Cars Data Analysis Project

In [ ]:
```
# For analyzing this data set of cars, we will be using the "pandas" data frame
# So in the below cell we will be importing the pandas data frame
```

In [15]:
```
import pandas as pd
```

In [22]:
```
cars = pd.read_csv("C:\Users\DELL\Desktop\Projects datasets\2. Cars Data1.csv")
```

In [23]:
```
cars
```

Out[23]:

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylinders	Horsepower	MPG_City	MPG_Highway	Weight	Wheelbase	Length
0	Acura	MDX	SUV	Asia	All	$36,945	$33,337	3.5	6.0	265	17	23	4451	106	189
1	Acura	RSX Type S 2dr	Sedan	Asia	Front	$23,820	$21,761	2.0	4.0	200	24	31	2778	101	172
2	Acura	TSX 4dr	Sedan	Asia	Front	$26,990	$24,647	2.4	4.0	200	22	29	3230	105	183
3	Acura	TL 4dr	Sedan	Asia	Front	$33,195	$30,299	3.2	6.0	270	20	28	3575	108	186
4	Acura	3.5 RL 4dr	Sedan	Asia	Front	$43,755	$39,014	3.5	6.0	225	18	24	3880	115	197
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
423	Volvo	C70 LPT convertible 2dr	Sedan	Europe	Front	$40,565	$38,203	2.4	5.0	197	21	28	3450	105	186
424	Volvo	C70 HPT convertible 2dr	Sedan	Europe	Front	$42,565	$40,083	2.3	5.0	242	20	26	3450	105	186
425	Volvo	S80 T6 4dr	Sedan	Europe	Front	$45,210	$42,573	2.9	6.0	268	19	26	3653	110	190
426	Volvo	V40	Wagon	Europe	Front	$26,135	$24,641	1.9	4.0	170	22	29	2822	101	180
427	Volvo	XC70	Wagon	Europe	All	$35,145	$33,112	2.5	5.0	208	20	27	3823	109	186

428 rows × 15 columns

In [ ]:
```
# To get the total rows and columns of the data use the .shape command
```

In [24]:
```
cars.shape
```

Out[24]:
```
(428, 15)
```

Instruction 1 (Data cleaning)

Find all the null values in the data. If there are any null values in the column, enter the mean value of that column in place of the null value.

In [ ]:
```
# First to find the null values .isnull() command should be used as follows.
```

In [26]:
```
cars.isnull()
```

Out[26]:

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylinders	Horsepower	MPG_City	MPG_Highway	Weight	Wheelbase	Length
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
423	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
424	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
425	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
426	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
427	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False

428 rows × 15 columns

In [ ]:
```
# To find the total number of null values in each column .isnull().sum() command should be used
```

In [27]:
```
cars.isnull().sum()
```

Out[27]:
```
Make          0
Model         0
Type          0
Origin        0
DriveTrain    0
MSRP          0
Invoice       0
EngineSize    0
Cylinders     2
Horsepower    0
MPG_City      0
MPG_Highway   0
Weight        0
Wheelbase     0
Length        0
dtype: int64
```

In [28]:
```
# From the above table we see that there are 2 null vaules in the cylinders columns and they have to be replaced with the
#------mean value of the cylinders column.
```

In [31]:
```
cars["Cylinders"].mean()
```

Out[31]:
```
5.807511737089202
```

In [34]:
```
cars["Cylinders"].isnull().sum()
```

Out[34]:
```
2
```

In [35]:
```
cars["Cylinders"].fillna(cars["Cylinders"].mean())
```

Out[35]:
```
0      6.0
1      4.0
2      4.0
3      6.0
4      6.0
      ... 
423    5.0
424    5.0
425    6.0
426    4.0
427    5.0
Name: Cylinders, Length: 428, dtype: float64
```

To make the permanent change in the cylinders column by changing null values with the mean value of the column, inplace = true command has to be used.

In [37]:
```
cars["Cylinders"].fillna(cars["Cylinders"].mean() , inplace = True)
```

In [40]:
```
cars.isnull().sum()
```

Out[40]:
```
Make          0
Model         0
Type          0
Origin        0
DriveTrain    0
MSRP          0
Invoice       0
EngineSize    0
Cylinders     0
Horsepower    0
MPG_City      0
MPG_Highway   0
Weight        0
Wheelbase     0
Length        0
dtype: int64
```

2) question(based on value counts)

Check the different type of Make that are available in the data set and what is the count(occurence) of each Make in the data set.

In [52]:
```
cars["Make"].value_counts()
```

Out[52]:
```
Toyota           28
Chevrolet        27
Mercedes-Benz    26
Ford             23
BMW              20
Dodge            17
Honda            17
Nissan           17
Volkswagen       15
Chrysler         15
Dodge            13
Mitsubishi       13
Audi             13
Jaguar           12
Hyundai          12
Subaru           11
Pontiac          11
Mazda            10
Lexus            10
Kia               9
Buick             9
Mercury           9
Lincoln           9
Saturn            8
Saab              8
Suzuki            8
Infiniti          8
GMC               8
Acura             7
Porsche           7
Jaguar            7
Land Rover        3
Oldsmobile        3
Jeep              3
Scion             2
Isuzu             2
MINI              2
Hummer            2
Name: Make, dtype: int64
```

3) Instruction(Filtering)

Show all the records where origin is Asia or Europe

In [59]:
```
cars.head(2)
```

Out[59]:

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylinders	Horsepower	MPG_City	MPG_Highway	Weight	Wheelbase	Length
0	Acura	MDX	SUV	Asia	All	$36,945	$33,337	3.5	6.0	265	17	23	4451	106	189
1	Acura	RSX Type S 2dr	Sedan	Asia	Front	$23,820	$21,761	2.0	4.0	200	24	31	2778	101	172

In [60]:
```
cars[(cars["Origin"] == "Asia") | (cars["Origin"] == "Europe")]
```

Out[60]:

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylinders	Horsepower	MPG_City	MPG_Highway	Weight	Wheelbase	Length
0	Acura	MDX	SUV	Asia	All	$36,945	$33,337	3.5	6.0	265	17	23	4451	106	189
1	Acura	RSX Type S 2dr	Sedan	Asia	Front	$23,820	$21,761	2.0	4.0	200	24	31	2778	101	172
2	Acura	TSX 4dr	Sedan	Asia	Front	$26,990	$24,647	2.4	4.0	200	22	29	3230	105	183
3	Acura	TL 4dr	Sedan	Asia	Front	$33,195	$30,299	3.2	6.0	270	20	28	3575	108	186
4	Acura	3.5 RL 4dr	Sedan	Asia	Front	$43,755	$39,014	3.5	6.0	225	18	24	3880	115	197
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
423	Volvo	C70 LPT convertible 2dr	Sedan	Europe	Front	$40,565	$38,203	2.4	5.0	197	21	28	3450	105	186
424	Volvo	C70 HPT convertible 2dr	Sedan	Europe	Front	$42,565	$40,083	2.3	5.0	242	20	26	3450	105	186
425	Volvo	S80 T6 4dr	Sedan	Europe	Front	$45,210	$42,573	2.9	6.0	268	19	26	3653	110	190
426	Volvo	V40	Wagon	Europe	Front	$26,135	$24,641	1.9	4.0	170	22	29	2822	101	180
427	Volvo	XC70	Wagon	Europe	All	$35,145	$33,112	2.5	5.0	208	20	27	3823	109	186

281 rows × 15 columns

In [ ]:
```
# Other method is using the .isin() as follows
```

In [72]:
```
cars[(cars["Origin"]).isin(["Asia" , "Europe"])]
```

Out[72]:

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylinders	Horsepower	MPG_City	MPG_Highway	Weight	Wheelbase	Length
0	Acura	MDX	SUV	Asia	All	$36,945	$33,337	3.5	6.0	265	17	23	4451	106	189
1	Acura	RSX Type S 2dr	Sedan	Asia	Front	$23,820	$21,761	2.0	4.0	200	24	31	2778	101	172
2	Acura	TSX 4dr	Sedan	Asia	Front	$26,990	$24,647	2.4	4.0	200	22	29	3230	105	183
3	Acura	TL 4dr	Sedan	Asia	Front	$33,195	$30,299	3.2	6.0	270	20	28	3575	108	186
4	Acura	3.5 RL 4dr	Sedan	Asia	Front	$43,755	$39,014	3.5	6.0	225	18	24	3880	115	197
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
423	Volvo	C70 LPT convertible 2dr	Sedan	Europe	Front	$40,565	$38,203	2.4	5.0	197	21	28	3450	105	186
424	Volvo	C70 HPT convertible 2dr	Sedan	Europe	Front	$42,565	$40,083	2.3	5.0	242	20	26	3450	105	186
425	Volvo	S80 T6 4dr	Sedan	Europe	Front	$45,210	$42,573	2.9	6.0	268	19	26	3653	110	190
426	Volvo	V40	Wagon	Europe	Front	$26,135	$24,641	1.9	4.0	170	22	29	2822	101	180
427	Volvo	XC70	Wagon	Europe	All	$35,145	$33,112	2.5	5.0	208	20	27	3823	109	186

281 rows × 15 columns

4) Instruction - Removing unwanted rocords

Remove all the records(rows) where weight is above 4000.

In [ ]:
```
# First to get all the records of where the weight is above 4000 - following code is written
```

In [79]:
```
cars.head(2)
```

Out[79]:

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylinders	Horsepower	MPG_City	MPG_Highway	Weight	Wheelbase	Length
0	Acura	MDX	SUV	Asia	All	$36,945	$33,337	3.5	6.0	265	17	23	4451	106	189
1	Acura	RSX Type S 2dr	Sedan	Asia	Front	$23,820	$21,761	2.0	4.0	200	24	31	2778	101	172

In [74]:
```
cars[cars["Weight"] > 4000]
```

Out[74]:

	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylinders	Horsepower	MPG_City	MPG_Highway	Weight	Wheelbase	Length
0	Acura	MDX	SUV	Asia	All	$36,945	$33,337	3.5	6.0	265	17	23	4451	106	189
15	Audi	A4 3.0 Quattro convertible 2dr	Sedan	Europe	All	$44,240	$40,075	3.0	6.0	220	18	25	4013	105	180
17	Audi	A6 4.2 Quattro 4dr	Sedan	Europe	All	$49,690	$44,936	4.2	8.0	300	17	24	4024	109	193
18	Audi	A8 L Quattro 4dr	Sedan	Europe	All	$69,190	$64,740	4.2	8.0	330	17	24	4399	121	204
20	Audi	RS 6 4dr	Sports	Europe	Front	$84,600	$76,417	4.2	8.0	450	15	22	4024	109	191
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
401	Volkswagen	Touareg V6	SUV	Europe	All	$35,515	$32,243	3.2	6.0	220	15	20	5086	112	187
411	Volkswagen	Phaeton 4dr	Sedan	Europe	Front	$65,000	$59,912	4.2	8.0	335	16	22	5194	118	204
412	Volkswagen	Phaeton W12 4dr	Sedan	Europe	Front	$75,000	$69,130	6.0	12.0	420	12	19	5399	118	204
426	Volkswagen	Passat W8	Wagon	Europe	Front	$40,235	$36,956	4.0	8.0	270	18	25	4067	106	194
458	Volvo	XC90 T6	SUV	Europe	All	$41,250	$38,851	2.9	6.0	268	15	20	4638	113	189

103 rows × 15 columns

In [76]:
```
# The method used to remove the records from the data is by attching the '~' symbol before the syntax
```

In [77]:
```
cars.head(2)
```

Out[77]:

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylinders	Horsepower	MPG_City	MPG_Highway	Weight	Wheelbase	Length
0	Acura	MDX	SUV	Asia	All	$36,945	$33,337	3.5	6.0	265	17	23	4451	106	189
1	Acura	RSX Type S 2dr	Sedan	Asia	Front	$23,820	$21,761	2.0	4.0	200	24	31	2778	101	172

In [78]:
```
cars[~(cars["weight"] > 4000)]
```

Out[78]:

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylinders	Horsepower	MPG_City	MPG_Highway	Weight	Wheelbase	Length
1	Acura	RSX Type S 2dr	Sedan	Asia	Front	$23,820	$21,761	2.0	4.0	200	24	31	2778	101	172
2	Acura	TSX 4dr	Sedan	Asia	Front	$26,990	$24,647	2.4	4.0	200	22	29	3230	105	183
3	Acura	TL 4dr	Sedan	Asia	Front	$33,195	$30,299	3.2	6.0	270	20	28	3575	108	186
4	Acura	3.5 RL 4dr	Sedan	Asia	Front	$43,755	$39,014	3.5	6.0	225	18	24	3880	115	197
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
423	Volvo	C70 LPT convertible 2dr	Sedan	Europe	Front	$40,565	$38,203	2.4	5.0	197	21	28	3450	105	186
424	Volvo	C70 HPT convertible 2dr	Sedan	Europe	Front	$42,565	$40,083	2.3	5.0	242	20	26	3450	105	186
425	Volvo	S80 T6 4dr	Sedan	Europe	Front	$45,210	$42,573	2.9	6.0	268	19	26	3653	110	190
426	Volvo	V40	Wagon	Europe	Front	$26,135	$24,641	1.9	4.0	170	22	29	2822	101	180
427	Volvo	XC70	Wagon	Europe	All	$35,145	$33,112	2.5	5.0	208	20	27	3823	109	186

325 rows × 15 columns

In [81]:
```
cars
```

Out[81]:

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylinders	Horsepower	MPG_City	MPG_Highway	Weight	Wheelbase	Length
0	Acura	MDX	SUV	Asia	All	$36,945	$33,337	3.5	6.0	265	17	23	4451	106	189
1	Acura	RSX Type S 2dr	Sedan	Asia	Front	$23,820	$21,761	2.0	4.0	200	24	31	2778	101	172
2	Acura	TSX 4dr	Sedan	Asia	Front	$26,990	$24,647	2.4	4.0	200	22	29	3230	105	183
3	Acura	TL 4dr	Sedan	Asia	Front	$33,195	$30,299	3.2	6.0	270	20	28	3575	108	186
4	Acura	3.5 RL 4dr	Sedan	Asia	Front	$43,755	$39,014	3.5	6.0	225	18	24	3880	115	197
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
423	Volvo	C70 LPT convertible 2dr	Sedan	Europe	Front	$40,565	$38,203	2.4	5.0	197	21	28	3450	105	186
424	Volvo	C70 HPT convertible 2dr	Sedan	Europe	Front	$42,565	$40,083	2.3	5.0	242	20	26	3450	105	186
425	Volvo	S80 T6 4dr	Sedan	Europe	Front	$45,210	$42,573	2.9	6.0	268	19	26	3653	110	190
426	Volvo	V40	Wagon	Europe	Front	$26,135	$24,641	1.9	4.0	170	22	29	2822	101	180
427	Volvo	XC70	Wagon	Europe	All	$35,145	$33,112	2.5	5.0	208	20	27	3823	109	186

428 rows × 15 columns