Carros Usados

Importando bibliotecas

In [106]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib import pyplot
```

Carregando os dados

In [107]:

```
dados = pd.read_csv('USA_cars_datasets.csv')
dados.head()
```

Out[107]:

	Unnamed: 0	price	brand	model	year	title_status	mileage	color	vin	lot	state	country	condition
0	0	6300	toyota	cruiser	2008	clean vehicle	274117.0	black	jtezu11f88k007763	159348797	new jersey	usa	10 days left
1	1	2899	ford	se	2011	clean vehicle	190552.0	silver	2fmdk3gc4bbb02217	166951262	tennessee	usa	6 days left
2	2	5350	dodge	mpv	2018	clean vehicle	39590.0	silver	3c4pdcgg5jt346413	167655728	georgia	usa	2 days left
3	3	25000	ford	door	2014	clean vehicle	64146.0	blue	1ftfw1et4efc23745	167753855	virginia	usa	22 hours left
4	4	27700	chevrolet	1500	2018	clean vehicle	6654.0	red	3gcpcrec2jg473991	167763266	florida	usa	22 hours left

Trabalhando os dados

Excluindo colunas

In [108]:

```
#Tirarndo colunas que não necessarias
dados = dados.drop(columns=['Unnamed: 0', 'vin', 'lot', 'condition', 'country'])
dados.head()
```

Out[108]:

	price	brand	model	year	title_status	mileage	color	state
0	6300	toyota	cruiser	2008	clean vehicle	274117.0	black	new jersey
1	2899	ford	se	2011	clean vehicle	190552.0	silver	tennessee
2	5350	dodge	mpv	2018	clean vehicle	39590.0	silver	georgia
3	25000	ford	door	2014	clean vehicle	64146.0	blue	virginia
4	27700	chevrolet	1500	2018	clean vehicle	6654.0	red	florida

Verificando desconformidades

In [109]:

```
dados.describe()
```

Out[109]:

	price	year	mileage
count	2499.000000	2499.000000	2.499000e+03
mean	18767.671469	2016.714286	5.229869e+04
std	12116.094936	3.442656	5.970552e+04
min	0.000000	1973.000000	0.000000e+00
25%	10200.000000	2016.000000	2.146650e+04
50%	16900.000000	2018.000000	3.536500e+04
75%	25555.500000	2019.000000	6.347250e+04
max	84900.000000	2020.000000	1.017936e+06

In [110]:

```
dados.isnull().sum()
#dados.info()
```

Out[110]:

```
price         0
brand         0
model         0
year          0
title_status  0
mileage       0
color         0
state         0
dtype: int64
```

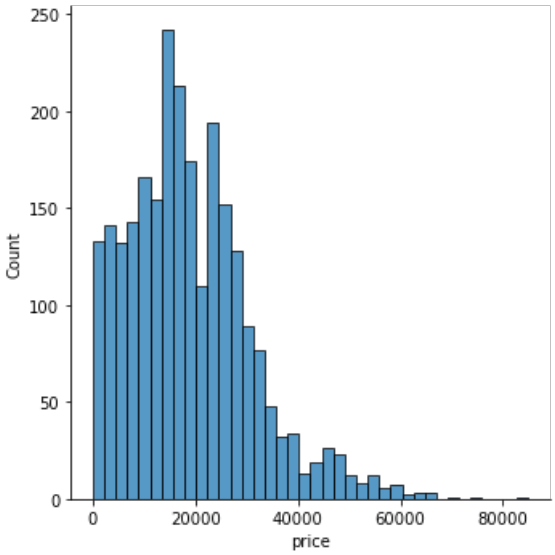
Visualizando graficamente

In [111]:

```
sns.displot(dados['price'])
```

Out[111]:

```
<seaborn.axisgrid.FacetGrid at 0x184207d0310>
```

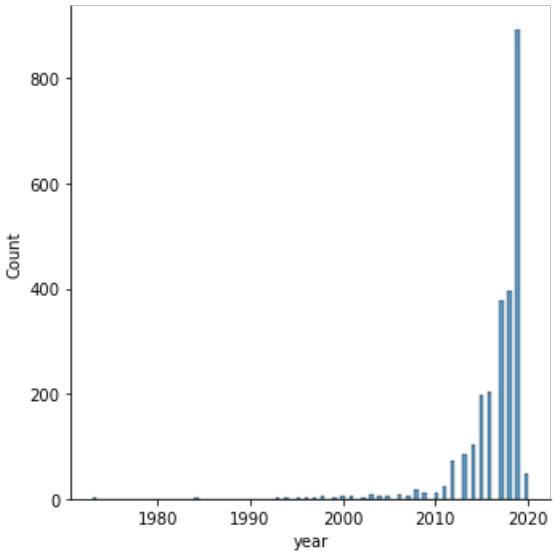


In [112]:

```
sns.displot(dados['year'])
```

Out[112]:

<seaborn.axisgrid.FacetGrid at 0x184206ae100>

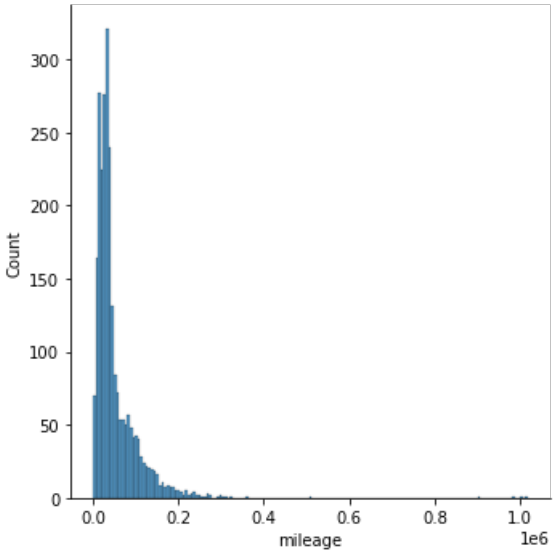


In [113]:

```
sns.displot(dados['mileage'])
```

Out[113]:

<seaborn.axisgrid.FacetGrid at 0x1841f0c0df0>



Tirando dados discrepantes

Excluindo valores de preços iguais a zero

In [114]:

```
q = dados['price'].quantile(0.01)
dados_ref = dados[dados['price']>q]
dados_ref.describe()
```

Out[114]:

	price	year	mileage
count	2456.000000	2456.000000	2.456000e+03
mean	19096.258550	2016.931189	5.011143e+04
std	11962.176006	2.957497	5.460446e+04
min	25.000000	1973.000000	0.000000e+00
25%	10500.000000	2016.000000	2.127675e+04
50%	17050.000000	2018.000000	3.504850e+04
75%	25800.000000	2019.000000	6.005075e+04
max	84900.000000	2020.000000	1.017936e+06

Excluindo milhas iguais a 0 e valores altos muito discrepantes

In [115]:

```
q2 = dados_ref['mileage'].quantile(0.99)
dados_ref2 = dados_ref[dados_ref['mileage']<q2]
dados_ref2.describe()
```

Out[115]:

	price	year	mileage
count	2431.000000	2431.000000	2431.000000
mean	19258.171946	2017.030440	46957.299877
std	11909.820415	2.759327	38217.983286
min	25.000000	1973.000000	0.000000
25%	10800.000000	2016.000000	21125.500000
50%	17200.000000	2018.000000	34837.000000
75%	25900.000000	2019.000000	58711.500000
max	84900.000000	2020.000000	217290.000000

In [116]:

```
q3 = dados_ref2['mileage'].quantile(0.01)
dados_ref3 = dados_ref2[dados_ref2['mileage']>q3]
dados_ref3.describe()
```

Out[116]:

	price	year	mileage
count	2406.000000	2406.000000	2406.000000
mean	19246.721114	2017.029925	47443.313799
std	11803.660284	2.755138	38115.849785
min	25.000000	1973.000000	1091.000000
25%	10900.000000	2016.000000	21677.500000
50%	17200.000000	2018.000000	35048.500000
75%	25868.750000	2019.000000	59478.750000
max	84900.000000	2020.000000	217290.000000

Excluindo datas antigas discrepantes

In [117]:

```
q4 = dados_ref3['year'].quantile(0.01)
dados_ref4 = dados_ref3[dados_ref3['year']>q4]
dados_ref4.describe()
```

Out[117]:

	price	year	mileage
count	2372.000000	2372.000000	2372.000000
mean	19462.780776	2017.219646	46262.244941
std	11716.634996	2.102454	36683.687010
min	25.000000	2009.000000	1091.000000
25%	11000.000000	2016.000000	21336.750000
50%	17500.000000	2018.000000	34799.000000
75%	25923.750000	2019.000000	57605.750000
max	84900.000000	2020.000000	217290.000000

Verificando marcas e cores mais populares

In [118]:

```
dados['color'].value_counts().head()
```

Out[118]:

```
white     707
black     516
gray      395
silver    300
red       192
Name: color, dtype: int64
```

In [119]:

```
dados['brand'].value_counts().head()
```

Out[119]:

```
ford         1235
dodge         432
nissan        312
chevrolet     297
gmc            42
Name: brand, dtype: int64
```

Tratamento de dados finalizado

In [120]:

```
dados_limpos = dados_ref4
```

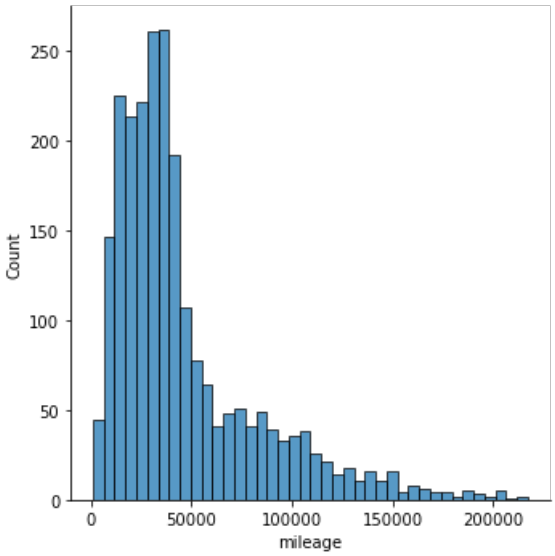
Graficos ajustados

In [121]:

```
sns.displot(dados_limpos['mileage'])
```

Out[121]:

<seaborn.axisgrid.FacetGrid at 0x18420c1f880>

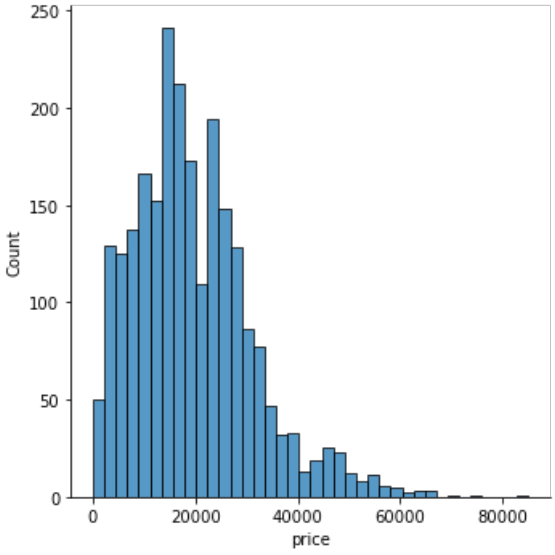


In [122]:

```
sns.displot(dados_limpos['price'])
```

Out[122]:

<seaborn.axisgrid.FacetGrid at 0x18420c55f10>

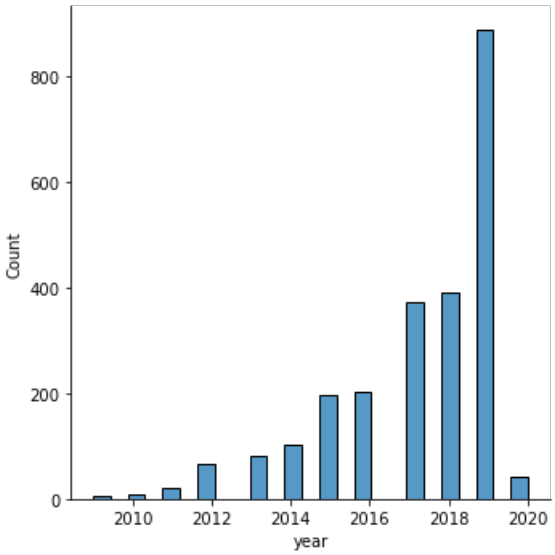


In [123]:

```
sns.displot(dados_limpos['year'])
```

Out[123]:

<seaborn.axisgrid.FacetGrid at 0x18420bf7130>



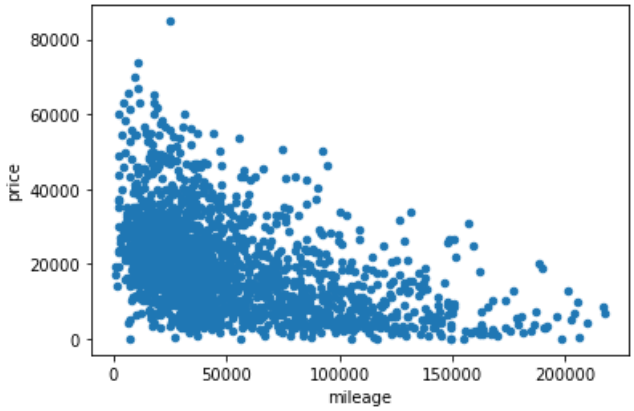
Relação preço por milhas

In [124]:

```
dados_limpos.plot.scatter(x="mileage", y="price")
```

Out[124]:

<AxesSubplot:xlabel='mileage', ylabel='price'>



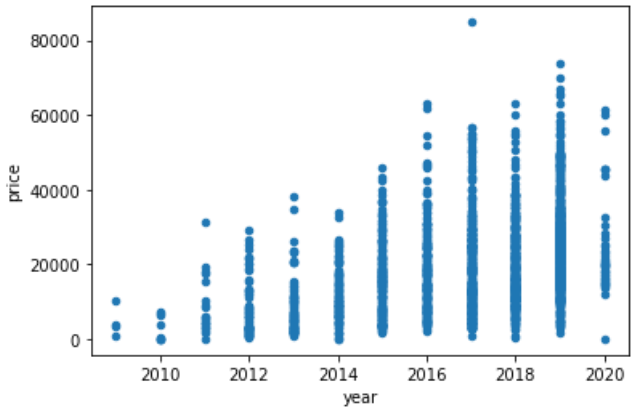
Relação preço por ano

In [125]:

```
dados_limpos.plot.scatter(x="year", y="price")
```

Out[125]:

<AxesSubplot:xlabel='year', ylabel='price'>



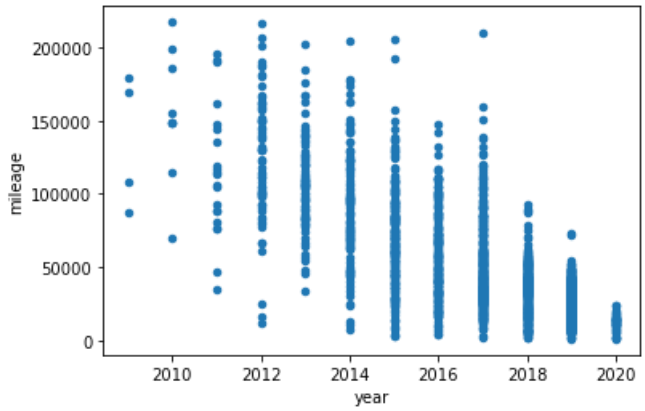
Relação milhas por ano

In [126]:

```
dados_limpos.plot.scatter(x="year", y="mileage")
```

Out[126]:

<AxesSubplot:xlabel='year', ylabel='mileage'>



Relação preço médio por marca

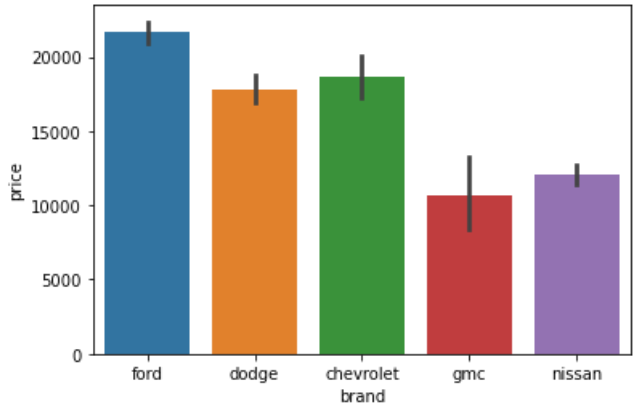
In [130]:

```
dados_marca = dados[
    (dados['brand'] == 'ford') |
    (dados['brand'] == 'dodge') |
    (dados['brand'] == 'nissan') |
    (dados['brand'] == 'gmc') |
    (dados['brand'] == 'chevrolet')
]

sns.barplot(x=dados_marca['brand'], y=dados_marca['price'])
```

Out[130]:

<AxesSubplot:xlabel='brand', ylabel='price'>



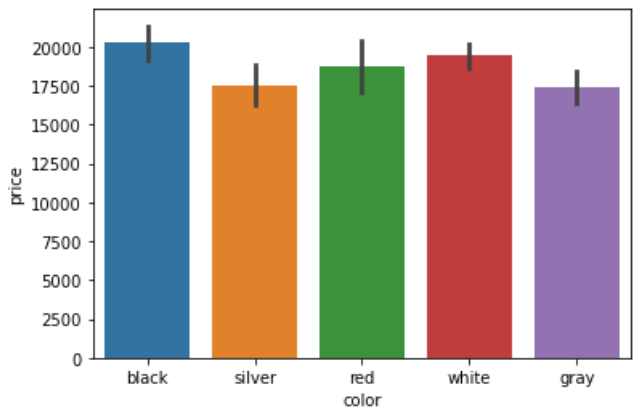
Relação preço médio por cor

In [131]:

```
dados_cor = dados[
    (dados['color'] == 'white') |
    (dados['color'] == 'black') |
    (dados['color'] == 'gray') |
    (dados['color'] == 'red') |
    (dados['color'] == 'silver')
]
sns.barplot(x=dados_cor['color'], y=dados_cor['price'])
```

Out[131]:

<AxesSubplot:xlabel='color', ylabel='price'>



In [ ]: