

```
In [1]: import pandas as pd
data=pd.read_csv("/home/placement/Downloads/fiat500.csv")
print(data)
```

	ID	model	engine_power	age_in_days	km	previous_owners	\
0	1	lounge	51	882	25000	1	
1	2	pop	51	1186	32500	1	
2	3	sport	74	4658	142228	1	
3	4	lounge	51	2739	160000	1	
4	5	pop	73	3074	106880	1	
...	...	...	...	...	...	...	...
1533	1534	sport	51	3712	115280	1	
1534	1535	lounge	74	3835	112000	1	
1535	1536	pop	51	2223	60457	1	
1536	1537	lounge	51	2557	80750	1	
1537	1538	pop	51	1766	54276	1	

	lat	lon	price
0	44.907242	8.611560	8900
1	45.666359	12.241890	8800
2	45.503300	11.417840	4200
3	40.633171	17.634609	6000
4	41.903221	12.495650	5700
...	...	...	...
1533	45.069679	7.704920	5200
1534	45.845692	8.666870	4600
1535	45.481541	9.413480	7500
1536	45.000702	7.682270	5990
1537	40.323410	17.568270	7900

[1538 rows x 9 columns]

## loc[]

```
In [2]: data1=data.loc[(data.km<=50000)]
```

In [3]: data1

Out[3]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon	price
0	1	lounge	51	882	25000	1	44.907242	8.61156	8900
1	2	pop	51	1186	32500	1	45.666359	12.24189	8800
6	7	lounge	51	731	11600	1	44.907242	8.61156	10750
7	8	lounge	51	1521	49076	1	41.903221	12.49565	9190
10	11	pop	51	790	43286	1	40.871429	14.43896	8950
...	...	...	...	...	...	...	...	...	...
1525	1526	lounge	51	790	41870	1	45.707249	11.47760	9500
1526	1527	lounge	51	1705	23600	1	38.122070	13.36112	9300
1527	1528	pop	51	517	3000	1	40.748241	14.52835	9999
1529	1530	lounge	51	731	22551	1	38.122070	13.36112	9900
1530	1531	lounge	51	670	29000	1	45.764648	8.99450	10800

907 rows × 9 columns

## groupby()

In [4]: data2=data1.groupby(["model"]).count()

```
In [5]: data2
```

```
Out[5]:
```

	ID	engine_power	age_in_days	km	previous_owners	lat	lon	price
<hr/>								
model								
<b>lounge</b>	734	734	734	734	734	734	734	734
<b>pop</b>	162	162	162	162	162	162	162	162
<b>sport</b>	11	11	11	11	11	11	11	11

## rename the column

```
In [6]: data3=data.rename(columns={'model':'model_name'})
```

```
In [7]: list(data3)
```

```
Out[7]: ['ID',  
         'model_name',  
         'engine_power',  
         'age_in_days',  
         'km',  
         'previous_owners',  
         'lat',  
         'lon',  
         'price']
```

## drop the column

```
In [8]: data3.drop(columns='lon')
```

```
Out[8]:
```

	ID	model_name	engine_power	age_in_days	km	previous_owners	lat	price
0	1	lounge	51	882	25000	1	44.907242	8900
1	2	pop	51	1186	32500	1	45.666359	8800
2	3	sport	74	4658	142228	1	45.503300	4200
3	4	lounge	51	2739	160000	1	40.633171	6000
4	5	pop	73	3074	106880	1	41.903221	5700
...	...	...	...	...	...	...	...	...
1533	1534	sport	51	3712	115280	1	45.069679	5200
1534	1535	lounge	74	3835	112000	1	45.845692	4600
1535	1536	pop	51	2223	60457	1	45.481541	7500
1536	1537	lounge	51	2557	80750	1	45.000702	5990
1537	1538	pop	51	1766	54276	1	40.323410	7900

1538 rows × 8 columns

In [9]: data3

Out[9]:

	ID	model_name	engine_power	age_in_days	km	previous_owners	lat	lon	price
0	1	lounge	51	882	25000	1	44.907242	8.611560	8900
1	2	pop	51	1186	32500	1	45.666359	12.241890	8800
2	3	sport	74	4658	142228	1	45.503300	11.417840	4200
3	4	lounge	51	2739	160000	1	40.633171	17.634609	6000
4	5	pop	73	3074	106880	1	41.903221	12.495650	5700
...	...	...	...	...	...	...	...	...	...
1533	1534	sport	51	3712	115280	1	45.069679	7.704920	5200
1534	1535	lounge	74	3835	112000	1	45.845692	8.666870	4600
1535	1536	pop	51	2223	60457	1	45.481541	9.413480	7500
1536	1537	lounge	51	2557	80750	1	45.000702	7.682270	5990
1537	1538	pop	51	1766	54276	1	40.323410	17.568270	7900

1538 rows × 9 columns

In [10]: data4=data3.drop(columns='lat')*#drop the 'lat' column*

In [11]: data4

Out[11]:

	ID	model_name	engine_power	age_in_days	km	previous_owners	lon	price
0	1	lounge	51	882	25000	1	8.611560	8900
1	2	pop	51	1186	32500	1	12.241890	8800
2	3	sport	74	4658	142228	1	11.417840	4200
3	4	lounge	51	2739	160000	1	17.634609	6000
4	5	pop	73	3074	106880	1	12.495650	5700
...	...	...	...	...	...	...	...	...
1533	1534	sport	51	3712	115280	1	7.704920	5200
1534	1535	lounge	74	3835	112000	1	8.666870	4600
1535	1536	pop	51	2223	60457	1	9.413480	7500
1536	1537	lounge	51	2557	80750	1	7.682270	5990
1537	1538	pop	51	1766	54276	1	17.568270	7900

1538 rows × 8 columns

In [12]: data.groupby(['model']).count() *#count the each model using groupby function*

Out[12]:

	ID	engine_power	age_in_days	km	previous_owners	lat	lon	price
<b>model</b>								
<b>lounge</b>	1094	1094	1094	1094	1094	1094	1094	1094
<b>pop</b>	358	358	358	358	358	358	358	358
<b>sport</b>	86	86	86	86	86	86	86	86

```
In [13]: data.sample(5)
```

```
Out[13]:
```

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon	price
<b>821</b>	822	lounge	51	1127	37000	1	40.672379	14.728220	10500
<b>807</b>	808	lounge	51	701	19000	1	40.672379	14.728220	9999
<b>1395</b>	1396	lounge	51	456	20941	1	39.291222	16.252514	9700
<b>1121</b>	1122	lounge	51	1035	25000	1	41.055920	14.299440	9500
<b>928</b>	929	pop	51	4627	148000	1	45.356602	9.203450	3500

```
In [19]: data1=data.drop(columns=['model'])
```

```
In [20]: data1
```

```
Out[20]:
```

	ID	engine_power	age_in_days	km	previous_owners	lat	lon	price
<b>0</b>	1	51	882	25000	1	44.907242	8.611560	8900
<b>1</b>	2	51	1186	32500	1	45.666359	12.241890	8800
<b>2</b>	3	74	4658	142228	1	45.503300	11.417840	4200
<b>3</b>	4	51	2739	160000	1	40.633171	17.634609	6000
<b>4</b>	5	73	3074	106880	1	41.903221	12.495650	5700
...	...	...	...	...	...	...	...	...
<b>1533</b>	1534	51	3712	115280	1	45.069679	7.704920	5200
<b>1534</b>	1535	74	3835	112000	1	45.845692	8.666870	4600
<b>1535</b>	1536	51	2223	60457	1	45.481541	9.413480	7500
<b>1536</b>	1537	51	2557	80750	1	45.000702	7.682270	5990
<b>1537</b>	1538	51	1766	54276	1	40.323410	17.568270	7900

1538 rows × 8 columns

## correlation and Heatmap

```
In [21]: cor=data1.corr()
```

```
In [22]: cor
```

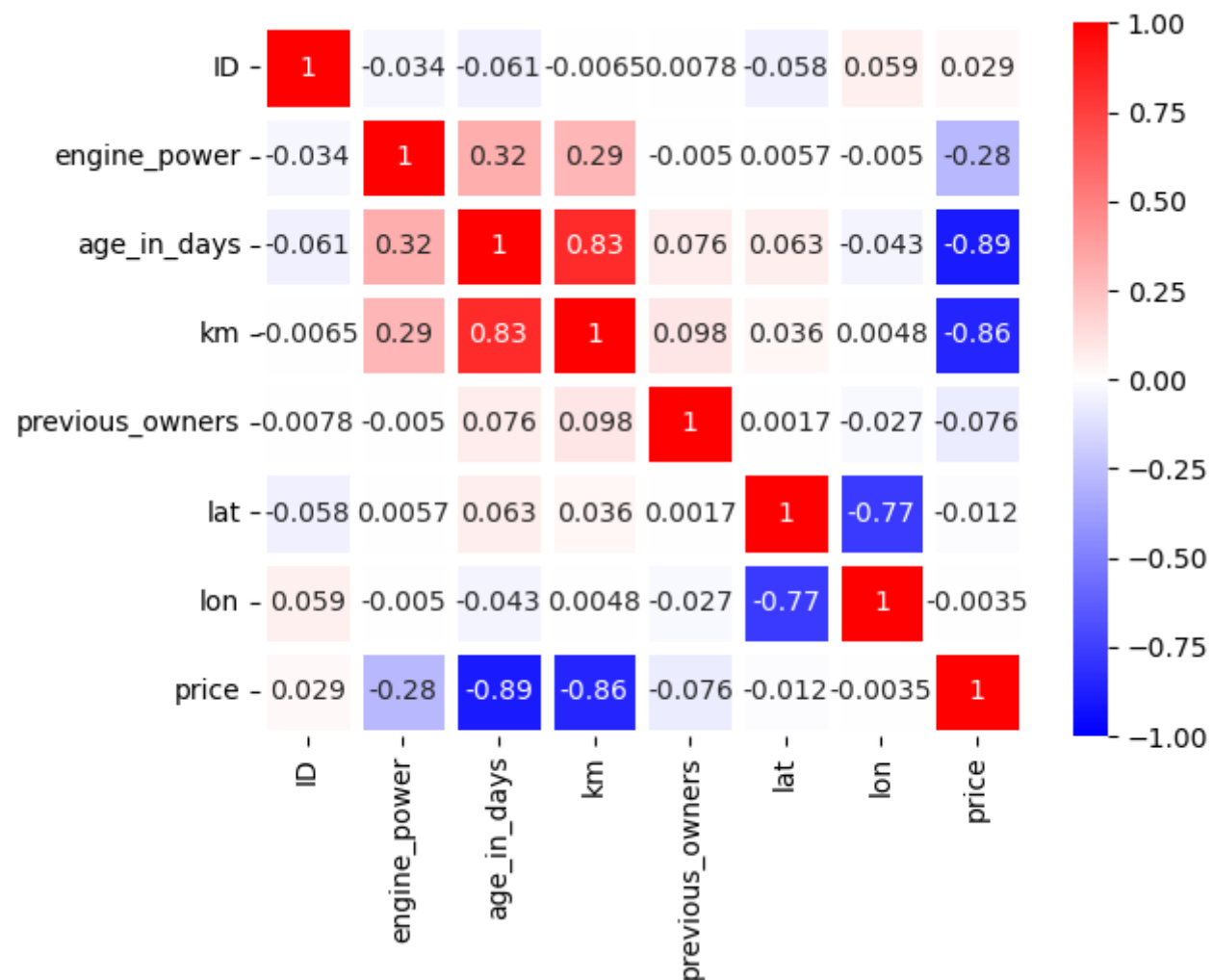
```
Out[22]:
```

	ID	engine_power	age_in_days	km	previous_owners	lat	lon	price
ID	1.000000	-0.034059	-0.060753	-0.006537	0.007803	-0.058207	0.058941	0.028516
engine_power	-0.034059	1.000000	0.319190	0.285495	-0.005030	0.005721	-0.005032	-0.277235
age_in_days	-0.060753	0.319190	1.000000	0.833890	0.075775	0.062982	-0.042667	-0.893328
km	-0.006537	0.285495	0.833890	1.000000	0.097539	0.035519	0.004839	-0.859373
previous_owners	0.007803	-0.005030	0.075775	0.097539	1.000000	0.001697	-0.026836	-0.076274
lat	-0.058207	0.005721	0.062982	0.035519	0.001697	1.000000	-0.766646	-0.011733
lon	0.058941	-0.005032	-0.042667	0.004839	-0.026836	-0.766646	1.000000	-0.003541
price	0.028516	-0.277235	-0.893328	-0.859373	-0.076274	-0.011733	-0.003541	1.000000



```
In [23]: import seaborn as sns
sns.heatmap(cor, vmax=1, vmin=-1, annot=True, linewidth=5, cmap='bwr')
```

Out[23]: <Axes: >



In [ ]: