Aim: Write a Python program to demonstrate various Data Visualization Techniques.

Data visualization is a graphical representation of quantitative information and data by using visual elements like graphs, charts, and maps.

Data visualization convert large and small data sets into visuals, which is easy to understand and process for humans.

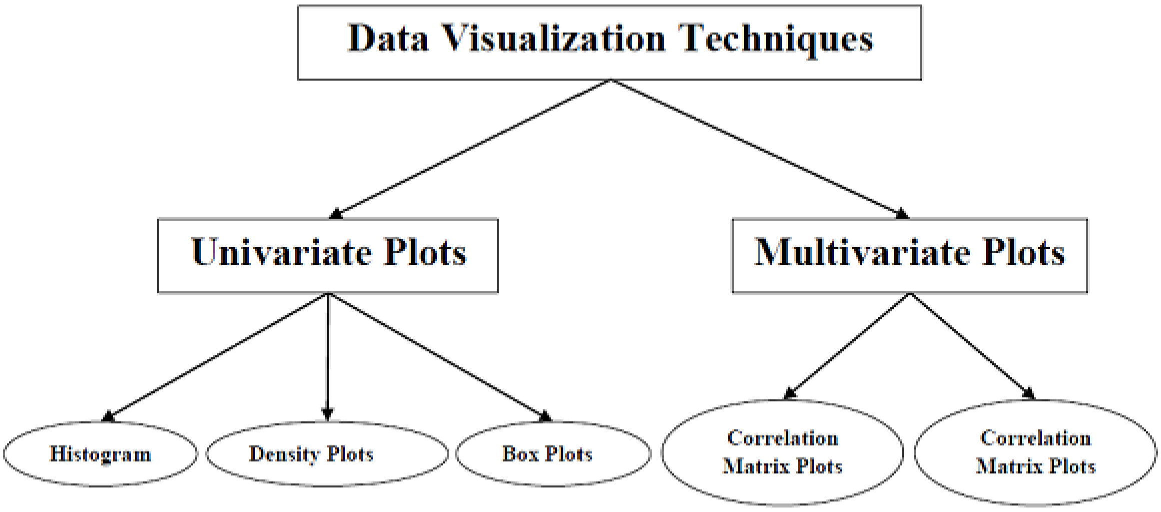
Data visualization tools provide accessible ways to understand outliers, patterns, and trends in the data.

In the world of Big Data, the data visualization tools and technologies are required to analyze vast amounts of information.

Data visualizations are common in your everyday life, but they always appear in the form of graphs and charts. The combination of multiple visualizations and bits of information are still referred to as Infographics.

Data visualizations are used to discover unknown facts and trends. You can see visualizations in the form of line charts to display change over time. Bar and column charts are useful for observing relationships and making comparisons. A pie chart is a great way to show parts-of-a-whole. And maps are the best way to share geographical data visually.

Today's data visualization tools go beyond the charts and graphs used in the Microsoft Excel spreadsheet, which displays the data in more sophisticated ways such as dials and gauges, geographic maps, heat maps, pie chart, and fever chart.



1. Matplotlib

Matplotlib is one of the best python data visualization libraries for generating powerful yet simple visualization. It is a 2-D plotting library that can be used in various ways, including Python, iPython sheets, and Jupyter notebooks.

Key Features

It supports various types of graphical representation, including line graphs, bar graphs, and histograms.
It can work with the NumPy arrays and border SciPy stack.
It has a huge amount of plots for understanding trends and making correlations.

Pros And Cons

Interactive platform
Versatile library
Not ideal for time series data

Installation

Click here to install Matplotlib

2. Plotly

The most popular data visualization library in Python is Plotly, which delivers an interactive plot and is easily readable to beginners. It is widely used for handling financial, geographical, statistical, and scientific data.

Key Features

Its robust API functions effectively in both local and web browser modes.

It is an interactive, open-source, and high-level visualization library.

It can be viewed in Jupyter notebooks, standalone HTML files, or even hosted online.

Pros And Cons

Offers contour plots, dimension chars, and dendrograms.

Allows 40 unique chart and plot types

Difficult to use

Installation

Click here to install Plotly

3. Seaborn

Seaborn is the best python library for data visualization, which offers a variety of visualized patterns. It is designed to work more compatible with Pandas data form and is widely used for statistical visualization.

Key Features

It performs the necessary mapping and aggregation to form information visuals.

It is integrated to explore and understand data in a better and more detailed way.

It offers a high level of a crossing point for creating beautiful and informative algebraic graphics.

Pros And Cons

Much more visually appealing representation

Switch to any other data format

Limited customizable options

Installation

Click here to install Seaborn

4. GGplot

GGplot is another popular data visualization library in Python, known as the python implementation of graphics grammar. It refers to the map of the data, with its aesthetic attributes including color, shape, and geometric objects like points and bars.

Key Features

It allows you to build informative visualization substantially with improved representations

It is integrated with Panda to store data in a data frame.

It is based on ggplot2, an R programming language plotting system.

Pros And Cons

Documentation is simple and easy to follow.
Save method to discuss and exhibit plots
Not suitable for creating highly customized graphics.

Installation

Click here to install GGplot

5. Altair

Altair is regarded as one of the declarative statistical visualization libraries for data visualization in Python. It indicates that we need to define the links in the data columns, which are the x-axis and y-axis, while creating any visuals.

Key Features

It has a user-friendly and consistent API built on Vega-lite JSON specification.
Its source is present on GitHub
It is dependent on python 3.6, entry points, jsonschema, NumPy, Pandas, and Toolz

Pros And Cons

Create the best visuals with minimal code
Holds declarative grammar on both visuals and interaction

Installation

Click here to install Altair

6. Bokeh

Bokeh is another interactive python library for data visualized for modern web browsers. This is best suitable for developing interactive plots and dashboards for complex or streaming data assets.

Key Features

It has a broad range of intuitive graphs which can be leveraged to form solutions.
It is well-known for creating custom-made visualizations.
It includes various generation and plot chart methods, including box plots, bar plots, and histograms.

Pros And Cons

Highest level of control for the rapid creation of charts
Many graphs with fewer codes and higher resolution
No pre-set defaults, and users have to define them each time.

Installation

Click here to install Boken

7. Pygal

Pygal is one of the most popular and best python data visualization libraries where the interactive plots are built using the pygal library. It allows you to download your visualization in various formations, including SVG. PNG, Browser, PyQuery, and more.

Key Features

It has three individual map packages to keep the compact module size.

It offers an interactive experience with data explorations, filtration, and more.
It is optimized with rich support, allowing users to be more visionary even in multiple complex problems.

Pros And Cons

It can provide output chats of data like SVGs.
Attractive char in a few code lines
Become slow with a large number of data points

Installation

Click here to install Pygal

8. Geoplotlib

Geoplotlib is another data visualization library of Python that allows the user to build maps and plot geographical data. This library is designed to handle the entire dataset, map projection, and tile download of the map automatically.

Key Features

It has the toolbox used to draw various maps, including heatmaps, dot-density maps, and choropleths.
It has an interface of an object-oriented programming language.
It also has excellent zooming and panning maps for distinct views.

Pros And Cons

Graphics rendering from OpenGL
Large datasets can be performed with excellent resolution
Enables hardware acceleration

Installation

Click here to install Geoplotlib

9. Folium

Folium is one of the easy-to-use data visualization python libraries to visualize and analyze data on an interactive leaflet map. The library utilizes OpenStreetMap, which provides the user with an excellent Google Maps experience with few codings.

Key Features

It has numerous built-in tilesets from various platforms, including Stamen, Mapbox, and OpenStreetMaps.
It is easy to add locations of other users with markets.
It also has different plugins and is capable of creating maps similar to plotly, Altari, and broken.

Pros And Cons

Employ various plugins
Easy to draw maps with markers
Complex to handle shapefiles

Installation

Click here to install Folium

10. Gleam

Gleam is the perfect python library for data visualization, made from the inspiration of the Shiney package of programming language. The users can develop the basic plot with gleam while building various fields on top, enabling easy filtration and sorting of data.

Key Features

It is used in the visualization and analysis of data in interactive web applications that take only python scripts.
It can work with any kind of data visualization of python language.
It doesn't require the knowledge of HTML, CSS, or JavaScript.

Pros And Cons

Suits all types of library
Simple to filter and sort data

Installation

Click here to install Gleam

Usage Of Python Libraries For Data Visualization

Python is a dynamic, portable, interpreted, and object-oriented programming language that has its advantages in Computer vision, data science, machine learning, robotics, and so on. Data visualization in python libraries gives you many insights throughout the entire process of analysis.

Here are some benefits of python libraries for data visualization.

1. Due to Python's popularity, the features of python libraries inherited the simplicity and readability, making it user-friendly for people.
2. There are numerous python data visualization libraries where some of which are tailor-made to fit your requirements.
3. From importing data from excel sheets to processing datasets for analyzing time series, python libraries have all. For instance, Matplotlib is equipped with the necessary tools for effective data visualization.

Career Options In Data Visualization

There are various options for a data visualization career available across multiple industries. If you are interested in the field, it will be helpful to learn the roles and responsibilities of each position to make more informed career choices. Here are some data visualizations of career options.

1. Data specialist (Rs. 7.0 Lakhs) – They convert data of different forms, including paperwork and physical documents, to digital formations for further data visualization.
2. Data visualization engineer (Rs. 12.6 Lakhs.)– They transform complicated data into easy-to-understand information which is accessible to everyone.
3. Data Visualization consultant (Rs. 6.1 Lakhs.) – They perform various tasks including better handling the data, extracting more useful insight, finding solutions to issues, and making better decisions.

Program

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

```
cars_data = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/dataset /Toyota1.csv',index_col=0,na_values=['??','????'
])
# Print the shape of the dataset
print ('cars_data: %s'%(str(cars_data.shape)))
```

O/p:

```
cars_data: (1436, 10)
```

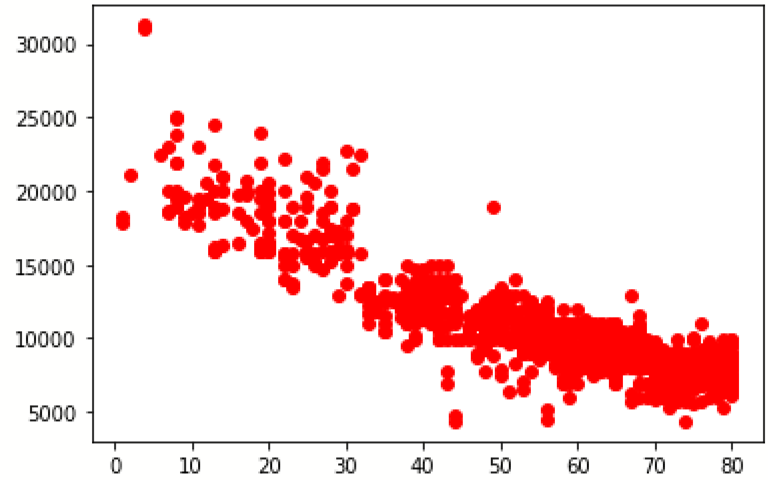
```
cars_data.dropna(axis=0,inplace=True)
# Print the shape of the dataset
print ('cars_data: %s'%(str(cars_data.shape)))
```

O/p

```
cars_data: (1096, 10)
```

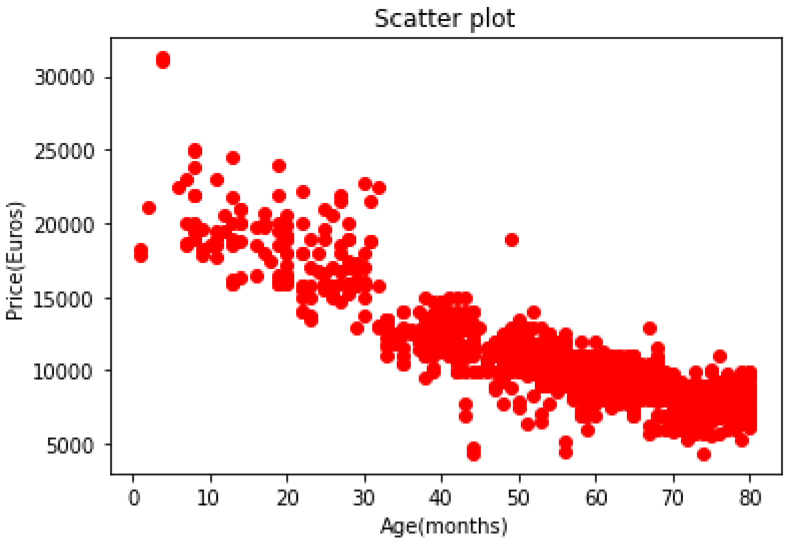
scatter plot

```
plt.scatter(cars_data['Age'],cars_data['Price'],c='red')
```



```
plt.scatter(cars_data['Age'],cars_data['Price'],c='red')
plt.title('Scatter plot')
plt.xlabel('Age(months)')
plt.ylabel('Price(Euros)')
plt.show()
```

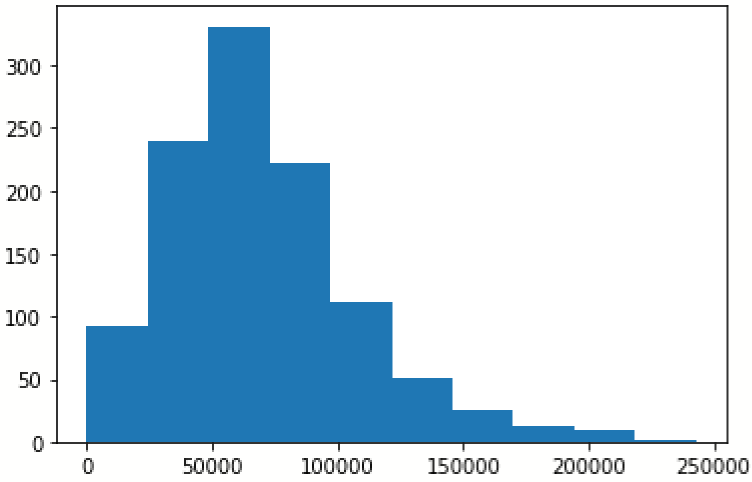
O/p



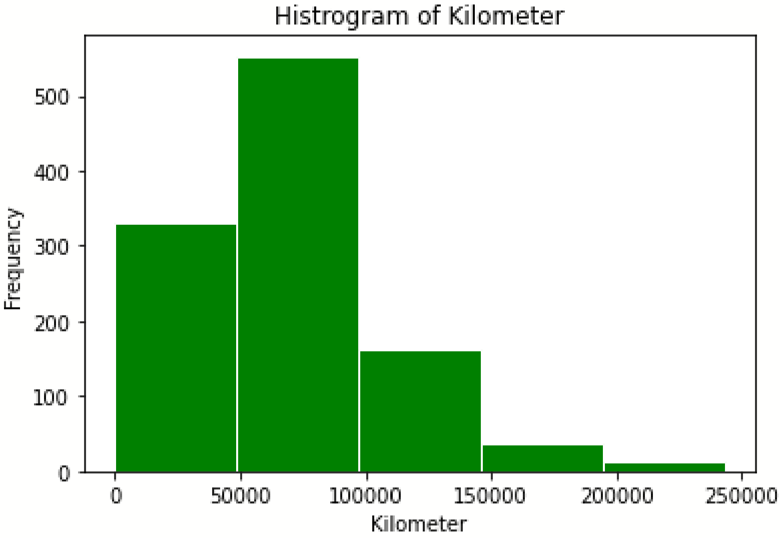
Histograms

```
plt.hist(cars_data['KM'])# Histogram with default argument
```

1

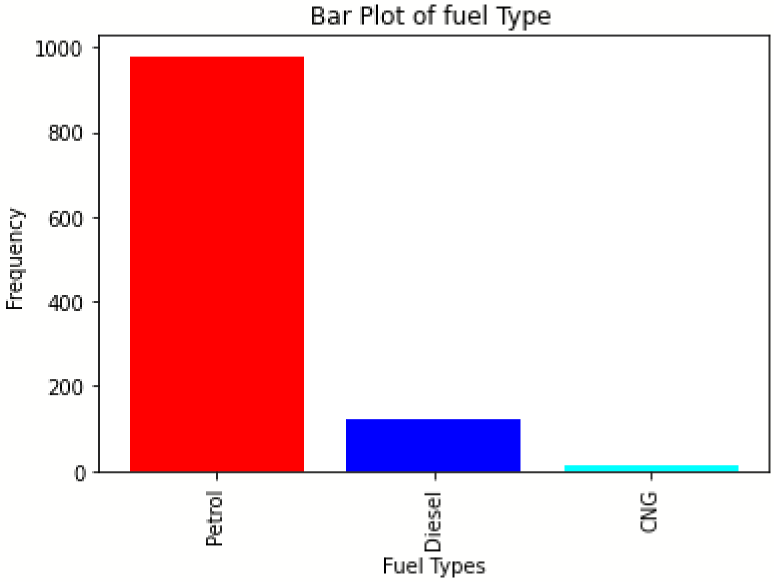


```
plt.hist(cars_data['KM'],
        color = 'green',
        edgecolor = 'white',
        bins = 5)
plt.title('Histrogram of Kilometer')
plt.xlabel('Kilometer')
plt.ylabel('Frequency')
plt.show()
```

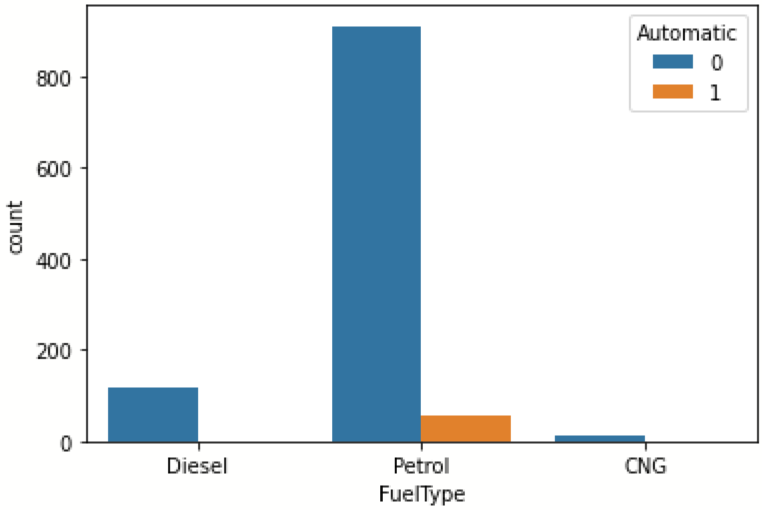


```
#Bar Plot
counts = [979, 120, 12]
fuelType =('Petrol','Diesel','CNG')
index = np.arange(len(fuelType))

plt.bar(index,counts, color=['red','blue','cyan'])
plt.title('Bar Plot of fuel Type ')
plt.xlabel('Fuel Types')
plt.ylabel('Frequency')
plt.xticks(index,fuelType,rotation=90)
plt.show()
```

1

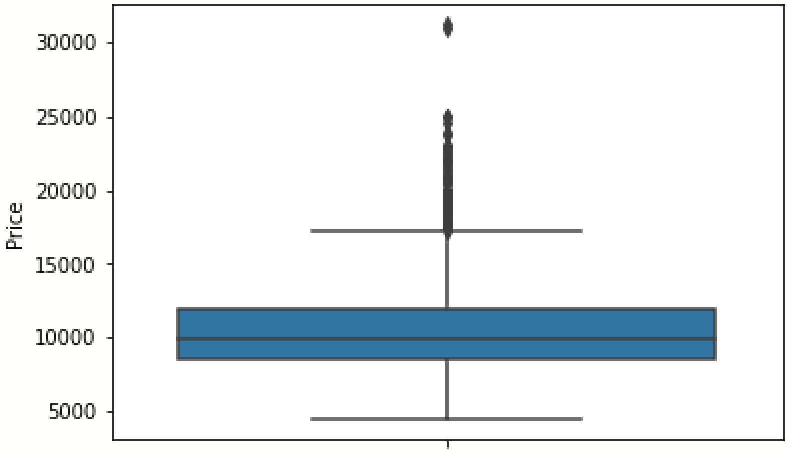


```
import seaborn as sns

#Grouped Bar plot
sns.countplot(x="FuelType",data=cars_data,hue="Automatic")
```



Box plots

```
sns.boxplot(y=cars_data["Price"])
```



Conclusion: