importing libraries

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl
%matplotlib inline
mpl.style.use('ggplot')
from sklearn.model_selection import train_test_split
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import make_column_transformer
from sklearn.pipeline import make_pipeline
from sklearn.metrics import r2_score
```

importing data

In [2]:

```
car=pd.read_csv('C:/Users/vella/Downloads/Book1.csv')
```

In [3]:

```
car.head()
```

Out[3]:

	name	company	year	Price	kms_driven	fuel_type
0	Hyundai Santro Xing	Hyundai	2007	80000	45000	Petrol
1	Mahindra Jeep CL550	Mahindra	2006	425000	40	Diesel
2	Hyundai Grand i10	Hyundai	2014	325000	28000	Petrol
3	Ford EcoSport Titanium	Ford	2014	575000	36000	Diesel
4	Ford Figo	Ford	2012	175000	41000	Diesel

In [4]:

```
car.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 816 entries, 0 to 815
Data columns (total 6 columns):
name          816 non-null object
company       816 non-null object
year          816 non-null int64
Price         816 non-null int64
kms_driven    816 non-null int64
fuel_type     816 non-null object
dtypes: int64(3), object(3)
memory usage: 38.3+ KB
```

removing outlier

In [5]:

```
car = car[car['Price']<6e6].reset_index(drop = True)
```

In [6]:

```
car.describe(include='all')
```

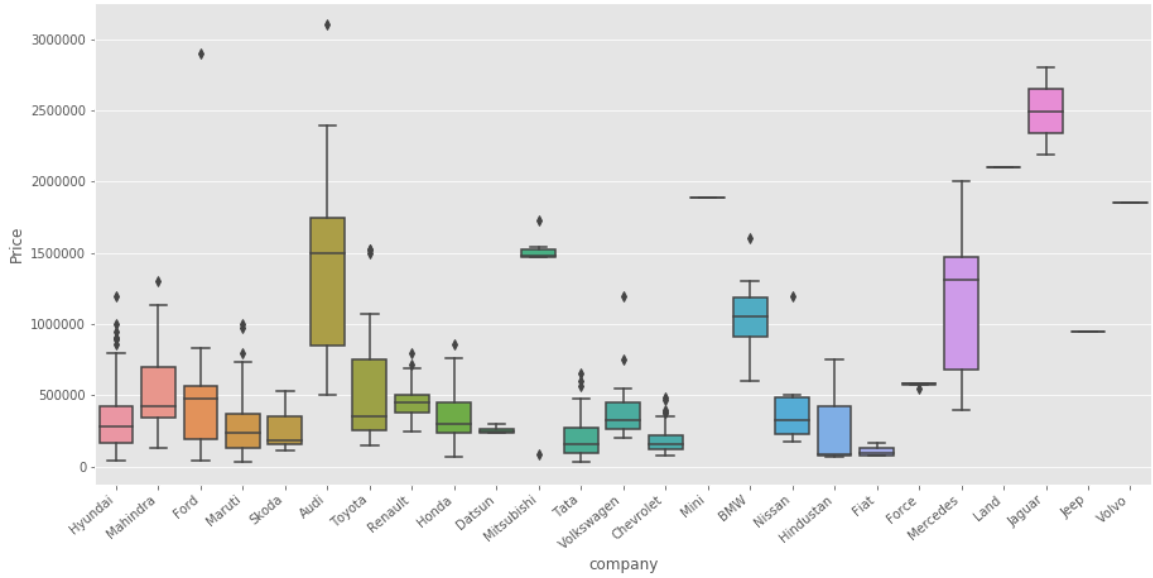
Out[6]:

	name	company	year	Price	kms_driven	fuel_type
count	815	815	815.000000	8.150000e+02	815.000000	815
unique	254	25	NaN	NaN	NaN	3
top	Maruti Suzuki Swift	Maruti	NaN	NaN	NaN	Petrol
freq	51	221	NaN	NaN	NaN	428
mean	NaN	NaN	2012.442945	4.017933e+05	46277.096933	NaN
std	NaN	NaN	4.005079	3.815888e+05	34318.459638	NaN
min	NaN	NaN	1995.000000	3.000000e+04	0.000000	NaN
25%	NaN	NaN	2010.000000	1.750000e+05	27000.000000	NaN
50%	NaN	NaN	2013.000000	2.999990e+05	41000.000000	NaN
75%	NaN	NaN	2015.000000	4.900000e+05	56879.000000	NaN
max	NaN	NaN	2019.000000	3.100000e+06	400000.000000	NaN

Checking relationship of Company with Price

In [7]:

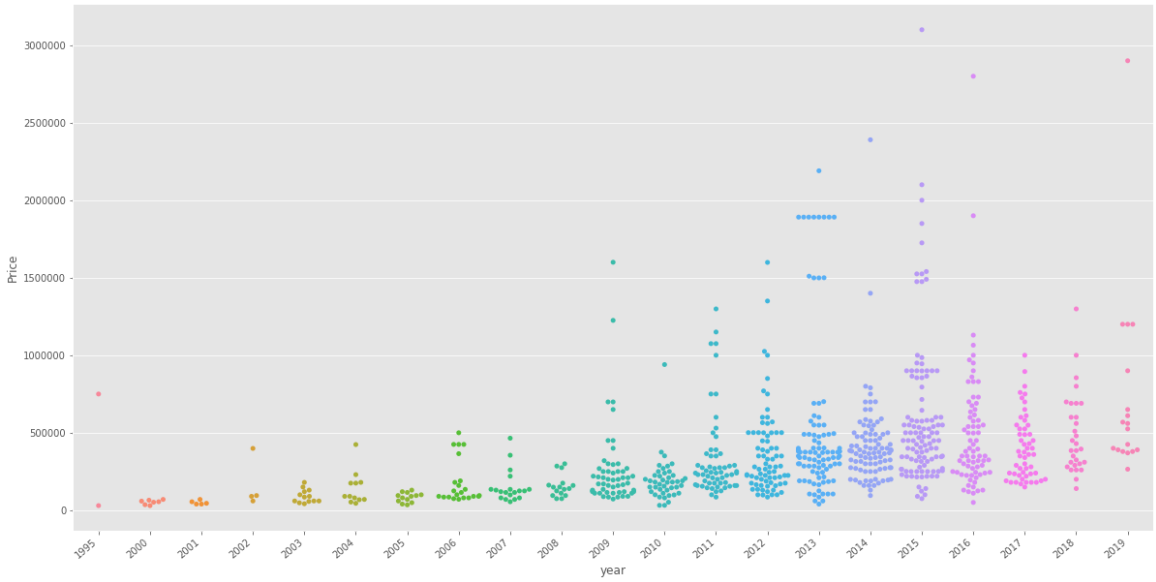
```
plt.subplots(figsize=(15,7))
ax=sns.boxplot(x='company',y='Price',data=car)
ax.set_xticklabels(ax.get_xticklabels(),rotation=40,ha='right')
plt.show()
```



Checking relationship of Year with Price

In [8]:

```
plt.subplots(figsize=(20,10))
ax=sns.swarmplot(x='year',y='Price',data=car)
ax.set_xticklabels(ax.get_xticklabels(),rotation=40,ha='right')
plt.show()
```



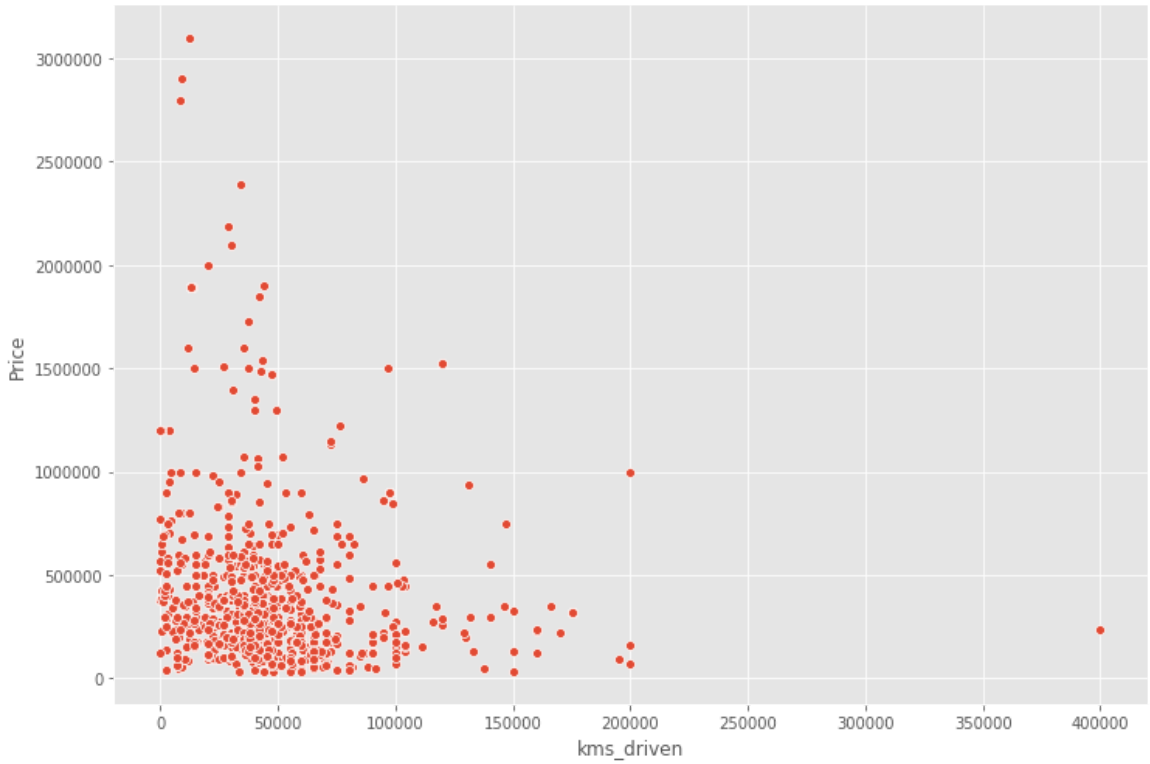
Checking relationship of kms_driven with Price

In [9]:

```
sns.relplot(x='kms_driven',y='Price',data=car,height=7,aspect=1.5)
```

Out[9]:

```
<seaborn.axisgrid.FacetGrid at 0x2332e7eedd8>
```



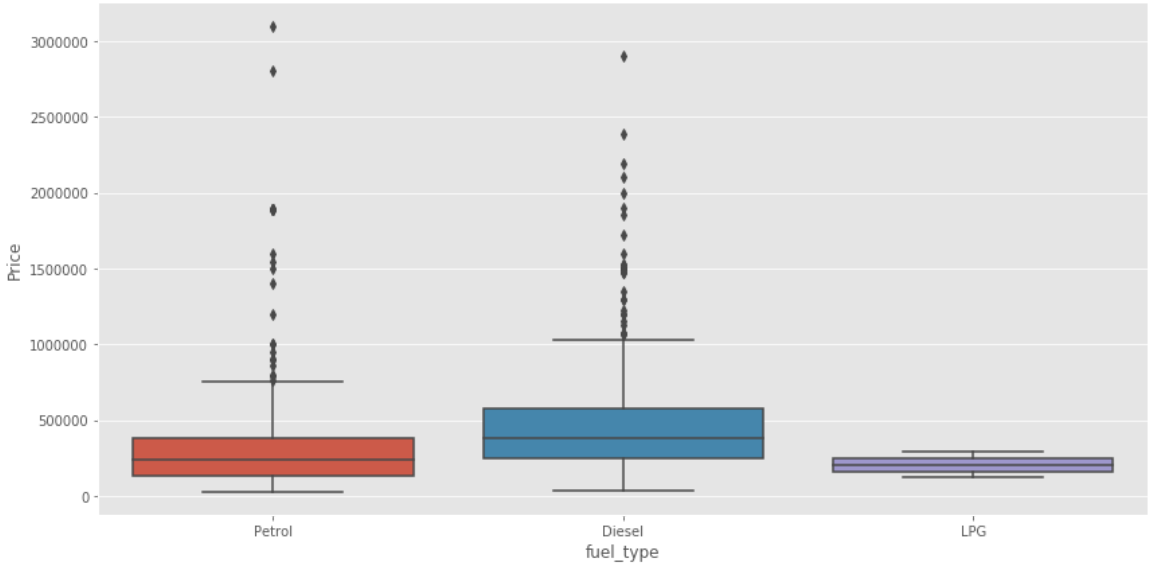
Checking relationship of Fuel Type with Price

In [10]:

```
plt.subplots(figsize=(14,7))
sns.boxplot(x='fuel_type',y='Price',data=car)
```

Out[10]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2332ee1e908>
```



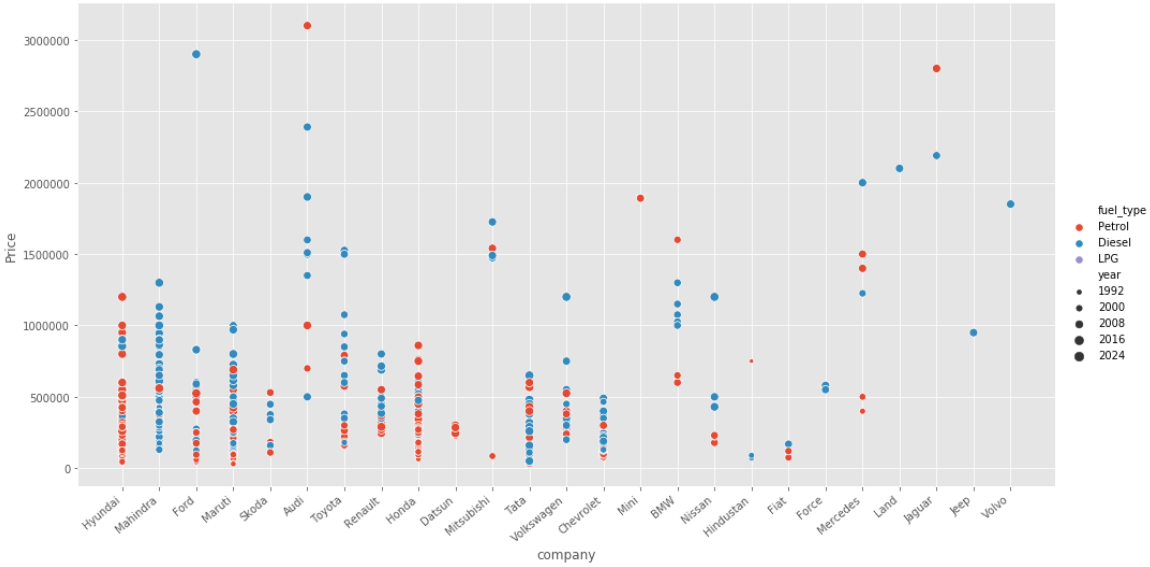
Relationship of Price with FuelType, Year and Company mixed

In [11]:

```
ax=sns.relplot(x='company',y='Price',data=car,hue='fuel_type',size='year',height=7,a
spect=2)
ax.set_xticklabels(rotation=40,ha='right')
```

Out[11]:

```
<seaborn.axisgrid.FacetGrid at 0x2332ee0e828>
```



Extracting Training Data

In [12]:

```
X=car[['name','company','year','kms_driven','fuel_type']]
y=car['Price']
```

In [13]:

```
X
```

Out[13]:

	name	company	year	kms_driven	fuel_type
0	Hyundai Santro Xing	Hyundai	2007	45000	Petrol
1	Mahindra Jeep CL550	Mahindra	2006	40	Diesel
2	Hyundai Grand i10	Hyundai	2014	28000	Petrol
3	Ford EcoSport Titanium	Ford	2014	36000	Diesel
4	Ford Figo	Ford	2012	41000	Diesel
5	Hyundai Eon	Hyundai	2013	25000	Petrol
6	Ford EcoSport Ambiente	Ford	2016	24530	Diesel
7	Maruti Suzuki Alto	Maruti	2015	60000	Petrol
8	Skoda Fabia Classic	Skoda	2010	60000	Petrol
9	Maruti Suzuki Stingray	Maruti	2015	30000	Petrol
10	Hyundai Elite i20	Hyundai	2014	32000	Petrol
11	Mahindra Scorpio SLE	Mahindra	2015	48660	Diesel
12	Hyundai Santro Xing	Hyundai	2007	45000	Petrol
13	Mahindra Jeep CL550	Mahindra	2006	40	Diesel
14	Audi A8	Audi	2017	4000	Petrol
15	Audi Q7	Audi	2014	16934	Diesel
16	Mahindra Scorpio S10	Mahindra	2016	43000	Diesel
17	Maruti Suzuki Alto	Maruti	2014	35550	Petrol
18	Mahindra Scorpio S10	Mahindra	2016	43000	Diesel
19	Mahindra Scorpio S10	Mahindra	2016	39522	Diesel
20	Maruti Suzuki Alto	Maruti	2015	39000	Petrol
21	Hyundai i20 Sportz	Hyundai	2012	55000	Petrol
22	Hyundai i20 Sportz	Hyundai	2012	55000	Petrol
23	Hyundai i20 Sportz	Hyundai	2012	55000	Petrol
24	Maruti Suzuki Alto	Maruti	2017	72000	Petrol
25	Maruti Suzuki Vitara	Maruti	2016	15975	Diesel
26	Maruti Suzuki Alto	Maruti	2008	70000	Petrol
27	Mahindra Bolero DI	Mahindra	2017	23452	Diesel
28	Maruti Suzuki Swift	Maruti	2014	35522	Diesel
29	Mahindra Scorpio S10	Mahindra	2015	48508	Diesel
...	...	...	...	...	...
785	Maruti Suzuki Wagon	Maruti	2006	7000	Petrol
786	Hyundai Eon	Hyundai	2018	25000	Petrol
787	Tata Manza	Tata	2015	100000	Diesel
788	Toyota Etios G	Toyota	2013	42000	Petrol
789	Hyundai Getz Prime	Hyundai	2009	20000	Petrol

	name	company	year	kms_driven	fuel_type
790	Toyota Qualis	Toyota	2003	100000	Diesel
791	Hyundai Santro Xing	Hyundai	2004	137495	Petrol
792	Tata Indica eV2	Tata	2016	91200	Diesel
793	Honda City 1.5	Honda	2009	55000	Petrol
794	Tata Zest XE	Tata	2017	120000	Diesel
795	Mahindra Quanto C4	Mahindra	2013	63000	Diesel
796	Tata Indigo eCS	Tata	2016	104000	Diesel
797	Maruti Suzuki Swift	Maruti	2016	146000	Diesel
798	Hyundai Elite i20	Hyundai	2011	40000	Petrol
799	Hyundai i20 Select	Hyundai	2011	40000	Petrol
800	Chevrolet Tavera Neo	Chevrolet	2007	100800	Diesel
801	Maruti Suzuki Dzire	Maruti	2016	150000	Diesel
802	Hyundai Elite i20	Hyundai	2018	2100	Petrol
803	Honda City VX	Honda	2016	95000	Petrol
804	Maruti Suzuki Dzire	Maruti	2016	2500	Diesel
805	Hyundai Getz	Hyundai	2006	80000	Petrol
806	Mercedes Benz C	Mercedes	2006	15000	Petrol
807	Maruti Suzuki Alto	Maruti	2005	65000	Petrol
808	Maruti Suzuki Swift	Maruti	2009	51000	Diesel
809	Skoda Fabia	Skoda	2009	45000	Petrol
810	Maruti Suzuki Ritz	Maruti	2011	50000	Petrol
811	Tata Indica V2	Tata	2009	30000	Diesel
812	Toyota Corolla Altis	Toyota	2009	132000	Petrol
813	Tata Zest XM	Tata	2018	27000	Diesel
814	Mahindra Quanto C8	Mahindra	2013	40000	Diesel

815 rows × 5 columns

In [14]:

```
y.shape
```

Out[14]:

```
(815,)
```

Applying Train Test Split

In [15]:

```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2)
```

In [16]:

```
ohe=OneHotEncoder()
ohe.fit(X[['name','company','fuel_type']])
```

Out[16]:

```
OneHotEncoder(categorical_features=None, categories=None, drop=None,
              dtype=<class 'numpy.float64'>, handle_unknown='error',
              n_values=None, sparse=True)
```

Creating a column transformer to transform categorical columns

In [17]:

```
column_trans=make_column_transformer((OneHotEncoder(categories=ohe.categories_),['name','company','fuel_type']),
                                     remainder='passthrough')
```

Linear Regression Model

In [18]:

```
lr=LinearRegression()
```

Making a pipeline

In [19]:

```
pipe=make_pipeline(column_trans,lr)
```

Fitting the model

In [20]:

```
pipe.fit(X_train,y_train)
```

Out[20]:

```
Pipeline(memory=None,
         steps=[('columntransformer',
                 ColumnTransformer(n_jobs=None, remainder='passthrough',
                                   sparse_threshold=0.3,
                                   transformer_weights=None,
                                   transformers=[('onehotencoder',
                                                  OneHotEncoder(categori
cal_features=None,
                                                                categori
es=[array(['Audi A3 Cabriolet', 'Audi A4 1.8', 'Audi A4 2.0', 'Audi A6
2.0',
       'Audi A8', 'Audi Q3 2.0', 'Audi Q5 2.0', 'Audi Q7', 'BMW 3 Serie
s',
       'B...
       'Renault', 'Skoda', 'Tata', 'Toyota', 'Volkswagen', 'Volvo'],
      dtype=object),

array(['Diesel', 'LPG', 'Petrol'], dtype=object)],
                                                                drop=Non
e,
                                                                dtype=<c
lass 'numpy.float64'>,
                                                                handle_u
nknown='error',
                                                                n_values
=None,
                                                                sparse=T
rue),
                                                  ['name', 'company',
                                                   'fuel_type'])],
                                   verbose=False)),
                ('linearregression',
                 LinearRegression(copy_X=True, fit_intercept=True, n_job
s=None,
                                  normalize=False))],
         verbose=False)
```

In [21]:

```
y_pred=pipe.predict(X_test)
```

Checking R2 Score

In [22]:

```
r2_score(y_test,y_pred)
```

Out[22]:

```
0.6413860541329858
```

Finding the model with a random state of TrainTestSplit

In [ ]:

```
scores=[]
for i in range(1000):
    X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.1,random_state=i)
    lr=LinearRegression()
    pipe=make_pipeline(column_trans,lr)
    pipe.fit(X_train,y_train)
    y_pred=pipe.predict(X_test)
    scores.append(r2_score(y_test,y_pred))
```

In [ ]:

```
np.argmax(scores)
```

In [ ]:

```
scores[np.argmax(scores)]
```

The best model is found at a certain random state

In [42]:

```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.1,random_state=np.arg
max(scores))
lr=LinearRegression()
pipe=make_pipeline(column_trans,lr)
pipe.fit(X_train,y_train)
y_pred=pipe.predict(X_test)
r2_score(y_test,y_pred)
```

Out[42]:

```
0.920088412025344
```

Trying to predict the cost of car

Method 1

In [43]:

```
pipe.predict(pd.DataFrame(columns=X_test.columns,data=np.array(['Maruti Suzuki Swif
t','Maruti',2019,100,'Petrol']).reshape(1,5)))
```

Out[43]:

```
array([416109.14071676])
```

Method 2

In [44]:

```
pipe.predict(pd.DataFrame(columns=['name','company','year','kms_driven','fuel_type'
],data=np.array(['Maruti Suzuki Swift','Maruti',2019,100,'Petrol']).reshape(1,5)))
```

Out[44]:

```
array([416109.14071676])
```