car-sales-eda

August 7, 2023

[1]:
```
pip install numpy
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: numpy in c:\programdata\anaconda3\lib\site-
packages (1.21.5)
Note: you may need to restart the kernel to use updated packages.
```

[2]:
```
pip install pandas
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: pandas in c:\programdata\anaconda3\lib\site-
packages (1.4.2)
Requirement already satisfied: numpy>=1.18.5 in
c:\programdata\anaconda3\lib\site-packages (from pandas) (1.21.5)
Requirement already satisfied: pytz>=2020.1 in
c:\programdata\anaconda3\lib\site-packages (from pandas) (2021.3)
Requirement already satisfied: python-dateutil>=2.8.1 in
c:\programdata\anaconda3\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\programdata\anaconda3\lib\site-
packages (from python-dateutil>=2.8.1->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

[3]:
```
import numpy as np
import pandas as pd
import pandas_profiling
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
C:\Users\PC\AppData\Local\Temp\ipykernel_8812\3978922422.py:3:
DeprecationWarning: `import pandas_profiling` is going to be deprecated by April
1st. Please use `import ydata_profiling` instead.
  import pandas_profiling
```

[4]:
```
pip install ydata_profiling
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: ydata_profiling in
```

```
c:\users\pc\appdata\roaming\python\python39\site-packages (4.1.2)
Requirement already satisfied: statsmodels<0.14,>=0.13.2 in
c:\programdata\anaconda3\lib\site-packages (from ydata_profiling) (0.13.2)
Requirement already satisfied: PyYAML<6.1,>=5.0.0 in
c:\programdata\anaconda3\lib\site-packages (from ydata_profiling) (6.0)
Requirement already satisfied: jinja2<3.2,>=2.11.1 in
c:\programdata\anaconda3\lib\site-packages (from ydata_profiling) (2.11.3)
Requirement already satisfied: typeguard<2.14,>=2.13.2 in
c:\users\pc\appdata\roaming\python\python39\site-packages (from ydata_profiling)
(2.13.3)
Requirement already satisfied: matplotlib<3.7,>=3.2 in
c:\programdata\anaconda3\lib\site-packages (from ydata_profiling) (3.5.1)
Requirement already satisfied: tqdm<4.65,>=4.48.2 in
c:\programdata\anaconda3\lib\site-packages (from ydata_profiling) (4.64.0)
Requirement already satisfied: pandas!=1.4.0,<1.6,>1.1 in
c:\programdata\anaconda3\lib\site-packages (from ydata_profiling) (1.4.2)
Requirement already satisfied: multimethod<1.10,>=1.4 in
c:\users\pc\appdata\roaming\python\python39\site-packages (from ydata_profiling)
(1.9.1)
Requirement already satisfied: visions[type_image_path]==0.7.5 in
c:\users\pc\appdata\roaming\python\python39\site-packages (from ydata_profiling)
(0.7.5)
Requirement already satisfied: requests<2.29,>=2.24.0 in
c:\programdata\anaconda3\lib\site-packages (from ydata_profiling) (2.27.1)
Requirement already satisfied: imagehash==4.3.1 in
c:\users\pc\appdata\roaming\python\python39\site-packages (from ydata_profiling)
(4.3.1)
Requirement already satisfied: seaborn<0.13,>=0.10.1 in
c:\programdata\anaconda3\lib\site-packages (from ydata_profiling) (0.11.2)
Requirement already satisfied: scipy<1.10,>=1.4.1 in
c:\programdata\anaconda3\lib\site-packages (from ydata_profiling) (1.7.3)
Requirement already satisfied: htmlmin==0.1.12 in
c:\users\pc\appdata\roaming\python\python39\site-packages (from ydata_profiling)
(0.1.12)
Requirement already satisfied: phik<0.13,>=0.11.1 in
c:\users\pc\appdata\roaming\python\python39\site-packages (from ydata_profiling)
(0.12.3)
Requirement already satisfied: numpy<1.24,>=1.16.0 in
c:\programdata\anaconda3\lib\site-packages (from ydata_profiling) (1.21.5)
Requirement already satisfied: pydantic<1.11,>=1.8.1 in
c:\users\pc\appdata\roaming\python\python39\site-packages (from ydata_profiling)
(1.10.7)
Requirement already satisfied: pillow in c:\programdata\anaconda3\lib\site-
packages (from imagehash==4.3.1->ydata_profiling) (9.0.1)
Requirement already satisfied: PyWavelets in c:\programdata\anaconda3\lib\site-
packages (from imagehash==4.3.1->ydata_profiling) (1.3.0)
Requirement already satisfied: networkx>=2.4 in
c:\programdata\anaconda3\lib\site-packages (from
```

```
visions[type_image_path]==0.7.5->ydata_profiling) (2.7.1)
Requirement already satisfied: tangled-up-in-unicode>=0.0.4 in
c:\users\pc\appdata\roaming\python\python39\site-packages (from
visions[type_image_path]==0.7.5->ydata_profiling) (0.2.0)
Requirement already satisfied: attrs>=19.3.0 in
c:\programdata\anaconda3\lib\site-packages (from
visions[type_image_path]==0.7.5->ydata_profiling) (21.4.0)
Requirement already satisfied: MarkupSafe>=0.23 in
c:\programdata\anaconda3\lib\site-packages (from
jinja2<3.2,>=2.11.1->ydata_profiling) (2.0.1)
Requirement already satisfied: pyparsing>=2.2.1 in
c:\programdata\anaconda3\lib\site-packages (from
matplotlib<3.7,>=3.2->ydata_profiling) (3.0.4)
Requirement already satisfied: kiwisolver>=1.0.1 in
c:\programdata\anaconda3\lib\site-packages (from
matplotlib<3.7,>=3.2->ydata_profiling) (1.3.2)
Requirement already satisfied: fonttools>=4.22.0 in
c:\programdata\anaconda3\lib\site-packages (from
matplotlib<3.7,>=3.2->ydata_profiling) (4.25.0)
Requirement already satisfied: cycler>=0.10 in
c:\programdata\anaconda3\lib\site-packages (from
matplotlib<3.7,>=3.2->ydata_profiling) (0.11.0)
Requirement already satisfied: python-dateutil>=2.7 in
c:\programdata\anaconda3\lib\site-packages (from
matplotlib<3.7,>=3.2->ydata_profiling) (2.8.2)
Requirement already satisfied: packaging>=20.0 in
c:\programdata\anaconda3\lib\site-packages (from
matplotlib<3.7,>=3.2->ydata_profiling) (21.3)
Requirement already satisfied: pytz>=2020.1 in
c:\programdata\anaconda3\lib\site-packages (from
pandas!=1.4.0,<1.6,>1.1->ydata_profiling) (2021.3)
Requirement already satisfied: joblib>=0.14.1 in
c:\programdata\anaconda3\lib\site-packages (from
phik<0.13,>=0.11.1->ydata_profiling) (1.1.0)
Requirement already satisfied: typing-extensions>=4.2.0 in
c:\users\pc\appdata\roaming\python\python39\site-packages (from
pydantic<1.11,>=1.8.1->ydata_profiling) (4.5.0)
Requirement already satisfied: six>=1.5 in c:\programdata\anaconda3\lib\site-
packages (from python-dateutil>=2.7->matplotlib<3.7,>=3.2->ydata_profiling)
(1.16.0)
Requirement already satisfied: charset-normalizer~=2.0.0 in
c:\programdata\anaconda3\lib\site-packages (from
requests<2.29,>=2.24.0->ydata_profiling) (2.0.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
c:\programdata\anaconda3\lib\site-packages (from
requests<2.29,>=2.24.0->ydata_profiling) (1.26.9)
Requirement already satisfied: idna<4,>=2.5 in
c:\programdata\anaconda3\lib\site-packages (from
```

```
requests<2.29,>=2.24.0->ydata_profiling) (3.3)
Requirement already satisfied: certifi>=2017.4.17 in
c:\programdata\anaconda3\lib\site-packages (from
requests<2.29,>=2.24.0->ydata_profiling) (2021.10.8)
Requirement already satisfied: patsy>=0.5.2 in
c:\programdata\anaconda3\lib\site-packages (from
statsmodels<0.14,>=0.13.2->ydata_profiling) (0.5.2)
Requirement already satisfied: colorama in c:\programdata\anaconda3\lib\site-
packages (from tqdm<4.65,>=4.48.2->ydata_profiling) (0.4.4)
Note: you may need to restart the kernel to use updated packages.
```

[5]:
```
import ydata_profiling as pandas_profiling
```

[6]:
```
carsales=pd.read_csv("D:\P EDA\car_sales.csv")
```

[7]:
```
car_data=carsales.copy()
```

[8]:
```
car_data.head(10)
```

[8]:

	car	price	body	mileage	engV	engType	registration	year	model	drive
0	Ford	15500.0	crossover	68	2.5	Gas	yes	2010	Kuga	full
1	Mercedes-Benz	20500.0	sedan	173	1.8	Gas	yes	2011	E-Class	rear
2	Mercedes-Benz	35000.0	other	135	5.5	Petrol	yes	2008	CL 550	rear
3	Mercedes-Benz	17800.0	van	162	1.8	Diesel	yes	2012	B 180	front
4	Mercedes-Benz	33000.0	vagon	91	NaN	Other	yes	2013	E-Class	NaN
5	Nissan	16600.0	crossover	83	2.0	Petrol	yes	2013	X-Trail	full
6	Honda	6500.0	sedan	199	2.0	Petrol	yes	2003	Accord	front
7	Renault	10500.0	vagon	185	1.5	Diesel	yes	2011	Megane	front
8	Mercedes-Benz	21500.0	sedan	146	1.8	Gas	yes	2012	E-Class	rear
9	Mercedes-Benz	22700.0	sedan	125	2.2	Diesel	yes	2010	E-Class	rear

[9]:
```
car_data.shape
```

[9]:
```
(9576, 10)
```

[10]:
```
car_data.dtypes
```

[10]:
```
car              object
price           float64
body             object
mileage           int64
engV            float64
engType          object
registration     object
year              int64
model            object
drive            object
dtype: object
```

[13]:
```
car_data.describe(include='all')
```

[13]:

	car	price	body	mileage	engV	engType	\
count	9576	9576.000000	9576	9576.000000	9142.000000	9576	
unique	87	NaN	6	NaN	NaN	4	
top	Volkswagen	NaN	sedan	NaN	NaN	Petrol	
freq	936	NaN	3646	NaN	NaN	4379	
mean	NaN	15633.317316	NaN	138.862364	2.646344	NaN	
std	NaN	24106.523436	NaN	98.629754	5.927699	NaN	
min	NaN	0.000000	NaN	0.000000	0.100000	NaN	
25%	NaN	4999.000000	NaN	70.000000	1.600000	NaN	
50%	NaN	9200.000000	NaN	128.000000	2.000000	NaN	
75%	NaN	16700.000000	NaN	194.000000	2.500000	NaN	
max	NaN	547800.000000	NaN	999.000000	99.990000	NaN	

	registration	year	model	drive
count	9576	9576.000000	9576	9065
unique	2	NaN	888	3
top	yes	NaN	E-Class	front
freq	9015	NaN	199	5188
mean	NaN	2006.605994	NaN	NaN
std	NaN	7.067924	NaN	NaN
min	NaN	1953.000000	NaN	NaN
25%	NaN	2004.000000	NaN	NaN
50%	NaN	2008.000000	NaN	NaN
75%	NaN	2012.000000	NaN	NaN
max	NaN	2016.000000	NaN	NaN

[14]:
```
car_data.isnull().sum()
```

[14]:
```
car              0
price            0
body             0
```

```
mileage           0
engV            434
engType           0
registration      0
year              0
model             0
drive           511
dtype: int64
```

[ ]:
```
profile=pandas_profiling.ProfileReport(car_data)
profile.to_file(output_file='car_data_before_EDA.html')
```

[15]:
```
car_data['price'].isnull().sum()
```

[15]:
```
0
```

[16]:
```
car_data.duplicated().sum()
```

[16]:
```
113
```

[19]:
```
car_data.drop_duplicates(keep='first', inplace=True)
```

[20]:
```
car_data.duplicated().sum()
```

[20]:
```
0
```

[21]:
```
car_data['car'].value_counts()
```

[21]:
```
Volkswagen       927
Mercedes-Benz    885
BMW              684
Toyota           529
VAZ              488
                ...
ZX                 1
Other-Retro        1
Mercury            1
Maserati           1
Buick              1
Name: car, Length: 87, dtype: int64
```

[22]:
```
car_data['body'].value_counts()
```

[22]:
```
sedan        3622
crossover    2007
hatch        1248
van          1038
```

```
other         829
vagon         719
Name: body, dtype: int64
```

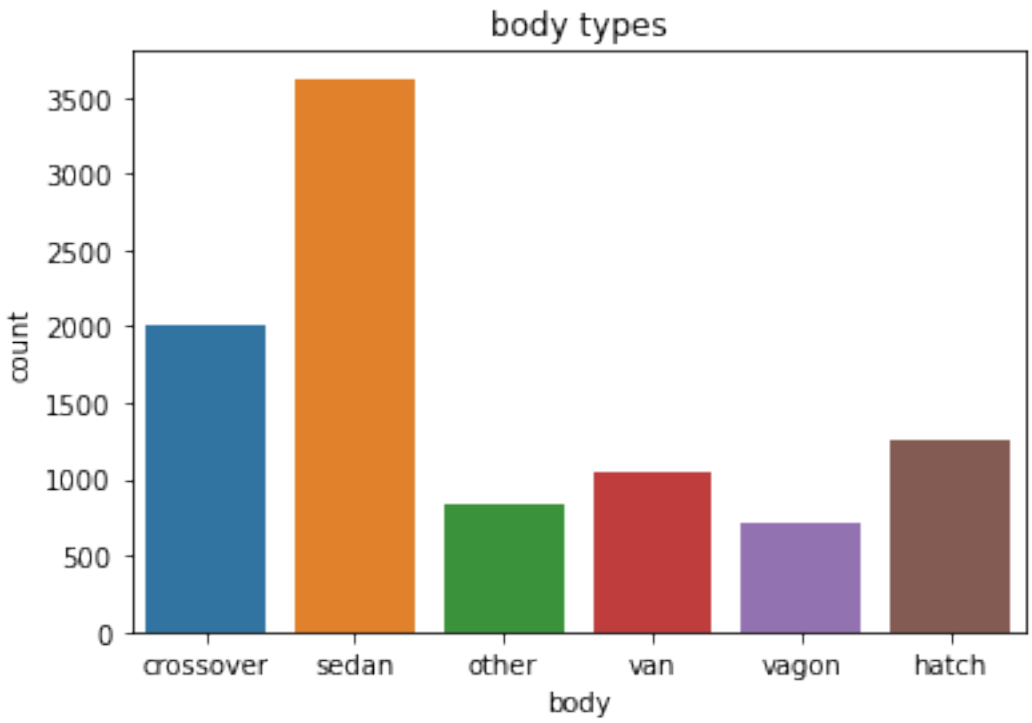
[23]:
```
car_data.sort_values(by=['price'],ascending=False)
```

[23]:
```
                car     price       body  mileage  engV engType registration  \
7621        Bentley  547800.0      sedan        0  6.75  Petrol          yes
1611        Bentley  499999.0  crossover        0  6.00  Petrol          yes
4134        Bentley  449999.0  crossover        0  6.00  Petrol          yes
4325  Mercedes-Benz  300000.0      sedan       68  6.00  Petrol          yes
5849  Mercedes-Benz  300000.0      other       37  5.00  Petrol          yes
…               …       …         …       …    …      …           …
4107            VAZ       0.0      vagon       39  1.30     Gas          yes
656            Audi       0.0  crossover        1  3.00  Diesel          yes
5563            BMW       0.0      sedan       65  2.00  Petrol          yes
2229     Volkswagen       0.0      vagon      160  1.90  Diesel          yes
8772            BMW       0.0      sedan       99  4.40  Petrol          yes

      year       model  drive
7621  2016    Mulsanne   rear
1611  2016    Bentayga   full
4134  2016    Bentayga   full
4325  2011       S 600    NaN
5849  2012       G 500   full
…     …          …    …
4107  1988        2104   rear
656   2016  A6 Allroad   full
5563  2012         320   rear
2229  2003   Passat B5  front
8772  2013      Alpina   full

[9463 rows x 10 columns]
```

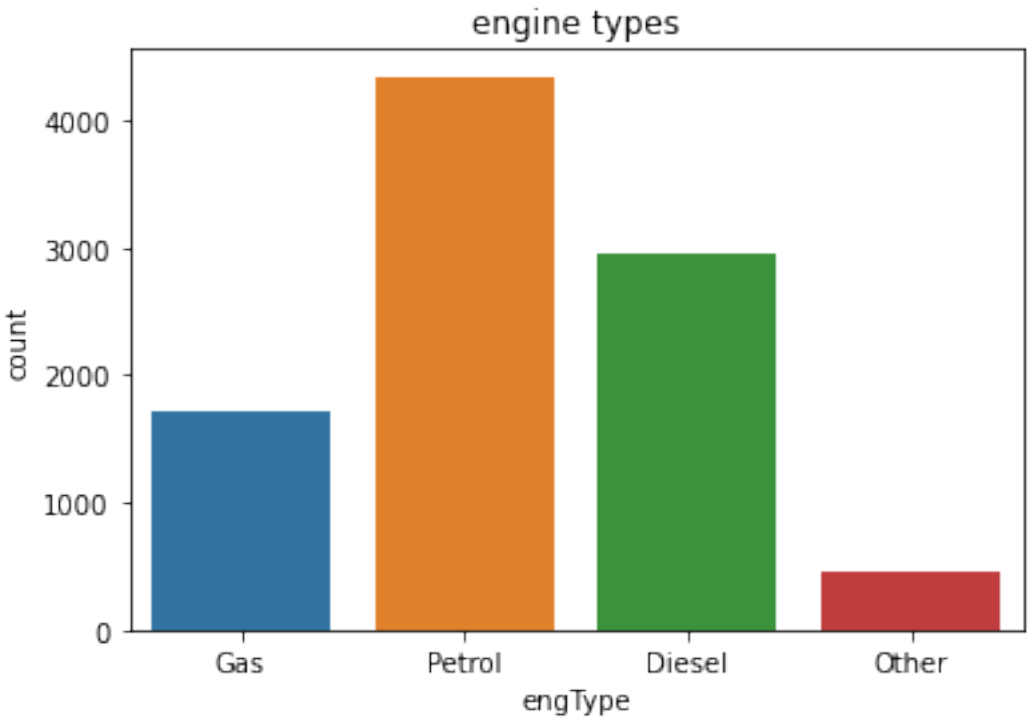
[25]:
```
sns.countplot(x='body',data=car_data)
plt.title('body types')
plt.show()
```



[26]:
```
car_data['engType'].value_counts()
```

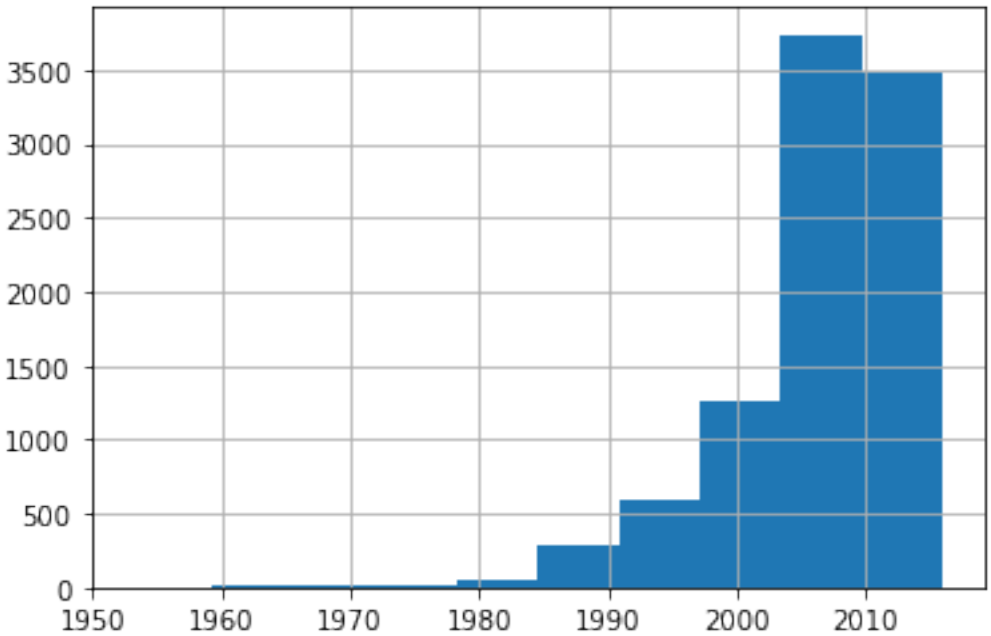
[26]:
```
Petrol    4341
Diesel    2950
Gas       1710
Other      462
Name: engType, dtype: int64
```

[28]:
```
sns.countplot(x='engType',data=car_data)
plt.title('engine types')
plt.show()
```



[29]:
```
car_data['year'].hist()
```

[29]: <AxesSubplot:>



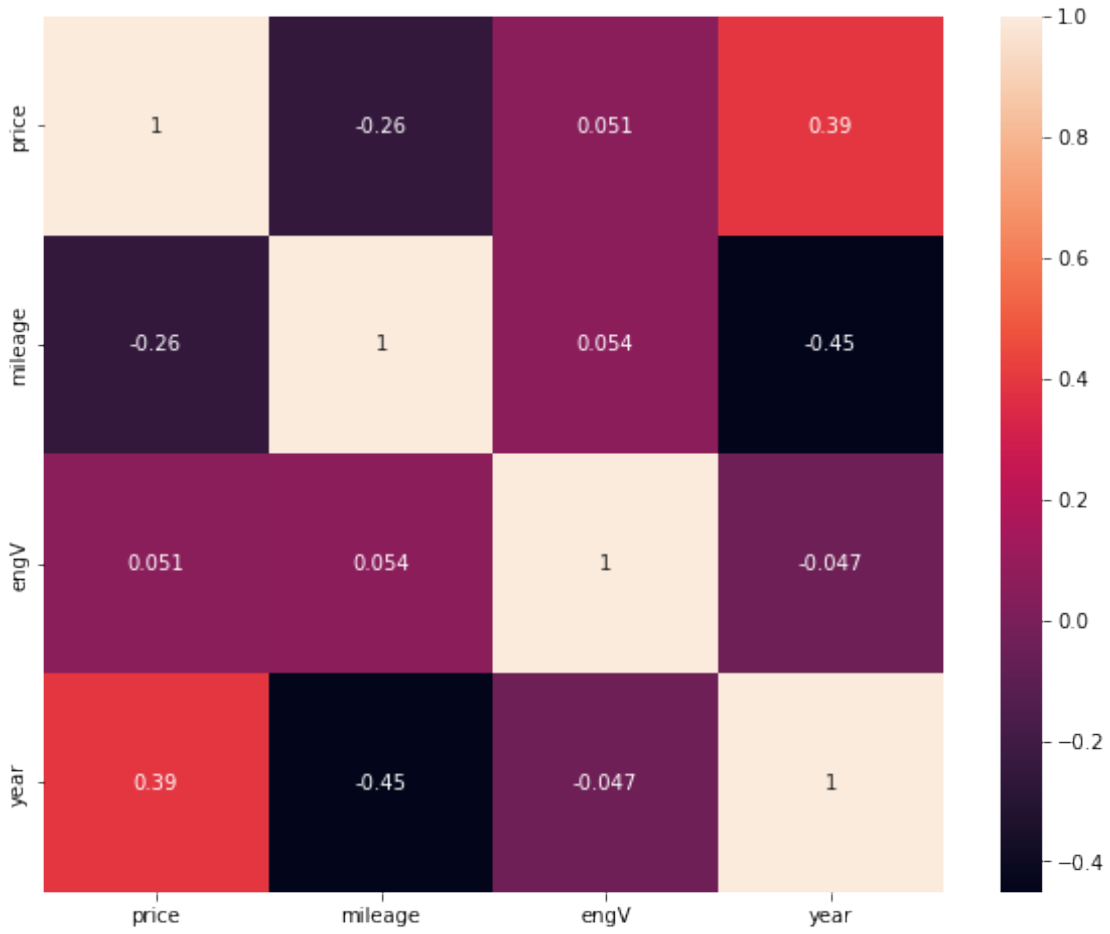
[30]:
```
car_data.corr()
```

[30]:

	price	mileage	engV	year
price	1.000000	-0.312965	0.049641	0.375440
mileage	-0.312965	1.000000	0.048746	-0.489326
engV	0.049641	0.048746	1.000000	-0.043509
year	0.375440	-0.489326	-0.043509	1.000000

[95]:
```
plt.subplots(figsize=(10,8))
sns.heatmap(car_data.corr(),annot=True)
```

[95]: <AxesSubplot:>



[32]:
```
car_data.isnull().sum()
```

[32]:
```
car               0
price             0
body              0
mileage           0
engV            434
engType           0
registration      0
year              0
model             0
drive           510
dtype: int64
```

[33]:
```
car_data['drive'].isnull().sum()
```

[33]:
```
510
```

[34]:
```
car_data['drive'].mode()
```

[34]:
```
0    front
Name: drive, dtype: object
```

[35]:
```
car_data.loc[car_data['drive'].isnull()]
```

[35]:

	car	price	body	mileage	engV	engType	registration	year	model	drive
4	Mercedes-Benz	33000.0	vagon	91	NaN	Other	yes	2013	E-Class	NaN
37	Audi	2850.0	sedan	260	NaN	Other	no	1999	A6	NaN
44	BMW	39333.0	sedan	6	2.00	Petrol	yes	2016	520	NaN
52	Mercedes-Benz	31500.0	sedan	123	2.20	Diesel	yes	2011	E-Class	NaN
103	Volkswagen	10000.0	van	231	1.90	Diesel	yes	2005	T5 (Transporter) ïàññ.	NaN
…	…	…	…	…	…	…	…	…	…	…
9445	Nissan	5000.0	sedan	260	3.00	Gas	yes	2000	Maxima	NaN
9450	VAZ	750.0	sedan	123	1.20	Petrol	yes	1990	2105	NaN
9469	Renault	5650.0	hatch	175	99.99	Other	yes	2002	Laguna	NaN
9537	Volkswagen	11500.0	other	51	1.60	Petrol	yes	2013	Polo	NaN
9566	UAZ	850.0	van	255	NaN	Other	yes			

```
9566  1981                    3962   NaN

[510 rows x 10 columns]
```

[36]:
```
car_data['drive'].fillna('front',inplace=True)
```

[37]:
```
car_data['drive'].isnull().sum()
```

[37]:
```
0
```

[38]:
```
car_data['engV'].isnull().sum()
```

[38]:
```
434
```

[39]:
```
car_data['engV'].mode().loc[0]
```

[39]:
```
2.0
```

[40]:
```
eng_mode=car_data['engV'].mode().loc[0]
```

[41]:
```
car_data['engV'].fillna(eng_mode,inplace=True)
```

[42]:
```
car_data['engV'].isnull().sum()
```

[42]:
```
0
```

[43]:
```
car_data.isnull().sum()
```

[43]:
```
car             0
price           0
body            0
mileage         0
engV            0
engType         0
registration    0
year            0
model           0
drive           0
dtype: int64
```

[44]:
```
(car_data['price']<=0).value_counts()
```

[44]:
```
False    9223
True      240
Name: price, dtype: int64
```

[45]:
```
car_data[car_data['price']<=0]
```

[45]:
```
                car  price       body  mileage  engV engType registration  \
20       Land Rover    0.0  crossover        0   4.4  Diesel          yes
53    Mercedes-Benz    0.0  crossover        0   3.0  Diesel          yes
71           Toyota    0.0  crossover        0   4.5  Diesel          yes
90          Porsche    0.0      sedan       22   4.8  Petrol          yes
92             Audi    0.0  crossover        0   3.0  Diesel          yes
…               …    …        …       …   …      …           …
9019         Toyota    0.0      hatch       76   1.0  Petrol          yes
9025  Mercedes-Benz    0.0  crossover        1   3.0  Petrol          yes
9036           Ford    0.0      other        1   5.0  Petrol          yes
9442        Renault    0.0      vagon      137   1.9  Diesel          yes
9470       Chrysler    0.0      vagon      198   2.0  Petrol          yes

      year             model  drive
20    2016       Range Rover   full
53    2016         GLE-Class   full
71    2016  Land Cruiser 200   full
90    2014          Panamera   full
92    2015                Q7   full
…     …                …    …
9019  2007              Aygo  front
9025  2016         GLE-Class   full
9036  2014           Mustang   rear
9442  2008      Kangoo ïàññ.  front
9470  2001        PT Cruiser  front

[240 rows x 10 columns]
```

[46]:
```
car_data.drop(car_data[car_data['price'] <= 0].index,inplace=True)
```

[47]:
```
car_data[car_data['price']<=0]
```

[47]:
```
Empty DataFrame
Columns: [car, price, body, mileage, engV, engType, registration, year, model,
drive]
Index: []
```

[48]:
```
car_data[car_data['mileage']<=0]
```

[48]:
```
                car       price       body  mileage  engV engType  \
10           Nissan  20447.1540  crossover        0   1.2  Petrol
17    Mercedes-Benz  99999.0000  crossover        0   3.0  Petrol
21           Nissan  26033.5530  crossover        0   1.6  Diesel
24              BMW  65099.0000  crossover        0   2.0  Diesel
26    Mercedes-Benz  69999.0000  crossover        0   2.2  Diesel
…               …         …       …       …   …      …
9234        Hyundai  12800.7750      hatch        0   1.4  Petrol
```

```
9268         Subaru  37500.0000  crossover        0   2.0  Diesel
9382         Suzuki  15486.9000      hatch        0   1.2  Petrol
9483           Opel  20120.0000      sedan        0   1.6  Diesel
9484         Nissan  29077.9515  crossover        0   1.6  Diesel

     registration  year       model  drive
10            yes  2016     Qashqai  front
17            yes  2016   GLE-Class   full
21            yes  2016     X-Trail   full
24            yes  2016          X5   full
26            yes  2016   GLE-Class   full
…             …    …          …     …
9234          yes  2016     Solaris  front
9268          yes  2016    Forester   full
9382          yes  2016       Swift  front
9483          yes  2016     Astra J  front
9484          yes  2016     X-Trail  front

[283 rows x 10 columns]
```

[49]:
```
a=car_data['mileage'].median()
a
```

[49]:
```
130.0
```

[50]:
```
car_data['mileage']=car_data['mileage'].replace(0, a)
```

[51]:
```
car_data[car_data['mileage']<=0]
```

[51]:
```
Empty DataFrame
Columns: [car, price, body, mileage, engV, engType, registration, year, model,
drive]
Index: []
```

[52]:
```
car_data['price']=car_data['price'].round(2)
```

[53]:
```
car_data['price']
```

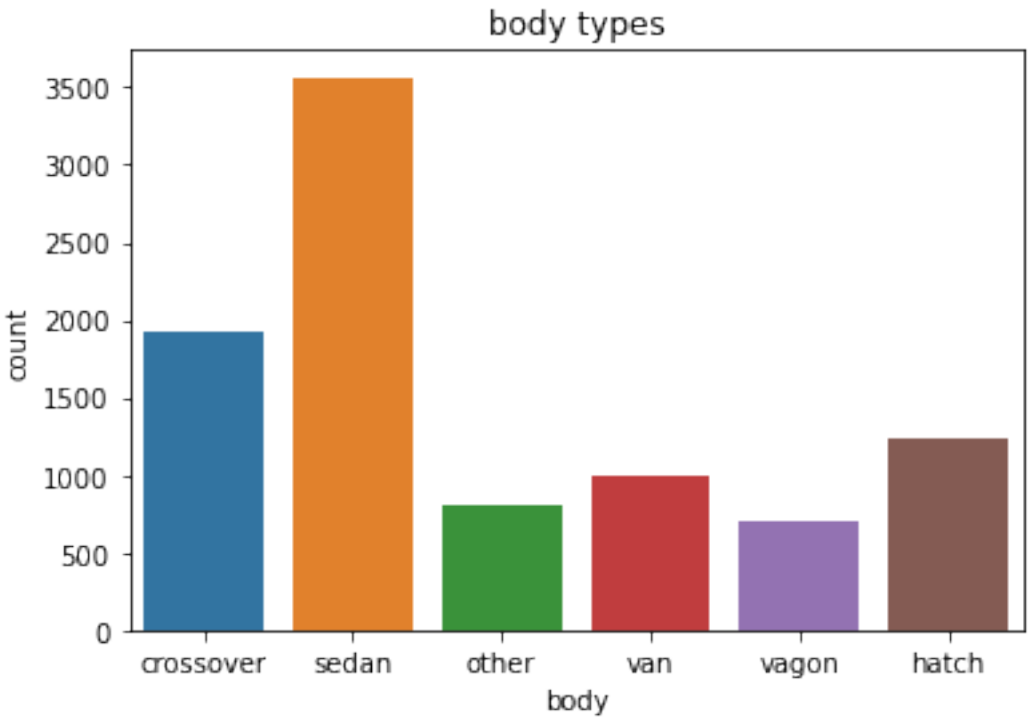
[53]:
```
0       15500.0
1       20500.0
2       35000.0
3       17800.0
4       33000.0
         …
9571    14500.0
9572     2200.0
9573    18500.0
```

```
9574    16999.0
9575    22500.0
Name: price, Length: 9223, dtype: float64
```

[ ]:
```
profile=pandas_profiling.ProfileReport(car_data)
profile.to_file(output_file='car_data_after_EDA.html')
```

1 1.Which type of cars are sold maximum?

[54]:
```
sns.countplot(x='body',data=car_data)
plt.title('body types')
plt.show()
```



2 What is the co relation between price and mileage

[55]:
```
car_data.corr()
```

[55]:
```
             price   mileage      engV      year
price     1.000000 -0.255992  0.050786  0.391502
mileage  -0.255992  1.000000  0.053541 -0.451259
engV      0.050786  0.053541  1.000000 -0.046806
```

```
year      0.391502 -0.451259 -0.046806  1.000000
```

[ ]:
```
#As the 'mileage' of the car increases,the 'price' will slightly decreases
```

3 How many cars are registered

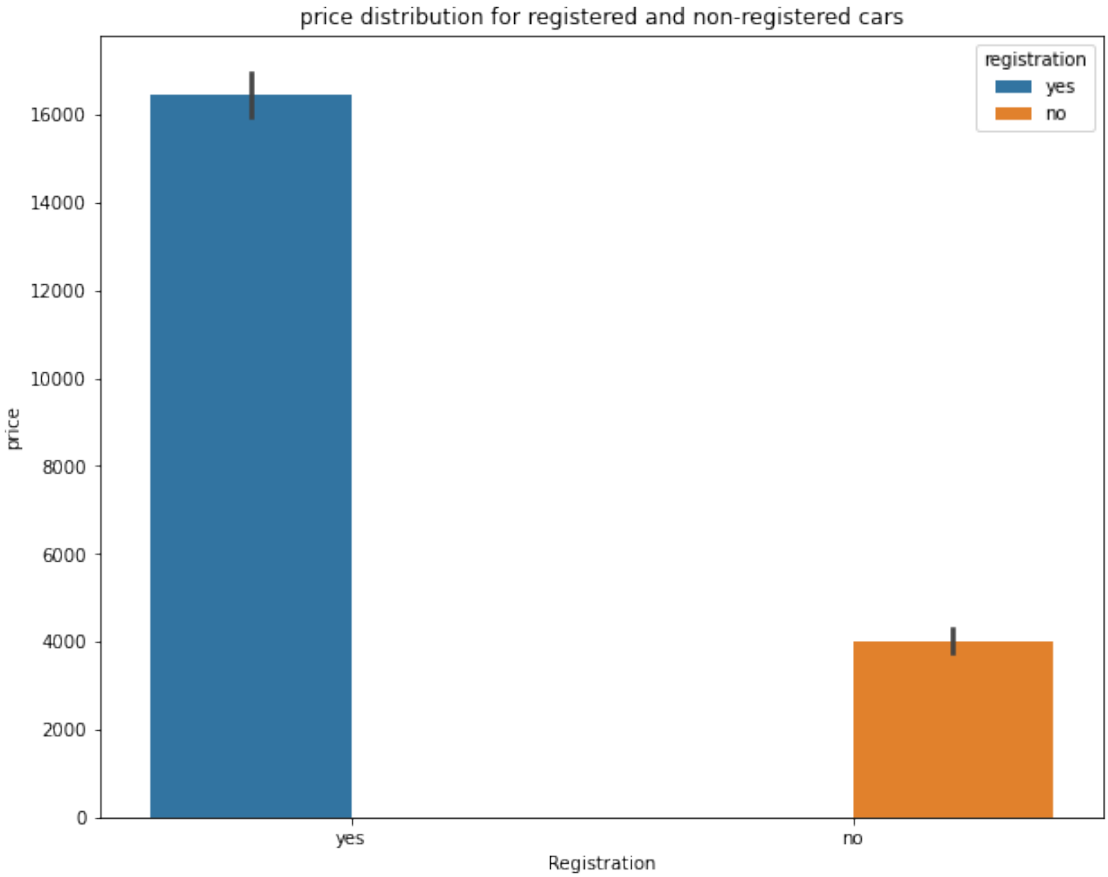
[56]:
```
car_data['registration'].value_counts()
```

[56]:
```
yes    8669
no      554
Name: registration, dtype: int64
```

[ ]:
```
# 8669 cars are registered
```

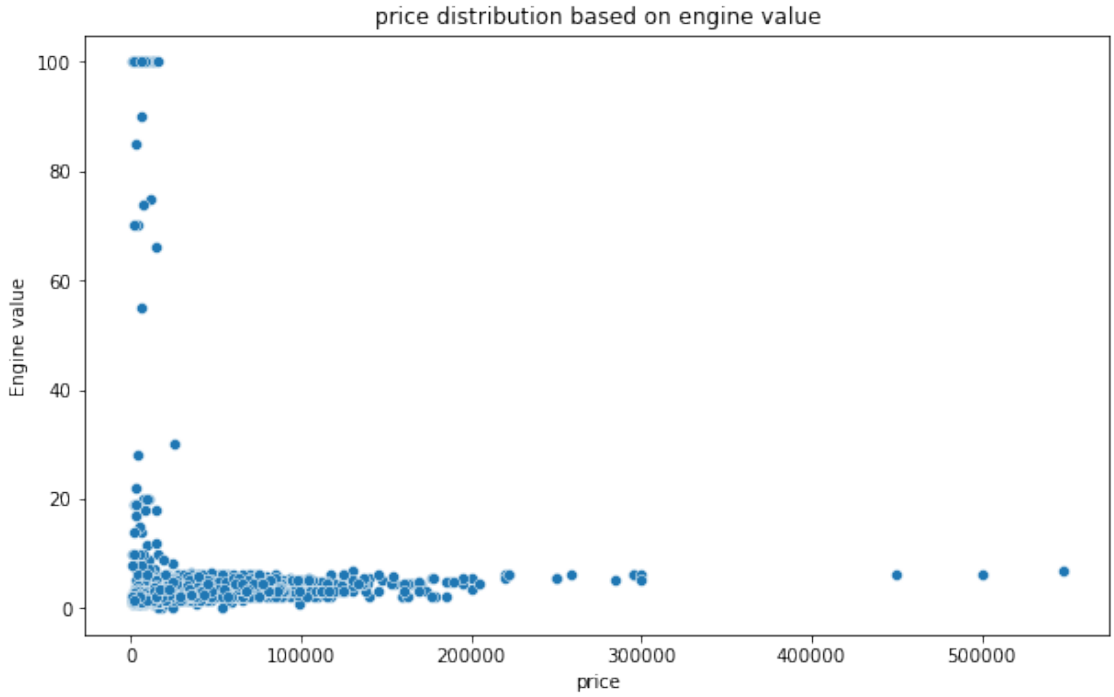
4 Price distribution between registered and non-registered cars

[57]:
```
plt.figure(figsize=(10,8))
sns.barplot(data=car_data,x='registration',y='price',hue='registration')
plt.xlabel('Registration')
plt.ylabel('price')
plt.title('price distribution for registered and non-registered cars')
plt.show()
```



5 What is the car price distribution based on Engine value

[85]:
```
plt.figure(figsize=(10,6))
sns.scatterplot(data=car_data,x='price',y='engV')
plt.xlabel('price')
plt.ylabel('Engine value')
plt.title('price distribution based on engine value')
plt.show()
```



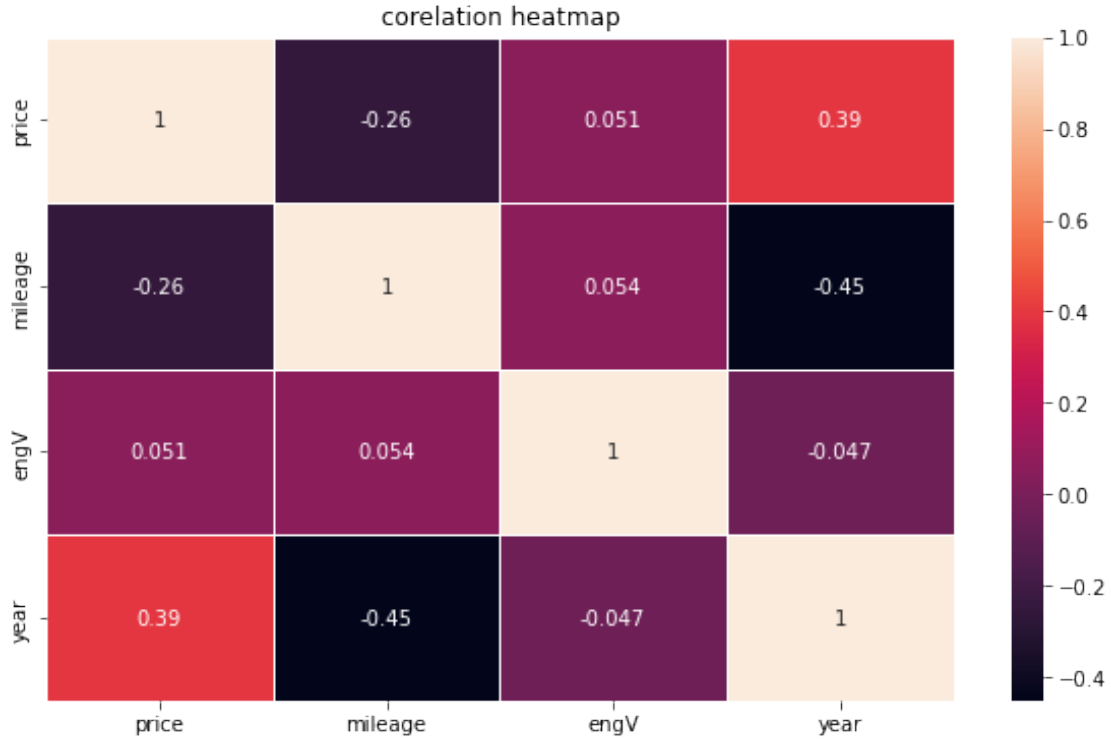
6 Which engine type of car users prefered maximum

[89]:
```
a=car_data['body'].value_counts().head(1)
a
```

[89]:
```
sedan    3564
Name: body, dtype: int64
```

7 Establish corelation between all features using heatmap

[94]:
```
plt.figure(figsize=(10,6))
sns.heatmap(data=car_data.corr(),annot=True,linewidths=0.5)
plt.title('corelation heatmap')
plt.show()
```


corelation heatmap
price
mileage
engV
year
1
-0.26
0.051
0.39
-0.26
1
0.054
-0.45
0.051
0.054
1
-0.047
0.39
-0.45
-0.047
1
price
mileage
engV
year
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

[ ]: