```
import numpy as np
import pandas as pd
```

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

```
df.head()
```

	brand	km_driven	fuel	owner	selling_price
0	Maruti	145500	Diesel	First Owner	450000
1	Skoda	120000	Diesel	Second Owner	370000
2	Honda	140000	Petrol	Third Owner	158000
3	Hyundai	127000	Diesel	First Owner	225000
4	Maruti	120000	Petrol	First Owner	130000

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df['brand'].value_counts()
```

brand	count
Maruti	2448
Hyundai	1415
Mahindra	772
Tata	734
Toyota	488
Honda	467
Ford	397
Chevrolet	230
Renault	228
Volkswagen	186
BMW	120
Skoda	105
Nissan	81
Jaguar	71
Volvo	67
Datsun	65
Mercedes-Benz	54
Fiat	47
Audi	40
Lexus	34
Jeep	31
Mitsubishi	14
Land	6
Force	6
Isuzu	5
Ambassador	4
Kia	4
MG	3
Daewoo	3
Ashok	1
Opel	1
Peugeot	1

dtype: int64

```
df['brand'].nunique()
```

```
32
```

Suggested code may be subject to a license | abhi-kuks/100-days-of-ML

```
df['fuel'].nunique()
```

```
4
```

```
df['owner'].value_counts()
```

owner	count
First Owner	5289
Second Owner	2105
Third Owner	555
Fourth & Above Owner	174
Test Drive Car	5

dtype: int64

1. OneHotEncoding using Pandas

```
pd.get_dummies(df,columns=['fuel','owner'])
```

	brand	km_driven	selling_price	fuel_CNG	fuel_Diesel	fuel_LPG	fuel_Petrol	owner_First Owner	owner_Fourth & Above Owner	owner_Second Owner	owner_Te Drive C
0	Maruti	145500	450000	False	True	False	False	True	False	False	Fal
1	Skoda	120000	370000	False	True	False	False	False	False	True	Fal
2	Honda	140000	158000	False	False	False	True	False	False	False	Fal
3	Hyundai	127000	225000	False	True	False	False	True	False	False	Fal
4	Maruti	120000	130000	False	False	False	True	True	False	False	Fal
...	...	...	...	...	...	...	...	...	...	...	
8123	Hyundai	110000	320000	False	False	False	True	True	False	False	Fal
8124	Hyundai	119000	135000	False	True	False	False	False	True	False	Fal
8125	Maruti	120000	382000	False	True	False	False	True	False	False	Fal
8126	Tata	25000	290000	False	True	False	False	True	False	False	Fal
8127	Tata	25000	290000	False	True	False	False	True	False	False	Fal

8128 rows × 12 columns

2. K-1 OneHotEncoding

```
pd.get_dummies(df,columns=['fuel','owner'],drop_first=True)
```

	brand	km_driven	selling_price	fuel_Diesel	fuel_LPG	fuel_Petrol	owner_Fourth & Above Owner	owner_Second Owner	owner_Test Drive Car	owner_Third Owner
0	Maruti	145500	450000	True	False	False	False	False	False	False
1	Skoda	120000	370000	True	False	False	False	True	False	False
2	Honda	140000	158000	False	False	True	False	False	False	True
3	Hyundai	127000	225000	True	False	False	False	False	False	False
4	Maruti	120000	130000	False	False	True	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...
8123	Hyundai	110000	320000	False	False	True	False	False	False	False
8124	Hyundai	119000	135000	True	False	False	True	False	False	False
8125	Maruti	120000	382000	True	False	False	False	False	False	False
8126	Tata	25000	290000	True	False	False	False	False	False	False
8127	Tata	25000	290000	True	False	False	False	False	False	False

Becaue during ML project we does not use pandas because it does not remember the coloumn name which we skip or use.So during ml project we use sklearn

3. OneHotEncoding using Sklearn

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(df.iloc[:,0:4],df.iloc[:,-1],test_size=0.2,random_state=2)
```

```
X_train.head()
```

	brand	km_driven	fuel	owner
5571	Hyundai	35000	Diesel	First Owner
2038	Jeep	60000	Diesel	First Owner
2957	Hyundai	25000	Petrol	First Owner
7618	Mahindra	130000	Diesel	Second Owner
6684	Hyundai	155000	Diesel	First Owner

Next steps: Generate code with X_train | View recommended plots | New interactive sheet

```
from sklearn.preprocessing import OneHotEncoder
```

```
ohe = OneHotEncoder(drop='first',sparse=False,dtype=np.int32)
# ohe = OneHotEncoder()
```

```
X_train_new = ohe.fit_transform(X_train[['fuel','owner']])
X_train_new
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:975: FutureWarning: `sparse` was renamed to `sparse_output`
  warnings.warn(
array([[1, 0, 0, ..., 0, 0, 0],
       [1, 0, 0, ..., 0, 0, 0],
       [0, 0, 1, ..., 0, 0, 0],
       ...,
       [0, 0, 1, ..., 0, 0, 0],
       [1, 0, 0, ..., 1, 0, 0],
       [1, 0, 0, ..., 0, 0, 0]], dtype=int32)
```

```
X_test_new = ohe.transform(X_test[['fuel','owner']])
```

```
X_train_new.shape
```

```
(6502, 7)
```

```
np.hstack((X_train[['brand','km_driven']].values,X_train_new))
```

```
array([['Hyundai', 35000, 1, ..., 0, 0, 0],
       ['Jeep', 60000, 1, ..., 0, 0, 0],
       ['Hyundai', 25000, 0, ..., 0, 0, 0],
```