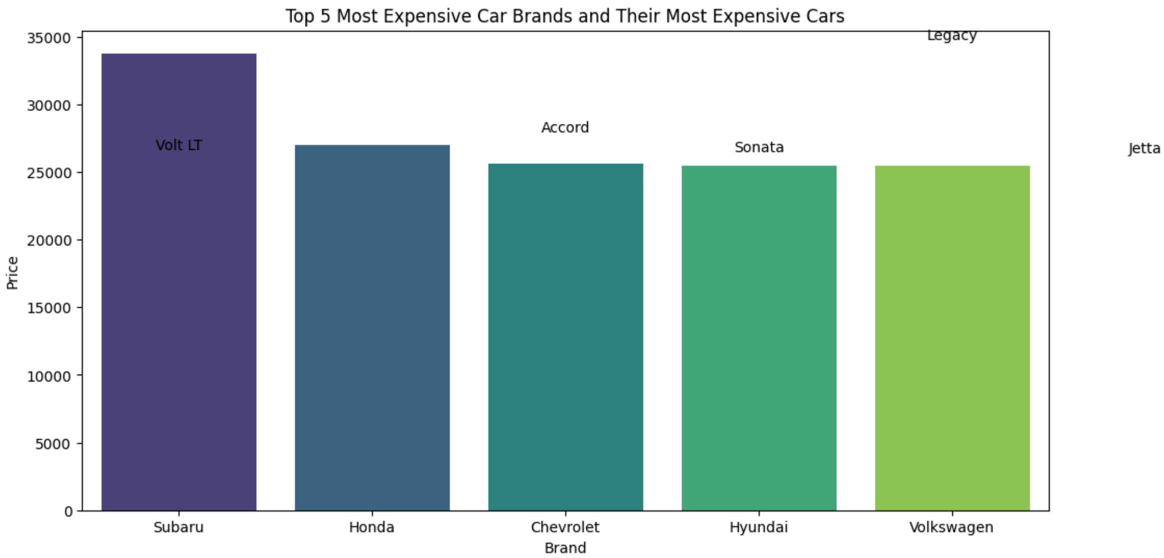
```
                            id      year     miles  city_mileage  \
id                    1.000000  0.314289 -0.140389     -0.274989
year                  0.314289  1.000000 -0.570330     -0.081093
miles                -0.140389 -0.570330  1.000000      0.109841
city_mileage         -0.274989 -0.081093  0.109841      1.000000
highway_mileage      -0.213916 -0.165681  0.122973      0.852120
horsepower            0.251407  0.251688 -0.072269     -0.493823
torque                0.217796  0.322014 -0.207391     -0.519817
engine_capacity_litre  0.033168  0.076018 -0.051521     -0.517143
fuel_capacity         0.245833  0.274241 -0.130851     -0.754301
num_cylinder         -0.049330  0.033104  0.026134     -0.286206
num_seat                   NaN       NaN       NaN           NaN
num_owners           -0.476279 -0.431124  0.415837      0.107537
price                -0.003663  0.581254 -0.500769      0.131767
condition                  NaN       NaN       NaN           NaN
doors                      NaN       NaN       NaN           NaN
wheel_drive          -0.106434  0.276995 -0.033973     -0.295957
speed_levels          0.187717  0.056380 -0.285384     -0.107265
front_headroom        0.367012  0.305153  0.002602     -0.173121
front_legroom         0.141772  0.051646  0.088820     -0.323113
rear_headroom         0.371326  0.175567  0.025208     -0.650003
rear_legroom          0.254622  0.334853 -0.045425     -0.169064
service_records      -0.195053 -0.501725  0.412596     -0.129836
mileage              -0.259388 -0.120172  0.119586      0.974863

                      highway_mileage  horsepower    torque  \
id                          -0.213916    0.251407  0.217796
year                        -0.165681    0.251688  0.322014
miles                        0.122973   -0.072269 -0.207391
city_mileage                 0.852120   -0.493823 -0.519817
highway_mileage              1.000000   -0.758562 -0.663951
horsepower                  -0.758562    1.000000  0.702836
torque                      -0.663951    0.702836  1.000000
engine_capacity_litre       -0.649320    0.399601  0.122738
fuel_capacity               -0.719426    0.438834  0.465158
num_cylinder                -0.458816    0.464994  0.227627
num_seat                          NaN         NaN       NaN
num_owners                   0.018155   -0.082548 -0.011073
price                       -0.041956    0.301800  0.199764
condition                         NaN         NaN       NaN
doors                             NaN         NaN       NaN
wheel_drive                 -0.417376    0.344031  0.213281
speed_levels                -0.037628    0.109859  0.298219
front_headroom              -0.321700    0.146378  0.011929
front_legroom               -0.441630    0.400069  0.227339
rear_headroom               -0.688770    0.509124  0.351548
rear_legroom                -0.138336    0.300567  0.255183
service_records              0.031982   -0.132643 -0.030602
mileage                      0.947304   -0.625207 -0.600839

                      engine_capacity_litre  fuel_capacity  num_cylinder  \
id                                 0.033168       0.245833     -0.049330
year                               0.076018       0.274241      0.033104
miles                             -0.051521      -0.130851      0.026134
city_mileage                      -0.517143      -0.754301     -0.286206
highway_mileage                   -0.649320      -0.719426     -0.458816
horsepower                         0.399601       0.438834      0.464994
torque                             0.122738       0.465158      0.227627
engine_capacity_litre              1.000000       0.475591      0.620809
fuel_capacity                      0.475591       1.000000      0.258862
```

```
num_cylinder                 0.620809     0.258862     1.000000
num_seat                          NaN          NaN          NaN
num_owners                   0.224878    -0.076055     0.375592
price                        0.038197     0.027480     0.049216
condition                         NaN          NaN          NaN
doors                             NaN          NaN          NaN
wheel_drive                  0.512227     0.329365     0.485071
speed_levels                -0.236593    -0.004260    -0.058857
front_headroom               0.420380     0.264980     0.049111
front_legroom                0.420684     0.417100    -0.076745
rear_headroom                0.402731     0.737816     0.178744
rear_legroom                -0.216792     0.343792     0.041187
service_records             -0.174761     0.095907    -0.189762
mileage                     -0.592974    -0.767981    -0.370516

                       ...  condition  doors  wheel_drive  speed_levels  \
id                     ...        NaN    NaN    -0.106434      0.187717
year                   ...        NaN    NaN     0.276995      0.056380
miles                  ...        NaN    NaN    -0.033973     -0.285384
city_mileage           ...        NaN    NaN    -0.295957     -0.107265
highway_mileage        ...        NaN    NaN    -0.417376     -0.037628
horsepower             ...        NaN    NaN     0.344031      0.109859
torque                 ...        NaN    NaN     0.213281      0.298219
engine_capacity_litre  ...        NaN    NaN     0.512227     -0.236593
fuel_capacity          ...        NaN    NaN     0.329365     -0.004260
num_cylinder           ...        NaN    NaN     0.485071     -0.058857
num_seat               ...        NaN    NaN          NaN           NaN
num_owners             ...        NaN    NaN     0.189074     -0.236658
price                  ...        NaN    NaN     0.268273     -0.026826
condition              ...        NaN    NaN          NaN           NaN
doors                  ...        NaN    NaN          NaN           NaN
wheel_drive            ...        NaN    NaN     1.000000     -0.121413
speed_levels           ...        NaN    NaN    -0.121413      1.000000
front_headroom         ...        NaN    NaN     0.212249      0.080521
front_legroom          ...        NaN    NaN     0.155533     -0.260503
rear_headroom          ...        NaN    NaN     0.171625      0.107920
rear_legroom           ...        NaN    NaN     0.084910     -0.092717
service_records        ...        NaN    NaN    -0.112962     -0.083293
mileage                ...        NaN    NaN    -0.358842     -0.079967

                       front_headroom  front_legroom  rear_headroom  \
id                           0.367012       0.141772       0.371326
year                         0.305153       0.051646       0.175567
miles                        0.002602       0.088820       0.025208
city_mileage                -0.173121      -0.323113      -0.650003
highway_mileage             -0.321700      -0.441630      -0.688770
horsepower                   0.146378       0.400069       0.509124
torque                       0.011929       0.227339       0.351548
engine_capacity_litre        0.420380       0.420684       0.402731
fuel_capacity                0.264980       0.417100       0.737816
num_cylinder                 0.049111      -0.076745       0.178744
num_seat                          NaN            NaN            NaN
num_owners                  -0.176602      -0.054584      -0.119983
price                       -0.129595       0.015819      -0.145403
condition                         NaN            NaN            NaN
doors                             NaN            NaN            NaN
wheel_drive                  0.212249       0.155533       0.171625
speed_levels                 0.080521      -0.260503       0.107920
front_headroom               1.000000       0.455325       0.477803
front_legroom                0.455325       1.000000       0.526027
```

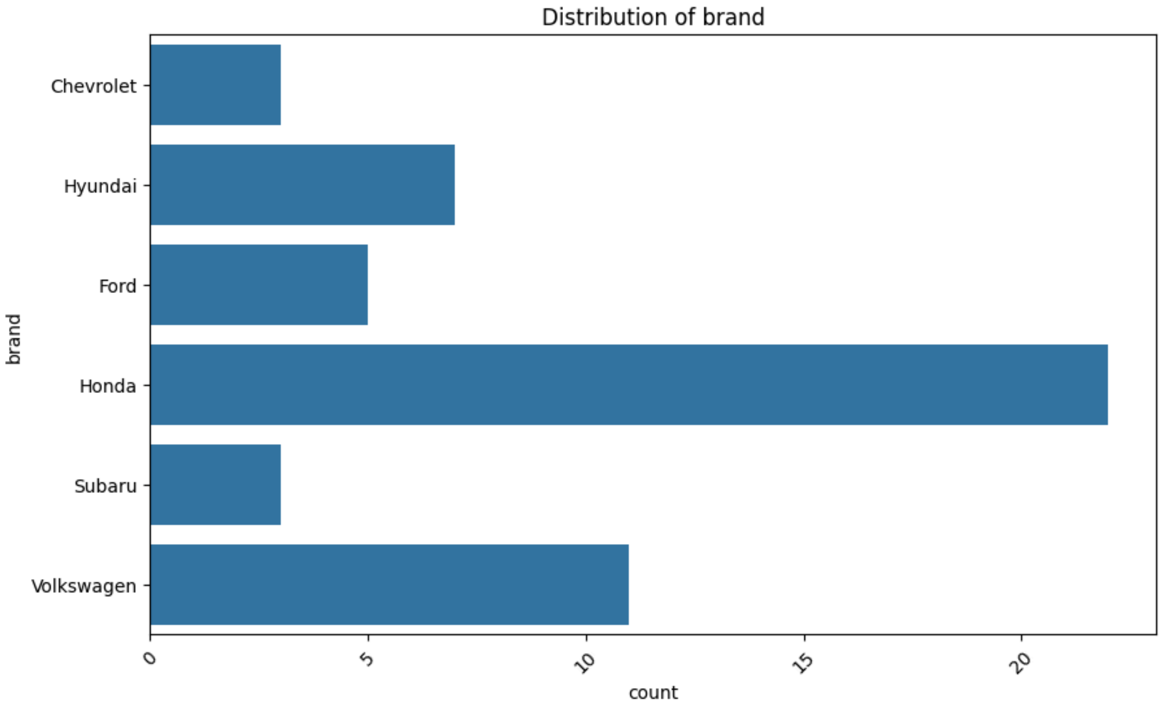
```
rear_headroom           0.477803     0.526027     1.000000
rear_legroom           -0.338809    -0.229369     0.120095
service_records        -0.232937     0.029669    -0.008931
mileage                -0.242924    -0.385790    -0.691091

                      rear_legroom  service_records   mileage
id                        0.254622        -0.195053 -0.259388
year                      0.334853        -0.501725 -0.120172
miles                    -0.045425         0.412596  0.119586
city_mileage             -0.169064        -0.129836  0.974863
highway_mileage          -0.138336         0.031982  0.947304
horsepower                0.300567        -0.132643 -0.625207
torque                    0.255183        -0.030602 -0.600839
engine_capacity_litre    -0.216792        -0.174761 -0.592974
fuel_capacity             0.343792         0.095907 -0.767981
num_cylinder              0.041187        -0.189762 -0.370516
num_seat                       NaN              NaN       NaN
num_owners               -0.304058         0.187823  0.073551
price                     0.377287        -0.327306  0.062791
condition                      NaN              NaN       NaN
doors                          NaN              NaN       NaN
wheel_drive               0.084910        -0.112962 -0.358842
speed_levels             -0.092717        -0.083293 -0.079967
front_headroom           -0.338809        -0.232937 -0.242924
front_legroom            -0.229369         0.029669 -0.385790
rear_headroom             0.120095        -0.008931 -0.691091
rear_legroom              1.000000        -0.057183 -0.162376
service_records          -0.057183         1.000000 -0.065856
mileage                  -0.162376        -0.065856  1.000000

[23 rows x 23 columns]
```

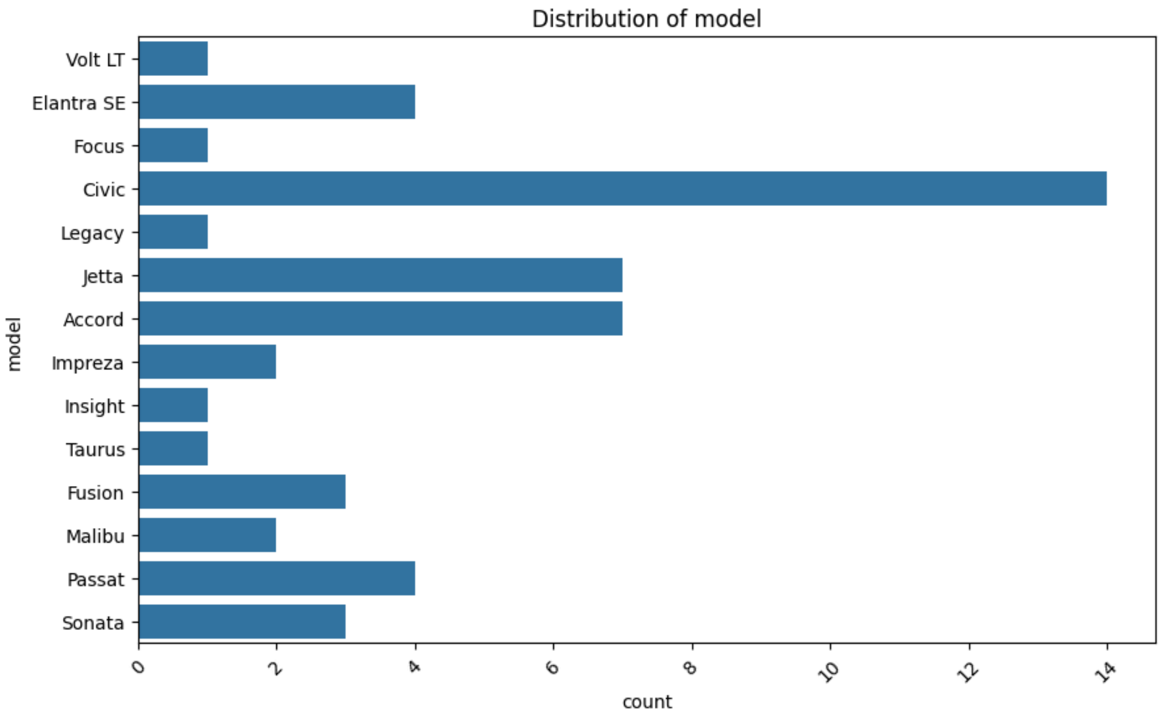


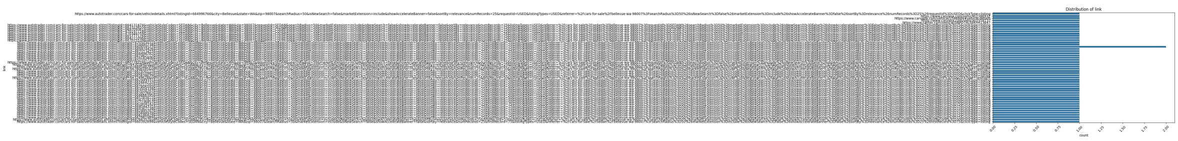
In [49]:
```
# Plotting Top 5 Most Expensive Car Brands and Their Most Expensive Cars
plt.figure(figsize=(12, 6))
sns.barplot(x='brand', y='price', data=top_brands, palette='viridis', hue='brand

# Adding labels for car models on top of bars
for index, row in top_brands.iterrows():
    plt.text(index, row['price'] + 1000, row['model'], color='black', ha='center

plt.title('Top 5 Most Expensive Car Brands and Their Most Expensive Cars')
plt.xlabel('Brand')
plt.ylabel('Price')
plt.show()
```

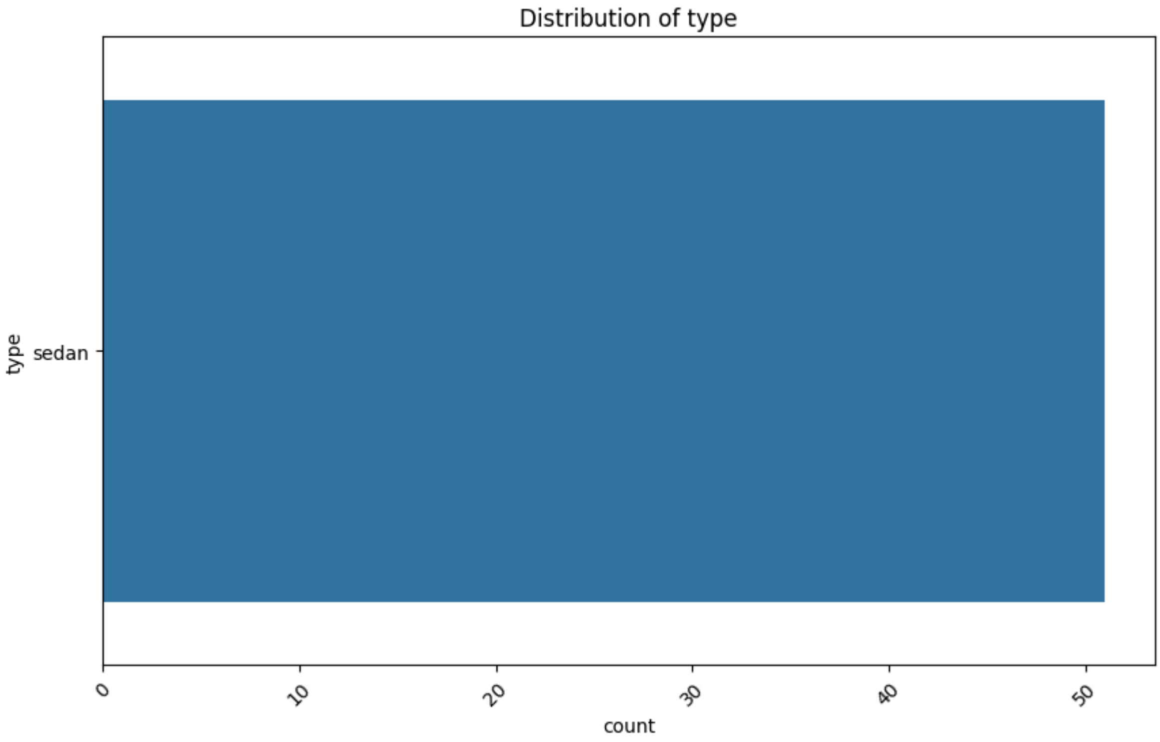


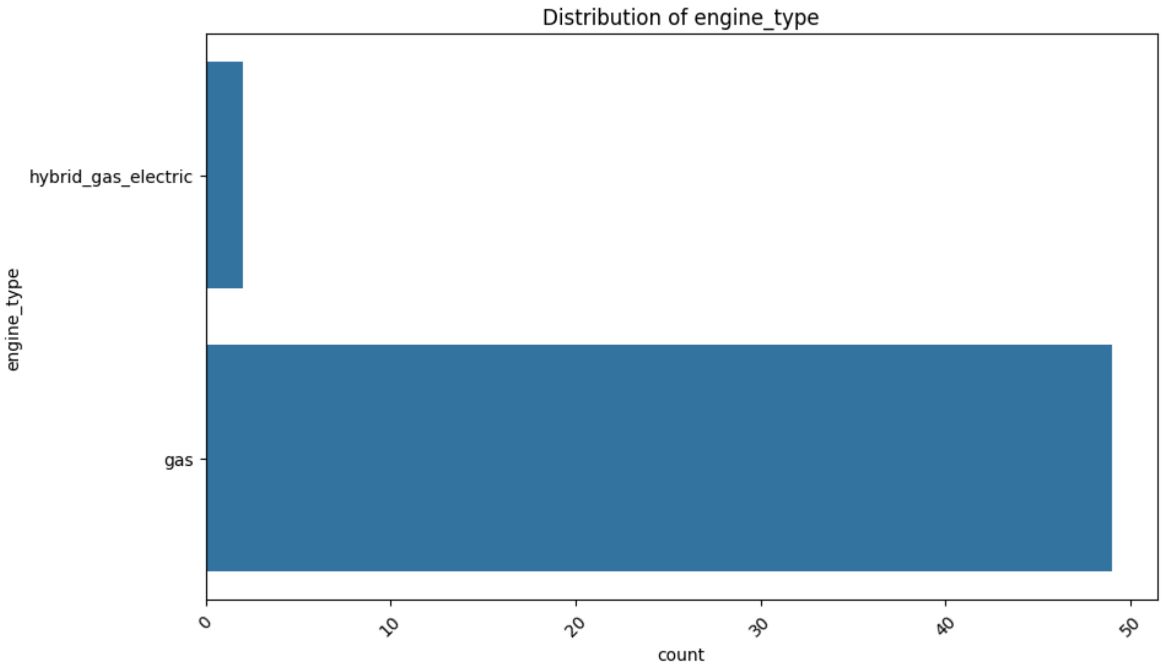
In [45]:
```
# 8. Plotting categorical variables
categorical_cols = data.select_dtypes(include=['object']).columns
for col in categorical_cols:
    plt.figure(figsize=(10, 6))
    sns.countplot(data[col])
    plt.title(f'Distribution of {col}')
    plt.xticks(rotation=45)
    plt.show()
```


Distribution of brand
Chevrolet
Hyundai
Ford
Honda
Subaru
Volkswagen
brand
0
5
10
15
20
count


Distribution of model
Volt LT
Elantra SE
Focus
Civic
Legacy
Jetta
Accord
Impreza
Insight
Taurus
Fusion
Malibu
Passat
Sonata
model
0
2
4
6
8
10
12
14
count







In [26]:
```
numeric_data = data.select_dtypes(include=['float64', 'int64'])
print(data.isnull().sum())  # Check for null values
print(np.isinf(numeric_data).sum())  # Check for infinite values
```

```
id                      0
brand                   0
model                   0
year                    0
miles                   0
city_mileage            0
highway_mileage         0
horsepower              0
torque                  0
engine_capacity_litre   0
fuel_capacity           0
num_cylinder            0
num_seat                0
num_owners              0
price                   0
link                    0
type                    0
doors                   0
wheel_drive             0
engine_type             0
speed_levels            0
front_headroom          0
front_legroom           0
rear_headroom           0
rear_legroom            0
service_records         0
mileage                 0
dtype: int64
id                      0
year                    0
miles                   0
city_mileage            0
highway_mileage         0
horsepower              0
torque                  0
engine_capacity_litre   0
fuel_capacity           0
num_cylinder            0
num_seat                0
num_owners              0
price                   0
doors                   0
wheel_drive             0
speed_levels            0
front_headroom          0
front_legroom           0
rear_headroom           0
rear_legroom            0
service_records         0
mileage                 0
dtype: int64
```

In [23]:

```
print(data[data['condition'].isnull()])
print(data[data['speed_levels'].isnull()])

data.drop(columns=['condition'], inplace=True)
data.dropna(subset=['speed_levels'], inplace=True)
```

	id	brand	model	year	miles	city_mileage	highway_mileage	\
0	3	Chevrolet	Volt LT	2019	27173	43	42	
1	6	Hyundai	Elantra SE	2017	76941	29	38	
2	7	Ford	Focus	2014	97027	27	37	
3	8	Honda	Civic	2016	95396	31	42	
5	11	Honda	Civic	2016	61459	31	41	
6	12	Subaru	Legacy	2022	6811	27	35	
7	13	Honda	Clarity	2018	29674	44	40	
8	14	Volkswagen	Jetta	2019	25044	30	40	
9	15	Volkswagen	Jetta	2017	26215	28	38	
10	16	Honda	Civic	2015	25939	29	37	
11	17	Honda	Civic	2019	32270	32	42	
12	18	Honda	Civic	2018	19950	31	40	
13	19	Honda	Accord	2020	19719	30	38	
14	20	Honda	Civic	2020	16076	32	42	
15	21	Subaru	Impreza	2019	28214	28	38	
16	22	Honda	Accord	2020	12395	30	38	
17	23	Honda	Civic	2019	33322	30	38	
18	24	Honda	Insight	2019	56233	55	49	
19	25	Volkswagen	Jetta	2019	24250	30	40	
20	26	Ford	Taurus	2019	38097	17	24	
21	27	Ford	Fusion	2020	35743	20	29	
22	28	Chevrolet	Malibu	2019	32946	29	36	
23	29	Volkswagen	Passat	2018	26905	25	36	
24	30	Volkswagen	Passat	2017	66329	23	34	
25	31	Volkswagen	Passat	2020	27457	23	34	
26	32	Honda	Accord	2020	12395	30	38	
27	33	Honda	Civic	2019	41799	30	38	
28	34	Hyundai	Sonata	2019	26910	25	33	
29	35	Honda	Civic	2018	16581	31	40	
30	36	Hyundai	Sonata	2019	9736	25	33	
31	37	Honda	Civic	2017	7743	31	40	
32	38	Subaru	Impreza	2019	44371	28	38	
33	39	Volkswagen	Jetta	2020	12944	25	32	
34	40	Volkswagen	Jetta	2019	5000	30	40	
35	41	Honda	Civic	2019	41799	30	38	
36	42	Chevrolet	Malibu	2018	13726	22	32	
37	43	Volkswagen	Jetta	2019	18898	30	40	
38	44	Honda	Accord	2018	51279	30	38	
39	45	Ford	Fusion	2020	36506	21	31	
40	46	Honda	Accord	2018	58126	30	38	
41	47	Honda	Civic	2019	27440	32	42	
42	48	Honda	Accord	2020	59090	30	38	
43	49	Honda	Civic	2019	47715	30	38	
44	50	Volkswagen	Passat	2020	9495	23	34	
45	51	Volkswagen	Jetta	2019	13721	30	40	
46	52	Honda	Civic	2018	72827	31	40	
47	53	Hyundai	Sonata	2019	33412	23	32	
48	54	Hyundai	Elantra SE	2020	13830	30	40	
49	55	Ford	Fusion	2019	23507	20	29	
50	56	Hyundai	Elantra SE	2020	11296	30	40	
51	57	Honda	Accord	2018	82729	30	38	

	horsepower	torque	engine_capacity_litre	...	doors	wheel_drive	\
0	149	294	1.5	...	4	2	
1	146	132	2.0	...	4	2	
2	159	146	2.0	...	4	2	
3	158	138	1.5	...	4	2	
5	158	138	2.0	...	4	2	
6	182	176	2.5	...	4	4	

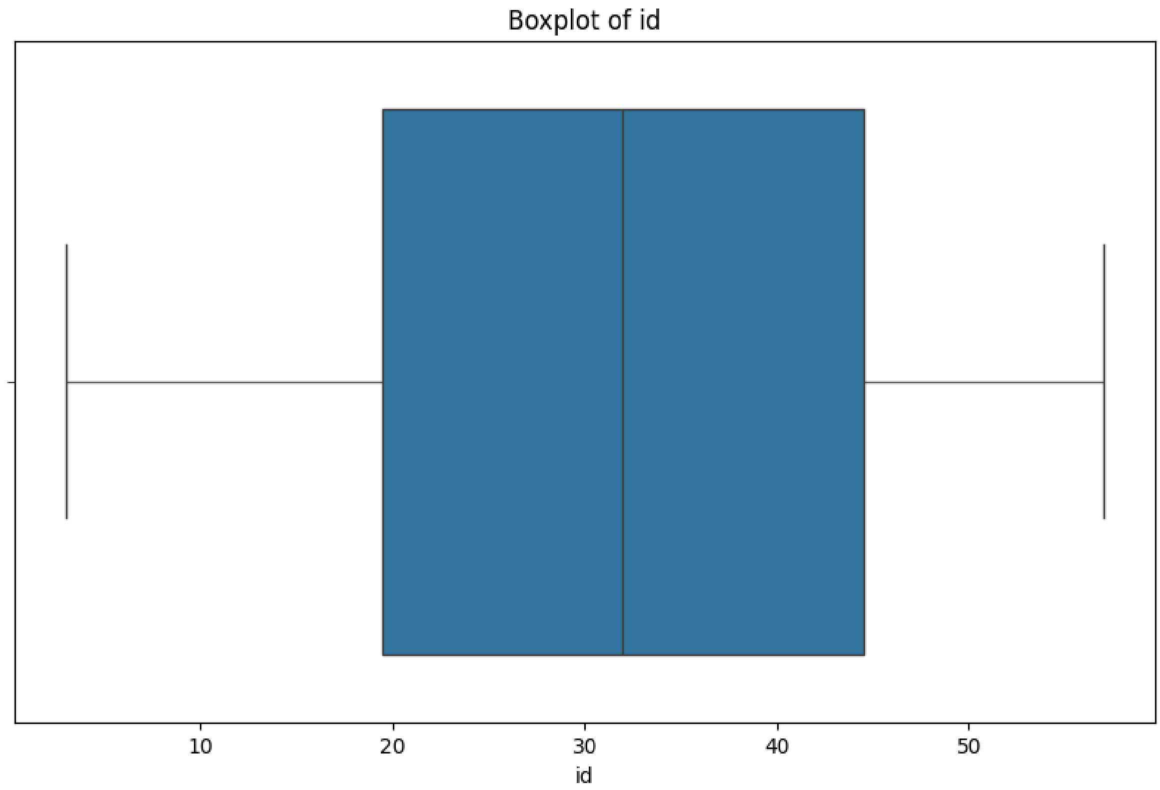
```
7          212     99               1.5  ...      4            2
8          147    184               1.4  ...      4            2
9          150    184               1.4  ...      4            2
10         143    129               1.8  ...      4            2
11         174    162               1.5  ...      4            2
12         158    138               2.0  ...      4            2
13         192    192               1.5  ...      4            2
14         174    162               1.5  ...      4            2
15         152    145               2.0  ...      4            2
16         192    192               1.5  ...      4            2
17         158    138               2.0  ...      4            2
18         151     99               1.5  ...      4            2
19         147    184               1.4  ...      4            2
20         288    254               3.5  ...      4            4
21         245    275               2.0  ...      4            4
22         160    184               1.5  ...      4            2
23         174    184               2.0  ...      4            2
24         170    184               1.8  ...      4            2
25         174    206               2.0  ...      4            2
26         192    192               1.5  ...      4            2
27         158    138               2.0  ...      4            2
28         185    178               2.4  ...      4            2
29         158    138               2.0  ...      4            2
30         185    178               2.4  ...      4            2
31         158    138               2.0  ...      4            2
32         152    145               2.0  ...      4            4
33         228    258               2.0  ...      4            2
34         147    184               1.4  ...      4            2
35         158    138               2.0  ...      4            2
36         250    260               2.0  ...      4            2
37         158    184               1.4  ...      4            2
38         192    192               1.5  ...      4            2
39         245    275               2.0  ...      4            2
40         192    192               1.5  ...      4            2
41         174    162               1.5  ...      4            2
42         192    192               1.5  ...      4            2
43         158    138               2.0  ...      4            2
44         174    206               2.0  ...      4            2
45         147    184               1.4  ...      4            2
46         158    138               2.0  ...      4            2
47         245    260               2.0  ...      4            2
48         147    132               2.0  ...      4            2
49         245    275               2.0  ...      4            2
50         147    132               2.0  ...      4            2
51         192    192               1.5  ...      4            2

            engine_type  speed_levels  front_headroom front_legroom  \
0   hybrid_gas_electric           6.0            37.8          42.1
1                   gas           6.0            39.0          42.2
2                   gas           6.0            38.3          43.7
3                   gas           6.0            37.5          42.3
5                   gas           6.0            37.5          42.3
6                   gas           6.0            39.4          42.8
7   hybrid_gas_electric           NaN            39.1          42.2
8                   gas           8.0            38.5          41.1
9                   gas           6.0            38.2          41.2
10                  gas           6.0            37.9          42.0
11                  gas           6.0            37.5          42.3
12                  gas           6.0            37.5          42.3
13                  gas           6.0            37.5          42.3
```

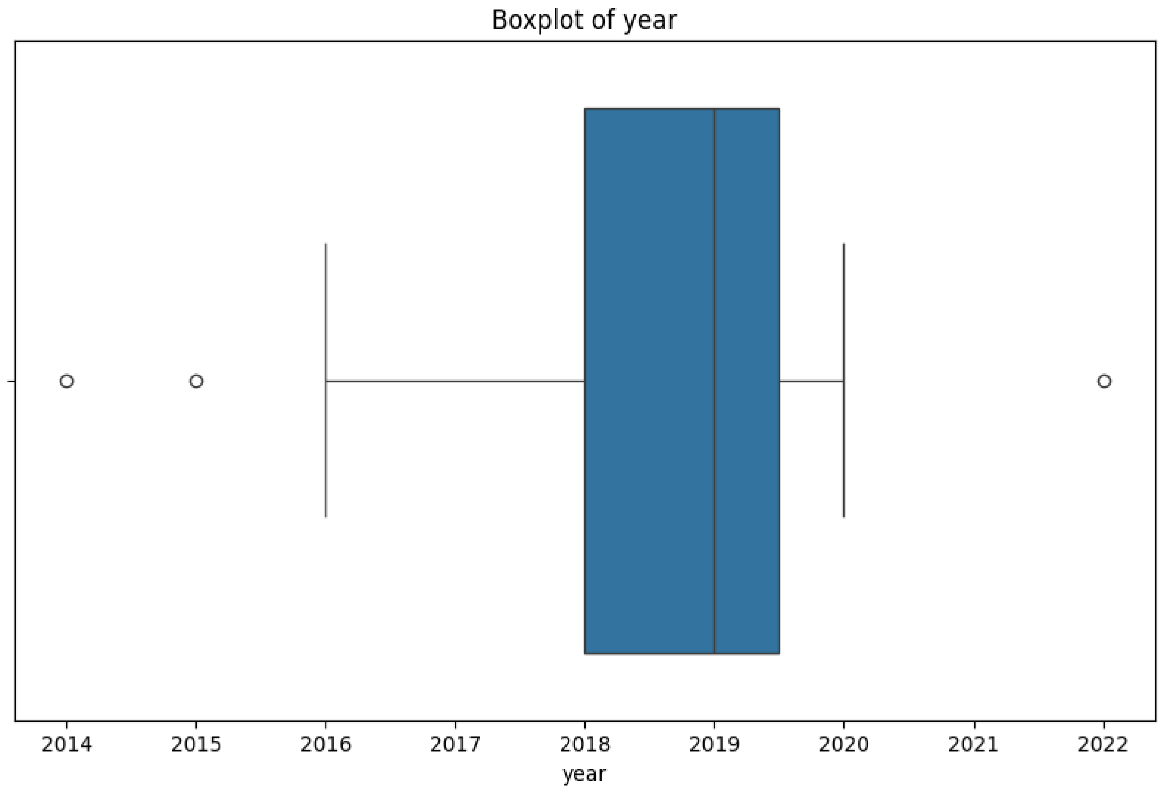
14	gas	6.0	37.5	42.3
15	gas	6.0	39.8	43.1
16	gas	6.0	37.5	42.3
17	gas	6.0	39.3	42.3
18	hybrid_gas_electric	6.0	39.3	42.3
19	gas	8.0	38.5	41.1
20	gas	6.0	39.0	41.9
21	gas	6.0	39.2	44.3
22	gas	6.0	39.1	41.5
23	gas	6.0	38.3	42.4
24	gas	6.0	38.3	42.4
25	gas	6.0	38.3	42.4
26	gas	6.0	37.5	42.3
27	gas	6.0	39.3	42.3
28	gas	6.0	40.4	45.5
29	gas	6.0	37.5	42.3
30	gas	6.0	40.4	45.5
31	gas	6.0	37.5	42.3
32	gas	6.0	39.8	43.1
33	gas	7.0	38.5	41.1
34	gas	8.0	38.5	41.1
35	gas	6.0	39.3	42.3
36	gas	9.0	39.1	42.0
37	gas	8.0	38.5	41.1
38	gas	6.0	37.5	42.3
39	gas	6.0	39.2	44.3
40	gas	6.0	37.5	42.3
41	gas	6.0	37.5	42.3
42	gas	6.0	39.5	42.3
43	gas	6.0	39.3	42.3
44	gas	6.0	38.3	42.4
45	gas	8.0	38.5	41.1
46	gas	6.0	39.3	42.3
47	gas	8.0	40.4	45.5
48	gas	6.0	40.3	42.2
49	gas	6.0	39.2	44.3
50	gas	6.0	40.3	42.2
51	gas	6.0	39.5	42.3

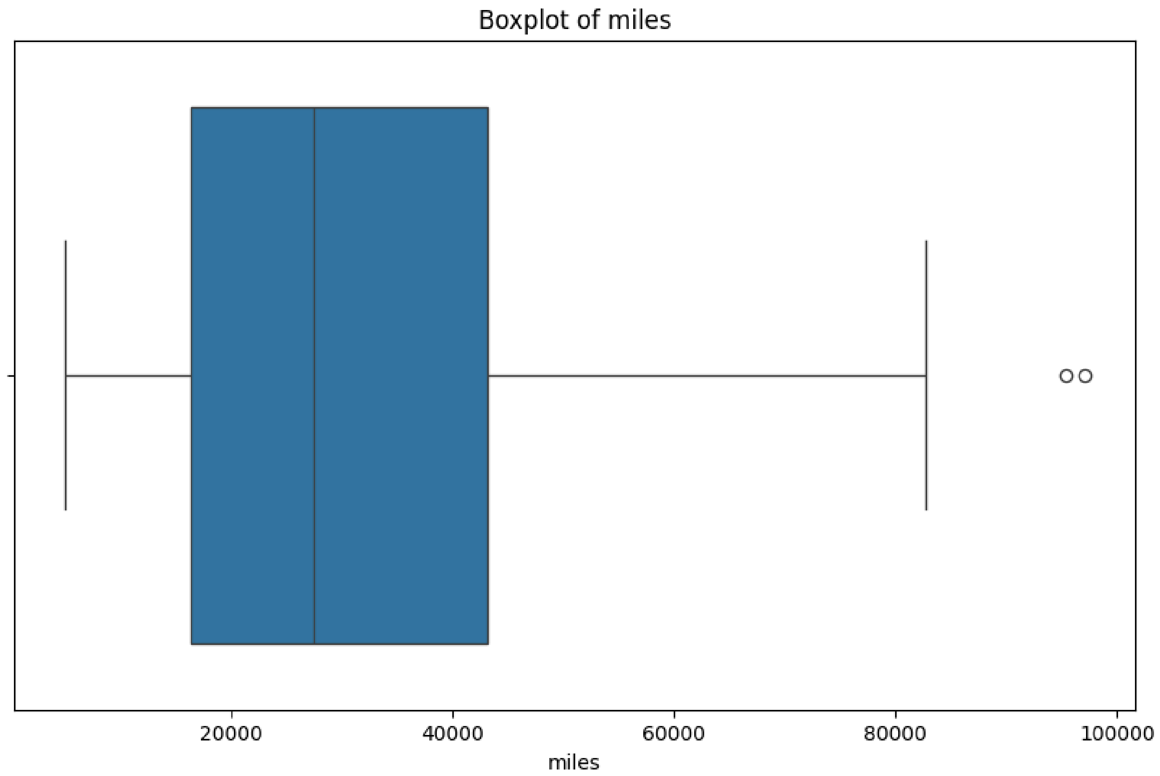
	rear_headroom	rear_legroom	service_records	mileage
0	35.8	34.7	4	42.5
1	37.0	35.7	16	33.5
2	38.0	33.2	13	32.0
3	36.8	37.4	26	36.5
5	36.8	37.4	13	36.0
6	37.2	39.5	6	31.0
7	37.1	36.2	2	42.0
8	37.2	37.4	6	35.0
9	37.1	38.1	13	33.0
10	36.2	36.2	15	33.0
11	36.8	37.4	4	37.0
12	36.8	37.4	9	35.5
13	37.2	40.4	4	34.0
14	36.8	37.4	5	37.0
15	37.2	36.5	4	33.0
16	37.2	40.4	4	34.0
17	37.1	37.4	4	34.0
18	36.9	37.4	5	52.0
19	37.2	37.4	8	35.0
20	37.8	38.1	1	20.5

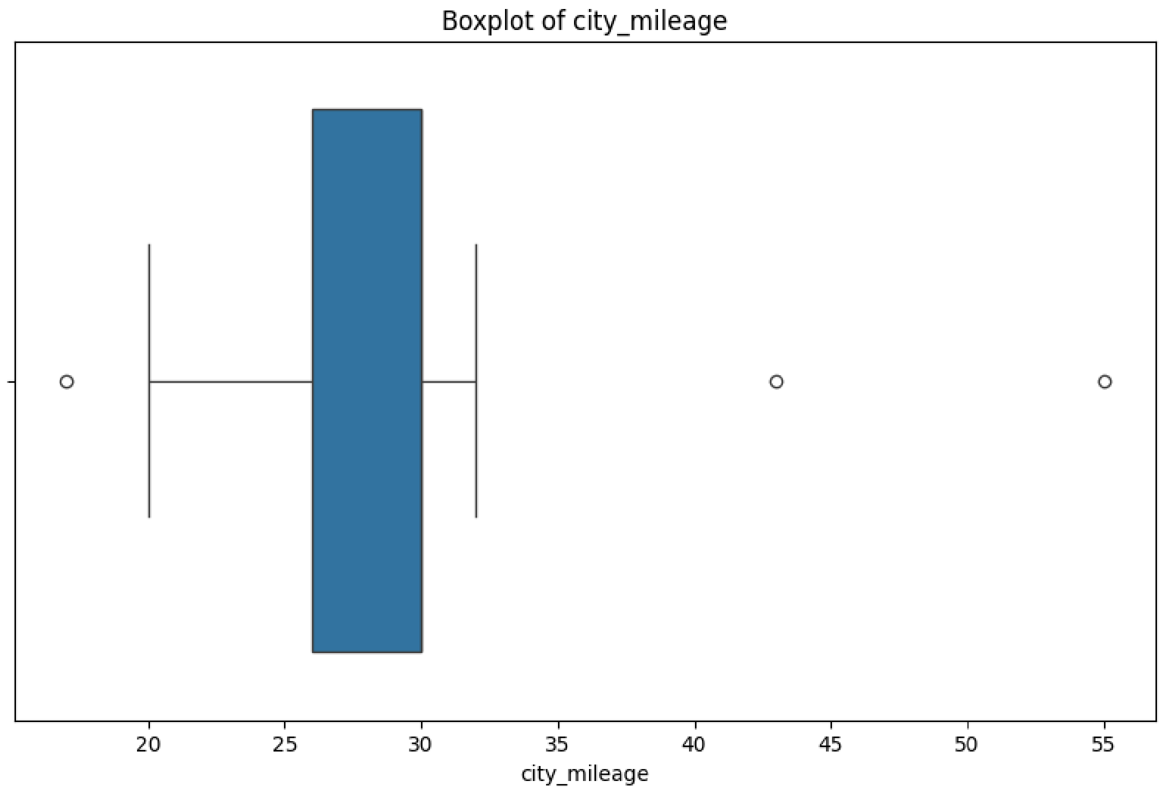
```
21          37.8          38.3            8     24.5
22          37.5          38.1           15     32.5
23          37.8          39.1           11     30.5
24          37.8          39.1            5     28.5
25          37.8          39.1           11     28.5
26          37.2          40.4            4     34.0
27          37.1          37.4            4     34.0
28          38.0          35.6            4     29.0
29          36.8          37.4            4     35.5
30          38.0          35.6            6     29.0
31          36.8          37.4            2     35.5
32          37.2          36.5            7     33.0
33          37.2          37.4            5     28.5
34          37.2          37.4            6     35.0
35          37.1          37.4            4     34.0
36          37.5          38.1            7     27.0
37          37.2          37.4            4     35.0
38          37.2          40.4            8     34.0
39          37.8          38.3            7     26.0
40          37.2          40.4           14     34.0
41          36.8          37.4           13     37.0
42          37.3          40.4            1     34.0
43          37.1          37.4            1     34.0
44          37.8          39.1           10     28.5
45          37.2          37.4            5     35.0
46          37.1          37.4            5     35.5
47          38.0          35.6           10     27.5
48          37.3          35.7            6     35.0
49          37.8          38.3           11     24.5
50          37.3          35.7            7     35.0
51          37.3          40.4            9     34.0

[51 rows x 28 columns]
   id  brand    model  year  miles  city_mileage  highway_mileage  horsepower  \
7  13  Honda  Clarity  2018  29674            44               40         212   

   torque  engine_capacity_litre  ...  doors  wheel_drive  \
7      99                    1.5  ...      4            2   

           engine_type  speed_levels  front_headroom front_legroom  \
7  hybrid_gas_electric           NaN            39.1          42.2   

   rear_headroom rear_legroom  service_records  mileage  
7           37.1         36.2                2     42.0  

[1 rows x 28 columns]
```

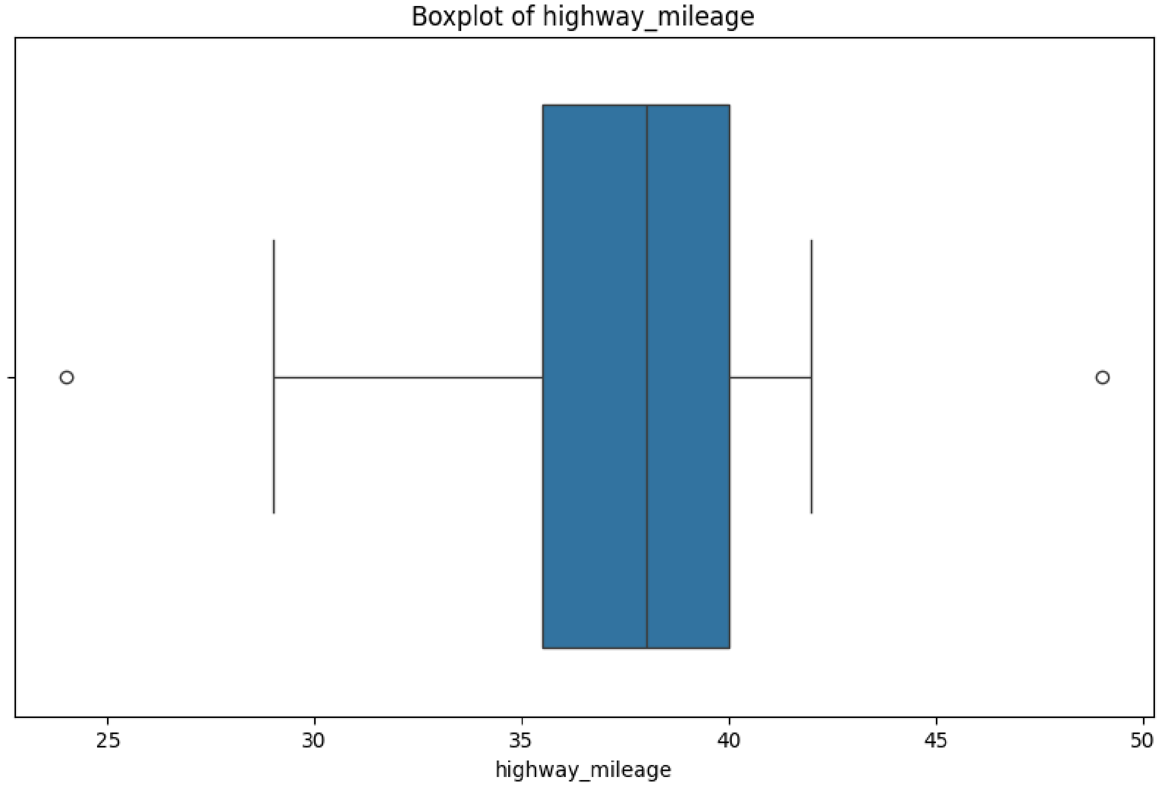
In [27]:
```python
# 10. Boxplots for outliers in numerical variables
numerical_cols = data.select_dtypes(include=['float64', 'int64']).columns
for col in numerical_cols:
    plt.figure(figsize=(10, 6))
    sns.boxplot(x=data[col])
    plt.title(f'Boxplot of {col}')
    plt.show()
```

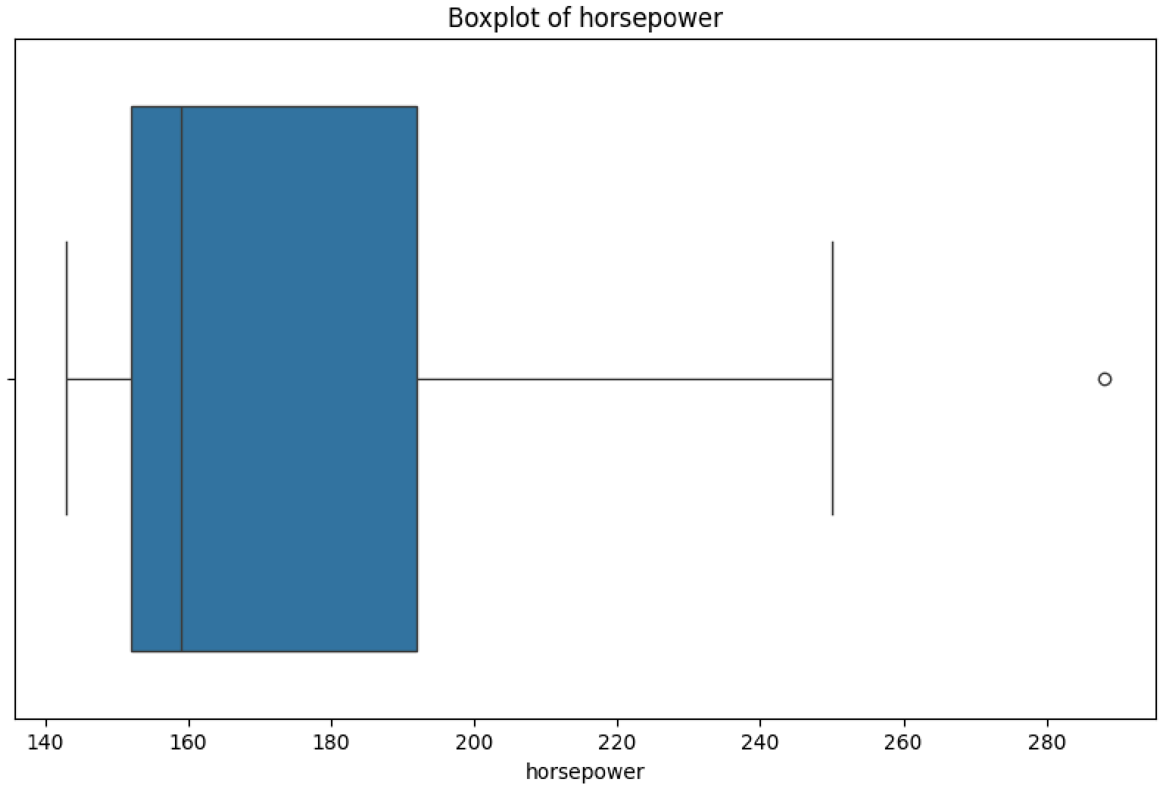

Boxplot of id
10
20
30
40
50
id

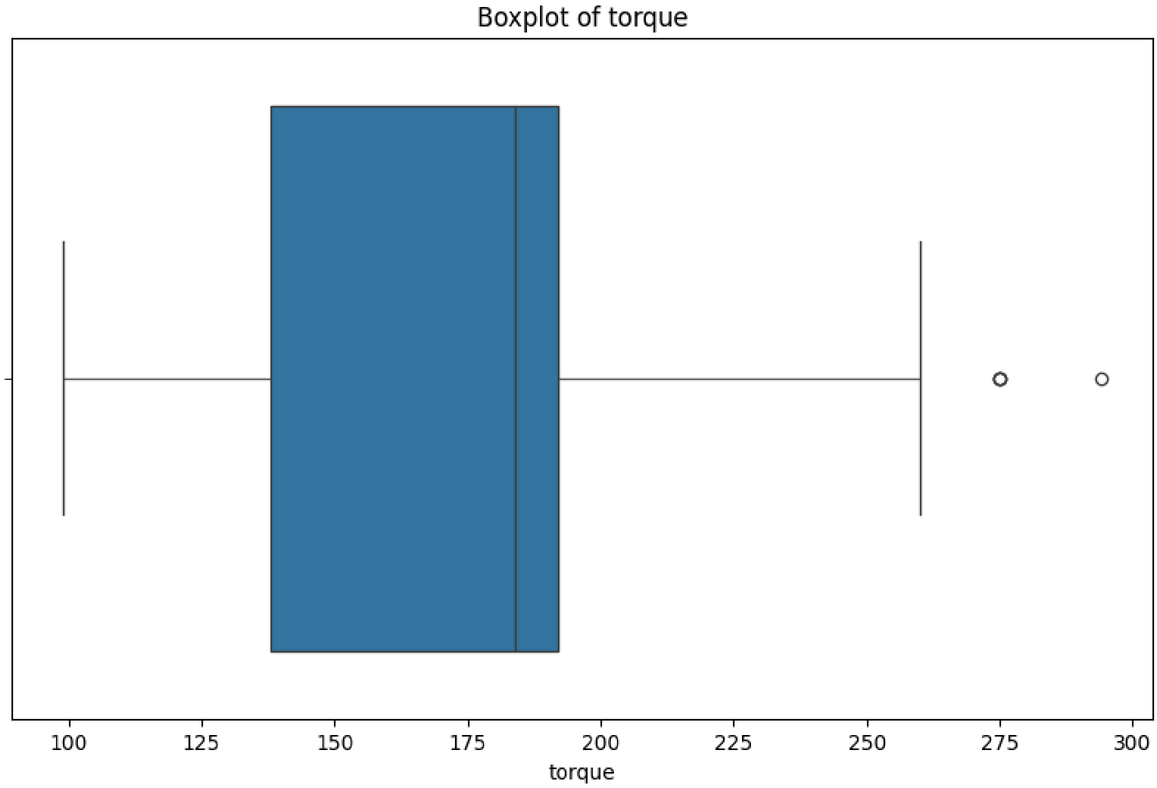

Boxplot of year
2014
2015
2016
2017
2018
2019
2020
2021
2022
year

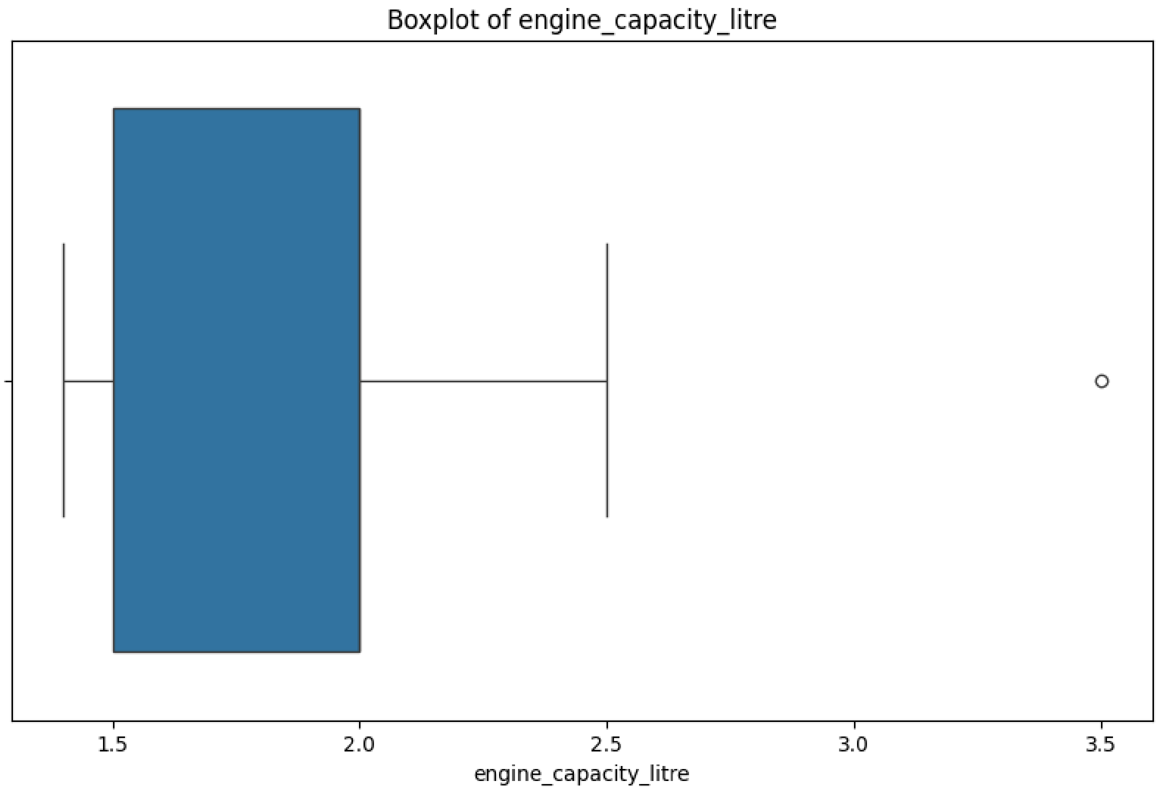

Boxplot of miles
20000
40000
60000
80000
100000
miles

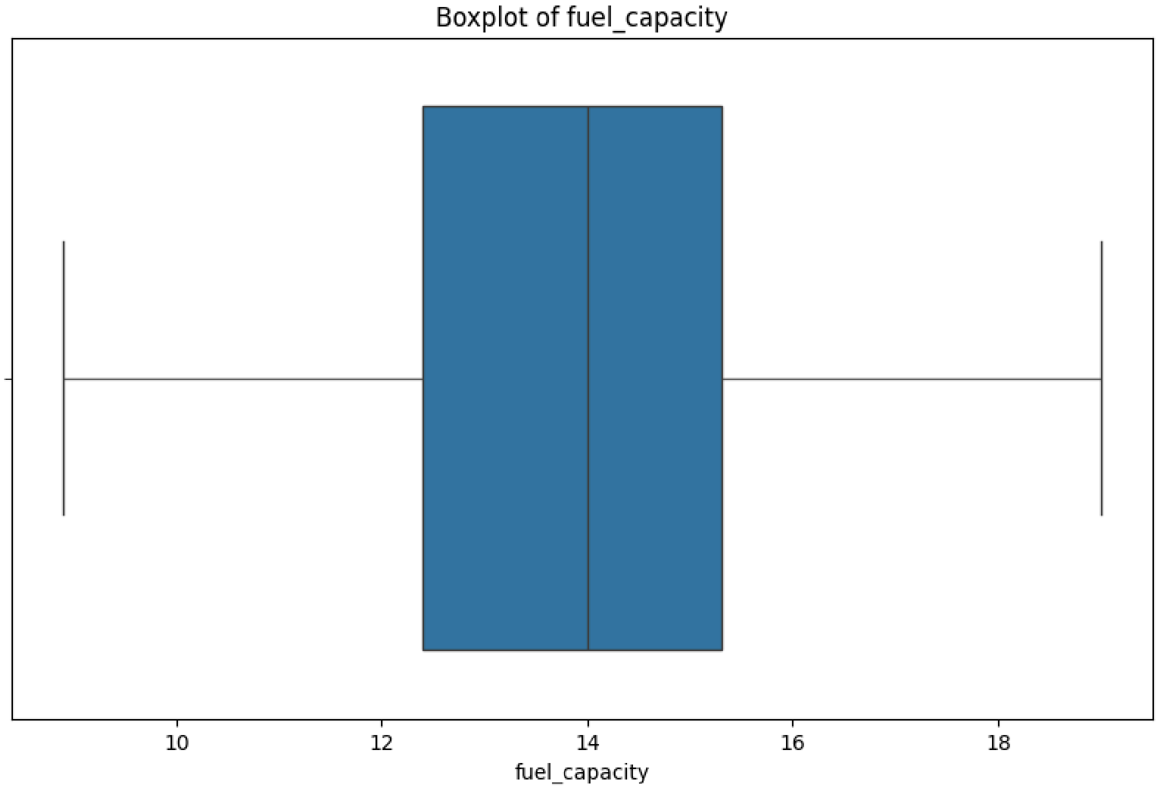

Boxplot of city_mileage
20
25
30
35
40
45
50
55
city_mileage

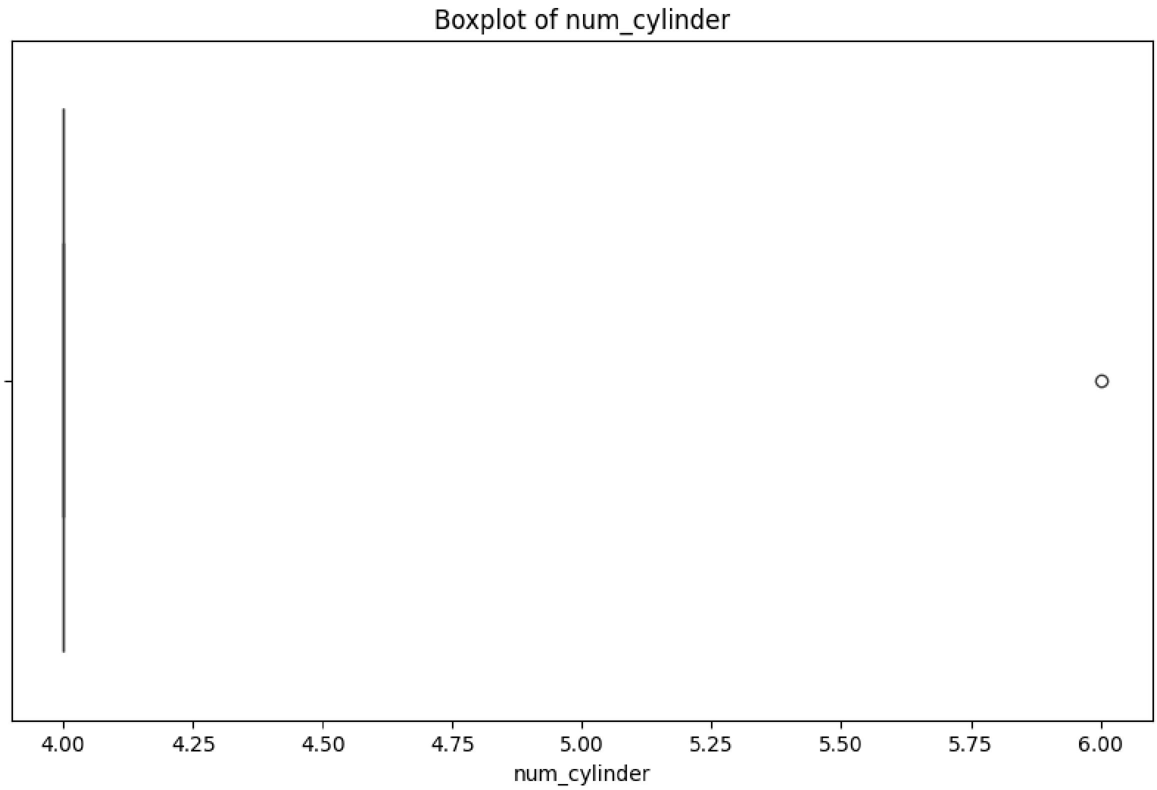

Boxplot of highway_mileage
25
30
35
40
45
50
highway_mileage

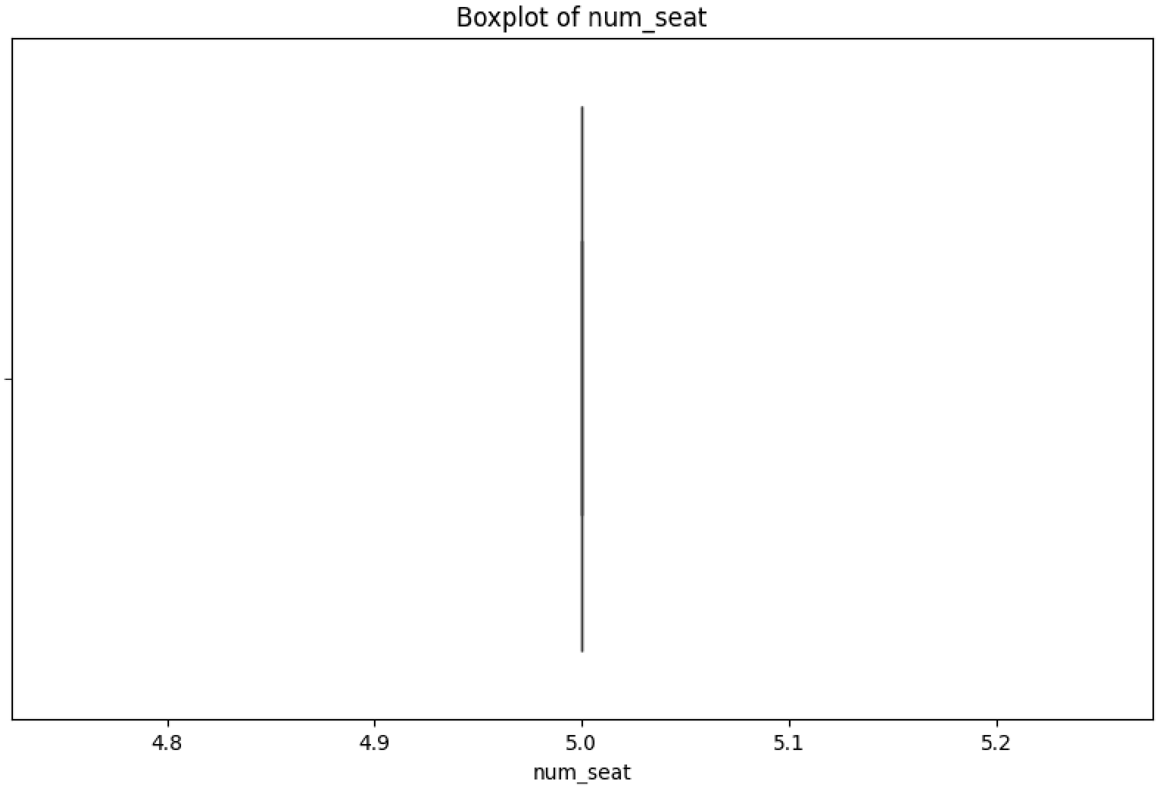

Boxplot of horsepower
140
160
180
200
220
240
260
280
horsepower

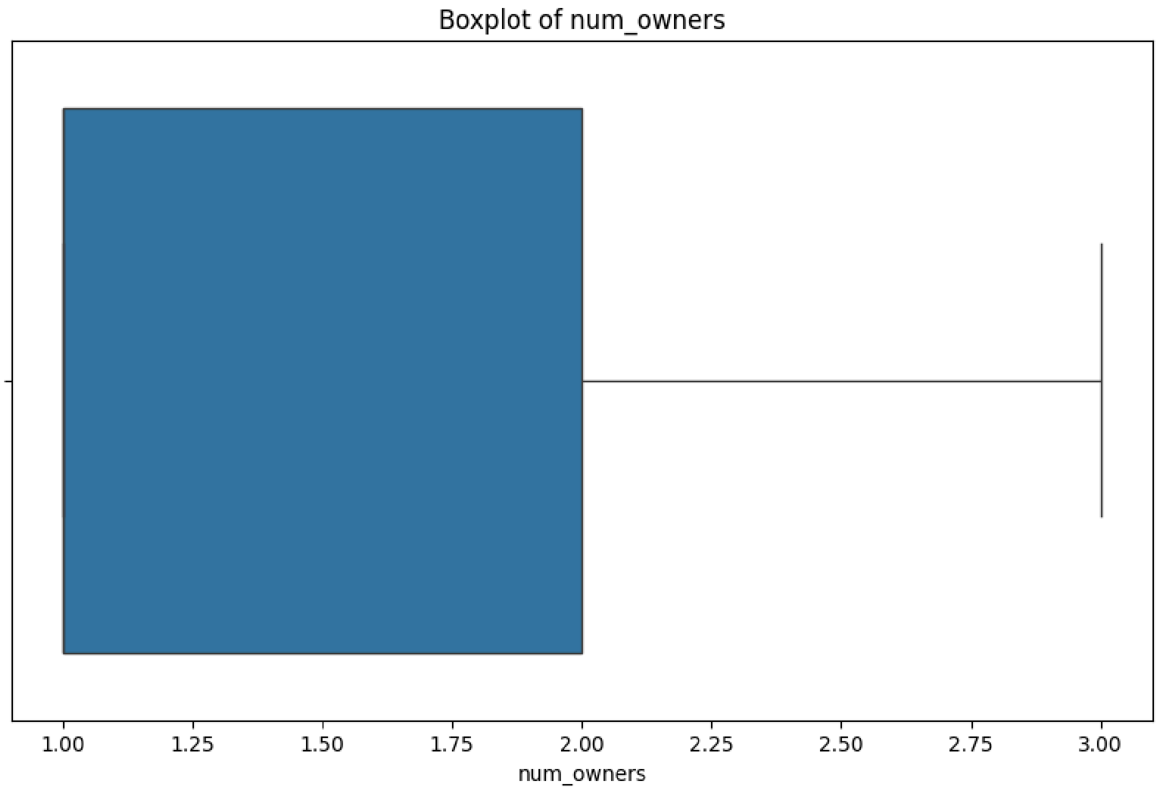

Boxplot of torque
100
125
150
175
200
225
250
275
300
torque

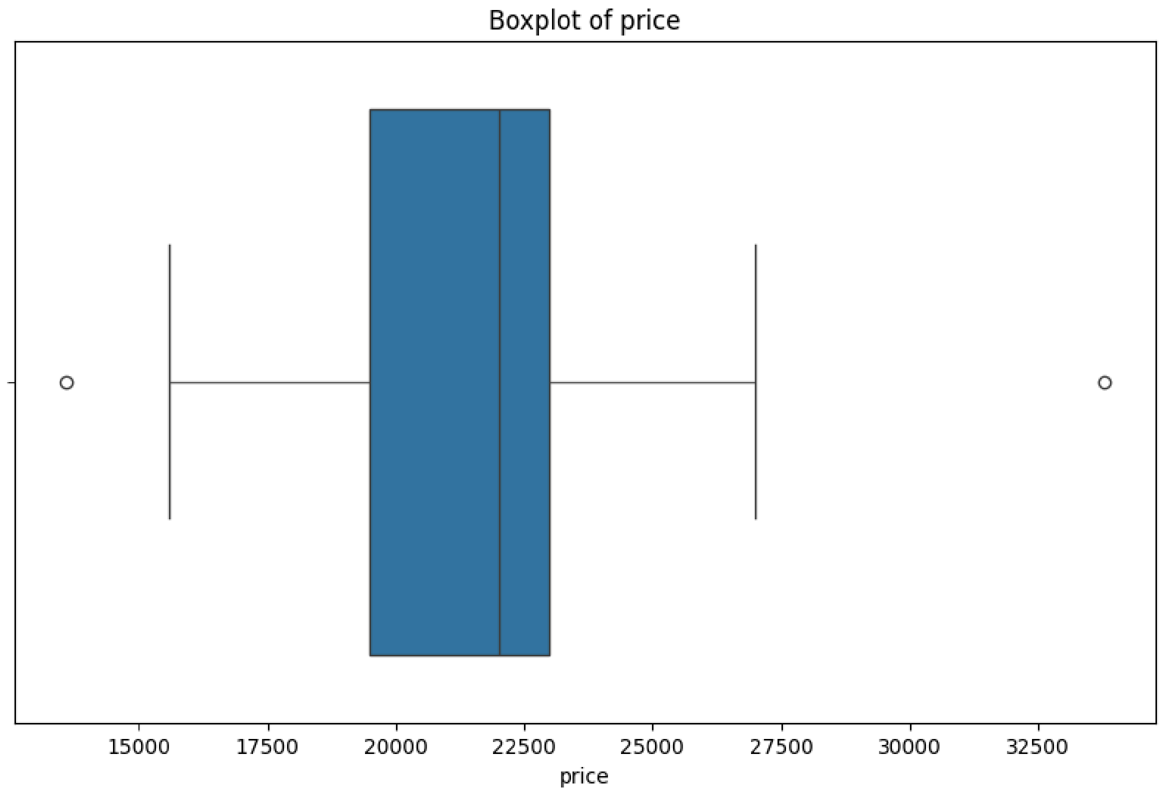

Boxplot of engine_capacity_litre
1.5
2.0
2.5
3.0
3.5
engine_capacity_litre

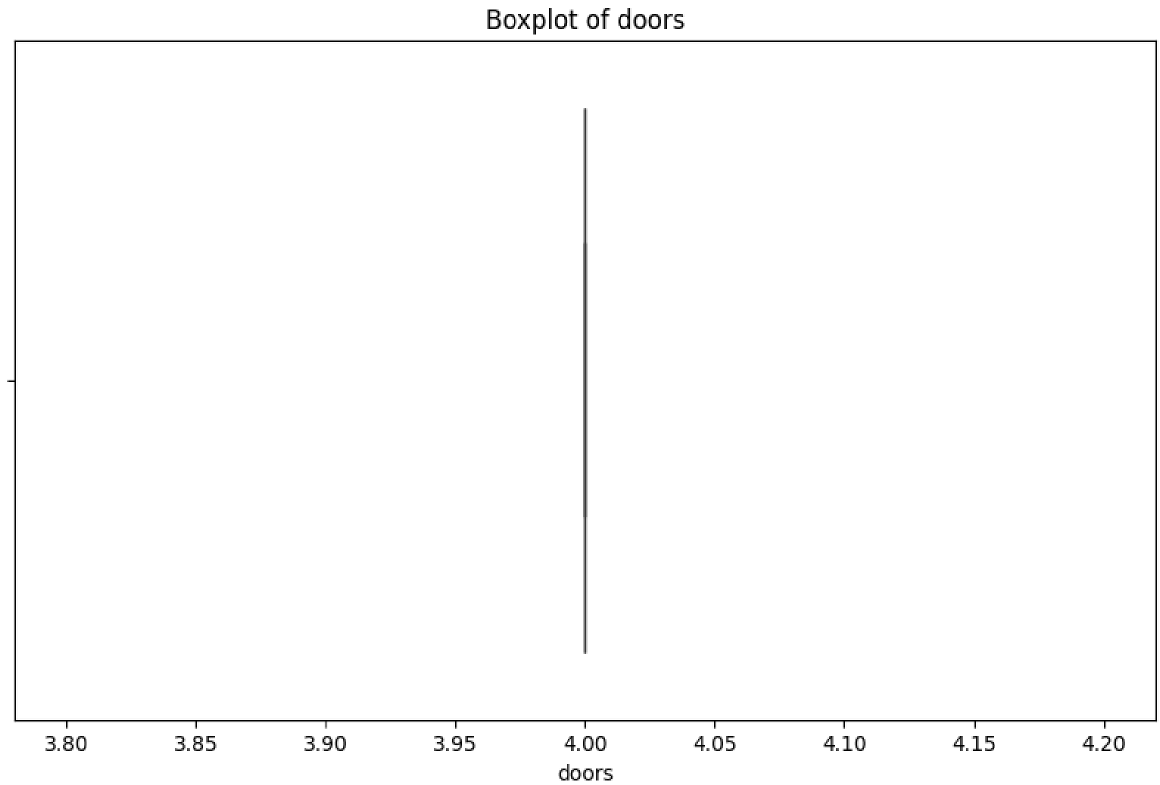

Boxplot of fuel_capacity
10
12
14
16
18
fuel_capacity

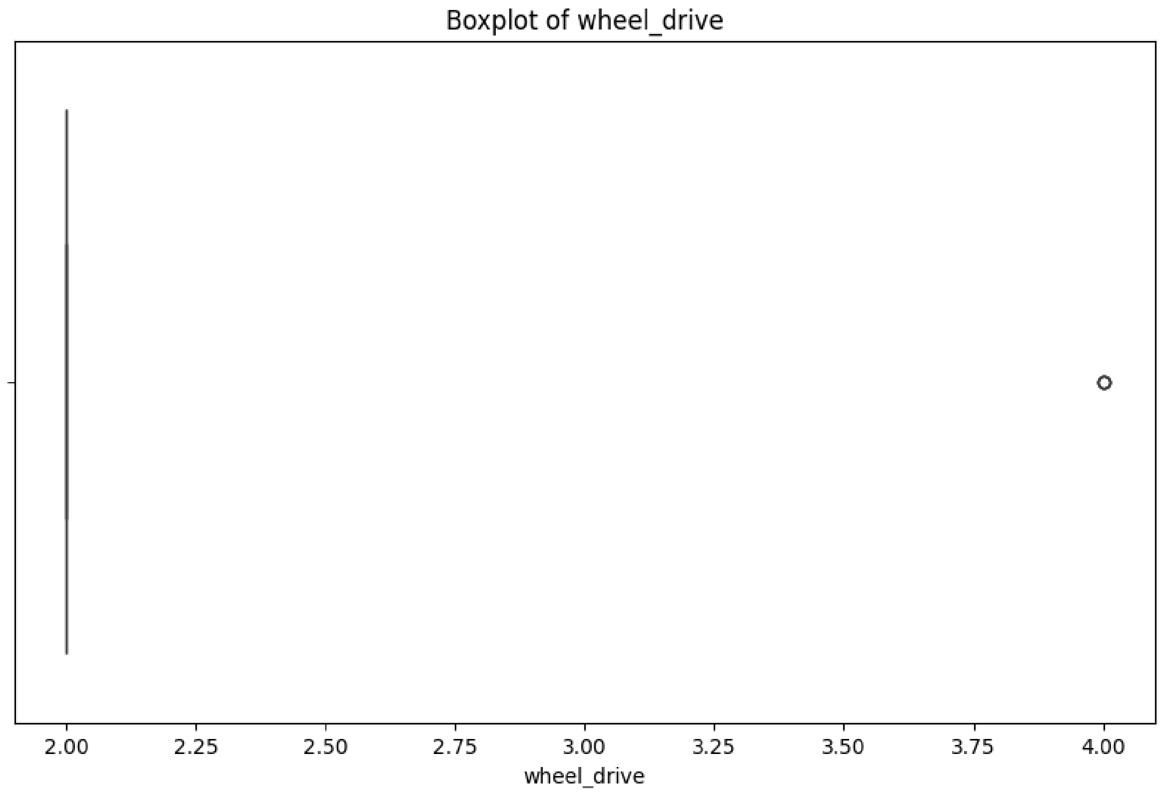

Boxplot of num_cylinder
4.00
4.25
4.50
4.75
5.00
5.25
5.50
5.75
6.00
num_cylinder

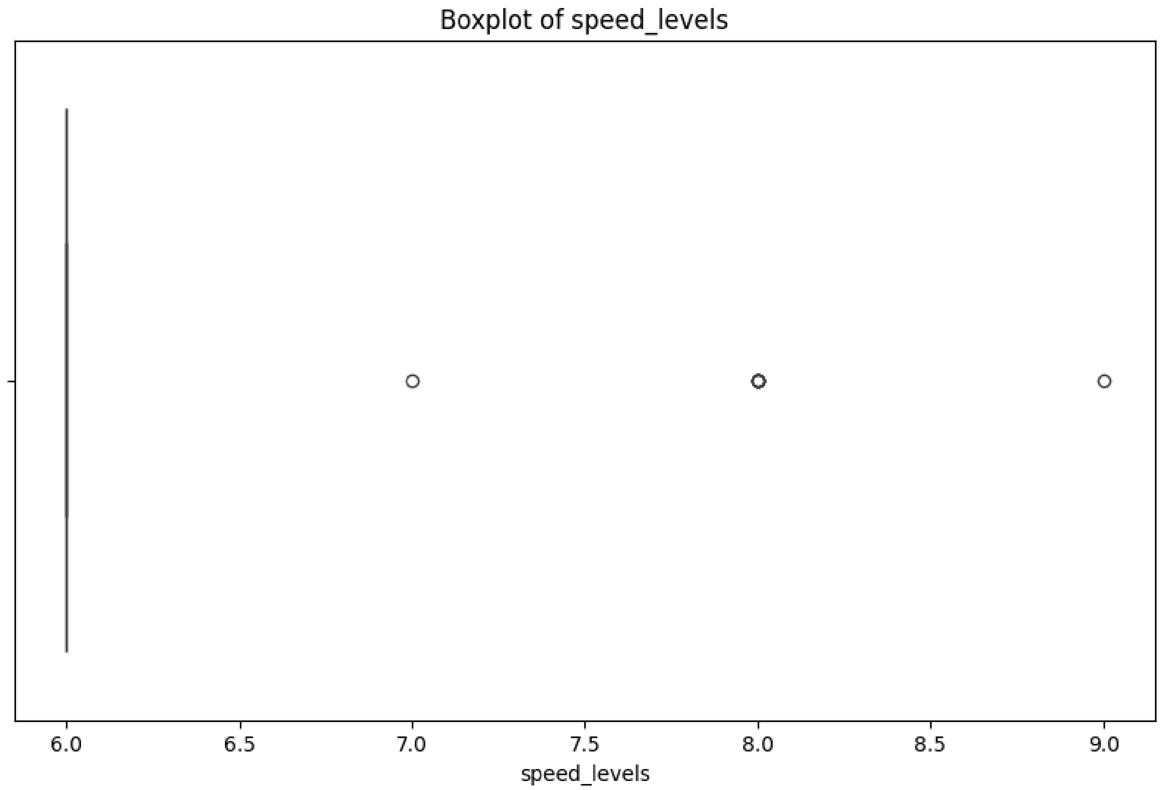

Boxplot of num_seat
4.8
4.9
5.0
5.1
5.2
num_seat

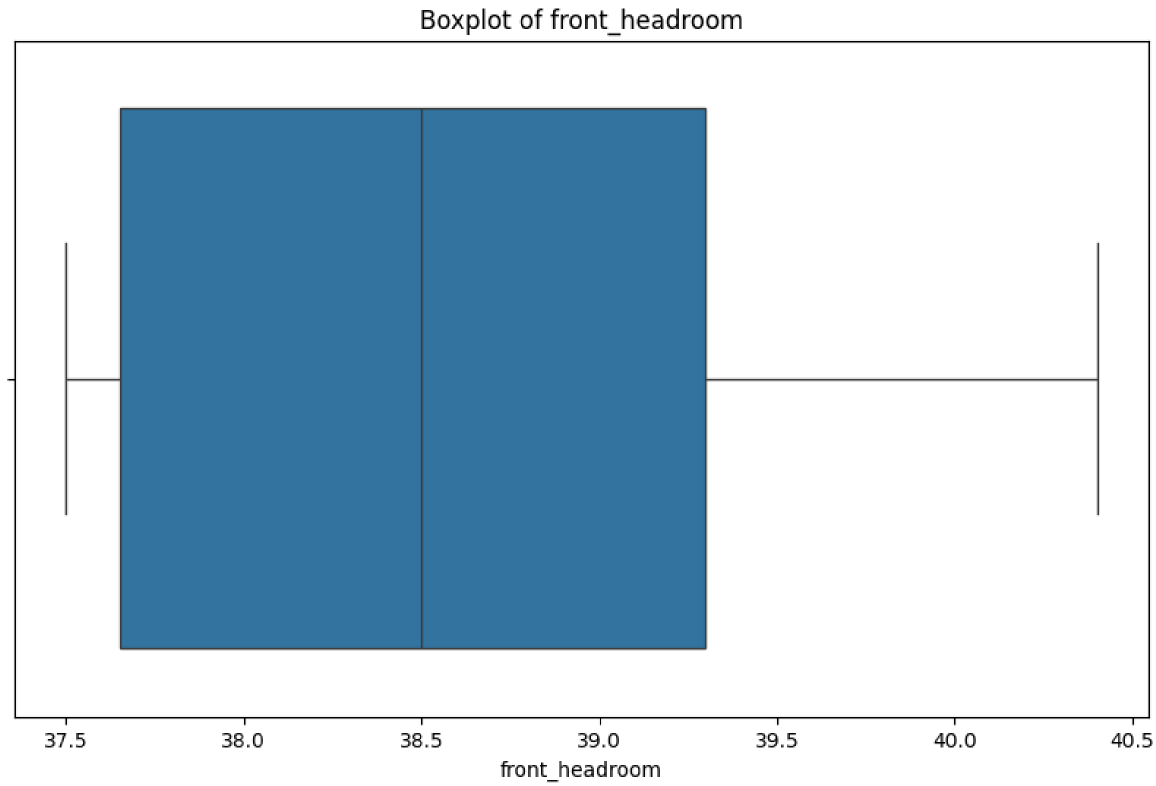

Boxplot of num_owners
1.00
1.25
1.50
1.75
2.00
2.25
2.50
2.75
3.00
num_owners

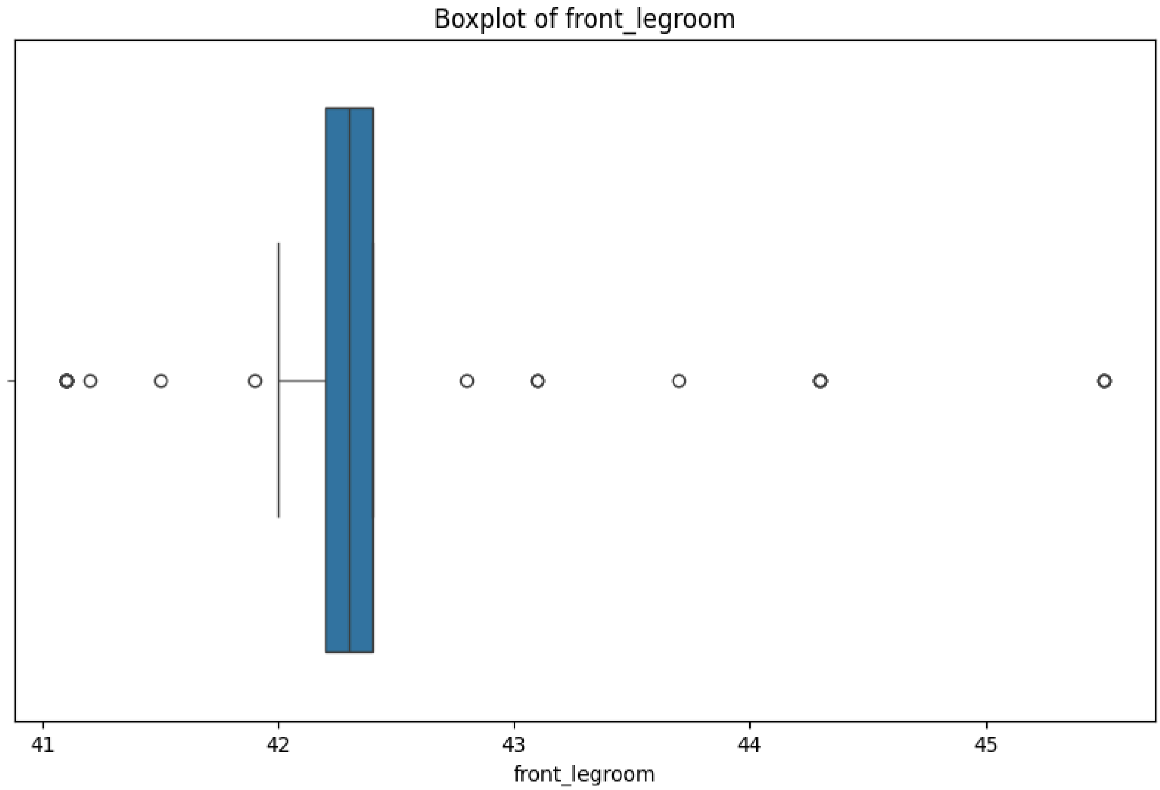

Boxplot of price
15000
17500
20000
22500
25000
27500
30000
32500
price

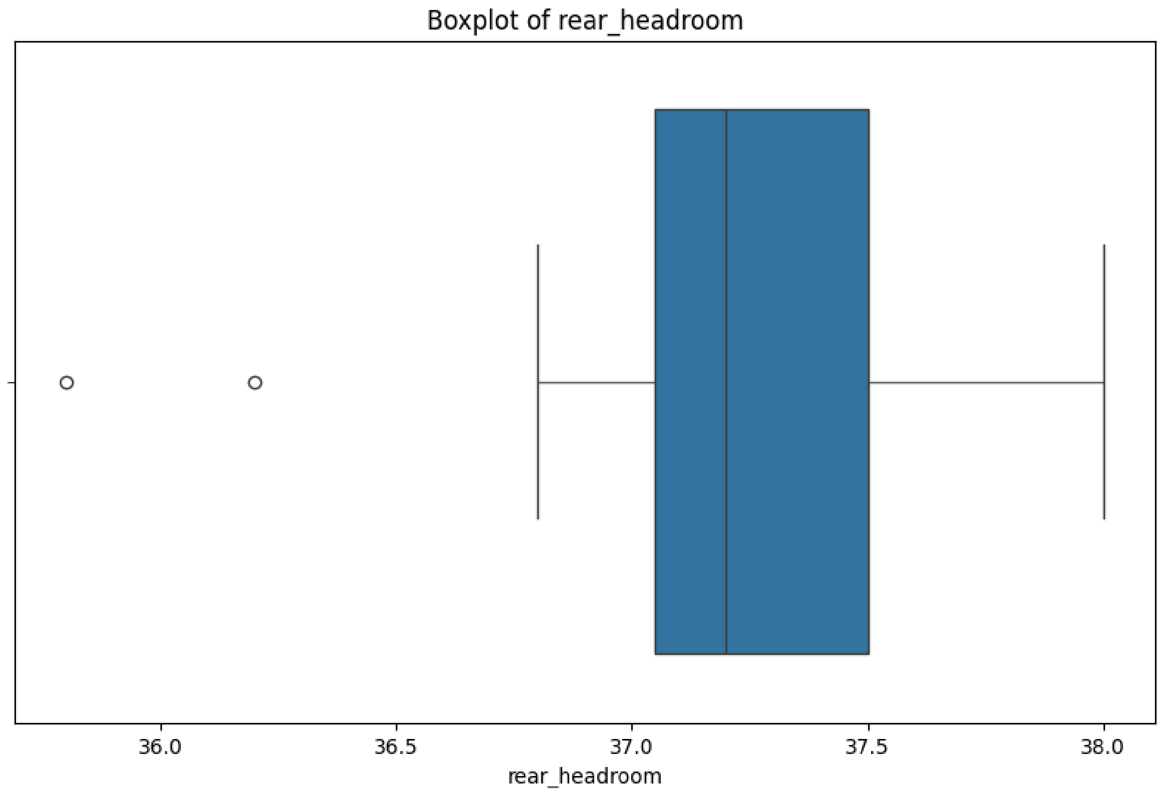

Boxplot of doors
3.80
3.85
3.90
3.95
4.00
4.05
4.10
4.15
4.20
doors

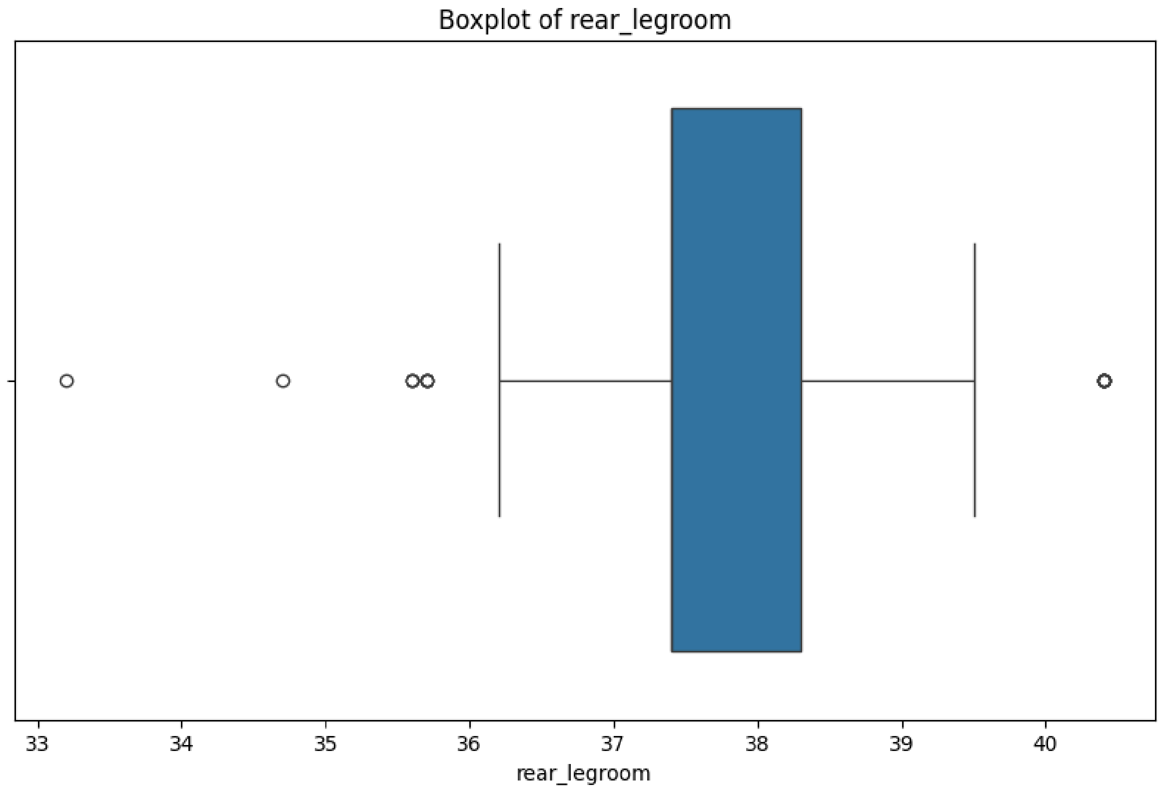

Boxplot of wheel_drive
2.00
2.25
2.50
2.75
3.00
3.25
3.50
3.75
4.00
wheel_drive

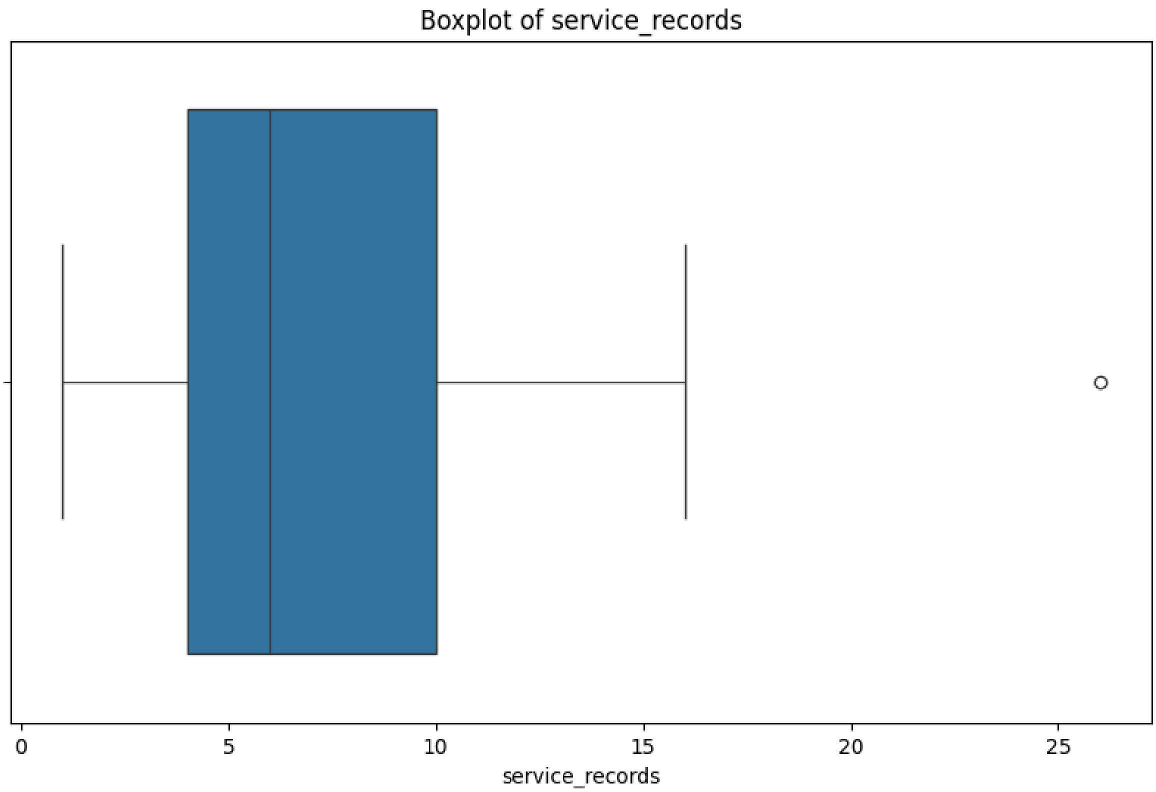

Boxplot of speed_levels
6.0
6.5
7.0
7.5
8.0
8.5
9.0
speed_levels

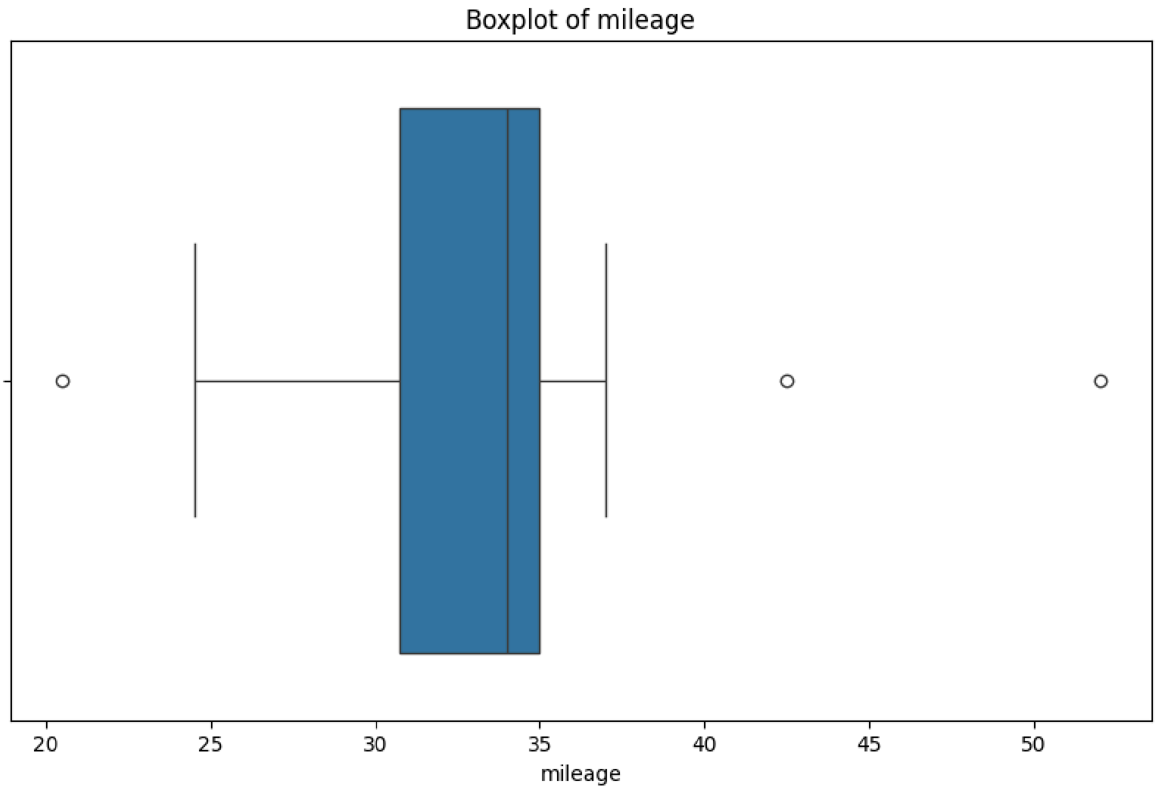

Boxplot of front_headroom
37.5
38.0
38.5
39.0
39.5
40.0
40.5
front_headroom


Boxplot of front_legroom
41
42
43
44
45
front_legroom


Boxplot of rear_headroom
36.0
36.5
37.0
37.5
38.0
rear_headroom


Boxplot of rear_legroom
33
34
35
36
37
38
39
40
rear_legroom

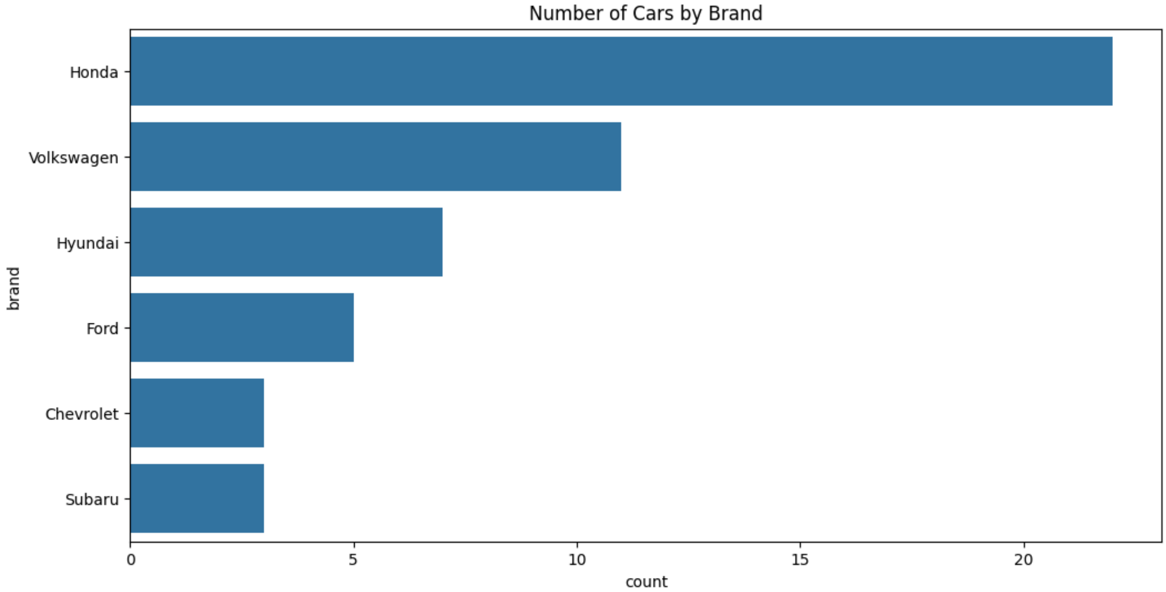




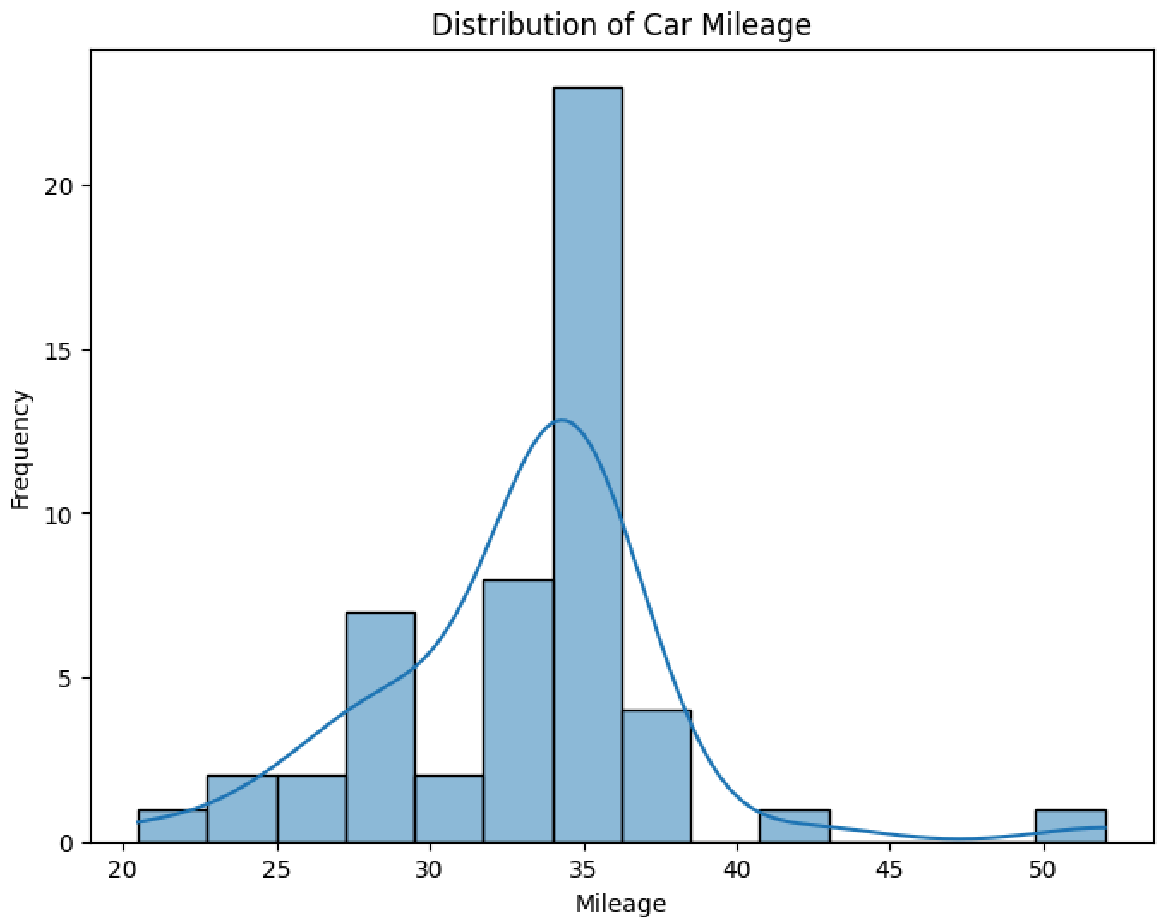
In [28]:

```
# 11. Count of cars by brand (if brand column exists)
if 'brand' in data.columns:
    print('\n--- Count of Cars by Brand ---')
    print(data['brand'].value_counts())
    plt.figure(figsize=(12, 6))
    sns.countplot(y=data['brand'], order=data['brand'].value_counts().index)
    plt.title('Number of Cars by Brand')
    plt.show()
```

```
--- Count of Cars by Brand ---
brand
Honda         22
Volkswagen    11
Hyundai        7
Ford           5
Chevrolet      3
Subaru         3
Name: count, dtype: int64
```



In [29]:
```
# 12. Mileage distribution
if 'mileage' in data.columns:
    plt.figure(figsize=(8, 6))
    sns.histplot(data['mileage'], kde=True)
    plt.title('Distribution of Car Mileage')
    plt.xlabel('Mileage')
    plt.ylabel('Frequency')
    plt.show()
```



In [31]:

```
# 13. Transmission type count
if 'transmission' in data.columns:
    plt.figure(figsize=(6, 6))
    sns.countplot(data['transmission'])
    plt.title('Distribution of Transmission Types')
    plt.show()

# 14. Fuel type distribution
if 'fuelType' in data.columns:
    plt.figure(figsize=(6, 6))
    sns.countplot(data['fuelType'])
    plt.title('Distribution of Fuel Types')
    plt.show()
```

In [38]:

```
# 15. Saving cleaned data (if needed)
data.to_csv('C:/Users/Komal Bhati/Desktop/data visualization proj/cleaned_used_c


print('My Full Data Analysis Completed')
```

```
My Full Data Analysis Completed
```

In [ ]: