



**Bharatiya Vidya Bhavan's  
Sardar Patel Institute of Technology**

Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058-India  
(Autonomous College Affiliated to University of Mumbai)

**BE-ETRX**

**UID:2019110039**

**Sub-AIML**

**NAME: Devansh Palliyath**

**Exploratory Data Analysis**

Aim: To perform Exploratory Data Analysis on a data set of cars in India in the year 2021.

Objective:

1. To gather a data set.
2. Perform cleaning on the data.
3. Analyze the data.
4. To find the relationship between the attributes of the dataset.

Software Used: Google Colab.

Dataset Link:

<https://www.kaggle.com/datasets/medhekarabhinav5/indian-cars-dataset>

Steps:

## 1. Understand the data

```
data.head()
```

Unnamed: 0	Make	Model	Variant	Ex-Showroom_Price	Displacement	Cylinders	Valves_Per_Cylinder	Drivetrain	Cylinder_Configuration	...	Leather_Wrapped_Steering	Automatic_Headlamps
0	0	Tata	Nano Genx	Rs. 2,92,667	624 cc	2.0	2.0	RWD (Rear Wheel Drive)	In-line	...	NaN	NaN
1	1	Tata	Nano Genx	Rs. 2,36,447	624 cc	2.0	2.0	RWD (Rear Wheel Drive)	In-line	...	NaN	NaN
2	2	Tata	Nano Genx	Rs. 2,96,661	624 cc	2.0	2.0	RWD (Rear Wheel Drive)	In-line	...	NaN	NaN
3	3	Tata	Nano Genx	Rs. 3,34,768	624 cc	2.0	2.0	RWD (Rear Wheel Drive)	In-line	...	NaN	NaN
4	4	Tata	Nano Genx	Rs. 2,72,223	624 cc	2.0	2.0	RWD (Rear Wheel Drive)	In-line	...	NaN	NaN

5 rows x 141 columns

This prints all the columns along with only the first 5 rows of the data set.

## 2. data.tail()

```
data.tail()
```

Variant	Ex-Showroom_Price	Displacement	Cylinders	Valves_Per_Cylinder	Drivetrain	Cylinder_Configuration	...	Leather_Wrapped_Steering	Automatic_Headlamps	Engine_Type	ASR/_Tractio
Vx Mt Diesel	Rs. 13,02,000	1498 cc	4.0	4.0	FWD (Front Wheel Drive)	In-line	...	Yes	NaN	NaN	
Zx Mt Diesel	Rs. 14,21,000	1498 cc	4.0	4.0	FWD (Front Wheel Drive)	In-line	...	Yes	Yes	NaN	
Zx Cvt Petrol	Rs. 14,31,000	1497 cc	4.0	4.0	FWD (Front Wheel Drive)	In-line	...	Yes	Yes	NaN	
V Cvt Petrol	Rs. 12,01,000	1497 cc	4.0	4.0	FWD (Front Wheel Drive)	In-line	...	NaN	NaN	NaN	
3.2 At	Rs. 68,62,560	3200 cc	4.0	4.0	AWD (All Wheel Drive)	In-line	...	Yes	NaN	NaN	

Prints bottom five rows.

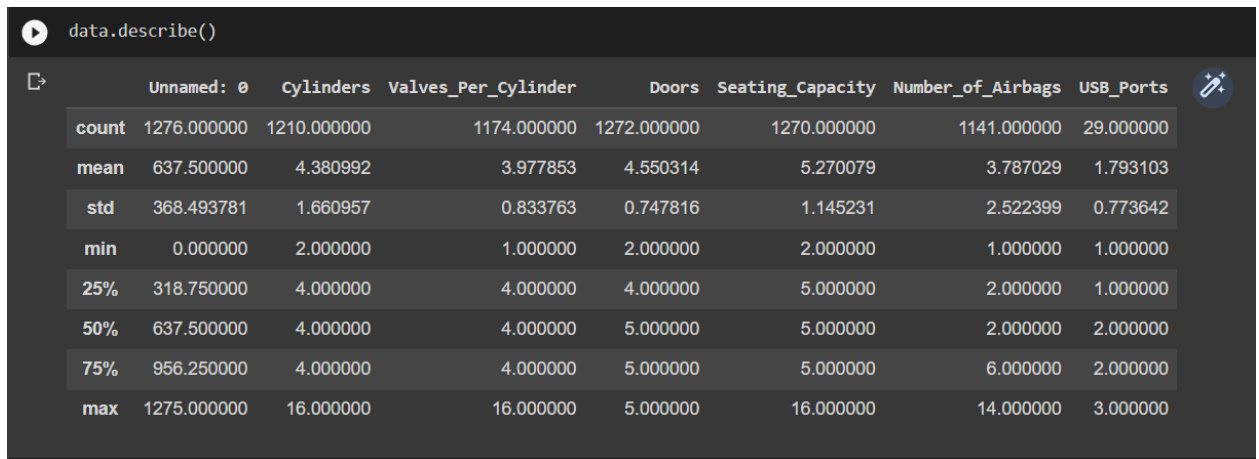
## 3. data.shape

```
data.shape
```

```
(1276, 141)
```

Prints the total number of rows and columns of the data set.

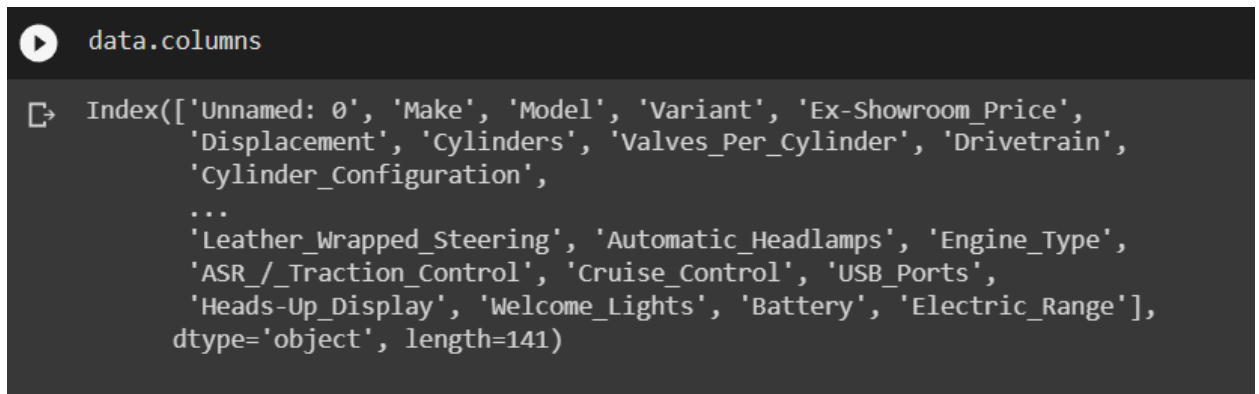
#### 4. data.describe()



	Unnamed: 0	Cylinders	Valves_Per_Cylinder	Doors	Seating_Capacity	Number_of_Airbags	USB_Ports
count	1276.000000	1210.000000	1174.000000	1272.000000	1270.000000	1141.000000	29.000000
mean	637.500000	4.380992	3.977853	4.550314	5.270079	3.787029	1.793103
std	368.493781	1.660957	0.833763	0.747816	1.145231	2.522399	0.773642
min	0.000000	2.000000	1.000000	2.000000	2.000000	1.000000	1.000000
25%	318.750000	4.000000	4.000000	4.000000	5.000000	2.000000	1.000000
50%	637.500000	4.000000	4.000000	5.000000	5.000000	2.000000	2.000000
75%	956.250000	4.000000	4.000000	5.000000	5.000000	6.000000	2.000000
max	1275.000000	16.000000	16.000000	5.000000	16.000000	14.000000	3.000000

It shows the data inside the dataset along with which data is the max and min also standard deviation, mean etc.

#### 5. data.columns



```
data.columns
Index(['Unnamed: 0', 'Make', 'Model', 'Variant', 'Ex-Showroom_Price',
      'Displacement', 'Cylinders', 'Valves_Per_Cylinder', 'Drivetrain',
      'Cylinder_Configuration',
      ...,
      'Leather_Wrapped_Steering', 'Automatic_Headlamps', 'Engine_Type',
      'ASR_/Traction_Control', 'Cruise_Control', 'USB_Ports',
      'Heads-Up_Display', 'Welcome_Lights', 'Battery', 'Electric_Range'],
      dtype='object', length=141)
```

It shows all the columns of the data set along with there names.

## 6. data.nunique()

```
data.nunique()

Unnamed: 0      1276
Make            39
Model           263
Variant         1064
Ex-Showroom_Price 1179
...
USB_Ports        3
Heads-Up_Display 1
Welcome_Lights   1
Battery          5
Electric_Range   8
Length: 141, dtype: int64
```

It is used in showing how many unique values a particular attribute or a feature has.

## 7. data['Model'].unique

```
data['Model'].unique()

array(['Nano Genx', 'Redi-Go', 'Kwid', 'Eeco', 'Alto K10', 'Go',
      'Celerio Tour', 'Santro', 'Tiago', 'Celerio X', 'Ignis', 'Triber',
      'Rio', 'Etios Liva', 'Micra Active', 'Bolt', 'Xcent Prime',
      'Dzire Tour', 'Elite I20', 'Aura', 'Polo', 'Dzire', 'Freestyle',
      'Ameo', 'Aspire', 'Platinum Etios', 'Etios Cross', 'Verito Vibe',
      'Urban Cross', 'Glanza', 'Avventura', 'Jazz', 'Compass Trailhawk',
      'Mu-X', 'Alturas G4', 'Tiguan', 'Cr-V', 'Superb Sportline', 'A3',
      'Mercedes-Benz B-Class', 'Mercedes-Benz Cla-Class', 'Kodiaq',
      'Avanti', 'Q3', 'Cooper 5 Door', 'Convertible', 'Xc40', 'Clubman',
      'A4', 'John Cooper Works', 'Xe', 'Xf', 'A3 Cabriolet', 'A6', 'X3',
      'Discovery Sport', 'S90', 'S5', 'X5', 'Mustang', 'Grand Cherokee',
      'Mercedes-Benz E-Class Cabriolet', 'M2 Competition', '718',
      'Mercedes-Benz Gls', 'Land Cruiser Prado', 'Rx 450H', 'Rs5',
      '7-Series', 'Q8', 'Mercedes-Benz S-Class', 'Levante',
      'Mercedes-Benz G-Class', 'A8 L', 'Granturismo', 'Quattroporte',
      'Lc 500H', 'Mercedes-Benz Maybach', 'Panamera', 'Lx 450D',
      'Mercedes-Benz S-Class Cabriolet', 'R8', 'Urus', 'Continental Gt',
      'Portofino', 'Bentayga', 'Db 11', '458 Speciale',
      'Rolls-Royce Ghost Series II', 'Rolls-Royce Wraith', 'Mulsanne',
      'Rolls-Royce Cullinan', 'Rolls-Royce Phantom Coupe', 'Chiron',
      'Qute (Re60)', 'Alto', 'S-Presso', 'Celerio', 'Grand I10 Prime',
      'Kuv100 Nxt', 'Swift', 'Altroz', 'Extreme', 'Tigor', 'Zest',
      'Amaze', 'Gypsy', 'Venue', 'Nexon', 'Linea', 'Bolero Power Plus',
      'Vitara Brezza', 'I20 Active', 'Ecosport', 'Duster', 'Verna',
      'Xuv300', 'Lodgy', 'Vento', 'E20 Plus', 'Tigor EV', 'Brv', 'Thar',
      'Gurkha', 'Xl6', 'Abarth Avventura', 'Tuv300 Plus', 'Marazzo',
      'Scorpio', 'Monte Carlo', 'Xuv500', 'E Verito', 'Hexa',
      'Innova Crysta', 'Compass', 'Corolla Altis', 'Civic', 'Zs EV',
      'Carnival', 'Superb', 'V40', 'Fortuner', 'Endeavour',
      'Cooper 3 Door', 'Kodiaq Scout', 'X1', 'S60', '3-Series']
```

It shows the total number of different values that the column 'Model' has.

Cleaning the Data:

8. `data.isnull().sum()`

```
data.isnull().sum()

Unnamed: 0      0
Make           75
Model          0
Variant        0
Ex-Showroom_Price  0
...
USB_Ports     1247
Heads-Up_Display 1225
Welcome_Lights 1207
Battery       1263
Electric_Range 1259
Length: 141, dtype: int64
```

It shows the total number of null values in the data set along with the columns in which the null values are present.

9. `data1=data.fillna(method='pad')`

This is used to replace the null values by the values present before them in the data set.

10. `data1.isnull().sum()`

```
data1.isnull().sum()

Unnamed: 0      0
Make           0
Model          0
Variant        0
Ex-Showroom_Price  0
...
USB_Ports      275
Heads-Up_Display 219
Welcome_Lights  219
Battery        615
Electric_Range  319
Length: 141, dtype: int64
```

This shows that the null values have been reduced compared to the earlier values.

11. `data2=data1.dropna()`

This is used to drop the remaining rows containing the null values.

12. `data2.isnull().sum()`

```
[ ] data2.isnull().sum()

Unnamed: 0      0
Make           0
Model          0
Variant        0
Ex-Showroom_Price  0
..
USB_Ports      0
Heads-Up_Display  0
Welcome_Lights  0
Battery        0
Electric_Range  0
Length: 141, dtype: int64
```

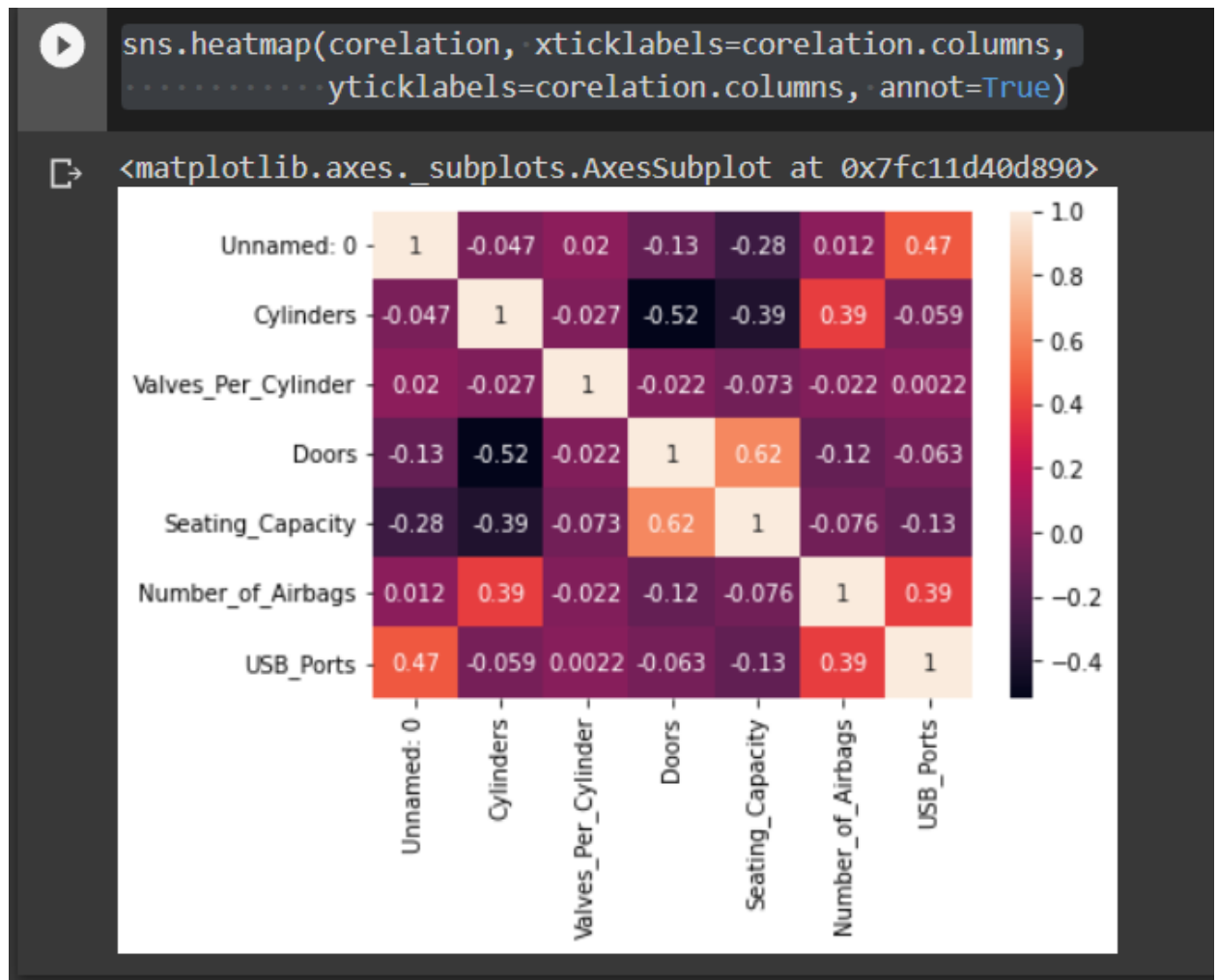
This shows that we have handled the null values successfully.  
And there are no more null values present in this dataset.

Relationship Analysis:

13. `correlation=data2.corr()`

This is used to form the correlation between the columns.

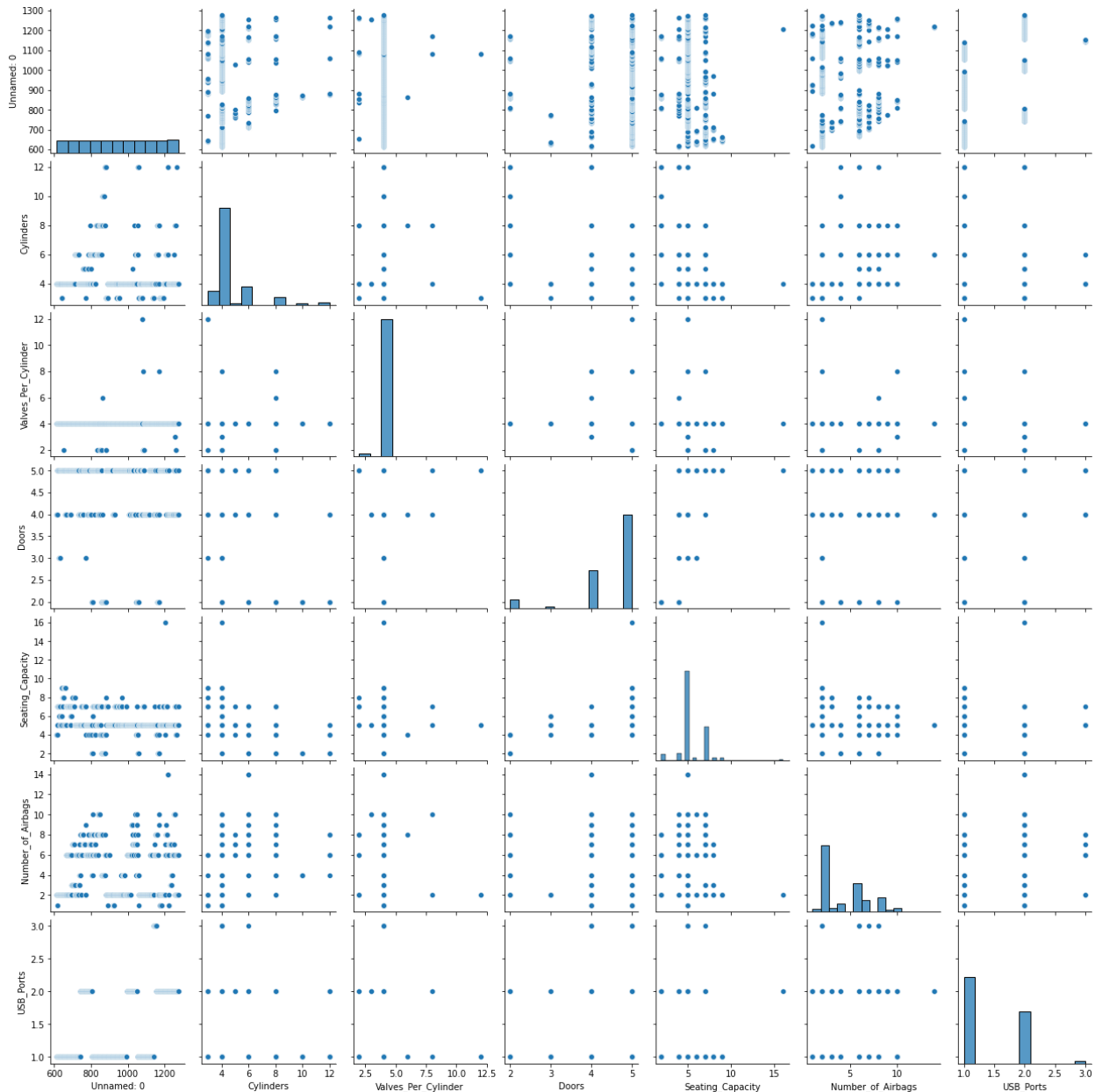
HeatMap:



A heat map helps you visualize density. And in the case of web design and analysis, it helps you visualize how far people scroll on your site, where they click and even sometimes where they're looking.

PairPlot:

```
sns.pairplot(data2)
```

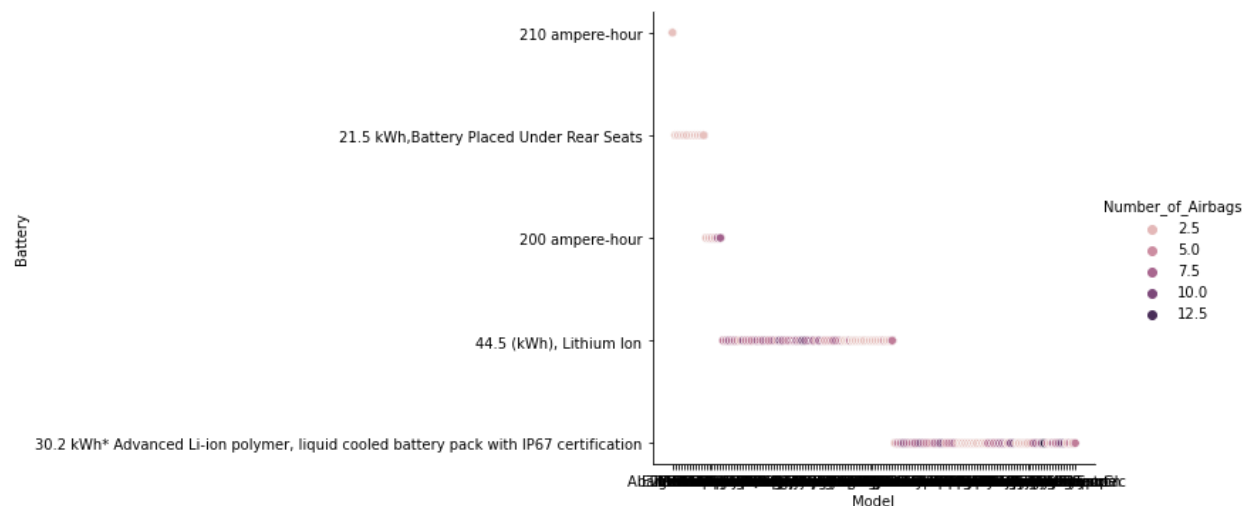


Pair plot is used to understand the best set of features to explain a relationship between two variables or to form the most separated clusters. It also helps to form some simple classification models by drawing some simple lines or making linear separation in our data-set.



Relation Plot:

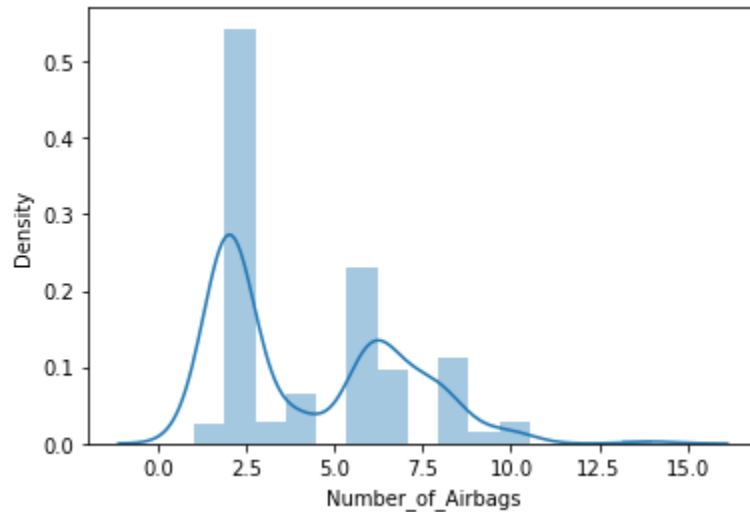
```
sns.relplot(x='Model', y='Battery', hue='Number_of_Airbags', data=data2)
```



The Seaborn Relational Plot (relplot) allows us to visualize how variables within a dataset relate to each other. Data visualization is an essential part of any data analysis or machine learning workflow. It allows us to gain insights about our data. This plot shows the relationship between model and battery of the vehicle. Also the number of airbags. For eq there is a single model having 210 ampere hour battery containing only one type of airbag which is 2.5. Models containing the lithium ion batteries have many airbag options.

Dist Plot:

```
sns.distplot(data2['Number_of_Airbags'])
```



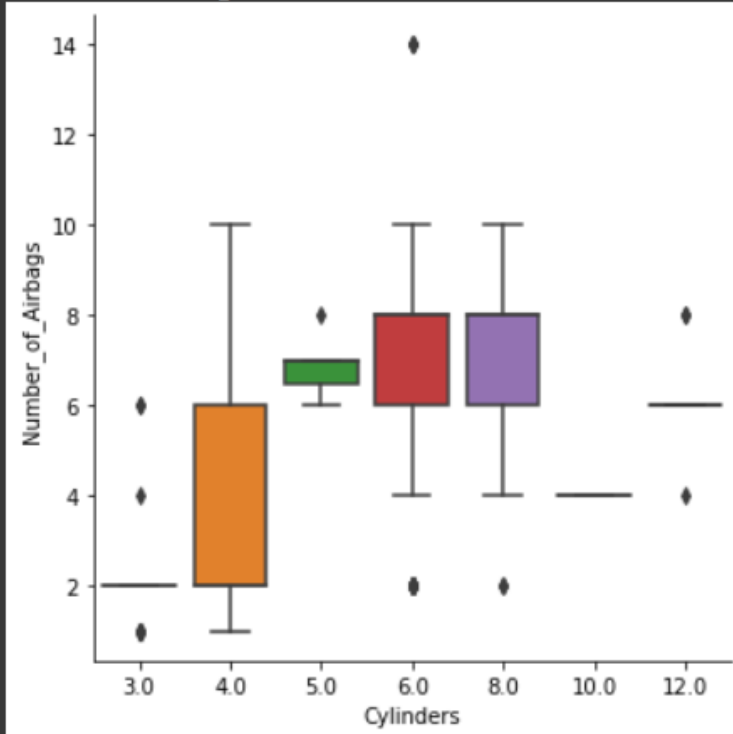
Seaborn Distplot represents the overall distribution of continuous data variables. The Seaborn module along with the Matplotlib module is used to depict the distplot with different variations in it. The Distplot depicts the data by a histogram and a line in combination to it.

This plot shows that the 2.5 airbags option in the cars is far more than any other airbag option. Cars having 10 airbags are very few.

## Category Plot (Box Plot):

```
sns.catplot(x='Cylinders',y='Number_of_Airbags',kind='box',data=data2)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f75b503bd90>
```



link code. Catplot can handle 8 different plots currently available in Seaborn. catplot function can do all these types of plots and one can specify the type of plot one needs with the kind parameter. So it is kind of one stop for every plot you will require for bivariate analysis. A small rectangular box is drawn with a line representing the median, while the top and bottom of the box represent the 75th and 25th percentiles (3rd and 1st quartiles), respectively. If the median is not in the middle of the box the distribution is skewed.

## Outcomes:

1. I understand that whenever we gather a dataset for any machine learning application, the first and very important thing that we need to do is to make the dataset simple and straightforward by removing the nullities and redundancies.
2. We can also delete the rows which contain the null values because the null values anyways wouldn't be useful for our applications because we first need to train the ML model and then test successfully.
3. By training and testing the ML model we need to develop a model which will provide a very good prediction accuracy.
4. We can plot the data in the dataset as we want and how exactly we need to see the data.
5. EDA is very important as it is the basic and the most important process in ML because as said initially this data that we filter and make clean is used to train and test the model.