Basic Analysis using Numpy and Pandas

import libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as pp
```

import dataset

In [2]:
```
data=pd.read_csv(r"E:\154\fiat500_VehicleSelection_Dataset - fiat500_VehicleSe
```

In [3]:
```
display(data)
```

	ID	model	engine_power	age_in_days	km	previous_owners	lat	l
0	1.0	lounge	51.0	882.0	25000.0	1.0	44.907242	8.6115598
1	2.0	pop	51.0	1186.0	32500.0	1.0	45.666359	12.241889
2	3.0	sport	74.0	4658.0	142228.0	1.0	45.503300	11.417
3	4.0	lounge	51.0	2739.0	160000.0	1.0	40.633171	17.634609
4	5.0	pop	73.0	3074.0	106880.0	1.0	41.903221	12.495650
...	...	...	...	...	...	...	...	
1544	NaN	NaN	NaN	NaN	NaN	NaN	NaN	leng
1545	NaN	NaN	NaN	NaN	NaN	NaN	NaN	con
1546	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Null valu
1547	NaN	NaN	NaN	NaN	NaN	NaN	NaN	fi
1548	NaN	NaN	NaN	NaN	NaN	NaN	NaN	sear

1549 rows × 11 columns

To display top 10 rows

In [4]:
```
data.head()
```

Out[4]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon
0	1.0	lounge	51.0	882.0	25000.0	1.0	44.907242	8.611559868
1	2.0	pop	51.0	1186.0	32500.0	1.0	45.666359	12.24188995
2	3.0	sport	74.0	4658.0	142228.0	1.0	45.503300	11.41784
3	4.0	lounge	51.0	2739.0	160000.0	1.0	40.633171	17.63460922
4	5.0	pop	73.0	3074.0	106880.0	1.0	41.903221	12.49565029

To display last 5 rows

In [5]:
```
data.tail()
```

Out[5]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon	price	Ur
1544	NaN	NaN	NaN	NaN	NaN	NaN	NaN	length	5	
1545	NaN	NaN	NaN	NaN	NaN	NaN	NaN	concat	lonprice	
1546	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Null values	NO	
1547	NaN	NaN	NaN	NaN	NaN	NaN	NaN	find	1	
1548	NaN	NaN	NaN	NaN	NaN	NaN	NaN	search	1	

In [6]:
```
data.dtypes
```

Out[6]:
```
ID                float64
model              object
engine_power      float64
age_in_days       float64
km                float64
previous_owners   float64
lat               float64
lon                object
price              object
Unnamed: 9        float64
Unnamed: 10        object
dtype: object
```

To view statistical summary

In [7]:
```
data.describe()
```

Out[7]:

	ID	engine_power	age_in_days	km	previous_owners	lat	l
count	1538.000000	1538.000000	1538.000000	1538.000000	1538.000000	1538.000000	
mean	769.500000	51.904421	1650.980494	53396.011704	1.123537	43.541361	
std	444.126671	3.988023	1289.522278	40046.830723	0.416423	2.133518	
min	1.000000	51.000000	366.000000	1232.000000	1.000000	36.855839	
25%	385.250000	51.000000	670.000000	20006.250000	1.000000	41.802990	
50%	769.500000	51.000000	1035.000000	39031.000000	1.000000	44.394096	
75%	1153.750000	51.000000	2616.000000	79667.750000	1.000000	45.467960	
max	1538.000000	77.000000	4658.000000	235000.000000	4.000000	46.795612	

To Print no of elements

In [8]:
```
data.size
```

Out[8]:
```
17039
```

In [9]:
```
data.ndim
```

Out[9]:
```
2
```

To print no of rows and columns

In [10]:
```
data.shape
```

Out[10]:
```
(1549, 11)
```

To find missing values

In [11]:
```
data.isna()
```

Out[11]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon	price	Un
0	False	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	...	
1544	True	True	True	True	True	True	True	False	False	
1545	True	True	True	True	True	True	True	False	False	
1546	True	True	True	True	True	True	True	False	False	
1547	True	True	True	True	True	True	True	False	False	
1548	True	True	True	True	True	True	True	False	False	

1549 rows × 11 columns

To fill nulll values with constatns

In [12]:
```
data.fillna(5)
```

Out[12]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lo
0	1.0	lounge	51.0	882.0	25000.0	1.0	44.907242	8.61155986
1	2.0	pop	51.0	1186.0	32500.0	1.0	45.666359	12.2418899
2	3.0	sport	74.0	4658.0	142228.0	1.0	45.503300	11.4178
3	4.0	lounge	51.0	2739.0	160000.0	1.0	40.633171	17.6346092
4	5.0	pop	73.0	3074.0	106880.0	1.0	41.903221	12.4956502
...	...	...	...	...	...	...	...	.
1544	5.0	5	5.0	5.0	5.0	5.0	5.000000	lengt
1545	5.0	5	5.0	5.0	5.0	5.0	5.000000	conca
1546	5.0	5	5.0	5.0	5.0	5.0	5.000000	Null value
1547	5.0	5	5.0	5.0	5.0	5.0	5.000000	fin
1548	5.0	5	5.0	5.0	5.0	5.0	5.000000	searc

1549 rows × 11 columns

In [13]:
```
data=data[['km','price']]
```

In [14]:
```
data
```

Out[14]:

	km	price
0	25000.0	8900
1	32500.0	8800
2	142228.0	4200
3	160000.0	6000
4	106880.0	5700
...	...	...
1544	NaN	5
1545	NaN	lonprice
1546	NaN	NO
1547	NaN	1
1548	NaN	1

1549 rows × 2 columns

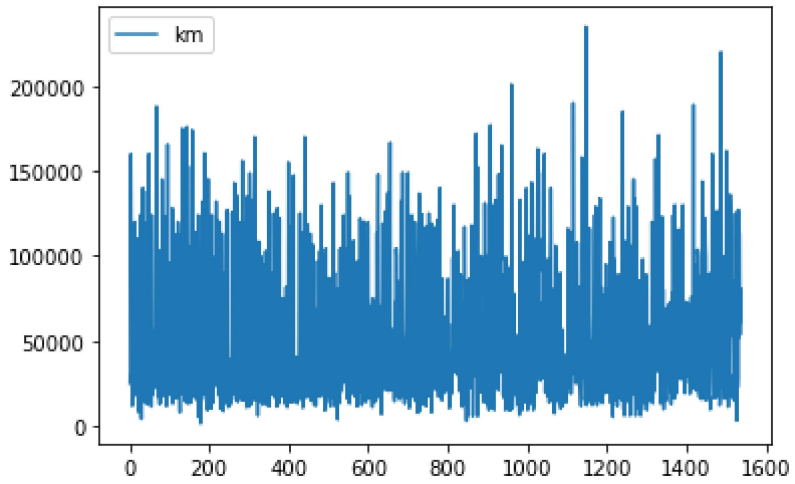
Type *Markdown* and LaTeX: α^2

In [15]:
```
data=data[['km','price']]
```

Line Chart

In [22]:
```
data.plot.line()
```

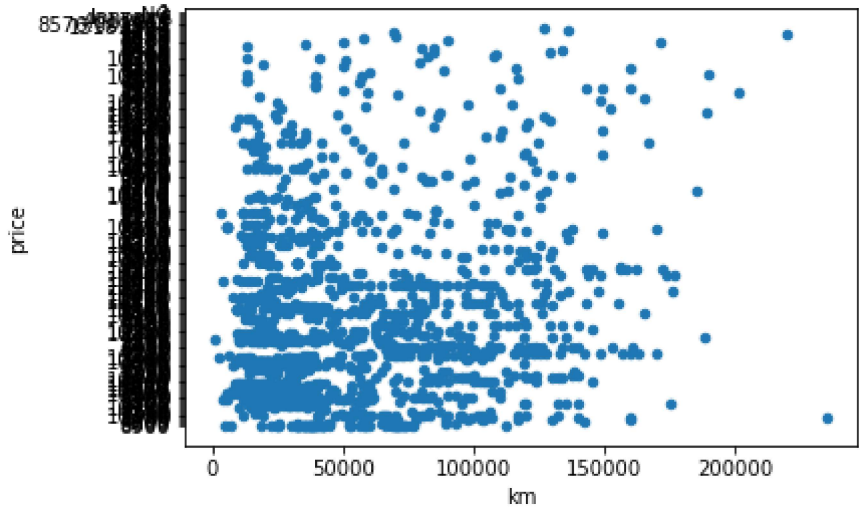
Out[22]: <AxesSubplot:>



Scatter Plot

In [27]:
```
data.plot.scatter(x='km',y='price')
```

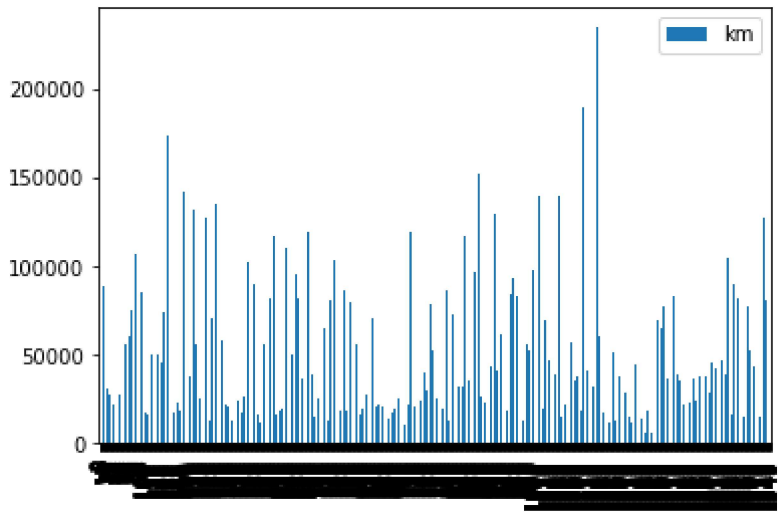
Out[27]: <AxesSubplot:xlabel='km', ylabel='price'>



Bar Chart

In [30]:
```
data.plot.bar()
```

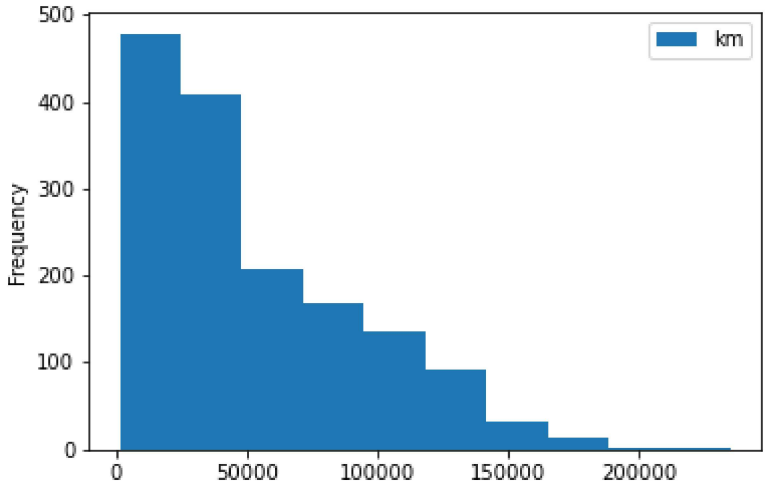
Out[30]: <AxesSubplot:>



Histogram

In [31]:
```
data.plot.hist()
```

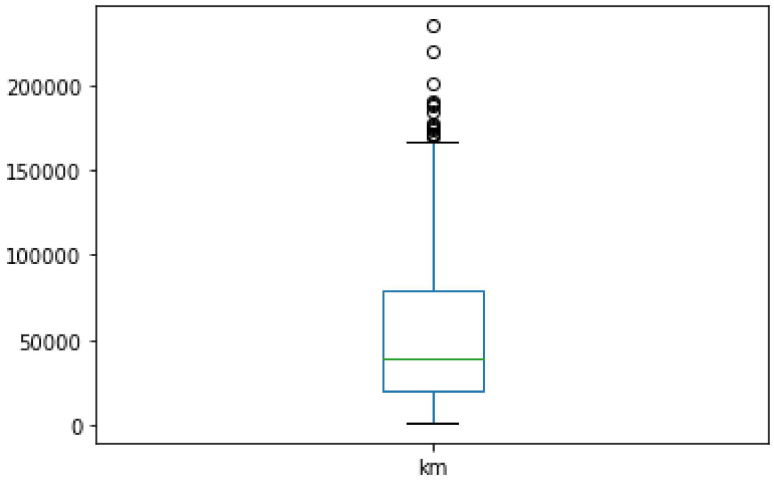
Out[31]: <AxesSubplot:ylabel='Frequency'>



Box Plot

In [33]:
```
data.plot.box()
```

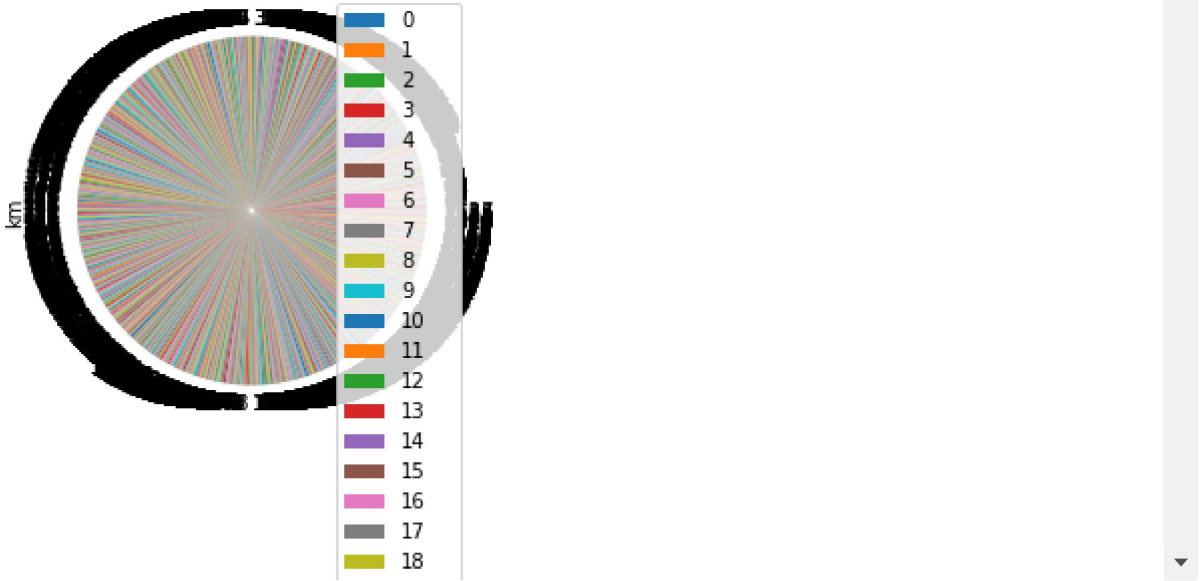
Out[33]: <AxesSubplot:>



Pie Chart

In [35]:
```
data.plot.pie(y="km")
```

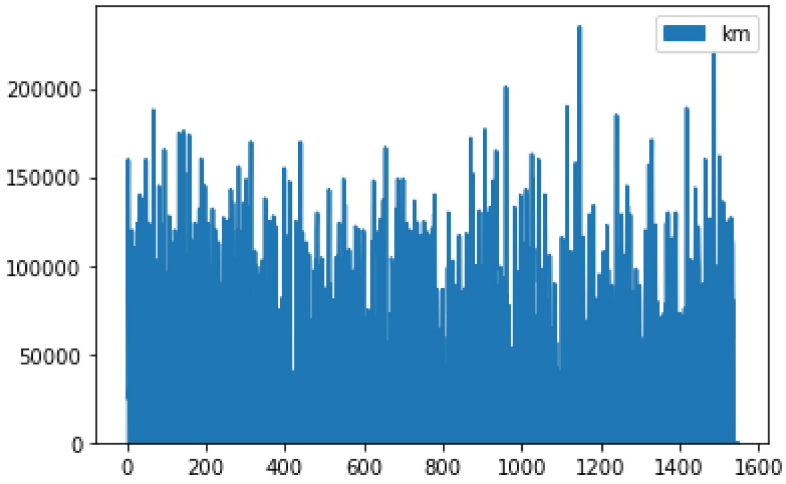
Out[35]: <AxesSubplot:ylabel='km'>



Area

In [36]:
```
data.plot.area()
```

Out[36]: <AxesSubplot:>



In [ ]:
```

```