

## EX : 1 - Setting up the Python environment and libraries Jupyter Notebook

- Create a new notebook for Python

```
!pip install ipywidgets --quiet
!jupyter nbextension enable
import ipywidgets as widgets
from IPython.display import display
```

- Write and execute Python code

```
name = "Sanjai"
print(f"Hello, {name}! Welcome to Jupyter Notebook.")
```

- Create new cells for code and Markdown

This is a **Markdown** cell.

You can use it to:

- Add **headings**
- Format **text**
- Insert images, tables, and equations

Example of a math equation:

$E = mc^2$

- Demonstrate the application of Jupyter Widgets, Jupyter AI

```
slider = widgets.IntSlider(
    value=5,
    min=0,
    max=10,
    step=1,
    description='Number:',
    continuous_update=False)
output = widgets.Output()
def update_slider(change):
    with output:
        output.clear_output()
        print(f"Slider value: {change['new']}")
slider.observe(update_slider, names='value')
display(slider, output)
```

## **EX : 2 - EDA-Data Import and Export**

- Importing data from CSV, Excel, SQL databases, and web scraping

### **CSV FILE:**

```
import pandas as pd
df_csv = pd.read_csv('/content/survey_lung_cancer.csv')
print(df_csv.head())
df_csv.info()
```

### **EXCEL FILE:**

```
import pandas as pd
df = pd.read_excel("/content/student_marks.xlsx")
df.head()
```

### **SQL DATABASE:**

```
import sqlite3
import pandas as pd
conn = sqlite3.connect('/content/Chinook_Sqlite.sqlite')
query = "SELECT * FROM Album"
df_sql = pd.read_sql_query(query, conn)
df_sql.head()
```

### **WEB SCRAPING:**

```
import pandas as pd
import requests
url = "https://www.worldometers.info/world-population/population-by-country/"
headers = {
    "User-Agent": "Mozilla/5.0"
}
response = requests.get(url, headers=headers)
df = pd.read_html(response.text)[0]
df.head()
```

- Handling different data formats

```
df_json = pd.read_json("sample_data.json") # replace with your file path
print("JSON Data:")
display(df_json.head())
```

- Export a DataFrame to an Excel file.

```
df_csv.to_excel("exported_data.xlsx", index=False)
print("Data exported successfully to 'exported_data.xlsx'")
```

### **EX : 3 - EDA-Data Cleaning**

- Handling missing values: detection, filling, and dropping

```
import pandas as pd
df_csv = pd.read_csv('/content/survey_lung_cancer.csv')
print(df_csv.head())
df_csv.info()
df_csv.isnull().sum()
df_csv.info()
```

- Removing duplicates and unnecessary data

```
import pandas as pd
df_csv = pd.read_csv('/content/survey_lung_cancer.csv')
print(df_csv.head())
df_csv.info()

duplicates=df_csv.duplicated().sum()
print(duplicates)

df_csv.drop_duplicates(inplace=True)
df_csv.isnull().sum()
df_csv.info()
print(df_csv.isnull().sum())
```

- Data type conversion and ensuring consistency

```
import pandas as pd
df_csv = pd.read_csv('/content/survey_lung_cancer.csv')
print(df_csv.head())
df_csv.info()

df_csv['GENDER'] =
df_csv['GENDER'].str.strip().str.upper().astype('category')
df_csv['LUNG_CANCER'] =
df_csv['LUNG_CANCER'].str.strip().str.upper().astype('category')
print(df_csv.dtypes)
print(df_csv.describe())
```

## **EX : 4 - EDA-Data Inspection and Analysis**

- Viewing and inspecting DataFrames

```
import pandas as pd
df_csv = pd.read_csv('/content/survey_lung_cancer.csv')
print(df_csv.head())

df_csv.info()
df_csv.tail()
df_csv.head()
df_csv.describe()
```

- Filtering and subsetting data using conditions

```
import pandas as pd
df_csv = pd.read_csv('/content/survey_lung_cancer.csv')
print(df_csv.head())

subset = df_csv[(df_csv['GENDER'] == 'M') & (df_csv['AGE'] > 50) &
(df_csv['LUNG_CANCER'] == 'YES')]
print(subset)
```

- Descriptive statistics: measures of central tendency (mean, median, mode) and measures of dispersion (range, variance, standard deviation)

```
import pandas as pd
df_csv = pd.read_csv('/content/survey_lung_cancer.csv')
print(df_csv.head())

print("Mean age:", df_csv['AGE'].mean())
print("Median age:", df_csv['AGE'].median())
print("Mode age:", df_csv['AGE'].mode()[0])
print("Minimum age:", df_csv['AGE'].min())
print("Maximum age:", df_csv['AGE'].max())

age_range = df_csv['AGE'].max() - df_csv['AGE'].min()
print("Age range:", age_range)

age_variance = df_csv['AGE'].var()
print("Age variance:", age_variance)

age_std = df_csv['AGE'].std()
print("Age standard deviation:", age_std)
```

### **EX-5 EDA-Data Visualization with Matplotlib**

Basic plotting: line charts, bar charts, histograms

```
import matplotlib.pyplot as plt

import numpy as np

# For better style
plt.style.use('seaborn-v0_8')

x = np.arange(1, 11)

y = np.random.randint(10, 50, size=10)
```

231501183

```
plt.figure(figsize=(8,5))

plt.plot(x, y, marker='o', color='b', label='Random Data')

plt.title("Line Chart Example")

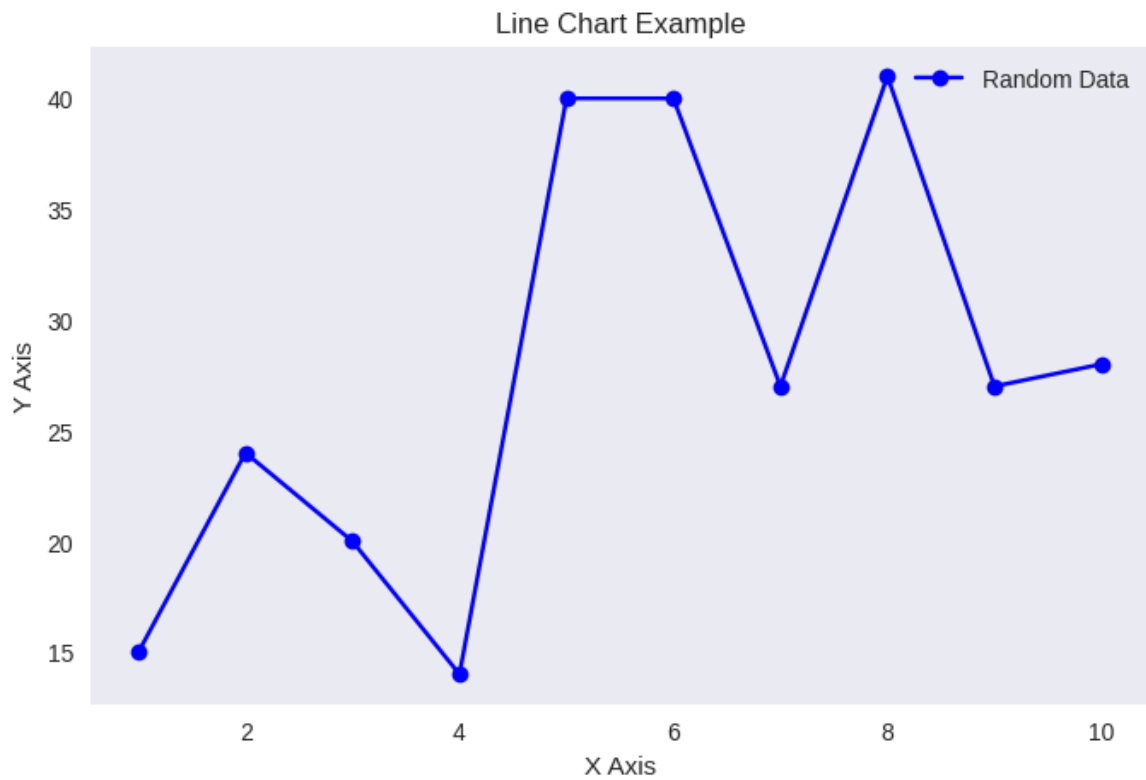
plt.xlabel("X Axis")

plt.ylabel("Y Axis")

plt.legend()

plt.grid(False)

plt.show()
```



```
categories = ['A', 'B', 'C', 'D', 'E']

values = [23, 45, 56, 78, 12]

plt.figure(figsize=(8,5))

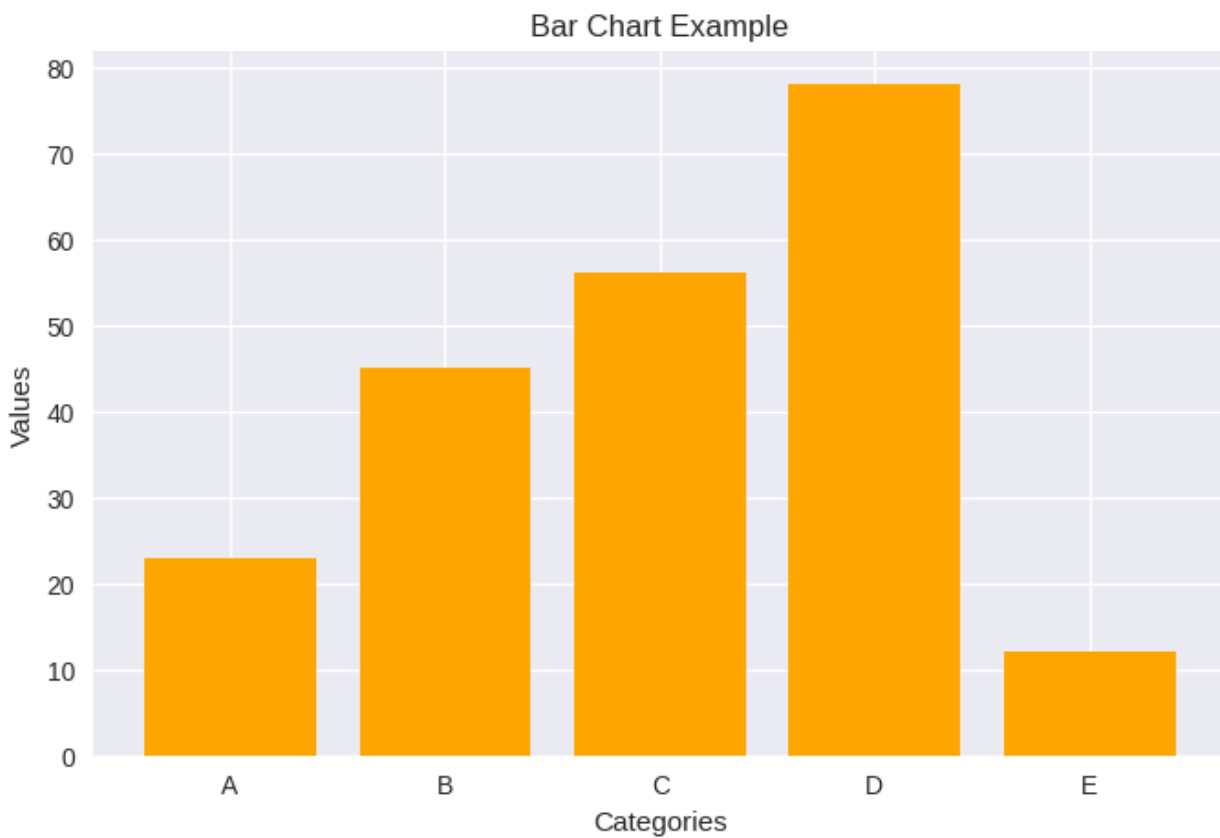
plt.bar(categories, values, color='orange')

plt.title("Bar Chart Example")

plt.xlabel("Categories")

plt.ylabel("Values")

plt.show()
```



---

```
data = np.random.randn(1000) # 1000 normally distributed values
```

```
plt.figure(figsize=(8,5))

plt.hist(data, bins=20, color='green', edgecolor='black')

plt.title("Histogram Example")

plt.xlabel("Value")

plt.ylabel("Frequency")

plt.show()
```

