

Assignment No. 4

Title: Visualization of the Netflix Dataset

Aim: To plot interactive visualizations using the netflix dataset.

Problem Statement: Use Netflix Movies and TV Shows dataset from Kaggle and perform following operation :

1. Make a visualization showing the total number of movies watched by child.
2. Make a visualization showing the total number of standup comedies .
3. Make a visualization showing most watched shows.
4. Make a visualization showing highest rated show Make a dashboard (DASHBOARD A) containing all of these above visualizations.

Theory:

For Visualizing we use basically two different libraries Matplotlib and Seaborn.

Matplotlib: Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy .Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in 2002.

One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

Seaborn: Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on the top of matplotlib library and also closely integrated to the data structures from pandas. Seaborn aims to make visualization the central part of exploring and understanding data. It provides dataset-oriented APIs, so that we can switch between different visual representations for same variables for better understanding of dataset.

Data Visualization using Matplotlib:

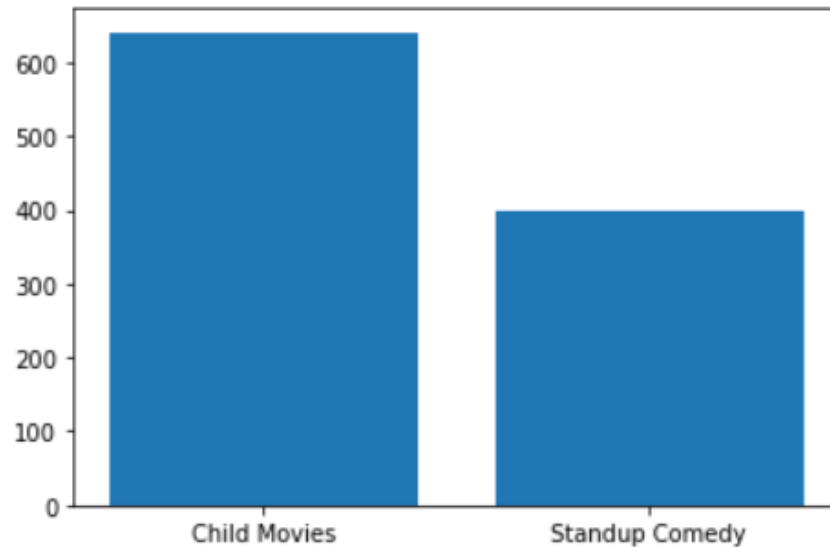
1. Bar Plot:

Syntax- `plt.bar(x, height, width, bottom, align)`

In the Netflix dataset for visualizing the total number of movies watched by children we use bar plot.

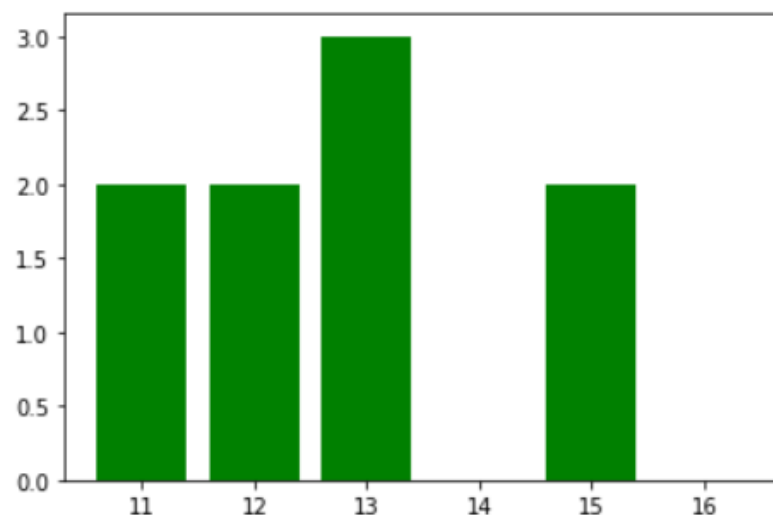
Code Snippet:

```
In [17]: import matplotlib.pyplot as plt
plt.bar(['Child Movies','Standup Comedy'],
        [total_child, standup_comedies])
plt.show()
```



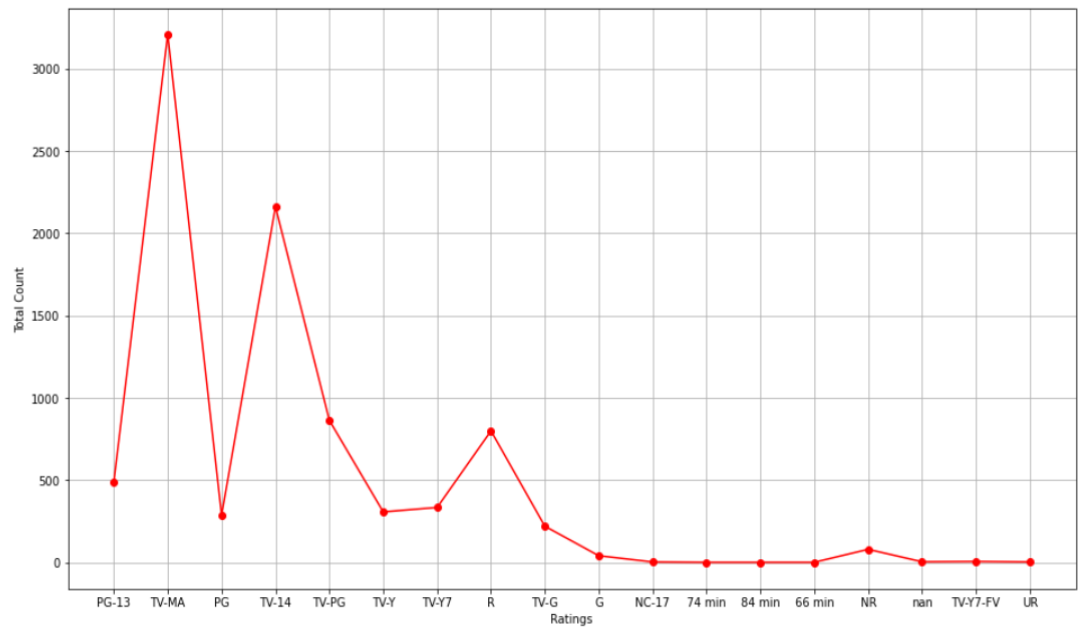
```
In [21]: plt.bar([11,12,13,15,16],
                  [len(seasons11),len(seasons12),len(seasons13),
                   len(seasons15),len(seasons16)],
                  color='green')
```

Out[21]: <BarContainer object of 5 artists>

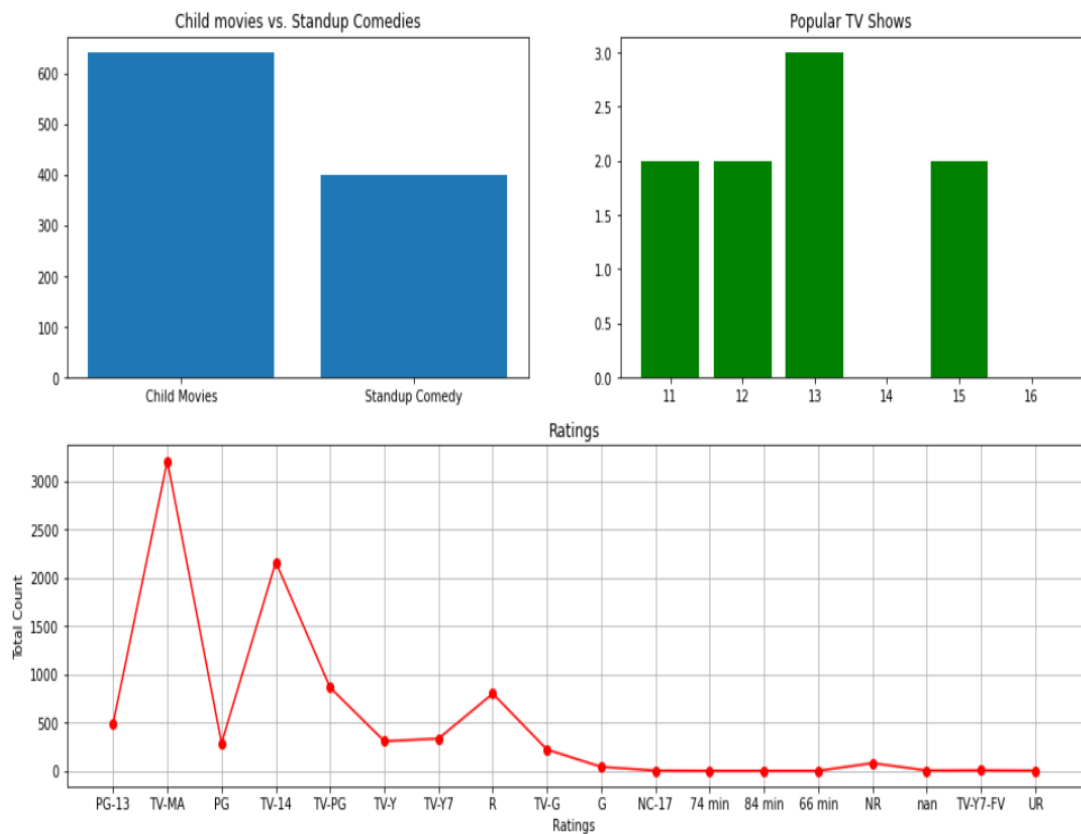


2. Line Plot: We have used line plot for getting the ratings of the movie.

```
In [27]: plt.figure(figsize=(16,9))
plt.plot(ratings.keys(), ratings.values(), color = 'red', marker='o')
plt.xlabel('Ratings'); plt.ylabel('Total Count')
plt.grid()
```



For dashboard we plot all the plots in one frame:



Conclusion: We have successfully performed the visualization tasks and made a dashboard depicting the same.

Assignment 5

Title: Visualization using seaborn library

Aim: To plot interactive visualizations using the seaborn library.

Problem Statement: Create following visualizations(ANY 4) using **SEABORN** for any dataset of your choice and draw appropriate conclusions

- a. scatterplot
- b. countplot
- c. displot
- d. KDEplot
- e. Regplot

Theory:

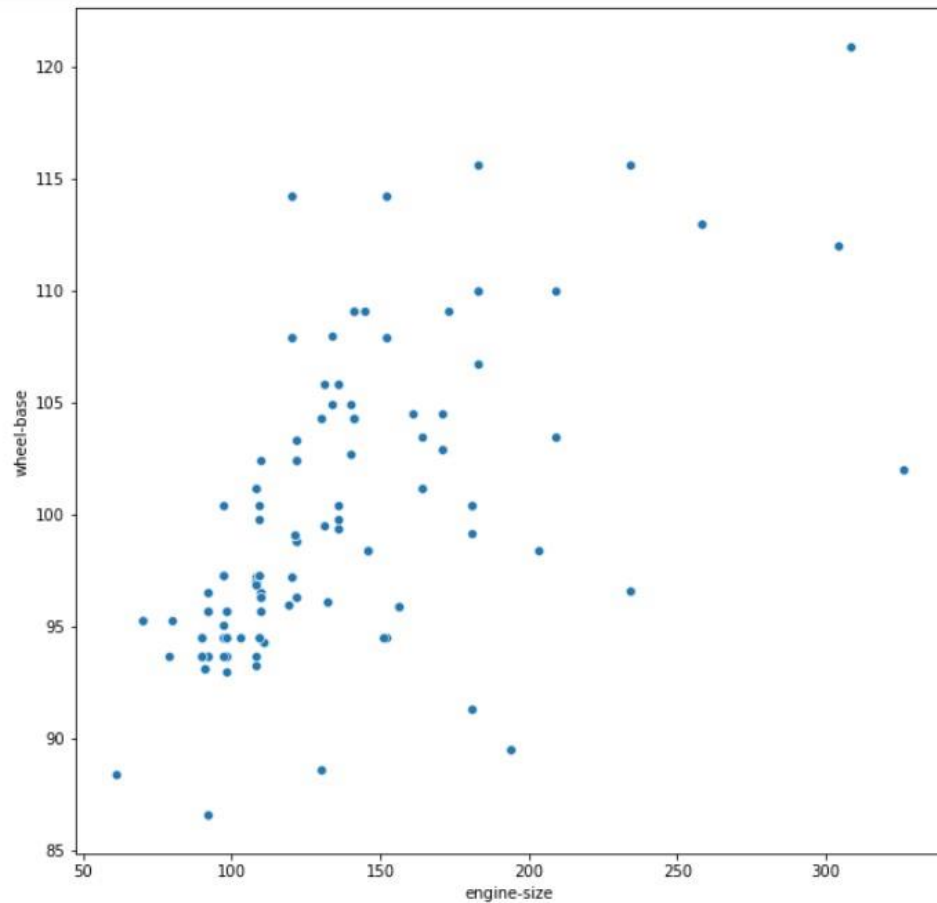
Seaborn: Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and colour palettes to make statistical plots more attractive. It is built on the top of matplotlib library and also closely integrated to the data structures from pandas. Seaborn aims to make visualization the central part of exploring and understanding data. It provides dataset-oriented APIs, so that we can switch between different visual representations for same variables for better understanding of dataset.

- A. Scatter Plot:- **Scatterplot** can be used with several semantic groupings which can help to understand well in a graph. They can plot two-dimensional graphics that can be enhanced by mapping up to three additional variables while using the semantics of hue, size, and style parameters. All the parameter control visual semantic which are used to identify the different subsets. Using redundant semantics can be helpful for making graphics more accessible.

Syntax:

```
seaborn.scatterplot(x=None, y=None, hue=None, style=None, size=None,
data=None, palette=None, hue_order=None, hue_norm=None, sizes=None,
size_order=None, size_norm=None, markers=True, style_order=None,
x_bins=None, y_bins=None, units=None, estimator=None, ci=95, n_boot=1000,
alpha='auto', x_jitter=None, y_jitter=None, legend='brief', ax=None, **kwargs)
```

Code Snippet:

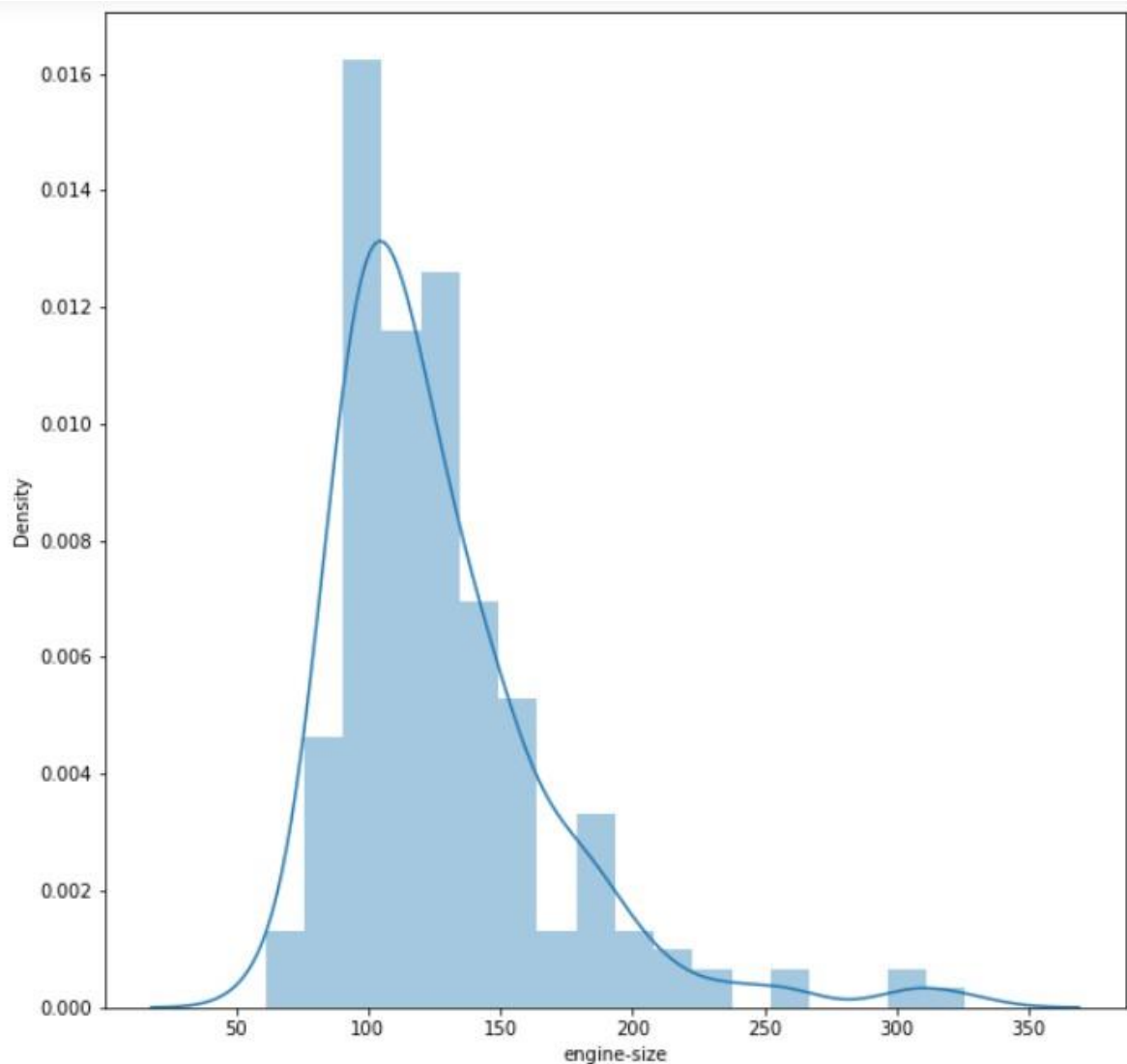


- B. **Distplot:** It is used basically for univariant set of observations and visualizes it through a histogram i.e. only one observation and hence we choose one particular column of the dataset.

Syntax:

`Sns.distplot(a[, bins, hist, kde, rug, fit, ...])`

Code Snippet:

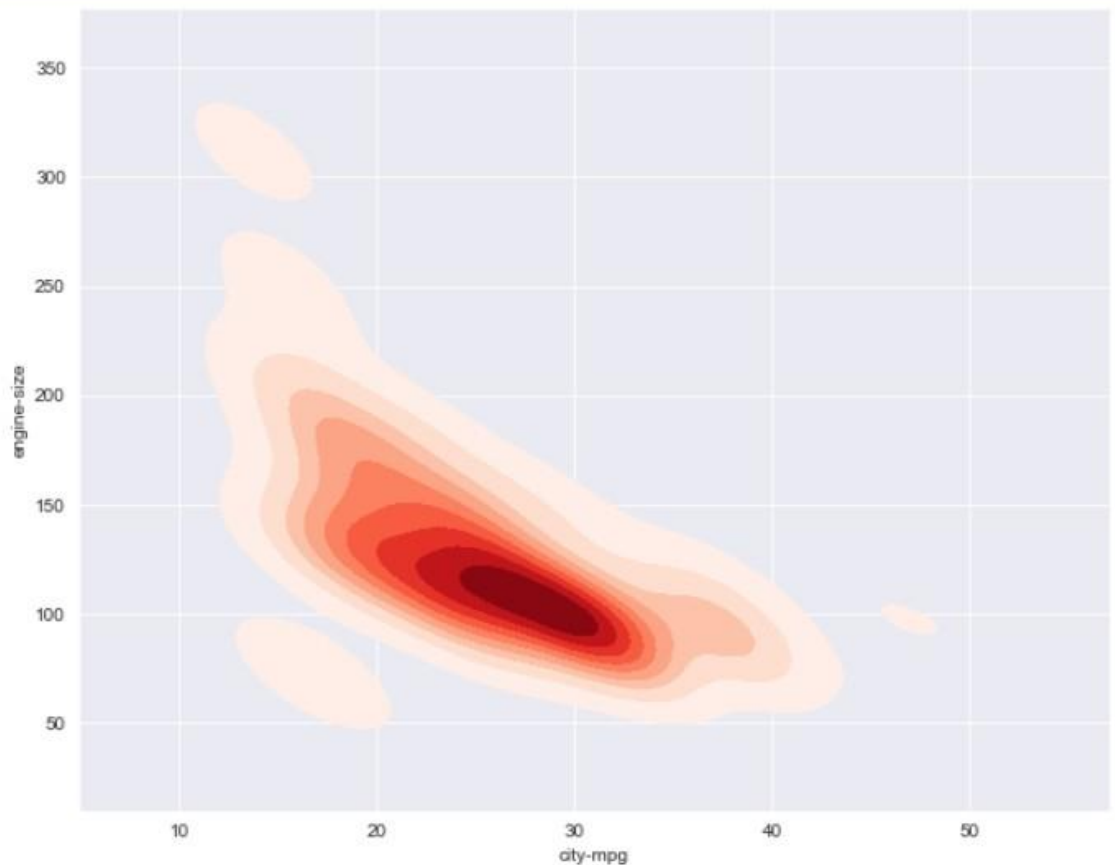


- C. KDE Plot: Kdeplot allows us to estimate the probability density function of the continuous or non-parametric from our data set curve in one or more dimensions it means we can create plot a single graph for multiple samples which helps in more efficient data visualization.

Syntax:

```
seaborn.kdeplot(x=None, *, y=None, vertical=False, palette=None, **kwargs)
```

Code Snippet:



- D. Reg Plot: This method is used to plot data and a linear regression model fit. There are a number of mutually exclusive options for estimating the regression model.

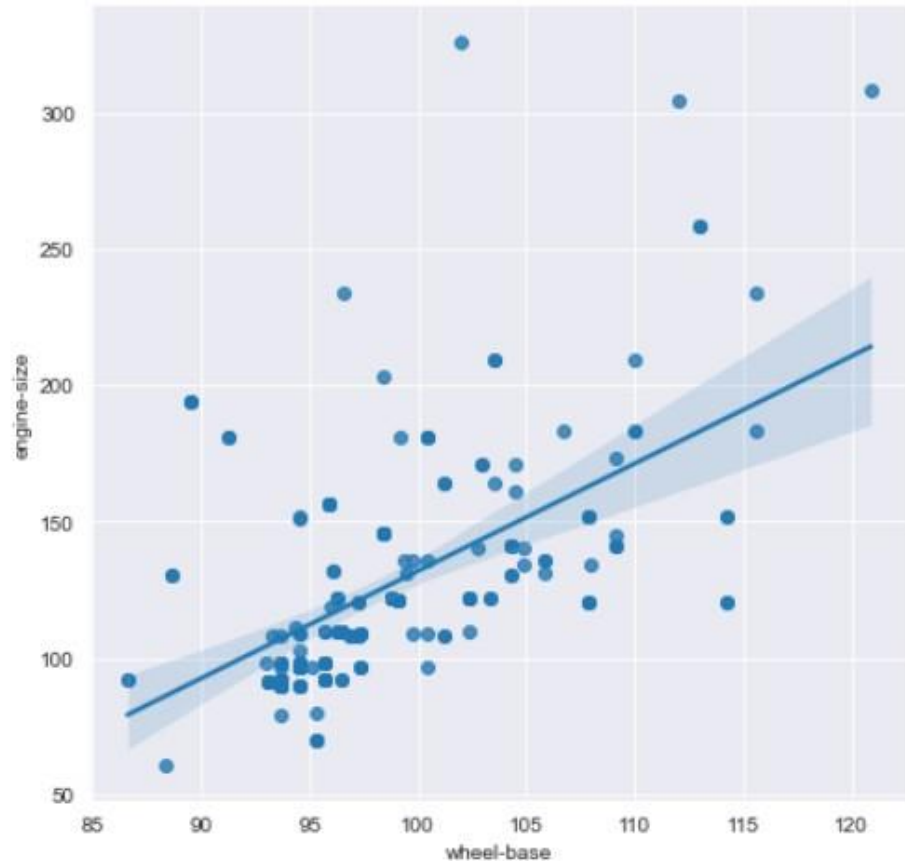
Syntax:

```
seaborn.regplot(x, y, data=None, x_estimator=None,
x_bins=None, x_ci='ci', scatter=True, fit_reg=True, ci=95,
n_boot=1000, units=None, order=1, logistic=False, lowess=False,
robust=False, logx=False, x_partial=None, y_partial=None,
truncate=False, dropna=True, x_jitter=None, y_jitter=None,
label=None, color=None, marker='o', scatter_kws=None,
line_kws=None, ax=None)
```

Code Snippet:

In [16]: *# Reg Plot*

```
plt.figure(figsize=(7,7))  
sns.regplot(x=auto["wheel-base"], y=auto["engine-size"])  
plt.show()
```



Conclusion: We have successfully implemented Scatter plot, Dist plot, KDE plot and Reg plot using seaborn library.

Assignment 6

Title: Visualization using matplotlib library.

Aim: To plot interactive visualizations using the matplotlib library.

Problem Statement: Create following visualizations(ANY 4) using **MATPLOTLIB** for any dataset of your choice and draw appropriate conclusions

- a. scatterplot
- b. barchart
- c. histogram
- d. piechart
- e. plot(plotting a line)

Theory:

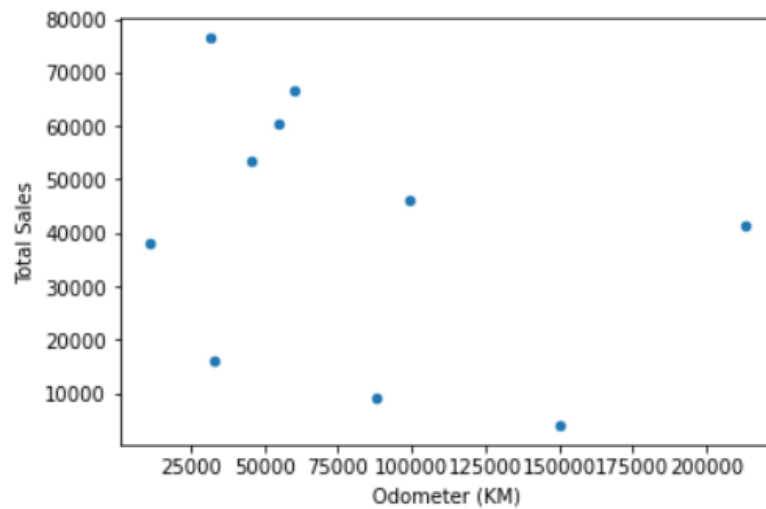
- A. Scatter Plot:- **Scatterplot** can be used with several semantic groupings which can help to understand well in a graph. They can plot two-dimensional graphics that can be enhanced by mapping up to three additional variables while using the semantics of hue, size, and style parameters. All the parameter control visual semantic which are used to identify the different subsets. Using redundant semantics can be helpful for making graphics more accessible.

Syntax:

```
matplotlib.pyplot.scatter(x_axis_data, y_axis_data, s=None, c=None, marker=None,
cmap=None, vmin=None, vmax=None, alpha=None, linewidths=None,
edgecolors=None)
```

Code Snippet:

```
In [14]: #car_sales["Price"] = car_sales["Price"].astype(int)
car_sales.plot(x="Odometer (KM)",y="Total Sales",kind="scatter");
```



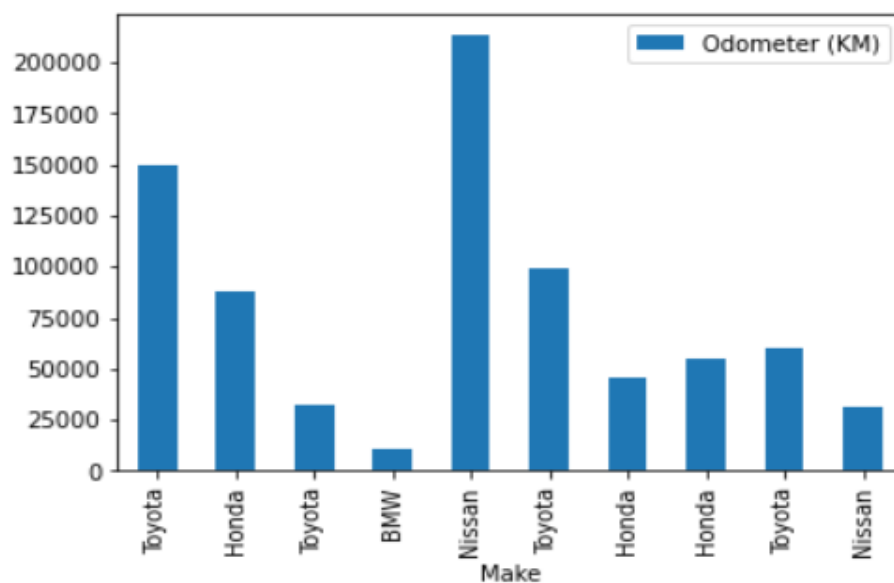
- B. **Bar chart:** A bar plot or bar chart is a graph that represents the category of data with rectangular bars with lengths and heights that is proportional to the values which they represent. The bar plots can be plotted horizontally or vertically. A bar chart describes the comparisons between the discrete categories. One of the axis of the plot represents the specific categories being compared, while the other axis represents the measured values corresponding to those categories.

Syntax:

```
plt.bar(x, height, width, bottom, align)
```

Code Snippet:

```
In [17]: car_sales.plot(x="Make",y="Odometer (KM)",kind="bar");
```



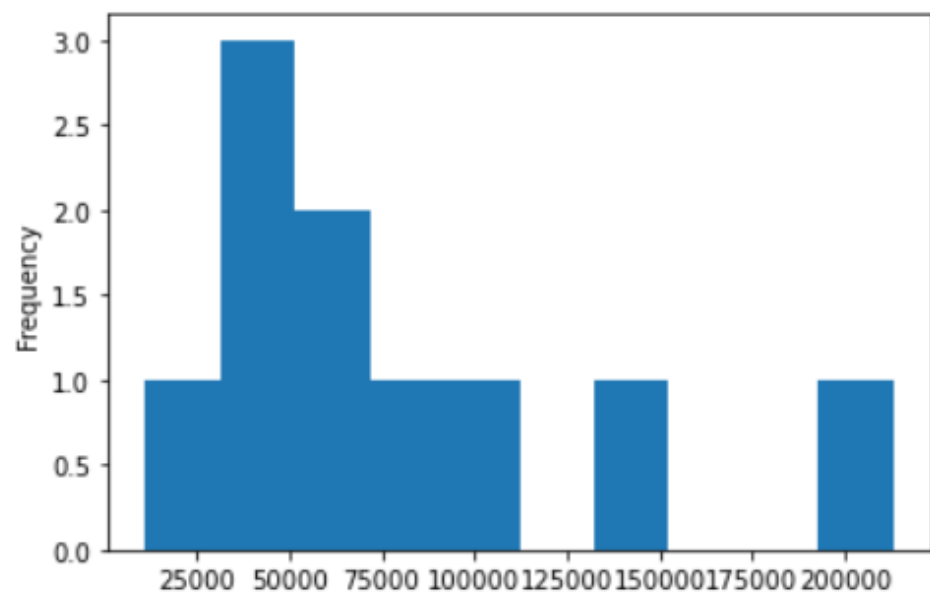
- C. Histogram: A histogram is basically used to represent data provided in a form of some groups. It is an accurate method for the graphical representation of numerical data distribution. It is a type of bar plot where X-axis represents the bin ranges while Y-axis gives information about frequency.

Syntax:

```
matplotlib.pyplot.hist(x, bins=None, range=None, density=False, weights=None,
cumulative=False, bottom=None, histtype='bar', align='mid',
orientation='vertical', rwidth=None, log=False, color=None, label=None,
stacked=False, *, data=None, **kwargs)
```

Code Snippet:

```
In [18]: car_sales["Odometer (KM)"].plot.hist();
```



- D. Plot(plotting a line): *Line charts* are used to represent the relation between two data X and Y on a different axis.

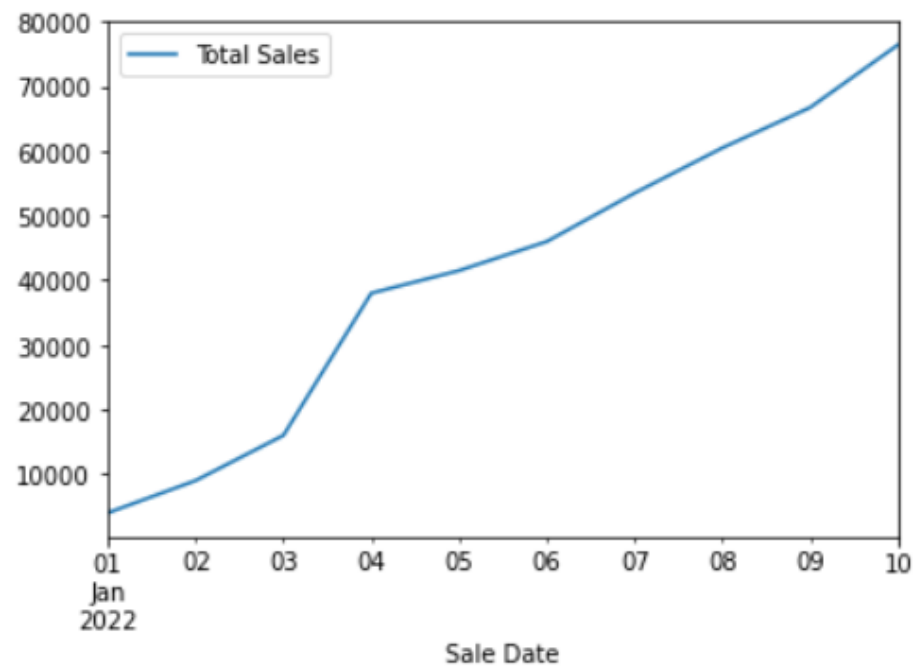
Syntax:

```
plt.plot(x, y)
```

Code Snippet:

```
In [12]: #Plotting the total sales graph  
car_sales.plot(x="Sale Date",y="Total Sales")
```

```
Out[12]: <AxesSubplot:xlabel='Sale Date'>
```



Conclusion: We have successfully implemented line plot, bar plot, histogram and scatter plot using matplotlib library.