```
In [1]: import pandas as pd
        #Import data
        car_sales=pd.read_csv("car-sales.csv")
```

Manipulating Data

```
In [2]: car_sales["Make"].str.lower()
```

```
Out[2]: 0    toyota
        1     honda
        2    toyota
        3       bmw
        4    nissan
        5    toyota
        6     honda
        7     honda
        8    toyota
        9    nissan
        Name: Make, dtype: object
```

```
In [3]: #Here you can check that it is not updtaed, thus to save your changes to columns
        car_sales["Make"]=car_sales["Make"].str.lower()
        car_sales
```

Out[3]:

	Make	Colour	Odometer (KM)	Doors	Price
0	toyota	White	150043	4	$4,000.00
1	honda	Red	87899	4	$5,000.00
2	toyota	Blue	32549	3	$7,000.00
3	bmw	Black	11179	5	$22,000.00
4	nissan	White	213095	4	$3,500.00
5	toyota	Green	99213	4	$4,500.00
6	honda	Blue	45698	4	$7,500.00
7	honda	Blue	54738	4	$7,000.00
8	toyota	White	60000	4	$6,250.00
9	nissan	White	31600	4	$9,700.00

```
In [4]: car_sales_missing=pd.read_csv("car-sales-missing-data.csv")
        car_sales_missing
```

KUSHAGRA GUPTA

Out[4]:

	Make	Colour	Odometer	Doors	Price
0	Toyota	White	150043.0	4.0	$4,000
1	Honda	Red	87899.0	4.0	$5,000
2	Toyota	Blue	NaN	3.0	$7,000
3	BMW	Black	11179.0	5.0	$22,000
4	Nissan	White	213095.0	4.0	$3,500
5	Toyota	Green	NaN	4.0	$4,500
6	Honda	NaN	NaN	4.0	$7,500
7	Honda	Blue	NaN	4.0	NaN
8	Toyota	White	60000.0	NaN	NaN
9	NaN	White	31600.0	4.0	$9,700

In [5]:

```
car_sales_missing["Odometer"].mean()
```

Out[5]:

```
92302.66666666667
```

In [6]:

```
car_sales_missing["Odometer"].fillna(car_sales_missing["Odometer"].mean())
```

Out[6]:

```
0    150043.000000
1     87899.000000
2     92302.666667
3     11179.000000
4    213095.000000
5     92302.666667
6     92302.666667
7     92302.666667
8     60000.000000
9     31600.000000
Name: Odometer, dtype: float64
```

In [7]:

```
# But if we open car_sales_missing it still wont show the updated values
car_sales_missing
```

Out[7]:

	Make	Colour	Odometer	Doors	Price
0	Toyota	White	150043.0	4.0	$4,000
1	Honda	Red	87899.0	4.0	$5,000
2	Toyota	Blue	NaN	3.0	$7,000
3	BMW	Black	11179.0	5.0	$22,000
4	Nissan	White	213095.0	4.0	$3,500
5	Toyota	Green	NaN	4.0	$4,500
6	Honda	NaN	NaN	4.0	$7,500
7	Honda	Blue	NaN	4.0	NaN
8	Toyota	White	60000.0	NaN	NaN
9	NaN	White	31600.0	4.0	$9,700

In [8]:

```
# First way
```

KUSHAGRA GUPTA

```
car_sales_missing["Odometer"].fillna(car_sales_missing["Odometer"].mean(),inplace=
# inplace by default FALSE hota hai,but agar True krde tho changes change directly


#Second way of doing the smae thing
# car_sales_missing["Odometer"]=car_sales_missing["Odometer"].fillna(car_sales_mis
```

In [9]:
```
car_sales_missing
```

Out[9]:

	Make	Colour	Odometer	Doors	Price
0	Toyota	White	150043.000000	4.0	$4,000
1	Honda	Red	87899.000000	4.0	$5,000
2	Toyota	Blue	92302.666667	3.0	$7,000
3	BMW	Black	11179.000000	5.0	$22,000
4	Nissan	White	213095.000000	4.0	$3,500
5	Toyota	Green	92302.666667	4.0	$4,500
6	Honda	NaN	92302.666667	4.0	$7,500
7	Honda	Blue	92302.666667	4.0	NaN
8	Toyota	White	60000.000000	NaN	NaN
9	NaN	White	31600.000000	4.0	$9,700

In [10]:
```
car_sales_missing.dropna()
```

Out[10]:

	Make	Colour	Odometer	Doors	Price
0	Toyota	White	150043.000000	4.0	$4,000
1	Honda	Red	87899.000000	4.0	$5,000
2	Toyota	Blue	92302.666667	3.0	$7,000
3	BMW	Black	11179.000000	5.0	$22,000
4	Nissan	White	213095.000000	4.0	$3,500
5	Toyota	Green	92302.666667	4.0	$4,500

In [11]:
```
car_sales_missing
```

KUSHAGRA GUPTA

Out[11]:

	Make	Colour	Odometer	Doors	Price
0	Toyota	White	150043.000000	4.0	$4,000
1	Honda	Red	87899.000000	4.0	$5,000
2	Toyota	Blue	92302.666667	3.0	$7,000
3	BMW	Black	11179.000000	5.0	$22,000
4	Nissan	White	213095.000000	4.0	$3,500
5	Toyota	Green	92302.666667	4.0	$4,500
6	Honda	NaN	92302.666667	4.0	$7,500
7	Honda	Blue	92302.666667	4.0	NaN
8	Toyota	White	60000.000000	NaN	NaN
9	NaN	White	31600.000000	4.0	$9,700

In [12]:

```
#Again it hasn't happened again.
car_sales_missing.dropna(inplace=True)
car_sales_missing
```

Out[12]:

	Make	Colour	Odometer	Doors	Price
0	Toyota	White	150043.000000	4.0	$4,000
1	Honda	Red	87899.000000	4.0	$5,000
2	Toyota	Blue	92302.666667	3.0	$7,000
3	BMW	Black	11179.000000	5.0	$22,000
4	Nissan	White	213095.000000	4.0	$3,500
5	Toyota	Green	92302.666667	4.0	$4,500

In [13]:

```
car_sales_missing=pd.read_csv("car-sales-missing-data.csv")
car_sales_missing
#we reinitialized our data, thus it agains contains missing values
```

Out[13]:

	Make	Colour	Odometer	Doors	Price
0	Toyota	White	150043.0	4.0	$4,000
1	Honda	Red	87899.0	4.0	$5,000
2	Toyota	Blue	NaN	3.0	$7,000
3	BMW	Black	11179.0	5.0	$22,000
4	Nissan	White	213095.0	4.0	$3,500
5	Toyota	Green	NaN	4.0	$4,500
6	Honda	NaN	NaN	4.0	$7,500
7	Honda	Blue	NaN	4.0	NaN
8	Toyota	White	60000.0	NaN	NaN
9	NaN	White	31600.0	4.0	$9,700

In [14]:

```
car_sales_missing_dropped=car_sales_missing.dropna()
```

KUSHAGRA
GUPTA

In [15]:

```
car_sales_missing_dropped
```

Out[15]:

	Make	Colour	Odometer	Doors	Price
0	Toyota	White	150043.0	4.0	$4,000
1	Honda	Red	87899.0	4.0	$5,000
3	BMW	Black	11179.0	5.0	$22,000
4	Nissan	White	213095.0	4.0	$3,500

In [16]:

```
car_sales_missing_dropped.to_csv("car-sales-missing-data-dropped.csv")
```

In [17]:

```
#column from series
seats_column=pd.Series([5,5,5,5,5])

#New Column called seats
car_sales["Seats"]=seats_column
car_sales
```

Out[17]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats
0	toyota	White	150043	4	$4,000.00	5.0
1	honda	Red	87899	4	$5,000.00	5.0
2	toyota	Blue	32549	3	$7,000.00	5.0
3	bmw	Black	11179	5	$22,000.00	5.0
4	nissan	White	213095	4	$3,500.00	5.0
5	toyota	Green	99213	4	$4,500.00	NaN
6	honda	Blue	45698	4	$7,500.00	NaN
7	honda	Blue	54738	4	$7,000.00	NaN
8	toyota	White	60000	4	$6,250.00	NaN
9	nissan	White	31600	4	$9,700.00	NaN

In [18]:

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

KUSHAGRA
GUPTA

Out[18]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats
0	toyota	White	150043	4	$4,000.00	5.0
1	honda	Red	87899	4	$5,000.00	5.0
2	toyota	Blue	32549	3	$7,000.00	5.0
3	bmw	Black	11179	5	$22,000.00	5.0
4	nissan	White	213095	4	$3,500.00	5.0
5	toyota	Green	99213	4	$4,500.00	5.0
6	honda	Blue	45698	4	$7,500.00	5.0
7	honda	Blue	54738	4	$7,000.00	5.0
8	toyota	White	60000	4	$6,250.00	5.0
9	nissan	White	31600	4	$9,700.00	5.0

In [19]:

```
#Column from Python List
fuel_economy=[7.5 , 9.2 , 5.0 , 9.6 ]
car_sales["Fuel per 100KM"]=fuel_economy
car_sales
```

KUSHAGRA GUPTA

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [19], in <cell line: 3>()
      1 #Column from Python List
      2 fuel_economy=[7.5 , 9.2 , 5.0 , 9.6 ]
----> 3 car_sales["Fuel per 100KM"]=fuel_economy
      4 car_sales

File ~\anaconda3\lib\site-packages\pandas\core\frame.py:3655, in DataFrame.__setit
em__(self, key, value)
   3652     self._setitem_array([key], value)
   3653 else:
   3654     # set column
-> 3655     self._set_item(key, value)

File ~\anaconda3\lib\site-packages\pandas\core\frame.py:3832, in DataFrame._set_it
em(self, key, value)
   3822 def _set_item(self, key, value) -> None:
   3823     """
   3824     Add series to DataFrame in specified column.
   3825
   (...)
   3830     ensure homogeneity.
   3831     """
-> 3832     value = self._sanitize_column(value)
   3834     if (
   3835         key in self.columns
   3836         and value.ndim == 1
   3837         and not is_extension_array_dtype(value)
   3838     ):
   3839         # broadcast across multiple columns if necessary
   3840         if not self.columns.is_unique or isinstance(self.columns, MultiInd
ex):

File ~\anaconda3\lib\site-packages\pandas\core\frame.py:4535, in DataFrame._saniti
ze_column(self, value)
   4532     return _reindex_for_setitem(value, self.index)
   4534 if is_list_like(value):
-> 4535     com.require_length_match(value, self.index)
   4536 return sanitize_array(value, self.index, copy=True, allow_2d=True)

File ~\anaconda3\lib\site-packages\pandas\core\common.py:557, in require_length_ma
tch(data, index)
    553 """
    554 Check the length of data matches the length of the index.
    555 """
    556 if len(data) != len(index):
--> 557     raise ValueError(
    558         "Length of values "
    559         f"({len(data)}) "
    560         "does not match length of index "
    561         f"({len(index)})"
    562     )

ValueError: Length of values (4) does not match length of index (10)
```

In [20]:

```
'''
Thus from above we get to know that ,if you are filling data with the help of serie
existing dataframe but if you are filling data with the help of python list then it
as existing dataframe.
'''
#Column from Python List
fuel_economy=[7.5 , 9.2 , 5.0 , 9.6 , 8.7 , 4.7 , 7.6 , 8.7 , 3.0 ,4.5]
```

KUSHAGRA GUPTA

```
car_sales["Fuel per 100KM"]=fuel_economy
car_sales
```

Out[20]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	Fuel per 100KM
0	toyota	White	150043	4	$4,000.00	5.0	7.5
1	honda	Red	87899	4	$5,000.00	5.0	9.2
2	toyota	Blue	32549	3	$7,000.00	5.0	5.0
3	bmw	Black	11179	5	$22,000.00	5.0	9.6
4	nissan	White	213095	4	$3,500.00	5.0	8.7
5	toyota	Green	99213	4	$4,500.00	5.0	4.7
6	honda	Blue	45698	4	$7,500.00	5.0	7.6
7	honda	Blue	54738	4	$7,000.00	5.0	8.7
8	toyota	White	60000	4	$6,250.00	5.0	3.0
9	nissan	White	31600	4	$9,700.00	5.0	4.5

In [21]:

```
car_sales['Total Fuel Used(L)']=car_sales["Odometer (KM)"]/100 * car_sales["Fuel pe
```

In [22]:

```
car_sales
```

Out[22]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	Fuel per 100KM	Total Fuel Used(L)
0	toyota	White	150043	4	$4,000.00	5.0	7.5	11253.225
1	honda	Red	87899	4	$5,000.00	5.0	9.2	8086.708
2	toyota	Blue	32549	3	$7,000.00	5.0	5.0	1627.450
3	bmw	Black	11179	5	$22,000.00	5.0	9.6	1073.184
4	nissan	White	213095	4	$3,500.00	5.0	8.7	18539.265
5	toyota	Green	99213	4	$4,500.00	5.0	4.7	4663.011
6	honda	Blue	45698	4	$7,500.00	5.0	7.6	3473.048
7	honda	Blue	54738	4	$7,000.00	5.0	8.7	4762.206
8	toyota	White	60000	4	$6,250.00	5.0	3.0	1800.000
9	nissan	White	31600	4	$9,700.00	5.0	4.5	1422.000

In [23]:

```
car_sales["Number Of Wheels"]=4
car_sales
```

KUSHAGRA GUPTA

Out[23]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	Fuel per 100KM	Total Fuel Used(L)	Number Of Wheels
0	toyota	White	150043	4	$4,000.00	5.0	7.5	11253.225	4
1	honda	Red	87899	4	$5,000.00	5.0	9.2	8086.708	4
2	toyota	Blue	32549	3	$7,000.00	5.0	5.0	1627.450	4
3	bmw	Black	11179	5	$22,000.00	5.0	9.6	1073.184	4
4	nissan	White	213095	4	$3,500.00	5.0	8.7	18539.265	4
5	toyota	Green	99213	4	$4,500.00	5.0	4.7	4663.011	4
6	honda	Blue	45698	4	$7,500.00	5.0	7.6	3473.048	4
7	honda	Blue	54738	4	$7,000.00	5.0	8.7	4762.206	4
8	toyota	White	60000	4	$6,250.00	5.0	3.0	1800.000	4
9	nissan	White	31600	4	$9,700.00	5.0	4.5	1422.000	4

In [24]:
```
car_sales["Passed Road Safety"]=True
car_sales.dtypes
```

Out[24]:
```
Make                   object
Colour                 object
Odometer (KM)           int64
Doors                   int64
Price                  object
Seats                 float64
Fuel per 100KM        float64
Total Fuel Used(L)    float64
Number Of Wheels        int64
Passed Road Safety       bool
dtype: object
```

In [25]:
```
car_sales
```

Out[25]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	Fuel per 100KM	Total Fuel Used(L)	Number Of Wheels	Passed Road Safety
0	toyota	White	150043	4	$4,000.00	5.0	7.5	11253.225	4	True
1	honda	Red	87899	4	$5,000.00	5.0	9.2	8086.708	4	True
2	toyota	Blue	32549	3	$7,000.00	5.0	5.0	1627.450	4	True
3	bmw	Black	11179	5	$22,000.00	5.0	9.6	1073.184	4	True
4	nissan	White	213095	4	$3,500.00	5.0	8.7	18539.265	4	True
5	toyota	Green	99213	4	$4,500.00	5.0	4.7	4663.011	4	True
6	honda	Blue	45698	4	$7,500.00	5.0	7.6	3473.048	4	True
7	honda	Blue	54738	4	$7,000.00	5.0	8.7	4762.206	4	True
8	toyota	White	60000	4	$6,250.00	5.0	3.0	1800.000	4	True
9	nissan	White	31600	4	$9,700.00	5.0	4.5	1422.000	4	True

In [26]:
```
car_sales=car_sales.drop("Total Fuel Used(L)",axis=1)
```

In [27]:
```
car_sales
```

KUSHAGRA GUPTA

Out[27]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	Fuel per 100KM	Number Of Wheels	Passed Road Safety
0	toyota	White	150043	4	$4,000.00	5.0	7.5	4	True
1	honda	Red	87899	4	$5,000.00	5.0	9.2	4	True
2	toyota	Blue	32549	3	$7,000.00	5.0	5.0	4	True
3	bmw	Black	11179	5	$22,000.00	5.0	9.6	4	True
4	nissan	White	213095	4	$3,500.00	5.0	8.7	4	True
5	toyota	Green	99213	4	$4,500.00	5.0	4.7	4	True
6	honda	Blue	45698	4	$7,500.00	5.0	7.6	4	True
7	honda	Blue	54738	4	$7,000.00	5.0	8.7	4	True
8	toyota	White	60000	4	$6,250.00	5.0	3.0	4	True
9	nissan	White	31600	4	$9,700.00	5.0	4.5	4	True

In [28]:

```
car_sales_shuffled=car_sales.sample(frac=1)
#Suppose we wrote frac=0.5 that meant it had shuffled 50% of the data
car_sales_shuffled
```

Out[28]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	Fuel per 100KM	Number Of Wheels	Passed Road Safety
2	toyota	Blue	32549	3	$7,000.00	5.0	5.0	4	True
6	honda	Blue	45698	4	$7,500.00	5.0	7.6	4	True
8	toyota	White	60000	4	$6,250.00	5.0	3.0	4	True
4	nissan	White	213095	4	$3,500.00	5.0	8.7	4	True
0	toyota	White	150043	4	$4,000.00	5.0	7.5	4	True
5	toyota	Green	99213	4	$4,500.00	5.0	4.7	4	True
7	honda	Blue	54738	4	$7,000.00	5.0	8.7	4	True
1	honda	Red	87899	4	$5,000.00	5.0	9.2	4	True
9	nissan	White	31600	4	$9,700.00	5.0	4.5	4	True
3	bmw	Black	11179	5	$22,000.00	5.0	9.6	4	True

In [29]:

```
car_sales
```

KUSHAGRA GUPTA

Out[29]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	Fuel per 100KM	Number Of Wheels	Passed Road Safety
0	toyota	White	150043	4	$4,000.00	5.0	7.5	4	True
1	honda	Red	87899	4	$5,000.00	5.0	9.2	4	True
2	toyota	Blue	32549	3	$7,000.00	5.0	5.0	4	True
3	bmw	Black	11179	5	$22,000.00	5.0	9.6	4	True
4	nissan	White	213095	4	$3,500.00	5.0	8.7	4	True
5	toyota	Green	99213	4	$4,500.00	5.0	4.7	4	True
6	honda	Blue	45698	4	$7,500.00	5.0	7.6	4	True
7	honda	Blue	54738	4	$7,000.00	5.0	8.7	4	True
8	toyota	White	60000	4	$6,250.00	5.0	3.0	4	True
9	nissan	White	31600	4	$9,700.00	5.0	4.5	4	True

In [31]:

```
#suppose you only want to see changes made on random 2-% of the data
car_sales_shuffled.sample(frac=0.2)
```

Out[31]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	Fuel per 100KM	Number Of Wheels	Passed Road Safety
2	toyota	Blue	32549	3	$7,000.00	5.0	5.0	4	True
4	nissan	White	213095	4	$3,500.00	5.0	8.7	4	True

In [32]:

```
car_sales_shuffled
```

Out[32]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	Fuel per 100KM	Number Of Wheels	Passed Road Safety
2	toyota	Blue	32549	3	$7,000.00	5.0	5.0	4	True
6	honda	Blue	45698	4	$7,500.00	5.0	7.6	4	True
8	toyota	White	60000	4	$6,250.00	5.0	3.0	4	True
4	nissan	White	213095	4	$3,500.00	5.0	8.7	4	True
0	toyota	White	150043	4	$4,000.00	5.0	7.5	4	True
5	toyota	Green	99213	4	$4,500.00	5.0	4.7	4	True
7	honda	Blue	54738	4	$7,000.00	5.0	8.7	4	True
1	honda	Red	87899	4	$5,000.00	5.0	9.2	4	True
9	nissan	White	31600	4	$9,700.00	5.0	4.5	4	True
3	bmw	Black	11179	5	$22,000.00	5.0	9.6	4	True

In [34]:

```
car_sales_shuffled.reset_index(inplace=True)
car_sales_shuffled
```

KUSHAGRA GUPTA

Out[34]:

	level_0	index	Make	Colour	Odometer (KM)	Doors	Price	Seats	Fuel per 100KM	Number Of Wheels	Passed Road Safety
0	0	2	toyota	Blue	32549	3	$7,000.00	5.0	5.0	4	True
1	1	6	honda	Blue	45698	4	$7,500.00	5.0	7.6	4	True
2	2	8	toyota	White	60000	4	$6,250.00	5.0	3.0	4	True
3	3	4	nissan	White	213095	4	$3,500.00	5.0	8.7	4	True
4	4	0	toyota	White	150043	4	$4,000.00	5.0	7.5	4	True
5	5	5	toyota	Green	99213	4	$4,500.00	5.0	4.7	4	True
6	6	7	honda	Blue	54738	4	$7,000.00	5.0	8.7	4	True
7	7	1	honda	Red	87899	4	$5,000.00	5.0	9.2	4	True
8	8	9	nissan	White	31600	4	$9,700.00	5.0	4.5	4	True
9	9	3	bmw	Black	11179	5	$22,000.00	5.0	9.6	4	True

In [35]:

```
#It has created all together a new index i.e. someting we dont want to happen
car_sales_shuffled.reset_index(inplace=True,drop=True)
car_sales_shuffled
```

Out[35]:

	level_0	index	Make	Colour	Odometer (KM)	Doors	Price	Seats	Fuel per 100KM	Number Of Wheels	Passed Road Safety
0	0	2	toyota	Blue	32549	3	$7,000.00	5.0	5.0	4	True
1	1	6	honda	Blue	45698	4	$7,500.00	5.0	7.6	4	True
2	2	8	toyota	White	60000	4	$6,250.00	5.0	3.0	4	True
3	3	4	nissan	White	213095	4	$3,500.00	5.0	8.7	4	True
4	4	0	toyota	White	150043	4	$4,000.00	5.0	7.5	4	True
5	5	5	toyota	Green	99213	4	$4,500.00	5.0	4.7	4	True
6	6	7	honda	Blue	54738	4	$7,000.00	5.0	8.7	4	True
7	7	1	honda	Red	87899	4	$5,000.00	5.0	9.2	4	True
8	8	9	nissan	White	31600	4	$9,700.00	5.0	4.5	4	True
9	9	3	bmw	Black	11179	5	$22,000.00	5.0	9.6	4	True

In [37]:

```
car_sales
```

KUSHAGRA GUPTA

Out[37]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	Fuel per 100KM	Number Of Wheels	Passed Road Safety
0	toyota	White	150043	4	$4,000.00	5.0	7.5	4	True
1	honda	Red	87899	4	$5,000.00	5.0	9.2	4	True
2	toyota	Blue	32549	3	$7,000.00	5.0	5.0	4	True
3	bmw	Black	11179	5	$22,000.00	5.0	9.6	4	True
4	nissan	White	213095	4	$3,500.00	5.0	8.7	4	True
5	toyota	Green	99213	4	$4,500.00	5.0	4.7	4	True
6	honda	Blue	45698	4	$7,500.00	5.0	7.6	4	True
7	honda	Blue	54738	4	$7,000.00	5.0	8.7	4	True
8	toyota	White	60000	4	$6,250.00	5.0	3.0	4	True
9	nissan	White	31600	4	$9,700.00	5.0	4.5	4	True

In [38]:

```
# Suppose we have to convert KM to Miles
car_sales["Odometer (KM)"]=car_sales["Odometer (KM)"].apply(lambda x:x/1.6)
#we use apply function when we need to apply same type of operation throughout the
#lambda is a keyword in python,which is short for an anonymous function, so this i
#apply this function to X divided by 1.6
car_sales
```

Out[38]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	Fuel per 100KM	Number Of Wheels	Passed Road Safety
0	toyota	White	93776.875	4	$4,000.00	5.0	7.5	4	True
1	honda	Red	54936.875	4	$5,000.00	5.0	9.2	4	True
2	toyota	Blue	20343.125	3	$7,000.00	5.0	5.0	4	True
3	bmw	Black	6986.875	5	$22,000.00	5.0	9.6	4	True
4	nissan	White	133184.375	4	$3,500.00	5.0	8.7	4	True
5	toyota	Green	62008.125	4	$4,500.00	5.0	4.7	4	True
6	honda	Blue	28561.250	4	$7,500.00	5.0	7.6	4	True
7	honda	Blue	34211.250	4	$7,000.00	5.0	8.7	4	True
8	toyota	White	37500.000	4	$6,250.00	5.0	3.0	4	True
9	nissan	White	19750.000	4	$9,700.00	5.0	4.5	4	True

In [ ]:

KUSHAGRA GUPTA