



SRM VALLIAMMAI ENGINEERING COLLEGE
(An Autonomous Institution)



SRM Nagar, Kattankulathur-603203

DEPARTMENT OF INFORMATION TECHNOLOGY

ACADEMIC YEAR: 2025-2026 ODD

SEMESTER

LAB MANUAL

(REGULATION - 2023)

**AD3364 – DATA EXPLORATION AND
VISUALIZATION LABORATORY**

THIRD SEMESTER

B. Tech – INFORMATION TECHNOLOGY

Prepared By

Ms. S. Jonisha, Assistant Professor (O.G) / IT

Ms. M. Kanmani, Assistant Professor (O.G) /IT

Ms. V. Abarna, Assistant Professor (O.G) / AI&DS

Mr. B. Yogesh kumar, Assistant Professor (O.G) / AI&DS

INDEX

E.NO	EXPERIMENT NAME	Pg. No.
A	PEO, PO, PSO	3-5
B	Syllabus	6
C	Introduction/ Description of major Software & Hardware involved in lab	7
D	CO, CO-PO Matrix, CO-PSO Matrix	7
E	Mode of Assessment	8
1	Install Data Analysis and Visualization tools: R/Python/Tableau Public/Power BI.	9-12
2	Perform Exploratory Data Analysis (EDA) on with datasets like email data set. Export all your emails as a dataset, import them inside a pandas data frame, visualize them and get different insights from the data.	13-16
3	Working with Numpy arrays, Pandas data frames, Basic Plots using Matplotlib.	17-21
4	Explore various variable and row filters in R for cleaning data. Apply various plot features in R on sample data sets and visualize.	22-23
5	Perform Time Series Analysis and apply the various visualizations techniques.	24-26
6	Perform Data Analysis and representation on map using map data sets with Mouse Rollover effect, user interaction, etc.	27-28
7	Build Cartographic visualization for multiple datasets involving various countries of the world; states and districts in India etc.	29-31
8	Perform EDA on Wine Quality Data Set.	32-36
9	Use a case study on a data set and apply the various EDA and visualizations techniques and present an analysis report.	37

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

1. To afford the necessary background in the field of Information Technology to deal with engineering problems to excel as engineering professionals in industries.
2. To improve the qualities like creativity, leadership, teamwork and skill thus contributing towards the growth and development of society.
3. To develop ability among students towards innovation and entrepreneurship that caters to the needs of Industry and society.
4. To inculcate and attitude for life-long learning process through the use of information technology sources.
5. To prepare then to be innovative and ethical leaders, both in their chosen profession and in other activities.

PROGRAMME OUTCOMES (POs)

After going through the four years of study, Information Technology Graduates will exhibit ability to:

PO#	Graduate Attribute	Programme Outcome
1	Engineering knowledge	Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization for the solution of complex engineering problems.
2	Problem analysis	Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3	Design/development of solutions	Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations.
4	Conduct investigations of complex problems	Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5	Modern tool usage	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools, including prediction and modeling to complex engineering activities, with an understanding of the limitations.
6	The engineer and society	Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice
7	Environment and sustainability	Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8	Ethics	Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice
9	Individual and team work	Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings
10	Communication	Communicate effectively on complex engineering activities with the engineering community and with the society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions
11	Project management and finance	Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12	Life-long learning	Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change

PROGRAMME SPECIFIC OUTCOMES (PSOs)

By the completion Bachelor of Technology in Information Technology program the student will have the following Program specific outcomes.

1. Design secured database applications involving planning, development, and maintenance using state-of-the-art methodologies based on ethical values.
2. Design and develop solutions for modern business environments coherent with advanced technologies and tools.
3. Design, plan, and set up a network that is helpful for contemporary business environments using the latest hardware components.
4. Planning and defining test activities by preparing test cases that can predict and correct errors ensuring a socially transformed product catering all technological needs.

OBJECTIVES:

- ❖ To understand the key techniques behind data visualization.
- ❖ To learn about various visualization structures.
- ❖ To evaluate the information visualization systems.
- ❖ To design and build data visualization systems.
- ❖ To analyze and identify trends in data sets.

LIST OF EXPERIMENTS

1. Install Data Analysis and Visualization tools: R/Python/Tableau Public/Power BI.
2. Perform Exploratory Data Analysis (EDA) on with datasets like email data set. Export all your emails as a dataset, import them inside a pandas data frame, visualize them and get different insights from the data.
3. Working with Numpy arrays, Pandas data frames, Basic Plots using Matplotlib.
4. Explore various variable and row filters in R for cleaning data. Apply various plot features in R on sample data sets and visualize.
5. Perform Time Series Analysis and apply the various visualizations techniques.
6. Perform Data Analysis and representation on map using map data sets with Mouse Rollover effect, user interaction, etc.
7. Build Cartographic visualization for multiple datasets involving various countries of the world; states and districts in India etc.
8. Perform EDA on Wine Quality Data Set.
9. Use a case study on a data set and apply the various EDA and visualizations techniques and present an analysis report.

TOTAL: 45 PERIODS

LIST OF EQUIPMENTS FOR A BATCH OF 30 STUDENTS

SOFTWARE:

Standalone desktops with Python 3 interpreter for Windows / Linux 30 Nos. (or) Server with Python 3 interpreter for Windows/Linux supporting 30 terminals or more.

HARDWARE:

Standalone Desktops: 30 Nos.

COURSE OUTCOMES

AD3364.1	Understand the fundamentals of exploratory data analysis.
AD3364.2	Implement the data visualization using Matplotlib.
AD3364.3	Perform univariate data exploration and analysis.
AD3364.4	Apply bivariate data exploration and analysis.
AD3364.5	Use data exploration and visualization techniques for multivariate and time series data.

CO- PO-PSO MATRIX

CO	PO												PSO			
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4
1	3	-	3	-	-	-	-	-	-	-	-	-	2	-	-	1
2	-	2	-	-	1	-	-	-	-	2	-	-		-	-	-
3	-	-	3	1	-	-	-	-	-	3	-	2		-	-	-
4	2	-	-	-	-	-	-	-	1	-	-	-	2	-	1	2
5	-	3	1	3	1	-	-	-	1	-	-	2	3	1	-	-
Average	2.5	2.5	2.3	2.0	1.0	-	-	-	1.0	2.5	-	2.0		1.0	1.0	1.5

EVALUATION PROCEDURE FOR EACH EXPERIMENT

S. No	Description	Mark
1.	Aim & Procedure	20
2.	Observation	30
3.	Conduction and Execution	30
4.	Output & Result	10
5.	Viva	10
Total		100

INTERNAL ASSESSMENT FOR LABORATORY

S. No	Description	Mark
1.	Conduction & Execution of Experiment	30
2.	Record	10
3.	Model Test	20
Total		60

Ex. No: 1 INSTALLING DATA ANALYSIS AND VISUALIZATION TOOL

AIM

To write a step to install data analysis and visualization tool: R / Python / Tableau Public / Power BI.

PROCEDURE

1. R:

- Download R:
 - Visit the official R website (<https://cran.r-project.org/>) and download the installer for your operating system (Windows, macOS, or Linux).
- Install R by following the instructions provided in the installer.

2. Python:

- Download Python:
 - Visit the official website (<https://www.python.org/downloads/>) and download the Python installer for your OS (Windows, macOS, or Linux).
- Install Python by running the installer and making sure to check the option to add Python to your system's PATH during installation.

(i) INSTALL NUMPY WITH PIP

NumPy (Numerical Python) is an open-source core Python library for scientific computations. It is a general-purpose array and matrices processing package.

```
pip install numpy
```

(ii) INSTALL JUPYTER LAB

Install Jupyter Lab with pip:

```
pip install jupyterlab
```

Once installed, launch Jupyter Lab with:

```
jupyter-lab
```

(iii) JUPYTER NOTEBOOK

Install the classic Jupyter Notebook with:

```
pip install notebook
```

To run the notebook:

```
Jupyter notebook
```

(iv) INSTALL SCIPY

Scipy is a Python library that is useful in solving many mathematical equations and algorithms. It is designed on the top of Numpy library. SCIPY means Scientific Python.

```
pip install scipy
```

(v) INSTALL PANDAS

Pandas is a Python Package that provides fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive.

```
pip install pandas
```

(vi) INSTALL MATPLOTLIB

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Working with “relational” or “labeled” data both easy and intuitive.

```
pip install matplotlib
```

3. Tableau Public:

- Tableau Public
- 4. It is a web-based tool, so there's no installation required. Simply visit the Tableau Public Website (<https://public.tableau.com/s/gallery>) and create an account to start using it. Power BI:
 - Download Power BI Desktop:
 - Go to the official Power BI website (<https://powerbi.microsoft.com/en-us/desktop/>) and download Power BI Desktop.
 - Install Power BI Desktop by running the installer.

PROGRAM 1

```
import numpy as np
import pandas as pd
hafeez = ['Hafeez', 19]
aslam = ['Aslam', 21]
kareem = ['kareem', 18]
dataframe = pd.DataFrame([hafeez, aslam, kareem], columns = ['Name', 'Age'])
print(dataframe)
```

Output 1

	Name	Age
0	Hafeez	19
1	Aslam	21
2	Kareem	18

PROGRAM 1

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv("CountryData.csv")
plt.hist(data)
plt.xlabel("code")
plt.ylabel("Total_personal_income")
plt.show()
```

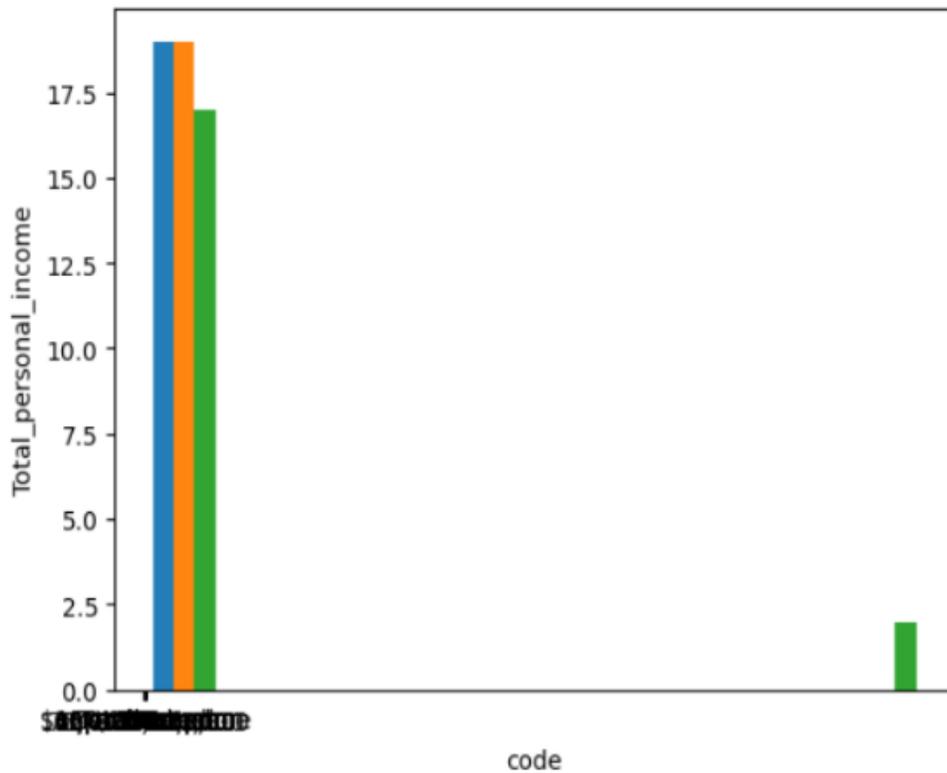
CREATE A CSV FILE IN EXCEL:

- First create a CSV file in excel with attributes 'code' and 'Total_personal_income'.
- Save the file with filename mentioned above "CountryData" with extension as .csv file.

CountryData - Microsoft Excel

Code	Total_per	Census_usually_resident_population_count_aged_15_years_and_over
11	Loss	20625
12	Zero incor	257310
13	\$1755,000	210705
14	\$5,001751	177423
15	\$10,00175	262197
16	\$15,00175	375282
17	\$20,00175	300639
18	\$25,00175	210132
19	\$30,00175	186087
20	\$35,00175	212724
21	\$40,00175	364719
22	\$50,00175	309375
23	\$60,00175	234606
24	\$70,00175	361317
25	\$100,0017	176310
26	\$150,001 c	110910
99	Not stated	0
TotalState	Total state	3776355
Total	Total	3776355

Output 2



RESULT

Thus, the python program to install data analysis and visualization tools like R, Python, Tableau Public, or Power BI, and their features were explored successfully.

Ex. No: 2 EXPLORATORY DATA ANALYSIS (EDA) ON WITH DATASETS

AIM

To perform exploratory data analysis (EDA) on with datasets like email data set.

PROCEDURE

Exploratory Data Analysis (EDA) on email datasets involves importing the data, cleaning it, visualizing it, and extracting insights. Here's a step-by-step guide on how to perform EDA on an email dataset using Python and Pandas

1. Import Necessary Libraries:

Import the required Python libraries for data analysis and visualization.

2. Load Email Data:

Assuming you have a folder containing email files (e.g., .eml files), you can use the email library to parse and extract the email contents.

3. Data Cleaning:

Depending on your dataset, you may need to clean and preprocess the data. Common cleaning steps include handling missing values, converting dates to datetime format, and removing duplicates.

4. Data Exploration:

Now, you can start exploring the dataset using various techniques. Here are some common EDA tasks:

Basic Statistics:

Get summary statistics of the dataset.

Distribution of Dates:

Visualize the distribution of email dates.

5. Word Cloud for Subject or Message:

Create a word cloud to visualize common words in email subjects or messages.

6. Top Senders and Recipients:

Find the top email senders and recipients.

Depending on your dataset, you can explore further, analyze sentiment, perform network analysis, or any other relevant analysis to gain insights from your email data.

PROGRAM

Import necessary libraries

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```

import seaborn as sns
# Load the dataset
df = pd.read_csv('D:\ARCHANA\dxv\LAB\DXV\Emaildataset.csv')
# Display basic information about the dataset
print(df.info())
# Display the first few rows of the dataset
print(df.head())
# Descriptive statistics
print(df.describe())
# Check for missing values
print(df.isnull().sum())
# Visualize the distribution of numerical variables
sns.pairplot(df) plt.show()
# Visualize the distribution of categorical variables
sns.countplot(x='label', data=df) plt.show()
# Correlation matrix for numerical variables
correlation_matrix = df.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.show()
# Word cloud for text data (if you have a column with text data) from wordcloud
import WordCloud
text_data = ''.join(df['text_column'])
wordcloud = WordCloud(width=800, height=400, random_state=21,
max_font_size=110).generate(text_data)
plt.figure(figsize=(10, 7))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis('off')
plt.show()

```

OUTPUT

Data columns (total 4 columns):

```
# Column Non-Null Count Dtype
-----
```

0 Unnamed: 0 5171 non-null int64

1 label 5171 non-null object

2 text 5171 non-null object

3 label_num 5171 non-null int64

dtypes: int64(2), object(2)

memory usage: 161.7+ KB

None

Unnamed: 0 label text label_num

0 605 ham Subject: enron methanol; meter #: 988291\r\n... 0

1 2349 ham Subject: hpl nom for january 9 , 2001\r\n(see... 0

2 3624 ham Subject: neon retreat\r\nho ho ho , we ' re ar... 0

3 4685 spam Subject: photoshop , windows , office . cheap ... 1

4 2030 ham Subject: re : indian springs\r\nthis deal is t... 0

	Unnamed: 0	label_num
count	5171.000000	5171.000000
mean	2585.000000	0.289886
std	1492.883452	0.453753
min	0.000000	0.000000
25%	1292.500000	0.000000
50%	2585.000000	0.000000
75%	3877.500000	1.000000
max	5170.000000	1.000000



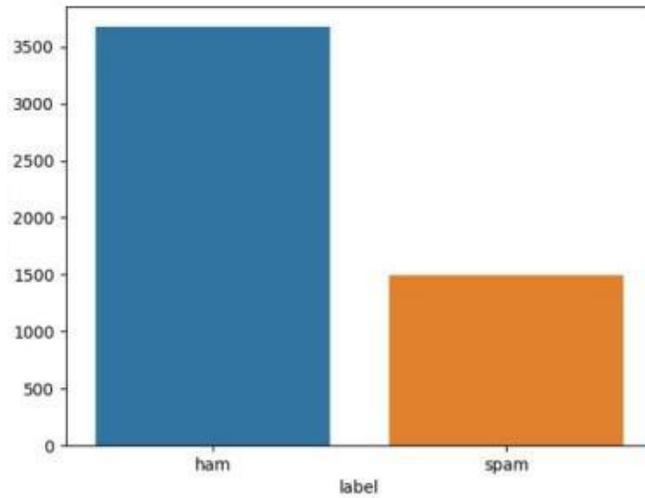
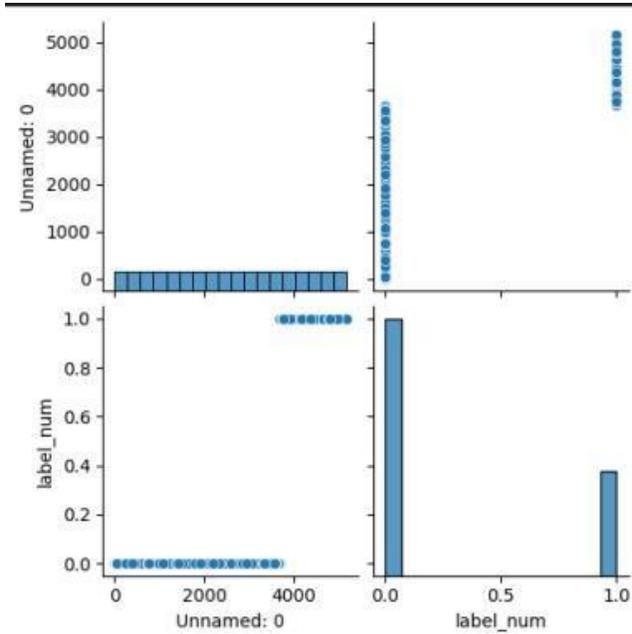
Unnamed: 0 0

label 0

text 0

label_num 0

dtype: int64



RESULT

Thus, the above Performing exploratory data analysis (EDA) on with datasets like email data set has been performed successfully.

**Ex. No:3 WORKING WITH NUMPY ARRAYS, PANDAS DATA FRAMES, BASIC PLOTS
USING MATPLOTLIB**

AIM

To write the steps for Working with Numpy arrays, Pandas data frames, Basic plots using Matplotlib

PROCEDURE

1. NumPy:

NumPy is a fundamental library for numerical computing in Python. It provides support for multi-dimensional arrays and various mathematical functions. To get started, you'll first need to install

pip install numpy

NumPy if you haven't already (you can use pip):

Once NumPy is installed, you can use it as follows:

```
import numpy as np
# Creating NumPy arrays
arr = np.array([1,2,3,4,5])
print(arr)
# Basic operations
mean = np.mean(arr)
sum = np.sum(arr)
# Mathematical functions
square_root = np.sqrt(arr)
exponential = np.exp(arr)
# Indexing and Slicing
first_element = arr[0]
sub_array = arr[1:4]
# Array Operations
Combined_array = np.concatenate([arr, sub_array])
```



OUTPUT

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

```
PS C:\Users\ADMIN\Desktop\python sample> & 'C:\Users\ADMIN\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\ADMIN\.vscode\extensions\ms-python.pytho  
n-2023.14.0\pythonFiles\lib\python\debuggy\adapter\..\..\debuggy\launcher' '58119' '--' 'C:\Users\ADMIN\Desktop\python sample\exno1.py'
```

```
[1 2 3 4 5]
```

```
PS C:\Users\ADMIN\Desktop\python sample>
```

2. Pandas:

Pandas is a powerful library for data manipulation and analysis.

You can install pandas using pip:

```
pip install pandas
```

Here's how to work with Pandas DataFrames:

```
import pandas as pd
```

```
# Creating a DataFrame from a dictionary
```

```
data = { 'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Emily'], 'Age': [25, 30, 35, 28, 22], 'City': ['New York',  
'Los Angeles', 'Chicago', 'Houston', 'Miami']  
}
```

```
df = pd.DataFrame(data)
```

```
# Display the entire DataFrame
```

```
print("DataFrame:")
```

```
print(df)
```

```
# Accessing specific columns
```

```
print("\n Accessing 'Name' Column:")
```

```
print(df['Name'])
```

```
# Adding a new column
```

```
df['Salary'] = [50000, 60000, 75000, 48000, 55000]
```

```
# Filtering data
```

```
print("\nPeople older than 30:")
```

```

print(df[df['Age'] > 30])

# Sorting by a column
print("\nSorting by 'Age' in descending order:")
print(df.sort_values(by='Age', ascending=False))

# Aggregating data
print("\nAverage age:")
print(df['Age'].mean())

# Grouping and aggregation
grouped_data = df.groupby('City')['Salary'].mean()
print("\nAverage salary by city:")
print(grouped_data)

# Applying a function to a column
df['Age_Squared'] = df['Age'].apply(lambda x: x ** 2)

2) # Removing a column
df = df.drop(columns=['Age_Squared'])

# Saving the DataFrame to a CSV file
df.to_csv('output.csv', index=False)

# Reading a CSV file into a DataFrame
new_df = pd.read_csv('output.csv')
print("\nDataFrame from CSV file:")
print(new_df)

```



OUTPUT

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL	PORTS
0	Alice	25	New York	
1	Bob	30	Los Angeles	
2	Charlie	35	Chicago	
3	David	28	Houston	
4	Emily	22	Miami	

```

Accessing 'Name' column:
0    Alice
1     Bob
2  Charlie
3   David
4    Emily
Name: Name, dtype: object

```

People older than 30:

	Name	Age	City	Salary
2	Charlie	35	Chicago	75000

Sorting by 'Age' in descending order:

	Name	Age	City	Salary
2	Charlie	35	Chicago	75000
1	Bob	30	Los Angeles	60000
3	David	28	Houston	48000
0	Alice	25	New York	50000
4	Emily	22	Miami	55000

Average age:

28.0

Average salary by city:

City

Chicago 75000.0

Houston 48000.0

Los Angeles 60000.0

Miami 55000.0

New York 50000.0

Name: Salary, dtype: float64

DataFrame from CSV file:

	Name	Age	City	Salary
0	Alice	25	New York	50000
1	Bob	30	Los Angeles	60000
2	Charlie	35	Chicago	75000
3	David	28	Houston	48000
4	Emily	22	Miami	55000

PS D:\ARCHANA\dxv\LAB>



3. Matplotlib:

Matplotlib is a popular library for creating static, animated, or interactive plots and graphs.

Install Matplotlib using pip:

```
pip install matplotlib
```

Here's a simple example of creating a basic plot:

```
import matplotlib.pyplot as plt
```

```
# Sample data
```

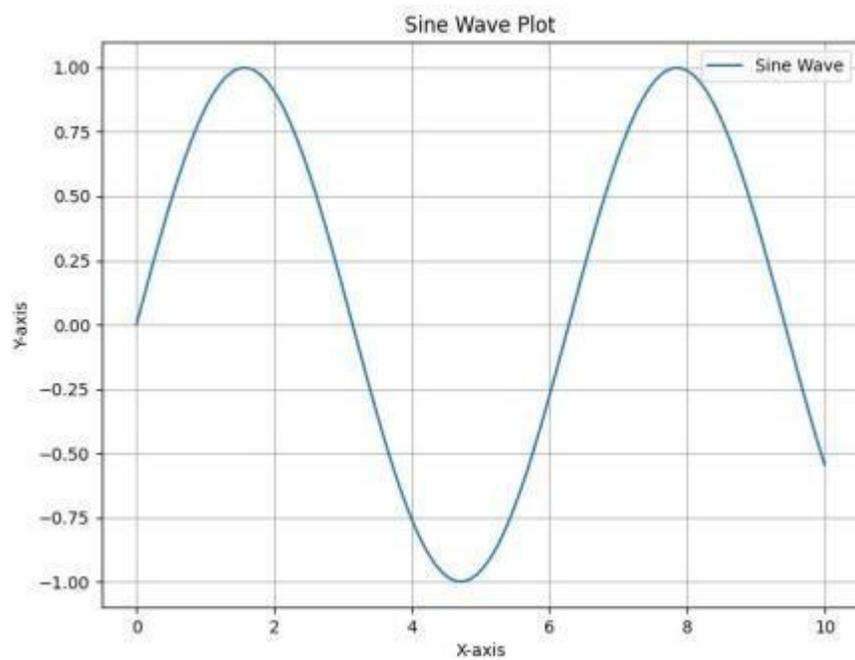
```
x = np.linspace(0, 10, 100)
```

```
y = np.sin(x)
```

```
# Create a line plot
```

```
plt.figure(figsize=(8, 6))
plt.plot(x, y, label='Sine
Wave') plt.title('Sine Wave
Plot') plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.legend()
plt.grid(True)
plt.show()
```

OUTPUT



RESULT

Thus, the above working with numpy, pandas, and matplotlib has been completed successfully.

Ex. No: 4 EXPLORING VARIOUS VARIABLE AND ROW FILTERS IN R FOR CLEANING DATA

AIM

To exploring various variable and row filters in R for cleaning data.

PROCEDURE

Data Preparation and Cleaning

First, let's create a sample dataset and then explore various variable and row filters to clean the data.

Create a sample dataset

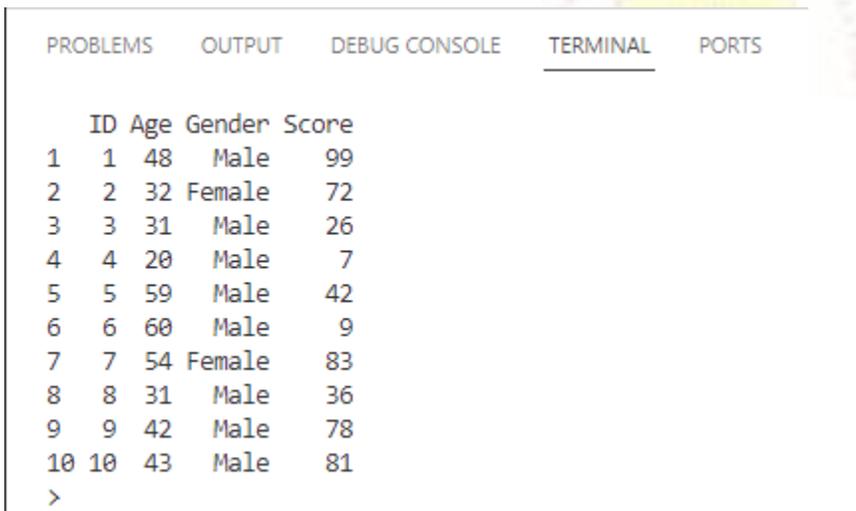
```
set.seed(123)
```

```
data <- data.frame( ID = 1:10, Age = sample(18:60, 10, replace = TRUE), Gender = sample(c("Male",  
"Female"), 10, replace = TRUE), Score = sample(1:100, 10) )
```

Print the sample data

```
print(data)
```

OUTPUT



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  
  
  ID Age Gender Score  
1   1  48  Male   99  
2   2  32 Female   72  
3   3  31  Male   26  
4   4  20  Male    7  
5   5  59  Male   42  
6   6  60  Male    9  
7   7  54 Female   83  
8   8  31  Male   36  
9   9  42  Male   78  
10  10  43  Male   81  
>
```

Variable Filters

1. Filtering by s Specific Value:

To filter rows based on a specific value in a variable (e.g., only show rows where Age is greater than 30): `filtered_data <- data [data$Age>30,]`

2. Filtering by Multiple Conditions:

You can filter row based on multiple conditions using the & (AND) or | (OR) operators (e.g., show rows

where Age is greater than 30 and Gender is "Male"): filtered_data <- data [data\$Age > 30 & data\$Gender == "Male",]

Row Filters

1. Removing Duplicate Rows:

To remove duplicate rows based on certain columns (e.g., remove duplicates based on 'ID'):

```
cleaned_data <- unique (data [, c("ID", "Age", "Gender")])
```

2. Removing Rows with Missing Values:

To remove rows with missing values (NA):

```
cleaned_data <- na.omit (data)
```

Data Visualization

Apply various plot features using the ggplot2 package to visualize the cleaned data.

Load the ggplot2 package library (ggplot2)

Create a scatterplot of Age vs. Score with points colored by Gender

```
Ggplot (data = cleaned_data, aes(x = Age, y = Score, color = Gender)) + geom_point () + labs(title = "Scatterplot of Age vs. Score", x = "Age", y = "Score")
```

Create a histogram of Age

```
Ggplot (data = cleaned_data, aes(x = Age)) + geom_histogram (binwidth = 5, fill = "blue", alpha = 0.5) + labs (title = "Histogram of Age", x = "Age", y = "Frequency")
```

Create a bar chart of Gender distribution

```
Ggplot (data = cleaned_data, aes(x = Gender)) + geom_bar (fill = "green", alpha = 0.7) + labs (title = "Gender Distribution", x = "Gender", y = "Count")
```

RESULT

Thus, the above exploring various variable and row filters in R for cleaning data was successfully completed.

Ex. No: 5 TIME SERIES ANALYSIS USING VARIOUS VISUALIZATION TECHNIQUES

AIM

To perform time series analysis and apply the various visualization techniques.

PROCEDURE

DOWNLOAD DATASET

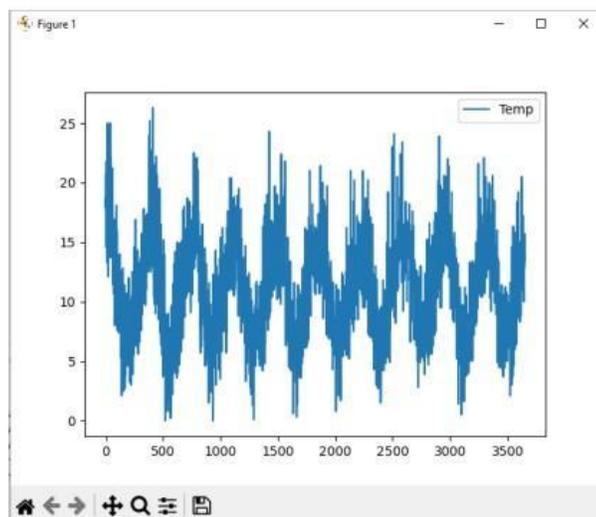
Step 1: Open google and type the following path in the address bar and download a dataset.

<http://github.com/jbrownlee/Datasets>.

Step 2: Write the following code to get the details.

```
from pandas import read_csv
from matplotlib import pyplot
series = read_csv ('pathname')
print (series.head ( ))
series.plot ( )
pyplot.show ( )
```

OUTPUT

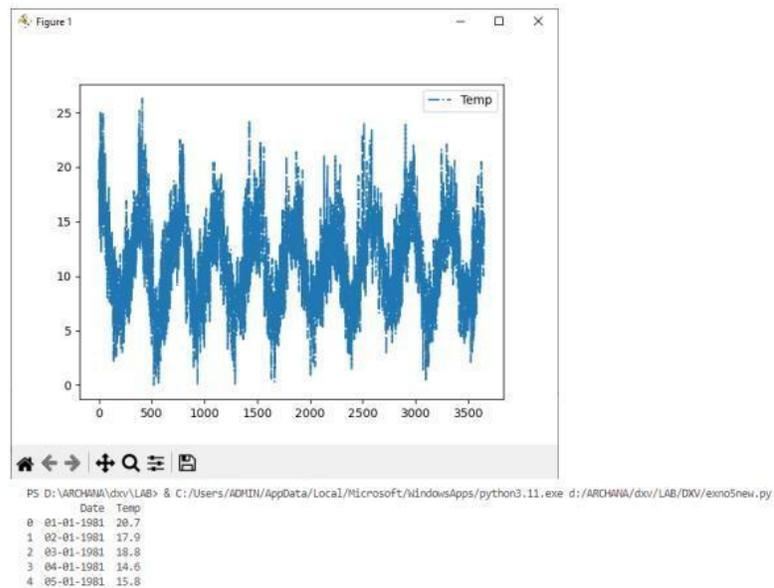


```
PS D:\ARCHANA\dxv\LAB> & C:/Users/ADITHI/AppData/Local/Microsoft/WindowsApps/python3.11.exe d:/ARCHANA/dxv/LAB/DXV/exno5nev.py
Date Temp
0 01-01-1981 28.7
1 02-01-1981 17.9
2 03-01-1981 18.8
3 04-01-1981 14.6
4 05-01-1981 15.8
```

Step 3: To get the time series line plot:

```
series.plot (style='-'.')  
pyplot.show ()
```

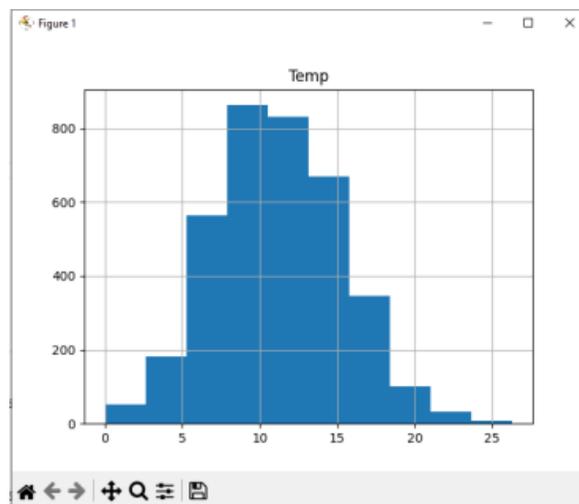
OUTPUT



Step 4: To create a Histogram:

```
series.hist ()  
pyplot.show ()
```

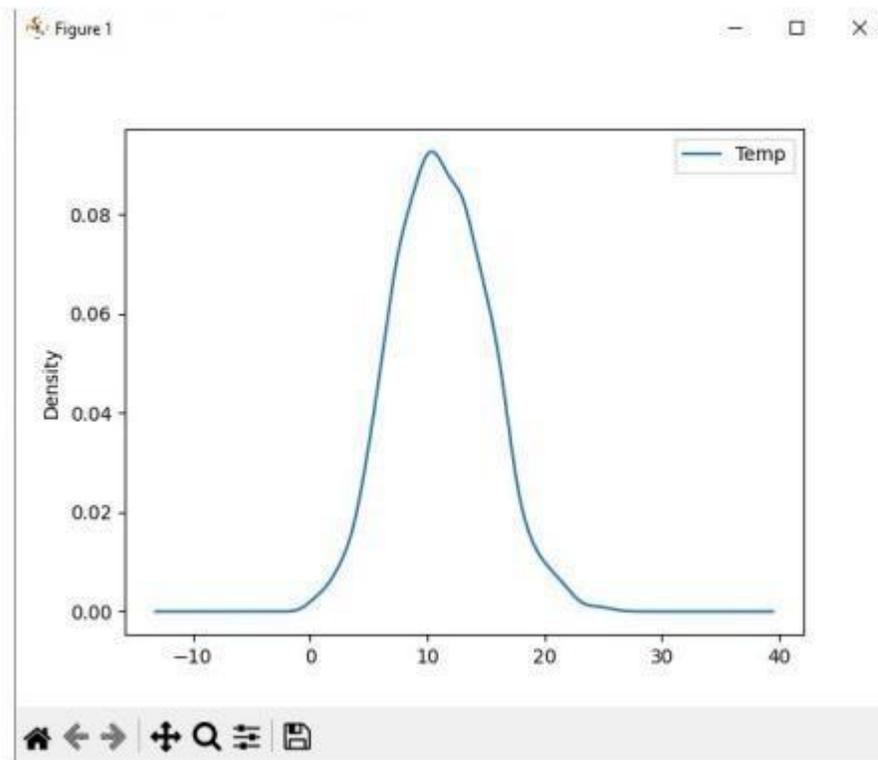
OUTPUT



Step 5: To create density plot:

```
series.plot(kind = 'kde')  
pyplot.show ( )
```

OUTPUT



RESULT

Thus, the above time analysis has been checked with various visualization techniques.

Ex. No: 6

DATA ANALYSIS AND REPRESENTATION ON MAP

AIM

Write a program to perform data analysis and representation on a map using various map data sets with mouse rollover effect, user interaction etc.

PROCEDURE STEP

1:

- Make sure to install the necessary libraries.

```
pip install geopandas folium bokeh
```

PROGRAM

```
from bokeh.io import show
from bokeh.models import ColumnDataSource, HoverTool
from bokeh.plotting import figure
from bokeh.layouts import column
import pandas as pd
import folium

# Load your data
data = pd.read_csv('D:\ARCHANA\dxv\LAB\DXV\geographic.csv')

# Create a Bokeh figure
p = figure(width=800, height=400, tools='pan,wheel_zoom,reset')

# Create a ColumnDataSource to hold data
source = ColumnDataSource(data)

# Add circle markers to the figure
p.circle(x='Longitude', y='Latitude', size=10, source=source, color='orange')

# Create a hover tool for mouse rollover effect
hover = HoverTool() hover.tooltips = [("Info", "@Info"), ("Latitude", "@Latitude"), ("Longitude", "@Longitude")]
p.add_tools(hover)
```

Display the Bokeh plot

```
layout = column(p)show(layout)
```

Create a map centered at a specific location

```
m = folium.Map(location=[latitude, longitude], zoom_start=10)
```

Add markers for your data points

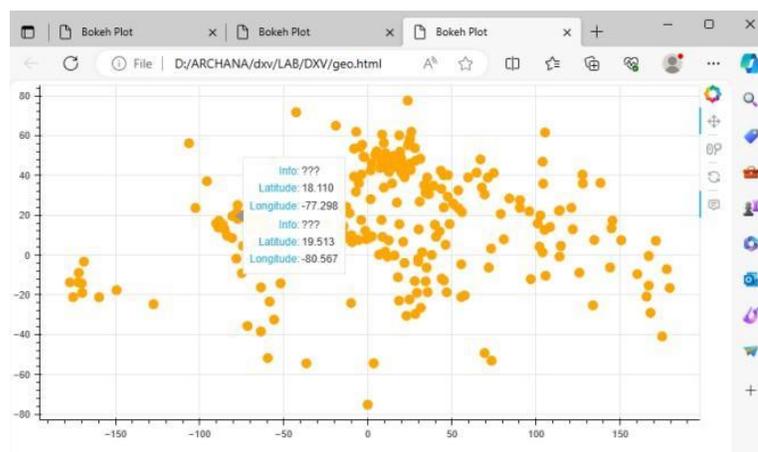
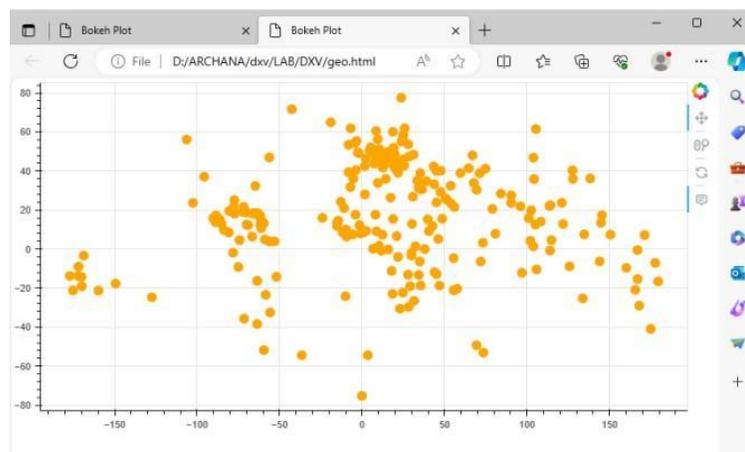
```
for index, row in data.iterrows():
```

```
    folium.Marker( location=[row['Latitude'], row['Longitude']], popup=row['Info'], # Display additional  
    info on mouse click ).add_to(m)
```

Save the map to an HTML file

```
m.save('map.html')
```

OUTPUT



RESULT

Thus, the data analysis and representation on a map using various map data sets with mouse rollover effect, use interaction has been completed successfully.

Ex. No: 7 BUILDING CARTOGRAPHIC VISUALIZATION

AIM

Build cartographic visualization for multiple datasets involving various countries of the world; states and districts in India etc.

PROCEDURE

STEP

1:

Collect Datasets

Gather the datasets containing geographical information for countries, states, or districts. Make sure these datasets include the necessary attributes for mapping (e.g., country/state/district names, codes, and relevant data).

STEP 2:

Install Required Libraries:

```
pip install geopandas matplotlib
```

STEP 3:

Load Geographic Data:

Use Geopandas to load the geographic data for countries, states, or districts. Make sure to match the geographical data with your datasets based on the common attributes.

STEP 4:

Merge Datasets:

Merge your datasets with the geographic data based on common attributes. This step is crucial for linking your data to the corresponding geographic regions.

STEP 5:

Create Cartographic Visualizations:

Use Matplotlib to create cartographic visualizations. You can create separate plots for different datasets or overlay them on a single map.

STEP 6:

Customize and Enhance:

Customize your visualizations based on your needs. You can add legends, labels, titles, and other elements to enhance the interpretability of your maps.

STEP 7:

Save and Share:

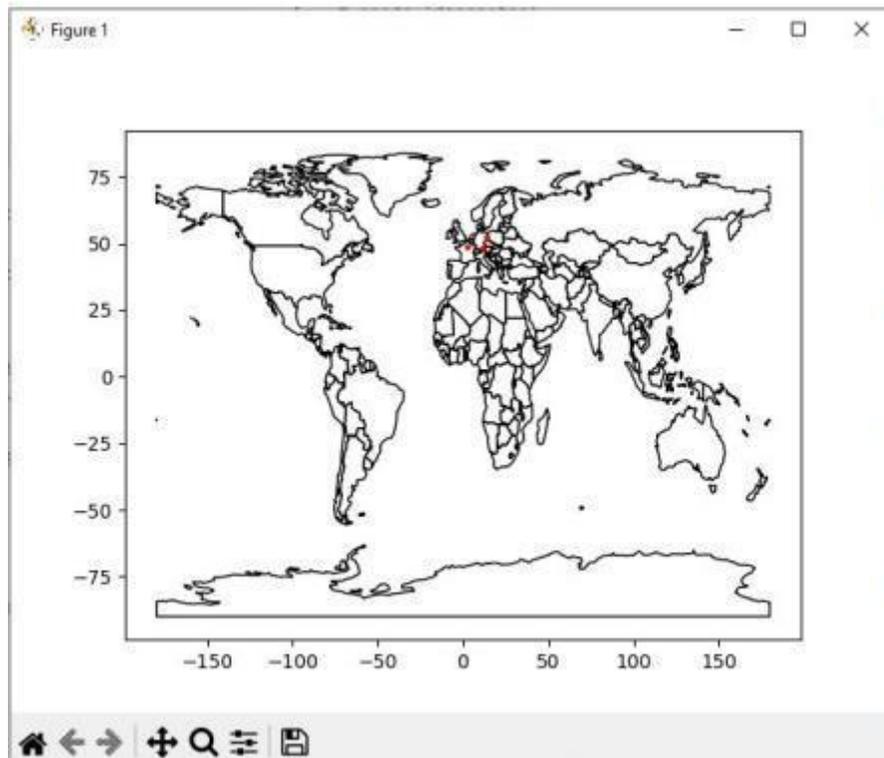
Save your visualizations as image files or interactive plots if needed. You can then share these visualizations with others.

PROGRAM:

```
import pandas as pd
import geopandas as gpd
import shapely
# needs 'descartes'
import matplotlib.pyplot as plt
df = pd.DataFrame({'city': ['Berlin', 'Paris', 'Munich'], 'latitude': [52.518611111111, 48.856666666667,
48.137222222222], 'longitude': [13.408333333333, 2.3516666666667, 11.575555555556]})
gdf = gpd.GeoDataFrame(df.drop(['latitude', 'longitude'], axis=1), crs={'init': 'epsg:4326'},
geometry=[shapely.geometry.Point(xy) for xy in zip(df.longitude, df.latitude)])
print (gdf)
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
base = world.plot(color='white', edgecolor='black')
gdf.plot (ax=base, marker='o', color='red', markersize=5)
plt.show ( )
```

OUTPUT

```
city          geometry
0  Berlin POINT (13.40833 52.51861)
1    Paris POINT (2.35167 48.85667)
2  Munich POINT (11.57556 48.13722)
```



RESULT

Build cartographic visualization for multiple datasets involving various countries of the world; has been visualized successfully.

Ex. No: 8 PERFORM EDA ON WINE QUALITY DATA SET.

AIM

To write a program to perform EDA on Wine Quality Data Set.

PROGRAM

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

# Load the dataset
data = pd.read_csv("pathname")

# Display the first few rows of the dataset
print(data.head())

# Get information about the dataset
print(data.info())

# Summary statistics
print (data.describe())

# Distribution of wine quality
sns.countplot (data ['quality'])
plt.title (" Wine Quality data set")
plt.show ( )

# Box plots for selected features by wine quality
features = ['alcohol', 'volatile acidity', 'citric acid', 'residual sugar']
for feature in features:
    plt.figure (figsize=(8, 6))
    sns.boxplot(x='quality', y=feature, data=data)
    plt.title (f'{feature} by Wine Quality')
    plt.show ( )

# Pair plot of selected features
sns.pairplot (data, vars= ['alcohol', 'volatile acidity', 'citric acid', 'residual sugar'], hue='quality', diag_kind='kde')
plt.suptitle ("Pair Plot of Selected Features")
plt.show ( )
```



Correlation heatmap

```
corr_matrix = data.corr ( )  
plt.figure (figsize = (10, 8))  
sns.heatmap (corr_matrix, annot=True, cmap="coolwarm", fmt=".2f")  
plt.title ("Correlation Heatmap")  
plt.show ( )
```

Histograms of selected features

```
features = ['alcohol', 'volatile acidity', 'citric acid', 'residual sugar']
```

for feature in features:

```
plt.figure (figsize = (6, 4))  
sns.histplot (data [feature], kde=True, bins=20)  
plt.title (f"Distribution of {feature}")  
plt.show ( )
```

OUTPUT



```
P5 D:\ARCHANA\dev\LAB> & C:\Users\ADMIN\AppData\Local\Microsoft\WindowsApps\python3.11.exe d:\ARCHANA\dev\LAB\DEV\exo6.py  
type alcohol volatile acidity citric acid residual sugar chlorides free sulfur dioxide total sulfur dioxide density pH sulphates quality fixed acidity  
0 white 8.8 0.27 0.36 20.7 0.045 45.0 170.0 1.0010 3.00 0.45 6 7.0  
1 white 9.5 0.30 0.34 1.6 0.040 14.0 132.0 0.9940 3.30 0.40 6 6.3  
2 white 10.1 0.28 0.40 6.9 0.050 30.0 97.0 0.9951 3.20 0.44 6 8.1  
3 white 9.9 0.23 0.32 8.5 0.050 47.0 180.0 0.9950 3.19 0.40 6 7.2  
4 white 9.9 0.23 0.32 8.5 0.050 47.0 180.0 0.9950 3.19 0.40 6 7.2  
<<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 6497 entries, 0 to 6496  
Data columns (total 13 columns):  
# Column Non-Null Count Dtype  
---  
0 type 6497 non-null object  
1 alcohol 6497 non-null float64  
2 volatile acidity 6489 non-null float64  
3 citric acid 6494 non-null float64  
4 residual sugar 6495 non-null float64  
5 chlorides 6495 non-null float64  
6 free sulfur dioxide 6497 non-null float64  
7 total sulfur dioxide 6497 non-null float64  
8 density 6497 non-null float64  
9 pH 6488 non-null float64  
10 sulphates 6493 non-null float64  
11 quality 6497 non-null int64  
12 fixed acidity 6487 non-null float64  
dtypes: float64(11), int64(1), object(1)  
memory usage: 600.0+ KB  
None  
alcohol volatile acidity citric acid residual sugar chlorides free sulfur dioxide total sulfur dioxide density pH sulphates quality fixed acidity  
count 6497.000000 6489.000000 6494.000000 6495.000000 6495.000000 6497.000000 6497.000000 6497.000000 6497.000000 6488.000000 6493.000000 6497.000000 6487.000000  
mean 10.401091 0.330691 0.318722 5.444326 0.056042 30.525319 115.744574 0.994697 3.218395 0.531215 5.818370 7.216579  
std 1.102712 0.164640 0.145265 4.758125 0.035036 17.740400 56.521855 0.002900 0.160748 0.148814 0.873255 1.296750  
min 8.000000 0.000000 0.000000 0.600000 0.000000 1.000000 6.000000 0.967118 2.720000 0.220000 3.000000 3.800000  
25% 9.500000 0.230000 0.250000 1.000000 0.030000 17.000000 77.000000 0.992340 3.110000 0.430000 5.000000 6.400000  
50% 10.300000 0.250000 0.310000 3.000000 0.047000 29.000000 118.000000 0.994000 3.210000 0.510000 6.000000 7.000000  
75% 11.300000 0.400000 0.390000 8.100000 0.055000 41.000000 156.000000 0.996990 3.320000 0.600000 6.000000 7.700000  
max 14.000000 1.500000 1.660000 65.000000 0.611000 209.000000 440.000000 1.030000 4.010000 2.000000 9.000000 15.000000
```

Figure 1

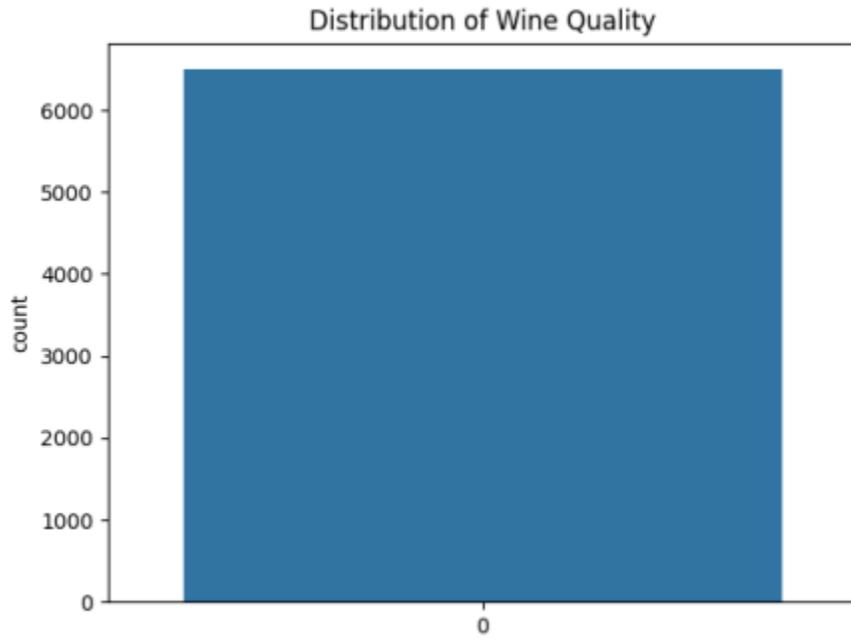
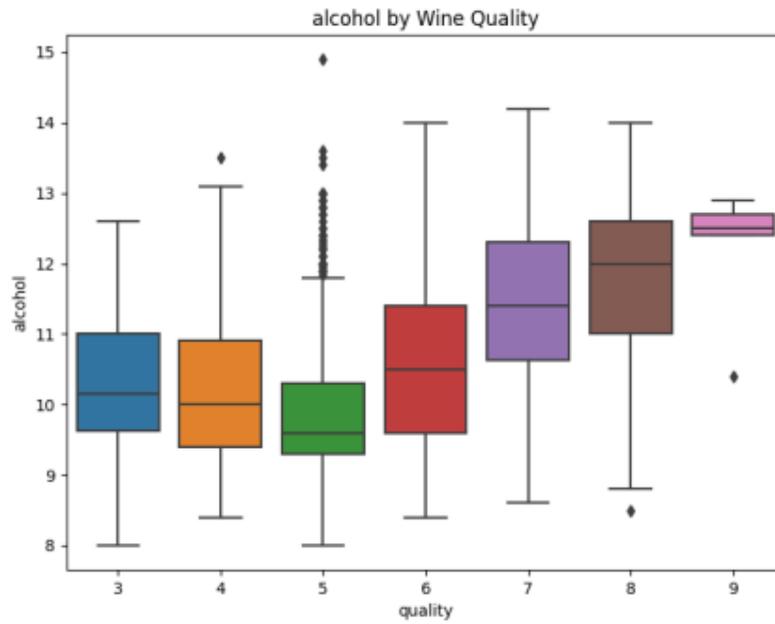
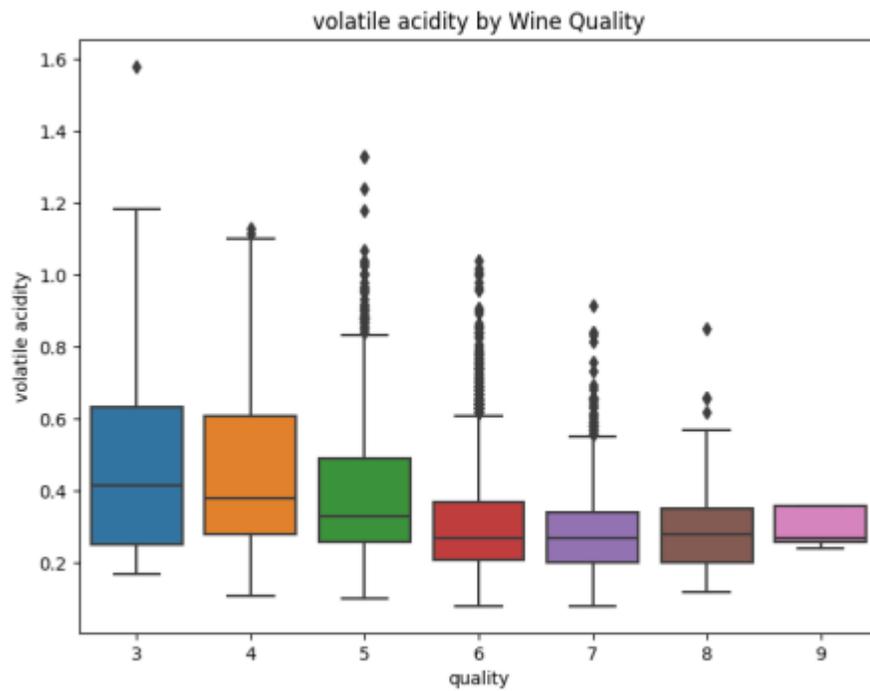


Figure 1

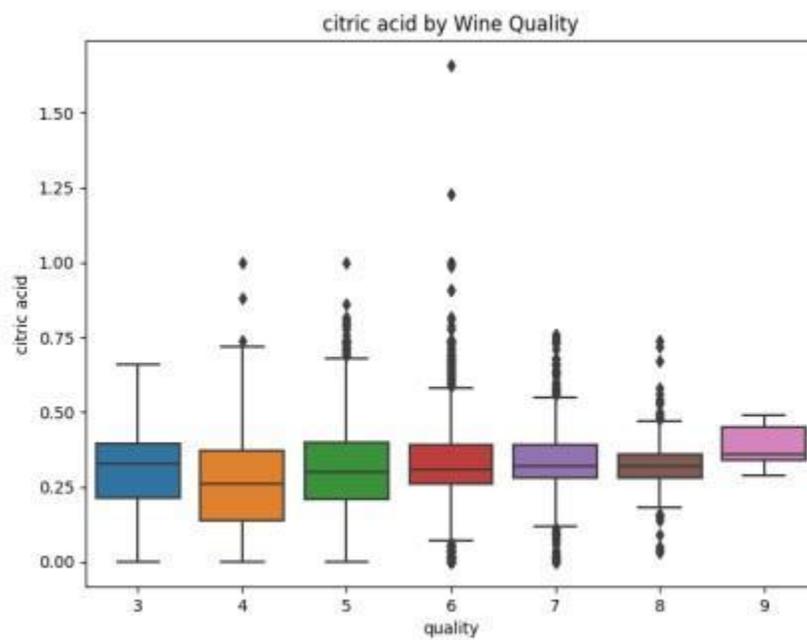




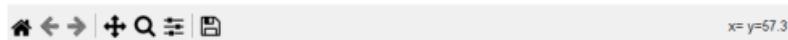
Navigation icons: Home, Left Arrow, Right Arrow, Zoom In, Zoom Out, Full Screen, Save. x= y=1.144

Figure 1

[-] [□] [X]



Navigation icons: Home, Left Arrow, Right Arrow, Zoom In, Zoom Out, Full Screen, Save. x= y=1.209



RESULT

Thus the above program to perform EDA on Wine Quality Data Set.

Ex. No: 9 VISUALIZING VARIOUS EDA TECHNIQUES AS CASE STUDY FOR IRIS DATASET

AIM

The Mini Project to predict the time taken to solve a problem given the current status of the user using Random Forest Regressor Model.

PROCEDURE

Import Libraries:

Start by importing the necessary libraries and loading the dataset.

Descriptive Statistics:

Compute and display descriptive statistics.

python

Check for Missing Values:

Verify if there are any missing values in the dataset.

Visualize Data Distributions:

Visualize the distribution of numerical variables.

python

Correlation Heatmap:

Examine the correlation between numerical variables.

Boxplots for Categorical Variables:

Use boxplots to visualize the distribution of features by species.

Violin Plots:

Combine box plots with kernel density estimation for better visualization.

Correlation between Features:

Visualize pair-wise feature correlations.

Conclusion and Summary:

Summarize key findings and insights from the analysis.

This case study provides a comprehensive analysis of the Iris dataset, including data exploration, descriptive statistics, and visualization of data distributions, correlation analysis, and feature-specific visualizations