EXPERIMENT-1

```
# Simple Python code
a = 10
b = 5
print("Sum is:", a + b)
```

```
Sum is: 15
```

```
### This is a heading
This is **bold**, *italic*, and a list:
- Point 1
- Point 2
```

This is a heading

This is **bold**, *italic*, and a list:

- Point 1
- Point 2

```
import ipywidgets as widgets
from IPython.display import display

slider = widgets.IntSlider(value=50, min=0, max=100)
display(slider)
```

EXPERIMENT-2

```
import pandas as pd
import sqlite3
import requests
from sqlalchemy import create_engine
import os
from io import StringIO
# Load CSV
df_csv = pd.read_csv(r"C:\Users\REC\Downloads\college_student_placement_dataset.csv")
print("📄 CSV Data:\n", df_csv)
```

```
📄 CSV Data:
     College_ID   IQ  Prev_Sem_Result  CGPA  Academic_Performance  \
0       CLG0030  107             6.61  6.28                     8
1       CLG0061   97             5.52  5.37                     8
2       CLG0036  109             5.36  5.83                     9
3       CLG0055  122             5.47  5.75                     6
4       CLG0004   96             7.91  7.69                     7
...         ...  ...              ...   ...                   ...
9995    CLG0021  119             8.41  8.29                     4
9996    CLG0098   70             9.25  9.34                     7
9997    CLG0066   89             6.08  6.25                     3
9998    CLG0045  107             8.77  8.92                     3
9999    CLG0060  109             9.41  9.77                     8

     Internship_Experience  Extra_Curricular_Score  Communication_Skills  \
0                       No                       8                     8
1                       No                       7                     8
2                       No                       3                     1
3                      Yes                       1                     6
4                       No                       8                    10
...                    ...                     ...                   ...
9995                    No                       1                     8
9996                    No                       0                     7
9997                   Yes                       3                     9
9998                    No                       7                     5
9999                    No                       3                     5

      Projects_Completed Placement
0                      4        No
1                      0        No
2                      1        No
3                      1        No
4                      2        No
...                  ...       ...
9995                   0       Yes
9996                   2        No
9997                   5        No
9998                   1        No
9999                   5        No

[10000 rows x 10 columns]
```

Create and Save Excel

df_csv.to_excel("data.xlsx", index=False)

Load Excel

df_excel = pd.read_excel("data.xlsx")

print("\n 📘 Excel Data:\n", df_excel)

```
📘 Excel Data:
     College_ID   IQ  Prev_Sem_Result  CGPA  Academic_Performance  \
0       CLG0030  107             6.61  6.28                     8
1       CLG0061   97             5.52  5.37                     8
2       CLG0036  109             5.36  5.83                     9
3       CLG0055  122             5.47  5.75                     6
4       CLG0004   96             7.91  7.69                     7
...         ...  ...              ...   ...                   ...
9995    CLG0021  119             8.41  8.29                     4
9996    CLG0098   70             9.25  9.34                     7
9997    CLG0066   89             6.08  6.25                     3
9998    CLG0045  107             8.77  8.92                     3
9999    CLG0060  109             9.41  9.77                     8

     Internship_Experience  Extra_Curricular_Score  Communication_Skills  \
0                       No                       8                     8
1                       No                       7                     8
2                       No                       3                     1
3                      Yes                       1                     6
4                       No                       8                    10
...                    ...                     ...                   ...
9995                    No                       1                     8
9996                    No                       0                     7
9997                   Yes                       3                     9
9998                    No                       7                     5
9999                    No                       3                     5

      Projects_Completed Placement
0                      4        No
1                      0        No
2                      1        No
3                      1        No
4                      2        No
...                  ...       ...
9995                   0       Yes
9996                   2        No
9997                   5        No
9998                   1        No
9999                   5        No

[10000 rows x 10 columns]
```

```
# Read from SQL
engine = create_engine('sqlite:///example.db')
df_sql = pd.read_sql("SELECT * FROM students", con=engine)
print("\n SQL Data:\n", df_sql)
```

```
 SQL Data:
    id     name  marks
0   1    Alice     88
1   2      Bob     92
2   3  Charlie     78
3   4    Alice     88
4   5      Bob     92
5   6  Charlie     78
6   7    Alice     88
7   8      Bob     92
8   9  Charlie     78
```

```
url = "https://www.worldometers.info/world-population/population-by-country/"
headers = {
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)"
}

# Step 2: Download HTML
response = requests.get(url, headers=headers)
html = response.text

# Step 3: Parse tables using pandas
tables = pd.read_html(html)

# Step 4: Use the first table (main data)
df_web = tables[0]
print("\n Web Scraped Data (Top 5 Rows):")
print(df_web.head())
```

```
🌎 Web Scraped Data (Top 5 Rows):
   # Country (or dependency)  Population 2025 Yearly Change  Net Change  \
0  1                   India       1463865525         0.89%    12929734
1  2                   China       1416096094        -0.23%  -3,225,184
2  3           United States        347275807         0.54%     1849236
3  4               Indonesia        285721236         0.79%     2233305
4  5                Pakistan        255219554         1.57%     3950390

   Density (P/Km²)  Land Area (Km²) Migrants (net)  Fert. Rate  Median Age  \
0              492         2973190       -495,753        1.94        28.8
1              151         9388211       -268,126        1.02        40.1
2               38         9147420        1230663        1.62        38.5
3              158         1811570        -39,509        2.10        30.4
4              331          770880     -1,235,336        3.50        20.6

  Urban Pop % World Share
0       37.1%      17.78%
1       67.5%      17.20%
2       82.8%       4.22%
3       59.6%       3.47%
4       34.4%       3.10%
```

```
data_dict = {'Name': ['David', 'Eve'], 'Age': [22, 24], 'Dept': ['Design', 'QA']}
df_dict = pd.DataFrame(data_dict)
print("\n📦 Dictionary Data:\n", df_dict)
```

```
📦 Dictionary Data:
    Name  Age    Dept
0  David   22  Design
1    Eve   24      QA
```

```
json_data = [
    {"Name": "Avi", "Age": 29, "Department": "Finance"},
    {"Name": "Nina", "Age": 26, "Department": "Legal"}
]

# Save to JSON file
import json
with open("data.json", "w") as f:
    json.dump(json_data, f)

# Load JSON
df_json = pd.read_json("data.json")
print("\n📜 JSON Data:\n", df_json)
```

```
📜 JSON Data:
    Name  Age Department
0   Avi   29    Finance
1  Nina   26      Legal
```

print("\n🎓 CGPA List:\n", df_csv[['College_ID', 'CGPA']])

```
🎓 CGPA List:
     College_ID  CGPA
0       CLG0030  6.28
1       CLG0061  5.37
2       CLG0036  5.83
3       CLG0055  5.75
4       CLG0004  7.69
...         ...   ...
9995    CLG0021  8.29
9996    CLG0098  9.34
9997    CLG0066  6.25
9998    CLG0045  8.92
9999    CLG0060  9.77

[10000 rows x 2 columns]
```

Export to Excel

df_sql.to_excel("students_output.xlsx", index=False)

print("\n✅ SQL Data exported to 'students_output.xlsx'")

```
✅ SQL Data exported to 'students_output.xlsx'
```

EXPERIMENT-3

```
# eda_data_cleaning.py

import pandas as pd
from sklearn.preprocessing import StandardScaler, MinMaxScaler

# Load your dataset (replace with your actual file)
df = pd.read_csv(r"C:\Users\REC\Downloads\final_dataset.csv")  # <-- Change this

print("\nMissing values:\n", df.isnull().sum())

# ----------------------------------------
# 1. Handling Missing Values
# ----------------------------------------

# Fill missing values (customize based on your data)
df.fillna(method='ffill', inplace=True)       # Forward fill
df.fillna(method='bfill', inplace=True)       # Backward fill

# Optional: Drop remaining NaNs if needed
df.dropna(inplace=True)

# ----------------------------------------
# 2. Remove Duplicates & Unnecessary Columns
# ----------------------------------------

# Remove duplicate rows
duplicates = df.duplicated().sum()
if duplicates > 0:
```

```
    df.drop_duplicates(inplace=True)

# Drop unwanted columns (edit these column names)
columns_to_drop = ['Unnamed: 0', 'ID', 'Notes']  # Example columns
df.drop(columns=[col for col in columns_to_drop if col in df.columns], inplace=True)

# ----------------------------------------
# 3. Data Type Conversion & Consistency
# ----------------------------------------

# Convert to datetime
if 'Date' in df.columns:
    df['Date'] = pd.to_datetime(df['Date'], errors='coerce')

# Convert numerical columns to correct types (customize)
for col in df.select_dtypes(include='object').columns:
    try:
        df[col] = pd.to_numeric(df[col])
    except:
        pass  # Skip if not convertible

# Strip spaces and lowercase for categorical text columns
df.columns = df.columns.str.strip()
for col in df.select_dtypes(include='object').columns:
    df[col] = df[col].str.strip().str.lower()

# ----------------------------------------
# 4. Normalization (Standardization & Min-Max)
# ----------------------------------------
```

```
# Choose numerical columns to normalize
numeric_cols = df.select_dtypes(include='number').columns.tolist()

# Standardization
scaler_std = StandardScaler()
df[numeric_cols] = scaler_std.fit_transform(df[numeric_cols])

# OR Min-Max Scaling
# scaler_minmax = MinMaxScaler()
# df[numeric_cols] = scaler_minmax.fit_transform(df[numeric_cols])

# ----------------------------------------
# Final Check
# ----------------------------------------

print("\n✅ Cleaned Data Info:")
print(df.info())

print("\n📊 Statistical Summary:")
print(df.describe())

print("\n🔍 Preview:")
print(df.head())

# Save cleaned data if needed
df.to_csv('cleaned_dataset.csv', index=False)
print("\n💾 Cleaned data saved to 'cleaned_dataset.csv'")
```

```
Missing values:
 Date              0
Month              0
Year               0
Holidays_Count     0
Days               0
PM2.5              0
PM10               0
NO2                0
SO2                0
CO                 0
Ozone              0
AQI                0
dtype: int64

✅ Cleaned Data Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1461 entries, 0 to 1460
Data columns (total 12 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Date            1461 non-null   datetime64[ns]
 1   Month           1461 non-null   float64
 2   Year            1461 non-null   float64
 3   Holidays_Count  1461 non-null   float64
 4   Days            1461 non-null   float64
 5   PM2.5           1461 non-null   float64
 6   PM10            1461 non-null   float64
 7   NO2             1461 non-null   float64
 8   SO2             1461 non-null   float64
 9   CO              1461 non-null   float64
 10  Ozone           1461 non-null   float64
 11  AQI             1461 non-null   float64
dtypes: datetime64[ns](1), float64(11)
memory usage: 137.1 KB
None
```

📊 Statistical Summary:

	Date	Month	Year	\
count	1461	1461.000000	1.461000e+03	
mean	1970-01-01 00:00:00.000000015	0.000000	7.143362e-14	
min	1970-01-01 00:00:00.000000001	-1.601451	-1.342192e+00	
25%	1970-01-01 00:00:00.000000008	-0.731559	-4.480094e-01	
50%	1970-01-01 00:00:00.000000016	0.138333	4.461733e-01	
75%	1970-01-01 00:00:00.000000023	1.008226	1.340356e+00	
max	1970-01-01 00:00:00.000000031	1.588154	1.340356e+00	
std	NaN	1.000342	1.000342e+00	

	Holidays_Count	Days	PM2.5	PM10	NO2	\
count	1.461000e+03	1.461000e+03	1.461000e+03	1.461000e+03	1.461000e+03	
mean	3.404380e-17	8.936497e-17	2.723504e-16	2.456017e-16	-1.021314e-16	
min	-4.836866e-01	-1.499445e+00	-1.266642e+00	-1.613336e+00	-9.946512e-01	
25%	-4.836866e-01	-9.997437e-01	-6.910130e-01	-7.977291e-01	-5.652676e-01	
50%	-4.836866e-01	-3.420266e-04	-2.612812e-01	-1.425050e-01	-1.901250e-01	
75%	-4.836866e-01	9.990597e-01	3.870863e-01	6.153083e-01	2.222196e-01	
max	2.067455e+00	1.498761e+00	1.269406e+01	6.048431e+00	1.126834e+01	
std	1.000342e+00	1.000342e+00	1.000342e+00	1.000342e+00	1.000342e+00	

	SO2	CO	Ozone	AQI
count	1.461000e+03	1.461000e+03	1.461000e+03	1.461000e+03
mean	1.556288e-16	-2.431700e-16	2.334432e-16	2.918040e-17
min	-1.142516e+00	-1.242946e+00	-1.775633e+00	-1.700109e+00
25%	-7.494815e-01	-6.838246e-01	-6.460308e-01	-8.742313e-01
50%	-2.826776e-01	-2.891507e-01	-2.042190e-01	-1.225900e-01
75%	3.939461e-01	3.521945e-01	4.957123e-01	7.589645e-01
max	5.641257e+00	6.042077e+00	4.198064e+00	2.763341e+00
std	1.000342e+00	1.000342e+00	1.000342e+00	1.000342e+00

🔍 Preview:

	Date	Month	Year	Holidays_Count	Days	\
0	1970-01-01 00:00:00.000000001	-1.601451	-1.342192	-0.483687	0.499359	
1	1970-01-01 00:00:00.000000002	-1.601451	-1.342192	-0.483687	0.999060	
2	1970-01-01 00:00:00.000000003	-1.601451	-1.342192	2.067455	1.498761	
3	1970-01-01 00:00:00.000000004	-1.601451	-1.342192	-0.483687	-1.499445	
4	1970-01-01 00:00:00.000000005	-1.601451	-1.342192	-0.483687	-0.999744	

	PM2.5	PM10	NO2	SO2	CO	Ozone	AQI
0	4.440081	1.734582	3.505073	-0.432635	2.868241	0.361638	2.410719
1	4.373624	2.659354	0.444863	-0.902463	2.588680	-1.050893	2.596310
2	1.874953	0.161085	3.798712	-0.554778	0.615310	0.419702	0.564095
3	-0.017096	-0.666437	3.316792	-0.585616	-0.026035	0.678349	0.044441
4	-0.512586	-1.258606	2.427354	-0.629152	-0.634490	0.661986	-0.493771

EXPERIMENT-4

```
# eda_inspection_analysis_fixed.py

import pandas as pd

# Load the dataset
df = pd.read_csv(r"C:\Users\REC\Downloads\final_dataset.csv")

print("\n🔍 First 5 Rows:")
print(df.head())

print("\n📄 DataFrame Info:")
print(df.info())

print("\n🧮 Column-wise Data Types:")
print(df.dtypes)

print("\n📊 Summary Statistics:")
print(df.describe())

# ----------------------------------------
# 2. Filtering and Subsetting Data
# ----------------------------------------

# Example: Filter rows where AQI > 100
if 'AQI' in df.columns:
    high_aqi = df[df['AQI'] > 100]
    print("\n📉 Rows where AQI > 100:")
    print(high_aqi.head())
```

```
# Example: Filter where PM2.5 > 60 (unhealthy)
if 'PM2.5' in df.columns:
    high_pm25 = df[df['PM2.5'] > 60]
    print("\n🔥 Rows where PM2.5 > 60:")
    print(high_pm25.head())

# Subset only pollutant columns
pollutants = ['PM2.5', 'PM10', 'NO2', 'SO2', 'CO', 'Ozone', 'AQI']
existing_pollutants = [col for col in pollutants if col in df.columns]
print("\n📌 Subset with pollutant columns:")
print(df[existing_pollutants].head())

# ----------------------------------------
# 3. Descriptive Statistics
# ----------------------------------------

# Central Tendency
print("\n📏 Measures of Central Tendency:")
print("\nMean:\n", df.mean(numeric_only=True))
print("\nMedian:\n", df.median(numeric_only=True))
print("\nMode:\n", df.mode(numeric_only=True).iloc[0])

# Dispersion
print("\n📐 Measures of Dispersion:")
print("\nRange (max - min):\n", df.max(numeric_only=True) - df.min(numeric_only=True))
print("\nVariance:\n", df.var(numeric_only=True))
print("\nStandard Deviation:\n", df.std(numeric_only=True))
```

🔍 First 5 Rows:

	Date	Month	Year	Holidays_Count	Days	PM2.5	PM10	NO2	SO2	\
0	1	1	2021	0	5	408.80	442.42	160.61	12.95	
1	2	1	2021	0	6	404.04	561.95	52.85	5.18	
2	3	1	2021	1	7	225.07	239.04	170.95	10.93	
3	4	1	2021	0	1	89.55	132.08	153.98	10.42	
4	5	1	2021	0	2	54.06	55.54	122.66	9.70	

	CO	Ozone	AQI
0	2.77	43.19	462
1	2.60	16.43	482
2	1.40	44.29	263
3	1.01	49.19	207
4	0.64	48.88	149

📄 DataFrame Info:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1461 entries, 0 to 1460
Data columns (total 12 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Date            1461 non-null   int64  
 1   Month           1461 non-null   int64  
 2   Year            1461 non-null   int64  
 3   Holidays_Count  1461 non-null   int64  
 4   Days            1461 non-null   int64  
 5   PM2.5           1461 non-null   float64
 6   PM10            1461 non-null   float64
 7   NO2             1461 non-null   float64
 8   SO2             1461 non-null   float64
 9   CO              1461 non-null   float64
 10  Ozone           1461 non-null   float64
 11  AQI             1461 non-null   int64  
dtypes: float64(6), int64(6)
memory usage: 137.1 KB
None
```

Column-wise Data Types:

```
Date                int64
Month               int64
Year                int64
Holidays_Count      int64
Days                int64
PM2.5             float64
PM10              float64
NO2               float64
SO2               float64
CO                float64
Ozone             float64
AQI                 int64
dtype: object
```

Summary Statistics:

```
              Date        Month         Year  Holidays_Count         Days  \
count  1461.000000  1461.000000  1461.000000     1461.000000  1461.000000
mean     15.729637     6.522930  2022.501027        0.189596     4.000684
std       8.803105     3.449884     1.118723        0.392116     2.001883
min       1.000000     1.000000  2021.000000        0.000000     1.000000
25%       8.000000     4.000000  2022.000000        0.000000     2.000000
50%      16.000000     7.000000  2023.000000        0.000000     4.000000
75%      23.000000    10.000000  2024.000000        0.000000     6.000000
max      31.000000    12.000000  2024.000000        1.000000     7.000000

             PM2.5         PM10          NO2          SO2           CO  \
count  1461.000000  1461.000000  1461.000000  1461.000000  1461.000000
mean     90.774538   218.219261    37.184921    20.104921     1.025832
std      71.650579   129.297734    35.225327    16.543659     0.608305
min       0.050000     9.690000     2.160000     1.210000     0.270000
25%      41.280000   115.110000    17.280000     7.710000     0.610000
50%      72.060000   199.800000    30.490000    15.430000     0.850000
75%     118.500000   297.750000    45.010000    26.620000     1.240000
max    1000.000000  1000.000000   433.980000   113.400000     4.700000
```

```
             Ozone          AQI
count  1461.000000  1461.000000
mean     36.338871   202.210815
std      18.951204   107.801076
min       2.700000    19.000000
25%      24.100000   108.000000
50%      32.470000   189.000000
75%      45.730000   284.000000
max     115.870000   500.000000

📄 Rows where AQI > 100:
   Date  Month  Year  Holidays_Count  Days   PM2.5    PM10     NO2    SO2  \
0     1      1  2021               0     5  408.80  442.42  160.61  12.95
1     2      1  2021               0     6  404.04  561.95   52.85   5.18
2     3      1  2021               1     7  225.07  239.04  170.95  10.93
3     4      1  2021               0     1   89.55  132.08  153.98  10.42
4     5      1  2021               0     2   54.06   55.54  122.66   9.70

     CO  Ozone  AQI
0  2.77  43.19  462
1  2.60  16.43  482
2  1.40  44.29  263
3  1.01  49.19  207
4  0.64  48.88  149

🔥 Rows where PM2.5 > 60:
   Date  Month  Year  Holidays_Count  Days   PM2.5    PM10     NO2    SO2  \
0     1      1  2021               0     5  408.80  442.42  160.61  12.95
1     2      1  2021               0     6  404.04  561.95   52.85   5.18
2     3      1  2021               1     7  225.07  239.04  170.95  10.93
3     4      1  2021               0     1   89.55  132.08  153.98  10.42
5     6      1  2021               0     3  155.59  180.14  142.71  10.29

     CO  Ozone  AQI
0  2.77  43.19  462
1  2.60  16.43  482
2  1.40  44.29  263
3  1.01  49.19  207
5  1.18  44.47  252
```

📌 Subset with pollutant columns:

	PM2.5	PM10	NO2	SO2	CO	Ozone	AQI
0	408.80	442.42	160.61	12.95	2.77	43.19	462
1	404.04	561.95	52.85	5.18	2.60	16.43	482
2	225.07	239.04	170.95	10.93	1.40	44.29	263
3	89.55	132.08	153.98	10.42	1.01	49.19	207
4	54.06	55.54	122.66	9.70	0.64	48.88	149

✏️ Measures of Central Tendency:

Mean:

Date	15.729637
Month	6.522930
Year	2022.501027
Holidays_Count	0.189596
Days	4.000684
PM2.5	90.774538
PM10	218.219261
NO2	37.184921
SO2	20.104921
CO	1.025832
Ozone	36.338871
AQI	202.210815

dtype: float64

Median:

Date	16.00
Month	7.00
Year	2023.00
Holidays_Count	0.00
Days	4.00
PM2.5	72.06
PM10	199.80
NO2	30.49
SO2	15.43
CO	0.85
Ozone	32.47
AQI	189.00

dtype: float64

```
Mode:
 Date                 1.00
Month                 1.00
Year               2024.00
Holidays_Count        0.00
Days                  1.00
PM2.5                31.18
PM10                116.78
NO2                   4.82
SO2                   8.01
CO                    0.69
Ozone                16.85
AQI                  55.00
Name: 0, dtype: float64

📐 Measures of Dispersion:

Range (max - min):
 Date                30.00
Month               11.00
Year                 3.00
Holidays_Count       1.00
Days                 6.00
PM2.5              999.95
PM10               990.31
NO2                431.82
SO2                112.19
CO                   4.43
Ozone              113.17
AQI                481.00
dtype: float64
```

```
Variance:
 Date                 77.494662
Month                11.901700
Year                  1.251540
Holidays_Count        0.153755
Days                  4.007534
PM2.5              5133.805518
PM10              16717.903971
NO2                1240.823645
SO2                 273.692644
CO                    0.370035
Ozone               359.148138
AQI               11621.071965
dtype: float64

Standard Deviation:
 Date                 8.803105
Month                3.449884
Year                 1.118723
Holidays_Count       0.392116
Days                 2.001883
PM2.5               71.650579
PM10               129.297734
NO2                 35.225327
SO2                 16.543659
CO                   0.608305
Ozone               18.951204
AQI                107.801076
dtype: float64
```