

In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import preprocessing, svm
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [2]:

```
df=pd.read_csv(r"C:\Users\Gowthami\Downloads\used_cars_data (1).csv")
df
```

Out[2]:

	S.No.	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owr
0	0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	
1	1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	
2	2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	
3	3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	
4	4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	
...	
7248	7248	Volkswagen Vento Diesel Trendline	Hyderabad	2011	89411	Diesel	Manual	
7249	7249	Volkswagen Polo GT TSI	Mumbai	2015	59000	Petrol	Automatic	
7250	7250	Nissan Micra Diesel XV	Kolkata	2012	28000	Diesel	Manual	
7251	7251	Volkswagen Polo GT TSI	Pune	2013	52262	Petrol	Automatic	
7252	7252	Mercedes-Benz E-Class 2009-2013 E 220 CDI Avan...	Kochi	2014	72443	Diesel	Automatic	

7253 rows × 14 columns

In [3]:

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

Out[3]:

S.No.		Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_
0	0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	
1	1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	
2	2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	
3	3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	
4	4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Se
5	5	Hyundai EON LPG Era Plus Option	Hyderabad	2012	75000	LPG	Manual	
6	6	Nissan Micra Diesel XV	Jaipur	2013	86999	Diesel	Manual	
7	7	Toyota Innova Crysta 2.8 GX AT 8S	Mumbai	2016	36000	Diesel	Automatic	
8	8	Volkswagen Vento Diesel Comfortline	Pune	2013	64430	Diesel	Manual	
9	9	Tata Indica Vista Quadrajet LS	Chennai	2012	65932	Diesel	Manual	Se



In [4]:

df.describe()

Out[4]:

	S.No.	Year	Kilometers_Driven	Seats	Price
count	7253.000000	7253.000000	7.253000e+03	7200.000000	6019.000000
mean	3626.000000	2013.365366	5.869906e+04	5.279722	9.479468
std	2093.905084	3.254421	8.442772e+04	0.811660	11.187917
min	0.000000	1996.000000	1.710000e+02	0.000000	0.440000
25%	1813.000000	2011.000000	3.400000e+04	5.000000	3.500000
50%	3626.000000	2014.000000	5.341600e+04	5.000000	5.640000
75%	5439.000000	2016.000000	7.300000e+04	5.000000	9.950000
max	7252.000000	2019.000000	6.500000e+06	10.000000	160.000000

In [5]:

df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7253 entries, 0 to 7252
Data columns (total 14 columns):
#   Column                Non-Null Count  Dtype
---  -
0   S.No.                 7253 non-null  int64
1   Name                  7253 non-null  object
2   Location              7253 non-null  object
3   Year                  7253 non-null  int64
4   Kilometers_Driven     7253 non-null  int64
5   Fuel_Type             7253 non-null  object
6   Transmission          7253 non-null  object
7   Owner_Type            7253 non-null  object
8   Mileage               7251 non-null  object
9   Engine                7207 non-null  object
10  Power                 7207 non-null  object
11  Seats                 7200 non-null  float64
12  New_Price             1006 non-null  object
13  Price                 6019 non-null  float64
dtypes: float64(2), int64(3), object(9)
memory usage: 793.4+ KB
```

In [6]:

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

Out[6]:

```
S.No.      0
Name       0
Location   0
Year       0
Kilometers_Driven  0
Fuel_Type  0
Transmission  0
Owner_Type  0
Mileage     2
Engine     46
Power      46
Seats      53
New_Price  6247
Price     1234
dtype: int64
```

In [7]:

```
df.fillna(value=0,inplace=True)
```

In [8]:

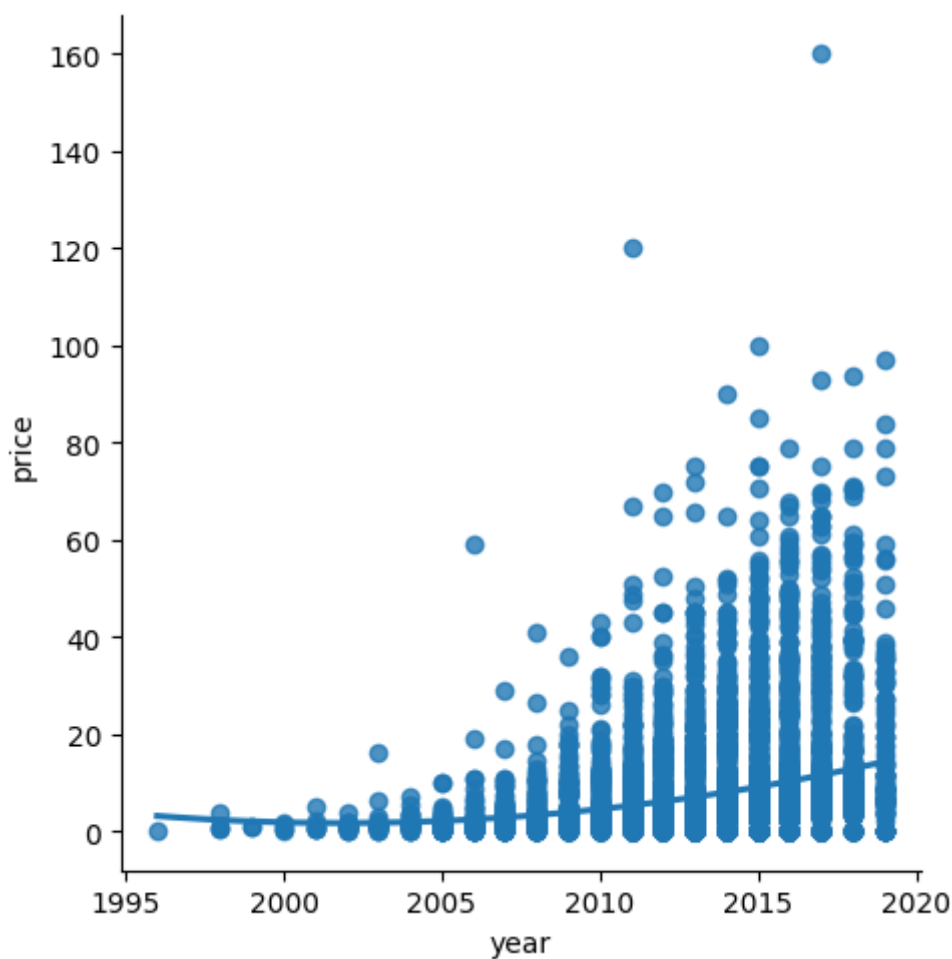
```
df=df[['Year','Price']]
#Taking only the selected two attributes from the dataset
df.columns=['year','price']
#Renaming the columns for easier writing of the code
```

In [9]:

```
sns.lmplot(x='year',y='price',data=df,order=2,ci=None)
```

Out[9]:

<seaborn.axisgrid.FacetGrid at 0x266559b2b50>



In [10]:

```
X=np.array(df['year']).reshape(-1,1)
```

In [11]:

```
y=np.array(df['price']).reshape(-1,1)
```

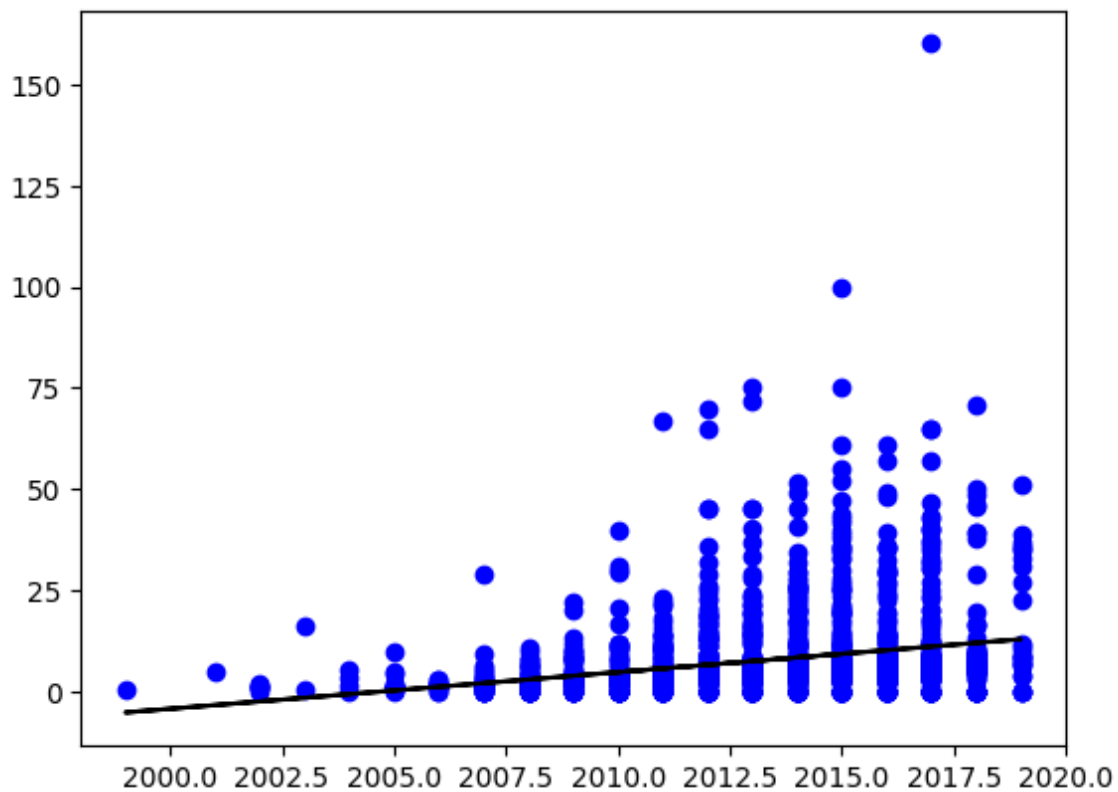
In [12]:

```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25)
reg=LinearRegression()
reg.fit(X_train,y_train)
print(reg.score(X_test,y_test))
```

0.046811645303847293

In [13]:

```
y_pred=reg.predict(X_test)
plt.scatter(X_test,y_test,color='b')
plt.plot(X_test,y_pred,color='k')
plt.show()
```

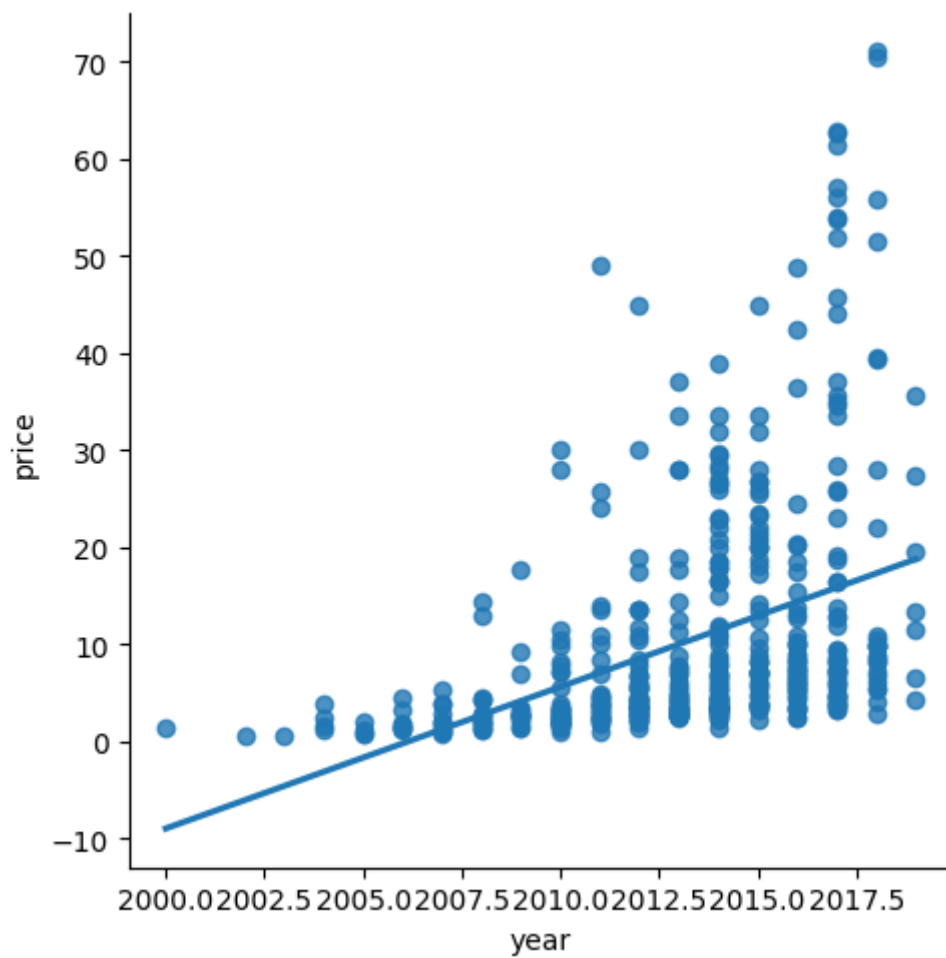


In [14]:

```
df500=df[:][:500]  
sns.lmplot(x='year',y='price',data=df500,order=1,ci=None)
```

Out[14]:

<seaborn.axisgrid.FacetGrid at 0x26655a51a50>



In [15]:

```

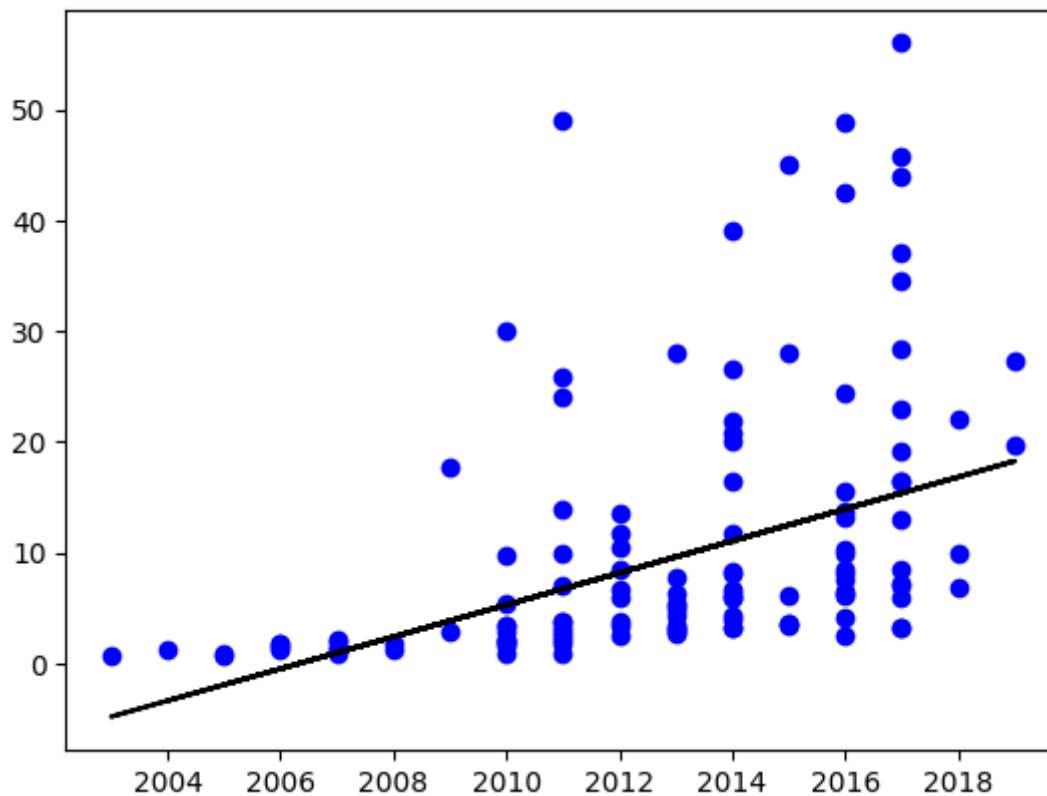
df500.fillna(method='ffill',inplace=True)
X=np.array(df500['year']).reshape(-1,1)
y=np.array(df500['price']).reshape(-1,1)
df500.dropna(inplace=True)
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25)
reg=LinearRegression()
reg.fit(X_train,y_train)
print("Regression:",reg.score(X_test,y_test))
y_pred=reg.predict(X_test)
plt.scatter(X_test,y_test,color='b')
plt.plot(X_test,y_pred,color='k')
plt.show

```

Regression: 0.17506618002968433

Out[15]:

<function matplotlib.pyplot.show(close=None, block=None)>



In [16]:

```

from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
model=LinearRegression()
model.fit(X_train,y_train)
y_pred=model.predict(X_test)
r2=r2_score(y_test,y_pred)
print("R2 score: ",r2)

```

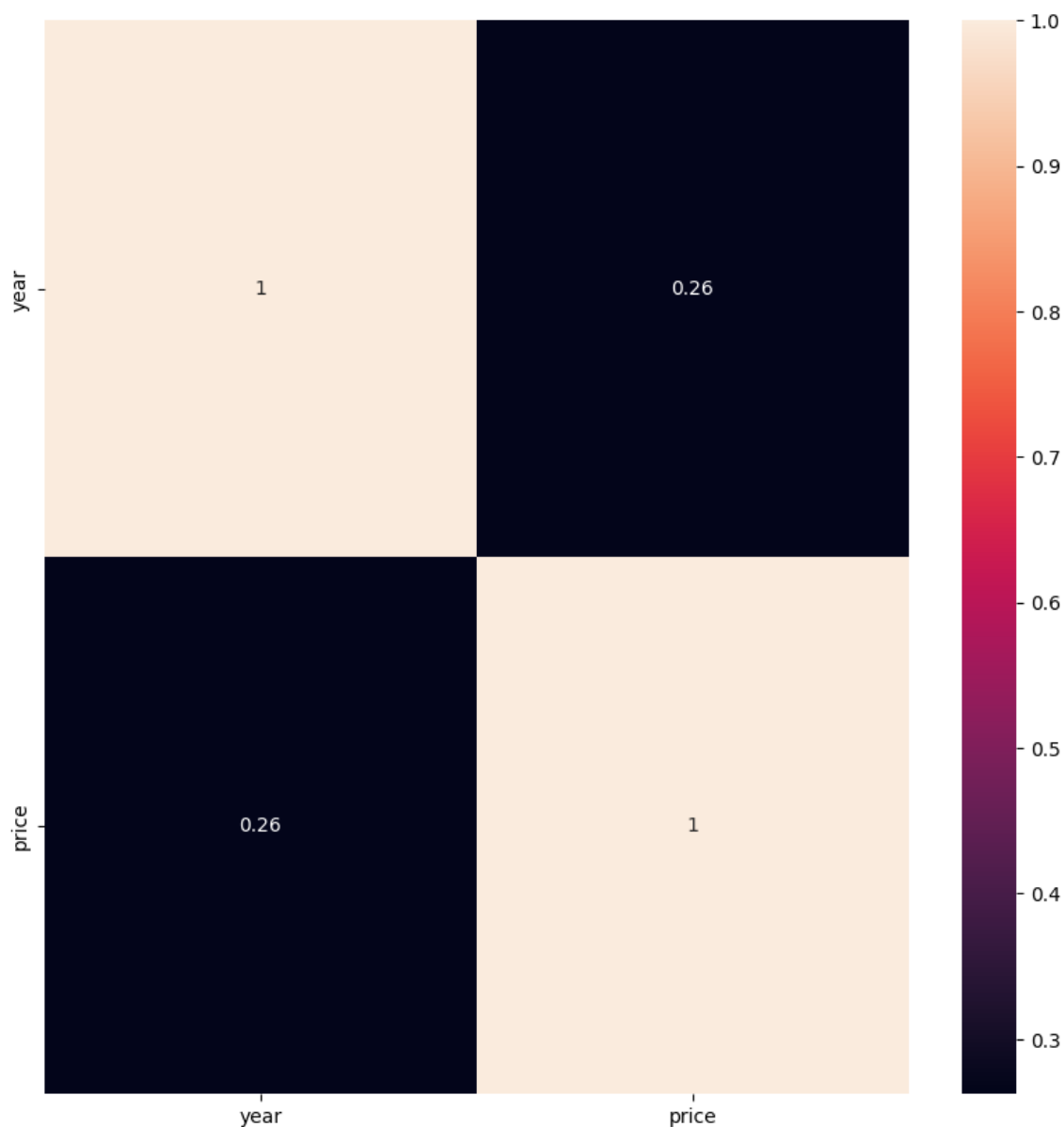
R2 score: 0.17506618002968433

In [17]:

```
plt.figure(figsize = (10, 10))  
sns.heatmap(df.corr(), annot = True)
```

Out[17]:

<Axes: >



In [18]:

```
import pandas as pd  
import numpy as np  
from sklearn import preprocessing  
import matplotlib.pyplot as plt  
import seaborn as sns  
sns.set(style="white")  
sns.set(style="whitegrid", color_codes=True)  
import warnings  
warnings.simplefilter(action='ignore')
```

In [19]:

```
db=pd.read_csv(r"C:\Users\Gowthami\Downloads\used_cars_data (1).csv")
```

db

Out[19]:

	S.No.	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owr
0	0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	
1	1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	
2	2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	
3	3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	
4	4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	
...
7248	7248	Volkswagen Vento Diesel Trendline	Hyderabad	2011	89411	Diesel	Manual	
7249	7249	Volkswagen Polo GT TSI	Mumbai	2015	59000	Petrol	Automatic	
7250	7250	Nissan Micra Diesel XV	Kolkata	2012	28000	Diesel	Manual	
7251	7251	Volkswagen Polo GT TSI	Pune	2013	52262	Petrol	Automatic	
7252	7252	Mercedes-Benz E-Class 2009-2013 E 220 CDI Avan...	Kochi	2014	72443	Diesel	Automatic	

7253 rows × 14 columns



In [20]:

```
db.head()
```

Out[20]:

	S.No.	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type
0	0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	F
1	1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	F
2	2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	F
3	3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	F
4	4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Sec

In [21]:

```
db.shape
```

Out[21]:

(7253, 14)

In [22]:

```
db.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7253 entries, 0 to 7252
Data columns (total 14 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   S.No.                 7253 non-null  int64
1   Name                  7253 non-null  object
2   Location              7253 non-null  object
3   Year                  7253 non-null  int64
4   Kilometers_Driven    7253 non-null  int64
5   Fuel_Type             7253 non-null  object
6   Transmission          7253 non-null  object
7   Owner_Type            7253 non-null  object
8   Mileage               7251 non-null  object
9   Engine                7207 non-null  object
10  Power                 7207 non-null  object
11  Seats                 7200 non-null  float64
12  New_Price             1006 non-null  object
13  Price                 6019 non-null  float64
dtypes: float64(2), int64(3), object(9)
memory usage: 793.4+ KB
```

In [23]:

```
db.describe()
```

Out[23]:

	S.No.	Year	Kilometers_Driven	Seats	Price
count	7253.000000	7253.000000	7.253000e+03	7200.000000	6019.000000
mean	3626.000000	2013.365366	5.869906e+04	5.279722	9.479468
std	2093.905084	3.254421	8.442772e+04	0.811660	11.187917
min	0.000000	1996.000000	1.710000e+02	0.000000	0.440000
25%	1813.000000	2011.000000	3.400000e+04	5.000000	3.500000
50%	3626.000000	2014.000000	5.341600e+04	5.000000	5.640000
75%	5439.000000	2016.000000	7.300000e+04	5.000000	9.950000
max	7252.000000	2019.000000	6.500000e+06	10.000000	160.000000

In [24]:

```
db.isnull().sum()
```

Out[24]:

```
S.No.      0
Name       0
Location   0
Year       0
Kilometers_Driven  0
Fuel_Type  0
Transmission  0
Owner_Type  0
Mileage     2
Engine     46
Power      46
Seats      53
New_Price  6247
Price     1234
dtype: int64
```

In [25]:

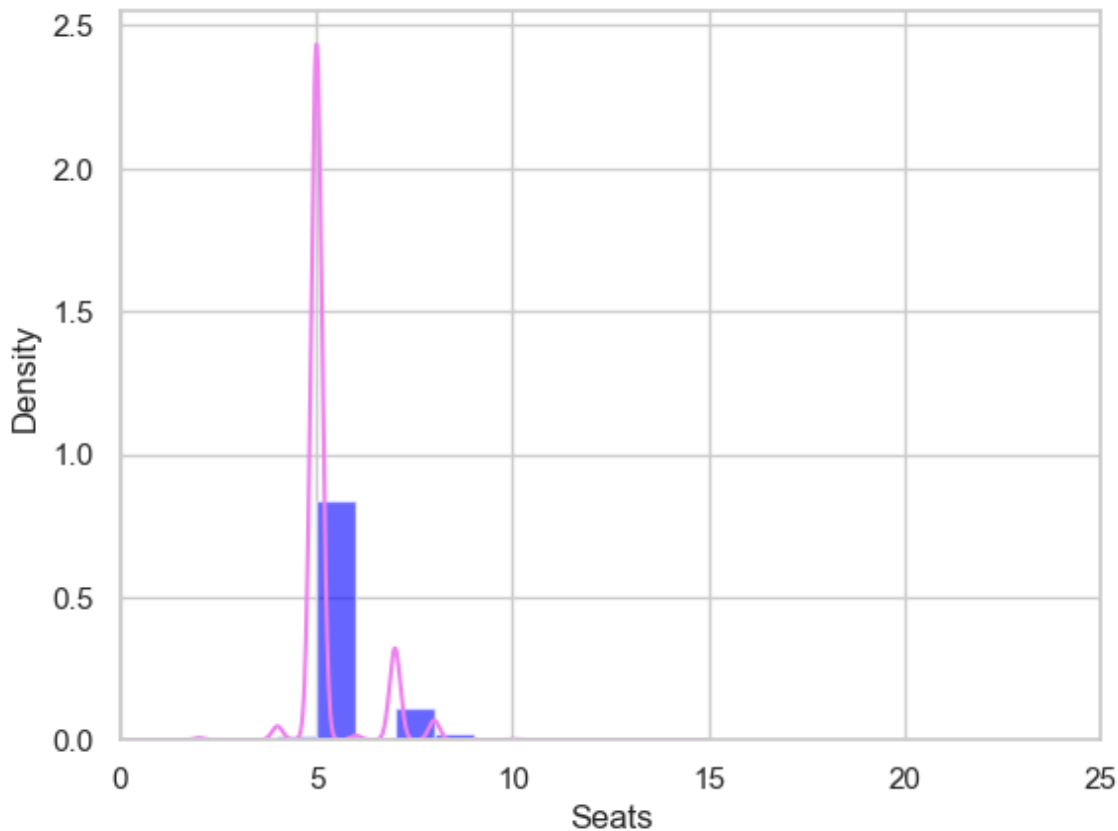
```
db.duplicated().any()
```

Out[25]:

```
False
```

In [26]:

```
ax=db['Seats'].hist(bins=10,density=True,stacked=True,color='blue',alpha=0.6)
db['Seats'].plot(kind='density',color='violet')
ax.set(xlabel='Seats')
plt.xlim(-0,25)
plt.show()
```



In [27]:

```
print(db["Seats"].mean(skipna=True))
print(db["Seats"].median(skipna=True))
```

```
5.279722222222222
5.0
```

In [28]:

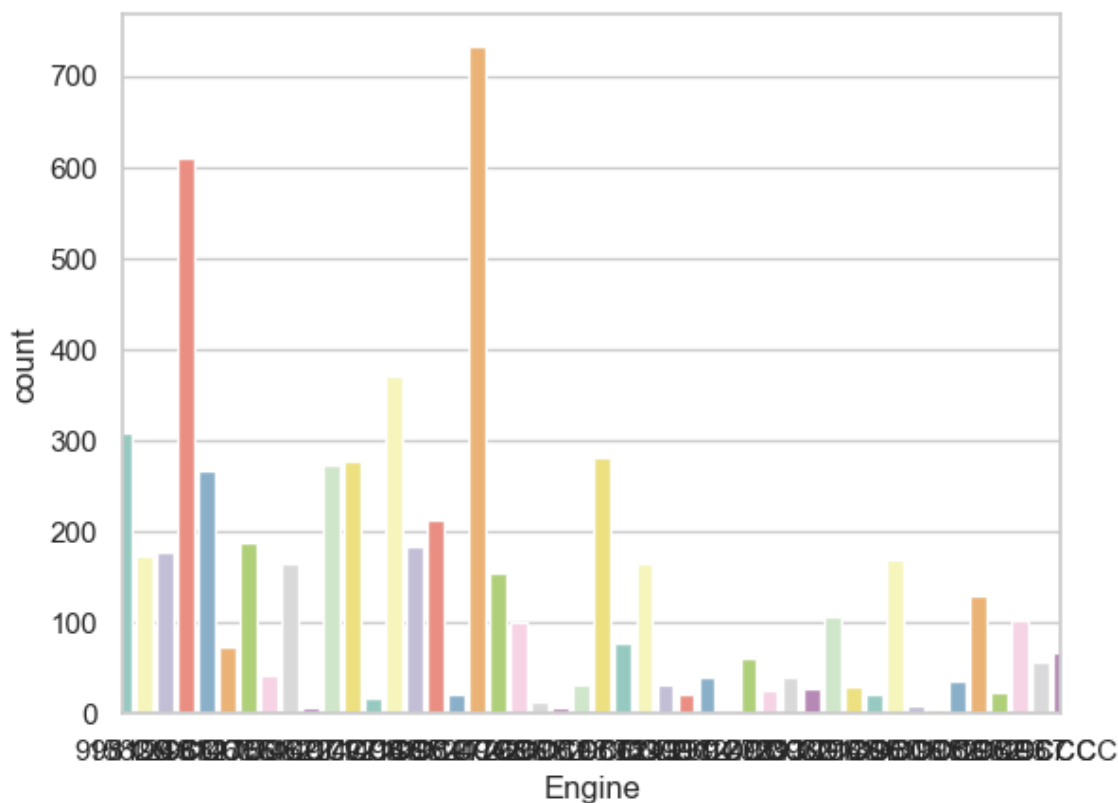
```
print(db["New_Price"].isnull().sum()/db.shape[0])
print(db["Price"].isnull().sum()/db.shape[0])
print(db["Mileage"].isnull().sum()/db.shape[0])
print(db["Engine"].isnull().sum()/db.shape[0])
print(db["Power"].isnull().sum()/db.shape[0])
```

```
0.8612987729215497
0.1701364952433476
0.0002757479663587481
0.006342203226251206
0.006342203226251206
```

In [29]:

```
print(db['Engine'].value_counts())
sns.countplot(x='Engine',data=db,palette='Set3')
plt.xlim(-0,45)
plt.show()
```

```
Engine
1197 CC      732
1248 CC      610
1498 CC      370
998 CC       309
1198 CC      281
...
1489 CC        1
1422 CC        1
2706 CC        1
1978 CC        1
1389 CC        1
Name: count, Length: 150, dtype: int64
```



In [30]:

```
data=db.copy()
data['Seats'].fillna(db['Seats'].median(skipna=True),inplace=True)
data.drop('New_Price',axis=1,inplace=True)
data['Price'].fillna(db['Price'].median(skipna=True),inplace=True)
data['Mileage'].fillna(db['Mileage'].value_counts().idxmax(),inplace=True)
data.drop('Engine',axis=1,inplace=True)
data.drop('Power',axis=1,inplace=True)
```

In [31]:

```
data.isnull().sum()
```

Out[31]:

```
S.No.      0
Name        0
Location    0
Year        0
Kilometers_Driven  0
Fuel_Type   0
Transmission  0
Owner_Type  0
Mileage      0
Seats        0
Price        0
dtype: int64
```

In [32]:

```
data.head()
```

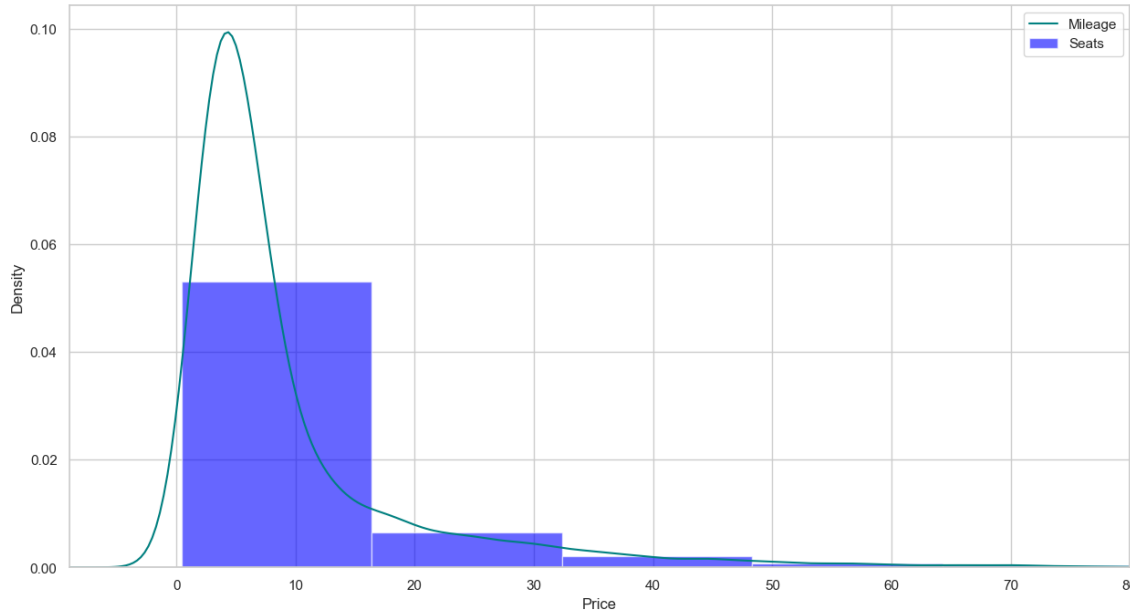
Out[32]:

	S.No.	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type
0	0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	F
1	1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	F
2	2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	F
3	3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	F
4	4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Sec



In [33]:

```
plt.figure(figsize=(15,8))
ax=db["Price"].hist(bins=10,density=True,stacked=True,color='blue',alpha=0.6)
db["Price"].plot(kind='density',color='teal')
ax.legend(['Mileage','Seats'])
ax.set(xlabel='Price')
plt.xlim(-9,80)
plt.show()
```



In [34]:

```
training=pd.get_dummies(data,columns=["S.No."])
final_train=training
final_train.head()
```

Out[34]:

	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mi
0	Maruti Wagon R LXi CNG	Mumbai	2010	72000	CNG	Manual	First	
1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	
2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	
3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	
4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	

5 rows × 7263 columns



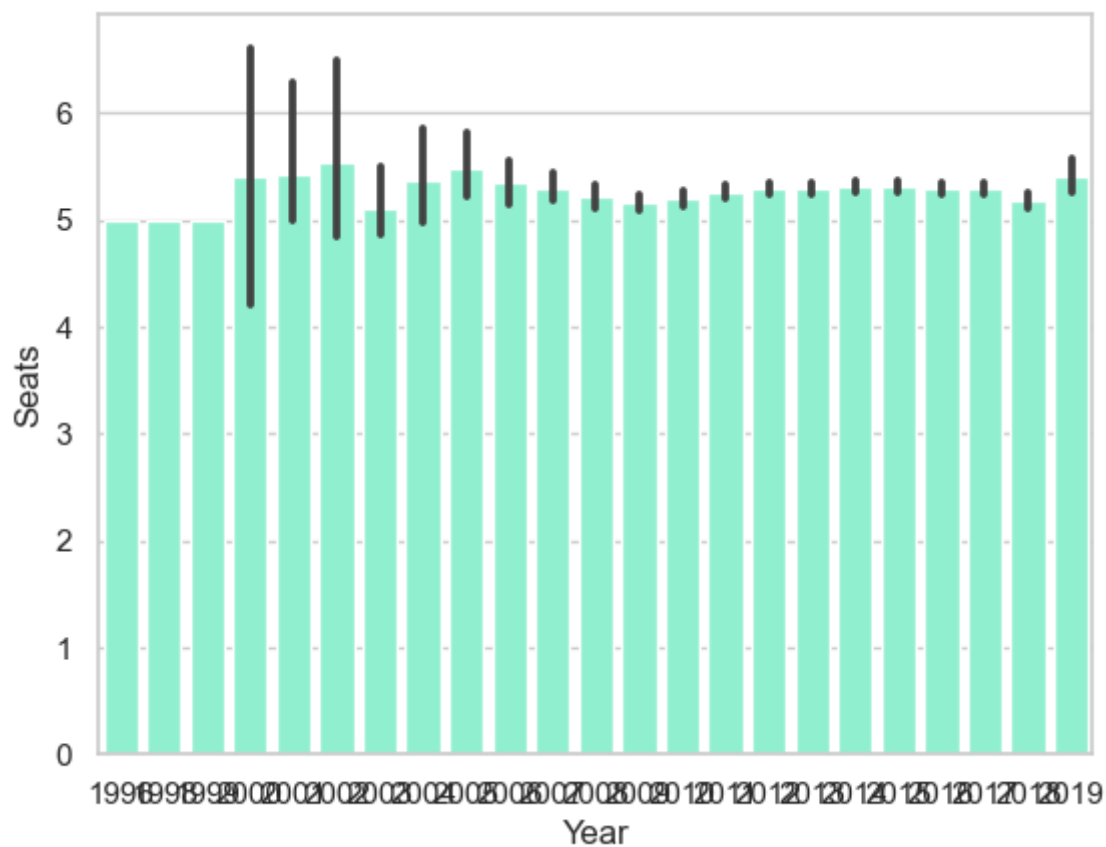
In [35]:

```
sns.barplot(x='Price',y='Year',data=final_train,color='mediumturquoise')  
plt.show()
```



In [36]:

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.barplot(x='Year',y='Seats',data=db,color='aquamarine')
plt.show()
```



In []: