Linear Regression Model

In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics
```

In [2]:

```
car_dataset = pd.read_csv("Car_Price.csv")
```

In [3]:

```
car_dataset.head()
```

Out[3]:

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

5 rows × 26 columns

In [4]:

```
car_dataset.shape
```

Out[4]:

```
(205, 26)
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 26 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   car_ID            205 non-null    int64  
 1   symboling         205 non-null    int64  
 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    int64  
 14  enginetype        205 non-null    object 
 15  cylindernumber    205 non-null    object 
 16  enginesize        205 non-null    int64  
 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    int64  
 22  peakrpm           205 non-null    int64  
 23  citympg           205 non-null    int64  
 24  highwaympg        205 non-null    int64  
 25  price             205 non-null    float64
dtypes: float64(8), int64(8), object(10)
memory usage: 41.8+ KB
```

In [6]:
```
car_dataset.describe()
```

Out[6]:

	car_ID	symboling	wheelbase	carlength	carwidth	carheight	curbweight	en
count	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	20
mean	103.000000	0.834146	98.756585	174.049268	65.907805	53.724878	2555.565854	12
std	59.322565	1.245307	6.021776	12.337289	2.145204	2.443522	520.680204	4
min	1.000000	-2.000000	86.600000	141.100000	60.300000	47.800000	1488.000000	6
25%	52.000000	0.000000	94.500000	166.300000	64.100000	52.000000	2145.000000	9
50%	103.000000	1.000000	97.000000	173.200000	65.500000	54.100000	2414.000000	12
75%	154.000000	2.000000	102.400000	183.100000	66.900000	55.500000	2935.000000	14
max	205.000000	3.000000	120.900000	208.100000	72.300000	59.800000	4066.000000	32

Extracting Some Features from the Dataset

```
In [7]: car_dataset.CarName.unique()
```

```
Out[7]: 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'], dtype=object)
```

```
In [8]: car_dataset.price.sum()
```

```
Out[8]: 2721725.667
```

```
In [9]: car_dataset.carbody.unique()
```

```
Out[9]: array(['convertible', 'hatchback', 'sedan', 'wagon', 'hardtop'],
              dtype=object)
```

In [10]:

```
car_dataset.isnull().sum()
```

Out[10]:

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

In [11]:

```
print(car_dataset.fueltype.value_counts())
print(car_dataset.carbody.value_counts())
```

```
gas       185
diesel     20
Name: fueltype, dtype: int64
sedan          96
hatchback      70
wagon          25
hardtop         8
convertible     6
Name: carbody, dtype: int64
```

Training the Model

In [12]:

```
y=car_dataset[["price"]]
x=car_dataset[["enginesize"]]
```

In [13]:

```
lm = LinearRegression()
```

In [15]:

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random
_state=101)
```

In [16]:

```
lm.fit(x_train,y_train)
```

Out[16]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

In [17]:

```
print(lm.intercept_,lm.coef_)
```

```
[-7107.69540612] [[161.34743434]]
```

Predicting the Value After the Training Phase

In [18]:

```
pred=lm.predict(x_test)
print(pred[0:10])
```

```
[[ 7413.57368461]
 [10479.1749371 ]
 [20482.71586628]
 [ 8704.35315935]
 [12253.99671486]
 [22096.19020969]
 [12415.3441492 ]
 [ 8543.005725  ]
 [10640.52237144]
 [ 7736.2685533 ]]
```

In [19]:

```
training_pred=lm.predict(x_test)
```

In [20]:

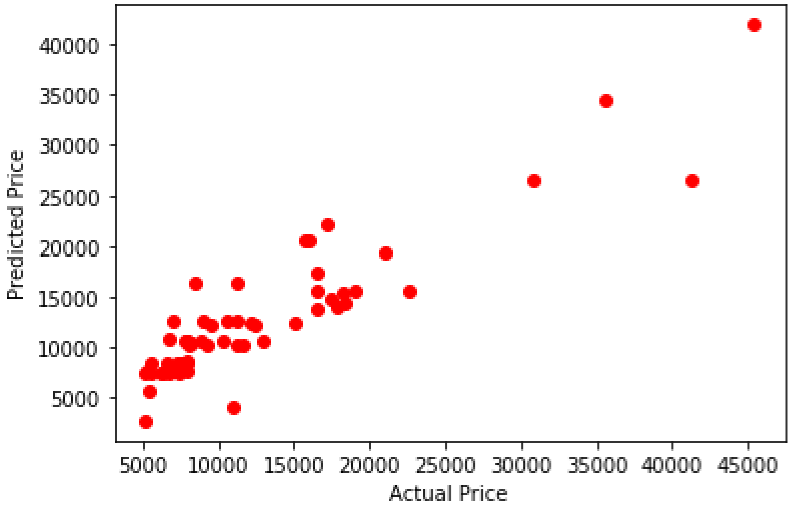
```
error_score=metrics.r2_score(y_test,training_pred)
print(error_score)
```

```
0.8258154601020361
```

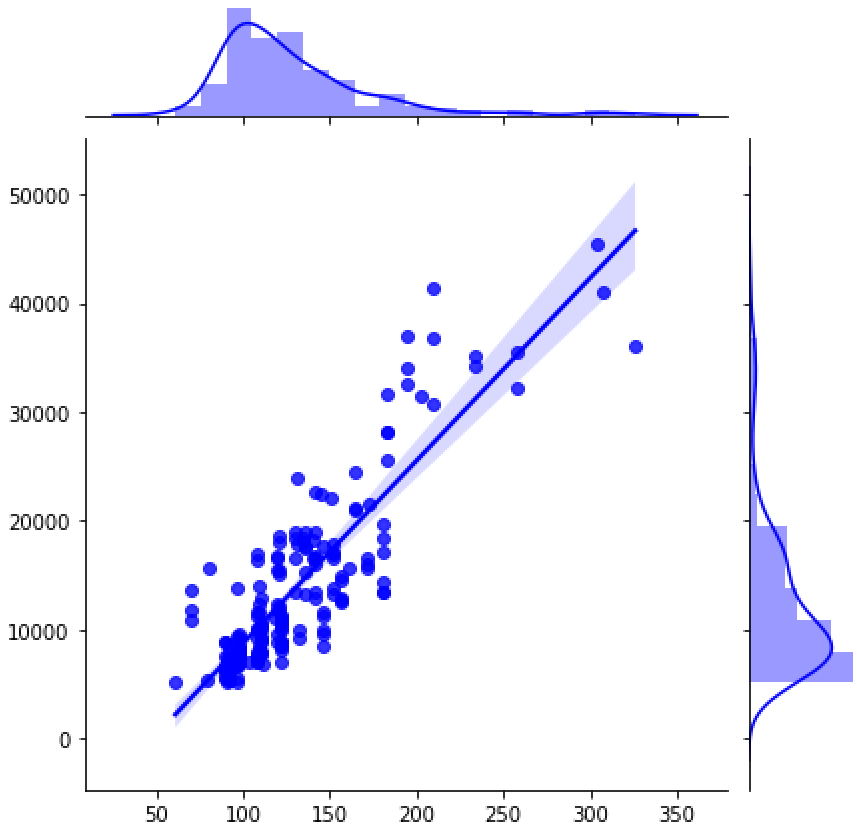
Plotting Some Graphs after the actual and the predicted Values

In [21]:

```
plt.scatter(y_test,training_pred,color="red")
plt.xlabel("Actual Price")
plt.ylabel("Predicted Price")
plt.show()
```

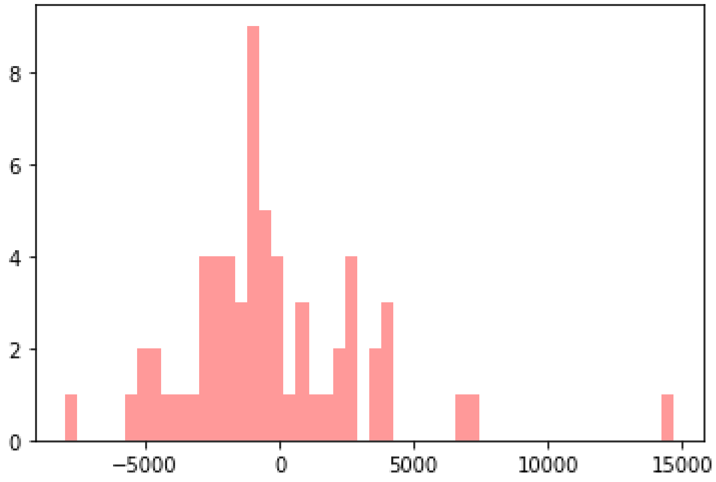


```
In [22]: sns.jointplot(y=car_dataset[["price"]],x=car_dataset[["enginesize",]],data=car_
         dataset,kind="reg",color="blue")

Out[22]: <seaborn.axisgrid.JointGrid at 0x11b9f445488>
```

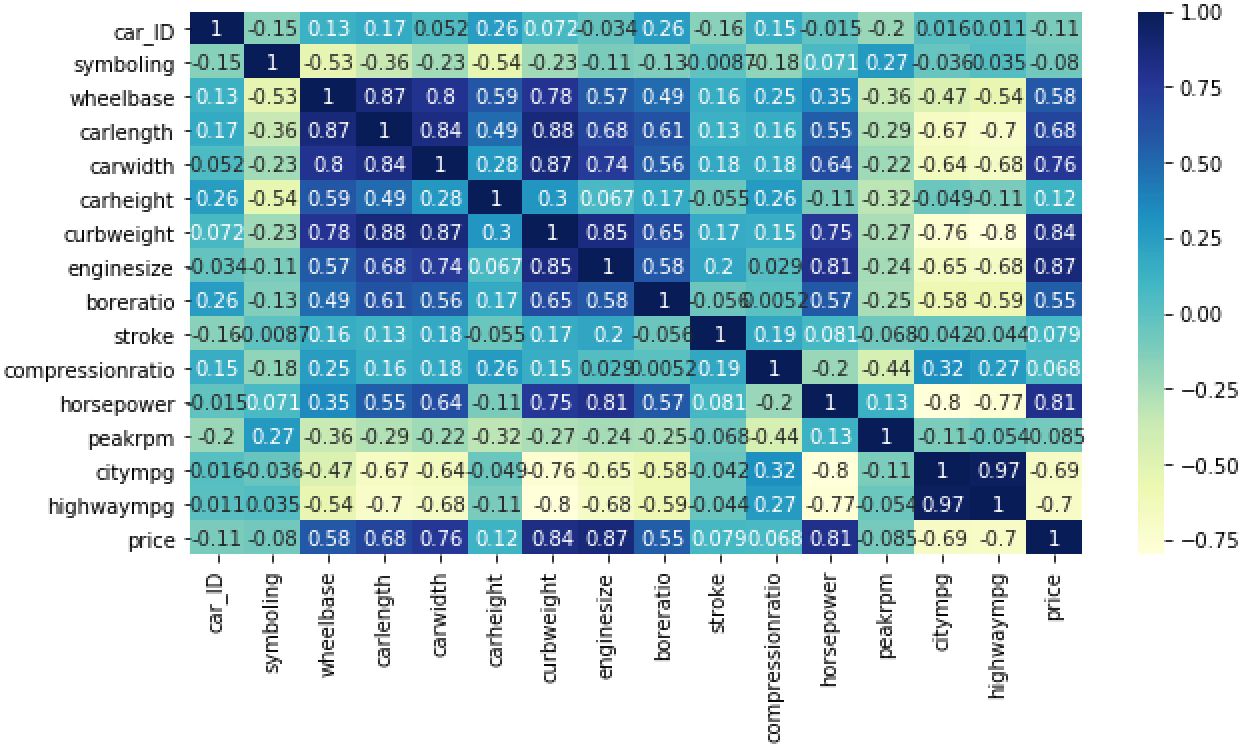


Plotting Some Graphs related to the Dataset

```
In [23]: sns.distplot(y_test-pred,bins=50,color="red",kde=False)

Out[23]: <matplotlib.axes._subplots.AxesSubplot at 0x11b9f665d48>
```

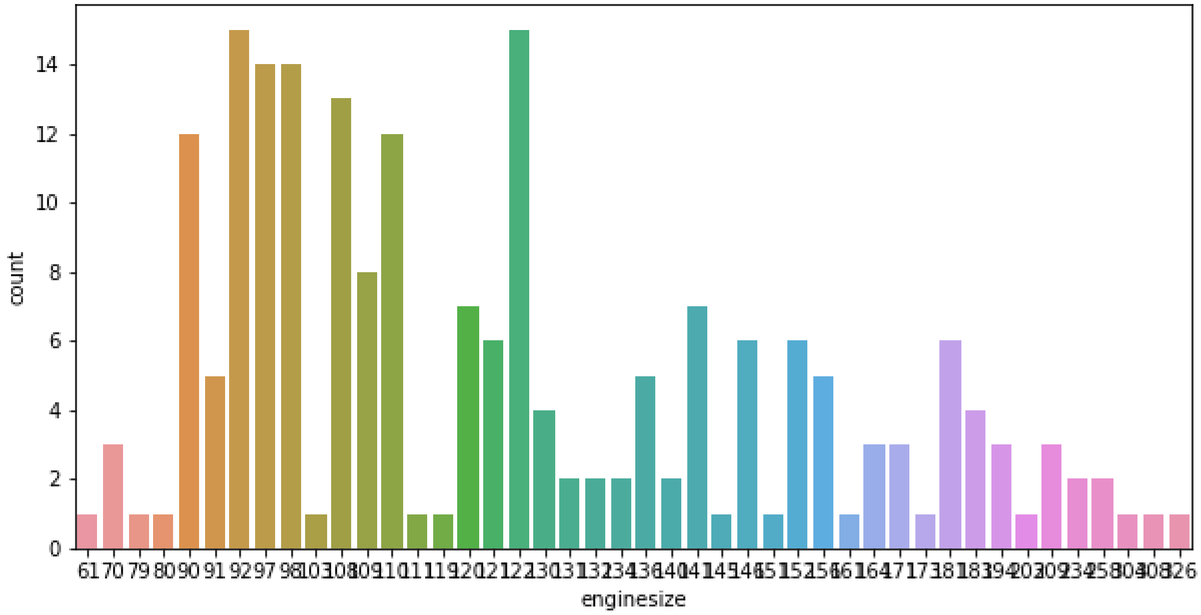


In [24]:
```
plt.figure(figsize = (10, 5))
sns.heatmap(car_dataset.corr(), annot = True, cmap="YlGnBu")
plt.show()
```



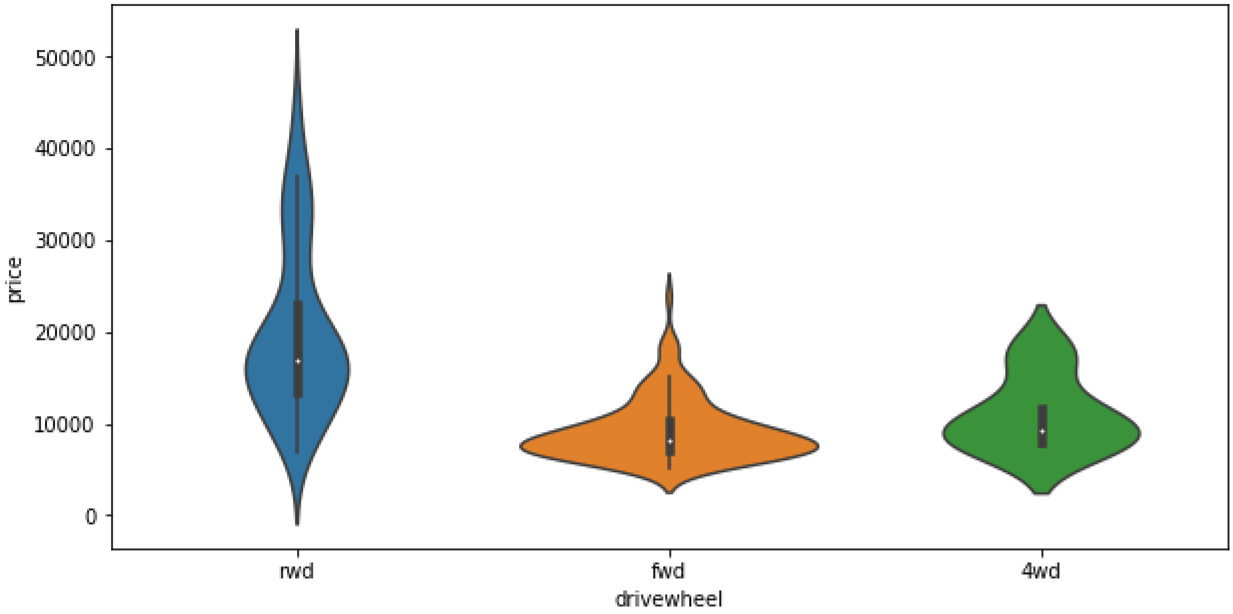
	car_ID	symboling	wheelbase	carlength	carwidth	carheight	curbweight	enginesize	boreratio	stroke	compressionratio	horsepower	peakrpm	citympg	highwaympg	price
car_ID	1	-0.15	0.13	0.17	0.052	0.26	0.072	-0.034	0.26	-0.16	0.15	-0.015	-0.2	0.016	0.011	-0.11
symboling	-0.15	1	-0.53	-0.36	-0.23	-0.54	-0.23	-0.11	-0.13	0.0087	-0.18	0.071	0.27	-0.036	0.035	-0.08
wheelbase	0.13	-0.53	1	0.87	0.8	0.59	0.78	0.57	0.49	0.16	0.25	0.35	-0.36	-0.47	-0.54	0.58
carlength	0.17	-0.36	0.87	1	0.84	0.49	0.88	0.68	0.61	0.13	0.16	0.55	-0.29	-0.67	-0.7	0.68
carwidth	0.052	-0.23	0.8	0.84	1	0.28	0.87	0.74	0.56	0.18	0.18	0.64	-0.22	-0.64	-0.68	0.76
carheight	0.26	-0.54	0.59	0.49	0.28	1	0.3	0.067	0.17	-0.055	0.26	-0.11	-0.32	-0.049	-0.11	0.12
curbweight	0.072	-0.23	0.78	0.88	0.87	0.3	1	0.85	0.65	0.17	0.15	0.75	-0.27	-0.76	-0.8	0.84
enginesize	-0.034	-0.11	0.57	0.68	0.74	0.067	0.85	1	0.58	0.2	0.029	0.81	-0.24	-0.65	-0.68	0.87
boreratio	0.26	-0.13	0.49	0.61	0.56	0.17	0.65	0.58	1	-0.056	0.0052	0.57	-0.25	-0.58	-0.59	0.55
stroke	-0.16	0.0087	0.16	0.13	0.18	-0.055	0.17	0.2	-0.056	1	0.19	0.081	-0.068	0.042	-0.044	0.079
compressionratio	0.15	-0.18	0.25	0.16	0.18	0.26	0.15	0.029	0.0052	0.19	1	-0.2	-0.44	0.32	0.27	0.068
horsepower	-0.015	0.071	0.35	0.55	0.64	-0.11	0.75	0.81	0.57	0.081	-0.2	1	0.13	-0.8	-0.77	0.81
peakrpm	-0.2	0.27	-0.36	-0.29	-0.22	-0.32	-0.27	-0.24	-0.25	-0.068	-0.44	0.13	1	-0.11	-0.054	-0.085
citympg	-0.016	-0.036	-0.47	-0.67	-0.64	-0.049	-0.76	-0.65	-0.58	-0.042	0.32	-0.8	-0.11	1	0.97	-0.69
highwaympg	0.011	0.035	-0.54	-0.7	-0.68	-0.11	-0.8	-0.68	-0.59	-0.044	0.27	-0.77	-0.054	0.97	1	-0.7
price	-0.11	-0.08	0.58	0.68	0.76	0.12	0.84	0.87	0.55	0.079	0.068	0.81	-0.085	-0.69	-0.7	1

In [25]:
```
plt.figure(figsize = (10,5))
sns.countplot(x="enginesize",data=car_dataset)
plt.show()
```



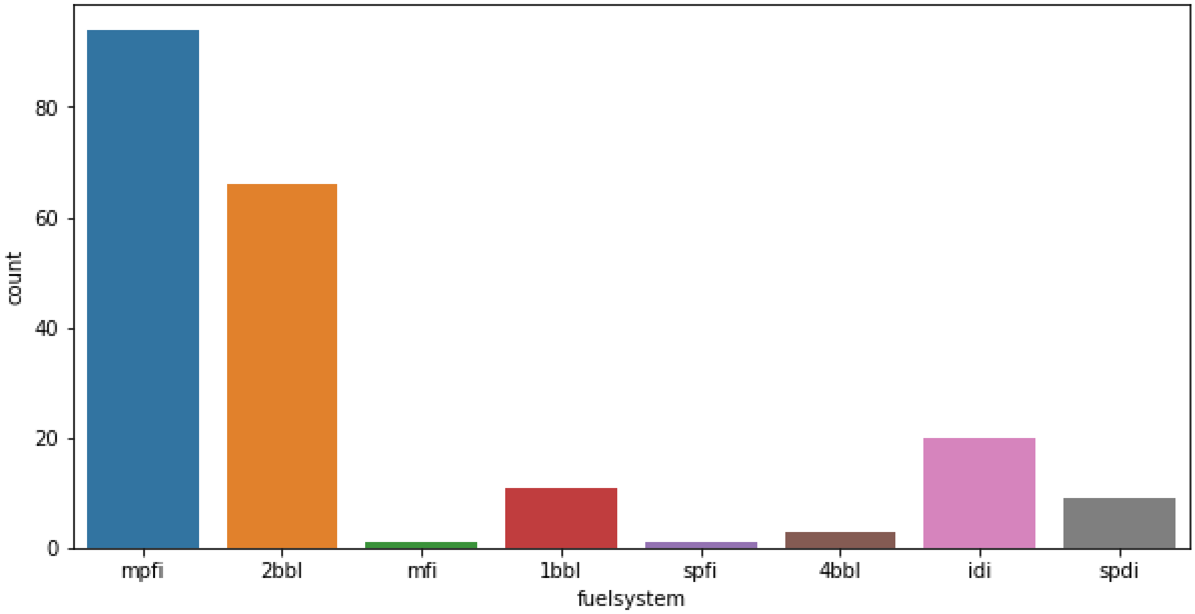
In [26]:

```
plt.figure(figsize = (10,5))
sns.violinplot(x="drivewheel",y="price",data=car_dataset)
plt.show()
```



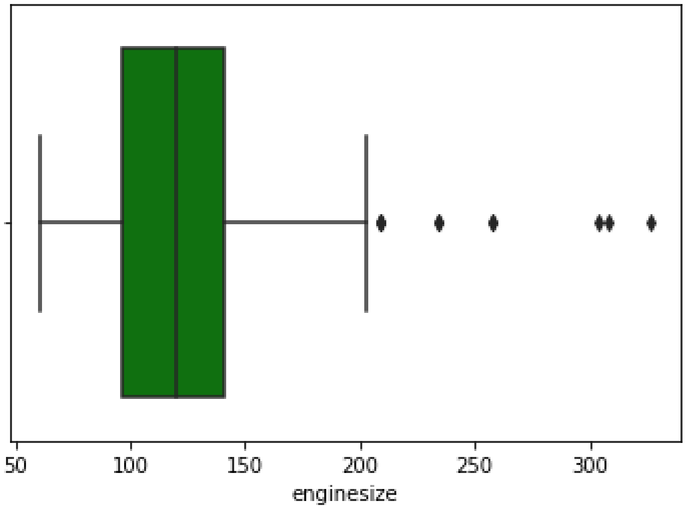
In [27]:

```
plt.figure(figsize = (10,5))
sns.countplot(x="fuelsystem",data=car_dataset)
plt.show()
```

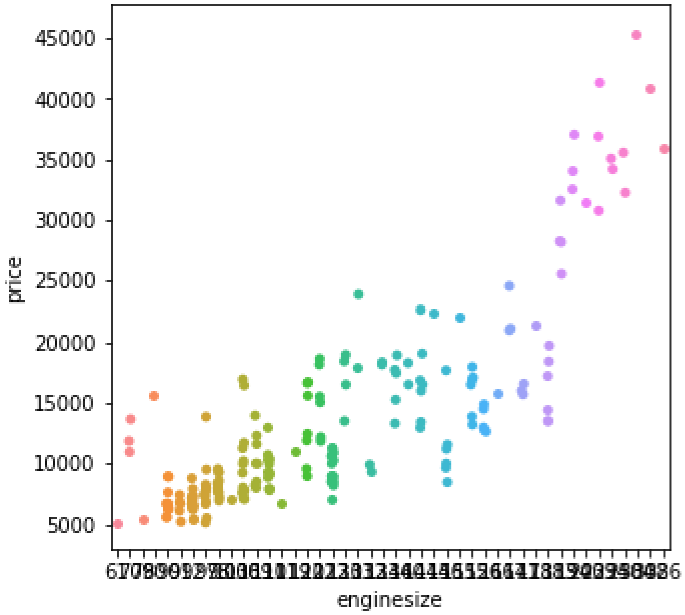


```
In [28]: sns.boxplot(x="enginesize",data=car_dataset,color="green")
```

```
Out[28]: <matplotlib.axes._subplots.AxesSubplot at 0x11b9fc27248>
```

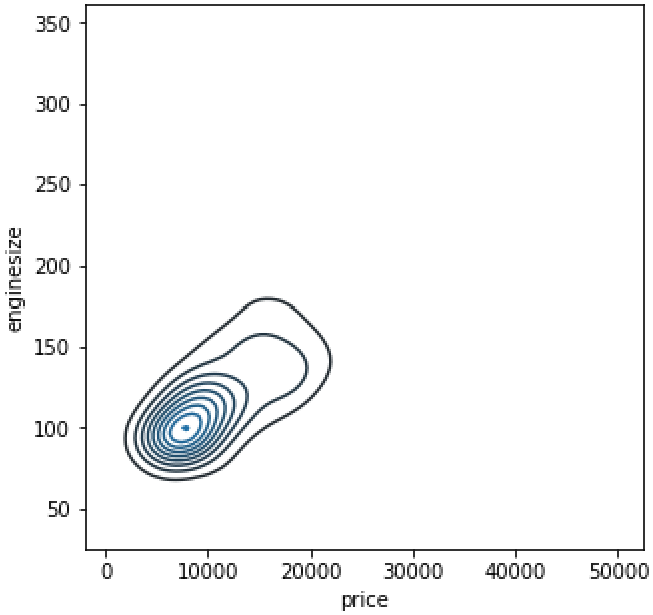


```
In [29]: plt.figure(figsize = (5,5))
         sns.stripplot(x="enginesize",y="price",data=car_dataset)
         plt.show()
```



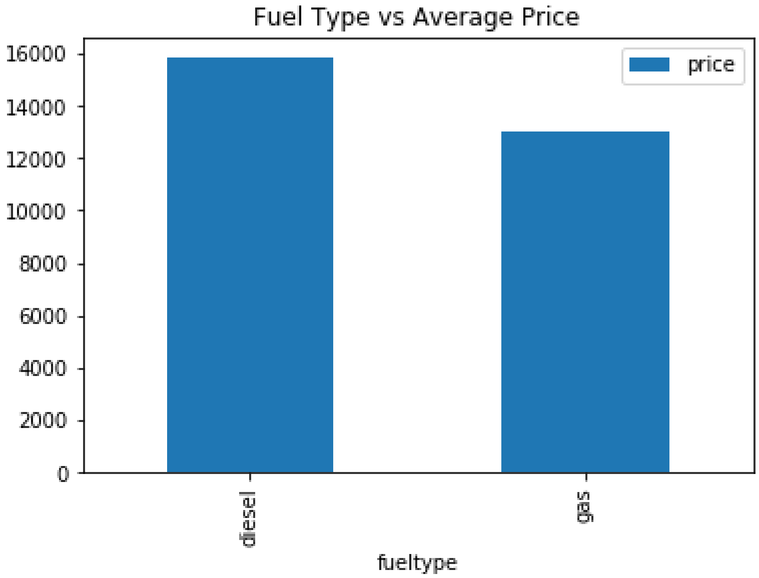
In [30]:

```
plt.figure(figsize = (5,5))
sns.kdeplot(car_dataset.price,car_dataset.enginesize)
plt.show()
```



In [31]:

```
df= pd.DataFrame(car_dataset.groupby(['fueltype'])['price'].mean().sort_values(
ascending = False))
df.plot.bar()
plt.title('Fuel Type vs Average Price')
plt.show()
```


Fuel Type vs Average Price
price
diesel
gas
fueltype