

```
In [1]: import pandas as pd

In [2]: home = pd.read_csv("pract.csv")

In [3]: home

Out[3]:
```

	make	colour	odometer	door	price
0	Toyota	Red	239970	3	\$,2000
1	BMW	Blue	67356	4	\$,7,000
2	Honda	Yellow	7734008	2	\$6,000
3	Benz	Green	9354	3	\$5,000

```
In [4]: pd.crosstab(home["make"], home["door"])

Out[4]:
```

	door	2	3	4
make				
BMW	0	0	1	
Benz	0	1	0	
Honda	1	0	0	
Toyota	0	1	0	

```
In [6]: %matplotlib inline
import matplotlib.pyplot as plt

In [7]: home

Out[7]:
```

	make	colour	odometer	door	price
0	Toyota	Red	239970	3	\$,2000
1	BMW	Blue	67356	4	\$,7,000
2	Honda	Yellow	7734008	2	\$6,000
3	Benz	Green	9354	3	\$5,000

Manipulating DATA

```
In [8]: home["make"].str.lower()

Out[8]:
```

0	toyota
1	bmw
2	honda
3	benz

Name: make, dtype: object

```
In [9]: home["make"] = home["make"].str.lower()
home

Out[9]:
```

	make	colour	odometer	door	price
0	toyota	Red	239970	3	\$,2000
1	bmw	Blue	67356	4	\$,7,000
2	honda	Yellow	7734008	2	\$6,000
3	benz	Green	9354	3	\$5,000

```
In [10]: # panda represent the missing data with NaN
home_missing = pd.read_csv("pract3.csv")
home_missing

Out[10]:
```

	make	colour	odometer	door	price
0	Toyota	Red	239970.0	3	\$,2000
1	BMW	Blue	67356.0	4	\$,7,000
2	Honda	Yellow	NaN	2	\$6,000
3	Benz	Green	9354.0	3	NaN

```
In [11]: home["odometer"]

Out[11]:
```

0	239970
1	67356
2	7734008
3	9354

Name: odometer, dtype: int64

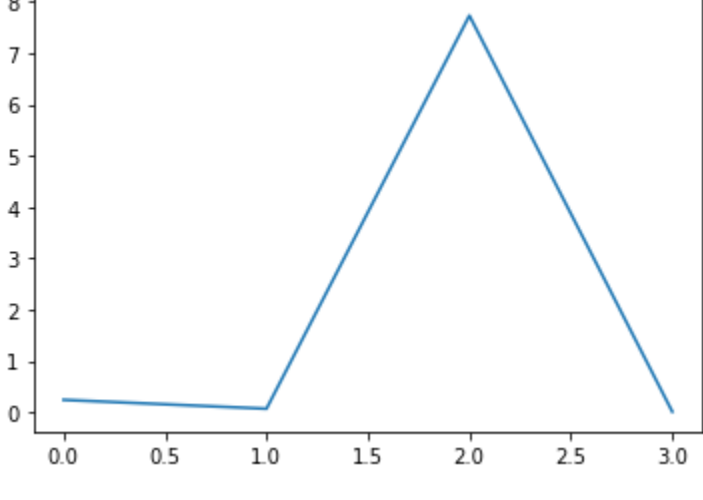
```
In [13]: home.groupby(["make"]).mean()

Out[13]:
```

	odometer	door
make		
benz	9354.0	3.0
bmw	67356.0	4.0
honda	7734008.0	2.0
toyota	239970.0	3.0

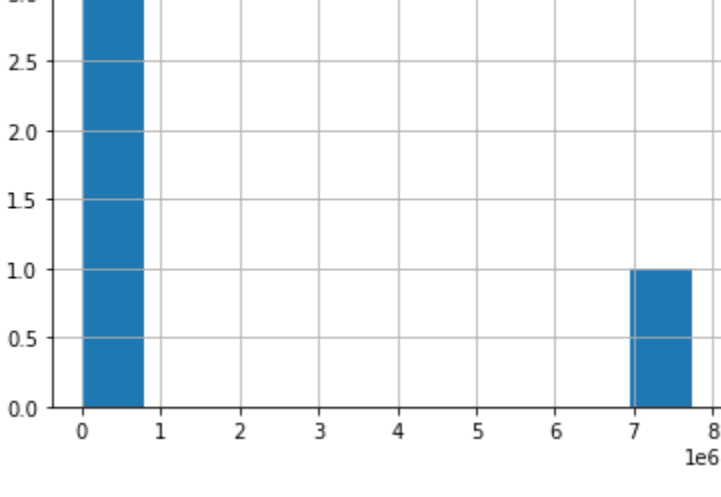
```
In [22]: home["odometer"].plot()

Out[22]: <AxesSubplot:~>
```



```
In [23]: home["odometer"].hist()

Out[23]: <AxesSubplot:~>
```



```
In [24]: home["make"].str.upper()

Out[24]:
```

0	TOYOTA
1	BMW
2	HONDA
3	BENZ

Name: make, dtype: object

```
In [26]: home["make"] = home["make"].str.upper()
home

Out[26]:
```

	make	colour	odometer	door	price
0	TOYOTA	Red	239970	3	\$,2000
1	BMW	Blue	67356	4	\$,7,000
2	HONDA	Yellow	7734008	2	\$6,000
3	BENZ	Green	9354	3	\$5,000

```
In [27]: # panda represent the missing data with NaN
home_missing = pd.read_csv("pract3.csv")
home_missing

Out[27]:
```

	make	colour	odometer	door	price
0	Toyota	Red	239970.0	3	\$,2000
1	BMW	Blue	67356.0	4	\$,7,000
2	Honda	Yellow	NaN	2	\$6,000
3	Benz	Green	9354.0	3	NaN

```
In [28]: #this is to fill in the missing data in "odometer"
home_missing["odometer"].fillna(home_missing["odometer"].mean())

Out[28]:
```

0	239970.0
1	67356.0
2	105560.0
3	9354.0

Name: odometer, dtype: float64

```
In [31]: #this is to fill in the missing data in "odometer"
home_missing["odometer"].fillna(home_missing["odometer"].mean(),inplace=True)
home

Out[31]:
```

	make	colour	odometer	door	price
0	TOYOTA	Red	239970	3	\$,2000
1	BMW	Blue	67356	4	\$,7,000
2	HONDA	Yellow	7734008	2	\$6,000
3	BENZ	Green	9354	3	\$5,000

```
In [32]: home_missing.dropna()

Out[32]:
```

	make	colour	odometer	door	price
0	Toyota	Red	239970.0	3	\$,2000
1	BMW	Blue	67356.0	4	\$,7,000
2	Honda	Yellow	105560.0	2	\$6,000

```
In [34]: home_missing.dropna(inplace=True)
home_missing

Out[34]:
```

	make	colour	odometer	door	price
0	Toyota	Red	239970.0	3	\$,2000
1	BMW	Blue	67356.0	4	\$,7,000
2	Honda	Yellow	105560.0	2	\$6,000

```
In [35]: home_missing.dropna(inplace=False)
home_missing

Out[35]:
```

	make	colour	odometer	door	price
0	Toyota	Red	239970.0	3	\$,2000
1	BMW	Blue	67356.0	4	\$,7,000
2	Honda	Yellow	105560.0	2	\$6,000

```
In [36]: #column from series
model_column = pd.Series([5,5,5])
#new column called model
home["model"] = model_column
home

Out[36]:
```

	make	colour	odometer	door	price	model
0	TOYOTA	Red	239970	3	\$,2000	5.0
1	BMW	Blue	67356	4	\$,7,000	5.0
2	HONDA	Yellow	7734008	2	\$6,000	5.0
3	BENZ	Green	9354	3	\$5,000	NaN

```
In [44]: home["model"].fillna(5,inplace=True)
home

Out[44]:
```

	make	colour	odometer	door	price	model
0	TOYOTA	Red	239970	3	\$,2000	5.0
1	BMW	Blue	67356	4	\$,7,000	5.0
2	HONDA	Yellow	7734008	2	\$6,000	5.0
3	BENZ	Green	9354	3	\$5,000	5.0

```
In [45]: # column from python list
brand = [2022,2021,2020,2019]
home["brand"] = brand
home

Out[45]:
```

	make	colour	odometer	door	price	model	brand
0	TOYOTA	Red	239970	3	\$,2000	5.0	2022
1	BMW	Blue	67356	4	\$,7,000	5.0	2021
2	HONDA	Yellow	7734008	2	\$6,000	5.0	2020
3	BENZ	Green	9354	3	\$5,000	5.0	2019

```
In [48]: #to create a new column from an existing one.
home["total door (T)"] = home["door"]/100*home["model"]
home

Out[48]:
```

	make	colour	odometer	door	price	model	brand	total door	total door (T)	unmber of wheels	passed road safty
0	TOYOTA	Red	239970	3	\$,2000	5.0	2022	0.15	0.15	4	True
1	BMW	Blue	67356	4	\$,7,000	5.0	2021	0.20	0.20	4	True
2	HONDA	Yellow	7734008	2	\$6,000	5.0	2020	0.10	0.10	4	True
3	BENZ	Green	9354	3	\$5,000	5.0	2019	0.15	0.15	4	True

```
In [50]: #CREATE A COLUMN FROM A SINGLE COLUMN
home["unmber of wheels"] = 4
home

Out[50]:
```

	make	colour	odometer	door	price	model	brand	total door	total door (T)	unmber of wheels	passed road safty
0	TOYOTA	Red	239970	3	\$,2000	5.0	2022	0.15	0.15	4	4
1	BMW	Blue	67356	4	\$,7,000	5.0	2021	0.20	0.20	4	4
2	HONDA	Yellow	7734008	2	\$6,000	5.0	2020	0.10	0.10	4	4
3	BENZ	Green	9354	3	\$5,000	5.0	2019	0.15	0.15	4	4

```
In [51]: home["passed road safty"] = True
home.dtypes

Out[51]:
```

make	object
colour	object
odometer	int64
door	int64
price	object
model	float64
brand	int64
total door	float64
total door (T)	float64
modela	int64
unmber of wheels	int64
passed road safty	bool
dtype:	object

```
In [52]: home

Out[52]:
```

	make	colour	odometer	door	price	model	brand	total door	total door (T)	modela	unmber of wheels	passed road safty
0	TOYOTA	Red	239970	3	\$,2000	5.0	2022	0.15	0.15	4	4	True
1	BMW	Blue	67356	4	\$,7,000	5.0	2021	0.20	0.20	4	4	True
2	HONDA	Yellow	7734008	2	\$6,000	5.0	2020	0.10	0.10	4	4	True
3	BENZ	Green	9354	3	\$5,000	5.0	2019	0.15	0.15	4	4	True

```
In [53]: how to drop a column#
home=home.drop("modela",axis=1)
home

Out[53]:
```

	make	colour	odometer	door	price	model	brand	total door	total door (T)	unmber of wheels	passed road safty
0	TOYOTA	Red	239970	3	\$,2000	5.0	2022	0.15	0.15	4	True
1	BMW	Blue	67356	4	\$,7,000	5.0	2021	0.20	0.20	4	True
2	HONDA	Yellow	7734008	2	\$6,000	5.0	2020	0.10	0.10	4	True
3	BENZ	Green	9354	3	\$5,000	5.0	2019	0.15	0.15	4	True

```
In [74]: #this means to show some percentage of the data frame(0.5 50% of the data then 1 for 100%)
home_shuffled=home.sample(frac=1)
home_shuffled

Out[74]:
```

	make	colour	odometer	door	price	model	brand	total door	total door (T)	unmber of wheels	passed road safty
3	BENZ	Green	9354	3	\$5,000	5.0	2019	0.15	0.15	4	True
1	BMW	Blue	67356	4	\$,7,000	5.0	2021	0.20	0.20	4	True
0	TOYOTA	Red	239970	3	\$,2000	5.0	2022	0.15	0.15	4	True
2	HONDA	Yellow	7734008	2	\$6,000	5.0	2020	0.10	0.10	4	True

```
In [75]: home

Out[75]:
```

	make	colour	odometer	door	price	model	brand	total door	total door (T)	unmber of wheels	passed road safty
0	TOYOTA	Red	239970	3	\$,2000	5.0	2022	0.15	0.15	4	True
1	BMW	Blue	67356	4	\$,7,000	5.0	2021	0.20	0.20	4	True
2	HONDA	Yellow	7734008	2	\$6,000	5.0	2020	0.10	0.10	4	True
3	BENZ	Green	9354	3	\$5,000	5.0	2019	0.15	0.15	4	True

```
In [76]: #to get your data frame to normal
home_shuffled.reset_index()

Out[76]:
```

	index	make	colour	odometer	door	price	model	brand	total door	total door (T)	unmber of wheels	passed road safty
0	3	BENZ	Green	9354	3	\$5,000	5.0	2019	0.15	0.15	4	True
1	1	BMW	Blue	67356	4	\$,7,000	5.0	2021	0.20	0.20	4	True
2	0	TOYOTA	Red	239970	3	\$,2000	5.0	2022	0.15	0.15	4	True
3	2	HONDA	Yellow	7734008	2	\$6,000	5.0	2020	0.10	0.10	4	True

```
In [77]: home_shuffled.reset_index(drop=True, inplace=True)
home_shuffled

Out[77]:
```

	make	colour	odometer	door	price	model	brand	total door	total door (T)	unmber of wheels	passed road safty
0	BENZ	Green	9354	3	\$5,000	5.0	2019	0.15	0.15	4	True
1	BMW	Blue	67356	4	\$,7,000	5.0	2021	0.20	0.20	4	True
2	TOYOTA	Red	239970	3	\$,2000	5.0	2022	0.15	0.15	4	True
3	HONDA	Yellow	7734008	2	\$6,000	5.0	2020	0.10	0.10	4	True

```
In [78]: #how to apply a function to a column, apply is a way to assign a function to a column
home["odometer"] = home["odometer"].apply(lambda x: x / 1.6)
home

Out[78]:
```

	make	colour	odometer	door	price	model	brand	total door	total door (T)	unmber of wheels	passed road safty
0	TOYOTA	Red	149981.25	3	\$,2000	5.0	2022	0.15	0.15	4	True
1	BMW	Blue	42097.50	4	\$,7,000	5.0	2021	0.20	0.20	4	True
2	HONDA	Yellow	4833755.00	2	\$6,000	5.0	2020	0.10	0.10	4	True
3	BENZ	Green	5846.25	3	\$5,000	5.0	2019	0.15	0.15	4	True

```
In [ ]: #try it, run your code
#search for it
#try again
#ask
```