1 - Problem Statement

1 - Analysis of most economic company car for City and highway in the context of price and mileage.

2 - Car Body specification (length,width,height) , wheel-base and engine specification (engine-type ,num-of-cylinders,engine-size) affect the car price.

2 - About Meta-Data

Source:- https://archive.ics.uci.edu/ml/datasets/automobile (https://archive.ics.uci.edu/ml/datasets/automobile)

Data Volume:- 205 Records , 26 Columns

2.1- Column Information and Range:

In [ ]:
```
1- symboling            : mark of the automobile [ -3, -2, -1, 0, 1, 2, 3.]
2- normalized-losses    : relative average loss payment per insured vehicle year
3- make                 : manufacturer name
4- fuel-type            : Type of Fule used [ diesel, gas.]
5- aspiration           : Type of Induction ( Supercharging and Turbocharging)
6- num-of-doors         : Total Door in vehicle [ four, two ]
7- body-style           : Body style of car [hardtop, wagon, sedan, hatchback, c
8- drive-wheels         : 4wd, fwd, rwd
9- engine-location      : engine location fron or back
10- wheel-base          : wheelbase is the distance between the centers of the fr
11- length              : length of car [continuous from 141.1 to 208.1]
12- width               : width of car [ continuous from 60.3 to 72.3 ]
13- Height              : Height of the car [continuous from 47.8 to 59.8]
14- curb-weight         : total mass of a vehicle with standard equipment and al
15- engine-type         : engine type [dohc, dohcv, l, ohc, ohcf, ohcv, rotor.]
16- num-of-cylinders    : Total cyclinders in engine [eight, five, four, six, tl
17- engine-size         : engine Size [continuous from 61 to 326]
18- fuel-system         : fuel system is to store and supply fuel to the engine
19- bore                : inside diameter of engine's cylinder [continuous from
20- stroke              : continuous from 2.07 to 4.17
21- compression-ratio   : continuous from 7 to 23
22- horsepower          : continuous from 48 to 288
23- peak-rpm            : continuous from 4150 to 6600
24- city-mpg            : average of fule in city ( Miles Per Gallon) [continuou
25- highway-mpg         : average of fule in highway [continuous from 16 to 54]
26- price               : Price of car [continuous from 5118 to 45400]
```

3- Data Loading

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import pandas_profiling

df = pd.read_csv("C:\Automobile_data.csv")
print( " Successfully loaded")
```

```
 Successfully loaded
```

4- Data Profiling

Statistics Discription of Dataset

In [3]:
```
info = df.shape
print( " number of rows = ",info[0])
print( " Number of column = ",info[1])
df.isnull().sum()
```

```
 number of rows =  205
 Number of column =  26
```

Out[3]:
```
symboling            0
normalized-losses    0
make                 0
fuel-type            0
aspiration           0
num-of-doors         0
body-style           0
drive-wheels         0
engine-location      0
wheel-base           0
length               0
width                0
height               0
curb-weight          0
engine-type          0
num-of-cylinders     0
engine-size          0
fuel-system          0
bore                 0
stroke               0
compression-ratio    0
horsepower           0
peak-rpm             0
city-mpg             0
highway-mpg          0
price                0
dtype: int64
```

In [24]:
```
df.head()
```

Out[24]:

	symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	.
0	3	?	alfa-romero	gas	std	two	convertible	rwd	front	88.6	.
1	3	?	alfa-romero	gas	std	two	convertible	rwd	front	88.6	.
2	1	?	alfa-romero	gas	std	two	hatchback	rwd	front	94.5	.
3	2	164	audi	gas	std	four	sedan	fwd	front	99.8	.
4	2	164	audi	gas	std	four	sedan	4wd	front	99.4	.

5 rows × 26 columns

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

Out[25]:

	symboling	wheel-base	length	width	height	curb-weight	engine-size	comp
count	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	20
mean	0.834146	98.756585	174.049268	65.907805	53.724878	2555.565854	126.907317	1
std	1.245307	6.021776	12.337289	2.145204	2.443522	520.680204	41.642693	
min	-2.000000	86.600000	141.100000	60.300000	47.800000	1488.000000	61.000000	
25%	0.000000	94.500000	166.300000	64.100000	52.000000	2145.000000	97.000000	
50%	1.000000	97.000000	173.200000	65.500000	54.100000	2414.000000	120.000000	
75%	2.000000	102.400000	183.100000	66.900000	55.500000	2935.000000	141.000000	
max	3.000000	120.900000	208.100000	72.300000	59.800000	4066.000000	326.000000	2

5- Pre Profiling

In [ ]:
```
profile = pandas_profiling.ProfileReport(df)
profile.to_file(output_file= "pre_automobile.html",silent=False)
```

6- Data Clean-Up (Pre Processing)

some of columns having "?" values ,It consider as missing values and it is replaced by appropriate value

6.1- "normalized-losses"

"?" is consider as missing vlaues. it can be replace ? with the mean values of all "normalized-losses"

In [2]:
```
print( "Number of record in column = normalized-losses having ?--> {0}".format(d
#convert string to float
df["normalized-losses"]  = pd.to_numeric(arg=df["normalized-losses"] ,errors = "
df["normalized-losses"].fillna(value= df["normalized-losses"].mean(),inplace =Tru
print( "Number of record in column = normalized-losses having ?--> {0}".format(d
```

```
Number of record in column = normalized-losses having ?--> 41
Number of record in column = normalized-losses having ?--> 0

C:\Users\Lenovo\Anaconda3\lib\site-packages\pandas\core\ops.py:1649: FutureWarn
ing: elementwise comparison failed; returning scalar instead, but in the future
will perform elementwise comparison
  result = method(y)
```

6.2- "num-of-doors"

Column "num-of-doors" is having categorical data. There are only 2 records having "?". Replace the ? to the mode of "num-of-doors" on the basis of column "make" and "body-style".

In [3]:
```
print(" Total number of ? in columns = num-of-doors ---> ",df[ df["num-of-doors"
df["num-of-doors" ].replace(to_replace="?",value= df[ (df["body-style"] == "sedan
df["num-of-doors" ].replace(to_replace="?",value= df[ (df["body-style"] == "sedan
print(" Total number of ? in columns = num-of-doors ---> ",df[ df["num-of-doors"
```

```
 Total number of ? in columns = num-of-doors --->  2
 Total number of ? in columns = num-of-doors --->  0
```

6.3- "bore"

Column "bore" is having numerical data,data "?" is replaced by NAN

In [4]:
```
print(" Total number of ? in column = bore ---> ",df[ df["bore" ] == "?" ]["bore"
df["bore"]  = pd.to_numeric(arg=df["bore"] ,errors = "coerce" )
print(" Total number of ? in column = bore ---> ",df[ df["bore" ] == "?" ]["bore"
```

```
 Total number of ? in column = bore --->  4
 Total number of ? in column = bore --->  0
```

6.4- "stroke"

Column "stroke" is having numerical data,data "?" is replaced by NAN

In [5]:
```
print(" Total number of ? in column = stroke ---> ",df[ df["stroke" ] == "?" ]["
df["stroke"]=pd.to_numeric(arg=df["stroke"] ,errors = "coerce")
print(" Total number of ? in column = stroke ---> ",df[ df["stroke" ] == "?" ]["
```

```
 Total number of ? in column = stroke --->  4
 Total number of ? in column = stroke --->  0
```

6.5- "peak-rpm"

Column "peak-rpm" is having numerical data,data "?" is replaced by NAN

In [6]:
```
#clean the column "peak-rpm" and set NAN
print(" Total number of ? in column = peak-rpm ---> ",df[ df["peak-rpm" ] == "?"
df["peak-rpm"]=pd.to_numeric(arg=df["peak-rpm"] ,errors = "coerce")
print(" Total number of ? in column = peak-rpm ---> ",df[ df["peak-rpm" ] == "?"
```

```
 Total number of ? in column = peak-rpm --->  2
 Total number of ? in column = peak-rpm --->  0
```

6.6- "price"

"?" in "price" column is replaced by mean value

In [7]:
```
# get the numeric and non numeric values
df["price"].str.isnumeric().value_counts()
#get the mean of numeric values
nPrice = df["price"].loc[df["price"].str.isnumeric()== True]
mPrice = nPrice.astype(int).mean()
#replce by mean
df['price'] = df['price'].replace('?',mPrice).astype(int)
```

6.7- "horsepower"

"?" in "horsepower" column is replaced by mean value

In [8]:
```
# get the numeric and non numeric values
df['horsepower'].str.isnumeric().value_counts()
nhorsepower = df['horsepower'].loc[ df['horsepower'].str.isnumeric() == True]
mhorsepower = nhorsepower.astype(int).mean()
mhorsepower
#replce by mean
df['horsepower'] = df['horsepower'].replace('?',mhorsepower).astype(int)
```

In [11]:
```
#save data set in csv file
df.to_csv("C:\Automobile_cleaned_data.csv")
```

7- Post Pandas Profiling

In [9]:
```
profile = pandas_profiling.ProfileReport(df)
profile.to_file(output_file= "post_automobile.html",silent=False)
```

In [14]:
```
df.head(df.shape[0]) # display complete data set
```

Out[14]:

	symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location	w
0	3	122.0	alfa-romero	gas	std	two	convertible	rwd	front	
1	3	122.0	alfa-romero	gas	std	two	convertible	rwd	front	
2	1	122.0	alfa-romero	gas	std	two	hatchback	rwd	front	
3	2	164.0	audi	gas	std	four	sedan	fwd	front	
4	2	164.0	audi	gas	std	four	sedan	4wd	front	
5	2	122.0	audi	gas	std	two	sedan	fwd	front	
6	1	158.0	audi	gas	std	four	sedan	fwd	front	
7	1	122.0	audi	gas	std	four	wagon	fwd	front	
8	1	158.0	audi	gas	turbo	four	sedan	fwd	front	
9	0	122.0	audi	gas	turbo	two	hatchback	4wd	front	
10	2	192.0	bmw	gas	std	two	sedan	rwd	front	
11	0	192.0	bmw	gas	std	four	sedan	rwd	front	
12	0	188.0	bmw	gas	std	two	sedan	rwd	front	
13	0	188.0	bmw	gas	std	four	sedan	rwd	front	
14	1	122.0	bmw	gas	std	four	sedan	rwd	front	
15	0	122.0	bmw	gas	std	four	sedan	rwd	front	
16	0	122.0	bmw	gas	std	two	sedan	rwd	front	
17	0	122.0	bmw	gas	std	four	sedan	rwd	front	
18	2	121.0	chevrolet	gas	std	two	hatchback	fwd	front	
19	1	98.0	chevrolet	gas	std	two	hatchback	fwd	front	
20	0	81.0	chevrolet	gas	std	four	sedan	fwd	front	
21	1	118.0	dodge	gas	std	two	hatchback	fwd	front	
22	1	118.0	dodge	gas	std	two	hatchback	fwd	front	
23	1	118.0	dodge	gas	turbo	two	hatchback	fwd	front	
24	1	148.0	dodge	gas	std	four	hatchback	fwd	front	
25	1	148.0	dodge	gas	std	four	sedan	fwd	front	
26	1	148.0	dodge	gas	std	four	sedan	fwd	front	
27	1	148.0	dodge	gas	turbo	four	sedan	fwd	front	
28	-1	110.0	dodge	gas	std	four	wagon	fwd	front	
29	3	145.0	dodge	gas	turbo	two	hatchback	fwd	front	
...	...	...	...	...	...	...	...	...	...	
175	-1	65.0	toyota	gas	std	four	hatchback	fwd	front	
176	-1	65.0	toyota	gas	std	four	sedan	fwd	front	

	symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location	w
177	-1	65.0	toyota	gas	std	four	hatchback	fwd	front	
178	3	197.0	toyota	gas	std	two	hatchback	rwd	front	
179	3	197.0	toyota	gas	std	two	hatchback	rwd	front	
180	-1	90.0	toyota	gas	std	four	sedan	rwd	front	
181	-1	122.0	toyota	gas	std	four	wagon	rwd	front	
182	2	122.0	volkswagen	diesel	std	two	sedan	fwd	front	
183	2	122.0	volkswagen	gas	std	two	sedan	fwd	front	
184	2	94.0	volkswagen	diesel	std	four	sedan	fwd	front	
185	2	94.0	volkswagen	gas	std	four	sedan	fwd	front	
186	2	94.0	volkswagen	gas	std	four	sedan	fwd	front	
187	2	94.0	volkswagen	diesel	turbo	four	sedan	fwd	front	
188	2	94.0	volkswagen	gas	std	four	sedan	fwd	front	
189	3	122.0	volkswagen	gas	std	two	convertible	fwd	front	
190	3	256.0	volkswagen	gas	std	two	hatchback	fwd	front	
191	0	122.0	volkswagen	gas	std	four	sedan	fwd	front	
192	0	122.0	volkswagen	diesel	turbo	four	sedan	fwd	front	
193	0	122.0	volkswagen	gas	std	four	wagon	fwd	front	
194	-2	103.0	volvo	gas	std	four	sedan	rwd	front	
195	-1	74.0	volvo	gas	std	four	wagon	rwd	front	
196	-2	103.0	volvo	gas	std	four	sedan	rwd	front	
197	-1	74.0	volvo	gas	std	four	wagon	rwd	front	
198	-2	103.0	volvo	gas	turbo	four	sedan	rwd	front	
199	-1	74.0	volvo	gas	turbo	four	wagon	rwd	front	
200	-1	95.0	volvo	gas	std	four	sedan	rwd	front	
201	-1	95.0	volvo	gas	turbo	four	sedan	rwd	front	
202	-1	95.0	volvo	gas	std	four	sedan	rwd	front	
203	-1	95.0	volvo	diesel	turbo	four	sedan	rwd	front	
204	-1	95.0	volvo	gas	turbo	four	sedan	rwd	front	

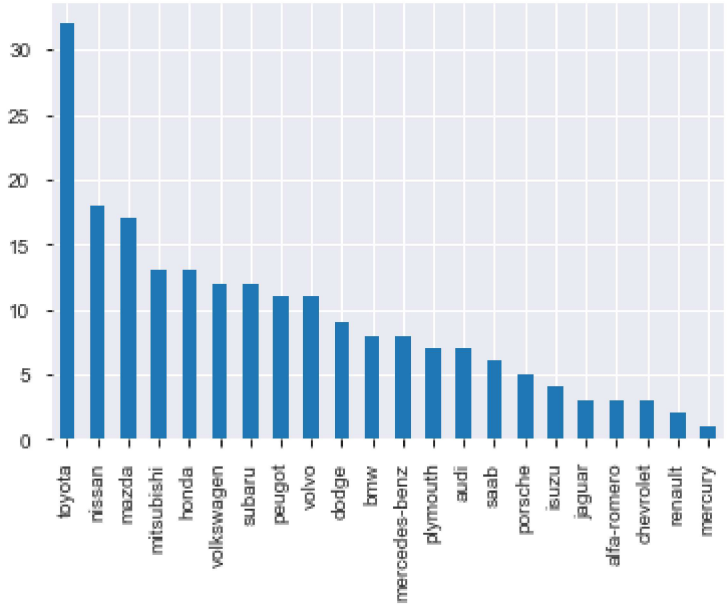
205 rows × 26 columns

8- Problem Statement 1- Most economic company car for City and highway

8.1- Frequency diagram for make

In [35]:
```
df.make.value_counts().plot.bar()
```

Out[35]: <matplotlib.axes._subplots.AxesSubplot at 0x29b813795c0>



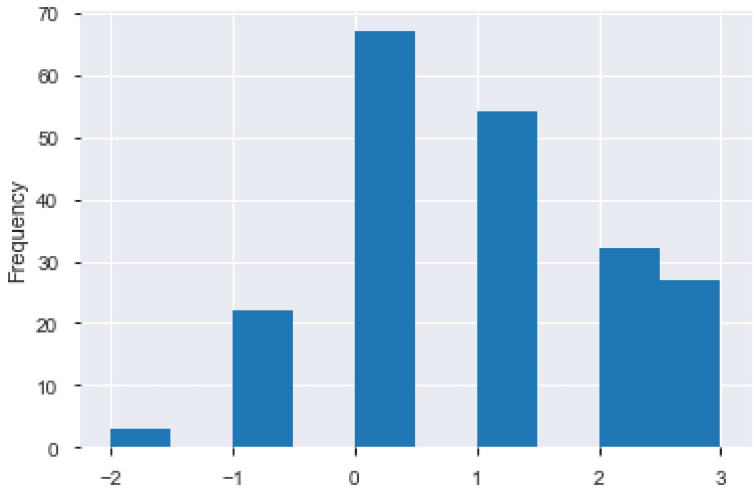
In [16]:
```
print(  "Bigest manufacturer of car is Toyota : %f" %((df[ df["make"] == "toyota
print(  "Least manufacturer of car is Mercury : %f" %((df[ df["make"] == "mercur
```

```
Bigest manufacturer of car is Toyota : 15.609756 %
Least manufacturer of car is Mercury : 0.487805 %
```

8.2- Risk rating (symboling) Histogram

In [39]:
```
df.symboling.plot(kind="hist")
# risk rating is from -3 to 3 , but it start from -2. most of car having risk ra
```

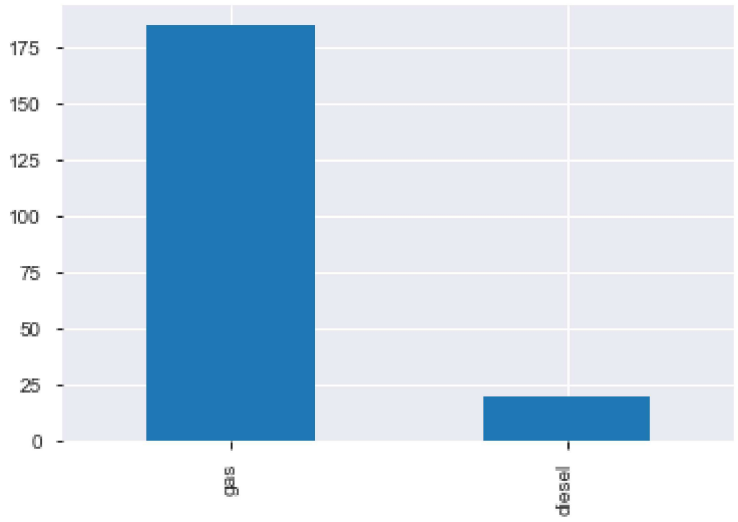
Out[39]: <matplotlib.axes._subplots.AxesSubplot at 0x29b81926e10>



8.3- Frequency diagram for fule type

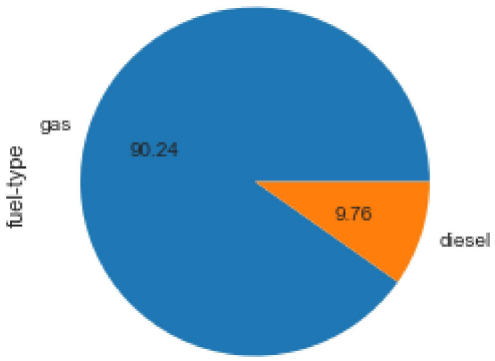
In [91]:
```
df["fuel-type"].value_counts().plot(kind="bar")
```

Out[91]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1b9849e0390>
```



In [36]:
```
df["fuel-type"].value_counts().plot(kind="pie",autopct='%.2f')
```

Out[36]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x29b817f42b0>
```



Answer:- percentage manufactured car according fuel Type

gas = 90.24%

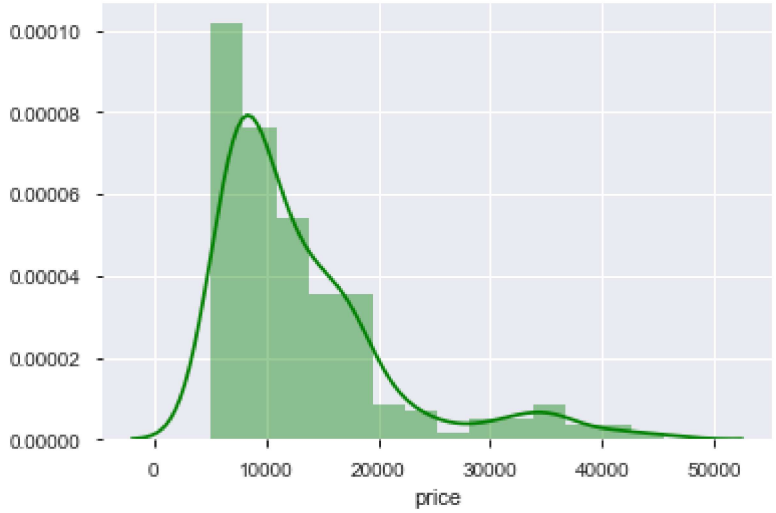
desel = 9.76%

gas fuel car are in most demand

8.4- Distribution of sales price

In [93]:
```
sns.distplot(df['price'],color ='g')

print("Skewness: %f" % df['price'].skew())
print("Kurtosis: %f" % df['price'].kurt())

#plot is skewed towards the left it means very less car having high price
```

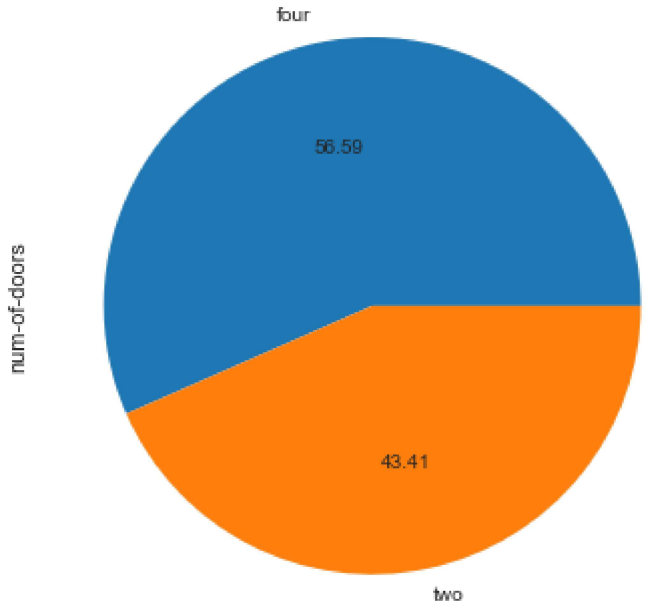
```
Skewness: 1.827325
Kurtosis: 3.354218
```



8.5- Number of doors - pie chart

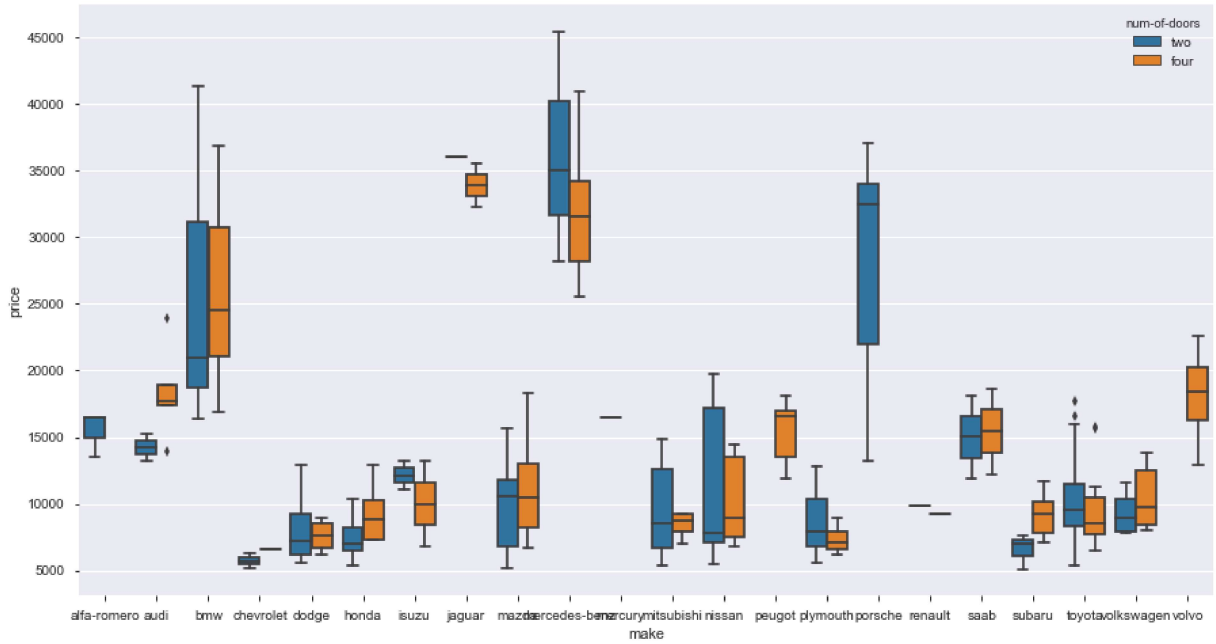
In [94]:
```
df["num-of-doors"].value_counts().plot(kind="pie",figsize = (6,9),autopct='%.2f'

# 56.59 % cars are four door
# 43.41 % cars are two  door
```

Out[94]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1b984af6b00>
```



8.6- Box Plot between make and Price

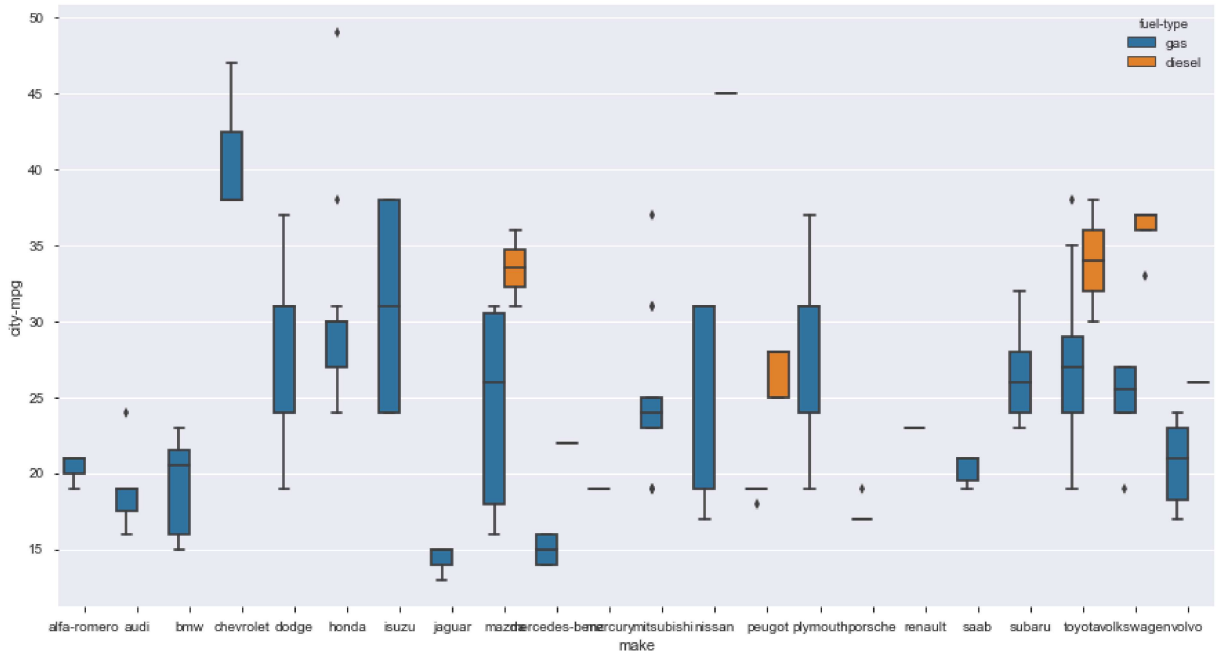
In [95]:
```
from matplotlib import pyplot as plt
plt.figure(figsize=(15,8))
ax = sns.boxplot( x="make",y="price",hue= "num-of-doors",data = df)

# 1 - Most expensive car is mercedes two door and least expensive is Chevrolet tw
# 2- Premium car above 20000 are bmw,Jaquar, Mercedes benz and Porsche.
# 3- car less then 10000 are Chevrolet, Dodge, Honda, Mitsubishi, Plymoth and Sul
# 4- Rest car are in middle range.
```



8.7- Box Plot between city-mpg , make and hue is Fule type

In [96]:
```
plt.figure(figsize=(15,8))
sns.boxplot(x="make",y="city-mpg",hue="fuel-type", data = df)
# least expensive car "Chevrolet" fule type = "gas" is very good in city mileage
```

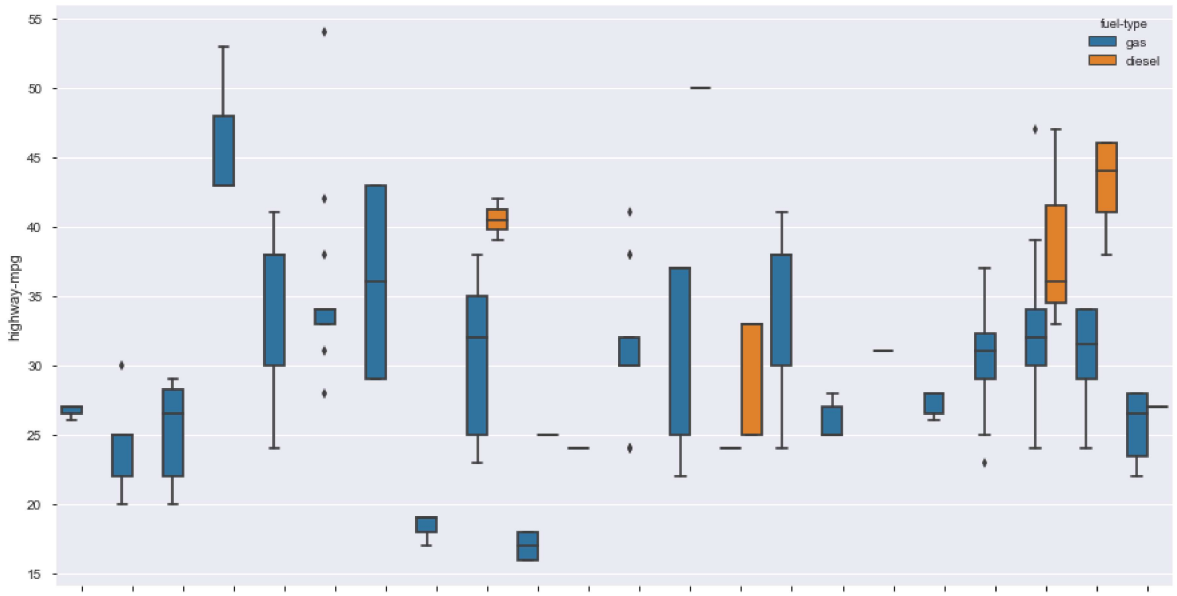
Out[96]: <matplotlib.axes._subplots.AxesSubplot at 0x1b984b30438>



8.8- Box Plot between highway-mpg , make and hue is Fule type

In [97]:
```
plt.figure(figsize=(15,8))
sns.boxplot(x="make",y="highway-mpg",hue="fuel-type", data = df)
# least expensive car "Chevrolet" fule type = "gas" is very good in High mileage
```

Out[97]: <matplotlib.axes._subplots.AxesSubplot at 0x1b9856f0e48>



8.9- Summary

1- "Toyota" manufactored most number of vehicles which is 15.60% than the 2nd "Nissan" that is 8.78 % and least is "Mercury".

2- Most manufactored fuel type car is "gass" which is 90.24%

3- Risk rating is from -3 to 3, in given data set it it starts from -2 and most of car having risk rating between 0 to 1.

4- Most expensive car is "mercedes" and least expensive is "Chevrolet". Both are two door car.

5- Premium car above 20000 are bmw,Jaquar, Mercedes benz and Porsche.

6- car less then 10000 are Chevrolet, Dodge, Honda, Mitsubishi, Plymoth and Subaru and rest are in middle range.

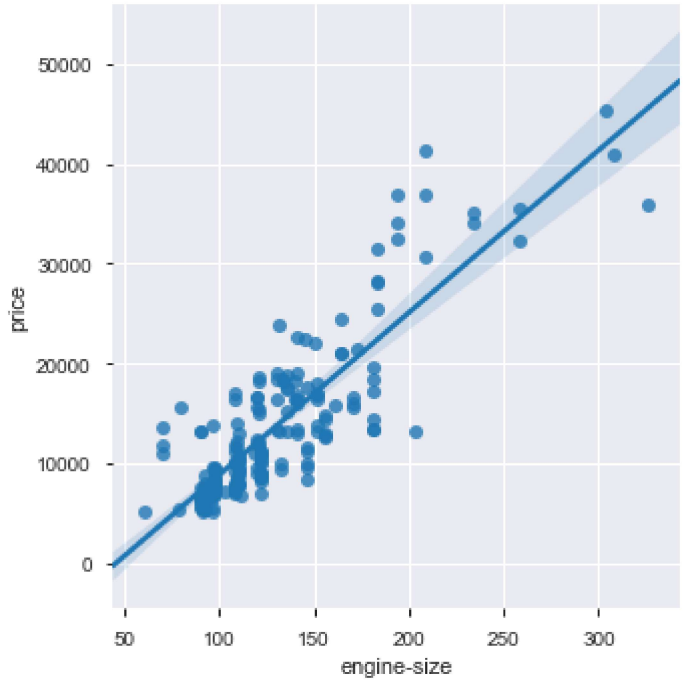
7- Least expensive car "Chevrolet" fule type = "gas" is very good in City/High mileage

9- Problem Statement 2:- Body specification , wheels base and engine specification affect the car price.

9.1- Scatter plot between engine size and price

In [72]:
```
sns.lmplot(x="engine-size",y="price",data=df)
# there is a linear releation between price  and engine size.
```

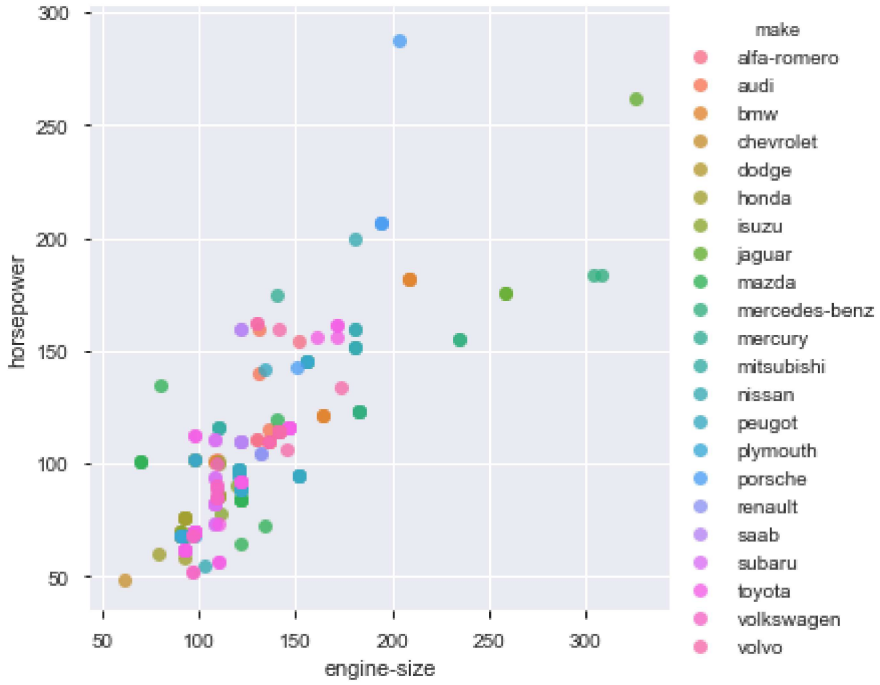
Out[72]: <seaborn.axisgrid.FacetGrid at 0x1b9e849ec50>



9.2- Scatter plot between engine size and horsepower

In [13]:
```
sns.lmplot(y="horsepower",x="engine-size", data=df, hue="make",fit_reg=False)
# as engine size increase power of engine increase
```

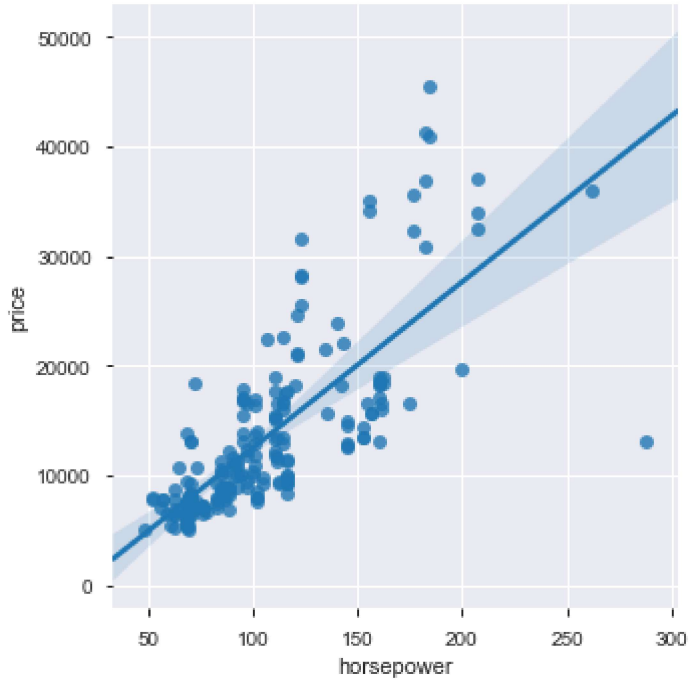
Out[13]: <seaborn.axisgrid.FacetGrid at 0x16a7d4e8438>



9.3- Scatter plot between horsepower and price

In [16]:
```
sns.lmplot(x="horsepower",y="price",data=df)
# there is a linear releation between price  and horsepower.
```

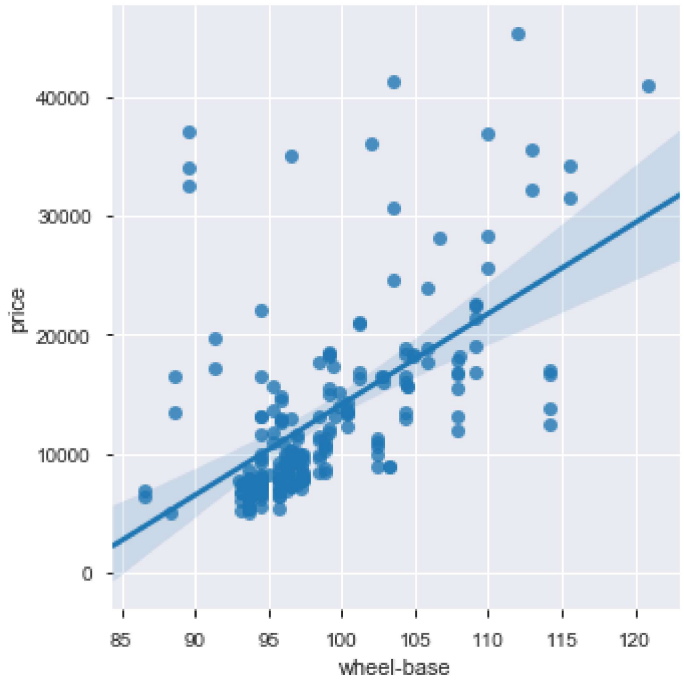
Out[16]: <seaborn.axisgrid.FacetGrid at 0x16a7d6a0b70>



9.4- Scatter plot between wheel-base and price

In [14]:
```
sns.lmplot(x="wheel-base",y="price",data=df)
# there is a linear releation between price  and wheel-base.
```

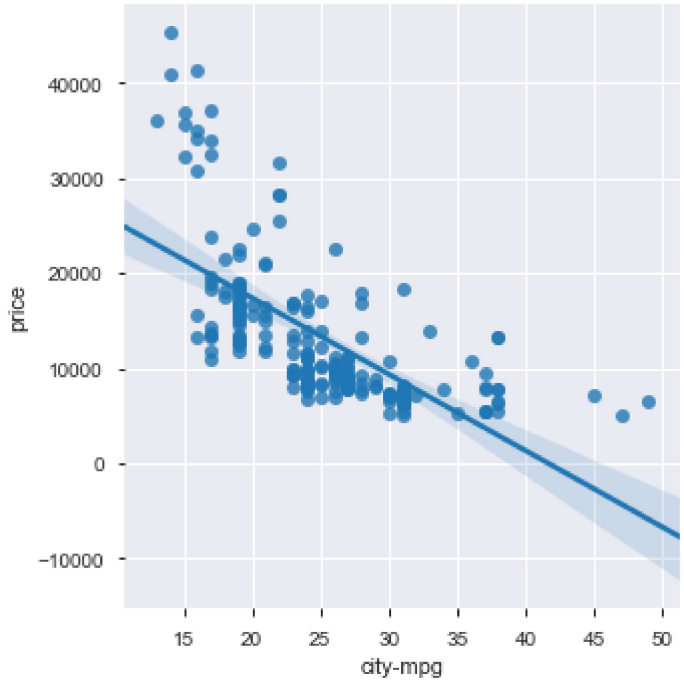
Out[14]: <seaborn.axisgrid.FacetGrid at 0x16a7d5cedd8>



9.5- Scatter plot between City mileage and price

In [71]:
```
sns.lmplot(x="city-mpg",y="price",data=df)
# there is a negetive correlation bewteen city mileage and price. if any car is h
```

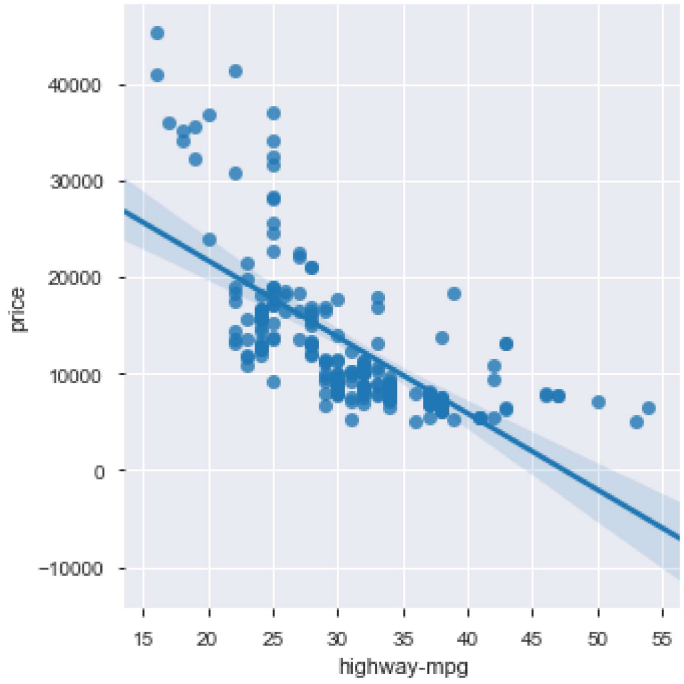
Out[71]:
```
<seaborn.axisgrid.FacetGrid at 0x1b9e8464a58>
```



9.6- Scatter plot between High mileage and price

In [70]:
```
sns.lmplot(x="highway-mpg",y="price",data=df)
# there is a negetive correlation bewteen highway mileage and price.
```

Out[70]: <seaborn.axisgrid.FacetGrid at 0x1b9e7dc8c50>



9.7- Summary

1- There is a positive linear releation between engine size and price, engine size and horsepower, horsepower and price and wheel-base and price.

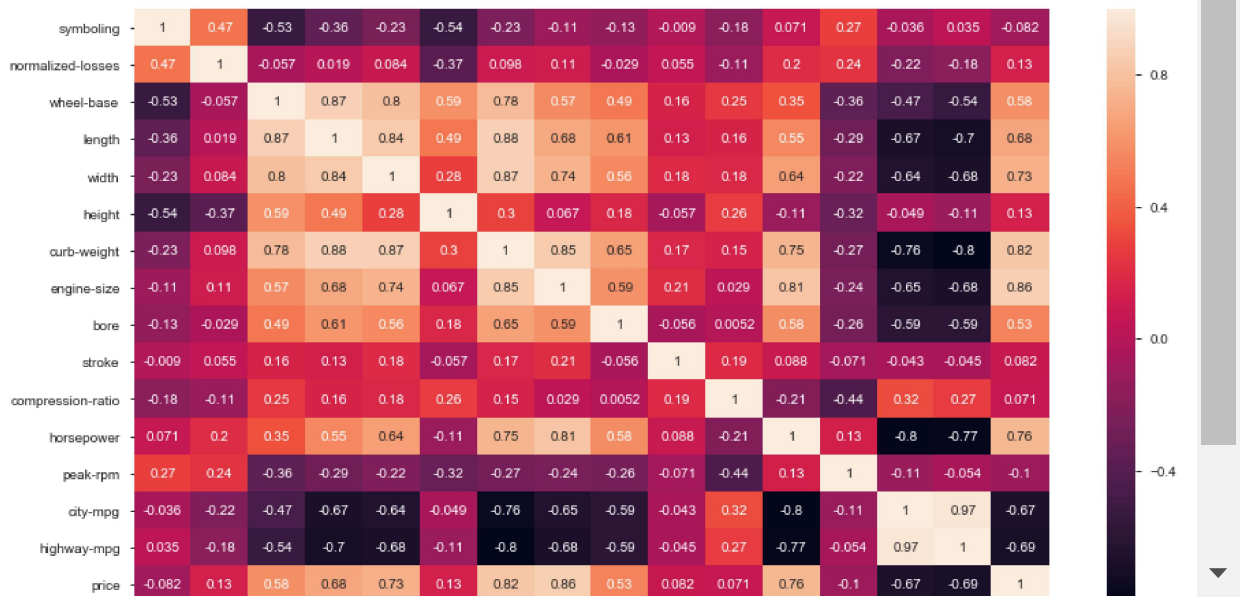
2- As engine and physical specificatin increases,price increases.

3- City and High mileage are having negative linear relation with price.

10- Correlation Analysis

In [100]:
```
plt.figure(figsize=(15,8))
sns.heatmap(  df.loc[:,  ["symboling","normalized-losses","wheel-base","length",
```

Out[100]: <matplotlib.axes._subplots.AxesSubplot at 0x1b985c5a5f8>



10.1- Summary

1- Engine size and crub weigth of car are highly correlated with price

2- Wheel base is highly correlated with length and width

3- Engine size is higly correlated with hoursepower

Conclusion

1- Engine specifications and body characters of car affect price. As specification and characters increase , prices increases.

2- Least expensive car manufacture is "Chevrolet" fule type = "gas" for city/Highway but Toyota 's car sale is high.