Pandas

Series, DataFrames and CSVs

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

In [2]:
```
# 2 main datatypes: A. Series 1-Dimentional B. DataFrame : 2-Dimentional
series = pd.Series(["BMW", "Toyota", "Honda"])
```

In [3]:
```
series
```

Out[3]:
```
0       BMW
1    Toyota
2     Honda
dtype: object
```

In [4]:
```
colours = pd.Series(["Red", "Blue", "Yellow"])
colours
```

Out[4]:
```
0       Red
1      Blue
2    Yellow
dtype: object
```

In [5]:
```
car_data = pd.DataFrame({"Car make": series, "Colour": colours})
car_data
```

Out[5]:

	Car make	Colour
0	BMW	Red
1	Toyota	Blue
2	Honda	Yellow

In [6]:
```
# Import Data (CSV)
car_sales = pd.read_csv("car-sales.csv")
car_sales
# Row: axis = 0, Column: axis = 1
```

Out[6]:

	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 [7]:
```
# Exporting a dataframe
# Avoid reidexing using index = False
car_sales.to_csv("exported-car-sales.csv", index=False)
exported_car_sales = pd.read_csv("exported-car-sales.csv")
exported_car_sales
```

Out[7]:

	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

Describing Data

In [8]:
```
# Attribute
car_sales.dtypes
# Function
#car_sales.to_csv()
```

Out[8]:
```
Make             object
Colour           object
Odometer (KM)     int64
Doors             int64
Price            object
dtype: object
```

In [9]:
```
car_sales.columns
```

Out[9]:
```
Index(['Make', 'Colour', 'Odometer (KM)', 'Doors', 'Price'], dtype='object')
```

In [10]:
```
car_sales.index
```

Out[10]:
```
RangeIndex(start=0, stop=10, step=1)
```

In [11]:
```
car_sales.describe()
```

Out[11]:

	Odometer (KM)	Doors
count	10.000000	10.000000
mean	78601.400000	4.000000
std	61983.471735	0.471405
min	11179.000000	3.000000
25%	35836.250000	4.000000
50%	57369.000000	4.000000
75%	96384.500000	4.000000
max	213095.000000	5.000000

In [12]:
```
car_sales.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   Make           10 non-null     object
 1   Colour         10 non-null     object
 2   Odometer (KM)  10 non-null     int64 
 3   Doors          10 non-null     int64 
 4   Price          10 non-null     object
dtypes: int64(2), object(3)
memory usage: 528.0+ bytes
```

In [13]:
```
car_sales["Doors"].mean()
```

Out[13]:
```
4.0
```

In [14]:
```
car_sales.sum()
```

Out[14]:
```
Make             ToyotaHondaToyotaBMWNissanToyotaHondaHondaToyo...
Colour               WhiteRedBlueBlackWhiteGreenBlueBlueWhiteWhite
Odometer (KM)                                                786014
Doors                                                            40
Price            $4,000.00$5,000.00$7,000.00$22,000.00$3,500.00...
dtype: object
```

In [15]:
```
len(car_sales)
```

Out[15]:
```
10
```

Selecting & Viewing Data

In [16]:
```
car_sales.head() # Top 5 rows in case of large data
```

Out[16]:

	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

In [17]:
```
car_sales.tail() # Bottom 5 rows
```

Out[17]:

	Make	Colour	Odometer (KM)	Doors	Price
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 [18]:
```
# .loc & .iloc
animals = pd.Series(["cat", "dog", "panda", "snake", "lion"], index=[0, 3, 9, 8,
animals
```

Out[18]:
```
0      cat
3      dog
9    panda
8    snake
3     lion
dtype: object
```

In [19]:
```
# loc refers to index
animals.loc[3]
```

Out[19]:
```
3     dog
3    lion
dtype: object
```

In [20]:
```
car_sales.loc[3]
```

Out[20]:
```
Make                    BMW
Colour                Black
Odometer (KM)         11179
Doors                     5
Price            $22,000.00
Name: 3, dtype: object
```

In [21]:
```
#iloc refers to position
animals.iloc[3]
```

Out[21]:
```
'snake'
```

In [22]:
```
animals.iloc[:3]
```

Out[22]:
```
0      cat
3      dog
9    panda
dtype: object
```

In [23]:
```
car_sales.loc[:5]
```

Out[23]:

	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

In [24]:
```
car_sales["Make"]
```

Out[24]:
```
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 [25]:
```
car_sales.Colour # Won't work in case of spaces in heading
```

Out[25]:
```
0    White
1      Red
2     Blue
3    Black
4    White
5    Green
6     Blue
7     Blue
8    White
9    White
Name: Colour, dtype: object
```

In [26]:
```
car_sales[car_sales["Make"] == "Toyota"] # Apply condition on data
```

Out[26]:

	Make	Colour	Odometer (KM)	Doors	Price
0	Toyota	White	150043	4	$4,000.00
2	Toyota	Blue	32549	3	$7,000.00
5	Toyota	Green	99213	4	$4,500.00
8	Toyota	White	60000	4	$6,250.00

In [27]:
```
pd.crosstab(car_sales["Make"], car_sales["Doors"]) # Crossovers/Aggregates columr
```

Out[27]:

Doors	3	4	5
Make			
BMW	0	0	1
Honda	0	3	0
Nissan	0	2	0
Toyota	1	3	0

In [28]:
```
# Groupby
car_sales.groupby(['Make']).mean()
```

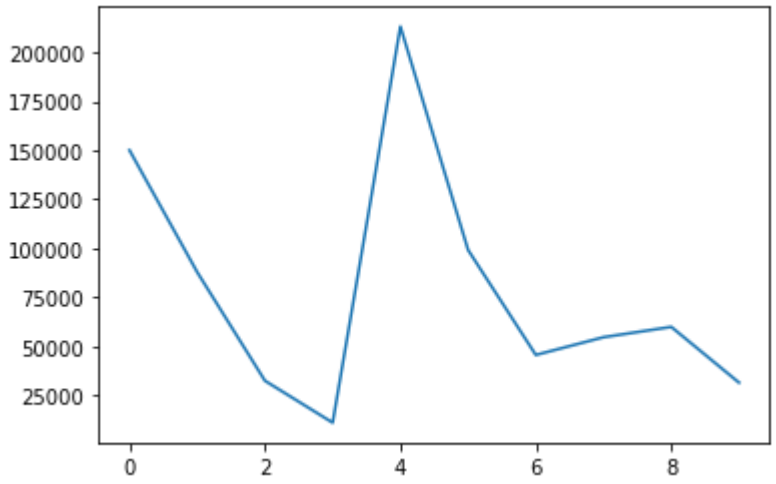
Out[28]:

Make	Odometer (KM)	Doors
BMW	11179.000000	5.00
Honda	62778.333333	4.00
Nissan	122347.500000	4.00
Toyota	85451.250000	3.75

In [29]:
```
# % is used to denote magic functions
%matplotlib inline
import matplotlib.pyplot as plt
```

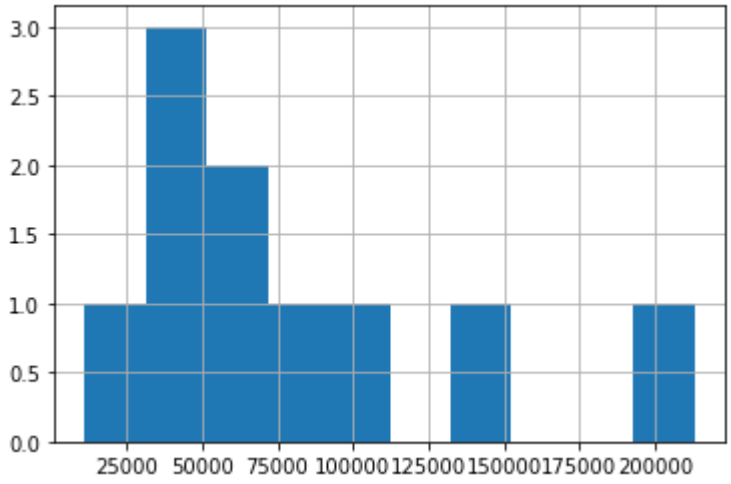
In [30]:
```
car_sales["Odometer (KM)"].plot() # Runs without matplotlib but if doesnt import
```

Out[30]: <AxesSubplot:>



In [31]:
```
car_sales["Odometer (KM)"].hist() # Histogram
```

Out[31]: <AxesSubplot:>

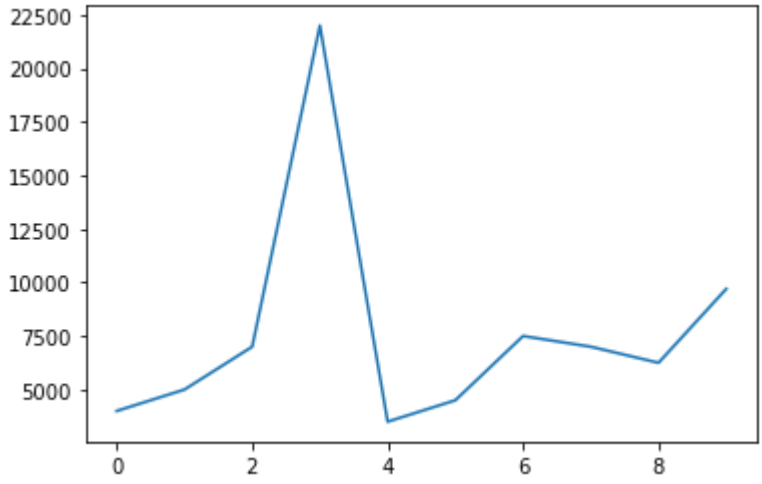


In [32]:
```
# convert price (Object datatype) to int
car_sales["Price"] = car_sales["Price"].str.replace('[\$\,\.]', '').astype(int)
car_sales ["Price"] = car_sales["Price"]/100
```

```
C:\Users\sonar\AppData\Local\Temp/ipykernel_11348/271307395.py:2: FutureWarnin
g: The default value of regex will change from True to False in a future versio
n.
  car_sales["Price"] = car_sales["Price"].str.replace('[\$\,\.]', '').astype(in
t)
```

In [33]:
```
car_sales["Price"].plot()
```

Out[33]:
```
<AxesSubplot:>
```



Manipulating Data

In [34]:
```
car_sales["Make"].str.lower() # Lowercase
```

Out[34]:
```
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 [35]:
```
car_sales_missing = pd.read_csv("car-sales-missing-data.csv")
car_sales_missing
```

Out[35]:

	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 [36]:
```
# Filling missing values with something
# Inplace changes are false by default but can be changed to true
car_sales_missing["Odometer"].fillna(car_sales_missing["Odometer"].mean(), inplac
car_sales_missing
```

Out[36]:

	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 [37]:
```
# Remove missing values
car_sales_missing.dropna(inplace = True)
car_sales_missing
```

Out[37]:

	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 [38]:
```
# Column from series
seats_column = pd.Series([5,5,5,5,5])

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

Out[38]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats
0	Toyota	White	150043	4	4000.0	5.0
1	Honda	Red	87899	4	5000.0	5.0
2	Toyota	Blue	32549	3	7000.0	5.0
3	BMW	Black	11179	5	22000.0	5.0
4	Nissan	White	213095	4	3500.0	5.0
5	Toyota	Green	99213	4	4500.0	NaN
6	Honda	Blue	45698	4	7500.0	NaN
7	Honda	Blue	54738	4	7000.0	NaN
8	Toyota	White	60000	4	6250.0	NaN
9	Nissan	White	31600	4	9700.0	NaN

In [39]:
```
car_sales["Seats"].fillna(5, inplace = True)
car_sales
```

Out[39]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats
0	Toyota	White	150043	4	4000.0	5.0
1	Honda	Red	87899	4	5000.0	5.0
2	Toyota	Blue	32549	3	7000.0	5.0
3	BMW	Black	11179	5	22000.0	5.0
4	Nissan	White	213095	4	3500.0	5.0
5	Toyota	Green	99213	4	4500.0	5.0
6	Honda	Blue	45698	4	7500.0	5.0
7	Honda	Blue	54738	4	7000.0	5.0
8	Toyota	White	60000	4	6250.0	5.0
9	Nissan	White	31600	4	9700.0	5.0

In [40]:
```
# Column from Python list
# List has to be the same length as data
fuel_economy = [7.5, 9.2, 5.0, 9.6, 8.7, 4.7, 7.6, 8.6, 3.0, 4.5]
car_sales["Fuel per 100KM"] = fuel_economy
car_sales
```

Out[40]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	Fuel per 100KM
0	Toyota	White	150043	4	4000.0	5.0	7.5
1	Honda	Red	87899	4	5000.0	5.0	9.2
2	Toyota	Blue	32549	3	7000.0	5.0	5.0
3	BMW	Black	11179	5	22000.0	5.0	9.6
4	Nissan	White	213095	4	3500.0	5.0	8.7
5	Toyota	Green	99213	4	4500.0	5.0	4.7
6	Honda	Blue	45698	4	7500.0	5.0	7.6
7	Honda	Blue	54738	4	7000.0	5.0	8.6
8	Toyota	White	60000	4	6250.0	5.0	3.0
9	Nissan	White	31600	4	9700.0	5.0	4.5

In [41]:
```
# Form new columns using data from existing ones
car_sales["Total fuel used (L)"] = car_sales["Odometer (KM)"]/100 * car_sales["Fu
car_sales
```

Out[41]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	Fuel per 100KM	Total fuel used (L)
0	Toyota	White	150043	4	4000.0	5.0	7.5	11253.225
1	Honda	Red	87899	4	5000.0	5.0	9.2	8086.708
2	Toyota	Blue	32549	3	7000.0	5.0	5.0	1627.450
3	BMW	Black	11179	5	22000.0	5.0	9.6	1073.184
4	Nissan	White	213095	4	3500.0	5.0	8.7	18539.265
5	Toyota	Green	99213	4	4500.0	5.0	4.7	4663.011
6	Honda	Blue	45698	4	7500.0	5.0	7.6	3473.048
7	Honda	Blue	54738	4	7000.0	5.0	8.6	4707.468
8	Toyota	White	60000	4	6250.0	5.0	3.0	1800.000
9	Nissan	White	31600	4	9700.0	5.0	4.5	1422.000

In [42]:
```
car_sales["Passing"] = True
car_sales
```

Out[42]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	Fuel per 100KM	Total fuel used (L)	Passing
0	Toyota	White	150043	4	4000.0	5.0	7.5	11253.225	True
1	Honda	Red	87899	4	5000.0	5.0	9.2	8086.708	True
2	Toyota	Blue	32549	3	7000.0	5.0	5.0	1627.450	True
3	BMW	Black	11179	5	22000.0	5.0	9.6	1073.184	True
4	Nissan	White	213095	4	3500.0	5.0	8.7	18539.265	True
5	Toyota	Green	99213	4	4500.0	5.0	4.7	4663.011	True
6	Honda	Blue	45698	4	7500.0	5.0	7.6	3473.048	True
7	Honda	Blue	54738	4	7000.0	5.0	8.6	4707.468	True
8	Toyota	White	60000	4	6250.0	5.0	3.0	1800.000	True
9	Nissan	White	31600	4	9700.0	5.0	4.5	1422.000	True

In [43]:
```
# Drop column
car_sales = car_sales.drop("Passing", axis = 1)
car_sales
```

Out[43]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	Fuel per 100KM	Total fuel used (L)
0	Toyota	White	150043	4	4000.0	5.0	7.5	11253.225
1	Honda	Red	87899	4	5000.0	5.0	9.2	8086.708
2	Toyota	Blue	32549	3	7000.0	5.0	5.0	1627.450
3	BMW	Black	11179	5	22000.0	5.0	9.6	1073.184
4	Nissan	White	213095	4	3500.0	5.0	8.7	18539.265
5	Toyota	Green	99213	4	4500.0	5.0	4.7	4663.011
6	Honda	Blue	45698	4	7500.0	5.0	7.6	3473.048
7	Honda	Blue	54738	4	7000.0	5.0	8.6	4707.468
8	Toyota	White	60000	4	6250.0	5.0	3.0	1800.000
9	Nissan	White	31600	4	9700.0	5.0	4.5	1422.000

In [44]:
```
# Shuffle data
# fact is percentage of data i.e 0.2 = 20%, 1= 100%
car_sales_shuffled = car_sales.sample(frac = 1)
car_sales_shuffled
```

Out[44]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	Fuel per 100KM	Total fuel used (L)
7	Honda	Blue	54738	4	7000.0	5.0	8.6	4707.468
0	Toyota	White	150043	4	4000.0	5.0	7.5	11253.225
8	Toyota	White	60000	4	6250.0	5.0	3.0	1800.000
4	Nissan	White	213095	4	3500.0	5.0	8.7	18539.265
9	Nissan	White	31600	4	9700.0	5.0	4.5	1422.000
1	Honda	Red	87899	4	5000.0	5.0	9.2	8086.708
2	Toyota	Blue	32549	3	7000.0	5.0	5.0	1627.450
3	BMW	Black	11179	5	22000.0	5.0	9.6	1073.184
5	Toyota	Green	99213	4	4500.0	5.0	4.7	4663.011
6	Honda	Blue	45698	4	7500.0	5.0	7.6	3473.048

In [45]:
```
car_sales_shuffled = car_sales_shuffled.reset_index(drop = True)
car_sales_shuffled
```

Out[45]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	Fuel per 100KM	Total fuel used (L)
0	Honda	Blue	54738	4	7000.0	5.0	8.6	4707.468
1	Toyota	White	150043	4	4000.0	5.0	7.5	11253.225
2	Toyota	White	60000	4	6250.0	5.0	3.0	1800.000
3	Nissan	White	213095	4	3500.0	5.0	8.7	18539.265
4	Nissan	White	31600	4	9700.0	5.0	4.5	1422.000
5	Honda	Red	87899	4	5000.0	5.0	9.2	8086.708
6	Toyota	Blue	32549	3	7000.0	5.0	5.0	1627.450
7	BMW	Black	11179	5	22000.0	5.0	9.6	1073.184
8	Toyota	Green	99213	4	4500.0	5.0	4.7	4663.011
9	Honda	Blue	45698	4	7500.0	5.0	7.6	3473.048

In [46]:
```
car_sales
```

Out[46]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	Fuel per 100KM	Total fuel used (L)
0	Toyota	White	150043	4	4000.0	5.0	7.5	11253.225
1	Honda	Red	87899	4	5000.0	5.0	9.2	8086.708
2	Toyota	Blue	32549	3	7000.0	5.0	5.0	1627.450
3	BMW	Black	11179	5	22000.0	5.0	9.6	1073.184
4	Nissan	White	213095	4	3500.0	5.0	8.7	18539.265
5	Toyota	Green	99213	4	4500.0	5.0	4.7	4663.011
6	Honda	Blue	45698	4	7500.0	5.0	7.6	3473.048
7	Honda	Blue	54738	4	7000.0	5.0	8.6	4707.468
8	Toyota	White	60000	4	6250.0	5.0	3.0	1800.000
9	Nissan	White	31600	4	9700.0	5.0	4.5	1422.000

In [47]:
```
# Apply function on certain column ex. lambda
car_sales["Odometer (KM)"] = car_sales["Odometer (KM)"].apply(lambda x: x/ 1.6)
car_sales
```

Out[47]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	Fuel per 100KM	Total fuel used (L)
0	Toyota	White	93776.875	4	4000.0	5.0	7.5	11253.225
1	Honda	Red	54936.875	4	5000.0	5.0	9.2	8086.708
2	Toyota	Blue	20343.125	3	7000.0	5.0	5.0	1627.450
3	BMW	Black	6986.875	5	22000.0	5.0	9.6	1073.184
4	Nissan	White	133184.375	4	3500.0	5.0	8.7	18539.265
5	Toyota	Green	62008.125	4	4500.0	5.0	4.7	4663.011
6	Honda	Blue	28561.250	4	7500.0	5.0	7.6	3473.048
7	Honda	Blue	34211.250	4	7000.0	5.0	8.6	4707.468
8	Toyota	White	37500.000	4	6250.0	5.0	3.0	1800.000
9	Nissan	White	19750.000	4	9700.0	5.0	4.5	1422.000

In [ ]:
```
```