Cars Prices Prediction

Linear Regression

A Chinese automobile company Geely Auto aspires to enter the US market by setting up their manufacturing unit there and producing cars locally to give competition to their US and European counterparts.

Business Goal

We are required to model the price of cars with the available independent variables. It will be used by the management to understand how exactly the prices vary with the independent variables. They can accordingly manipulate the design of the cars, the business strategy etc. to meet certain price levels. Further, the model will be a good way for management to understand the pricing dynamics of a new market.

In [1]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```python
data_df=pd.read_csv('CarPrice_Assignment.csv')
data_df.shape
```

Out[2]: (205, 26)

understanding the data

In [3]:
```python
data_df.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 [4]:
```python
data_df.describe()
```

Out[4]:

	car_ID	symboling	wheelbase	carlength	carwidth	carheight	curbweight	enginesize	boreratio	stroke	compressionratio	horsepowe
count	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.00000
mean	103.000000	0.834146	98.756585	174.049268	65.907805	53.724878	2555.565854	126.907317	3.329756	3.255415	10.142537	104.11707
std	59.322565	1.245307	6.021776	12.337289	2.145204	2.443522	520.680204	41.642693	0.270844	0.313597	3.972040	39.54416
min	1.000000	-2.000000	86.600000	141.100000	60.300000	47.800000	1488.000000	61.000000	2.540000	2.070000	7.000000	48.00000
25%	52.000000	0.000000	94.500000	166.300000	64.100000	52.000000	2145.000000	97.000000	3.150000	3.110000	8.600000	70.00000
50%	103.000000	1.000000	97.000000	173.200000	65.500000	54.100000	2414.000000	120.000000	3.310000	3.290000	9.000000	95.00000
75%	154.000000	2.000000	102.400000	183.100000	66.900000	55.500000	2935.000000	141.000000	3.580000	3.410000	9.400000	116.00000
max	205.000000	3.000000	120.900000	208.100000	72.300000	59.800000	4066.000000	326.000000	3.940000	4.170000	23.000000	288.00000

cleaning data

detecting nulls

In [5]:
```
df_null=data_df.isna().sum()
df_null
```

Out[5]:
```
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
```

check duplicates

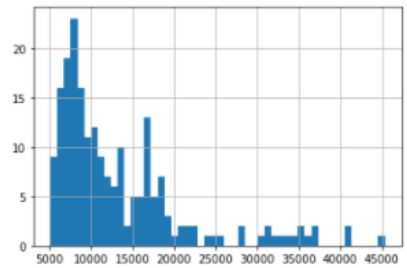
In [6]:
```
data_df.duplicated().value_counts()
```

Out[6]:
```
False    205
dtype: int64
```

detecting outliers

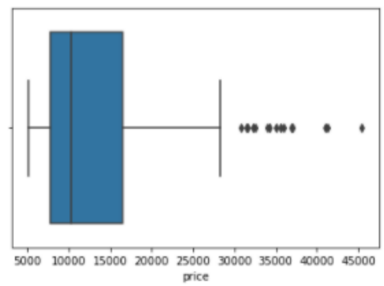
In [7]:
```
data_df['price'].hist(bins=50)
```

Out[7]: <AxesSubplot:>



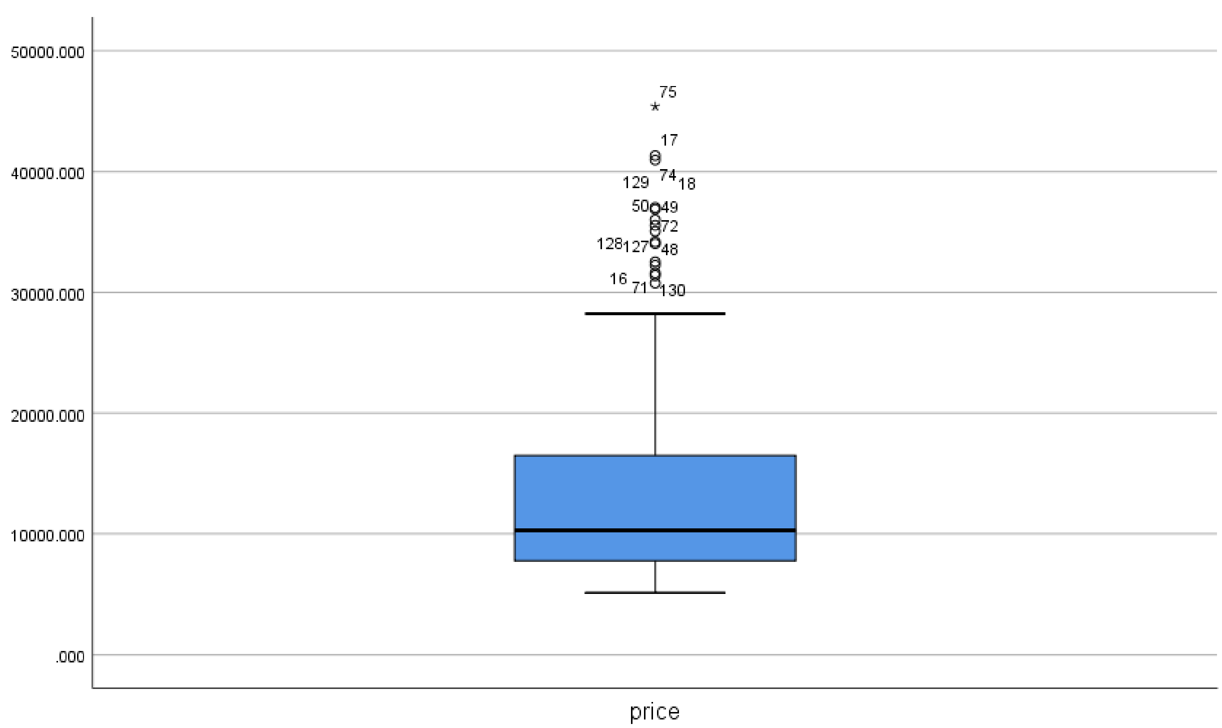
In [8]:
```
import seaborn as sns
sns.boxplot(x=data_df['price'])
```

Out[8]: <AxesSubplot:xlabel='price'>



In [9]:
```
data_df.shape
```

Out[9]: (205, 26)



removing outliers using SPSS

importing the new CSV file

In [10]:
```
data_cleaned=pd.read_csv('carprice.csv')
data_cleaned
```

Out[10]:

	car_ID	symboling	CarName	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	...	enginesize	fuelsystem	boreratio
0	1	3	alfa-romero giulia	gas	std	two	convertible	rwd	front	88.6	...	130	mpfi	3.47
1	2	3	alfa-romero stelvio	gas	std	two	convertible	rwd	front	88.6	...	130	mpfi	3.47
2	3	1	alfa-romero Quadrifoglio	gas	std	two	hatchback	rwd	front	94.5	...	152	mpfi	2.68
3	4	2	audi 100 ls	gas	std	four	sedan	fwd	front	99.8	...	109	mpfi	3.19
4	5	2	audi 100ls	gas	std	four	sedan	4wd	front	99.4	...	136	mpfi	3.19
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
183	201	-1	volvo 145e (sw)	gas	std	four	sedan	rwd	front	109.1	...	141	mpfi	3.78
184	202	-1	volvo 144ea	gas	turbo	four	sedan	rwd	front	109.1	...	141	mpfi	3.78
185	203	-1	volvo 244dl	gas	std	four	sedan	rwd	front	109.1	...	173	mpfi	3.58
186	204	-1	volvo 246	diesel	turbo	four	sedan	rwd	front	109.1	...	145	idi	3.01
187	205	-1	volvo 264gl	gas	turbo	four	sedan	rwd	front	109.1	...	141	mpfi	3.78

188 rows × 26 columns

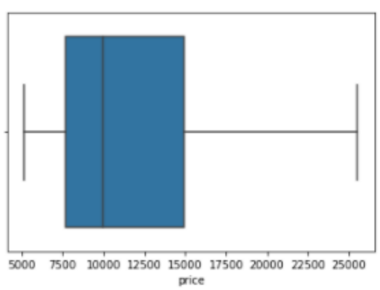
In [11]:
```
data_cleaned.shape
```

Out[11]: (188, 26)

box plot after removing outliers

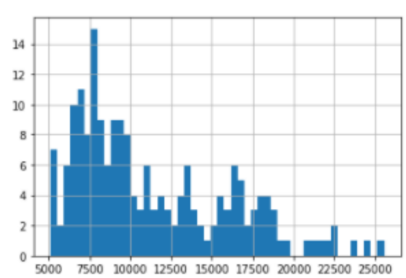
In [12]:
```
sns.boxplot(x=data_cleaned['price'])
```

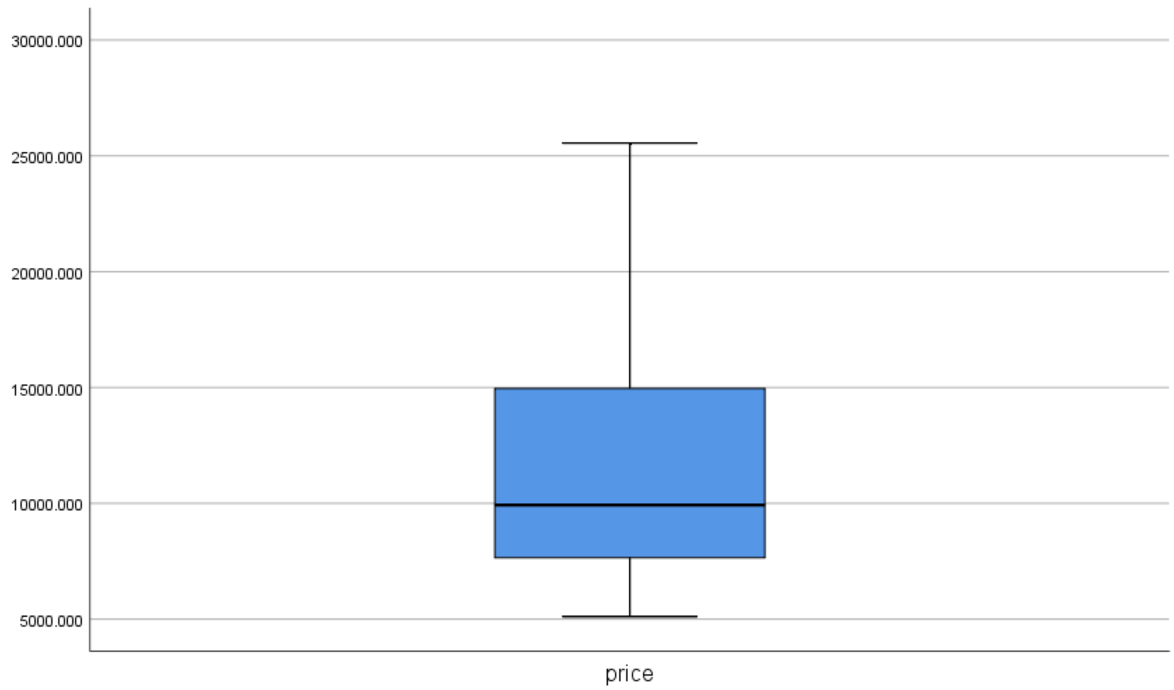
Out[12]: <AxesSubplot:xlabel='price'>



In [13]:
```
data_cleaned['price'].hist(bins=50)
```

Out[13]: <AxesSubplot:>




30000.000
25000.000
20000.000
15000.000
10000.000
5000.000
price

handling car name to be standard

In [29]:
```
data_cleaned['CarName'] = data_cleaned['CarName'].str.split(' ',expand=True)
data_cleaned['CarName'].unique()
```

Out[29]:
```
array(['alfa-romero', 'audi', 'bmw', 'chevrolet', 'dodge', 'honda',
       'isuzu', 'mazda', 'buick', 'mercury', 'mitsubishi', 'nissan',
       'peugeot', 'plymouth', 'porsche', 'renault', 'saab', 'subaru',
       'toyota', 'volkswagen', 'volvo'], dtype=object)
```

maxda:mazda , Nissan:nissan , toyouta:toyota , vokswagen:volkswagen , vw:volkswagen

In [16]:
```
data_cleaned["CarName"]=data_cleaned['CarName'].replace(to_replace =["maxda", "Nissan","toyouta","vokswagen","vw"],
                             value =["mazda","nissan","toyota","volkswagen","volkswagen"])
```

In [17]:
```
data_cleaned['CarName'].unique()
```

Out[17]:
```
array(['alfa-romero', 'audi', 'bmw', 'chevrolet', 'dodge', 'honda',
       'isuzu', 'mazda', 'buick', 'mercury', 'mitsubishi', 'nissan',
       'peugeot', 'plymouth', 'porsche', 'renault', 'saab', 'subaru',
       'toyota', 'volkswagen', 'volvo'], dtype=object)
```

dropping Car ID column

In [15]:
```
data_cleaned=data_cleaned.drop(['car_ID'],axis=1)
data_cleaned.info()
```

```
 7   enginelocation    188 non-null    object
 8   wheelbase         188 non-null    float64
 9   carlength         188 non-null    float64
 10  carwidth          188 non-null    float64
 11  carheight         188 non-null    float64
 12  curbweight        188 non-null    int64
 13  enginetype        188 non-null    object
 14  cylindernumber    188 non-null    object
 15  enginesize        188 non-null    int64
 16  fuelsystem        188 non-null    object
 17  boreratio         188 non-null    float64
 18  stroke            188 non-null    float64
 19  compressionratio  188 non-null    float64
 20  horsepower        188 non-null    int64
 21  peakrpm           188 non-null    int64
 22  citympg           188 non-null    int64
 23  highwaympg        188 non-null    int64
 24  price             188 non-null    float64
dtypes: float64(8), int64(7), object(10)
memory usage: 36.8+ KB
```

preparing data

setting symboling to be from 1 to 7 to be analayzed using SPSS

In [18]:
```
data_cleaned["symboling"]=data_cleaned['symboling'].replace(to_replace =[-3, -2,-1,0,1,2,3],
                             value =[1,2,3,4,5,6,7])
data_cleaned["symboling"].head(10)
```

Out[18]:
```
0    7
1    7
2    5
3    6
4    6
5    6
6    5
7    5
8    5
9    4
Name: symboling, dtype: int64
```

separating the data into 2 data frames , one for categorical and the othe for numerical

In [19]:
```
cat_col = data_cleaned.select_dtypes(include=['object']).columns
num_col = data_cleaned.select_dtypes(exclude=['object']).columns
df_cat = data_cleaned[cat_col]
df_num = data_cleaned[num_col]
```

exporting 3 csv files to show relationships between variables using SPSS (categorical data file ,numerical data file, cleaned data file)

In [20]:
```
df_cat.to_csv('df_cat.csv')
df_num.to_csv('df_num.csv')
```

In [21]:
```
data_cleaned.to_csv('data_cleaned.csv')
```

we found that the variables (Wheel base,Horse power,Car length,Car width and Engine size) have relashionship with Price

Statistics

		symboling	price
N	Valid	188	188
	Missing	0	0
Skewness		.148	.852
Std. Error of Skewness		.177	.177
Kurtosis		-.621-	-.117-
Std. Error of Kurtosis		.353	.353

```
CORRELATIONS
  /VARIABLES=symboling price
  /PRINT=TWOTAIL NOSIG
  /MISSING=PAIRWISE.
```

Correlations

Correlations

		symboling	price
symboling	Pearson Correlation	1	-.097-
	Sig. (2-tailed)		.188
	N	188	188
price	Pearson Correlation	-.097-	1
	Sig. (2-tailed)	.188	
	N	188	188

Statistics

		wheelbase	carlength	carwidth	carheight	enginesize	boreratio	stroke	compressionratio	horsepower	peakrpm	price
N	Valid	188	188	188	188	188	188	188	188	188	188	188
	Missing	0	0	0	0	0	0	0	0	0	0	0
Skewness		.982	-.037-	.675	.104	.664	.112	-1.032-	2.758	.861	.059	.852
Std. Error of Skewness		.177	.177	.177	.177	.177	.177	.177	.177	.177	.177	.177
Kurtosis		.977	-.134-	.800	-.425-	-.161-	-.671-	2.350	6.067	.164	.351	-.117-
Std. Error of Kurtosis		.353	.353	.353	.353	.353	.353	.353	.353	.353	.353	.353

Correlations

Correlations

		wheelbase	carlength	carwidth	carheight	enginesize	boreratio	horsepower	peakrpm	price
wheelbase	Pearson Correlation	1	.856**	.785**	.626**	.549**	.504**	.369**	-.281-**	.652**
	Sig. (2-tailed)		.000	.000	.000	.000	.000	.000	.000	.000
	N	188	188	188	188	188	188	188	188	188
carlength	Pearson Correlation	.856**	1	.832**	.526**	.678**	.593**	.560**	-.233-**	.731**
	Sig. (2-tailed)	.000		.000	.000	.000	.000	.000	.001	.000
	N	188	188	188	188	188	188	188	188	188
carwidth	Pearson Correlation	.785**	.832**	1	.324**	.677**	.521**	.603**	-.151-*	.775**
	Sig. (2-tailed)	.000	.000		.000	.000	.000	.000	.039	.000
	N	188	188	188	188	188	188	188	188	188
carheight	Pearson Correlation	.626**	.526**	.324**	1	.153*	.202**	-.052-	-.296-**	.226**
	Sig. (2-tailed)	.000	.000	.000		.036	.005	.477	.000	.002
	N	188	188	188	188	188	188	188	188	188
enginesize	Pearson Correlation	.549**	.678**	.677**	.153*	1	.574**	.762**	-.303-**	.731**
	Sig. (2-tailed)	.000	.000	.000	.036		.000	.000	.000	.000
	N	188	188	188	188	188	188	188	188	188
boreratio	Pearson Correlation	.504**	.593**	.521**	.202**	.574**	1	.492**	-.295-**	.495**
	Sig. (2-tailed)	.000	.000	.000	.005	.000		.000	.000	.000
	N	188	188	188	188	188	188	188	188	188
horsepower	Pearson Correlation	.369**	.560**	.603**	-.052-	.762**	.492**	1	.133	.746**
	Sig. (2-tailed)	.000	.000	.000	.477	.000	.000		.069	.000
	N	188	188	188	188	188	188	188	188	188
peakrpm	Pearson Correlation	-.281-**	-.233-**	-.151-*	-.296-**	-.303-**	-.295-**	.133	1	-.083-
	Sig. (2-tailed)	.000	.001	.039	.000	.000	.000	.069		.255
	N	188	188	188	188	188	188	188	188	188
price	Pearson Correlation	.652**	.731**	.775**	.226**	.731**	.495**	.746**	-.083-	1
	Sig. (2-tailed)	.000	.000	.000	.002	.000	.000	.000	.255	
	N	188	188	188	188	188	188	188	188	188

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed).

Nonparametric Correlations

Correlations

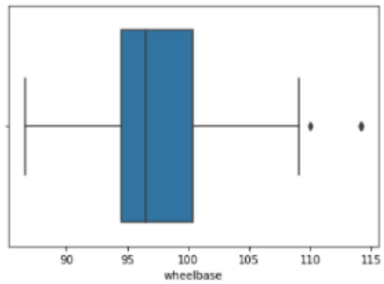
			stroke	compressionratio	price
Spearman's rho	stroke	Correlation Coefficient	1.000	-.056-	.123
		Sig. (2-tailed)	.	.449	.092
		N	188	188	188
	compressionratio	Correlation Coefficient	-.056-	1.000	-.200-**
		Sig. (2-tailed)	.449	.	.006
		N	188	188	188
	price	Correlation Coefficient	.123	-.200-**	1.000
		Sig. (2-tailed)	.092	.006	.
		N	188	188	188

**. Correlation is significant at the 0.01 level (2-tailed).

box plot visuals to the independent variables having relationship with price according to SPSS

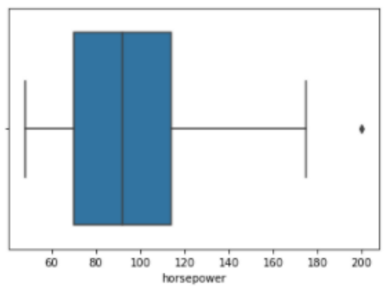
In [22]:
```
sns.boxplot(x=df_num['wheelbase'])
```

Out[22]: <AxesSubplot:xlabel='wheelbase'>



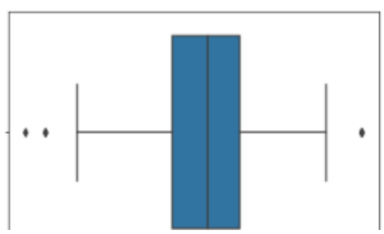
In [23]:
```
sns.boxplot(x=df_num['horsepower'])
```

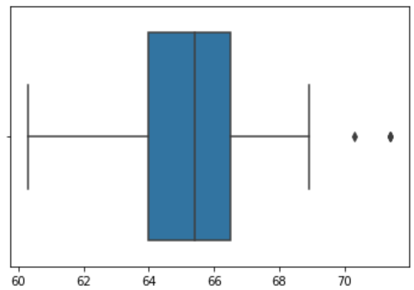
Out[23]: <AxesSubplot:xlabel='horsepower'>

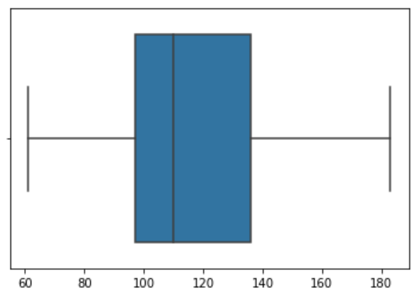


In [24]:
```
sns.boxplot(x=df_num['carlength'])
```

Out[24]: <AxesSubplot:xlabel='carlength'>



In [25]:
```
sns.boxplot(x=df_num['carwidth'])
```
Out[25]:
```
<AxesSubplot:xlabel='carwidth'>
```


In [26]:
```
sns.boxplot(x=df_num['enginesize'])
```
Out[26]:
```
<AxesSubplot:xlabel='enginesize'>
```


exporting a temporary csv file for the categorical data to deal with the categorical variables in SPSS

In [27]:
```
data_cleaned_temp=data_cleaned.copy()
```

In [28]:
```
data_cleaned_temp["carbody"]=data_cleaned_temp['carbody'].replace(to_replace =["convertible","hardtop","hatchback","sedan","wago
                            value =["1","2","3","4","5"])
data_cleaned_temp['carbody'] = data_cleaned_temp['carbody'].astype(int)
data_cleaned_temp["fueltype"]=data_cleaned_temp['fueltype'].replace(to_replace =["gas","diesel"],
                            value =["0","1"])
data_cleaned_temp['fueltype'] = data_cleaned_temp['fueltype'].astype(int)
data_cleaned_temp["aspiration"]=data_cleaned_temp['aspiration'].replace(to_replace =["std","turbo"],
                            value =["0","1"])
data_cleaned_temp['aspiration'] = data_cleaned_temp['aspiration'].astype(int)
data_cleaned_temp["cylindernumber"]=data_cleaned_temp['cylindernumber'].replace(to_replace =["two","three","four","five","six"],
                            value =["1","2","3","4","5"])
data_cleaned_temp['cylindernumber'] = data_cleaned_temp['cylindernumber'].astype(int)
data_cleaned_temp["doornumber"]=data_cleaned_temp['doornumber'].replace(to_replace =["two","four"],
                            value =["0","1"])
data_cleaned_temp['doornumber'] = data_cleaned_temp['doornumber'].astype(int)
data_cleaned_temp["enginetype"]=data_cleaned_temp['enginetype'].replace(to_replace =["ohc","dohc","ohcv","ohcf","rotor","l"],
                            value =["1","2","3","4","5","6"])
data_cleaned_temp['enginetype'] = data_cleaned_temp['enginetype'].astype(int)
data_cleaned_temp.to_csv('data_cleaned_temp.csv')
```

Descriptives

	fueltype			Statistic	Std. Error
price	"gas"	Mean		11120.60332	348.197067
		95% Confidence Interval for Mean	Lower Bound	10433.25649	
			Upper Bound	11807.95014	
		5% Trimmed Mean		10845.16450	
		Median		9549.00000	
		Variance		20732244.75	
		Std. Deviation		4553.267480	
		Minimum		5118.000	
		Maximum		24565.000	
		Range		19447.000	
		Interquartile Range		6990.000	
		Skewness		.844	.186
		Kurtosis		-.188-	.369
	"diesel"	Mean		13455.23529	1346.147385
		95% Confidence Interval for Mean	Lower Bound	10601.53032	
			Upper Bound	16308.94027	
		5% Trimmed Mean		13136.31699	
		Median		13200.00000	
		Variance		30805917.32	
		Std. Deviation		5550.307858	
		Minimum		7099.000	
		Maximum		25552.000	
		Range		18453.000	
		Interquartile Range		9566.000	
		Skewness		.721	.550
		Kurtosis		-.278-	1.063

Descriptives

	aspiration			Statistic	Std. Error
price	"std"	Mean		10489.31169	339.849981
		95% Confidence Interval for Mean	Lower Bound	9817.90736	
			Upper Bound	11160.71602	
		5% Trimmed Mean		10202.39394	
		Median		8948.50000	
		Variance		17786693.48	
		Std. Deviation		4217.427353	
		Minimum		5118.000	
		Maximum		24565.000	
		Range		19447.000	
		Interquartile Range		6030.000	
		Skewness		1.024	.195
		Kurtosis		.307	.389
	"turbo"	Mean		15147.29903	831.644551
		95% Confidence Interval for Mean	Lower Bound	13455.30547	
			Upper Bound	16839.29259	
		5% Trimmed Mean		15015.84206	
		Median		14679.00000	
		Variance		23515510.39	
		Std. Deviation		4849.279368	
		Minimum		7689.000	
		Maximum		25552.000	
		Range		17863.000	
		Interquartile Range		7351.250	
		Skewness		.224	.403
		Kurtosis		-.723-	.788

Descriptives

	doornumber			Statistic	Std. Error
price	"two"	Mean		10471.13169	483.166710
		95% Confidence Interval for Mean	Lower Bound	9509.59929	
			Upper Bound	11432.66409	
		5% Trimmed Mean		10220.66827	
		Median		9095.00000	
		Variance		18909455.67	
		Std. Deviation		4348.500393	
		Minimum		5118.000	
		Maximum		22018.000	
		Range		16900.000	
		Interquartile Range		6616.000	
		Skewness		.794	.267
		Kurtosis		-.384-	.529
	"four"	Mean		11983.18224	467.936772
		95% Confidence Interval for Mean	Lower Bound	11055.45208	
			Upper Bound	12910.91241	
		5% Trimmed Mean		11641.10644	
		Median		10245.00000	
		Variance		23429236.03	
		Std. Deviation		4840.375608	
		Minimum		6229.000	
		Maximum		25552.000	
		Range		19323.000	
		Interquartile Range		7852.000	
		Skewness		.862	.234
		Kurtosis		-.161-	.463

T-Test

Group Statistics

	fueltype	N	Mean	Std. Deviation	Std. Error Mean
price	"gas"	171	11120.60332	4553.267480	348.197067
	"diesel"	17	13455.23529	5550.307858	1346.147385

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
price	Equal variances assumed	1.200	.275	-1.975-	186	.050	-2334.63198-	1181.874037	-4666.23321-	-3.030743-
	Equal variances not assumed			-1.679-	18.205	.110	-2334.63198-	1390.450999	-5253.50565-	584.241696

T-Test

Group Statistics

	aspiration	N	Mean	Std. Deviation	Std. Error Mean
price	"std"	154	10489.31169	4217.427353	339.849981
	"turbo"	34	15147.29903	4849.279368	831.644551

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
price	Equal variances assumed	1.957	.163	-5.669-	186	.000	-4657.98734-	821.663117	-6278.96443-	-3037.01025-
	Equal variances not assumed			-5.185-	44.673	.000	-4657.98734-	898.404512	-6467.83242-	-2848.14226-

T-Test

Group Statistics

	doornumber	N	Mean	Std. Deviation	Std. Error Mean
price	"two"	81	10471.13169	4348.500393	483.166710
	"four"	107	11983.18224	4840.375608	467.936772

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
price	Equal variances assumed	1.520	.219	-2.215-	186	.028	-1512.05055-	682.676065	-2858.83397-	-165.267134-
	Equal variances not assumed			-2.248-	180.564	.026	-1512.05055-	672.617940	-2839.25293-	-184.848170-

	carbody			Statistic	Std. Error
price	"convertible"	Mean		14814.75000	1387.303762
		95% Confidence Interval for Mean	Lower Bound	10399.73027	
			Upper Bound	19229.76973	
		5% Trimmed Mean		14835.05556	
		Median		14997.50000	
		Variance		7698446.917	
		Std. Deviation		2774.607525	
		Minimum		11595.000	
		Maximum		17669.000	
		Range		6074.000	
		Interquartile Range		5306.750	
		Skewness		-.238-	1.014
		Kurtosis		-3.098-	2.619
	"hardtop"	Mean		9384.00000	678.337428
		95% Confidence Interval for Mean	Lower Bound	7225.22756	
			Upper Bound	11542.77244	
		5% Trimmed Mean		9346.22222	
		Median		9044.00000	
		Variance		1840566.667	
		Std. Deviation		1356.674857	
		Minimum		8249.000	
		Maximum		11199.000	
		Range		2950.000	
		Interquartile Range		2510.000	
		Skewness		.992	1.014
		Kurtosis		-.433-	2.619
	"hatchback"	Mean		10071.95894	508.832542
		95% Confidence Interval for Mean	Lower Bound	9056.59941	
			Upper Bound	11087.31847	
		5% Trimmed Mean		9803.72089	
		Median		8845.00000	
		Variance		17864828.38	
		Std. Deviation		4226.680539	
		Minimum		5118.000	
		Maximum		22018.000	
		Range		16900.000	
		Interquartile Range		6188.000	
		Skewness		.903	.289
		Kurtosis		-.099-	.570

	enginetype			Statistic	Std. Error
price	"ohc"	Mean		10675.91667	391.806574
		95% Confidence Interval for Mean	Lower Bound	9901.34192	
			Upper Bound	11450.49142	
		5% Trimmed Mean		10274.56703	
		Median		8921.00000	
		Variance		21798759.56	
		Std. Deviation		4668.914173	
		Minimum		5195.000	
		Maximum		25552.000	
		Range		20357.000	
		Interquartile Range		5643.250	
		Skewness		1.274	.203
		Kurtosis		.894	.404
	"dohc"	Mean		14959.70000	1023.794142
		95% Confidence Interval for Mean	Lower Bound	12643.71675	
			Upper Bound	17275.68325	
		5% Trimmed Mean		15070.88889	
		Median		15874.00000	
		Variance		10481544.46	
		Std. Deviation		3237.521344	
		Minimum		9298.000	
		Maximum		18620.000	
		Range		9322.000	
		Interquartile Range		4450.250	
		Skewness		-1.034-	.687
		Kurtosis		.018	1.334
	"ohcv"	Mean		16834.87500	1041.184944
		95% Confidence Interval for Mean	Lower Bound	14372.86383	
			Upper Bound	19296.88617	
		5% Trimmed Mean		16761.86111	
		Median		16849.50000	
		Variance		8672528.696	
		Std. Deviation		2944.915737	
		Minimum		13499.000	
		Maximum		21485.000	
		Range		7986.000	
		Interquartile Range		5650.000	
		Skewness		.292	.752
		Kurtosis		-1.145-	1.481

	cylindernumber			Statistic	Std. Error
price	"two"	Mean		13020.00000	1039.531144
		95% Confidence Interval for Mean	Lower Bound	9711.74795	
			Upper Bound	16328.25205	
		5% Trimmed Mean		12989.44444	
		Median		12745.00000	
		Variance		4322500.000	
		Std. Deviation		2079.062289	
		Minimum		10945.000	
		Maximum		15645.000	
		Range		4700.000	
		Interquartile Range		3975.000	
		Skewness		.577	1.014
		Kurtosis		-1.355-	2.619
	"four"	Mean		10285.75472	311.003940
		95% Confidence Interval for Mean	Lower Bound	9671.49330	
			Upper Bound	10900.01613	
		5% Trimmed Mean		10043.56115	
		Median		8948.00000	
		Variance		15379028.68	
		Std. Deviation		3921.610470	
		Minimum		5118.000	
		Maximum		22625.000	
		Range		17507.000	
		Interquartile Range		5091.000	
		Skewness		1.032	.192
		Kurtosis		.227	.383
	"five"	Mean		18738.89588	1452.417634
		95% Confidence Interval for Mean	Lower Bound	15304.47392	
			Upper Bound	22173.31783	
		5% Trimmed Mean		18662.82875	
		Median		17784.58350	
		Variance		16876135.86	
		Std. Deviation		4108.057431	
		Minimum		13295.000	
		Maximum		25552.000	
		Range		12257.000	
		Interquartile Range		6836.250	
		Skewness		.670	.752
		Kurtosis		-.227-	1.481

Kruskal-Wallis Test

Ranks

	carbody	N	Mean Rank
price	"convertible"	4	141.38
	"hardtop"	4	85.00
	"hatchback"	69	78.02
	"sedan"	87	103.21
	"wagon"	24	104.08
	Total	188	

Test Statistics[a,b]

	price
Kruskal-Wallis H	12.390
df	4
Asymp. Sig.	.015

a. Kruskal Wallis Test

b. Grouping Variable: carbody

Kruskal-Wallis Test

Ranks

	enginetype	N	Mean Rank
price	"ohc"	142	85.78
	"dohc"	10	141.00
	"ohcv"	8	156.94
	"ohcf"	12	65.13
	"rotor"	4	127.00
	"l"	12	135.88
	Total	188	

Test Statistics[a,b]

	price
Kruskal-Wallis H	33.345
df	5
Asymp. Sig.	.000

a. Kruskal Wallis Test

b. Grouping Variable: enginetype

Kruskal-Wallis Test

Ranks

	cylindernumber	N	Mean Rank
price	"two"	4	127.00
	"three"	1	2.00
	"four"	159	83.95
	"five"	8	165.00
	"six"	16	161.78
	Total	188	

Test Statistics[a,b]

	price
Kruskal-Wallis H	48.187
df	4
Asymp. Sig.	.000

a. Kruskal Wallis Test

b. Grouping Variable: cylindernumber

We found that :

aspiration

door number

car body

engine type

cylinder number

have significant change on price