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

In [2]:
```
#2 main datatype
Series=pd.Series(["bmw","tyota","honda"])
```

In [3]:
```
Series
```

Out[3]:
```
0      bmw
1    tyota
2    honda
dtype: object
```

In [4]:
```
#series=1 -dimensional
```

In [5]:
```
colours=pd.Series(["red","blue","pink"])
colours
```

Out[5]:
```
0     red
1    blue
2    pink
dtype: object
```

In [6]:
```
#dataframe=2- dimensional
car_data=pd.DataFrame({"car make" :Series,"colour":colours} )
car_data
```

Out[6]:

	car make	colour
0	bmw	red
1	tyota	blue
2	honda	pink

In [7]:
```
#import data
car_sales=pd.read_csv("car-sales-extended-missing-data.csv")
```

In [8]:
```
car_sales
```

Out[8]:

	Make	Colour	Odometer (KM)	Doors	Price
0	Honda	White	35431.0	4.0	15323.0
1	BMW	Blue	192714.0	5.0	19943.0
2	Honda	White	84714.0	4.0	28343.0
3	Toyota	White	154365.0	4.0	13434.0
4	Nissan	Blue	181577.0	3.0	14043.0
...	...	...	...	...	...
995	Toyota	Black	35820.0	4.0	32042.0
996	NaN	White	155144.0	3.0	5716.0
997	Nissan	Blue	66604.0	4.0	31570.0
998	Honda	White	215883.0	4.0	4001.0
999	Toyota	Blue	248360.0	4.0	12732.0

1000 rows × 5 columns

In [9]:
```
#exportinga dataframe
car_sales.to_csv("exported-car-sales.csv",index=False)
```

In [10]:
```
exported_car_sales=pd.read_csv("exported-car-sales.csv")
exported_car_sales
```

Out[10]:

	Make	Colour	Odometer (KM)	Doors	Price
0	Honda	White	35431.0	4.0	15323.0
1	BMW	Blue	192714.0	5.0	19943.0
2	Honda	White	84714.0	4.0	28343.0
3	Toyota	White	154365.0	4.0	13434.0
4	Nissan	Blue	181577.0	3.0	14043.0
...	...	...	...	...	...
995	Toyota	Black	35820.0	4.0	32042.0
996	NaN	White	155144.0	3.0	5716.0
997	Nissan	Blue	66604.0	4.0	31570.0
998	Honda	White	215883.0	4.0	4001.0
999	Toyota	Blue	248360.0	4.0	12732.0

1000 rows × 5 columns

describe data

In [11]:
```
#atrributes
#car_sales.dtypes
#function
car_sales.to_csv()
```

Out[11]:
```
',Make,Colour,Odometer (KM),Doors,Price\r\n0,Honda,White,35431.0,4.0,153
23.0\r\n1,BMW,Blue,192714.0,5.0,19943.0\r\n2,Honda,White,84714.0,4.0,283
43.0\r\n3,Toyota,White,154365.0,4.0,13434.0\r\n4,Nissan,Blue,181577.0,3.
0,14043.0\r\n5,Honda,Red,42652.0,4.0,23883.0\r\n6,Toyota,Blue,163453.0,
4.0,8473.0\r\n7,Honda,White,,4.0,20306.0\r\n8,,White,130538.0,4.0,9374.0
\r\n9,Honda,Blue,51029.0,4.0,26683.0\r\n10,Nissan,White,167421.0,4.0,162
59.0\r\n11,Nissan,Green,17119.0,4.0,6160.0\r\n12,Nissan,White,102303.0,
4.0,16909.0\r\n13,,White,134181.0,4.0,11121.0\r\n14,Honda,Blue,199833.0,
4.0,18946.0\r\n15,Toyota,Blue,205592.0,4.0,16290.0\r\n16,Toyota,Red,9674
2.0,4.0,34465.0\r\n17,BMW,White,194189.0,5.0,17177.0\r\n18,Nissan,White,
67991.0,3.0,9109.0\r\n19,Nissan,Blue,215820.0,4.0,6010.0\r\n20,Toyota,,1
24844.0,4.0,24130.0\r\n21,Honda,,30615.0,4.0,29653.0\r\n22,Toyota,White,
148744.0,4.0,22489.0\r\n23,Honda,Green,130075.0,4.0,21242.0\r\n24,Honda,
Blue,172718.0,4.0,14274.0\r\n25,Honda,Blue,125819.0,4.0,15686.0\r\n26,Ho
nda,White,180390.0,4.0,13344.0\r\n27,Honda,Green,82783.0,4.0,10984.0\r\n
28,Honda,White,56687.0,4.0,6135.0\r\n29,Toyota,White,112004.0,4.0,13586.
0\r\n30,Nissan,Blue,34024.0,4.0,23929.0\r\n31,Toyota,White,108569.0,,686
6.0\r\n32,Nissan,White,203795.0,4.0,9330.0\r\n33,Nissan,Green,153554.0,
3.0,9780.0\r\n34,Honda,Blue,71949.0,4.0,17516.0\r\n35,Honda,Blue,191622.
```

In [12]:
```
car_sales.columns
```

Out[12]:
```
Index(['Make', 'Colour', 'Odometer (KM)', 'Doors', 'Price'], dtype='objec
t')
```

In [13]:
```
car_columns=car_sales.columns
car_columns
```

Out[13]:
```
Index(['Make', 'Colour', 'Odometer (KM)', 'Doors', 'Price'], dtype='objec
t')
```

In [14]:
```
car_sales.index
```

Out[14]:
```
RangeIndex(start=0, stop=1000, step=1)
```

In [15]:
```
car_sales
```

Out[15]:

	Make	Colour	Odometer (KM)	Doors	Price
0	Honda	White	35431.0	4.0	15323.0
1	BMW	Blue	192714.0	5.0	19943.0
2	Honda	White	84714.0	4.0	28343.0
3	Toyota	White	154365.0	4.0	13434.0
4	Nissan	Blue	181577.0	3.0	14043.0
...	...	...	...	...	...
995	Toyota	Black	35820.0	4.0	32042.0
996	NaN	White	155144.0	3.0	5716.0
997	Nissan	Blue	66604.0	4.0	31570.0
998	Honda	White	215883.0	4.0	4001.0
999	Toyota	Blue	248360.0	4.0	12732.0

1000 rows × 5 columns

In [16]:
```
car_sales.describe()
```

Out[16]:

	Odometer (KM)	Doors	Price
count	950.000000	950.000000	950.000000
mean	131253.237895	4.011579	16042.814737
std	69094.857187	0.382539	8581.695036
min	10148.000000	3.000000	2796.000000
25%	70391.250000	4.000000	9529.250000
50%	131821.000000	4.000000	14297.000000
75%	192668.500000	4.000000	20806.250000
max	249860.000000	5.000000	52458.000000

In [17]:
```
car_sales.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Make           951 non-null    object 
 1   Colour         950 non-null    object 
 2   Odometer (KM)  950 non-null    float64
 3   Doors          950 non-null    float64
 4   Price          950 non-null    float64
dtypes: float64(3), object(2)
memory usage: 39.2+ KB
```

statical analysis

In [18]:
```
car_sales.mean()
```

```
C:\Users\satya shukla\AppData\Local\Temp\ipykernel_5572\4073448239.py:1: F
utureWarning: Dropping of nuisance columns in DataFrame reductions (with
'numeric_only=None') is deprecated; in a future version this will raise Ty
peError.  Select only valid columns before calling the reduction.
  car_sales.mean()
```

Out[18]:
```
Odometer (KM)    131253.237895
Doors                 4.011579
Price             16042.814737
dtype: float64
```

In [19]:
```
car_sales.sum
```

Out[19]:
```
<bound method NDFrame._add_numeric_operations.<locals>.sum of        Make
Colour  Odometer (KM)  Doors    Price
0     Honda  White        35431.0    4.0  15323.0
1       BMW   Blue       192714.0    5.0  19943.0
2     Honda  White        84714.0    4.0  28343.0
3    Toyota  White       154365.0    4.0  13434.0
4    Nissan   Blue       181577.0    3.0  14043.0
..      ...    ...            ...    ...      ...
995  Toyota  Black        35820.0    4.0  32042.0
996     NaN  White       155144.0    3.0   5716.0
997  Nissan   Blue        66604.0    4.0  31570.0
998   Honda  White       215883.0    4.0   4001.0
999  Toyota   Blue       248360.0    4.0  12732.0

[1000 rows x 5 columns]>
```

In [20]:
```
car_sales["Doors"].sum()
```

Out[20]:
```
3811.0
```

In [21]:
```
len(car_sales)
```

Out[21]:
```
1000
```

viewing and selecting data

In [22]:
```
car_sales.head()
```

Out[22]:

	Make	Colour	Odometer (KM)	Doors	Price
0	Honda	White	35431.0	4.0	15323.0
1	BMW	Blue	192714.0	5.0	19943.0
2	Honda	White	84714.0	4.0	28343.0
3	Toyota	White	154365.0	4.0	13434.0
4	Nissan	Blue	181577.0	3.0	14043.0

In [23]:
```
car_sales
```

Out[23]:

	Make	Colour	Odometer (KM)	Doors	Price
0	Honda	White	35431.0	4.0	15323.0
1	BMW	Blue	192714.0	5.0	19943.0
2	Honda	White	84714.0	4.0	28343.0
3	Toyota	White	154365.0	4.0	13434.0
4	Nissan	Blue	181577.0	3.0	14043.0
...	...	...	...	...	...
995	Toyota	Black	35820.0	4.0	32042.0
996	NaN	White	155144.0	3.0	5716.0
997	Nissan	Blue	66604.0	4.0	31570.0
998	Honda	White	215883.0	4.0	4001.0
999	Toyota	Blue	248360.0	4.0	12732.0

1000 rows × 5 columns

In [24]:
```
car_sales.head(10)
```

Out[24]:

	Make	Colour	Odometer (KM)	Doors	Price
0	Honda	White	35431.0	4.0	15323.0
1	BMW	Blue	192714.0	5.0	19943.0
2	Honda	White	84714.0	4.0	28343.0
3	Toyota	White	154365.0	4.0	13434.0
4	Nissan	Blue	181577.0	3.0	14043.0
5	Honda	Red	42652.0	4.0	23883.0
6	Toyota	Blue	163453.0	4.0	8473.0
7	Honda	White	NaN	4.0	20306.0
8	NaN	White	130538.0	4.0	9374.0
9	Honda	Blue	51029.0	4.0	26683.0

In [25]:
```
car_sales.tail(3)
```

Out[25]:

	Make	Colour	Odometer (KM)	Doors	Price
997	Nissan	Blue	66604.0	4.0	31570.0
998	Honda	White	215883.0	4.0	4001.0
999	Toyota	Blue	248360.0	4.0	12732.0

.loc and .iloc

In [26]:
```
animals=pd.Series(["cat","dog","bird","panda","snake"],
                  index=[0,3,9,8,3])
```

In [27]:
```
animals
```

Out[27]:
```
0      cat
3      dog
9     bird
8    panda
3    snake
dtype: object
```

In [28]:
```
animals.loc[3]
```

Out[28]:
```
3      dog
3    snake
dtype: object
```

In [29]:
```
animals.loc[9]
```

Out[29]:
```
'bird'
```

In [30]:
```
car_sales.loc[3]
```

Out[30]:
```
Make               Toyota
Colour              White
Odometer (KM)    154365.0
Doors                4.0
Price            13434.0
Name: 3, dtype: object
```

In [31]:
```
#.iloc refers to position
animals.iloc[3]
```

Out[31]:
```
'panda'
```

In [32]:
```
car_sales.iloc[3]
```

Out[32]:
```
Make               Toyota
Colour              White
Odometer (KM)    154365.0
Doors                4.0
Price            13434.0
Name: 3, dtype: object
```

In [33]:
```
animals.iloc[:3]
```

Out[33]:
```
0     cat
3     dog
9    bird
dtype: object
```

In [34]:
```
car_sales.head()
```

Out[34]:

	Make	Colour	Odometer (KM)	Doors	Price
0	Honda	White	35431.0	4.0	15323.0
1	BMW	Blue	192714.0	5.0	19943.0
2	Honda	White	84714.0	4.0	28343.0
3	Toyota	White	154365.0	4.0	13434.0
4	Nissan	Blue	181577.0	3.0	14043.0

In [35]:
```
car_sales["Make"]
```

Out[35]:
```
0       Honda
1         BMW
2       Honda
3      Toyota
4      Nissan
        ...
995    Toyota
996       NaN
997    Nissan
998     Honda
999    Toyota
Name: Make, Length: 1000, dtype: object
```

In [36]:
```
car_sales["Colour"]
```

Out[36]:
```
0      White
1       Blue
2      White
3      White
4       Blue
       ...
995    Black
996    White
997     Blue
998    White
999     Blue
Name: Colour, Length: 1000, dtype: object
```

In [37]:
```
car_sales[car_sales["Make"]=="Toyota"]
```

Out[37]:

	Make	Colour	Odometer (KM)	Doors	Price
3	Toyota	White	154365.0	4.0	13434.0
6	Toyota	Blue	163453.0	4.0	8473.0
15	Toyota	Blue	205592.0	4.0	16290.0
16	Toyota	Red	96742.0	4.0	34465.0
20	Toyota	NaN	124844.0	4.0	24130.0
...	...	...	...	...	...
983	Toyota	Red	NaN	4.0	14671.0
989	Toyota	Red	41735.0	4.0	13928.0
990	Toyota	White	173408.0	4.0	8082.0
995	Toyota	Black	35820.0	4.0	32042.0
999	Toyota	Blue	248360.0	4.0	12732.0

379 rows × 5 columns

In [38]:
```
car_sales[car_sales["Odometer (KM)"]>100000]
```

Out[38]:

	Make	Colour	Odometer (KM)	Doors	Price
1	BMW	Blue	192714.0	5.0	19943.0
3	Toyota	White	154365.0	4.0	13434.0
4	Nissan	Blue	181577.0	3.0	14043.0
6	Toyota	Blue	163453.0	4.0	8473.0
8	NaN	White	130538.0	4.0	9374.0
...	...	...	...	...	...
993	Nissan	Black	162523.0	4.0	4696.0
994	BMW	Blue	163322.0	3.0	31666.0
996	NaN	White	155144.0	3.0	5716.0
998	Honda	White	215883.0	4.0	4001.0
999	Toyota	Blue	248360.0	4.0	12732.0

597 rows × 5 columns

In [41]:
```
pd.crosstab(car_sales["Make"],car_sales["Doors"])
```

Out[41]:

Doors	3.0	4.0	5.0
Make			
BMW	21	0	72
Honda	0	276	0
Nissan	38	137	0
Toyota	0	359	0

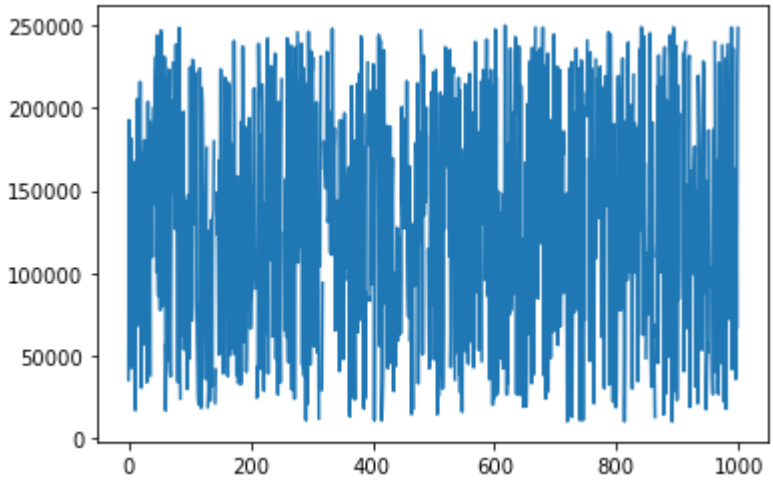
In [43]:
```
#groupby
car_sales.groupby(["Make"]).mean()
```

Out[43]:

Make	Odometer (KM)	Doors	Price
BMW	127491.439560	4.548387	26902.440860
Honda	125233.374101	4.000000	14618.661922
Nissan	136809.029070	3.782857	13664.502857
Toyota	135942.582873	4.000000	15715.257062

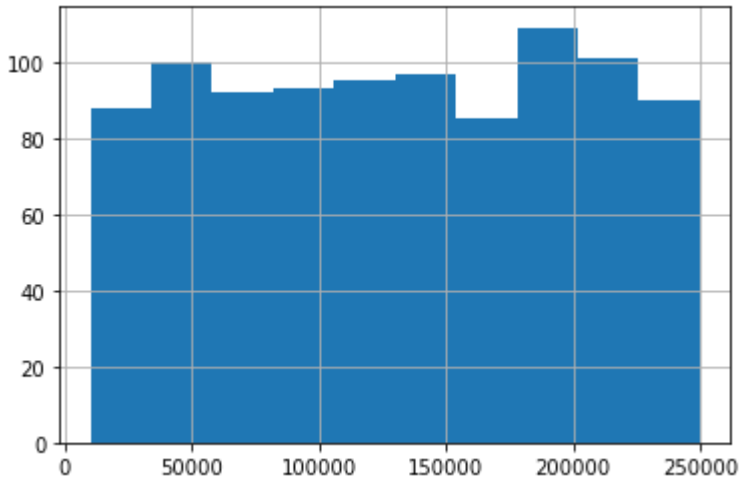
In [47]:
```
%matplotlib inline
import matplotlib.pyplot as plt
car_sales["Odometer (KM)"].plot()
```

Out[47]: <AxesSubplot:>



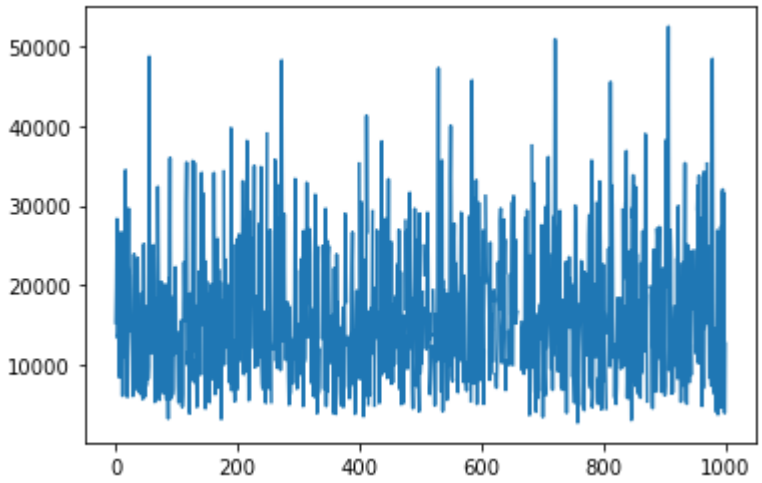
In [48]:
```
car_sales["Odometer (KM)"].hist()
```

Out[48]: <AxesSubplot:>



In [49]:
```
car_sales["Price"].plot()
```

Out[49]: <AxesSubplot:>



In [50]:
```
car_sales
```

Out[50]:

	Make	Colour	Odometer (KM)	Doors	Price
0	Honda	White	35431.0	4.0	15323.0
1	BMW	Blue	192714.0	5.0	19943.0
2	Honda	White	84714.0	4.0	28343.0
3	Toyota	White	154365.0	4.0	13434.0
4	Nissan	Blue	181577.0	3.0	14043.0
...	...	...	...	...	...
995	Toyota	Black	35820.0	4.0	32042.0
996	NaN	White	155144.0	3.0	5716.0
997	Nissan	Blue	66604.0	4.0	31570.0
998	Honda	White	215883.0	4.0	4001.0
999	Toyota	Blue	248360.0	4.0	12732.0

1000 rows × 5 columns

manipulating data

In [51]:
```
car_sales["Make"].str.lower()
```

Out[51]:
```
0       honda
1         bmw
2       honda
3      toyota
4      nissan
        ...
995    toyota
996       NaN
997    nissan
998     honda
999    toyota
Name: Make, Length: 1000, dtype: object
```

In [52]:
```
car_sales["Make"]=car_sales["Make"].str.lower()
```

In [53]:
```
car_sales
```

Out[53]:

	Make	Colour	Odometer (KM)	Doors	Price
0	honda	White	35431.0	4.0	15323.0
1	bmw	Blue	192714.0	5.0	19943.0
2	honda	White	84714.0	4.0	28343.0
3	toyota	White	154365.0	4.0	13434.0
4	nissan	Blue	181577.0	3.0	14043.0
...	...	...	...	...	...
995	toyota	Black	35820.0	4.0	32042.0
996	NaN	White	155144.0	3.0	5716.0
997	nissan	Blue	66604.0	4.0	31570.0
998	honda	White	215883.0	4.0	4001.0
999	toyota	Blue	248360.0	4.0	12732.0

1000 rows × 5 columns

In [55]:
```
#this is the method to how to fil missing data to use fillna method
car_sales_missing["Odometer"].fillna(car_sales_missing["odometer"].mean())
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call las
t)
Input In [55], in <cell line: 1>()
----> 1 car_sales_missing["Odometer"].fillna(car_sales_missing["odomete
r"].mean())

NameError: name 'car_sales_missing' is not defined
```

In [56]:
```
# The Pandas Series Object

# A Pandas Series is a one-dimensional array of indexed data. It can be cred

data=pd.Series([0.25,0.5,0.75,1])
data
```

Out[56]:
```
0    0.25
1    0.50
2    0.75
3    1.00
dtype: float64
```

In [58]:
```
data.values
```

Out[58]:
```
array([0.25, 0.5 , 0.75, 1.  ])
```

In [60]:
```
data.shape
```

Out[60]:
```
(4,)
```

In [61]:
```
data.index
```

Out[61]:
```
RangeIndex(start=0, stop=4, step=1)
```

In [62]:
```
data[1]
```

Out[62]:
```
0.5
```

In [63]:
```
data[:2]
```

Out[63]:
```
0    0.25
1    0.50
dtype: float64
```

In [64]:
```
data1=pd.Series([0.75,.50,.75,1],index=['a','b','c','d'])
data1
```

Out[64]:
```
a    0.75
b    0.50
c    0.75
d    1.00
dtype: float64
```

In [65]:
```
# Series as specialized dictionary

Population_dict={'California':123543,'Texas':87451,'Boston':986734,'Newyork'

Population=pd.Series(Population_dict)

Population
```

Out[65]:
```
California    123543
Texas          87451
Boston        986734
Newyork       907856
dtype: int64
```

In [66]:
```
pd.Series([1,2,3,4])
```

Out[66]:
```
0    1
1    2
2    3
3    4
dtype: int64
```

In [69]:
```
pd.Series(5,index=[100,200,300,400,500])
```

Out[69]:
```
100    5
200    5
300    5
400    5
500    5
dtype: int64
```

In [ ]:
```

```