Выбранный датасет: BMW Pricing Challenge

Представленные данные состоят из почти 5000 реальных автомобилей BMW, которые были проданы на аукционе b2b в 2018 году. Цена, указанная в таблице, является самой высокой ставкой, которая была достигнута в ходе аукциона.

In [58]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline

import seaborn as sns
```

Загрузка датасета

In [59]:

```
df = pd.read_csv('bmw_pricing_challenge.csv')
```

In [60]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4843 entries, 0 to 4842
Data columns (total 18 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   maker_key         4843 non-null   object
 1   model_key         4843 non-null   object
 2   mileage           4843 non-null   int64 
 3   engine_power      4843 non-null   int64 
 4   registration_date 4843 non-null   object
 5   fuel              4843 non-null   object
 6   paint_color       4843 non-null   object
 7   car_type          4843 non-null   object
 8   feature_1         4843 non-null   bool  
 9   feature_2         4843 non-null   bool  
 10  feature_3         4843 non-null   bool  
 11  feature_4         4843 non-null   bool  
 12  feature_5         4843 non-null   bool  
 13  feature_6         4843 non-null   bool  
 14  feature_7         4843 non-null   bool  
 15  feature_8         4843 non-null   bool  
 16  price             4843 non-null   int64 
 17  sold_at           4843 non-null   object
dtypes: bool(8), int64(3), object(7)
memory usage: 416.3+ KB
```

In [61]:

```
df.describe()
```

Out[61]:

	mileage	engine_power	price
count	4.843000e+03	4843.00000	4843.000000
mean	1.409628e+05	128.98823	15828.081767
std	6.019674e+04	38.99336	9220.285684
min	-6.400000e+01	0.00000	100.000000
25%	1.029135e+05	100.00000	10800.000000
50%	1.410800e+05	120.00000	14200.000000
75%	1.751955e+05	135.00000	18600.000000
max	1.000376e+06	423.00000	178500.000000

In [62]:

```
df.head(5)
```

Out[62]:

	maker_key	model_key	mileage	engine_power	registration_date	fuel	paint_color	car_type	feature_1	feature_2	feature_3	feature_4	feature_
0	BMW	118	140411	100	2012-02-01	diesel	black	convertible	True	True	False	False	Tru
1	BMW	M4	13929	317	2016-04-01	petrol	grey	convertible	True	True	False	False	Fals
2	BMW	320	183297	120	2012-04-01	diesel	white	convertible	False	False	False	False	Tru
3	BMW	420	128035	135	2014-07-01	diesel	red	convertible	True	True	False	False	Tru
4	BMW	425	97097	160	2014-12-01	diesel	silver	convertible	True	True	False	False	Fals

Подготовка данных

In [63]:

```
df.isnull().head(10)
```

Out[63]:

	maker_key	model_key	mileage	engine_power	registration_date	fuel	paint_color	car_type	feature_1	feature_2	feature_3	feature_4	feature_5
0	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False
5	False	False	False	False	False	False	False	False	False	False	False	False	False
6	False	False	False	False	False	False	False	False	False	False	False	False	False
7	False	False	False	False	False	False	False	False	False	False	False	False	False
8	False	False	False	False	False	False	False	False	False	False	False	False	False
9	False	False	False	False	False	False	False	False	False	False	False	False	False

In [64]:

```
df.isnull().sum()
```

Out[64]:

```
maker_key           0
model_key           0
mileage             0
engine_power        0
registration_date    0
fuel                0
paint_color         0
car_type            0
feature_1           0
feature_2           0
feature_3           0
feature_4           0
feature_5           0
feature_6           0
feature_7           0
feature_8           0
price               0
sold_at             0
dtype: int64
```

- 0 пропусков в колонках, значит вставка средних значений вместо значений "NaN" не требуется

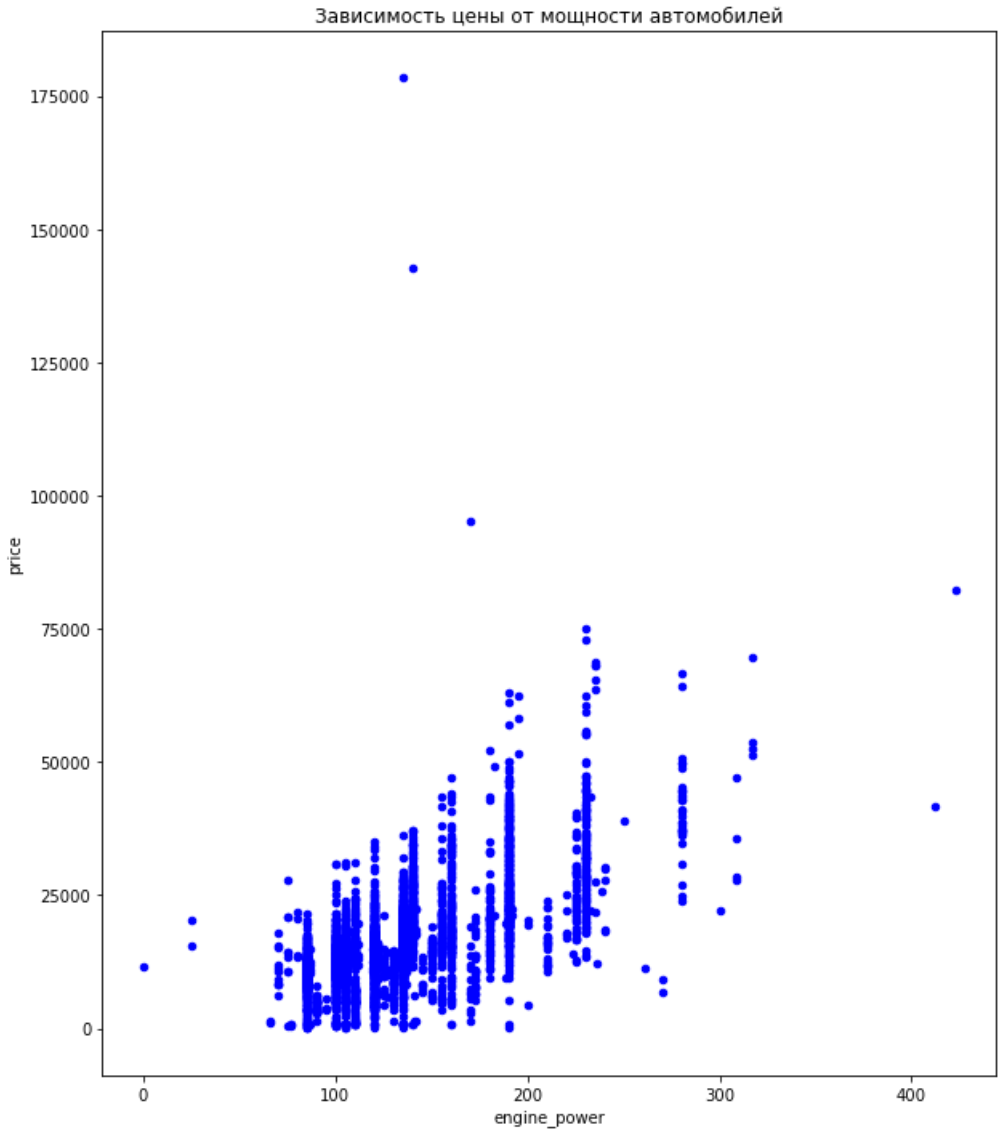
Описательная статистика + гипотезы

Гипотеза №1

Логично предположить, что мощность автомобиля влияет на его стоимость, ведь за лошадиные силы надо платить.

In [73]:

```
a_engine = np.average(df.engine_power)
df.plot.scatter(x='engine_power', y='price', c='blue', figsize = [10, 12], title='Зависимость цены от мощности автомобилей');
```



Данный график подтверждает гипотезу, по которой можно сделать такие выводы:

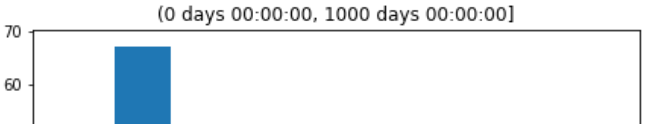
- Из концентрирования большей части значений прослеживается повышение цены автомобилей с повышением их мощности
- Единичные автомобили, проданные намного дороже средней цены на аукционе, имели мощность выше среднего

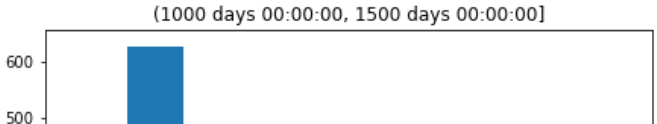
Гипотеза №2

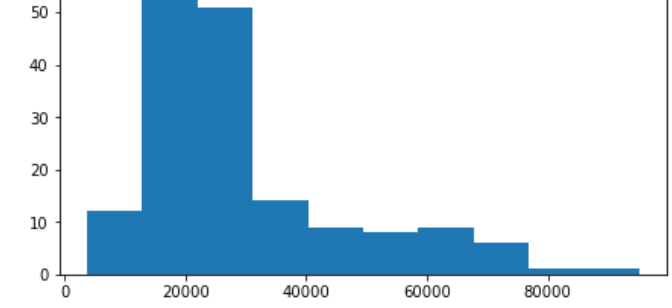
Предположим, что с увеличением возраста автомобиля он теряет свою стоимость. Для этого для каждого автомобиля вычислим его количество дней службы и построим соответсвующие гистограммы, сгруппировав полученные данные.

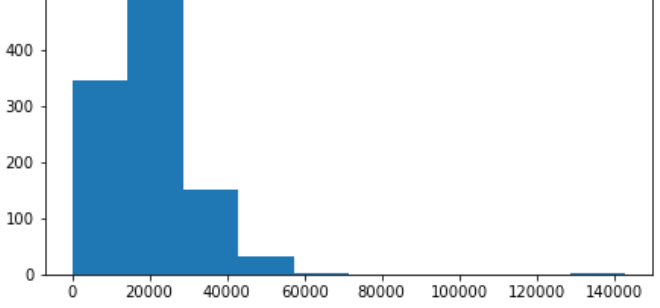
In [66]:

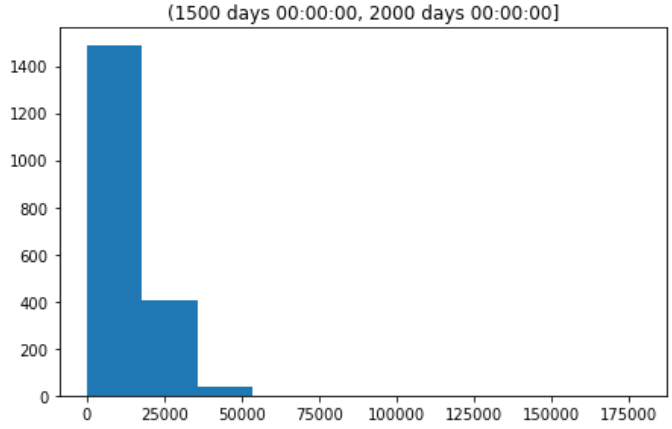
```
df['dates'] = pd.to_datetime(df['sold_at']) - pd.to_datetime(df['registration_date'])
bins = [pd.to_timedelta('0 days'), pd.to_timedelta('1000 days'), pd.to_timedelta('1500 days'), pd.to_timedelta('2000 days'), pd.to_timedelta('3000 days'), pd.to_timedelta('4000 days'), pd.to_timedelta('11000 days')]
df['dates_groups']=pd.cut(df.dates, bins)
df.hist('price', by='dates_groups', layout=[3,2], figsize = [16, 18], xrot=0);
```

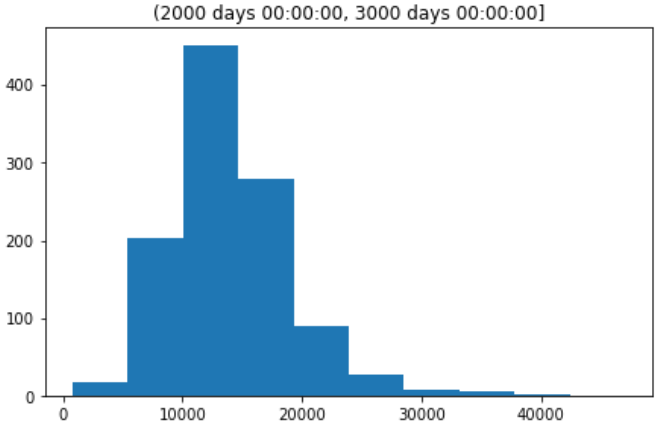


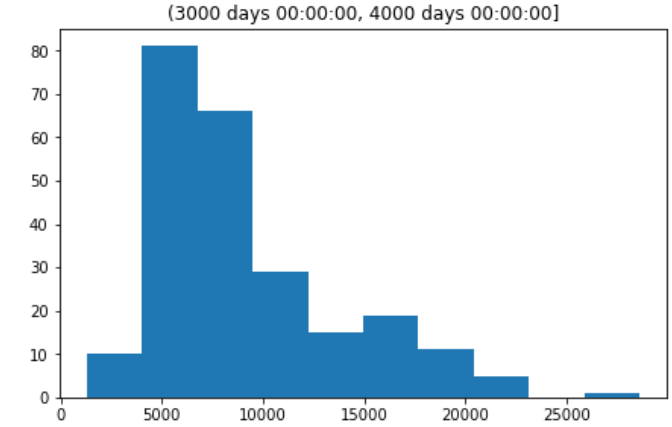


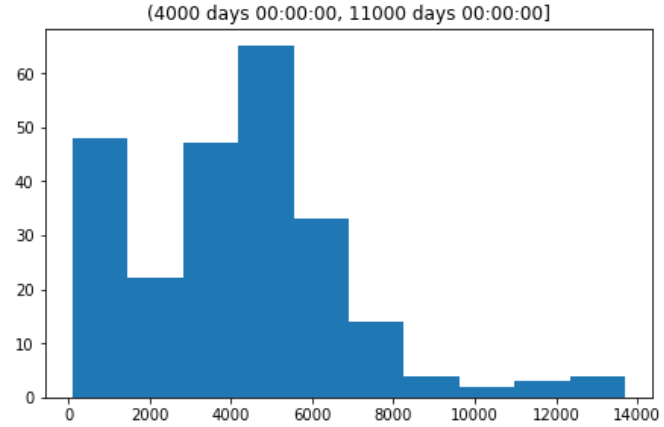












Вторая гипотеза также подтвердилась. Вот какие выводы можно сделать из полученных гистограмм:

- С увеличением возраста автомобиля он падает в цене
- Подавляющее большинство проданных на аукцион автомобилей были не старше 10 лет

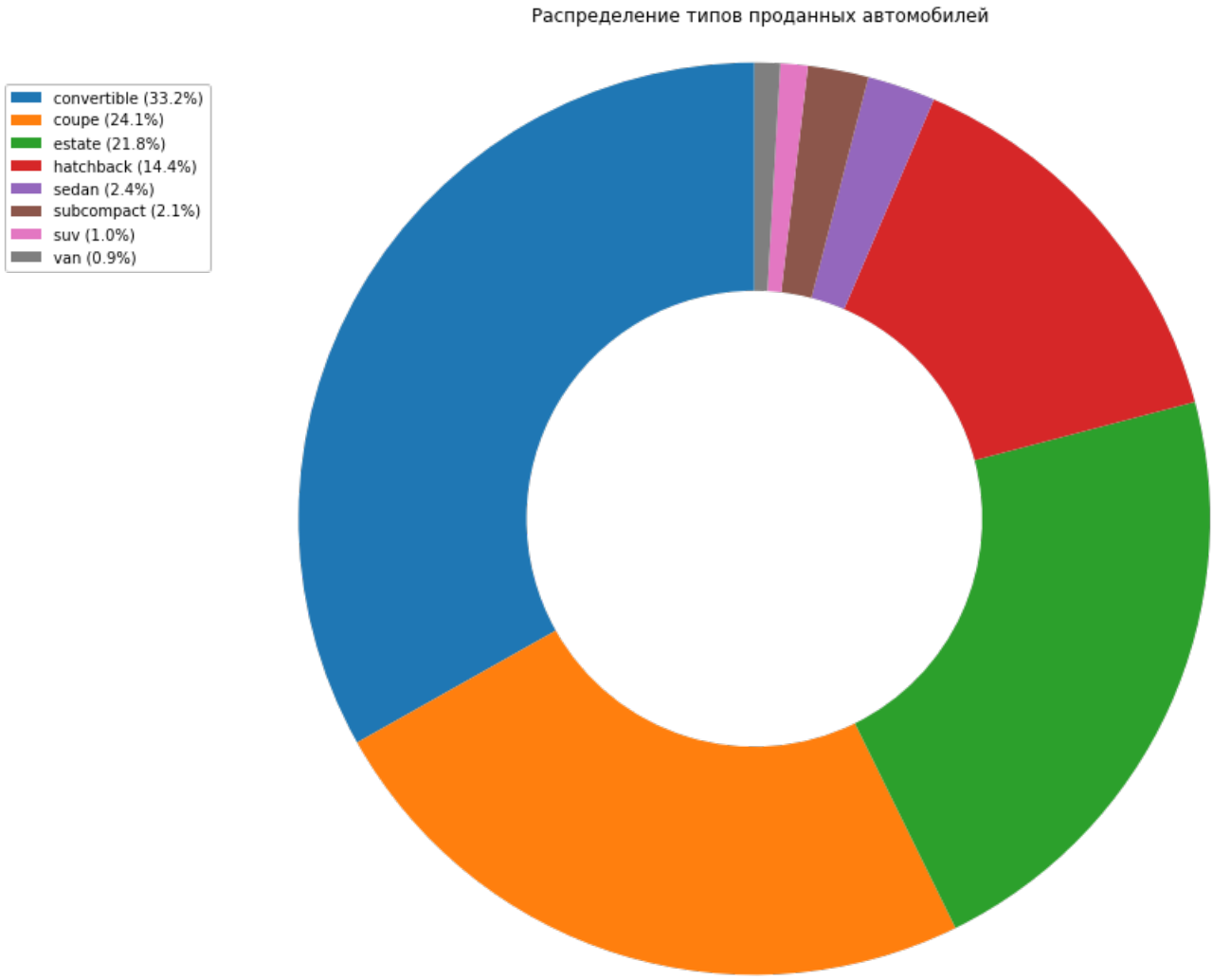
Гипотеза №3

Предположим, что на аукционе больше всего было продано седанов и купе, так как это одни из самых распространенных типов автомобилей марки BMW. Для этого построим круговую диаграмму.

In [67]:

```
plt.rcParams['figure.figsize'] = [12, 12]
labels = list(df['car_type'].unique())
values = list(dict(df['car_type'].value_counts()).values())
total = sum(values)
labels = [f"{n} ({v/total:.1%})" for n, v in zip(labels, values)]

fig1, ax1 = plt.subplots()
ax1.pie(values, shadow=False, startangle=90, wedgeprops=dict(width=0.5))
ax1.axis('equal')
plt.legend(
    bbox_to_anchor = (-0.25, 0.75, 0.25, 0.25),
    loc = 'lower left', labels = labels)
plt.title('Распределение типов проданных автомобилей')
plt.show()
```



Из диаграммы видим, что гипотеза подтвердилась чатично:

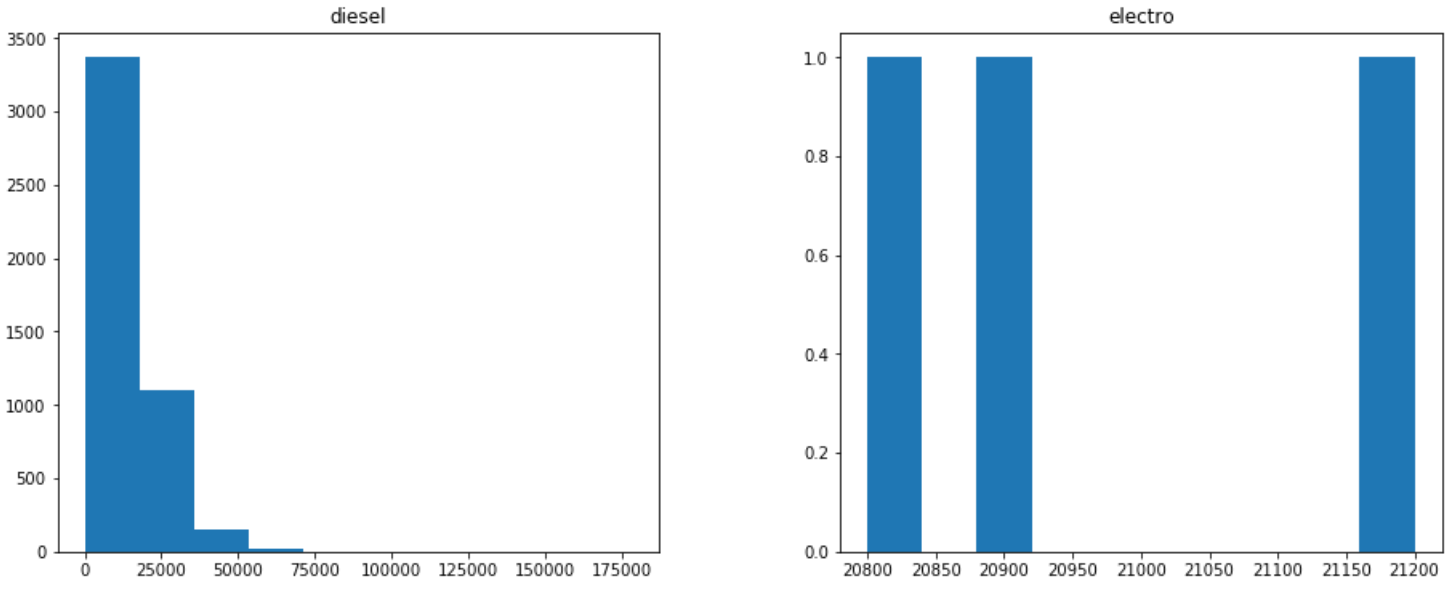
- Почти четверть из всех проданных автомобилей относилась к купе, а седанов, напротив, продано чуть больше двух процентов.
- Самый распространенный тип автомобиля на данном аукционе - кабриолет. Треть продаж из всех проданных автомобилей разных типов кузова.

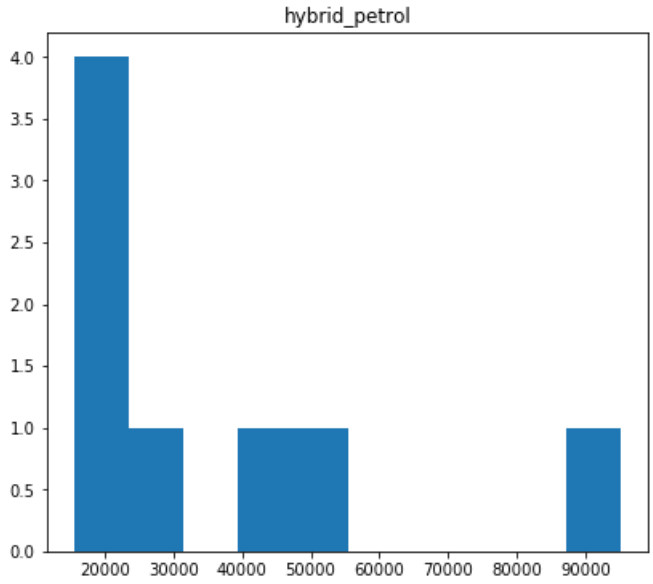
Гипотеза №4

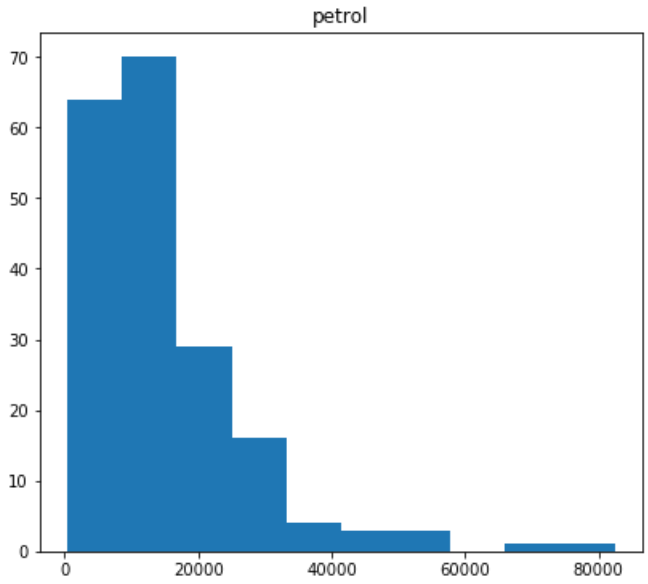
Разумно предположить, что автомобилей на электричестве и на гибридном топливе было продано меньше всего, так как это наименее распространенные автомобили. Проверим это с помощью столбчатых гистограмм.

In [68]:

```
df.hist('price', by='fuel', xrot=0, layout=[2,2], figsize = [15, 15]);
```







Гипотеза подтвердилась, на аукционе и правда были проданы единицы гибридных автомобилей и электрокаров. Кроме того, можем сделать такие выводы:

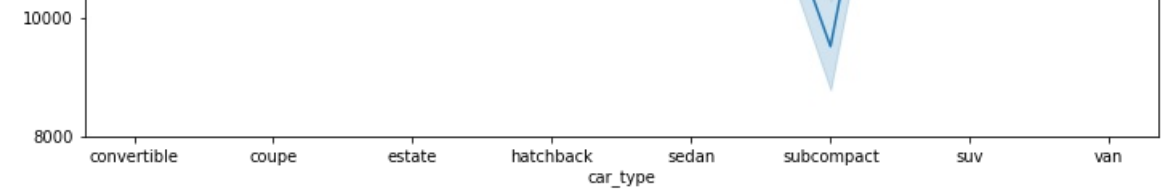
- У большей части проданных автомобилей были дизельные двигатели
- Автомобили с электрическим и гибридным типом топлива встречаются (и продаются) реже всего
- Нельзя проследить зависимость цены от типа топлива, так как дизельных автомобилей намного больше, чем машин с другими типами топлива

Гипотеза №5

Предположим, что наиболее дорогими автомобилями являются купе и кабриолеты. Построим график для проверки этой гипотезы.

In [69]:

```
sns.lineplot(x=df.car_type, y=df.price);
```



Из графика видно лишь частичное подтверждение гипотезы:

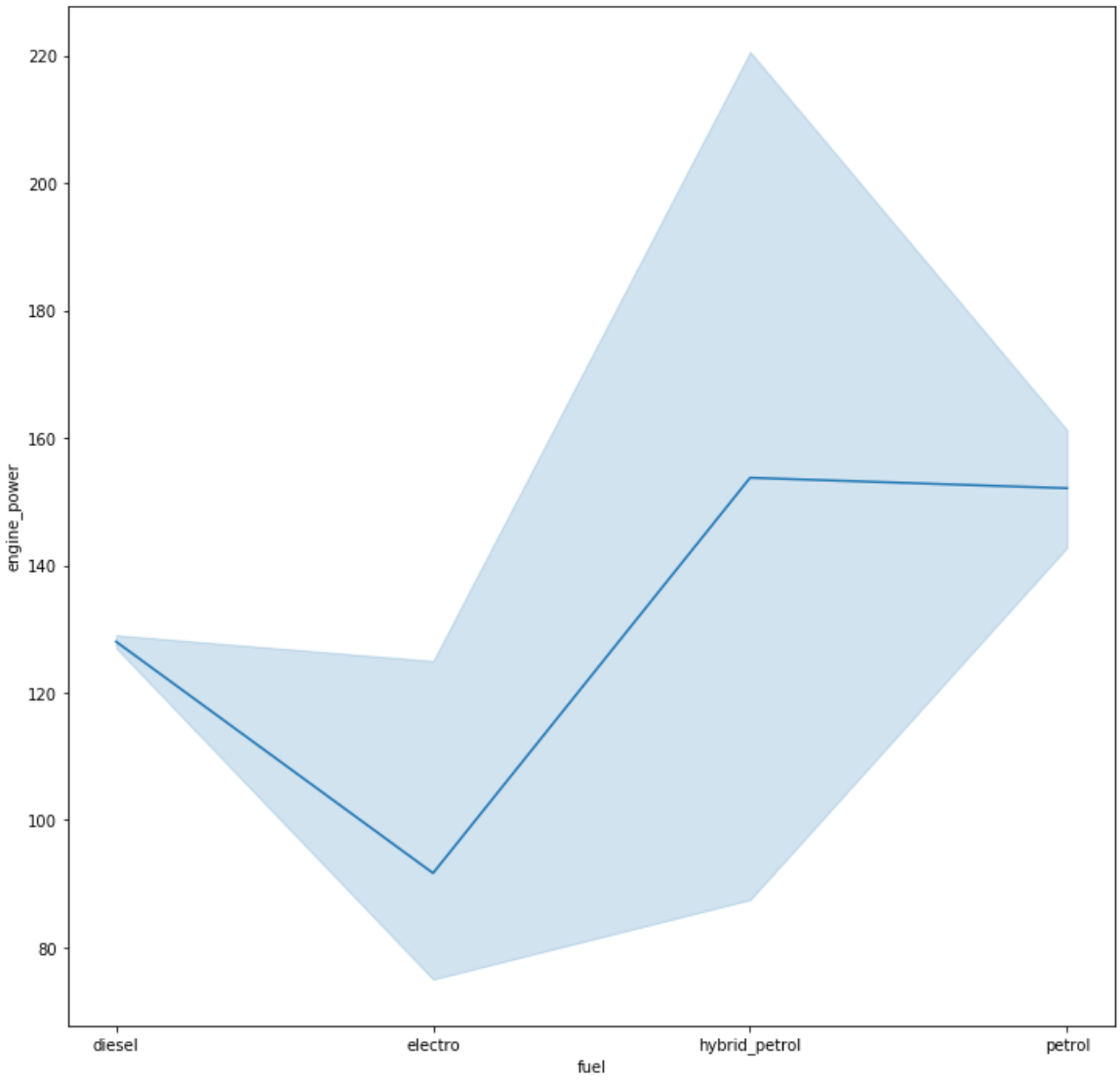
- Наиболее дорогими типами автомобилей являются купе и внедорожники. Кабриолеты стоят немного дешевле, занимая третье место по цене.

Статистические данные

Узнаем, на каком топливе ездят автомобили, имеющие наибольшую мощность двигателя.

In [70]:

```
sns.lineplot(x=df.fuel, y=df.engine_power);
```



Из построенного графика можно увидеть, что наиболее мощными двигателями обладают автомобили с гибридными и бензиновыми типами топлива.

In [ ]:

```

```