

In [3]:

```
import pandas as pd
```

In [4]:

```
# 2 main datatype  
series=pd.Series(["BMW", "Toyota", "Honda"])
```

In [5]:

```
series
```

Out[5]:

```
0      BMW  
1    Toyota  
2     Honda  
dtype: object
```

In [6]:

```
#series=1-dimensional
```

In [7]:

```
colours=pd.Series(["Red", "Blue", "White"])  
colours
```

Out[7]:

```
0      Red  
1     Blue  
2    White  
dtype: object
```

In [8]:

```
#Dataframe=2 dimensional  
car_data=pd.DataFrame({ "car make": series, "colour": colours})  
car_data
```

Out[8]:

	car make	colour
0	BMW	Red
1	Toyota	Blue
2	Honda	White

In [9]:

```
#import data  
car_sales=pd.read_csv("Car_sales.csv")
```

In [10]:

```
car_sales
```

Out[10]:

	Manufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_in_th
0	Acura	Integra	16.919	16.360	Passenger	
1	Acura	TL	39.384	19.875	Passenger	
2	Acura	CL	14.114	18.225	Passenger	
3	Acura	RL	8.588	29.725	Passenger	
4	Audi	A4	20.397	22.255	Passenger	
...
152	Volvo	V40	3.545	NaN	Passenger	
153	Volvo	S70	15.245	NaN	Passenger	
154	Volvo	V70	17.531	NaN	Passenger	
155	Volvo	C70	3.493	NaN	Passenger	
156	Volvo	S80	18.969	NaN	Passenger	

157 rows × 16 columns

In [11]:

```
#exporting a dataframe
car_sales.to_csv("exported-car-sales.csv")
```

In [12]:

```
exported_car_sales=pd.read_csv("exported-car-sales.csv")
exported_car_sales
```

Out[12]:

	Unnamed: 0	Manufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type
0	0	Acura	Integra	16.919	16.360	Passenger
1	1	Acura	TL	39.384	19.875	Passenger
2	2	Acura	CL	14.114	18.225	Passenger
3	3	Acura	RL	8.588	29.725	Passenger
4	4	Audi	A4	20.397	22.255	Passenger
...
152	152	Volvo	V40	3.545	NaN	Passenger
153	153	Volvo	S70	15.245	NaN	Passenger
154	154	Volvo	V70	17.531	NaN	Passenger
155	155	Volvo	C70	3.493	NaN	Passenger
156	156	Volvo	S80	18.969	NaN	Passenger

157 rows × 7 columns

describe data

In [13]:

```
car_sales.columns
```

Out[13]:

```
Index(['Manufacturer', 'Model', 'Sales_in_thousands', '__year_resale_value',
      'Vehicle_type', 'Price_in_thousands', 'Engine_size', 'Horsepower',
      'Wheelbase', 'Width', 'Length', 'Curb_weight', 'Fuel_capacity',
      'Fuel_efficiency', 'Latest_Launch', 'Power_perf_factor'],
      dtype='object')
```

In [14]:

```
car_sales.dtypes
```

Out[14]:

```
Manufacturer      object
Model             object
Sales_in_thousands  float64
__year_resale_value float64
Vehicle_type      object
Price_in_thousands float64
Engine_size       float64
Horsepower        float64
Wheelbase         float64
Width            float64
Length           float64
Curb_weight       float64
Fuel_capacity     float64
Fuel_efficiency   float64
Latest_Launch    object
Power_perf_factor float64
dtype: object
```

In [15]:

```
car_columns=car_sales.columns
car_columns
```

Out[15]:

```
Index(['Manufacturer', 'Model', 'Sales_in_thousands', '__year_resale_value',
      'Vehicle_type', 'Price_in_thousands', 'Engine_size', 'Horsepower',
      'Wheelbase', 'Width', 'Length', 'Curb_weight', 'Fuel_capacity',
      'Fuel_efficiency', 'Latest_Launch', 'Power_perf_factor'],
      dtype='object')
```

In [16]:

```
car_sales.index
```

Out[16]:

```
RangeIndex(start=0, stop=157, step=1)
```

In [17]:

```
car_sales
```

Out[17]:

	Manufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_in_th
0	Acura	Integra	16.919	16.360	Passenger	
1	Acura	TL	39.384	19.875	Passenger	
2	Acura	CL	14.114	18.225	Passenger	
3	Acura	RL	8.588	29.725	Passenger	
4	Audi	A4	20.397	22.255	Passenger	
...
152	Volvo	V40	3.545	NaN	Passenger	
153	Volvo	S70	15.245	NaN	Passenger	
154	Volvo	V70	17.531	NaN	Passenger	
155	Volvo	C70	3.493	NaN	Passenger	
156	Volvo	S80	18.969	NaN	Passenger	

157 rows × 6 columns

In [18]:

```
car_sales.describe()
```

Out[18]:

	Sales_in_thousands	__year_resale_value	Price_in_thousands	Engine_size	Horsepower
count	157.000000	121.000000	155.000000	156.000000	156.000000
mean	52.998076	18.072975	27.390755	3.060897	185.948718
std	68.029422	11.453384	14.351653	1.044653	56.700321
min	0.110000	5.160000	9.235000	1.000000	55.000000
25%	14.114000	11.260000	18.017500	2.300000	149.500000
50%	29.450000	14.180000	22.799000	3.000000	177.500000
75%	67.956000	19.875000	31.947500	3.575000	215.000000
max	540.561000	67.550000	85.500000	8.000000	450.000000

In [19]:

```
car_sales.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 157 entries, 0 to 156
Data columns (total 16 columns):
#   Column                      Non-Null Count  Dtype
---  -
0   Manufacturer                 157 non-null    object
1   Model                        157 non-null    object
2   Sales_in_thousands          157 non-null    float64
3   __year_resale_value          121 non-null    float64
4   Vehicle_type                 157 non-null    object
5   Price_in_thousands          155 non-null    float64
6   Engine_size                  156 non-null    float64
7   Horsepower                   156 non-null    float64
8   Wheelbase                    156 non-null    float64
9   Width                        156 non-null    float64
10  Length                       156 non-null    float64
11  Curb_weight                   155 non-null    float64
12  Fuel_capacity                 156 non-null    float64
13  Fuel_efficiency               154 non-null    float64
14  Latest_Launch                157 non-null    object
15  Power_perf_factor             155 non-null    float64
dtypes: float64(12), object(4)
memory usage: 19.8+ KB
```

In [20]:

```
car_sales.mean()
```

```
C:\Users\satya shukla\AppData\Local\Temp\ipykernel_8660\4073448239.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError. Select only valid columns before calling the reduction.
  car_sales.mean()
```

Out[20]:

```
Sales_in_thousands    52.998076
__year_resale_value    18.072975
Price_in_thousands    27.390755
Engine_size            3.060897
Horsepower             185.948718
Wheelbase              107.487179
Width                  71.150000
Length                187.343590
Curb_weight            3.378026
Fuel_capacity          17.951923
Fuel_efficiency         23.844156
Power_perf_factor      77.043591
dtype: float64
```

In [21]:

```
car_prices=pd.Series([3000,1500,111250])
car_prices.mean()
```

Out[21]:

38583.333333333336

In [22]:

```
car_sales.sum()
```

Out[22]:

```
Manufacturer      AcuraAcuraAcuraAcuraAudiAudiAudiBMWBMWBMWBuick...
Model             IntegraTLCLRLA4A6A8323i328i528iCenturyRegalPar...
Sales_in_thousands      8320.698
__year_resale_value      2186.83
Vehicle_type       PassengerPassengerPassengerPassengerPassengerP...
Price_in_thousands      4245.567
Engine_size          477.5
Horsepower          29008.0
Wheelbase           16768.0
Width               11099.4
Length              29225.6
Curb_weight          523.594
Fuel_capacity        2800.5
Fuel_efficiency       3672.0
Latest_Launch       2/2/20126/3/20111/4/20123/10/201110/8/20118/9/...
Power_perf_factor     11941.756636
dtype: object
```

In [23]:

```
car_sales["Engine_size"].sum()
```

Out[23]:

477.5

VIEWING AND selecting data

In [24]:

```
car_sales.head()
```

Out[24]:

	Manufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_in_thou
0	Acura	Integra	16.919	16.360	Passenger	
1	Acura	TL	39.384	19.875	Passenger	
2	Acura	CL	14.114	18.225	Passenger	
3	Acura	RL	8.588	29.725	Passenger	
4	Audi	A4	20.397	22.255	Passenger	

In [25]:

```
car_sales
```

Out[25]:

	Manufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_in_th
0	Acura	Integra	16.919	16.360	Passenger	
1	Acura	TL	39.384	19.875	Passenger	
2	Acura	CL	14.114	18.225	Passenger	
3	Acura	RL	8.588	29.725	Passenger	
4	Audi	A4	20.397	22.255	Passenger	
...
152	Volvo	V40	3.545	NaN	Passenger	
153	Volvo	S70	15.245	NaN	Passenger	
154	Volvo	V70	17.531	NaN	Passenger	
155	Volvo	C70	3.493	NaN	Passenger	
156	Volvo	S80	18.969	NaN	Passenger	

157 rows × 16 columns

In [26]:

```
car_sales.head(7)
```

Out[26]:

	Manufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_in_thou
0	Acura	Integra	16.919	16.360	Passenger	
1	Acura	TL	39.384	19.875	Passenger	
2	Acura	CL	14.114	18.225	Passenger	
3	Acura	RL	8.588	29.725	Passenger	
4	Audi	A4	20.397	22.255	Passenger	
5	Audi	A6	18.780	23.555	Passenger	
6	Audi	A8	1.380	39.000	Passenger	

In [27]:

```
car_sales.tail()
```

Out[27]:

	Manufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_in_th
152	Volvo	V40	3.545	NaN	Passenger	
153	Volvo	S70	15.245	NaN	Passenger	
154	Volvo	V70	17.531	NaN	Passenger	
155	Volvo	C70	3.493	NaN	Passenger	
156	Volvo	S80	18.969	NaN	Passenger	

In [28]:

```
#.loc & .iloc  
animals=pd.Series(["cat","dog","bird","panda","snake"],index=[0,3,9,8,4])
```

In [29]:

```
animals
```

Out[29]:

```
0      cat  
3      dog  
9      bird  
8      panda  
4      snake  
dtype: object
```

In [30]:

```
animals.loc[3]
```

Out[30]:

```
'dog'
```

In [31]:

```
animals.loc[4]
```

Out[31]:

```
'snake'
```

In [32]:

```
car_sales.loc[3]
```

Out[32]:

Manufacturer	Acura
Model	RL
Sales_in_thousands	8.588
__year_resale_value	29.725
Vehicle_type	Passenger
Price_in_thousands	42.0
Engine_size	3.5
Horsepower	210.0
Wheelbase	114.6
Width	71.4
Length	196.6
Curb_weight	3.85
Fuel_capacity	18.0
Fuel_efficiency	22.0
Latest_Launch	3/10/2011
Power_perf_factor	91.389779

Name: 3, dtype: object

In [33]:

```
#.iloc refers to the position  
animals.iloc[3]
```

Out[33]:

'panda'

In [34]:

```
animals
```

Out[34]:

0	cat
3	dog
9	bird
8	panda
4	snake

dtype: object

In [35]:

```
car_sales.iloc[3]
```

Out[35]:

```
Manufacturer      Acura
Model              RL
Sales_in_thousands  8.588
__year_resale_value 29.725
Vehicle_type       Passenger
Price_in_thousands 42.0
Engine_size        3.5
Horsepower         210.0
Wheelbase          114.6
Width              71.4
Length            196.6
Curb_weight        3.85
Fuel_capacity       18.0
Fuel_efficiency     22.0
Latest_Launch      3/10/2011
Power_perf_factor   91.389779
Name: 3, dtype: object
```

In [36]:

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

Out[36]:

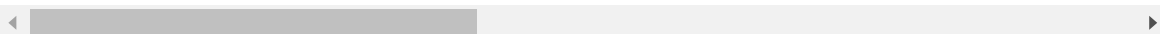
```
0    cat
3    dog
9    bird
dtype: object
```

In [37]:

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

Out[37]:

	Manufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_in_thou
0	Acura	Integra	16.919	16.360	Passenger	
1	Acura	TL	39.384	19.875	Passenger	
2	Acura	CL	14.114	18.225	Passenger	
3	Acura	RL	8.588	29.725	Passenger	



In [38]:

```
car_sales["Model"]
```

Out[38]:

```
0      Integra
1         TL
2         CL
3         RL
4         A4
...
152      V40
153      S70
154      V70
155      C70
156      S80
Name: Model, Length: 157, dtype: object
```

In [39]:

```
pd.crosstab(car_sales["Model"],car_sales["Model"])
```

Out[39]:

Model	3-Sep	3000GT	300M	323i	328i	4Runner	5-Sep	528i	A4	A6	...	Town & Country	Town car
Model													
3-Sep	1	0	0	0	0	0	0	0	0	0	...	0	0
3000GT	0	1	0	0	0	0	0	0	0	0	...	0	0
300M	0	0	1	0	0	0	0	0	0	0	...	0	0
323i	0	0	0	1	0	0	0	0	0	0	...	0	0
328i	0	0	0	0	1	0	0	0	0	0	...	0	0
...
Viper	0	0	0	0	0	0	0	0	0	0	...	0	0
Voyager	0	0	0	0	0	0	0	0	0	0	...	0	0
Windstar	0	0	0	0	0	0	0	0	0	0	...	0	0
Wrangler	0	0	0	0	0	0	0	0	0	0	...	0	0
Xterra	0	0	0	0	0	0	0	0	0	0	...	0	0

156 rows × 156 columns



In [40]:

```
car_sales.groupby(["Model"]).mean()
```

Out[40]:

	Sales_in_thousands	__year_resale_value	Price_in_thousands	Engine_size	Horsepower
Model					
3-Sep	12.115	NaN	26.100	2.0	1
3000GT	0.110	20.940	25.450	3.0	1
300M	30.696	NaN	29.185	3.5	2
323i	19.747	NaN	26.990	2.5	1
328i	9.231	28.675	33.400	2.8	1
...
Viper	0.916	58.470	69.725	8.0	4
Voyager	24.155	12.025	18.850	2.4	1
Windstar	155.787	13.175	21.410	3.0	1
Wrangler	55.557	13.475	14.460	2.5	1
Xterra	54.158	NaN	22.799	3.3	1

156 rows × 12 columns

In [41]:

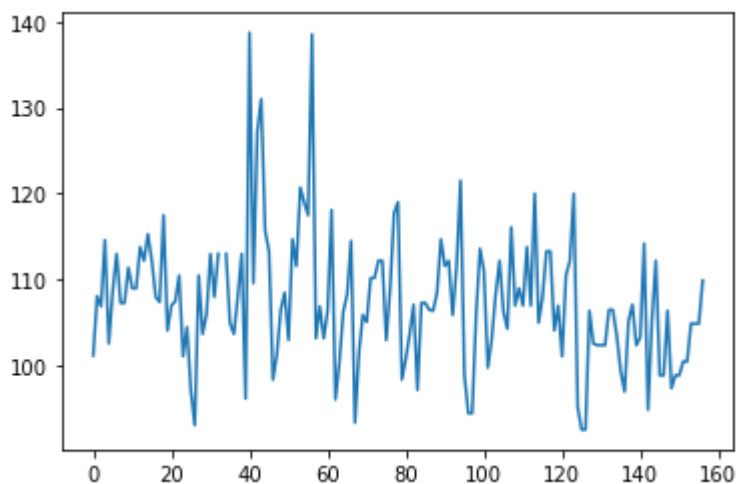
```
%matplotlib inline
import matplotlib.pyplot as plt
```

In [42]:

```
car_sales["Wheelbase"].plot()
```

Out[42]:

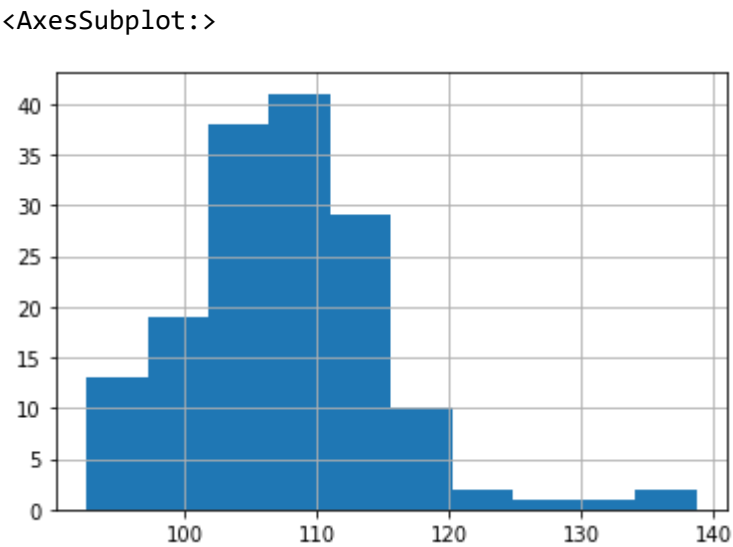
<AxesSubplot:>



In [43]:

```
car_sales["Wheelbase"].hist()
```

Out[43]:



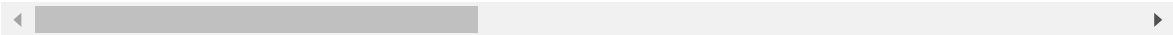
In [44]:

```
car_sales
```

Out[44]:

	Manufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_in_th
0	Acura	Integra	16.919	16.360	Passenger	
1	Acura	TL	39.384	19.875	Passenger	
2	Acura	CL	14.114	18.225	Passenger	
3	Acura	RL	8.588	29.725	Passenger	
4	Audi	A4	20.397	22.255	Passenger	
...
152	Volvo	V40	3.545	NaN	Passenger	
153	Volvo	S70	15.245	NaN	Passenger	
154	Volvo	V70	17.531	NaN	Passenger	
155	Volvo	C70	3.493	NaN	Passenger	
156	Volvo	S80	18.969	NaN	Passenger	

157 rows × 16 columns



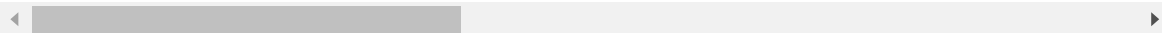
In [45]:

```
seats_column=pd.Series([5,5,5,5,5])
car_sales["seats"]=seats_column# new coloumn
car_sales
```

Out[45]:

	Manufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_in_th
0	Acura	Integra	16.919	16.360	Passenger	
1	Acura	TL	39.384	19.875	Passenger	
2	Acura	CL	14.114	18.225	Passenger	
3	Acura	RL	8.588	29.725	Passenger	
4	Audi	A4	20.397	22.255	Passenger	
...
152	Volvo	V40	3.545	NaN	Passenger	
153	Volvo	S70	15.245	NaN	Passenger	
154	Volvo	V70	17.531	NaN	Passenger	
155	Volvo	C70	3.493	NaN	Passenger	
156	Volvo	S80	18.969	NaN	Passenger	

157 rows × 7 columns



In [46]:

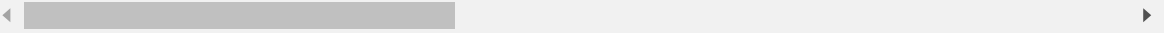
```
car_sales["seats"].fillna(5, inplace=True)
```

In [47]:

```
car_sales.head(13)
```

Out[47]:

	Manufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_in_th
0	Acura	Integra	16.919	16.360	Passenger	
1	Acura	TL	39.384	19.875	Passenger	
2	Acura	CL	14.114	18.225	Passenger	
3	Acura	RL	8.588	29.725	Passenger	
4	Audi	A4	20.397	22.255	Passenger	
5	Audi	A6	18.780	23.555	Passenger	
6	Audi	A8	1.380	39.000	Passenger	
7	BMW	323i	19.747	NaN	Passenger	
8	BMW	328i	9.231	28.675	Passenger	
9	BMW	528i	17.527	36.125	Passenger	
10	Buick	Century	91.561	12.475	Passenger	
11	Buick	Regal	39.350	13.740	Passenger	
12	Buick	Park Avenue	27.851	20.190	Passenger	



In [50]:

```
#create a coloumn for a single value
car_sales["number of wheels"]=4
car_sales
```

Out[50]:

_thousands	Engine_size	Horsepower	Wheelbase	Width	Length	Curb_weight	Fuel_capacity
21.50	1.8	140.0	101.2	67.3	172.4	2.639	13.2
28.40	3.2	225.0	108.1	70.3	192.9	3.517	17.2
NaN	3.2	225.0	106.9	70.6	192.0	3.470	17.2
42.00	3.5	210.0	114.6	71.4	196.6	3.850	18.0
23.99	1.8	150.0	102.6	68.2	178.0	2.998	16.4
...
24.40	1.9	160.0	100.5	67.6	176.6	3.042	15.8
27.50	2.4	168.0	104.9	69.3	185.9	3.208	17.9
28.80	2.4	168.0	104.9	69.3	186.2	3.259	17.9
45.50	2.3	236.0	104.9	71.5	185.7	3.601	18.5
36.00	2.9	201.0	109.9	72.1	189.8	3.600	21.1

In [51]:

```
car_sales["passed road safty"]= True
car_sales.dtypes
```

Out[51]:

```
Manufacturer      object
Model             object
Sales_in_thousands  float64
__year_resale_value float64
Vehicle_type      object
Price_in_thousands float64
Engine_size       float64
Horsepower        float64
Wheelbase         float64
Width             float64
Length           float64
Curb_weight       float64
Fuel_capacity     float64
Fuel_efficiency   float64
Latest_Launch     object
Power_perf_factor float64
seats            float64
number of wheels  int64
passed road safty bool
dtype: object
```

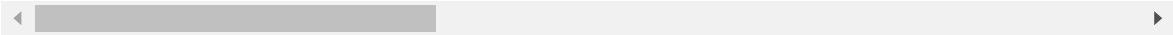
In [52]:

```
car_sales
```

Out[52]:

	Manufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_in_th
0	Acura	Integra	16.919	16.360	Passenger	
1	Acura	TL	39.384	19.875	Passenger	
2	Acura	CL	14.114	18.225	Passenger	
3	Acura	RL	8.588	29.725	Passenger	
4	Audi	A4	20.397	22.255	Passenger	
...
152	Volvo	V40	3.545	NaN	Passenger	
153	Volvo	S70	15.245	NaN	Passenger	
154	Volvo	V70	17.531	NaN	Passenger	
155	Volvo	C70	3.493	NaN	Passenger	
156	Volvo	S80	18.969	NaN	Passenger	

157 rows × 19 columns



In [54]:

```
car_sales.drop("Fuel_efficiency",axis= 1)
```

Out[54]:

	Manufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_in_th
0	Acura	Integra	16.919	16.360	Passenger	
1	Acura	TL	39.384	19.875	Passenger	
2	Acura	CL	14.114	18.225	Passenger	
3	Acura	RL	8.588	29.725	Passenger	
4	Audi	A4	20.397	22.255	Passenger	
...
152	Volvo	V40	3.545	NaN	Passenger	
153	Volvo	S70	15.245	NaN	Passenger	
154	Volvo	V70	17.531	NaN	Passenger	
155	Volvo	C70	3.493	NaN	Passenger	
156	Volvo	S80	18.969	NaN	Passenger	

157 rows × 18 columns



In [55]:

```
car_sales.sample(frac=1)
```

Out[55]:

	Manufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_in_
142	Toyota	RAV4	25.106	13.325	Car	
137	Toyota	Camry	247.994	13.245	Passenger	
79	Mitsubishi	Mirage	26.232	8.325	Passenger	
131	Saturn	SW	5.223	10.790	Passenger	
19	Chevrolet	Cavalier	145.519	9.250	Passenger	
...
73	Lexus	LS400	6.375	40.375	Passenger	
109	Oldsmobile	Intrigue	38.554	NaN	Passenger	
115	Plymouth	Breeze	5.240	9.800	Passenger	
119	Pontiac	Grand Am	131.097	10.290	Passenger	
16	Cadillac	Eldorado	6.536	25.725	Passenger	

157 rows × 19 columns



In [56]:

```
car_sales_shuffled=car_sales.sample(frac=1)
```

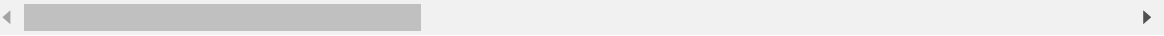
In [57]:

```
car_sales_shuffled
```

Out[57]:

	Manufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_in_
13	Buick	LeSabre	83.257	13.360	Passenger	
57	Honda	Civic	199.685	9.850	Passenger	
51	Ford	Crown Victoria	63.403	14.210	Passenger	
72	Lexus	GS400	3.334	NaN	Passenger	
82	Mitsubishi	Diamante	5.711	16.575	Passenger	
...	
134	Subaru	Outback	47.107	NaN	Passenger	
128	Saab	3-Sep	12.115	NaN	Passenger	
8	BMW	328i	9.231	28.675	Passenger	
79	Mitsubishi	Mirage	26.232	8.325	Passenger	
39	Dodge	Viper	0.916	58.470	Passenger	

157 rows × 19 columns



In [58]:

```
car_sales["Engine_size"]=car_sales["Engine_size"].apply(lambda x : x/1.6)
car_sales
```

Out[58]:

	Manufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_in_th
0	Acura	Integra	16.919	16.360	Passenger	
1	Acura	TL	39.384	19.875	Passenger	
2	Acura	CL	14.114	18.225	Passenger	
3	Acura	RL	8.588	29.725	Passenger	
4	Audi	A4	20.397	22.255	Passenger	
...
152	Volvo	V40	3.545	NaN	Passenger	
153	Volvo	S70	15.245	NaN	Passenger	
154	Volvo	V70	17.531	NaN	Passenger	
155	Volvo	C70	3.493	NaN	Passenger	
156	Volvo	S80	18.969	NaN	Passenger	

157 rows × 19 columns



In []: