carpricepred-3

September 28, 2023

[1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

[2]:
```
df=pd.read_csv(r"C:\Users\amits\Downloads\quikr_car.csv")
```

[3]:
```
df
```

[3]:
```
                                     name    company  year          Price  \
0      Hyundai Santro Xing XO eRLX Euro III    Hyundai  2007         80,000
1                   Mahindra Jeep CL550 MDI   Mahindra  2006       4,25,000
2                Maruti Suzuki Alto 800 Vxi     Maruti  2018  Ask For Price
3    Hyundai Grand i10 Magna 1.2 Kappa VTVT    Hyundai  2014       3,25,000
4          Ford EcoSport Titanium 1.5L TDCi       Ford  2014       5,75,000
..                                       …        …   …             …
887                                     Ta       Tara  zest       3,10,000
888                    Tata Zest XM Diesel       Tata  2018       2,60,000
889                     Mahindra Quanto C8   Mahindra  2013       3,90,000
890                Honda Amaze 1.2 E i VTEC      Honda  2014       1,80,000
891               Chevrolet Sail 1.2 LT ABS  Chevrolet  2014       1,60,000

     kms_driven fuel_type
0    45,000 kms    Petrol
1        40 kms    Diesel
2    22,000 kms    Petrol
3    28,000 kms    Petrol
4    36,000 kms    Diesel
..          …       …
887         NaN       NaN
888  27,000 kms    Diesel
889  40,000 kms    Diesel
890      Petrol       NaN
891      Petrol       NaN

[892 rows x 6 columns]
```

[4]:
```
df.isnull().sum()
```

[4]:
```
name           0
company        0
year           0
Price          0
kms_driven    52
fuel_type     55
dtype: int64
```

[5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 892 entries, 0 to 891
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   name         892 non-null    object
 1   company      892 non-null    object
 2   year         892 non-null    object
 3   Price        892 non-null    object
 4   kms_driven   840 non-null    object
 5   fuel_type    837 non-null    object
dtypes: object(6)
memory usage: 41.9+ KB
```

[6]:
```
df['kms_driven'].value_counts()
```

[6]:
```
45,000 kms      30
35,000 kms      30
55,000 kms      25
50,000 kms      23
20,000 kms      22
                ..
1,03,553 kms     1
29,685 kms       1
30,874 kms       1
49,800 kms       1
1,66,000 kms     1
Name: kms_driven, Length: 258, dtype: int64
```

[7]:
```
df['total_kms']=df['kms_driven'].str.split(' ').str[0]
```

[8]:
```
df['total_kms']=df['total_kms'].replace(',','', regex=True)
```

[9]:
```
df.drop('kms_driven', inplace=True, axis=1)
```

[10]:
```
df['total_kms']=pd.to_numeric(df['total_kms'], errors='coerce')
```

[11]:
```
df.isnull().sum()
```

[11]:
```
name          0
company       0
year          0
Price         0
fuel_type    55
total_kms    54
dtype: int64
```

[12]:
```
df.head()
```

[12]:

	name	company	year	Price	fuel_type	total_kms
0	Hyundai Santro Xing XO eRLX Euro III	Hyundai	2007	80,000	Petrol	45000.0
1	Mahindra Jeep CL550 MDI	Mahindra	2006	4,25,000	Diesel	40.0
2	Maruti Suzuki Alto 800 Vxi	Maruti	2018	Ask For Price	Petrol	22000.0
3	Hyundai Grand i10 Magna 1.2 Kappa VTVT	Hyundai	2014	3,25,000	Petrol	28000.0
4	Ford EcoSport Titanium 1.5L TDCi	Ford	2014	5,75,000	Diesel	36000.0

[13]:
```
df['Price']=df['Price'].replace(',','', regex=True)
```

[14]:
```
df['Price']=pd.to_numeric(df['Price'], errors='coerce')
```

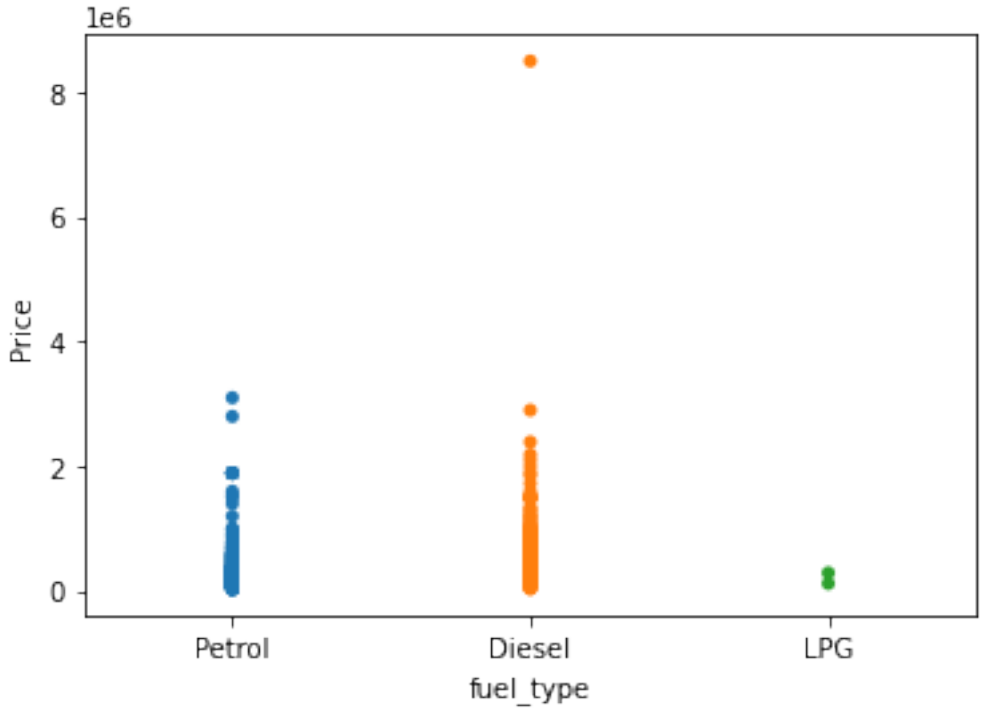
[15]:
```
df['year'].value_counts().head(20)
```

[15]:
```
2015    117
2014     94
2013     94
2016     76
2012     75
2011     60
2017     56
2009     56
2010     44
2018     34
2019     22
2006     22
2007     19
```

```
2008     16
2003     13
2005     13
2004     12
2000      7
2002      5
2001      5
Name: year, dtype: int64
```

[16]:
```
sns.stripplot(x='fuel_type', y='Price', data=df, jitter=False )
```

[16]:
```
<Axes: xlabel='fuel_type', ylabel='Price'>
```

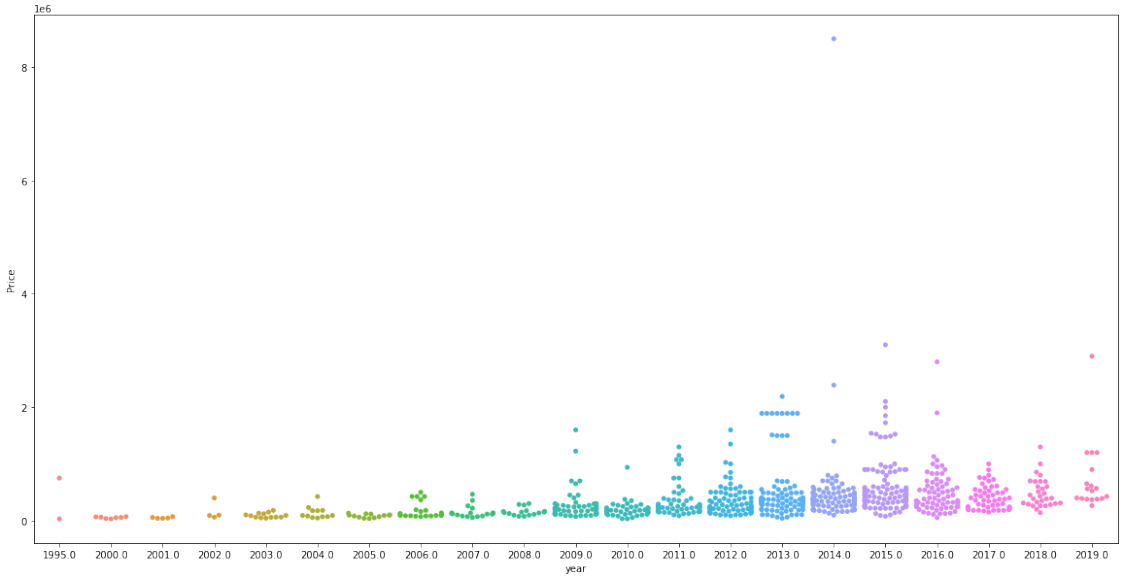


[17]:
```
df['year']=pd.to_numeric(df['year'], errors='coerce')
```

[18]:
```
plt.subplots(figsize=(20,10))
ax=sns.swarmplot(x='year',y='Price',data=df)
```

```
C:\Users\amits\anaconda3\lib\site-packages\seaborn\categorical.py:1296:
UserWarning: 15.4% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\amits\anaconda3\lib\site-packages\seaborn\categorical.py:1296:
UserWarning: 22.7% of the points cannot be placed; you may want to decrease the
```

```
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\amits\anaconda3\lib\site-packages\seaborn\categorical.py:1296:
UserWarning: 26.3% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\amits\anaconda3\lib\site-packages\seaborn\categorical.py:1296:
UserWarning: 12.5% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\amits\anaconda3\lib\site-packages\seaborn\categorical.py:1296:
UserWarning: 39.3% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\amits\anaconda3\lib\site-packages\seaborn\categorical.py:1296:
UserWarning: 34.1% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\amits\anaconda3\lib\site-packages\seaborn\categorical.py:1296:
UserWarning: 36.7% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\amits\anaconda3\lib\site-packages\seaborn\categorical.py:1296:
UserWarning: 28.0% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\amits\anaconda3\lib\site-packages\seaborn\categorical.py:1296:
UserWarning: 29.8% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\amits\anaconda3\lib\site-packages\seaborn\categorical.py:1296:
UserWarning: 39.4% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\amits\anaconda3\lib\site-packages\seaborn\categorical.py:1296:
UserWarning: 35.9% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\amits\anaconda3\lib\site-packages\seaborn\categorical.py:1296:
UserWarning: 14.5% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\amits\anaconda3\lib\site-packages\seaborn\categorical.py:1296:
UserWarning: 16.1% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```



[19]:
```
df.head()
```

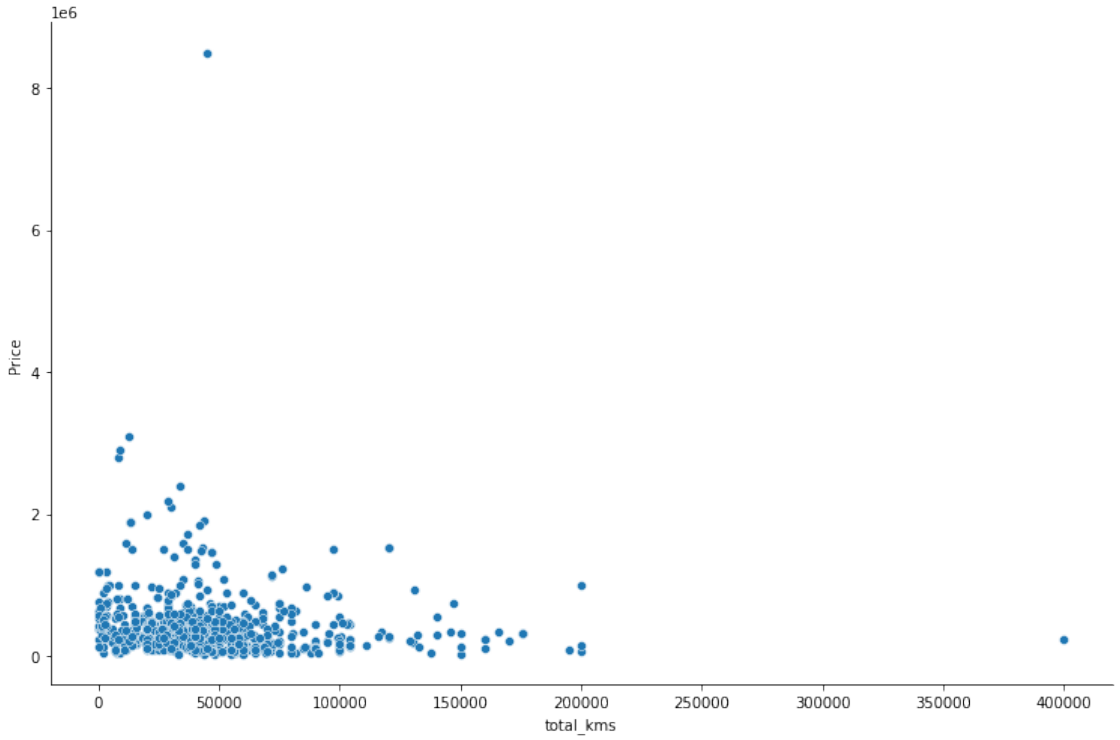
[19]:
```
                                  name    company    year     Price  \
0    Hyundai Santro Xing XO eRLX Euro III   Hyundai  2007.0   80000.0
1                 Mahindra Jeep CL550 MDI  Mahindra  2006.0  425000.0
2             Maruti Suzuki Alto 800 Vxi    Maruti  2018.0       NaN
3  Hyundai Grand i10 Magna 1.2 Kappa VTVT   Hyundai  2014.0  325000.0
4        Ford EcoSport Titanium 1.5L TDCi      Ford  2014.0  575000.0

  fuel_type  total_kms
0    Petrol    45000.0
1    Diesel       40.0
2    Petrol    22000.0
3    Petrol    28000.0
4    Diesel    36000.0
```

[20]:
```
sns.relplot(x='total_kms',y='Price',data=df,height=7,aspect=1.5, kind="scatter")
```

```
C:\Users\amits\anaconda3\lib\site-packages\seaborn\axisgrid.py:88: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

[20]:
```
<seaborn.axisgrid.FacetGrid at 0x205883f0fd0>
```



[21]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 892 entries, 0 to 891
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   name        892 non-null    object 
 1   company     892 non-null    object 
 2   year        842 non-null    float64
 3   Price       857 non-null    float64
 4   fuel_type   837 non-null    object 
 5   total_kms   838 non-null    float64
dtypes: float64(3), object(3)
memory usage: 41.9+ KB
```

[22]:
```
df['fuel_type'].unique()
```

[22]:
```
array(['Petrol', 'Diesel', nan, 'LPG'], dtype=object)
```

[23]:
```
df['fuel_type'].isnull().sum()
```

[23]:
```
55
```

[24]:
```
# df.drop('name', inplace=True, axis=1)
df['name']=df['name'].str.split().str.slice(start=0,stop=3).str.join(' ')
```

[25]:
```
df.head()
```

[25]:

	name	company	year	Price	fuel_type	total_kms
0	Hyundai Santro Xing	Hyundai	2007.0	80000.0	Petrol	45000.0
1	Mahindra Jeep CL550	Mahindra	2006.0	425000.0	Diesel	40.0
2	Maruti Suzuki Alto	Maruti	2018.0	NaN	Petrol	22000.0
3	Hyundai Grand i10	Hyundai	2014.0	325000.0	Petrol	28000.0
4	Ford EcoSport Titanium	Ford	2014.0	575000.0	Diesel	36000.0

[26]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 892 entries, 0 to 891
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   name        892 non-null    object 
 1   company     892 non-null    object 
 2   year        842 non-null    float64
 3   Price       857 non-null    float64
 4   fuel_type   837 non-null    object 
 5   total_kms   838 non-null    float64
dtypes: float64(3), object(3)
memory usage: 41.9+ KB
```

[27]:
```
df.head(20)
```

[27]:

	name	company	year	Price	fuel_type	total_kms
0	Hyundai Santro Xing	Hyundai	2007.0	80000.0	Petrol	45000.0
1	Mahindra Jeep CL550	Mahindra	2006.0	425000.0	Diesel	40.0
2	Maruti Suzuki Alto	Maruti	2018.0	NaN	Petrol	22000.0
3	Hyundai Grand i10	Hyundai	2014.0	325000.0	Petrol	28000.0
4	Ford EcoSport Titanium	Ford	2014.0	575000.0	Diesel	36000.0
5	Ford EcoSport Titanium	Ford	2015.0	NaN	Diesel	59000.0
6	Ford Figo	Ford	2012.0	175000.0	Diesel	41000.0
7	Hyundai Eon	Hyundai	2013.0	190000.0	Petrol	25000.0
8	Ford EcoSport Ambiente	Ford	2016.0	830000.0	Diesel	24530.0
9	Maruti Suzuki Alto	Maruti	2015.0	250000.0	Petrol	60000.0
10	Skoda Fabia Classic	Skoda	2010.0	182000.0	Petrol	60000.0
11	Maruti Suzuki Stingray	Maruti	2015.0	315000.0	Petrol	30000.0
12	Hyundai Elite i20	Hyundai	2014.0	415000.0	Petrol	32000.0
13	Mahindra Scorpio SLE	Mahindra	2015.0	320000.0	Diesel	48660.0
14	Hyundai Santro Xing	Hyundai	2007.0	80000.0	Petrol	45000.0
15	Mahindra Jeep CL550	Mahindra	2006.0	425000.0	Diesel	40.0

```
16                Audi A8      Audi  2017.0  1000000.0    Petrol     4000.0
17                Audi Q7      Audi  2014.0   500000.0    Diesel    16934.0
18    Mahindra Scorpio S10  Mahindra  2016.0   350000.0    Diesel    43000.0
19      Maruti Suzuki Alto    Maruti  2014.0   160000.0    Petrol    35550.0
```

[28]:
```
df['company'].replace('tara','tata',inplace=True)
```

[29]:
```
df.drop(labels='company',axis=1,inplace=True)
```

[30]:
```
df=pd.get_dummies(data=df, columns=['fuel_type'],␣
↪prefix=['fuel_type'],drop_first=True)
```

[31]:
```
df=pd.get_dummies(data=df, columns=['name'], prefix=['name'],drop_first=True)
```

[32]:
```
df
```

[32]:
```
        year     Price  total_kms  fuel_type_LPG  fuel_type_Petrol  \
0     2007.0   80000.0    45000.0              0                 1
1     2006.0  425000.0       40.0              0                 0
2     2018.0       NaN    22000.0              0                 1
3     2014.0  325000.0    28000.0              0                 1
4     2014.0  575000.0    36000.0              0                 0
..       ...       ...        ...            ...               ...
887      NaN  310000.0        NaN              0                 0
888   2018.0  260000.0    27000.0              0                 0
889   2013.0  390000.0    40000.0              0                 0
890   2014.0  180000.0        NaN              0                 0
891   2014.0  160000.0        NaN              0                 0

     name_7 SEATER MAHINDRA  name_9 SEATER MAHINDRA  name_Any type car  \
0                         0                       0                  0
1                         0                       0                  0
2                         0                       0                  0
3                         0                       0                  0
4                         0                       0                  0
..                      ...                     ...                ...
887                       0                       0                  0
888                       0                       0                  0
889                       0                       0                  0
890                       0                       0                  0
891                       0                       0                  0

     name_Audi A3 Cabriolet  name_Audi A4 1.8  ...  name_i want sale  \
0                         0                 0  ...                 0
1                         0                 0  ...                 0
2                         0                 0  ...                 0
3                         0                 0  ...                 0
```

```
4                          0                 0  …                 0
..                       …                 …  …               …
887                        0                 0  …                 0
888                        0                 0  …                 0
889                        0                 0  …                 0
890                        0                 0  …                 0
891                        0                 0  …                 0

     name_scratch less Tata  name_sell my car  name_selling car Ta  name_tata  \
0                          0                 0                    0          0
1                          0                 0                    0          0
2                          0                 0                    0          0
3                          0                 0                    0          0
4                          0                 0                    0          0
..                       …                 …                    …        …
887                        0                 0                    0          0
888                        0                 0                    0          0
889                        0                 0                    0          0
890                        0                 0                    0          0
891                        0                 0                    0          0

     name_tata Indica  name_tata zest 2017  name_urgent sale Ta  \
0                   0                    0                    0
1                   0                    0                    0
2                   0                    0                    0
3                   0                    0                    0
4                   0                    0                    0
..                …                    …                    …
887                 0                    0                    0
888                 0                    0                    0
889                 0                    0                    0
890                 0                    0                    0
891                 0                    0                    0

     name_urgent sell my  name_very good condition
0                      0                         0
1                      0                         0
2                      0                         0
3                      0                         0
4                      0                         0
..                   …                         …
887                    0                         0
888                    0                         0
889                    0                         0
890                    0                         0
891                    0                         0
```

```
[892 rows x 307 columns]
```

[33]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 892 entries, 0 to 891
Columns: 307 entries, year to name_very good condition
dtypes: float64(3), uint8(304)
memory usage: 285.8 KB
```

[34]:
```
df.isnull().sum()
```

[34]:
```
year                        50
Price                       35
total_kms                   54
fuel_type_LPG                0
fuel_type_Petrol             0
                            ..
name_tata Indica             0
name_tata zest 2017          0
name_urgent sale Ta          0
name_urgent sell my          0
name_very good condition     0
Length: 307, dtype: int64
```

[35]:
```
df['Price'].skew()
```

[35]:
```
7.655821693343426
```

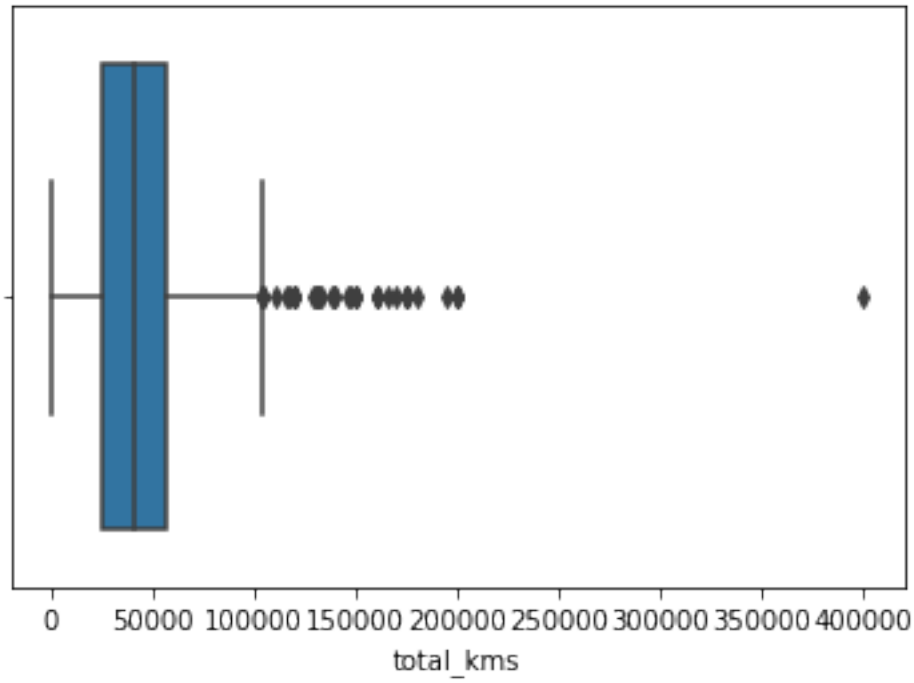
[36]:
```
mean=df['Price'].median()
```

[37]:
```
df['Price'].fillna(mean, inplace=True)
```

[38]:
```
sns.boxplot(df['total_kms'])
```

```
C:\Users\amits\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```

[38]:
```
<Axes: xlabel='total_kms'>
```



[39]:
```
q1=df['total_kms'].quantile(0.25)
q3=df['total_kms'].quantile(0.75)
iqr=q3-q1
print(iqr)
lower_limit=q1-1.5*iqr
upper_limit=q3+1.5*iqr
print(lower_limit,upper_limit)
```

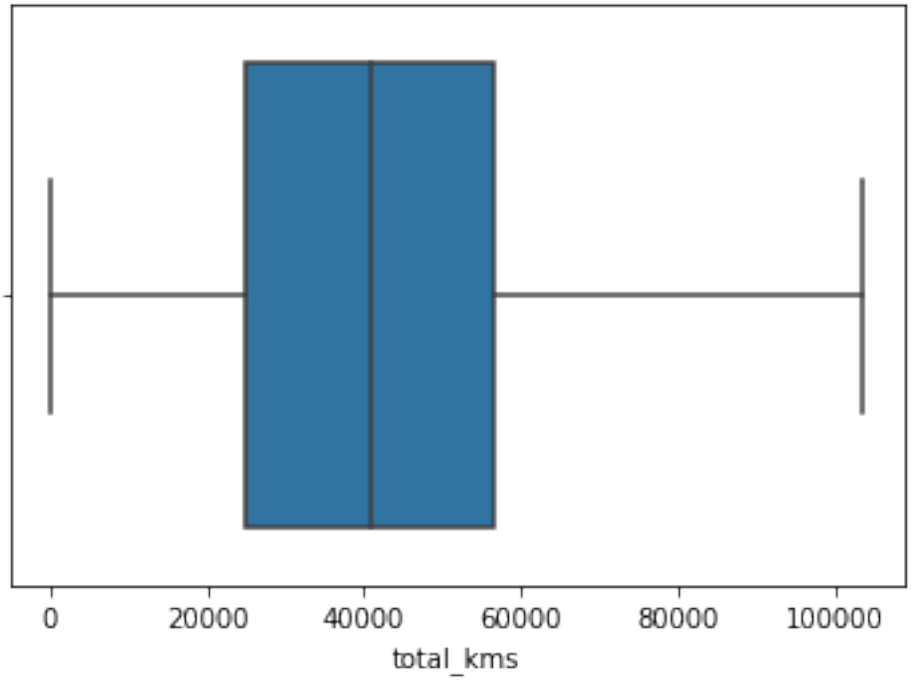
```
31437.5
-22156.25 103593.75
```

[40]:
```
df=df.copy()
df['total_kms']=np.where(df['total_kms'] > upper_limit, upper_limit,
                    np.where(df['total_kms'] < lower_limit, lower_limit,
                            df['total_kms']))
```

[41]:
```
sns.boxplot(df['total_kms'])
```

```
C:\Users\amits\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```

[41]: <Axes: xlabel='total_kms'>



[42]:
```
mean=df['total_kms'].median()
```

[43]:
```
df['total_kms'].fillna(mean, inplace=True)
```

[44]:
```
mode=df['year'].mode()
print(mode)
```

```
0    2015.0
Name: year, dtype: float64
```

[45]:
```
mode=df['year'].mode()[0]
print(mode)
```

```
2015.0
```

[46]:
```
df['year'].fillna(mode, inplace=True)
```

[47]:
```
df['year']=df['year'].astype(int)
```

[48]:
```
df['Price']=df['Price'].astype(int)
```

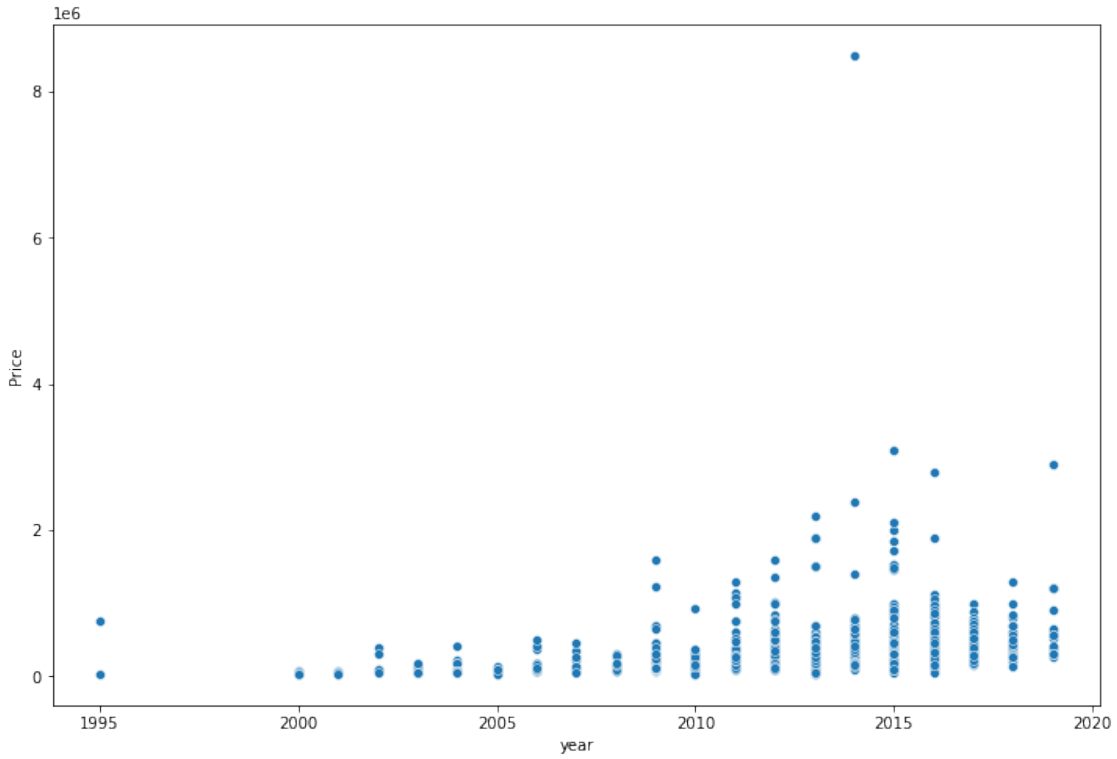
[49]:
```
df['total_kms']=df['total_kms'].astype(int)
```

[50]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 892 entries, 0 to 891
Columns: 307 entries, year to name_very good condition
dtypes: int32(3), uint8(304)
memory usage: 275.4 KB
```

[51]:
```
plt.figure(figsize=(12,8))
sns.scatterplot(x='year', y='Price', data=df)
```

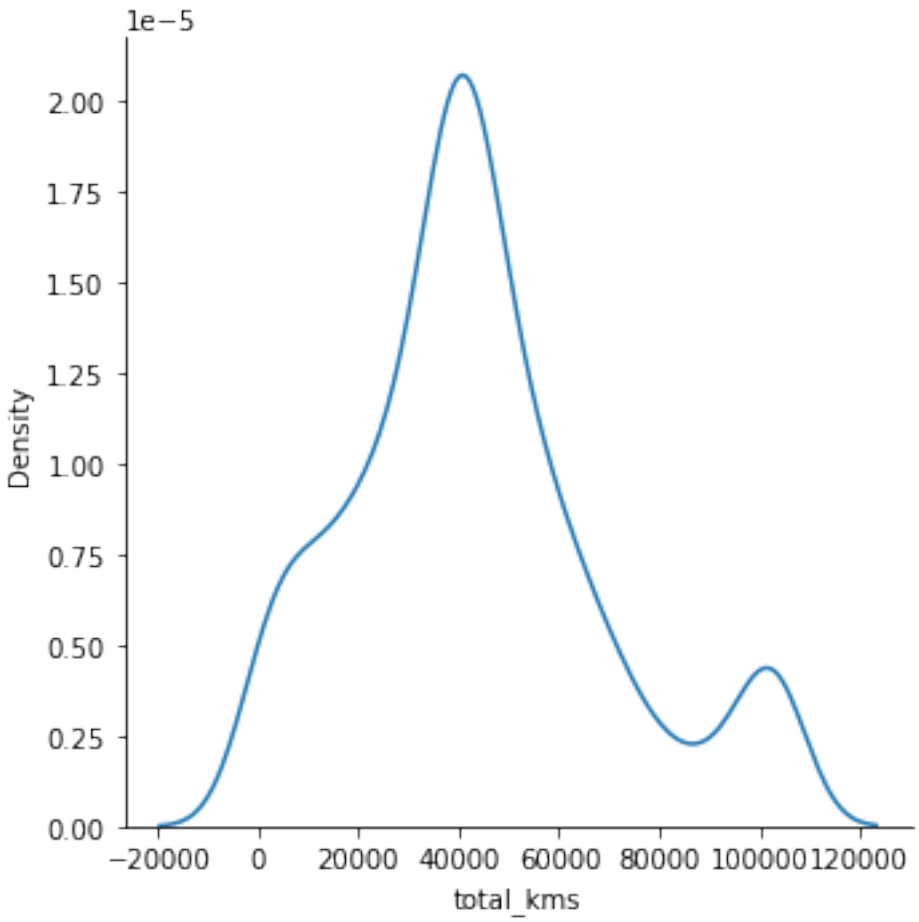
[51]: <Axes: xlabel='year', ylabel='Price'>



[52]:
```
sns.displot(x='total_kms', data=df, kind='kde')
```

```
C:\Users\amits\anaconda3\lib\site-packages\seaborn\axisgrid.py:88: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

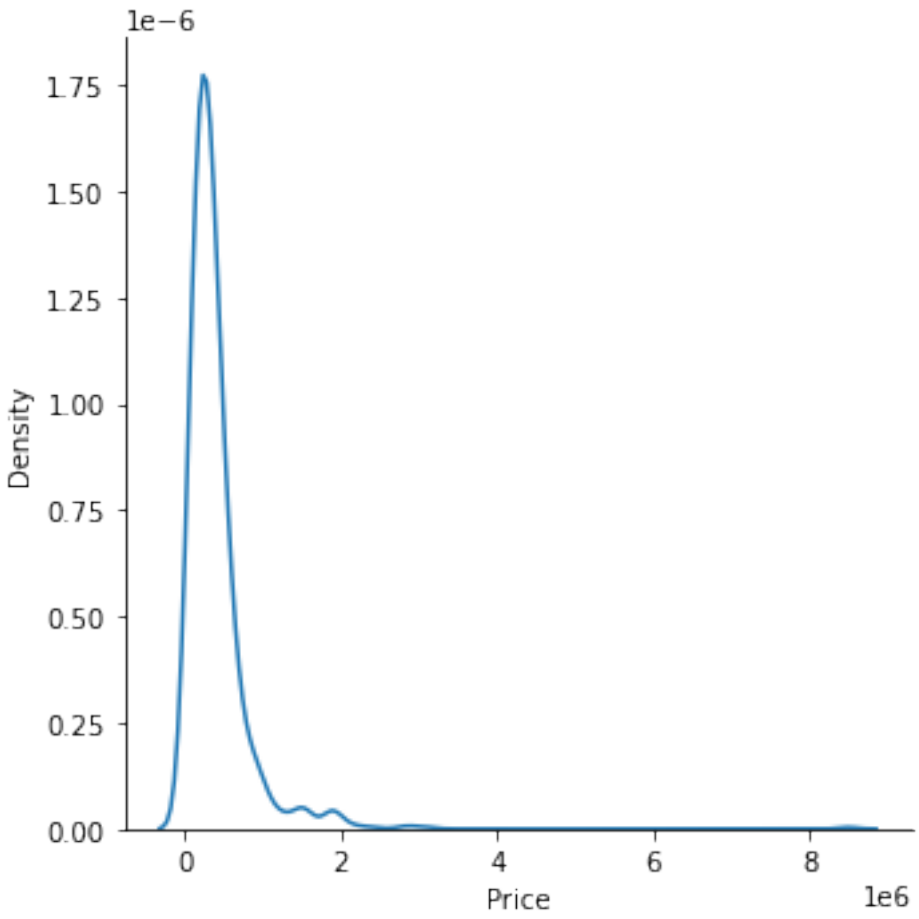
[52]: <seaborn.axisgrid.FacetGrid at 0x2058886c130>



[53]:
```
sns.displot(x='Price', data=df, kind='kde')
```

```
C:\Users\amits\anaconda3\lib\site-packages\seaborn\axisgrid.py:88: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

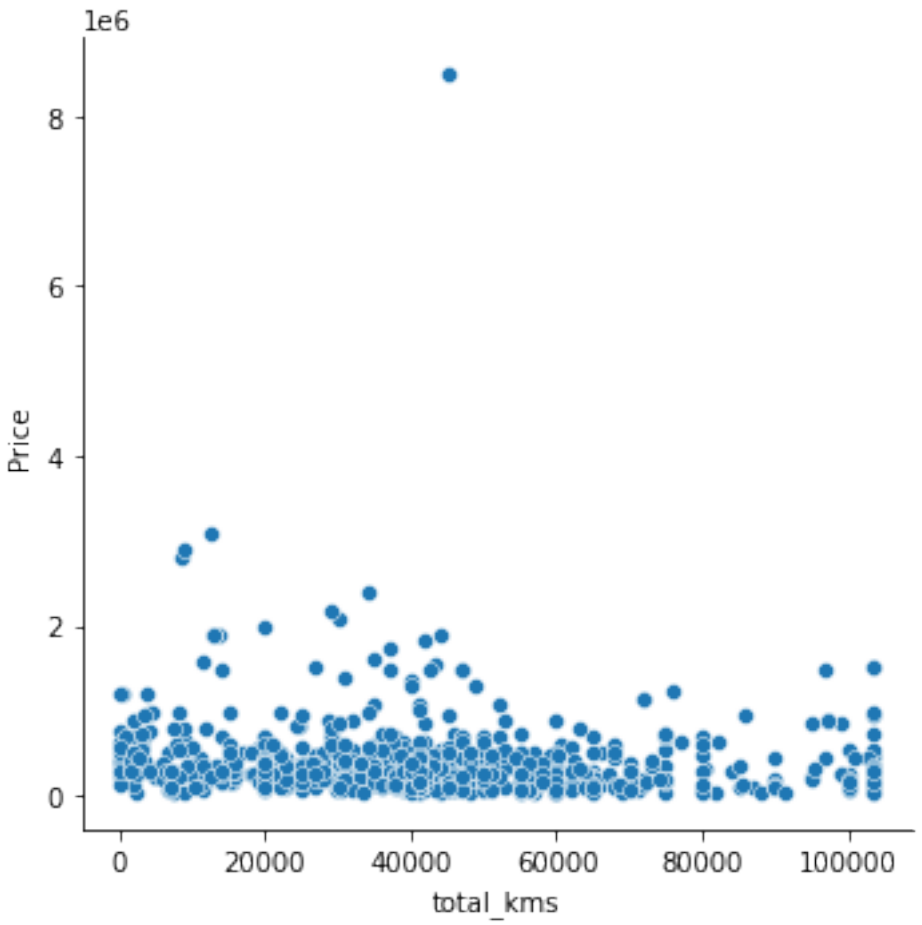
[53]:
```
<seaborn.axisgrid.FacetGrid at 0x205888d9ac0>
```



[54]:
```
sns.relplot(data=df, x='total_kms',y='Price', kind='scatter')
```

```
C:\Users\amits\anaconda3\lib\site-packages\seaborn\axisgrid.py:88: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

[54]:
```
<seaborn.axisgrid.FacetGrid at 0x205888d9220>
```



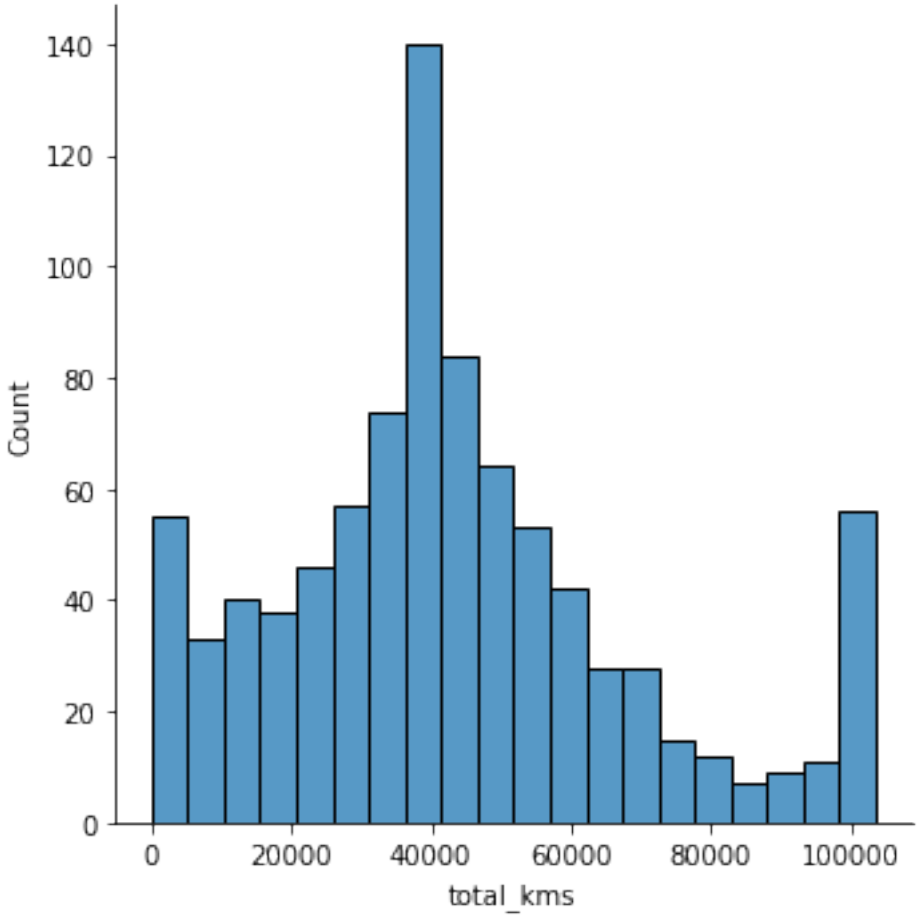
[55]:
```
df.columns
```

[55]:
```
Index(['year', 'Price', 'total_kms', 'fuel_type_LPG', 'fuel_type_Petrol',
       'name_7 SEATER MAHINDRA', 'name_9 SEATER MAHINDRA', 'name_Any type car',
       'name_Audi A3 Cabriolet', 'name_Audi A4 1.8',
       …
       'name_i want sale', 'name_scratch less Tata', 'name_sell my car',
       'name_selling car Ta', 'name_tata', 'name_tata Indica',
       'name_tata zest 2017', 'name_urgent sale Ta', 'name_urgent sell my',
       'name_very good condition'],
      dtype='object', length=307)
```

[56]:
```
sns.displot(data=df, x='total_kms', kind='hist', bins=20)
```

```
C:\Users\amits\anaconda3\lib\site-packages\seaborn\axisgrid.py:88: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

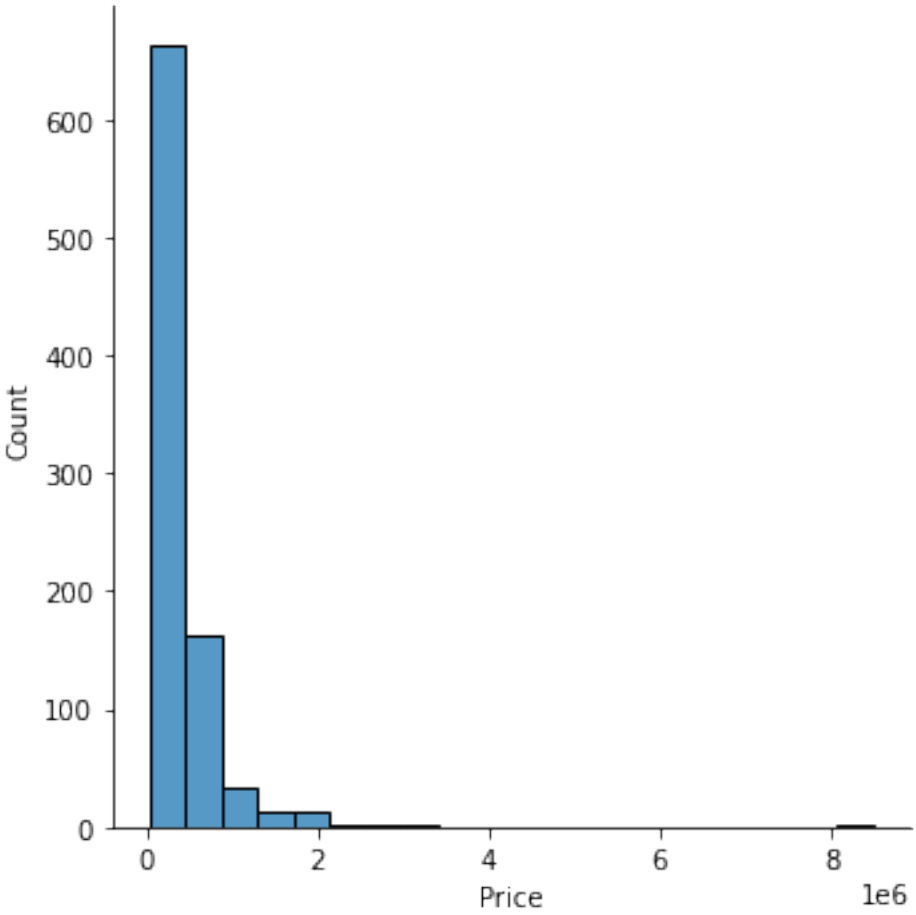
[56]: <seaborn.axisgrid.FacetGrid at 0x205888e14c0>



[57]:
```
sns.displot(data=df, x='Price', kind='hist', bins=20)
```

```
C:\Users\amits\anaconda3\lib\site-packages\seaborn\axisgrid.py:88: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

[57]: <seaborn.axisgrid.FacetGrid at 0x2058ac754f0>



[58]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 892 entries, 0 to 891
Columns: 307 entries, year to name_very good condition
dtypes: int32(3), uint8(304)
memory usage: 275.4 KB
```

[59]:
```
from sklearn.model_selection import train_test_split
```

[60]:
```
y= df['Price']
```

[61]:
```
x = df.drop('Price',axis=1)
```

[62]:
```
y
```

[62]:
```
0        80000
1       425000
```

```
2      299999
3      325000
4      575000
       ...
887    310000
888    260000
889    390000
890    180000
891    160000
Name: Price, Length: 892, dtype: int32
```

[63]:
```
x_train,x_test,y_train,y_test=train_test_split(x, y, test_size=20,␣
↪random_state=42)
```

[64]:
```
from sklearn.linear_model import LinearRegression
```

[65]:
```
model= LinearRegression()
```

[66]:
```
model.fit(x_train, y_train)
```

[66]:
```
LinearRegression()
```

[67]:
```
y_pred = model.predict(x_test)
```

[68]:
```
from sklearn.metrics import r2_score
```

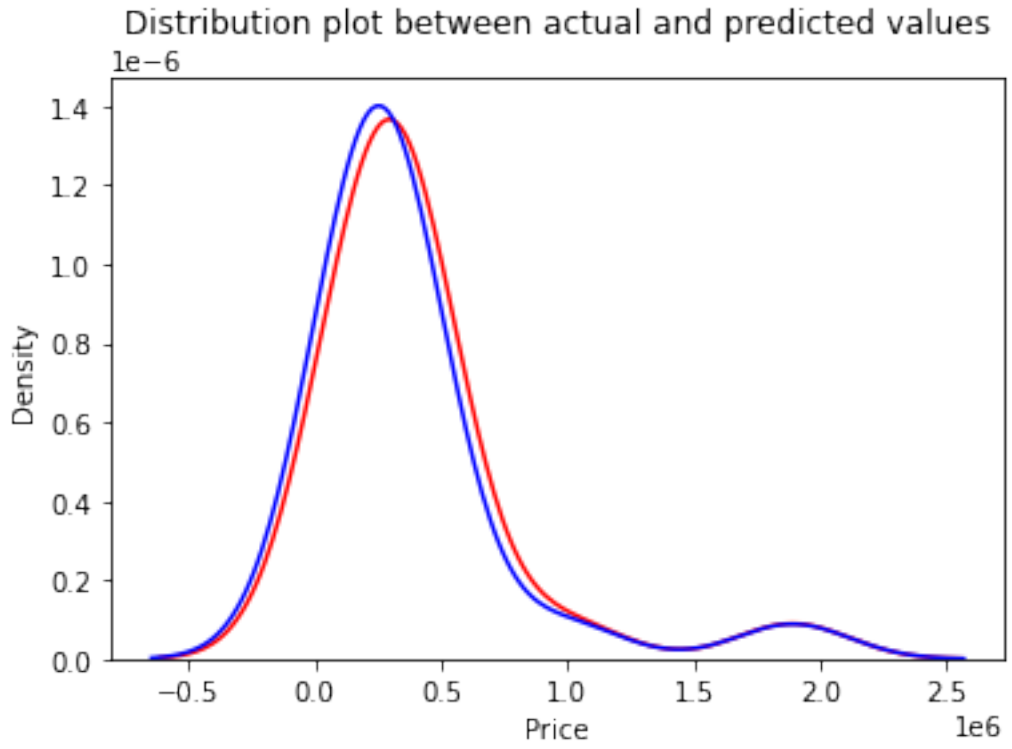
[69]:
```
r2_score(y_test, y_pred)
```

[69]:
```
0.9730805764681132
```

[70]:
```
# Distribution plot for actual and predicted values
ax = sns.distplot(y_test, hist=False, color="r", label="Actual Value")
sns.distplot(y_pred, color="b", hist=False, label="Predicted Value")
plt.title("Distribution plot between actual and predicted values")
plt.show()
```

```
C:\Users\amits\anaconda3\lib\site-packages\seaborn\distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `kdeplot` (an axes-level function for
kernel density plots).
  warnings.warn(msg, FutureWarning)
C:\Users\amits\anaconda3\lib\site-packages\seaborn\distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `kdeplot` (an axes-level function for
```

```
kernel density plots).
  warnings.warn(msg, FutureWarning)
```



[71]:
```
import xgboost as xgb
```

[72]:
```
xgb_regressor = xgb.XGBRegressor()

# Train the model on the training data
xgb_regressor.fit(X_train, y_train)
y_pred = xgb_regressor.predict(X_test)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_21556/3612386330.py in <module>
      2
      3 # Train the model on the training data
----> 4 xgb_regressor.fit(X_train, y_train)
      5 y_pred = xgb_regressor.predict(X_test)

NameError: name 'X_train' is not defined
```

[ ]:
```
r2 = r2_score(y_test, y_pred_)
```

[ ]:
```
import pickle
```

[ ]:
```
pickle.dump(model, open("carprice.pkl","wb"))
```