


School of Computing
Department of Computer Science & Engineering

(Artificial Intelligence and Machine Learning)

ACADEMIC YEAR 2025-26 (SUMMER SEMESTER)

LAB RECORD NOTEBOOK

10212CA214 - DATA VISUALIZATION

NAME: M.krishna Koushik

VTU.NO: 25106

REG.NO: 23UECL0033

BRANCH: CSE(AIML)

YEAR/SEM: III | V

School of Computing
Department of Computer Science & Engineering
(Artificial Intelligence and Machine Learning)

ACADEMIC YEAR 2025-26 (SUMMER SEMESTER)

BONAFIDE CERTIFICATE

NAME : M. Krishna Kowshik
BRANCH : CSE (AIML)
VTU NO. : 25106
REG.NO. : 23UECL0033
YEAR/SEM : III/V
SLOT NO. : S12L4

Certified that this is a bonafide record of work done by above student in the **"10212CA214 - DATA VISUALIZATION LABORATORY"** during the year 2025-2026 (Summer Semester).

E.T. 29/10/25
SIGNATURE OF LAB HANDLING FACULTY

SIGNATURE OF HOD

Submitted for the Semester Practical Examination held on **03.11.25** at Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology.

INTERNAL EXAMINER

EXTERNAL EXAMINER

INDEX

	Date	Title	Page No.	Marks	Faculty Signature
	21/07/25	Exploration of Data Visualization Tools like Tableau, Python libraries, D3.js	1-6	17	21/7
	28/07/25	To visualize and perform Univariate analysis using continuous and categorical data	8-12	17	28/7
	04/08/25	To visualize and perform Bivariate analysis using continuous and categorical data	13-16	17	4/8
	11/08/25	To visualize and perform Multivariate analysis using Multiple variables involving Multiple measures	17-18	16	11/8
	18/08/25	To design and perform visualization for Trees	19-22	16	18/8
	25/08/25	To design and perform visualization for Graphs and Networks	23-24	16	25/8
	08/09/25	To generate insight using Text Network Analysis and Visualization	25-28	18	8/9
.	15/09/25	To analyze and visualize Spatial and Geospatial data	29-30	19	15/9
.	29/09/25	To analyze and visualize Time Oriented Data	31-33	18	29/9
).	13/10/25	Use Case:1 Earthquake and Geospatial Data Analysis	34-37	17	13/10
.	27/10/25	Use Case:2 Performance of Different Departments over Year	38-41	18	27/10

Completed

Total Marks: 189 / 220

Signature of Faculty

21/7/25 TASK-1(A)

Aim:-
The primary aim of this task is to Perform a Preliminary Exploratory data Analysis on the EDA Processed Diabetic Dataset.

Algorithm:-
1. Load necessary python libraries (pandas, numpy)
2. Read the diabetic dataset using pandas
3. Display the first few rows to understand data set
4. Check each column name and its datatype
5. sort and show rows based on the first column in both ascending and descending order
6. Compute the number of unique values from the dataset
7. Use Pandas 'describe' to compute statistics like count mean, std, min, etc. for numerical features
8. Get the shape of the dataset to know number of rows and columns

Program:-

```
import pandas as pd
df = pd.read_csv("diabetic_data.csv")
Print("=== First 5 Rows ===")
Print(df.head(), "\n")
Print("=== Dataframe info ===")
df.info()
Print()
Print("=== Number of columns ===")
Print(len(df.columns))
Print("=== Column Datatypes ===")
Print(df.dtype, "\n")
print("=== TOP 5 Rows sorted by 'age' Ascending ===")
Print(df.uniqve(), "\n")
total_uniqve = df.stack().nuniqve()
Print(f"=== total uniqve values in Entire Dataframe:
                                    {total_Uniqve} ===\n")
Printf("=== Descriptive statics ===")
Print(df.describe().T, "\n")
rows, cols = df.shape
Print(f"=== Dataset shape ===\n Rows : {rows}
                                    {rows} Columns : {cols}")
```

Result:-

In this program I performed Exploratory data analysis on the C.U. processed diabetic Dataset by loading it into a dataframe and displaying its structure and contents.

3	1	89	66	23	94	
4	0	137	40	35	168	

	BMI	Diabatespedigreefunction	Age	outcome
0	33.6	0.627	50	1
1	26.6	0.351	31	0
2	23.3	0.672	32	1
3	28.1	0.167	21	0
4	43.1	2.288	35	1

=== Data frame Info ===

#	column	Non-Null Count	Dtype
0	Pregencies	768 non-null	int64
1	Glucose	768 non-null	int64
2	Blood Pressure	768 non-null	int64
3	skinthicknes	768 non-null	int64
4	Insulen	768 non-null	int64
5	BMI	768 non-null	int64
6	Diabetespedigreefunction	768 non-null	float64
7	Age	768 non-null	float64
8	outcome	768 non-null	int64

number of columns ===
9
=== column Datatypes ===
Pregnancies int64
Glucose int64
Blood Pressure int64

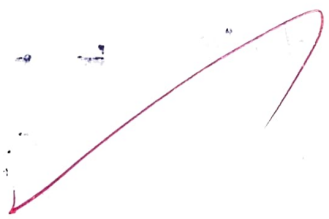
Task-1B

Aim:-

To perform data preprocessing on a dataset by identifying and handling null values and duplicate entries.

Algorithm:-

1. Import necessary python libraries
2. load the dataset using pandas
3. Display the first 5 rows of the dataset.
4. check for null values in each column
5. Remove null values if any
6. Identify duplicate rows
7. Remove duplicate rows.



Program

```
import pandas as pd
df: pd. read csv("https://raw.githubusercontent.com
            /uiuc-cse/data-fa14/gh-pages/data/iris.csv")
print ("First 5 rows of the dataset")
print (df. head())
print ("\n count of null values in each column:").
print (df. isnull ().sum())
df-cleaned = df. dropna()
print ("\n Dataframe after removing null values:)
print (df. cleaned . head ())
duplicates = df_cleaned[df-cleaned.duplicates()]
print ("\n Duplicates rows in the dataset:)
print (duplicates)
df_no_duplicates = df_cleaned.drop_duplicates()
print ("\n Dataframe after removing duplicates!")
print ("df_no-duplicates.head()).
```

VEL TECH	
EX No.	1
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	2
RECORD (5)	5
TOTAL (20)	12+5=17
SIGN WITH DATE	

21/7/25

Result:-

Thus, to perform data preprocessing on a dataset by indentifying and handling null values and duplicates

First 5 rows of the dataset

	speal_length	speal_width	petal_length	petal_width	
0	5.1	3.5	1.4	0.2	
1	4.9	3.0	1.4	0.2	
2	4.7	3.2	1.3	0.2	
3	4.6	3.1	1.5	0.2	
4	5.0	3.6	1.4	0.2	

count of null values in each column.

speal_length 0
speal_width 0
petal_length 0
petal_width 0
species 0
dtype: int64

o/p

Dataframe after removing null values (if any):

	speal_length	speal_width	petal length	petal width	
0	5.1	3.5	1.4	0.2	
1	4.9	3.0	1.4	0.2	
2	4.7	3.2	1.3	0.2	
3	4.6	3.1	1.5	0.2	
4	5.0	3.6	1.4	0.2	

Duplicate rows in the dataset

	speal_length	speal_width	petal_length	petal width	
34	4.9	3.1	1.5	0.1	
37	4.9	3.1	1.5	0.1	

Dataframe after removing duplicates

	speal_length	speal width	petal_length	petal width	
0	5.1	3.5	1.4	0.2	
1	4.9	3.0	1.4	0.2	
2	4.7	3.2	1.3	0.2	
3	4.6	3.1	1.5	0.2	
4	5.0	3.6	1.4	0.2	

Choose any [illegible] from ... chart, pie chart using univariate analysis of categorical data using identified data. construct a scatter plot, Line plot, strip plot, swarm plot using univariate analysing continuous data using identified data

Aim:-

To perform univariate analysis by identifying categorical and continuous attributes from a selected dataset and visualizing the insights using plots such as for bar charts, pie chart, scatter plot, line plot, strip plot, and swarm plot

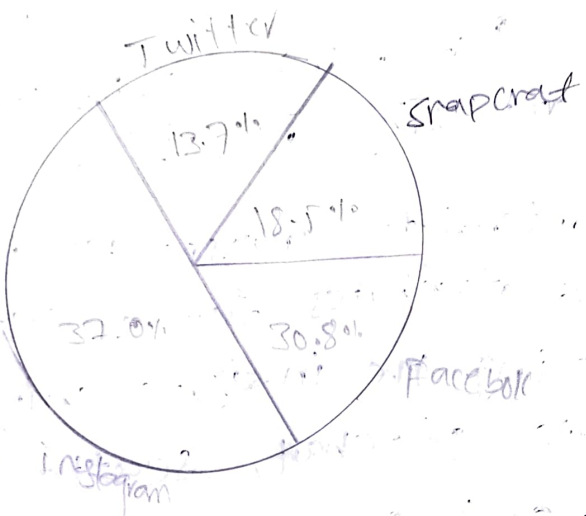
Algorithm

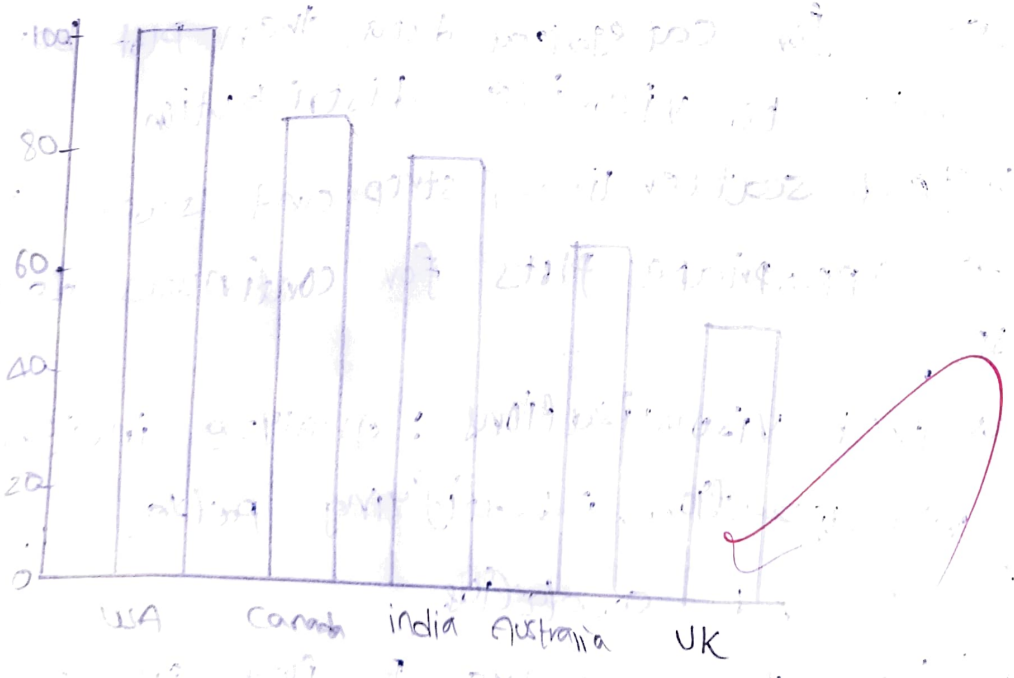
1. Choose Dataset: select a dataset with categorical and continuous attributes
2. Identify data types: Distinguish b/w categorical and continuous attributes
3. Construct Bar and pie charts: create frequency tables for categorical data, then plot bar and piecharts to visualize distribution
4. Construct scatter line, strip and swarm plots. utilize appropriate plots for continuous data analysis.
5. Interpret visualizations: Analyze insights gained from visualizations, identifying patterns, dominant categories and correlations
6. Consider further Analysis: Reflect on implications for further exploration or decision

Program:-

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv("Content | sentimes +dataset.csv")
plot form. likes_top5 = df.groupby('plot form')['likes']
                        sum(), sort values(ascending = false
                                                    head)

# create a pie chart
plt.figure_(figure = (4,4))
platform_likes.top5.plot(kind = 'pie', autopct = '%.1.1f%%')
                        % startangle = 140,
        colors = ['skyblue', 'light coral', 'light green', 'light salmon']
    leirnes sky blue])
plt.title("top5 platforms by total likes")
plt.ylabel('')
plt.show()
top_countries = df.nlargest(5, 'likes')
# create a bar chart
plt.bar(top5_countries['country'] top5, countries['Likes']
    ['Likes'], 'light salmon', 'light sky blue'])
plt.title("top 5 platforms by total likes')
plt.ylabel('')
plt.show()
top5_countries = df.nlargest(5, 'Likes')
# create a bar_chart
plt.figure(figsize = (10,6))
plt.bar(top5_countries['country'], top5_(countries['likes']
    light_salem', 'lights sky blue'])
plt.xlabel('country')
plt.ylabel("total liked")
```


Twitter
13.7%
Snapchat
18.5%
32.0%
30.8%
Facebook
Instagram

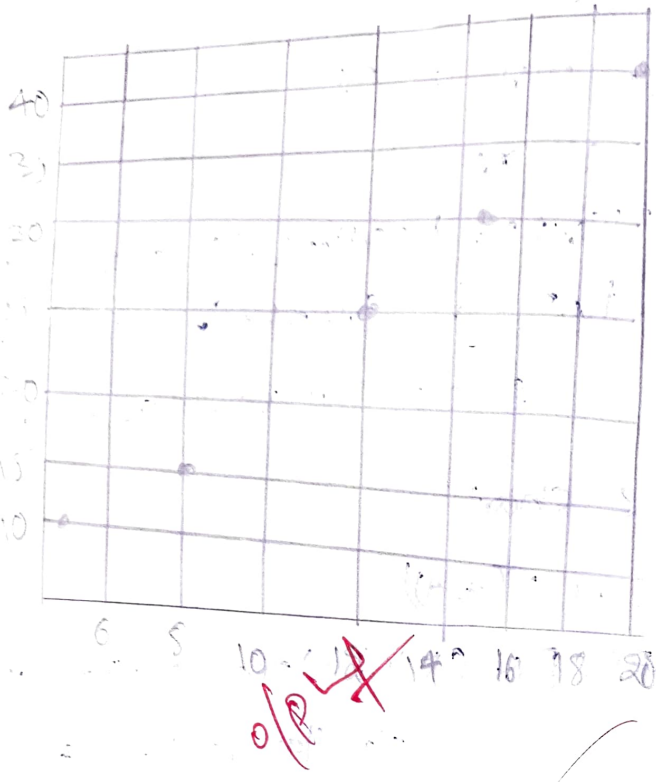

100
80
60
40
20
0
USA
Canada
India
Australia
UK

```
plt.title("top 5 countries by total likes")
plt.show()
data = {
    'Text': ['Enjoying a beautiful day of park!',
    'Traffic was terrible this morning', Just finished
an a merging workout', expected about upcoming weeks
'Retweet' = [15, 3, 20, 8, 12],
'Likes': [30, 10, 40, 15, 25]
}
df = pd.DataFrame(data)
#set the seaborn style
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Retweets', y='Likes', data=df, col='
        alpha' =0.7)
plt.title('Scatter Plot of Retweets Vs Likes')
plt.show().
```

RESULT:-

Thus to Perform univariate calculus analysis by identifying categorical and continuous attributes and visualising the insights are done and executed successfully.

Scatter Plot of Product Vs Price



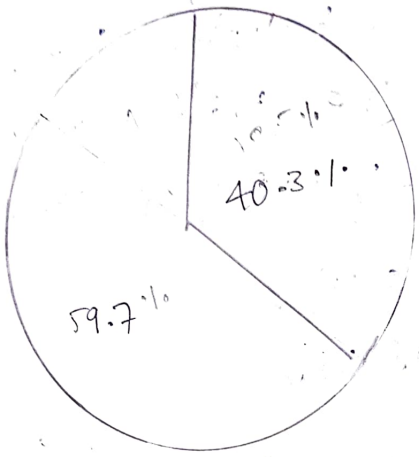
Aim:-
10 Perform univariate analysis on a given employee dataset by identifying categorical and continuous attributes and visualizing categorical data using bar and pie charts continuous data using histogram, density plot and rug plot.

Algorithm:-
1. load the employee dataset using pandas
2. Identify and seperate categorical and continuous attributes
3. For categorical attributes
   - Construct a Bar chart
   - Construct a pie chart
4. For continuous attributes
   - Construct a histogram
   - Construct a Density plot
   - Construct a Rug plot
5. Display and interpret each visualization

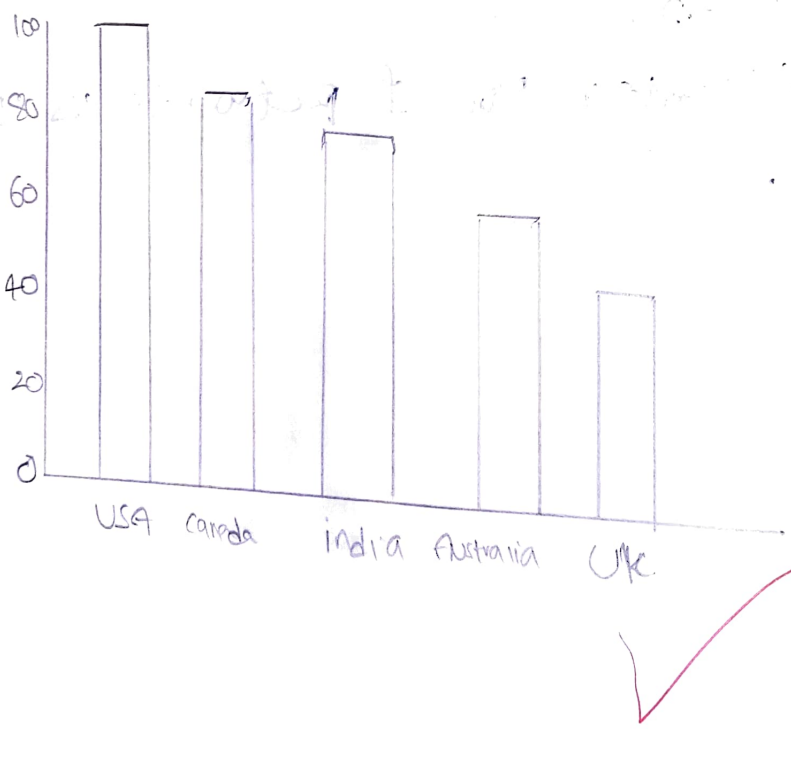
Program:-
```
import seaborn as sns
import Pandas as Pd
file_path = 'Employee.csv'
df = pd.read_csv(file_path)
categorical_counts = df[categorical_column].value_counts
plt.pie(category_counts, labels = category_counts.index,
autopct = '%1.1f%%', startangle =
```

output :-

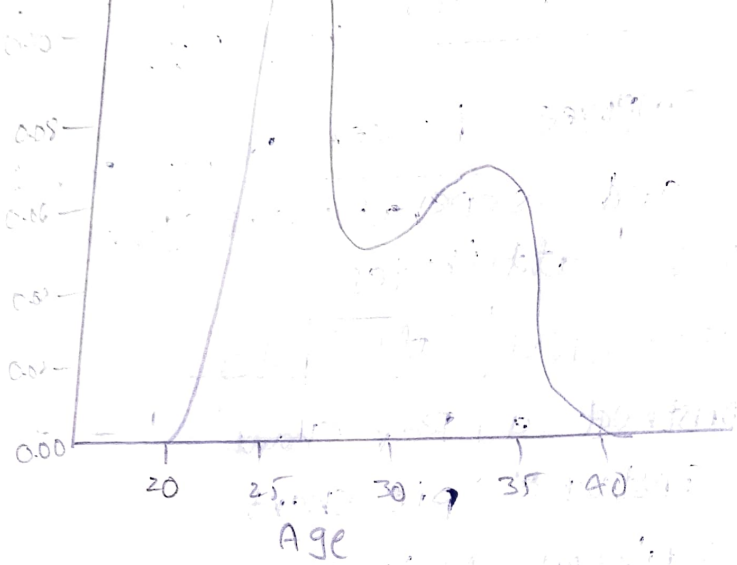
The pie chart of Gender



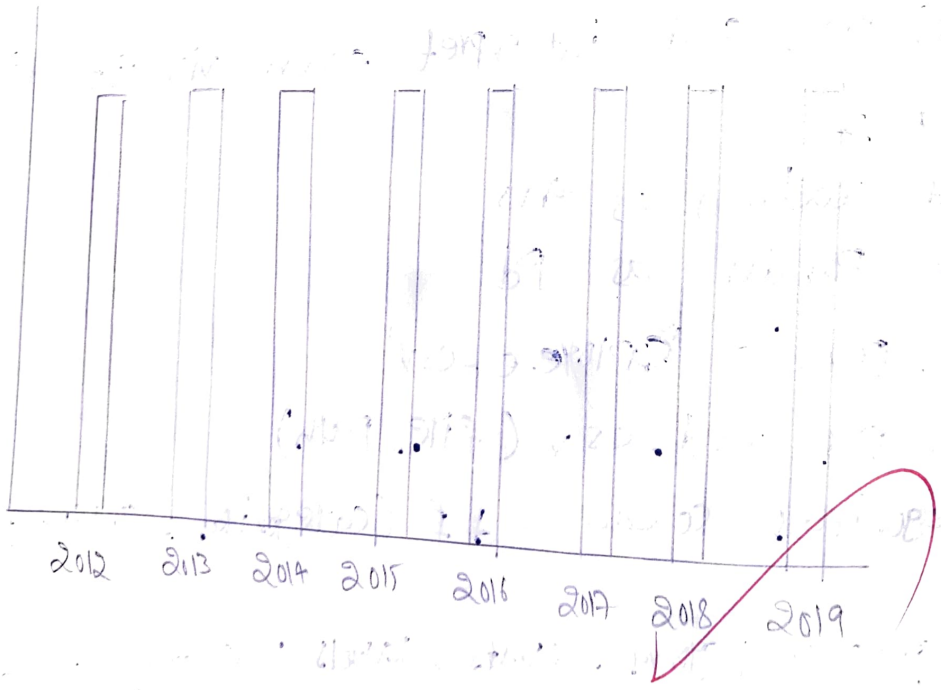
Bar city of city.



```
Plt.show()
Categorical_column = 'city'
categorical_counts = df[categorical_column].value_counts()
plt.bar(categorical_counts.index, categorical_counts,
    color=['skyblue', 'lightcoral', 'lightgreen', 'orange'])
plt.title(f'Bar Chart of {categorical_column}')
plt.xlabel(categorical_column)
plt.ylabel('Count')
plt.show()
Continuous_column = 'Age'
Plt hist(df[continuous_column], bins=10, edgecolor='black')
needed bins = blue bars
plt.title(f'{continuous_column} Destribution')
plt.xlabel(continuous_column)
plt.ylabel('freavency')
plt.show()
Continuous_column = 'Age'
sns.kdeplot(df[continuous_column], fill=True)
plt.title(f'Density plot of {continuous_column}')
plt.xlabel(continuous_column)
plt.ylabel('Density')
plt.show()
Continuous_column
```



Rug plots of Joining year



Color
Plt. hi
Plt. xl
Plt. yl
Plt. sh
Continuous
Plt hist(
needed bi
Plt. title
Plt. x labe
Plt. ylab
Plt. shou
Continuous_
sns. kdeplo
Plt. title(
Plt. xlabel
plt. y label
Plt. show
Continuous
sn. rug p

plt. yticks ()

plt. show()

VEL TECH	
EX No.	2
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	3
RECORD (5)	4
TOTAL (20)	12+5=17
SIGNATURE DATE	

28/8

Result:-

Thus to Perform univariate analysis on a given employee dataset by identifying categorical and continous attributes and visualizing and

11/8/25 Task-3 using continuous and categorical data 13

categorical vs categorical: stocked Barchart, Grouped Barchart, segmented Barchart, mosaic Plots

Continuous vs continous: Scatterplot fit lines

categorical vs continuous: Bar chart (summary statistics Grouped kernal Density Plots, Bar Plots, violin Plots, Ridgeline plots Beeswarm Plots

Tools: Tableau, Language: Python

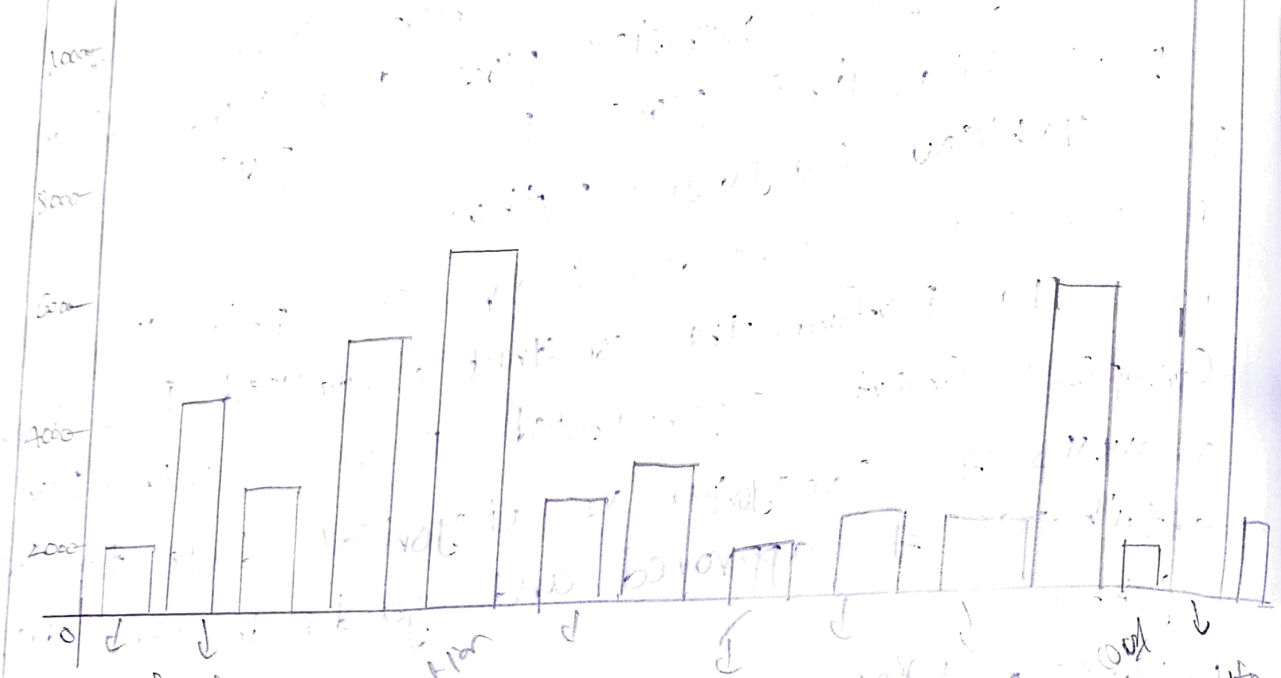
link for data set: student if coral consomphion (kaggle.com)

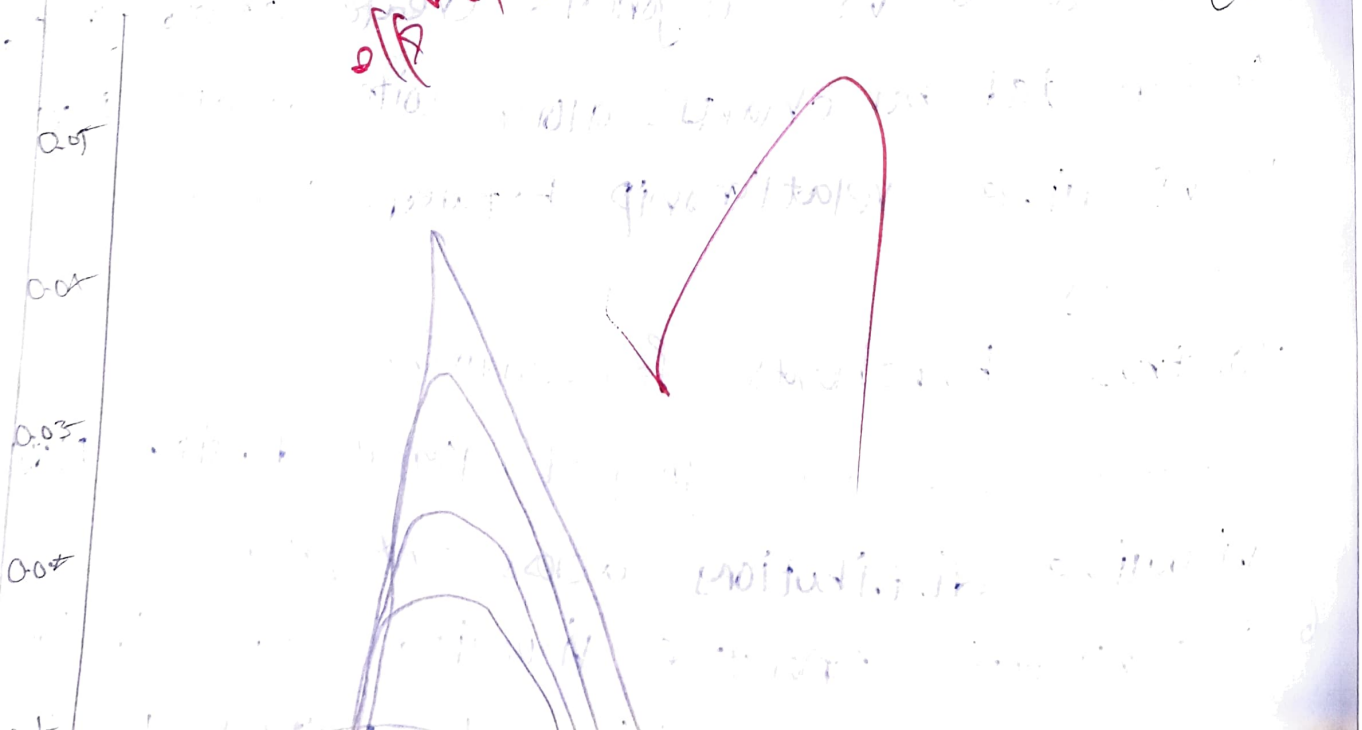
Aim:- To Perform the construct a stacked Bar chart GroupBar chart, segmented Barchat using Bivariate analysis of categorial vs categorical data for the attributes of approved and gender in above dataset

Algorithm:-

1. Select Dataset: choose a dataset containing both categorical and continuous variables
2. Differntiate variables:- Identify categorical and continuous variable
3. categorical vs categorical: create stacked grouped or segmented bar charts. along with mosaic Plots to visualize relationship between categorical variables
4. Construct barcharts for summary
5. statistics and use grouped kernal density plots to visualize distributions across categories

```
import matplotlib.pyplot as plt
df = pd.read_csv('/content/clean_dataset.csv')
industry = df['Industry']
income = df['Income']
plt.figure(figsize=(10,6))
plt.bar(industry, income, color='sky blue')
plt.xlabel('Industry')
plt.ylabel('Income')
plt.title('Income by Industry')
plt.xticks(rotation=45)
plt.show()
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
df = pd.read_csv('/content/clean_dataset.csv')
categorical_column = ['Ethnicity']
continuous_column = 'Income'
plt.figure(figsize=(14,8))
sns.kdeplot(data=df, x=continuous_column, null=
    categorical_column, fill=True, common_norm=False)
plt.xlabel(continuous_column)
plt.title('
```

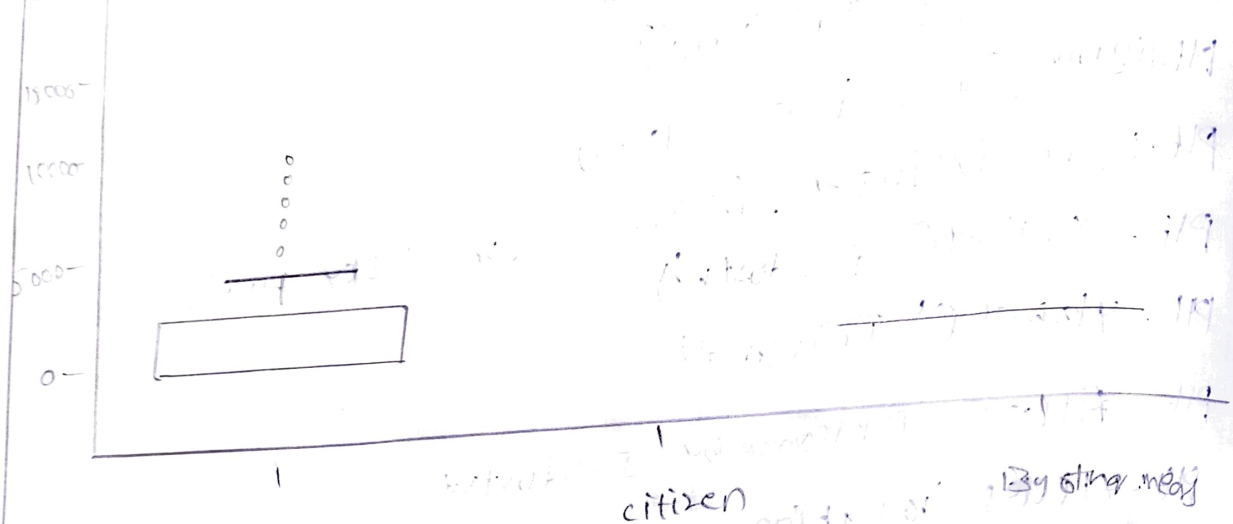

1000
500
0
2000
0
Communication service transfer
Health care

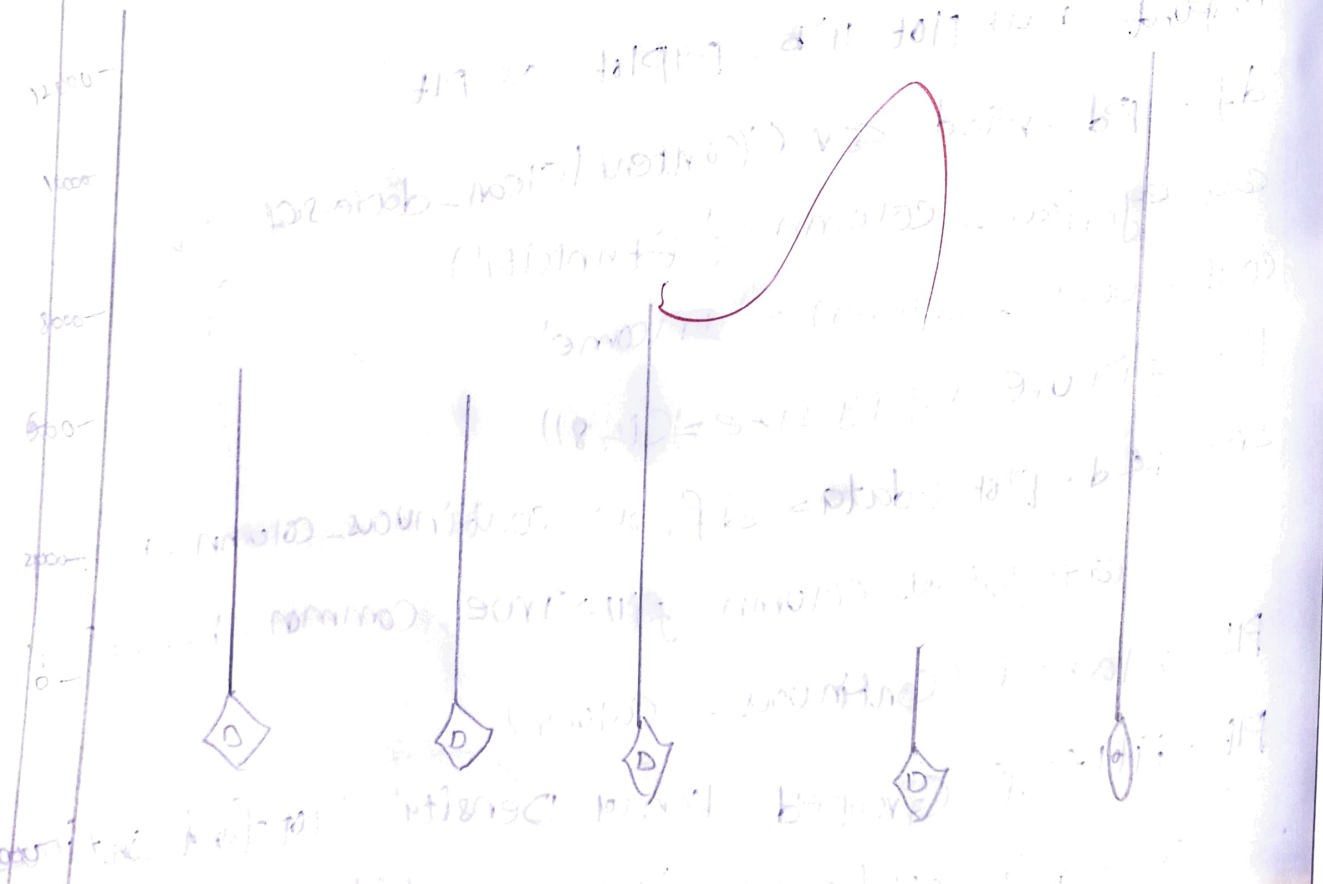

0.07
0.04
0.03
0.00

```
categorical_column = ['citizen']
continuous_column = ['Income']
plt.title(f'Grouped bar plot for {categorical_column}')
plt.show()
```

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
df = pd.read_csv('/content/clean_dataset.csv')
categorical_column = 'Efficiency'
continuous_column = 'Income'
plt.title(f'Column {categorical_column} by {continuous_column}')
plt.show()
```

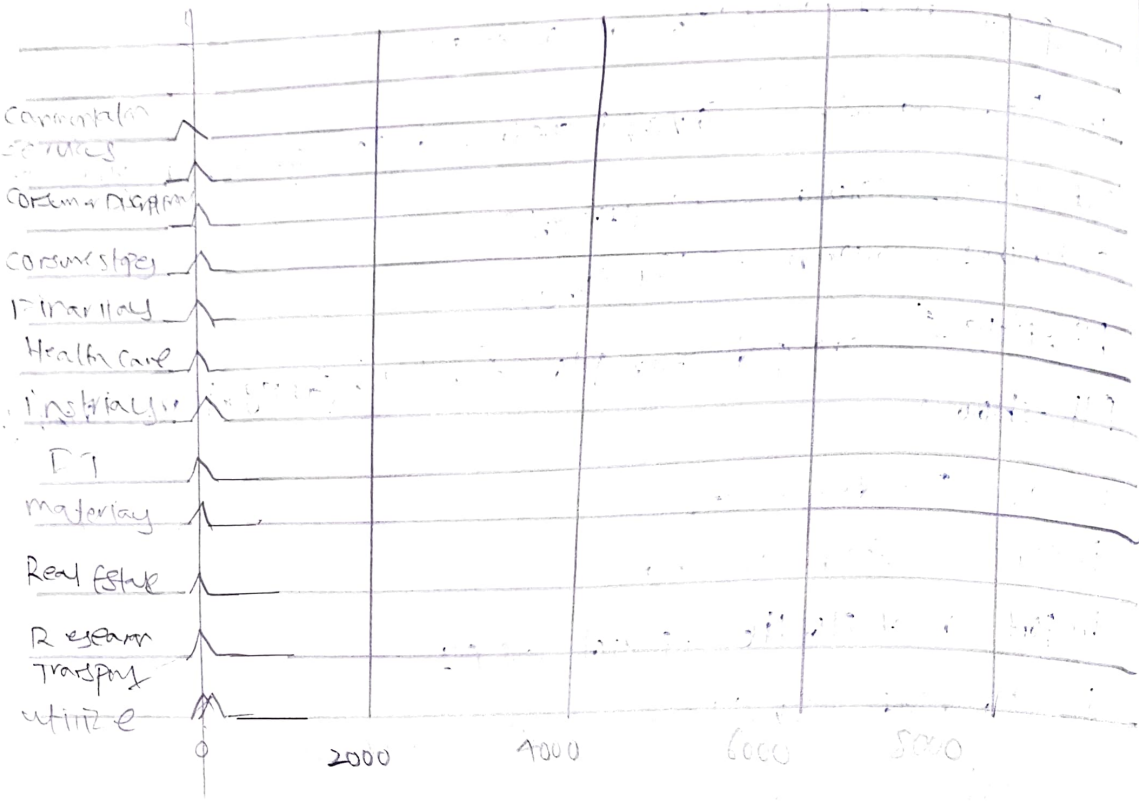
```
import pandas as pd
from joyplot import joyplot
import matplotlib.pyplot as plt
df = pd.read_csv('/content/clean_dataset.csv')
categorical_column = 'Industry'
continuous_column = 'Income'
plt.figure(figsize=(18,8))
plt.title(f'Ridge line plot for {continuous_column} by
                {categorical_column}')
```

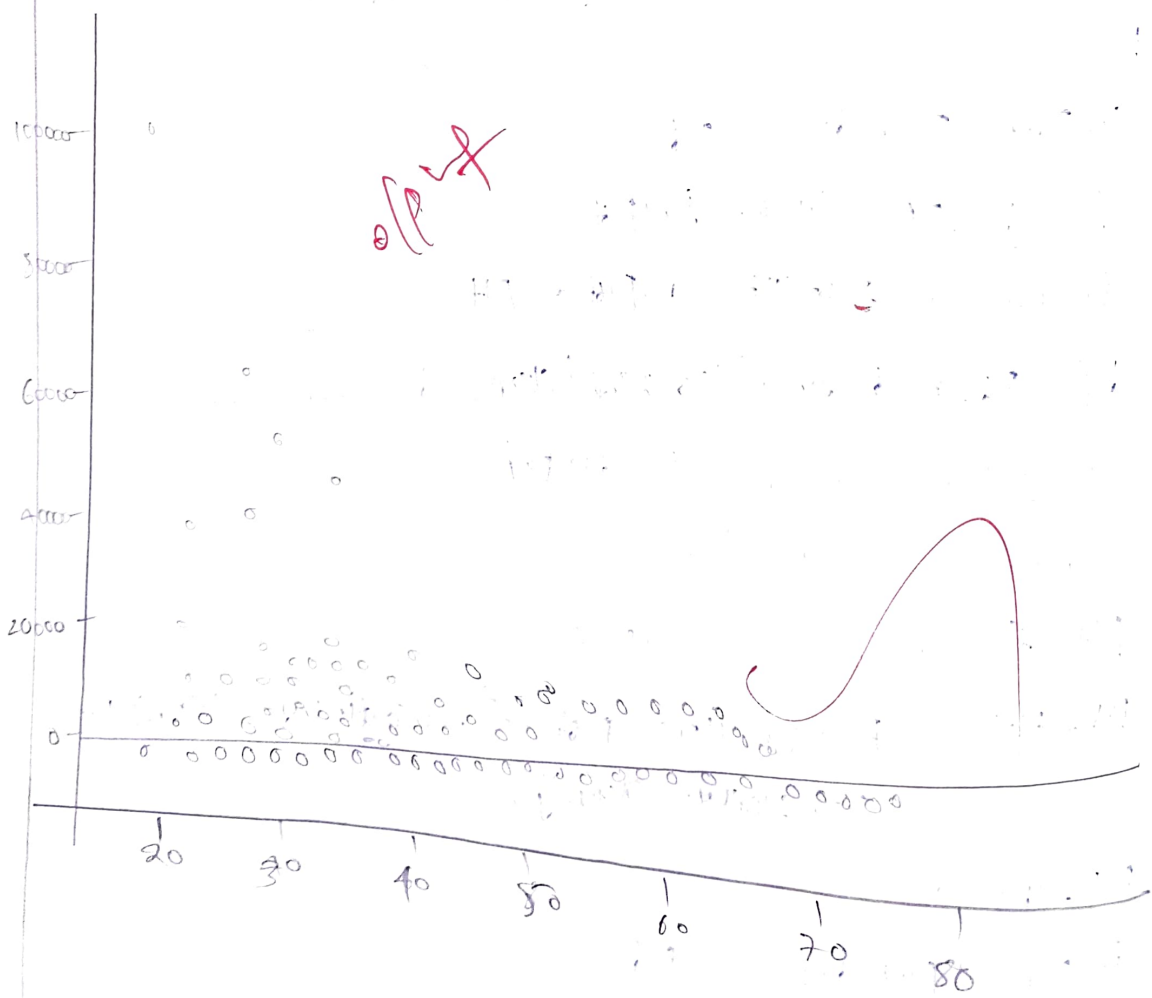

citizen
By Birth



```
import seaborn as sns
import matplotlib.pyplot as plt
df = pd.read_csv('/content/clean_data_set_csv', nrows=10)
categorical_column = ['citizen']
continuous_column = ['Income']
plt.title(f'Grouped bar plot for {categorical_column}')
plt.show()
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
df = pd.read.csv('/content/clean_dataset_csv')
categorical_column = 'Efficiency'
continuous_column = 'Income'
plt.title(f'column {categorical_column}')
plt.show()
import pandas as pd
from joypy import joyplot
import matplotlib.pyplot as plt
df = pd.read_csv('/content/clean_dataset_csv')
categorical_column = 'Industry'
continuous_column = 'Income'
plt.figure(figure=(18,8))
plt.title(f'Ridge line plot for {continuous_column} by
          {categorical_column}')
```

Ridgeline Plot for income by industry





```
import matplotlib.pyplot as plt
df = pd.read_csv('C:\Content\Dv task_4.csv')
sns.lmplot(data=df, x='Age', y='Income', height=6,
           line_kws={'color':'red'})
plt.title('Scatter Plot with Fit line')
plt.xlabel('Age')
plt.ylabel('Income')
plt.show()
```

VELTECH	
EX No.	3
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	15
TOTAL (20)	4
WITH DATE	13+4=17

Result:-

Thus, the visualize and perform Bivariate analysis using continuous and categorical data is can be executed successfully.

11/8/25 Task-4 analysis using multiple variables involving multiple measures

Aim:-

To visualize and perform multivariate analysis using multiple variables involving multiple measures

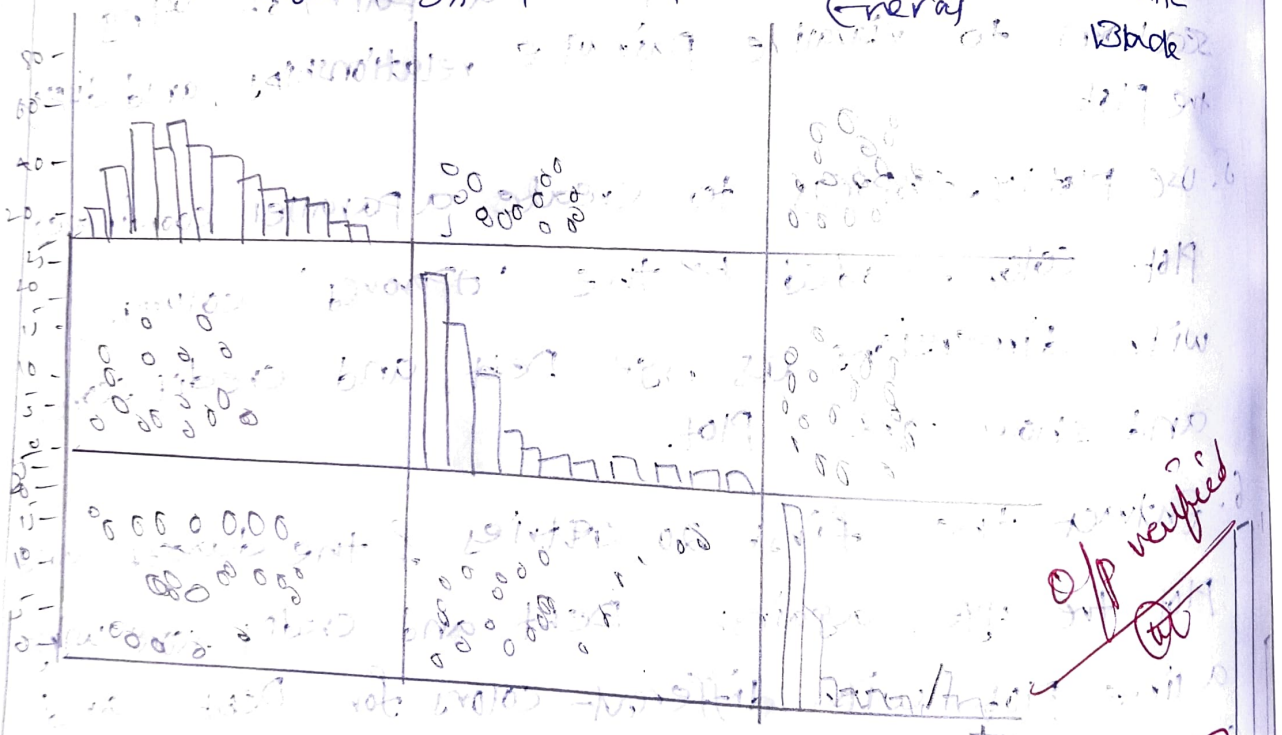
Algorithm:-

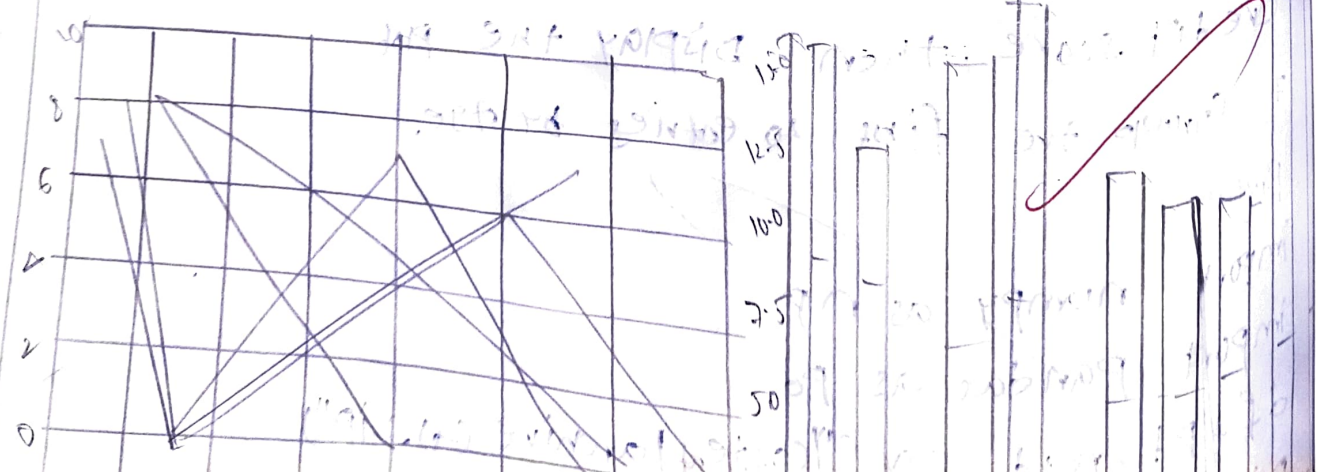
1. Import necessary libraries including Numpy, Pandas, seaborn, matplotlib, pyplot, and Plotly.express
2. Load the dataset from a zip file using using pandas 'read_csv' function
3. Print the loaded dataset to inspect its structure and contents
4. Select numeric columns, create a pairplot using seaborn to visualize pairwise relationships, and display the plot
5. Use Plotly.express to create a parallel coordinates plot, color-coded by the 'Approved' column, with dimensions as Age, Debt, and credit score, and show the plot.
6. Extract the first 20 entries of the dataset and plot the Age against Debt and credit score using a line plot with different colors for Debt and credit score, then Display the plot
7. Group the first 20 entries by Age.

Program:-

```
import numpy as np
import pandas as pd
df = pd.read_csv("/content/archive (4).zip")
print(df)
import seaborn as sns
import matplotlib.pyplot as plt
```

1	0	58.67	4.460			materials	white
2	0	24.50	0.500	1		materials	black
3	1	27.83	1.540	1		Industrials	black
4	1	20.17	5.625	1		Industrials	white
685	1	21.08	10.085	0	0	Education	Black
686	0	22.67	0.750	1	1	Energy	white
687	0	25.25	13.500	0	0	healthcare	Latino
688	1	17.92	0.205	1	1	Consumer staples	white
689	1	35.00	3.375	1	1	Energy	Black





```
numeric_subset = df[['Age', 'Debt', 'years employed']]
sns.pairplot(numeric_subset)
plt.show()
import pandas as pd
import plotly.express as px
attributes = ["Age", "Debt", "credit score"]
r_continuous_mid_point = 0)
fig.show()
first_20_entries = df.head(20)
age = first_20_entries['Age']
Debt = first_20_entries['Debt']
credit_score = first_20_entries['credit score']
plt.plot(age, credit_score, label='credit score', color='green')
plt.xlabel('Age')
plt.ylabel('value')
plt.title('Line Graph: Age, Debt and credit score')
plt.legend()
plt.grid(True)
plt.show()
grouped_data = first_20 entries.groupby('Age')[['Debt',
    'credit score']]
plt.xlabel('Age')
plt.ylabel('value')
plt.title('stacked Bar chart: Debt and credit score by Age')
plt.legend(title='Attribute')
plt.show()
```

VEL TECH	
EX No.	11
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	3
RECORD (5)	3
TOTAL (20)	3+3+3=16
SIGN WITH DATE	

11/8

Result:-

Thus, the visualize and perform multivariate analysis using multiple variables is executed successfully.

Aim:-

Visualization for Trees

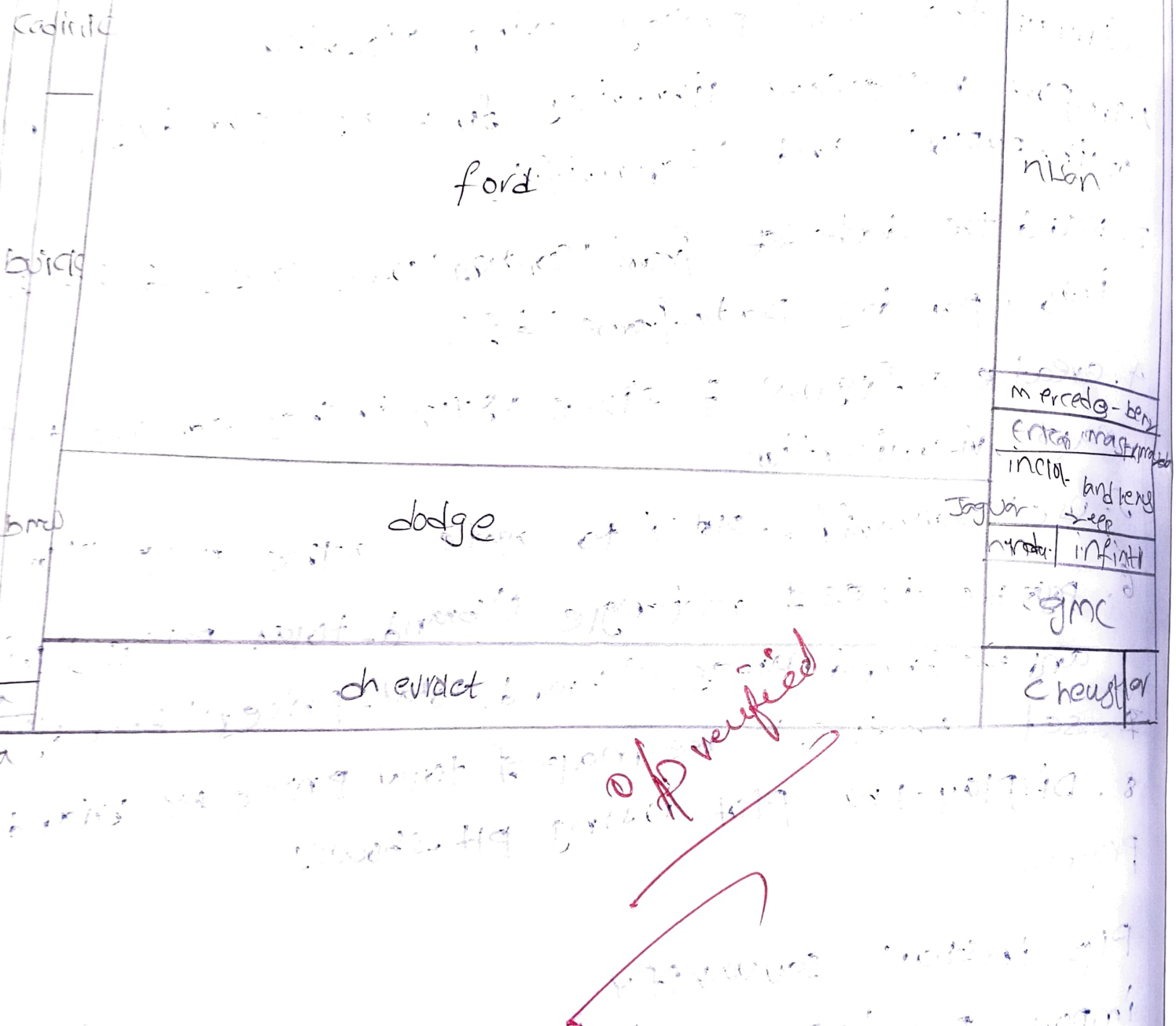
construct a treemap display on a real-world dataset. You can download the dataset link. Us cars' data was scraped from Auction Export.com.

Algorithm:-

1. Install the 'squarify' package using pip.rify
2. Import necessary libraries such as 'pandas', 'matplotlib.pyplot', and 'squarify'.
3. Read the dataset from '/content/USA_cars_dataset.csv' into a pandas Dataframe 'df'.
4. Create a figure of size 18x12 inches for the tree map visualization
5. Use 'squarify.plot()' to create a tree map plot
6. Pass the sizes of rectangle ('brand_total_price['price']) and the brand names. ('brand_total_price['brand']).
7. set the title. 'Tree map of total price by brand'.
8. Display the plot using plt.show()

Program:-

```
Pip install squarify
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import squarify,
```


ford
dodge
chevrolet
nissan
infiniti
gmc
chrysler
not verified

```
plt.figure(fig size=(18,12))
squarify.plot(sizes=brand_total_price['price'],
label=brand_total_price['brand'] alpha=0.4)
plt.axis('off')
plt.title('Tree Map of Total price by Brand')
plt.show()
```

Result:-

Thus, the design and perform visualization for Treess are executed successfully.

Aim:-

To build a sunburst display using the above program dataset.

Algorithm:-

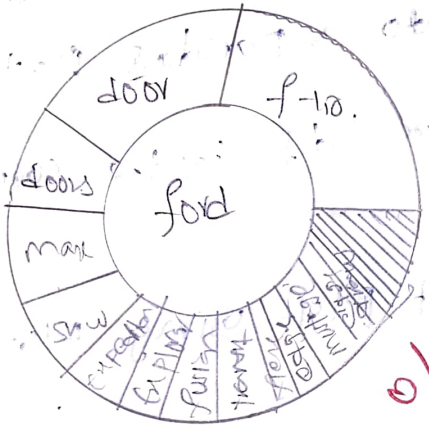
1) install the required packages 'squarify', and 'plotly' using pip
2) import necessary libraries: 'pandas', 'numpy', 'matplotlib.pyplot', 'seaborn', 'squarify', and 'plotly.express',
3. Read the dataset into a pandas Dataframe. 'df'
4. Group the Dataframe 'df' by the combination of 'brand' and 'model'
5. create a sunburst plot using plotly express. ('px.sunburst()');
6. Display the plot using 'fig.show()',

Program:-

```
pip install squarify
pip install plotly
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import squarify
```

output :-

sunburst. Display of total price by Brand and model



O/p verified

RESU
Thur,
Progra

and model)

fig.show()

VEL TECH	
EX No.	5
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	3
RECORD (5)	+3
TOTAL (20)	+3 +3 =16
SIGN WITH DATE	

Result:-

Thus, the Build a sunburst display using above program dataset are executed successfully

AIM:-

TO Design and Perform visualization for graphs and Networks

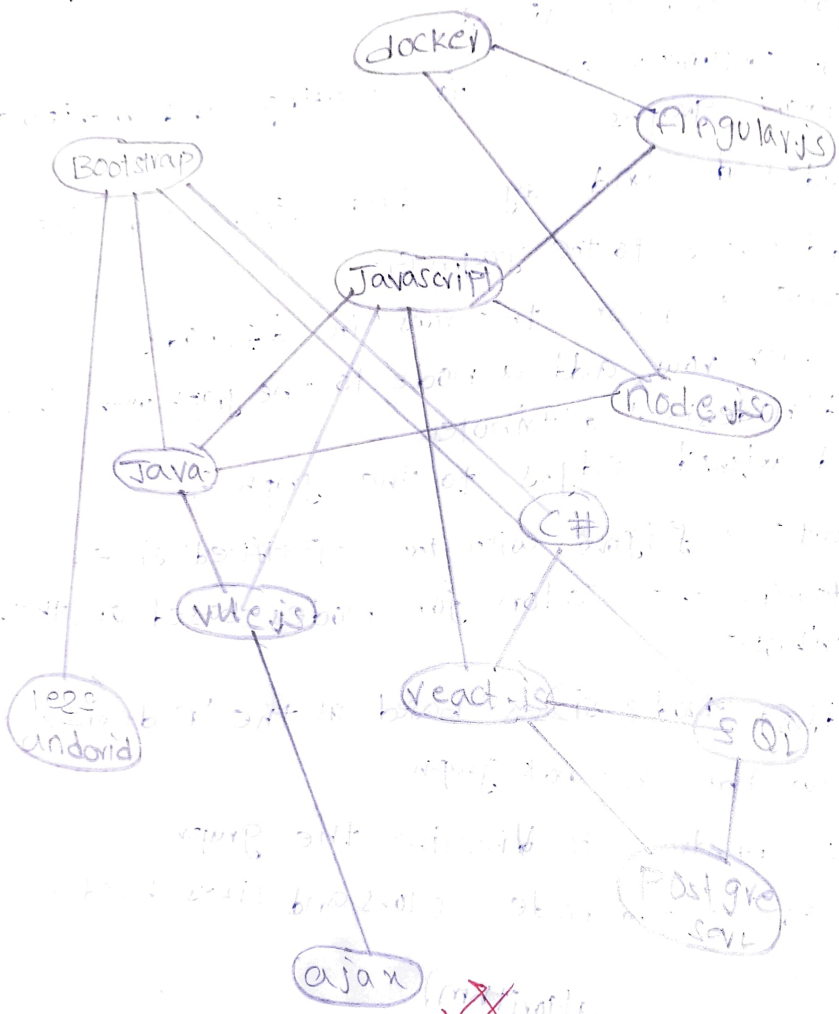
Algorithm:-

1. Import necessary libraries:
2. Import Network x as nx for creating and manipulating network graphs.
3. Read node and edge data from csv files
4. Add nodes to the graph G:
   Iterate through the rows in df_nodes.
   for each row, add a node to the graph with a 'name', 'group' and 'nodesize' attributes.
   Add weighted edges to the graph G.
5. Create a figure with the specified size.
   Determine the colors for nodes based on their 'group' attribute
6. Adjust node sizes based on the 'nodesize' attribute
   Draw the network graph
7. use nx.draw to visualize the graph
   customize the node colors and sizes based on the calculated values.
8. End of the algorithm.

program :-

```
import network as nx
import matplotlib as plt
import pandas as pd
G = nx.Graph(day = "stackoverflow
```

output:-


docker
Angularjs
Bootstrap
Javascript
nodejs
Java
C#
vuejs
reactjs
SQL
android
Postgre SQL
ajax

o/p

```
node size = row ('nodesize'))
for index, row in df_links.iterrows():
node size = row ('nodesize'))
for index, row in df_links.iterrows():
G.add_weighted_edges_from([row['source'], row['target'],
row ('value')])
plt.figure (figure size=(15,15))
options = {
    'edge_color': '#FFDEA2',
    'width': 1.5,
    'with_labels': True
    'font_weight': 'regular',
}
colors = [color_graph(map(G.nodes [node]('groups')) for
 node in G]
sizes = [G.nodes[node] ('nodesize')] *25 for node
 in G]
nx.draw (G, node_color = colors, node_size = sizes,
pos = nx.spring_layout (G, k=1.5, iterations =15),
** options)
ax = plt.gca()
ax.collections[0].set_edgecolor ('#555555')
plt.show()
```

VEL TECH	
EX No.	6
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	4
RECORD (5)	+2
TOTAL (20)	4+2=16
SIGNATURE	

25/8

Result:-

Thus, The design and perform visualizations for graphs and networks are executed successfully.

1/9/25 Task - 7(a) Analysis and visualization of

AIM:-

To build a word cloud using the tag cloud tool that visualises the frequency of words from text containing personal details; where the size of each word represents how frequently it appears

Algorithm:-

Import necessary libraries.

Import wordcloud from the word cloud library for word cloud creation

Import matplotlib.pyplot as plt for data visualization

Define the text from which you want to create a word cloud, in details variable

create a wordcloud object with specific parameters

provides list of stop words to exclude common words

set the minium font size is displayed words. Create a Matplotlib figure for displaying word cloud.

Define the figure size (5*5 inches) and specify the face color, which is transparent

Generate the word cloud image using the generate method of the word cloud object, passing in the text from the steps

Display the word cloud image:

use plt.imshow to display the word cloud. Turn off the axis to remove axis labels. Adjust the layout to minimize padding. show the word cloud using plt.sho

End of the algorithm.

output :-

effective
visualization
can use business students
will visualize the
for design and
graphic tools
data.

```
import matplotlib.pyplot as plt
details = "Data visualization is the visual and
interactive exploration and graphic representation
of data of any time - This course covers data
visualization concepts, practices, and tools particularly
for analyzing and presenting business idea data
students will evaluate, design, and develop effective
visualizations and dashboards, using various developme
tools".
word cloud = wordcloud (width = 800, height = 800,
background_color = 'white',
stopwords = [],
min_font_size = 10).generate (details)
plt.figure (figsize = (5,5), facecolor = None)
plt.axis ("off")
plt.tight_layout (pad = 0)
plt.show ()
```

RESULT:-

Thus, the Generation of insight portnetwork plot graph

TO Generate insight using text network Analysis and visualization.

AIM:-

TO Build a word cloud using the tag cloud tool that visualizes the frequency of words from text containing personal details. where the size of each word represents how freavency it appeas

Algorithm:-

1) Input Data
   - collect text data

2) preproces the data which is given in the text

3) count how many times each word appears in the text.

4) word cloud generation
   - map freavency value to word size
   - Generate the cloud image where more freavency word appears bigga

5) Display the generated word cloud.

program:-

```
From wordcloud import word cloud, stop words
import matplotlib.pyplot as plt
text = "My name is lavanik. I am cse(AIML) student.
I love python, ML&AI and Data visualization. my
hobbies are coding My skills include python"
stop words = set (stop WORDS)
word cloud = word cloud ( width = 800, weight = 400, bagroud.
  color = 'white', stopwords = stop words, color map = 'viridis'
          min_font size = 10 %. generate (text)
```

Output:-

computer
python
machine
Data Learning
science
web Passionate
Development Competitive

```
plt.imshow(word_cloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

VEL TECH	
EX No.	7
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	3
RECORD (5)	5
TOTAL (20)	18
Sign	

Result:-

Thus, the Generate insights using text network mappings and Data visualization are executed successfully

1508/9/25

Task-8

AIM :-

To analyze and visualize spatial and geospatial data using geographical maps and map projections, enabling better understanding of spatial relationships and geographic patterns.

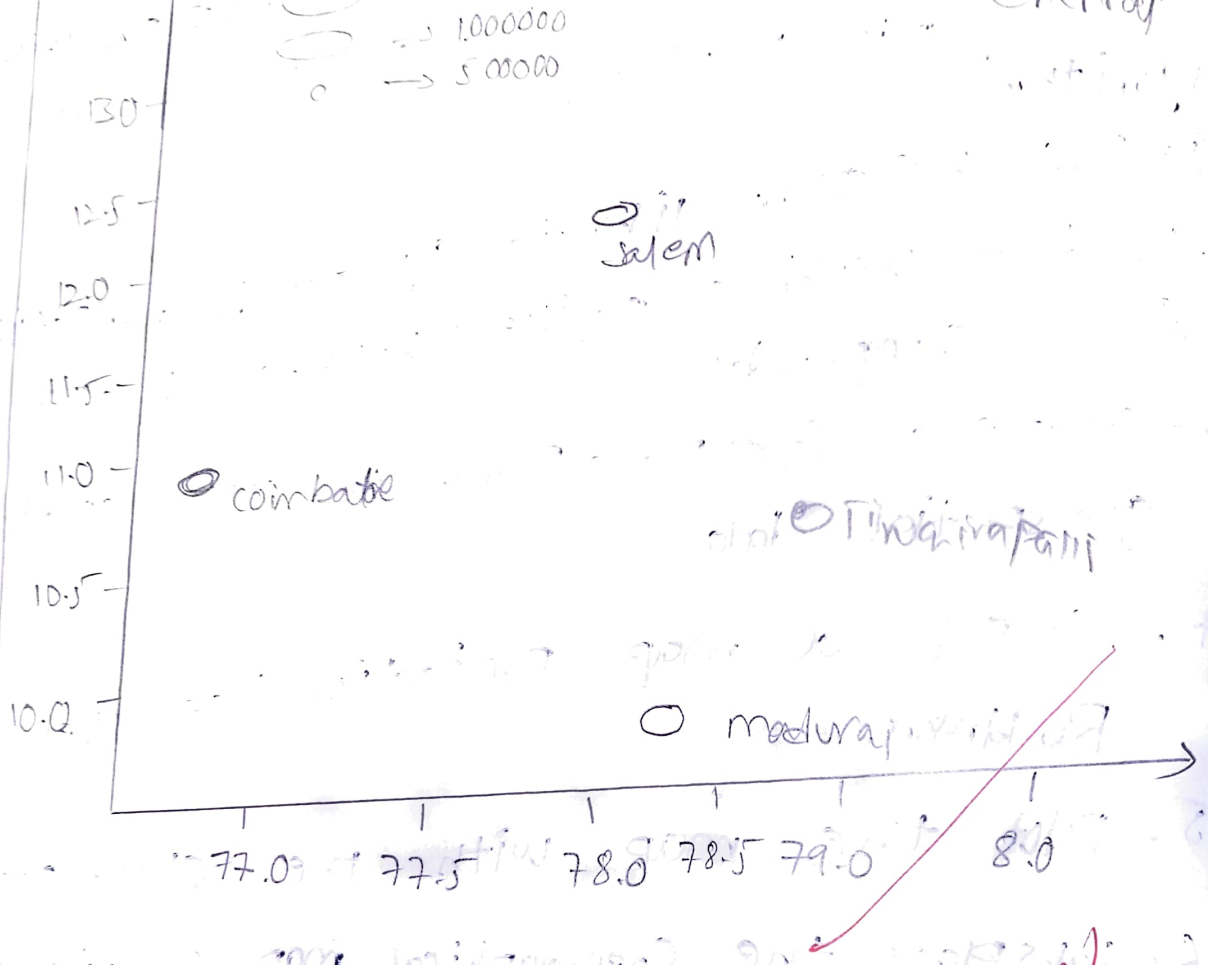
Algorithm

1. Import necessary libraries (geopandas, matplotlib).
2. Load a world shapefile (built-in dataset from geopandas)
3. Create a GeoDataFrame to store spatial/geospatial data
4. Select a map projection (eg.. Mercator, Robinson.
5. Plot the map with the chosen projection
6. Display the Geographical map for visualization

Program

```
import GeoPandas as Pd
import matplotlib.pyplot as PH

world =
 gPd.read_file(gPd.datasets.get_path('naqura
 kearth_lowres'))
 world.Plot()
PH.title("world Map - Default projection")
PH.show()
ax=
world.to_crs('+proj=robin').Plot(figsize=(edge
color="black")
```


1,000000
5 00000
13.0
12.5
12.0
11.5
11.0
10.5
10.0
salem
coimbatie
Tiruchirapalli
madurai
77.0
77.5
78.0
78.5
79.0
8.0

```
plt.show()                                    Projection")

ax =
world.to_crs("EPSG:3395").plot(figsize=(10,6)
color = "lightgreen",
edgecolor = "black")
plt.title("World Map - Mercator Projection")
plt.show()
```

VEL TECH	
EX No.	8
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	4
RECORD (5)	5
TOTAL (20)	19
SIGN WITH DATE	

Result:-

Thus, the visualization of spatial and Geospat data Geographical map, map projections are

AIM

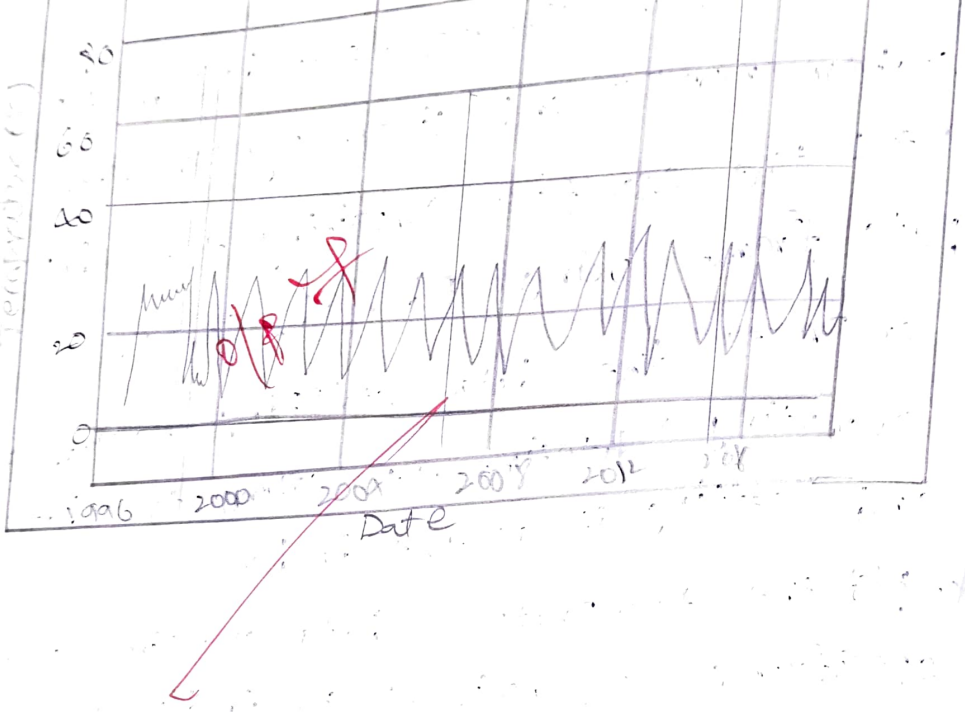
To analyze and visualize Time oriented Data using Line graph, trend lines, area chart

Algorithm

1. Import necessary libraries : pandas, matplotlib.pyplot, seaborn, numpy, and linear regression from sklearn.
2. Read the csv file containing the dataset into a Data frame using Pd.read_csv().
3. preprocess the data stripping column names, converting 'datetime_utc' column to datetime format and setting it as the index
4. Plot the temprature data over time using sns.lineplot() specifying the x-axis as the index of the Data frame and y-axis as the temprature column ('_tempm').
5. customize the plot by adding a title, labels for the x & y axis. enabling grid lines, rotating x-axis labels Plot using Plot.show().

Program

```
import pandas as Pd
import matplotlib.pyplot as Plt
import seaborn as sns
import numpy as np
from sklearn.linear_model import LinearRegression
data = Pd.read_csv('/content/test-set.csv')
data.columns = data.columns.str.strip()
data['datetime_utc'] = Pd.to_datetime(data['datetime_utc'])
data.set_index('datetime_utc', inplace=True)
Plt.figure(figsize=(10,6))
sns.lineplot(data=data, x=data.index, y='_tempm')
```



Trend li

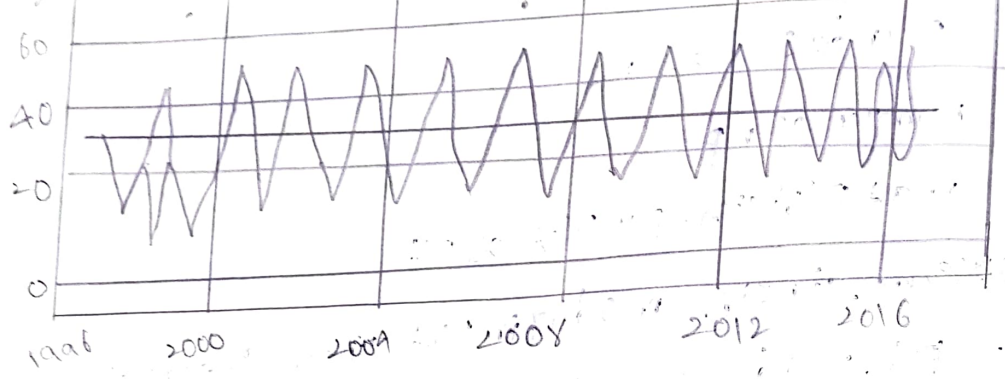
import
impor
impor
from s
data =
data =

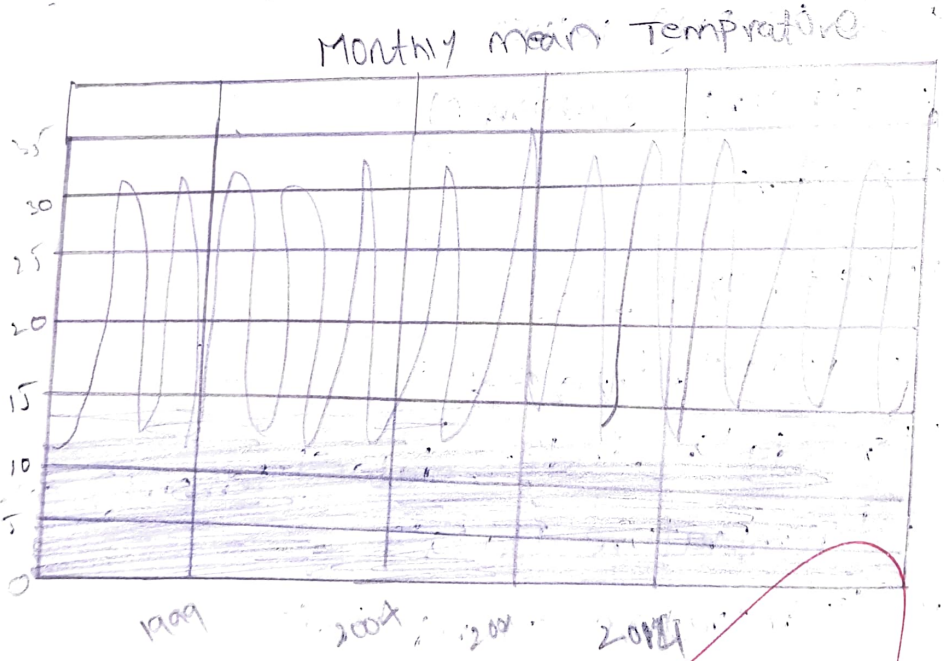
x = dat
y = dat
mode
mode.f
slope =
intercept
plt.scat
label
plt.title
plt.xlab
plt.ylab
plt.grid
plt.leger
plt.xtick

```
plt.title('Line Graph of Temp
    Temperature')
plt.tight_layout()
plt.show()
```

Trend lines

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
data = pd.read_csv('/content/testset.csv')
data = ['_tempm'].fillna(data['_tempm'].mean()
    inplace=True)
X = data.index.astype(int).values.reshape(-1,1)
Y = data['_tempm'].values
model = LinearRegression()
model.fit(X,Y)
slope = model.coef_[0]
intercept = model.intercept_
plt.scatter(data.index, data['_tempm'], color='
    label='Temprature Data')
plt.title('Temprature Trend Line')
plt.xlabel('Date')
plt.ylabel('Temprature (°C)')
plt.grid(True)
plt.legend()
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```





data
plt.f
plt.
plt.
plt.y
plt.t
plt.

Resu
THUS
Trend
and

Area chart

```
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv('/content/test-set.csv')
data['datetime_utc', inplace = True)
plt.figure(figsize=(10,6))
plt.title('Monthly Mean Temprature')
plt.xlabel('Date')
plt.ylabel('Temprature')
plt.tight_layout()
plt.show()
```

VEL TECH	
EX No.	9
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	3
RECORD (5)	5
TOTAL (25)	18
SIGN WITH DATE	

Result

Thus, we successfully implemented the line graph, Trend line, Area chart by Time oriented Data and out is verified.

Earthquake and Geospatial Data Analysis

Batch Mates:

1. VTU24801 JINKALA SHASHIDHAR
2. VTU24802 SHAIK JAKEER HUSSAIN
3. VTU24858 SHAIK MOHAMMED SHAHID
4. VTU24867 KUSUMARAJU BUKARI BABA
5. VTU24905 SHAIK RUBIYA
6. VTU24923 MALLU RUDRA TEJESHWAR REDDY
7. VTU25011 RAAVI NAVADEEP CHOWDARY
8. VTU25104 DONGALA JAYANDAR
9. VTU25106 MADAKKA GARI KRISHNA KOWSHIK
10. VTU25112 SEELAM NAVEEN

Aim:

To analyze and visualize earthquake occurrences using geospatial data.
The goal is to identify regions with the highest seismic activity, visualize earthquake magnitudes on maps,
and observe temporal patterns to support disaster risk assessment and preparedness.

Algorithm:

1. Load the dataset (**earthquake_data.csv**) containing Date, Latitude, Longitude, Magnitude, Depth, Region.
2. Convert **Date** into datetime format and handle missing or invalid entries.
3. Filter earthquakes based on magnitude thresholds (e.g., ≥4.0 for moderate and above).
4. Compute:
   - Average magnitude per region
   - Total earthquakes per region
   - Maximum depth and strongest quake recorded
5. Visualize data using:
   - Geographic Scatter Map (Latitude vs Longitude) color-coded by magnitude
   - Bar chart showing top earthquake-prone regions
   - Line plot showing monthly earthquake counts

6. Generate summary metrics (average magnitude, deepest quake, most active region).
7. Export visualizations and summary results as CSV/PNG files if needed.

Program:

```
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px

df = pd.read_csv("earthquake_data.csv")

df['Date'] = pd.to_datetime(df['Date'], errors='coerce')
df['Magnitude'] = pd.to_numeric(df['Magnitude'], errors='coerce')
df = df.dropna(subset=['Latitude', 'Longitude', 'Magnitude'])

avg_mag = df['Magnitude'].mean().round(2)
max_mag = df['Magnitude'].max()
most_active_region = df['Region'].value_counts().idxmax()

print("\n=== Earthquake Data Summary ===")
print(f"Average Magnitude     : {avg_mag}")
print(f"Strongest Earthquake  : {max_mag}")
print(f"Most Active Region    : {most_active_region}")

fig = px.scatter_geo(df, lat='Latitude', lon='Longitude', color='Magnitude',
                     title='Global Earthquake Distribution by Magnitude',
                     color_continuous_scale='Reds', size='Magnitude', projection='natural earth')
fig.show()
```

```
region_counts = df['Region'].value_counts().head(10)
plt.figure(figsize=(10,6))
plt.bar(region_counts.index, region_counts.values, color='orange')


plt.title("Top 10 Most Earthquake-Prone Regions")
plt.xlabel("Region")
plt.ylabel("Number of Earthquakes")
plt.xticks(rotation=45, ha='right')
plt.grid(axis='y', linestyle='--', alpha=0.4)
plt.tight_layout()
plt.show()

monthly = df.groupby(df['Date'].dt.to_period('M')).size()
monthly.plot(kind='line', figsize=(10,5), color='green')
plt.title("Monthly Earthquake Occurrence Trend")
plt.xlabel("Month")
plt.ylabel("Number of Earthquakes")
plt.grid(True)
plt.tight_layout()
plt.show()
```

Output:

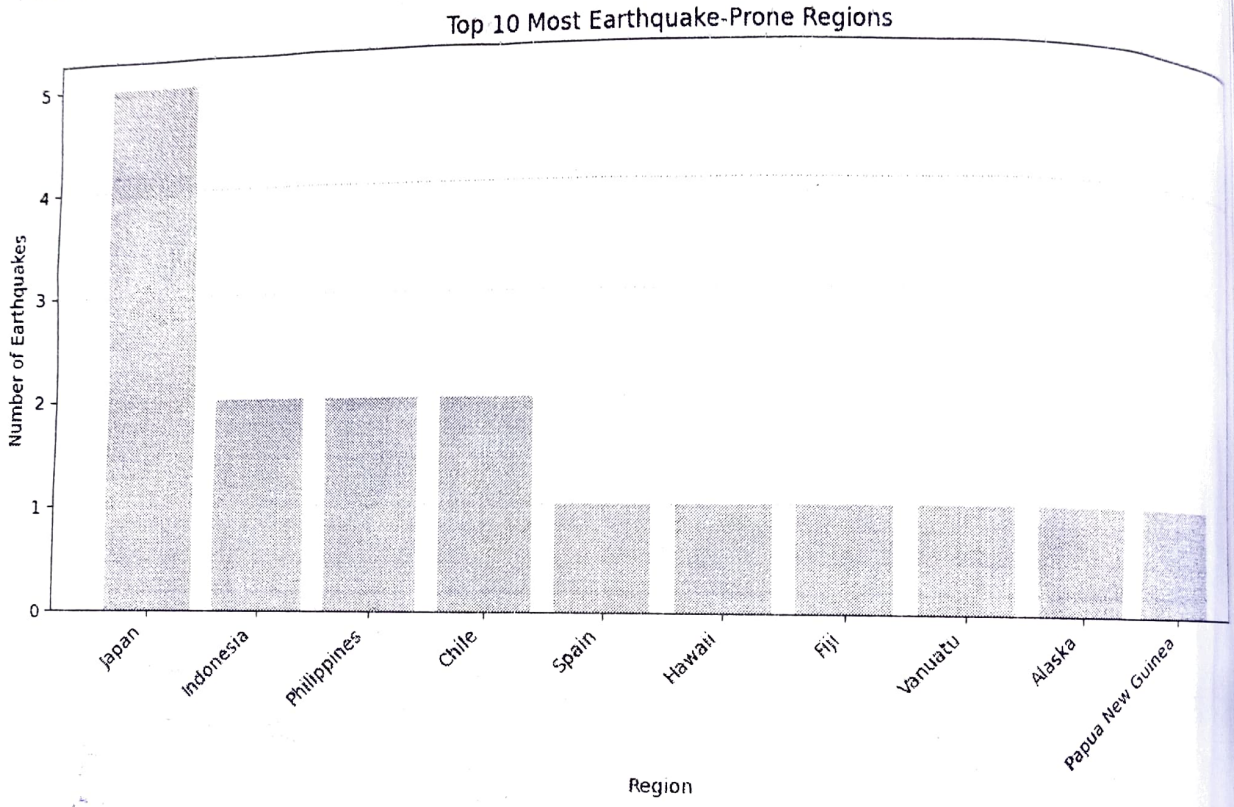
=== Earthquake Data Summary ===

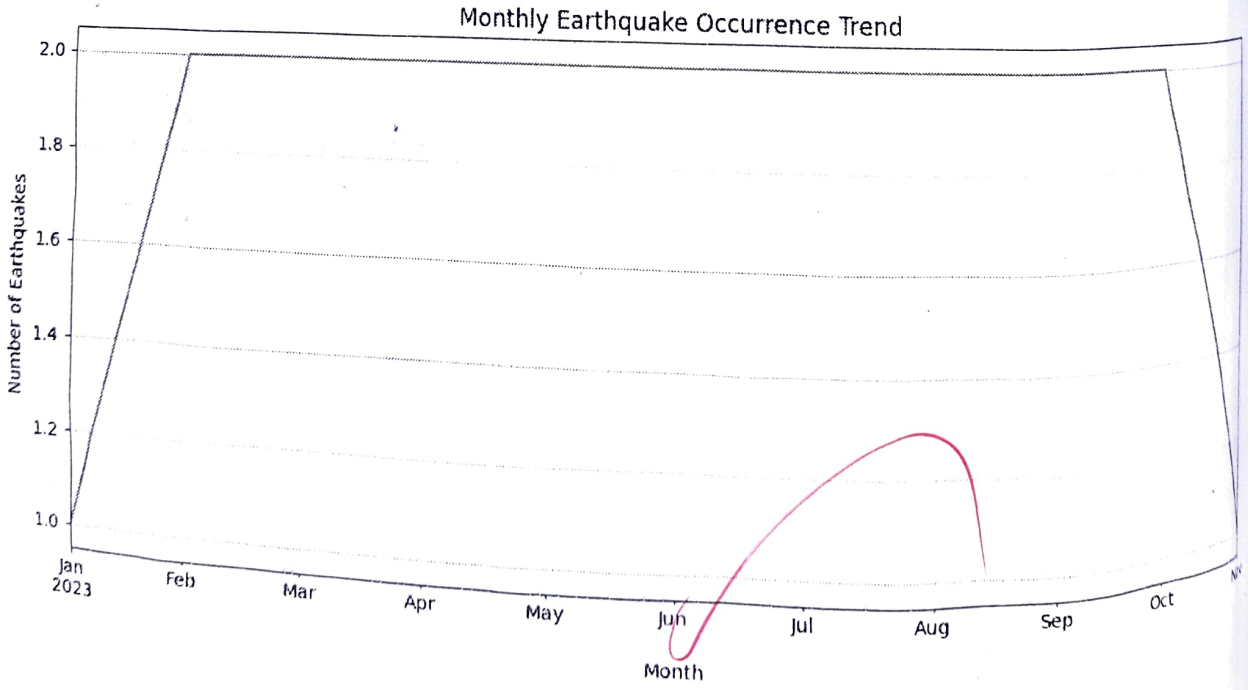
Average Magnitude : 5.98

Strongest Earthquake : 8.3

Most Active Region : Japan

4.555.566.577.58MagnitudeGlobal Earthquake Distribution by Magnitude





VEL TECH	
EX NO.	VC1
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	2
RECORD (5)	5
TOTAL (20)	17
SIGN WITH DATE	

23/10/25

Result:

Thus, earthquake occurrences were successfully analyzed and visualized using geospatial data. The analysis revealed regions with the highest seismic activity, identified the strongest earthquakes, and showcased temporal patterns in frequency, enabling better understanding of global earthquake behavior and aiding disaster management planning.

USE CASE – 2

Performance of Different Company Departments Over Year

Batch Mates :

1. VTU24801 JINKALA SHASHIDHAR
2. VTU24802 SHAIK JAKEER HUSSAIN
3. VTU24858 SHAIK MOHAMMED SHAHID
4. VTU24867 KUSUMARAJU BUKARI BABA
5. VTU24905 SHAIK RUBIYA
6. VTU24923 MALLU RUDRA TEJESHWAR REDDY
7. VTU25011 RAAVI NAVADEEP CHOWDARY
8. VTU25104 DONGALA JAYANDAR
9. VTU25106 MADAKKA GARI KRISHNA KOWSHIK
10. VTU25112 SEELAM NAVEEN

Aim:

To analyze and visualize yearly department performance using time-series plots and summary metrics to identify trends, best/worst months, and overall strengths or weaknesses.

Algorithm: 1. Define departments and the time dimension (months for the year).

2. Acquire or generate monthly performance metrics for each department (normalized scores, KPIs, etc.).
3. Build a tidy DataFrame: rows = months, columns = departments.
4. Compute derived metrics: yearly average per department, month-over-month changes, best/worst months.
5. Visualize:
   - Line chart: performance over months (one line per department).
   - Bar chart: average yearly performance per department.
6. Export charts and data for reporting (PNG/CSV/PDF).

Program:

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

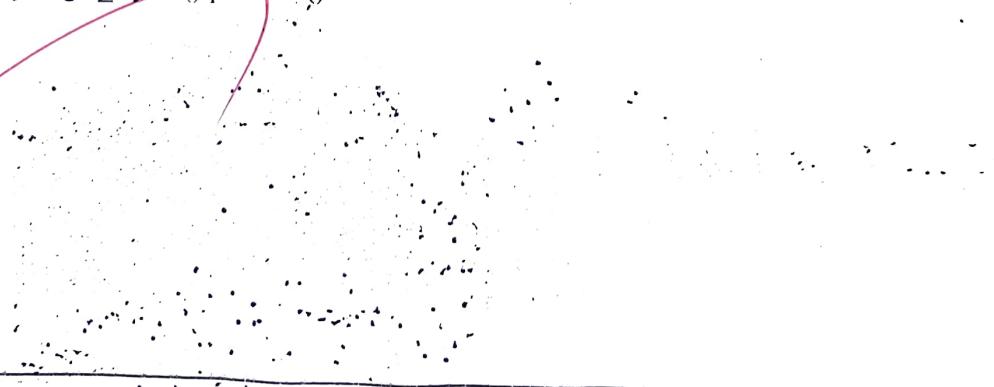
```
df=pd.read_csv("/content/drive/MyDrive/ColabNotebooks/department_performance_2024.csv")
df.set_index('Month', inplace=True) yearly_avg =
df.mean().round(2).sort_values(ascending=False)

best_month = df.idxmax() worst_month
= df.idxmin() monthly_changes =
df.diff().round(2)

print("\n===  Yearly  Average  Performance  (Descending)  ===")
print(yearly_avg) print("\n=== Best and Worst Month per Department ===")
for dept in df.columns:    print(f"{dept}: Best = {best_month[dept]},  Worst =
{worst_month[dept]}")

plt.figure(figsize=(10,6)) for
dept in df.columns:
   plt.plot(df.index, df[dept], marker='o', linewidth=2, label=dept)
plt.title("Monthly  Performance  by  Department")
plt.xlabel("Month")  plt.ylabel("Performance  Score  (0-100)")
plt.xticks(rotation=45)  plt.grid(axis='y', linestyle='--', alpha=0.4)
plt.legend() plt.tight_layout() plt.show()

plt.figure(figsize=(8,5))
plt.bar(yearly_avg.index,              yearly_avg.values)
plt.title("Average Yearly Performance by Department")
plt.xlabel("Department")              plt.ylabel("Average
Performance  Score  (0-100)")  plt.xticks(rotation=30)
plt.tight_layout() plt.show()
```



OUTPUT:

=== Yearly Average Performance (Descending) ===

Sales	70.30
Finance	57.68
Marketing	54.05
R&D	49.08
HR	46.83

=== Best and Worst Month per Department ===

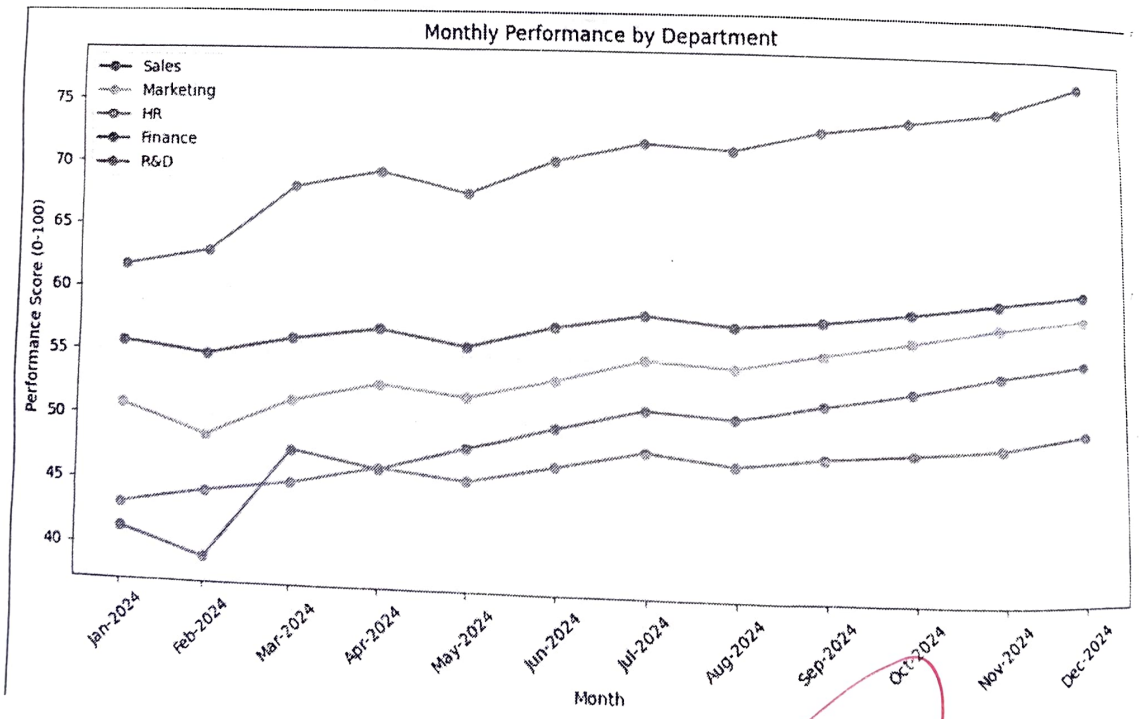
Sales: Best = Dec-2024, Worst = Jan-2024

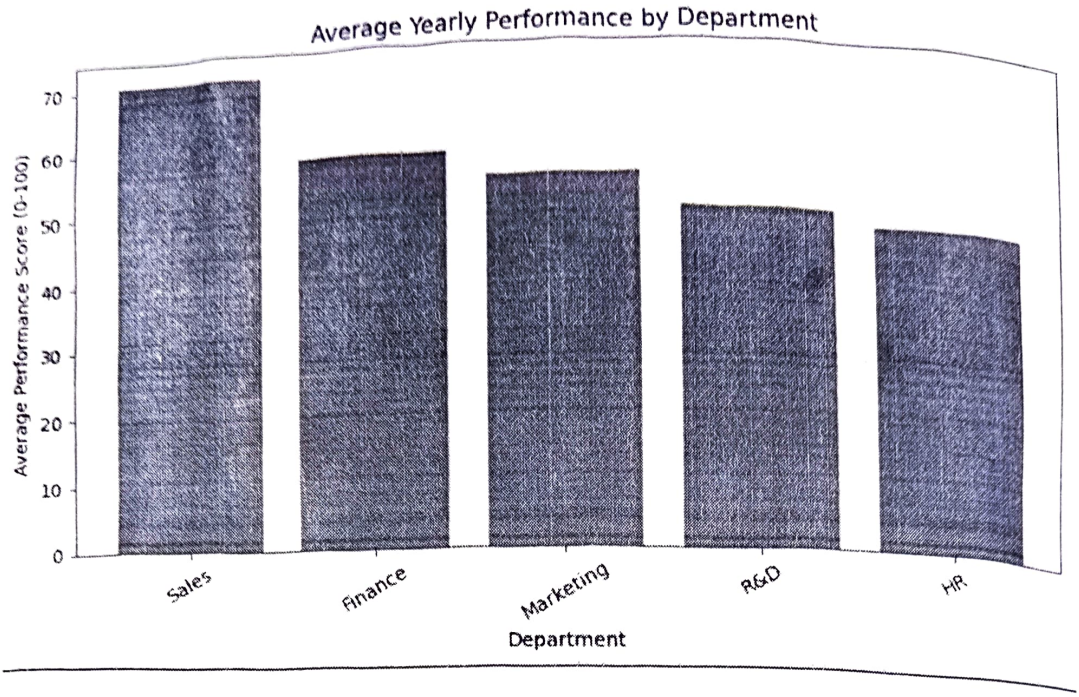
Marketing: Best = Dec-2024, Worst = Feb-2024

HR: Best = Dec-2024, Worst = Jan-2024

Finance: Best = Dec-2024, Worst = Feb-2024

R&D: Best = Dec-2024, Worst = Feb-2024




Average Yearly Performance by Department
Average Performance Score (0-100)
0
10
20
30
40
50
60
70
Sales
Finance
Marketing
R&D
HR
Department

VEL TECH	
EX NO.	UC 24
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	03
RECORD (5)	5
TOTAL (20)	18
SIGN WITH DATE	27/10

Result:

Thus, the yearly department performance was successfully analyzed and visualized, revealing key trends, best and worst months, and insights for data-driven decisions.