Задание

Выбрать набор данных (датасет). Создать ноутбук, который содержит следующие разделы: Текстовое описание выбранного Вами набора данных. Основные характеристики датасета. Визуальное исследование датасета. Информация о корреляции признаков. Сформировать отчет и разместить его в своем репозитории на github.

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
```

In [2]:
```
data = pd.read_csv('auto.csv', sep=",")
```

In [3]:
```
data.head()
```

Out[3]:

	symboling	normalized-losses	make	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	length	...
0	3	122	alfa-romero	std	two	convertible	rwd	front	88.6	0.811148	...
1	3	122	alfa-romero	std	two	convertible	rwd	front	88.6	0.811148	...
2	1	122	alfa-romero	std	two	hatchback	rwd	front	94.5	0.822681	...
3	2	164	audi	std	four	sedan	fwd	front	99.8	0.848630	...
4	2	164	audi	std	four	sedan	4wd	front	99.4	0.848630	...

5 rows × 29 columns

In [4]:
```
data.shape
```

Out[4]:
```
(201, 29)
```

In [5]:
```
total_count = data.shape[0]
print('Всего строк: {}'.format(total_count))
```

```
Всего строк: 201
```

In [6]:
```
data.columns
```

Out[6]:
```
Index(['symboling', 'normalized-losses', 'make', 'aspiration', 'num-of-doors',
       'body-style', 'drive-wheels', 'engine-location', 'wheel-base', 'length',
       'width', 'height', 'curb-weight', 'engine-type', 'num-of-cylinders',
       'engine-size', 'fuel-system', 'bore', 'stroke', 'compression-ratio',
       'horsepower', 'peak-rpm', 'city-mpg', 'highway-mpg', 'price',
       'city-L/100km', 'horsepower-binned', 'diesel', 'gas'],
      dtype='object')
```

In [7]:
```
data.dtypes
```

Out[7]:
```
symboling               int64
normalized-losses       int64
make                   object
aspiration             object
num-of-doors           object
body-style             object
drive-wheels           object
engine-location        object
wheel-base            float64
length                float64
width                 float64
height                float64
curb-weight             int64
engine-type            object
num-of-cylinders       object
engine-size             int64
fuel-system            object
bore                  float64
stroke                float64
compression-ratio     float64
horsepower            float64
peak-rpm              float64
city-mpg                int64
highway-mpg             int64
price                 float64
city-L/100km          float64
horsepower-binned      object
diesel                  int64
gas                     int64
dtype: object
```

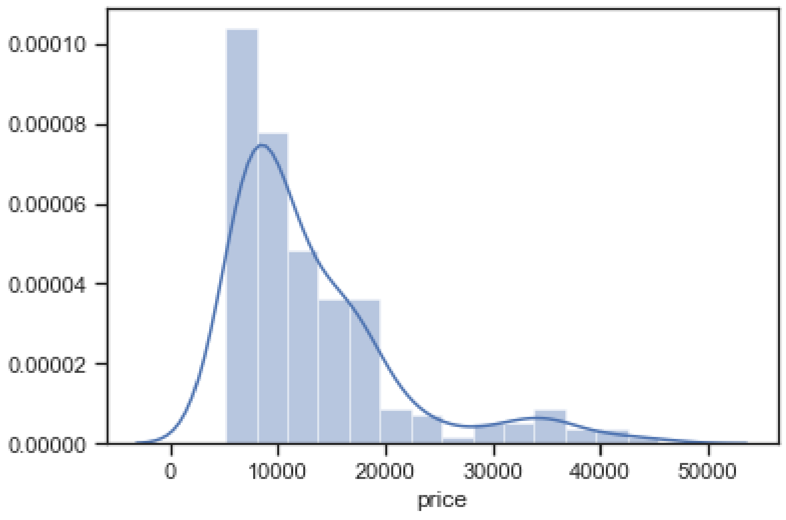
In [8]:
```
data.describe()
```

Out[8]:

	symboling	normalized-losses	wheel-base	length	width	height	curb-weight	engine-size
count	201.000000	201.00000	201.000000	201.000000	201.000000	201.000000	201.000000	201.000000
mean	0.840796	122.00000	98.797015	0.837102	0.915126	53.766667	2555.666667	126.875622
std	1.254802	31.99625	6.066366	0.059213	0.029187	2.447822	517.296727	41.546834
min	-2.000000	65.00000	86.600000	0.678039	0.837500	47.800000	1488.000000	61.000000
25%	0.000000	101.00000	94.500000	0.801538	0.890278	52.000000	2169.000000	98.000000
50%	1.000000	122.00000	97.000000	0.832292	0.909722	54.100000	2414.000000	120.000000
75%	2.000000	137.00000	102.400000	0.881788	0.925000	55.500000	2926.000000	141.000000
max	3.000000	256.00000	120.900000	1.000000	1.000000	59.800000	4066.000000	326.000000

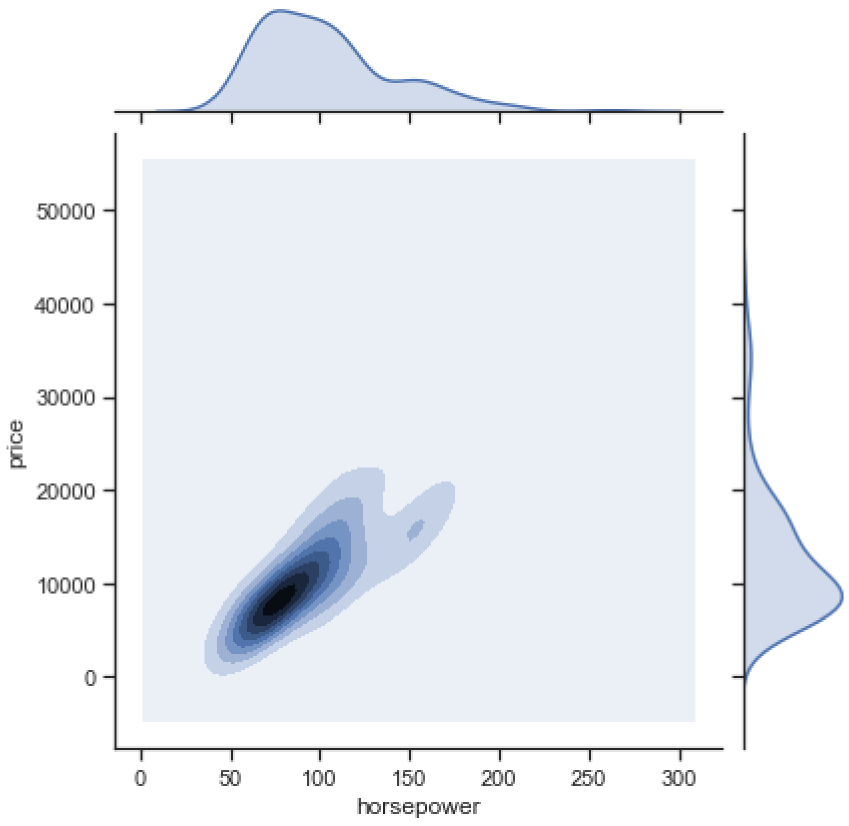
```
In [9]:  sns.distplot(data['price'])
```

```
Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x2b5dce67208>
```

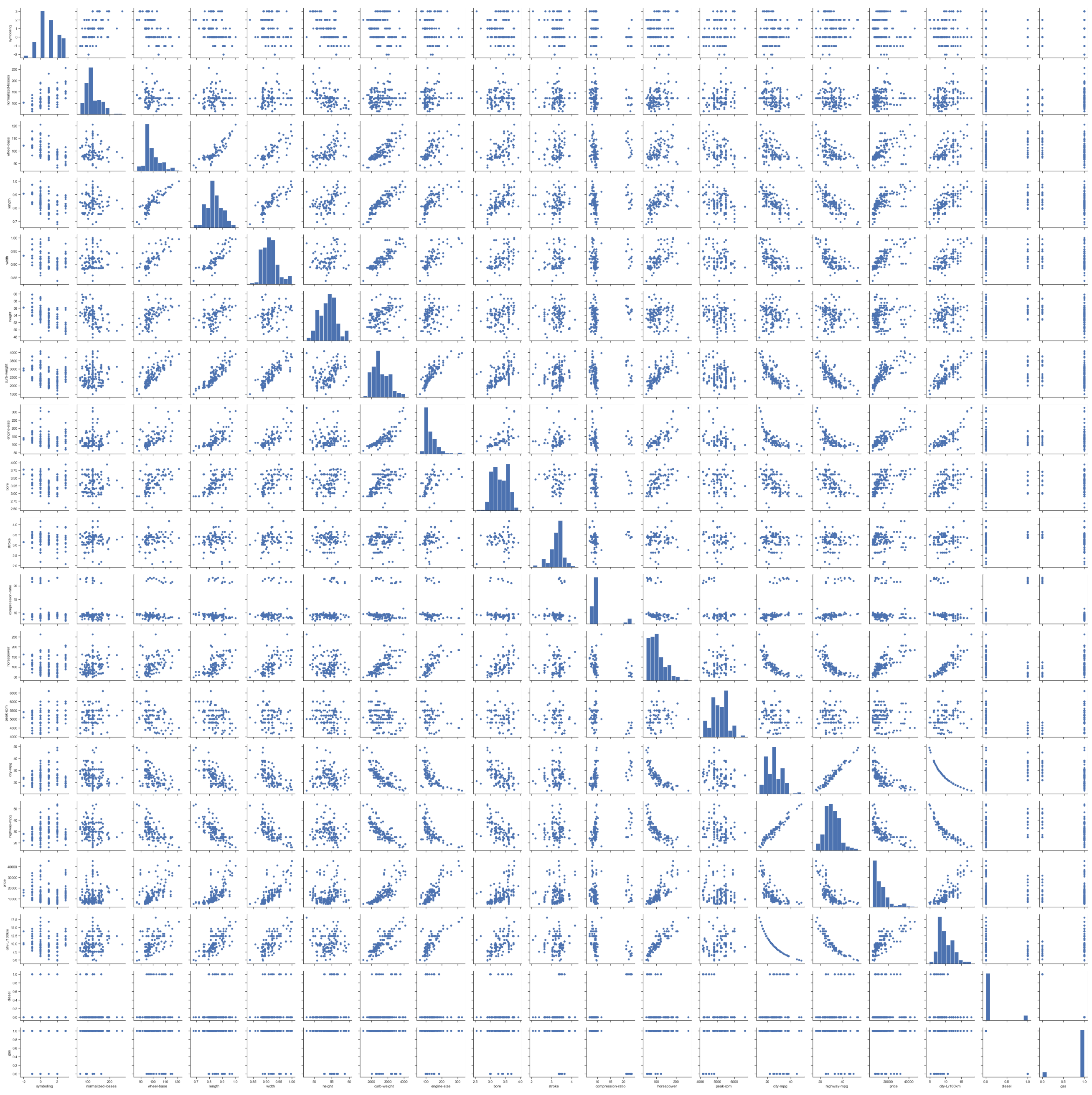


```
In [10]: sns.jointplot(x='horsepower', y='price', data=data, kind="kde")
```

```
Out[10]: <seaborn.axisgrid.JointGrid at 0x2b5deeaecc0>
```



In [11]:
```python
import warnings
warnings.filterwarnings('ignore')
sns.pairplot(data, plot_kws=dict(linewidth=0));
```



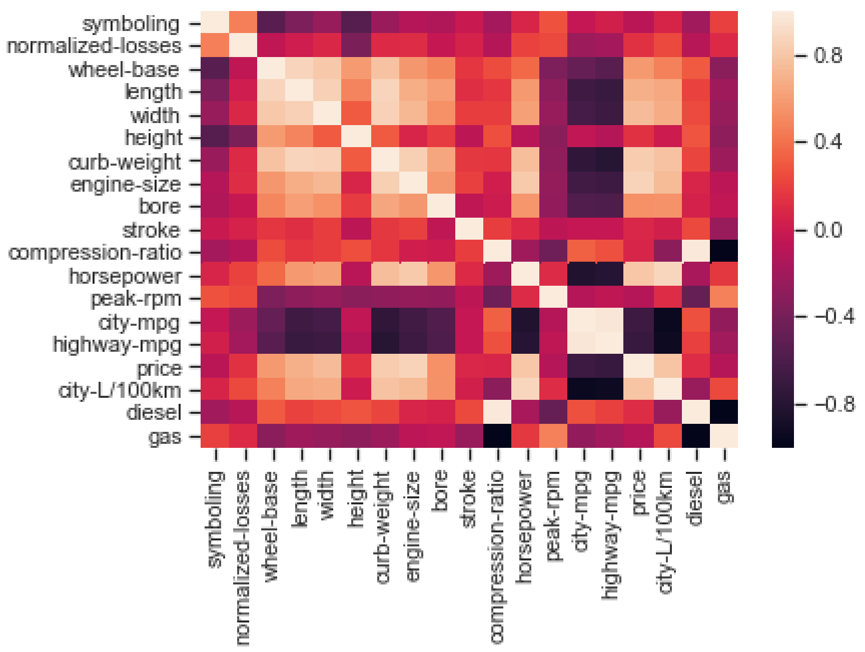
```
In [12]: data.corr()
```

Out[12]:

	symboling	normalized-losses	wheel-base	length	width	height	curb-weight	engine-size	
symboling	1.000000	0.466264	-0.535987	-0.365404	-0.242423	-0.550160	-0.233118	-0.110581	-
normalized-losses	0.466264	1.000000	-0.056661	0.019424	0.086802	-0.373737	0.099404	0.112360	-
wheel-base	-0.535987	-0.056661	1.000000	0.876024	0.814507	0.590742	0.782097	0.572027	
length	-0.365404	0.019424	0.876024	1.000000	0.857170	0.492063	0.880665	0.685025	
width	-0.242423	0.086802	0.814507	0.857170	1.000000	0.306002	0.866201	0.729436	
height	-0.550160	-0.373737	0.590742	0.492063	0.306002	1.000000	0.307581	0.074694	
curb-weight	-0.233118	0.099404	0.782097	0.880665	0.866201	0.307581	1.000000	0.849072	
engine-size	-0.110581	0.112360	0.572027	0.685025	0.729436	0.074694	0.849072	1.000000	
bore	-0.140019	-0.029862	0.493244	0.608971	0.544885	0.180449	0.644060	0.572609	
stroke	-0.008245	0.055563	0.158502	0.124139	0.188829	-0.062704	0.167562	0.209523	-
compression-ratio	-0.182196	-0.114713	0.250313	0.159733	0.189867	0.259737	0.156433	0.028889	
horsepower	0.075819	0.217299	0.371147	0.579821	0.615077	-0.087027	0.757976	0.822676	
peak-rpm	0.279740	0.239543	-0.360305	-0.285970	-0.245800	-0.309974	-0.279361	-0.256733	-
city-mpg	-0.035527	-0.225016	-0.470606	-0.665192	-0.633531	-0.049800	-0.749543	-0.650546	-
highway-mpg	0.036233	-0.181877	-0.543304	-0.698142	-0.680635	-0.104812	-0.794889	-0.679571	-
price	-0.082391	0.133999	0.584642	0.690628	0.751265	0.135486	0.834415	0.872335	
city-L/100km	0.066171	0.238567	0.476153	0.657373	0.673363	0.003811	0.785353	0.745059	
diesel	-0.196735	-0.101546	0.307237	0.211187	0.244356	0.281578	0.221046	0.070779	
gas	0.196735	0.101546	-0.307237	-0.211187	-0.244356	-0.281578	-0.221046	-0.070779	-

```
In [13]: sns.heatmap(data.corr())
```

```
Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x2b5e9f367b8>
```


symboling
normalized-losses
wheel-base
length
width
height
curb-weight
engine-size
bore
stroke
compression-ratio
horsepower
peak-rpm
city-mpg
highway-mpg
price
city-L/100km
diesel
gas