In [16]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
from datetime import datetime, timedelta
pd.options.display.float_format='{:.4f}'.format
plt.rcParams['figure.figsize'] = [8,8]
pd.set_option('display.max_columns', 500)
pd.set_option('display.max_colwidth', -1)
sns.set(style='darkgrid')
import matplotlib.ticker as ticker
import matplotlib.ticker as plticker
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
from sklearn.base import TransformerMixin
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
import statsmodels.api as sm
from sklearn.feature_selection import RFE
from sklearn.linear_model import LinearRegression
from statsmodels.stats.outliers_influence import variance_inflation_factor
from sklearn.metrics import r2_score
```

In [17]:
```
pip install autoviz --upgrade
```

```
Requirement already satisfied: autoviz in c:\python310\lib\site-packages
(0.1.58)
Requirement already satisfied: typing-extensions>=4.1.1 in c:\python310\li
b\site-packages (from autoviz) (4.5.0)
Requirement already satisfied: xgboost>=0.82 in c:\python310\lib\site-pack
ages (from autoviz) (1.7.5)
Requirement already satisfied: hvplot>=0.7.3 in c:\python310\lib\site-pack
ages (from autoviz) (0.8.3)
Requirement already satisfied: ipython in c:\python310\lib\site-packages
(from autoviz) (8.11.0)
Requirement already satisfied: fsspec>=0.8.3 in c:\python310\lib\site-pack
ages (from autoviz) (2023.4.0)
Requirement already satisfied: holoviews>=1.14.6 in c:\python310\lib\site-
packages (from autoviz) (1.14.9)
Requirement already satisfied: pandas in c:\python310\lib\site-packages (f
rom autoviz) (1.5.3)
Requirement already satisfied: jupyter in c:\python310\lib\site-packages
(from autoviz) (1.0.0)
Requirement already satisfied: pyamg in c:\python310\lib\site-packages (fr
```

In [18]:
```
pip install plotly
```

```
Requirement already satisfied: plotly in c:\python310\lib\site-packages (5.1
4.1)
Requirement already satisfied: packaging in c:\python310\lib\site-packages (f
rom plotly) (23.0)
Requirement already satisfied: tenacity>=6.2.0 in c:\python310\lib\site-packa
ges (from plotly) (8.2.2)
Note: you may need to restart the kernel to use updated packages.


[notice] A new release of pip available: 22.3.1 -> 23.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

In [19]:
```
import matplotlib.pyplot as plt
```

In [20]:
```
path = '../input/car-price-prediction/'
file = path + 'CarPrice_Assignment.csv'
file1 = path+ 'Data Dictionary - carprices.xlsx'
```

In [21]:
```
df_auto = pd.read_csv("Car Prediction.csv")

df_auto.head()
```

Out[21]:

	car_ID	symboling	CarName	fueltype	aspiration	doornumber	carbody	drivewheel	engi
0	1	3.0000	alfa-romero giulia	gas	std	two	convertible	rwd	
1	2	3.0000	alfa-romero stelvio	gas	std	two	convertible	rwd	
2	3	1.0000	alfa-romero Quadrifoglio	gas	std	two	hatchback	rwd	
3	4	2.0000	audi 100 ls	gas	std	four	sedan	fwd	
4	5	2.0000	audi 100ls	gas	std	four	sedan	4wd	

In [22]:
```
df_auto
```

Out[22]:

	car_ID	symboling	CarName	fueltype	aspiration	doornumber	carbody	drivewheel	ei
0	1	3.0000	alfa-romero giulia	gas	std	two	convertible	rwd	
1	2	3.0000	alfa-romero stelvio	gas	std	two	convertible	rwd	
2	3	1.0000	alfa-romero Quadrifoglio	gas	std	two	hatchback	rwd	
3	4	2.0000	audi 100 ls	gas	std	four	sedan	fwd	
4	5	2.0000	audi 100ls	gas	std	four	sedan	4wd	
...	...	...	...	...	...	...	...	...	
201	202	-1.0000	volvo 144ea	gas	turbo	four	sedan	rwd	
202	203	-1.0000	volvo 244dl	gas	std	four	sedan	rwd	
203	204	-1.0000	volvo 246	diesel	turbo	four	sedan	rwd	
204	205	-1.0000	volvo 264gl	gas	turbo	four	sedan	rwd	
205		NaN	NaN	NaN	NaN	NaN	NaN	NaN	

206 rows × 26 columns

In [23]:
```
df_auto.shape
```

Out[23]:
```
(206, 26)
```

In [24]:
```
df_auto.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 206 entries, 0 to 205
Data columns (total 26 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   car_ID            206 non-null    object 
 1   symboling         205 non-null    float64
 2   CarName           205 non-null    object 
 3   fueltype          205 non-null    object 
 4   aspiration        205 non-null    object 
 5   doornumber        205 non-null    object 
 6   carbody           205 non-null    object 
 7   drivewheel        205 non-null    object 
 8   enginelocation    205 non-null    object 
 9   wheelbase         205 non-null    float64
 10  carlength         205 non-null    float64
 11  carwidth          205 non-null    float64
 12  carheight         205 non-null    float64
 13  curbweight        205 non-null    float64
 14  enginetype        205 non-null    object 
 15  cylindernumber    205 non-null    object 
 16  enginesize        205 non-null    float64
 17  fuelsystem        205 non-null    object 
 18  boreratio         205 non-null    float64
 19  stroke            205 non-null    float64
 20  compressionratio  205 non-null    float64
 21  horsepower        205 non-null    float64
 22  peakrpm           205 non-null    float64
 23  citympg           205 non-null    float64
 24  highwaympg        205 non-null    float64
 25  price             205 non-null    float64
dtypes: float64(15), object(11)
memory usage: 42.0+ KB
```

In [25]:
```
df_auto.describe()
```

Out[25]:

	symboling	wheelbase	carlength	carwidth	carheight	curbweight	enginesize	boreratio
count	205.0000	205.0000	205.0000	205.0000	205.0000	205.0000	205.0000	205.0000
mean	0.8341	98.7566	174.0493	65.9078	53.7249	2555.5659	126.9073	3.3298
std	1.2453	6.0218	12.3373	2.1452	2.4435	520.6802	41.6427	0.2708
min	-2.0000	86.6000	141.1000	60.3000	47.8000	1488.0000	61.0000	2.5400
25%	0.0000	94.5000	166.3000	64.1000	52.0000	2145.0000	97.0000	3.1500
50%	1.0000	97.0000	173.2000	65.5000	54.1000	2414.0000	120.0000	3.3100
75%	2.0000	102.4000	183.1000	66.9000	55.5000	2935.0000	141.0000	3.5800
max	3.0000	120.9000	208.1000	72.3000	59.8000	4066.0000	326.0000	3.9400

In [26]:
```
df_auto = df_auto.drop('car_ID',axis=1)
```

In [27]:
```
df_null = df_auto.isna().mean().round(4) * 100

df_null.sort_values(ascending=False).head()
```

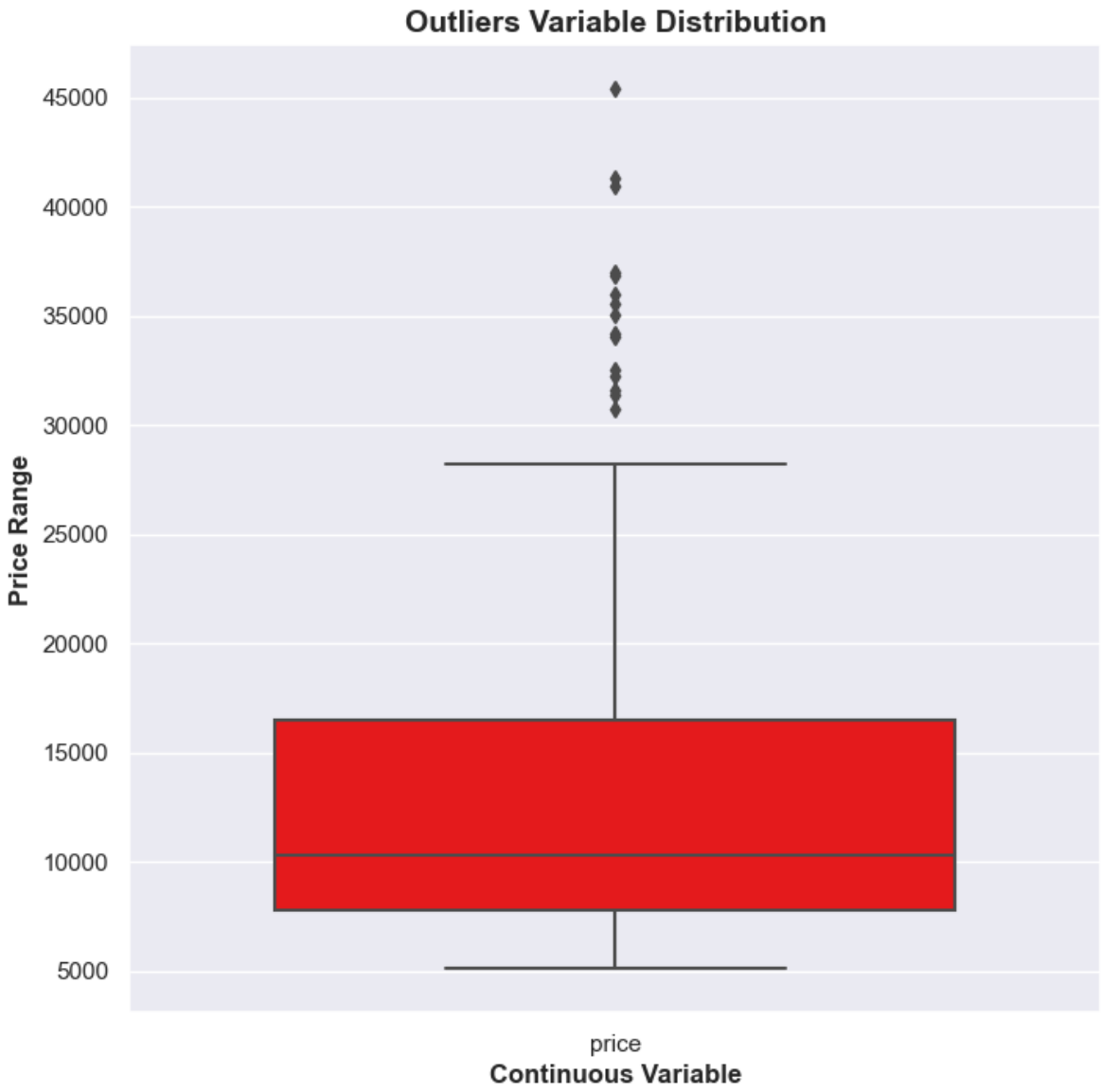
Out[27]:
```
symboling    0.4900
enginetype   0.4900
highwaympg   0.4900
citympg      0.4900
peakrpm      0.4900
dtype: float64
```

In [28]:
```
df_auto.dtypes
```

Out[28]:
```
symboling           float64
CarName              object
fueltype             object
aspiration           object
doornumber           object
carbody              object
drivewheel           object
enginelocation       object
wheelbase           float64
carlength           float64
carwidth            float64
carheight           float64
curbweight          float64
enginetype           object
cylindernumber       object
enginesize          float64
fuelsystem           object
boreratio           float64
stroke              float64
compressionratio    float64
horsepower          float64
peakrpm             float64
citympg             float64
highwaympg          float64
price               float64
dtype: object
```

In [29]:
```
outliers = ['price']
plt.rcParams['figure.figsize'] = [8,8]
sns.boxplot(data=df_auto[outliers], orient="v", palette="Set1" ,whis=1.5,satur
plt.title("Outliers Variable Distribution", fontsize = 14, fontweight = 'bold'
plt.ylabel("Price Range", fontweight = 'bold')
plt.xlabel("Continuous Variable", fontweight = 'bold')
df_auto.shape
```

Out[29]: (206, 25)



In [30]:
```
df_auto['CarName'].unique()
```

Out[30]:
```
array(['alfa-romero giulia', 'alfa-romero stelvio',
       'alfa-romero Quadrifoglio', 'audi 100 ls', 'audi 100ls',
       'audi fox', 'audi 5000', 'audi 4000', 'audi 5000s (diesel)',
       'bmw 320i', 'bmw x1', 'bmw x3', 'bmw z4', 'bmw x4', 'bmw x5',
       'chevrolet impala', 'chevrolet monte carlo', 'chevrolet vega 2300',
       'dodge rampage', 'dodge challenger se', 'dodge d200',
       'dodge monaco (sw)', 'dodge colt hardtop', 'dodge colt (sw)',
       'dodge coronet custom', 'dodge dart custom',
       'dodge coronet custom (sw)', 'honda civic', 'honda civic cvcc',
       'honda accord cvcc', 'honda accord lx', 'honda civic 1500 gl',
       'honda accord', 'honda civic 1300', 'honda prelude',
       'honda civic (auto)', 'isuzu MU-X', 'isuzu D-Max ',
       'isuzu D-Max V-Cross', 'jaguar xj', 'jaguar xf', 'jaguar xk',
       'maxda rx3', 'maxda glc deluxe', 'mazda rx2 coupe', 'mazda rx-4',
       'mazda glc deluxe', 'mazda 626', 'mazda glc', 'mazda rx-7 gs',
       'mazda glc 4', 'mazda glc custom l', 'mazda glc custom',
       'buick electra 225 custom', 'buick century luxus (sw)',
       'buick century', 'buick skyhawk', 'buick opel isuzu deluxe',
       'buick skylark', 'buick century special',
       'buick regal sport coupe (turbo)', 'mercury cougar',
       'mitsubishi mirage', 'mitsubishi lancer', 'mitsubishi outlander',
       'mitsubishi g4', 'mitsubishi mirage g4', 'mitsubishi montero',
       'mitsubishi pajero', 'Nissan versa', 'nissan gt-r', 'nissan rogue',
       'nissan latio', 'nissan titan', 'nissan leaf', 'nissan juke',
       'nissan note', 'nissan clipper', 'nissan nv200', 'nissan dayz',
       'nissan fuga', 'nissan otti', 'nissan teana', 'nissan kicks',
       'peugeot 504', 'peugeot 304', 'peugeot 504 (sw)', 'peugeot 604sl',
       'peugeot 505s turbo diesel', 'plymouth fury iii',
       'plymouth cricket', 'plymouth satellite custom (sw)',
       'plymouth fury gran sedan', 'plymouth valiant', 'plymouth duster',
       'porsche macan', 'porcshce panamera', 'porsche cayenne',
       'porsche boxter', 'renault 12tl', 'renault 5 gtl', 'saab 99e',
       'saab 99le', 'saab 99gle', 'subaru', 'subaru dl', 'subaru brz',
       'subaru baja', 'subaru r1', 'subaru r2', 'subaru trezia',
       'subaru tribeca', 'toyota corona mark ii', 'toyota corona',
       'toyota corolla 1200', 'toyota corona hardtop',
       'toyota corolla 1600 (sw)', 'toyota carina', 'toyota mark ii',
       'toyota corolla', 'toyota corolla liftback',
       'toyota celica gt liftback', 'toyota corolla tercel',
       'toyota corona liftback', 'toyota starlet', 'toyota tercel',
       'toyota cressida', 'toyota celica gt', 'toyouta tercel',
       'vokswagen rabbit', 'volkswagen 1131 deluxe sedan',
       'volkswagen model 111', 'volkswagen type 3', 'volkswagen 411 (sw)',
       'volkswagen super beetle', 'volkswagen dasher', 'vw dasher',
       'vw rabbit', 'volkswagen rabbit', 'volkswagen rabbit custom',
       'volvo 145e (sw)', 'volvo 144ea', 'volvo 244dl', 'volvo 245',
       'volvo 264gl', 'volvo diesel', 'volvo 246', nan], dtype=object)
```

In [31]:
```
df_auto['CarName'] = df_auto['CarName'].replace({'maxda': 'mazda', 'nissan': '
                                'vokswagen': 'volkswagen', 'vw': 'volkswagen'})
```

In [32]:
```
df_auto['symboling'] = df_auto['symboling'].astype(str)
```

In [33]:
```
df_auto.loc[df_auto.duplicated()]
```

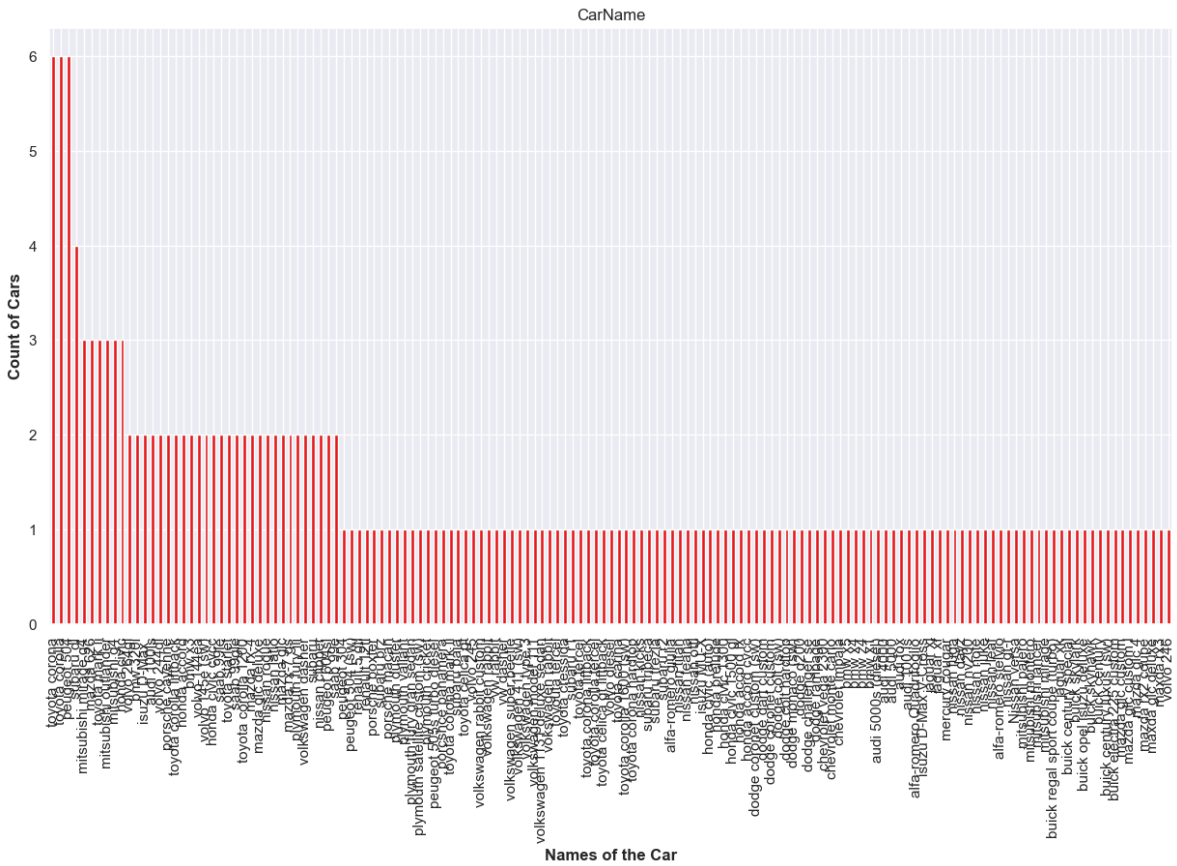
Out[33]:

	symboling	CarName	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	v

In [34]:
```
cat_col = df_auto.select_dtypes(include=['object']).columns
num_col = df_auto.select_dtypes(exclude=['object']).columns
df_cat = df_auto[cat_col]
df_num = df_auto[num_col]
```

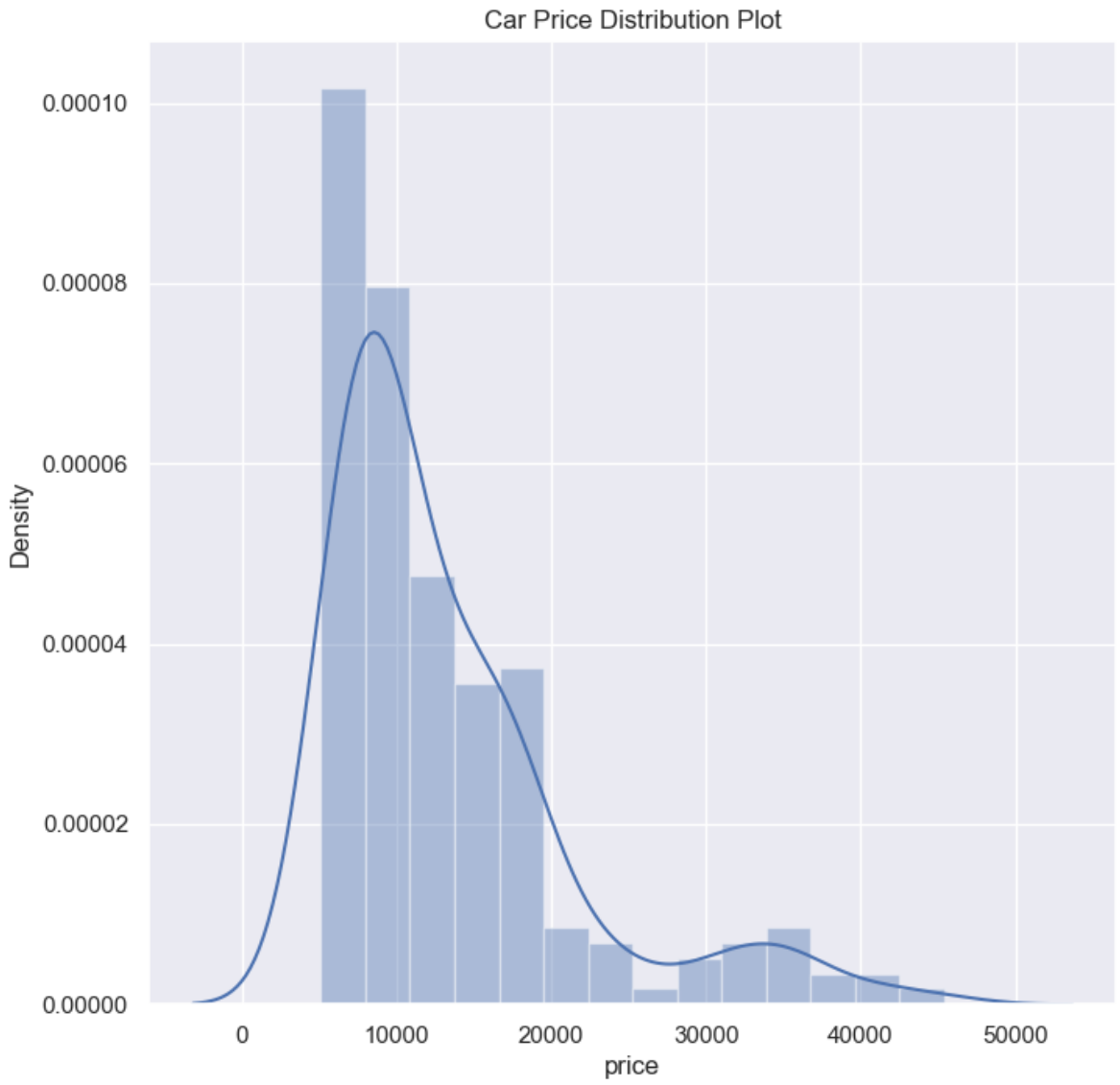
In [35]:
```
plt.rcParams['figure.figsize'] = [15,8]
ax=df_auto['CarName'].value_counts().plot(kind='bar',stacked=True, colormap =
ax.title.set_text('CarName')
plt.xlabel("Names of the Car",fontweight = 'bold')
plt.ylabel("Count of Cars",fontweight = 'bold')
```

Out[35]: Text(0, 0.5, 'Count of Cars')

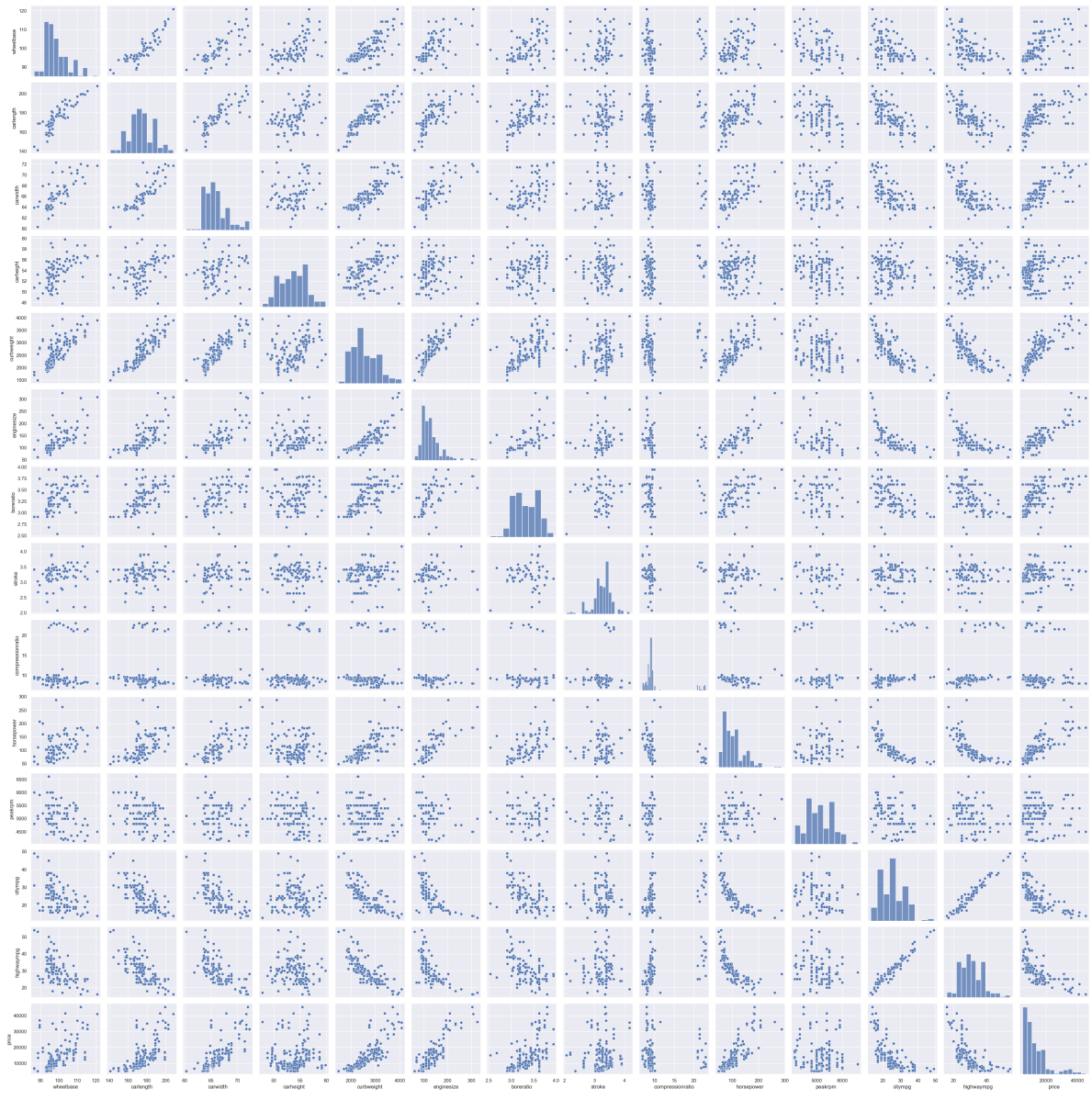


In [36]:
```
plt.figure(figsize=(8,8))

plt.title('Car Price Distribution Plot')
sns.distplot(df_auto['price'])
```

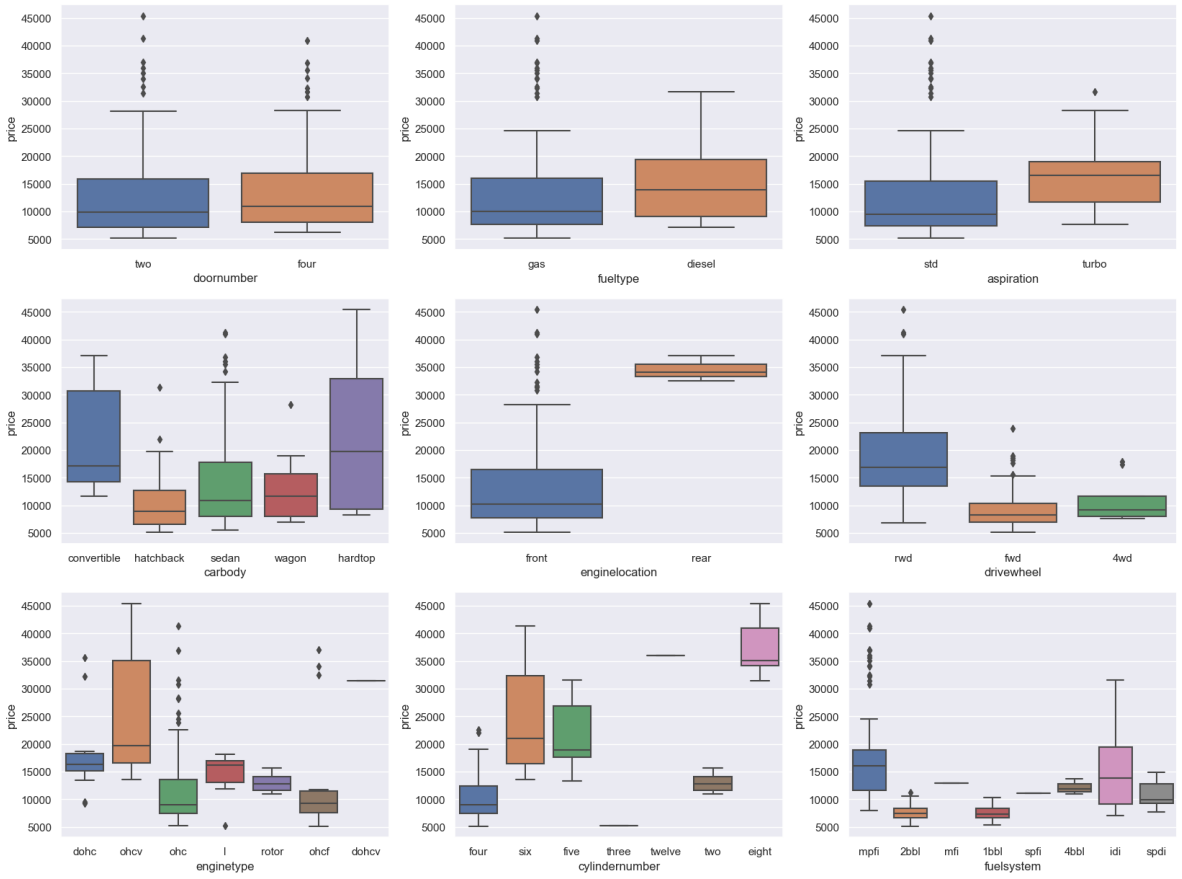
Out[36]:
```
<Axes: title={'center': 'Car Price Distribution Plot'}, xlabel='price', ylabe
l='Density'>
```



In [37]:
```
ax = sns.pairplot(df_auto[num_col])
```



In [38]:
```
plt.figure(figsize=(20, 15))
plt.subplot(3,3,1)
sns.boxplot(x = 'doornumber', y = 'price', data = df_auto)
plt.subplot(3,3,2)
sns.boxplot(x = 'fueltype', y = 'price', data = df_auto)
plt.subplot(3,3,3)
sns.boxplot(x = 'aspiration', y = 'price', data = df_auto)
plt.subplot(3,3,4)
sns.boxplot(x = 'carbody', y = 'price', data = df_auto)
plt.subplot(3,3,5)
sns.boxplot(x = 'enginelocation', y = 'price', data = df_auto)
plt.subplot(3,3,6)
sns.boxplot(x = 'drivewheel', y = 'price', data = df_auto)
plt.subplot(3,3,7)
sns.boxplot(x = 'enginetype', y = 'price', data = df_auto)
plt.subplot(3,3,8)
sns.boxplot(x = 'cylindernumber', y = 'price', data = df_auto)
plt.subplot(3,3,9)
sns.boxplot(x = 'fuelsystem', y = 'price', data = df_auto)
plt.show()
```

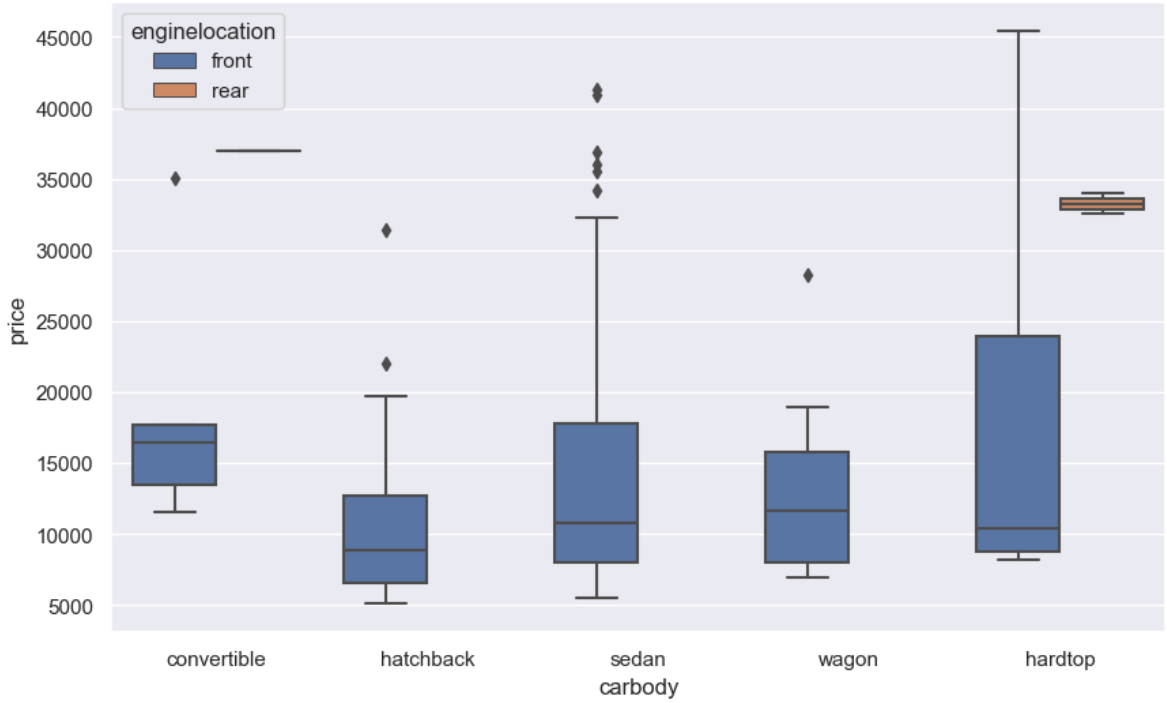


In [39]:
```
pip install matplotlib
```

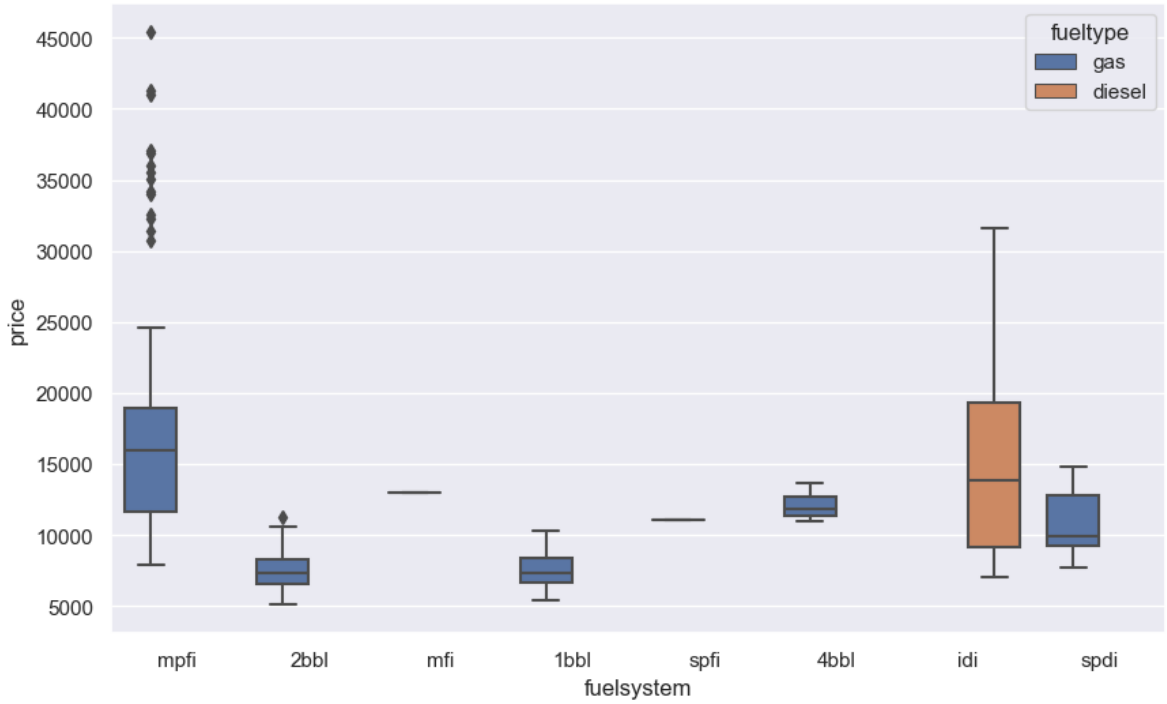
```
Requirement already satisfied: matplotlib in c:\python310\lib\site-packages
(3.7.1)
Requirement already satisfied: cycler>=0.10 in c:\python310\lib\site-packages
(from matplotlib) (0.11.0)
Requirement already satisfied: contourpy>=1.0.1 in c:\python310\lib\site-pack
ages (from matplotlib) (1.0.7)
Requirement already satisfied: python-dateutil>=2.7 in c:\python310\lib\site-
packages (from matplotlib) (2.8.2)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\python310\lib\site-pac
kages (from matplotlib) (1.4.4)
Requirement already satisfied: pillow>=6.2.0 in c:\python310\lib\site-package
s (from matplotlib) (9.5.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\python310\lib\site-pac
kages (from matplotlib) (4.39.3)
Requirement already satisfied: packaging>=20.0 in c:\python310\lib\site-packa
ges (from matplotlib) (23.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\python310\lib\site-pack
ages (from matplotlib) (3.0.9)
Requirement already satisfied: numpy>=1.20 in c:\python310\lib\site-packages
(from matplotlib) (1.24.2)
Requirement already satisfied: six>=1.5 in c:\python310\lib\site-packages (fr
om python-dateutil>=2.7->matplotlib) (1.16.0)
Note: you may need to restart the kernel to use updated packages.


[notice] A new release of pip available: 22.3.1 -> 23.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

In [44]:
```
import matplotlib.pyplot as plt
```

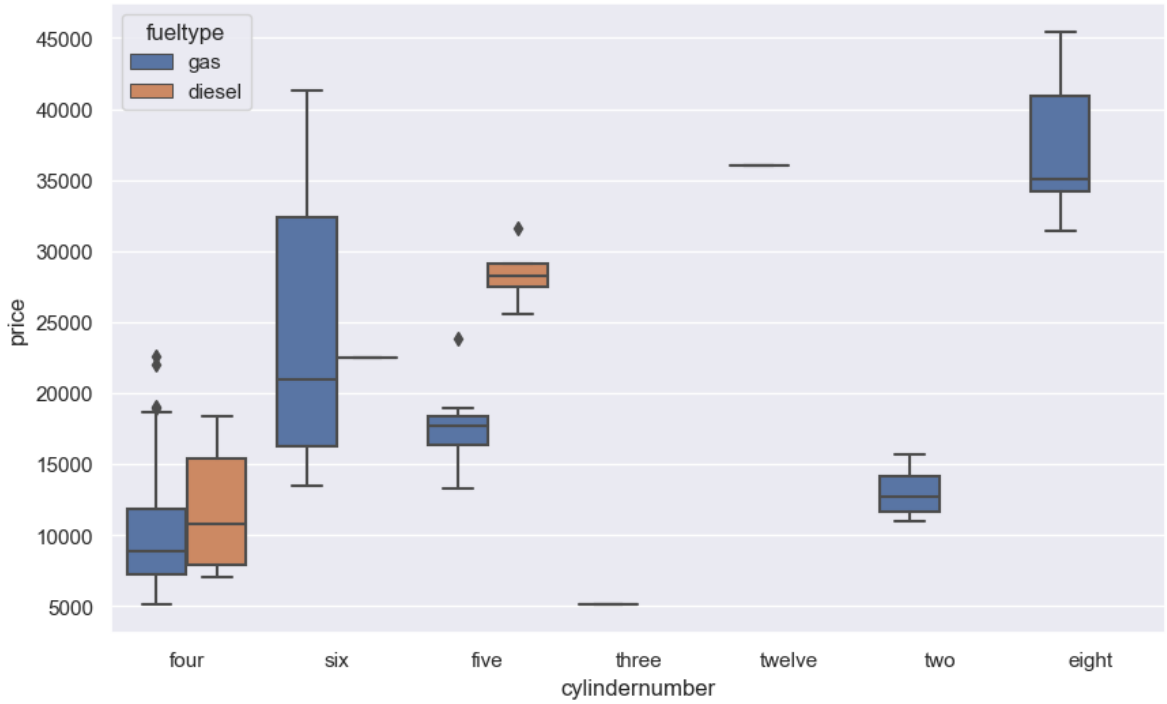
In [47]:
```
plt.figure(figsize = (10, 6))
sns.boxplot(x = 'carbody', y = 'price', hue = 'enginelocation', data = df_auto
plt.show()
```



In [46]:
```
plt.figure(figsize = (10, 6))
sns.boxplot(x = 'fuelsystem', y = 'price', hue = 'fueltype', data = df_auto)
plt.show()
```

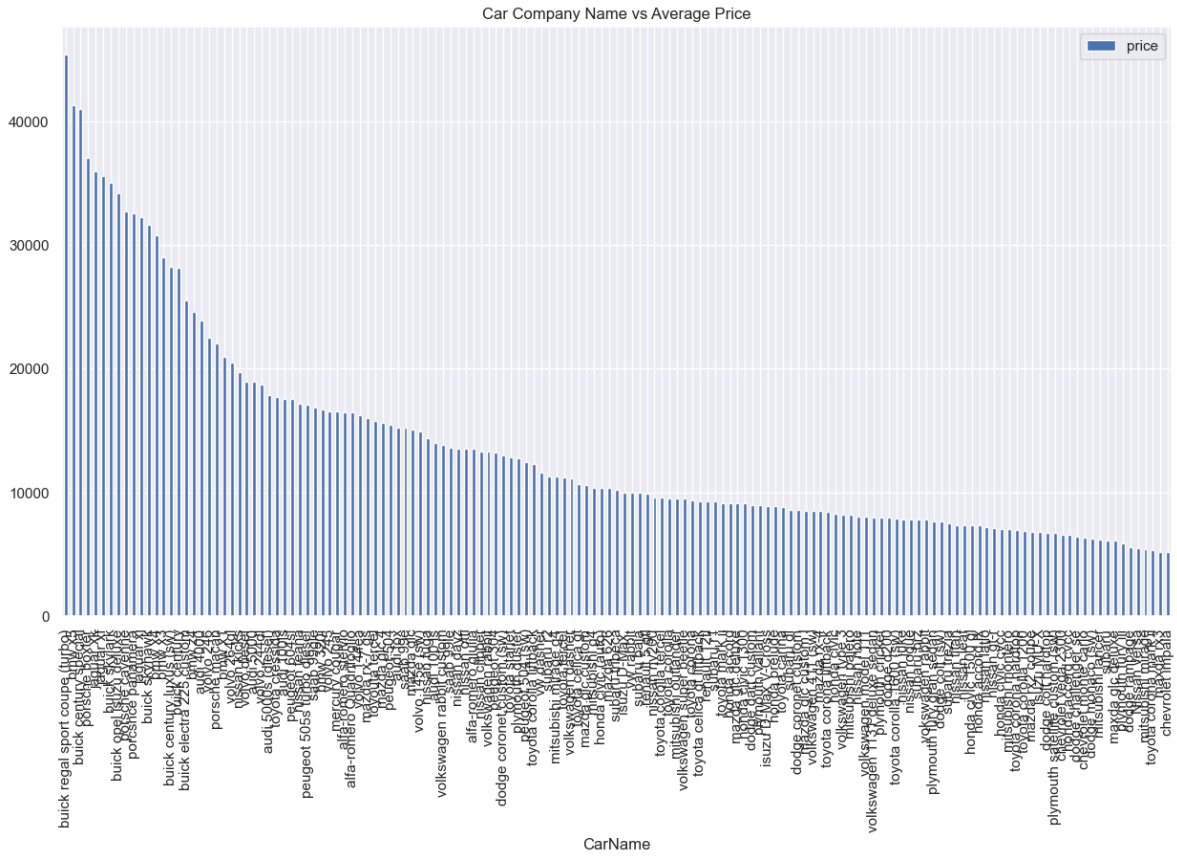


In [48]:
```
plt.figure(figsize = (10, 6))
sns.boxplot(x = 'cylindernumber', y = 'price', hue = 'fueltype', data = df_aut
plt.show()
```



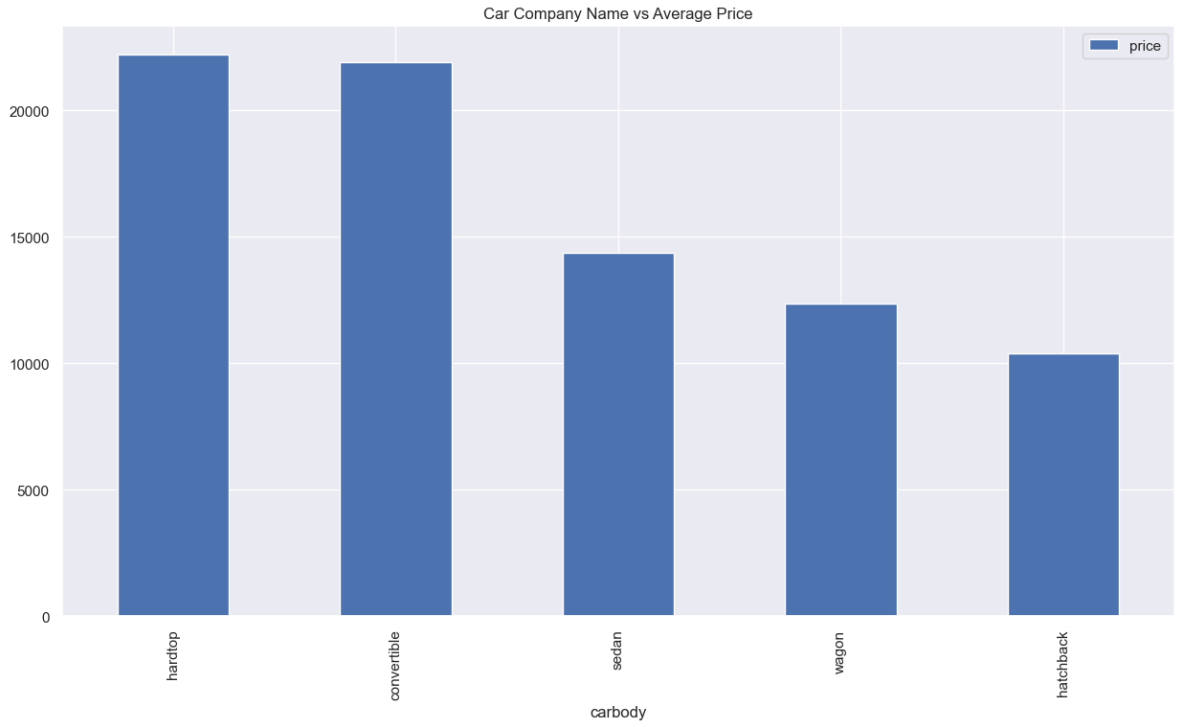
In [50]:
```
plt.figure(figsize=(10, 6))

df_autox = pd.DataFrame(df_auto.groupby(['CarName'])['price'].mean().sort_valu
df_autox.plot.bar()
plt.title('Car Company Name vs Average Price')
plt.show()
```

```
<Figure size 1000x600 with 0 Axes>
```



In [51]:
```
plt.figure(figsize=(20, 6))

df_autoy = pd.DataFrame(df_auto.groupby(['carbody'])['price'].mean().sort_valu
df_autoy.plot.bar()
plt.title('Car Company Name vs Average Price')
plt.show()
```

```
<Figure size 2000x600 with 0 Axes>
```



In [60]:
```
np.random.seed(0)
df_train, df_test = train_test_split(df_auto, train_size = 0.7, test_size = 0.
```

In [61]:
```
df_train.head()
```

Out[61]:

	symboling	CarName	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocati
123	-1.0	plymouth valiant	gas	std	four	wagon	fwd	fr
110	0.0	peugeot 504	diesel	turbo	four	wagon	rwd	fr
167	2.0	toyota corona liftback	gas	std	two	hardtop	rwd	fr
1	3.0	alfa-romero stelvio	gas	std	two	convertible	rwd	fr
166	1.0	toyota corolla tercel	gas	std	two	hatchback	rwd	fr

In [62]:
```
scaler = preprocessing.StandardScaler()
```

In [63]:
```
sig_num_col = ['wheelbase','carlength','carwidth','curbweight','enginesize','b
```

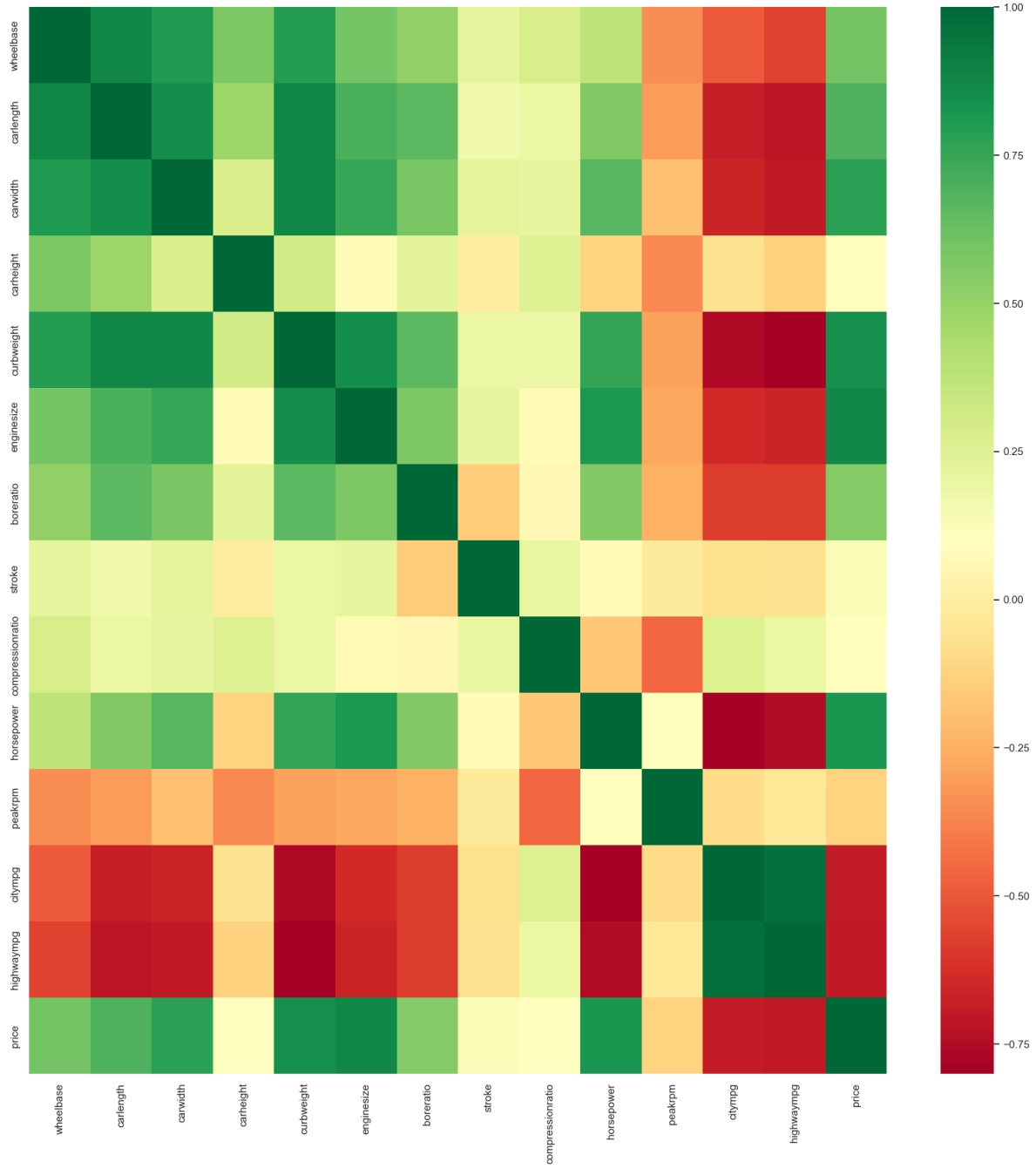
In [64]:
```
import warnings
warnings.filterwarnings("ignore")

df_train[sig_num_col] = scaler.fit_transform(df_train[sig_num_col])
```

In [65]:
```
df_train.head()
```

Out[65]:

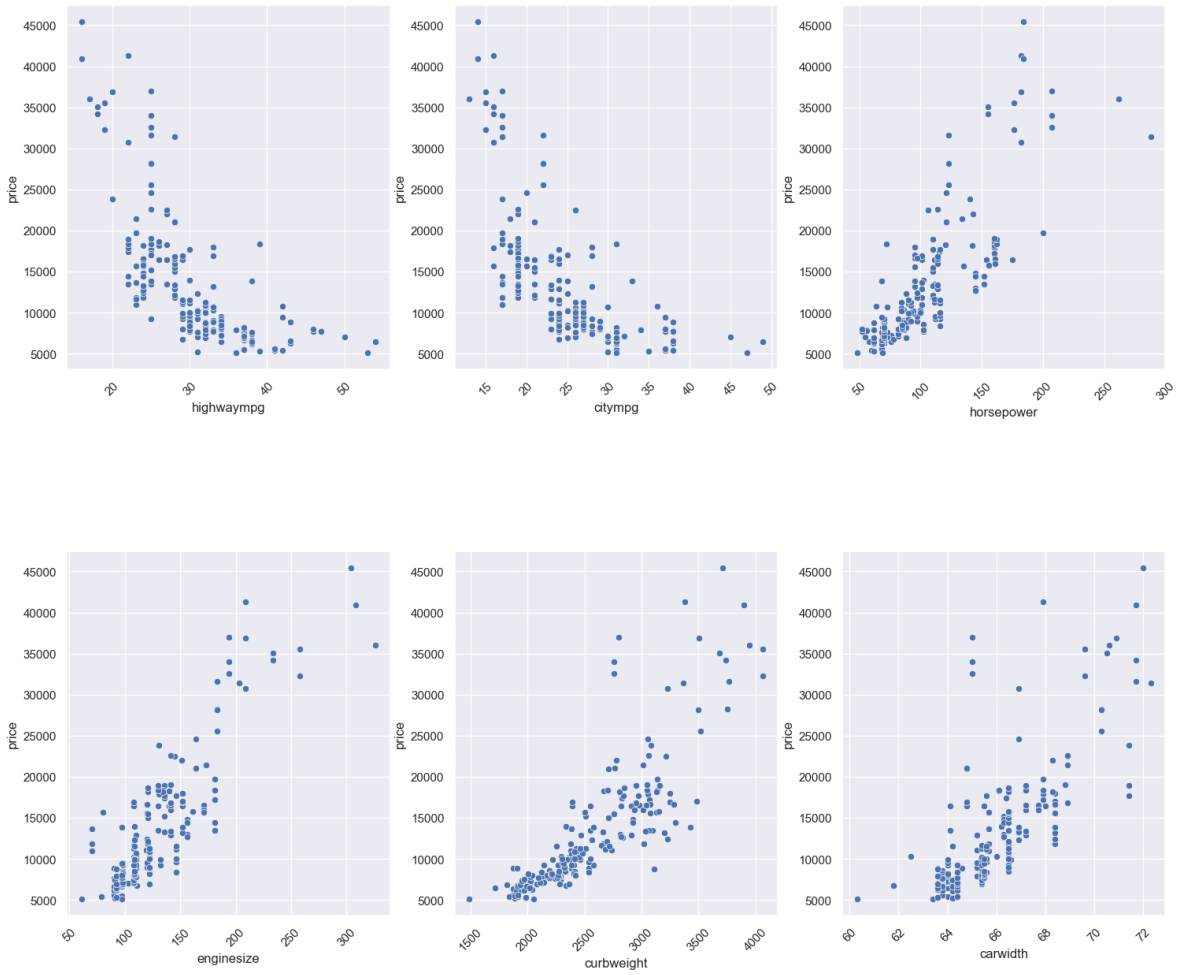
	symboling	CarName	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocati
123	-1.0	plymouth valiant	gas	std	four	wagon	fwd	fr
110	0.0	peugeot 504	diesel	turbo	four	wagon	rwd	fr
167	2.0	toyota corona liftback	gas	std	two	hardtop	rwd	fr
1	3.0	alfa-romero stelvio	gas	std	two	convertible	rwd	fr
166	1.0	toyota corolla tercel	gas	std	two	hatchback	rwd	fr

In [66]:
```
plt.figure(figsize = (20, 20))
sns.heatmap(df_train.corr(), cmap="RdYlGn")
plt.show()
```



In [67]:
```
col = ['highwaympg','citympg','horsepower','enginesize','curbweight','carwidth
```

In [68]:
```
fig,axes = plt.subplots(2,3,figsize=(18,15))
for seg,col in enumerate(col):
    x,y = seg//3,seg%3
    an=sns.scatterplot(x=col, y='price' ,data=df_auto, ax=axes[x,y])
    plt.setp(an.get_xticklabels(), rotation=45)

plt.subplots_adjust(hspace=0.5)
```



In [69]:
```
y_train = df_train.pop('price')
X_train = df_train
```

In [70]:
```
X_train_1 = X_train['horsepower']
```

In [72]:
```
X_train_1c = sm.add_constant(X_train_1)
```

In [75]:
```
plt.scatter(X_train_1c.iloc[:, 1], y_train)
plt.plot(X_train_1c.iloc[:, 1], 0.8062*X_train_1c.iloc[:, 1], 'r')
plt.show()
```

In [77]:
```
X_train_2 = X_train[['horsepower', 'curbweight']]
```

In [88]:
```
X_train_2c = sm.add_constant(X_train_2)
```

In [90]:
```
df_auto.isnull().sum()
```

Out[90]:
```
symboling           0
CarName             1
fueltype            1
aspiration          1
doornumber          1
carbody             1
drivewheel          1
enginelocation      1
wheelbase           1
carlength           1
carwidth            1
carheight           1
curbweight          1
enginetype          1
cylindernumber      1
enginesize          1
fuelsystem          1
boreratio           1
stroke              1
compressionratio    1
horsepower          1
peakrpm             1
citympg             1
highwaympg          1
price               1
dtype: int64
```

In [ ]:

In [ ]: