Course Name:	Data Analysis Laboratory (216H03L501)	Semester:	V
Date of Performance:	14 / 07 / 2025	DIV/ Batch No:	HDA2
Student Name:	Aaryan Sharma	Roll No:	16010123012

Experiment No: 1

Title: Studies on Pandas library of python.

Objectives of the Experiment:

1. To understand and apply the fundamental functionalities of the pandas library for data analysis.
2. To manipulate and transform datasets using filtering, sorting, and column operations.
3. To analyze data using grouping and aggregation techniques to derive meaningful insights.

COs to be achieved:

CO1: Understand basic concepts of data analytics to solve real-world problems

Books/ Journals/ Websites referred:

1. https://www.kaggle.com/datasets
2. https://www.w3schools.com/python/pandas

Theory:

Pandas is a Python library used for working with data sets.
It has functions for analyzing, cleaning, exploring, and manipulating data.

Problem statement/ Tasks

Task 1: Import Required Libraries and Dataset

- Import pandas and load a real-world CSV dataset (e.g., Titanic, Student Performance, COVID-19).
- Display the first and last 5 records using head() and tail().

Task 2: Basic Exploration of the Dataset

- Display the dataset shape using .shape, column names using .columns, and data types using .dtypes.
- Generate summary statistics using .describe() and data info using .info().

Task 3: Identify Missing and Duplicate Data

- Detect missing values using .isnull().sum().
- Remove or fill missing values using .dropna() or .fillna().

- Check for and remove duplicate rows using .duplicated() and .drop_duplicates().

Task 4: Filtering Records

- Extract rows based on specific conditions (e.g., students who scored more than 80%, passengers who survived).

Task 5: Sorting the Dataset

- Sort the dataset based on one or more columns using .sort_values().
  - Example: Sort by age or total score.

Task 6: Creating or Modifying Columns

- Create new columns from existing ones (e.g., Total Marks = Math + Science + English).
- Drop unnecessary columns using .drop().
- Rename columns using .rename().

Task 7: Grouping and Aggregation

- Use .groupby() to find average, count, or sum based on a categorical column.
- Example:
  - Average marks by gender: df.groupby('Gender')['Marks'].mean()
  - Survival rate by class: df.groupby('Pclass')['Survived'].mean()

Task 8: Pivot Tables or Multi-Level Grouping (Optional for advanced students)

- Create pivot tables using .pivot_table() to summarize complex data.
  - Example: Average score by gender and class.

Task 9: Insight Generation

- Write 3-5 key insights based on the group-by and aggregated data.
- Example:
  - "Female students have higher average marks in English."
  - "Survival rate is highest for first-class passengers."

Code:

```
[ ]  # Print the version
     print(f"pandas version: {pd.__version__}")

     pandas version: 2.2.2
```

```
csp = pd.read_csv(os.path.join(path, "college_student_placement_dataset.csv"))
csp
```

	College_ID	IQ	Prev_Sem_Result	CGPA	Academic_Performance	Internship_Experience	Extra_Curricular_Score	Communication_Skills	Projects_Completed	Placement
0	CLG0030	107	6.61	6.28	8	No	8	8	4	No
1	CLG0061	97	5.52	5.37	8	No	7	8	0	No
2	CLG0036	109	5.36	5.83	9	No	3	1	1	No
3	CLG0055	122	5.47	5.75	6	Yes	1	6	1	No
4	CLG0004	96	7.91	7.69	7	No	8	10	2	No
...	...	...	...	...	...	...	...	...	...	...
9995	CLG0021	119	8.41	8.29	4	No	1	8	0	Yes
9996	CLG0098	70	9.25	9.34	7	No	0	7	2	No
9997	CLG0066	89	6.08	6.25	3	Yes	3	9	5	No
9998	CLG0045	107	8.77	8.92	3	No	7	5	1	No
9999	CLG0060	109	9.41	9.77	8	No	3	5	5	No

10000 rows × 10 columns

```
csp.describe()
```

	IQ	Prev_Sem_Result	CGPA	Academic_Performance	Extra_Curricular_Score	Communication_Skills	Projects_Completed
count	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000
mean	99.471800	7.535673	7.532379	5.546400	4.970900	5.561800	2.513400
std	15.053101	1.447519	1.470141	2.873477	3.160103	2.900866	1.715959
min	41.000000	5.000000	4.540000	1.000000	0.000000	1.000000	0.000000
25%	89.000000	6.290000	6.290000	3.000000	2.000000	3.000000	1.000000
50%	99.000000	7.560000	7.550000	6.000000	5.000000	6.000000	3.000000
75%	110.000000	8.790000	8.770000	8.000000	8.000000	8.000000	4.000000
max	158.000000	10.000000	10.460000	10.000000	10.000000	10.000000	5.000000

```
csp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 10 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   College_ID              10000 non-null  object
 1   IQ                      10000 non-null  int64
 2   Prev_Sem_Result         10000 non-null  float64
 3   CGPA                    10000 non-null  float64
 4   Academic_Performance    10000 non-null  int64
 5   Internship_Experience   10000 non-null  object
 6   Extra_Curricular_Score  10000 non-null  int64
 7   Communication_Skills    10000 non-null  int64
 8   Projects_Completed      10000 non-null  int64
 9   Placement               10000 non-null  object
dtypes: float64(2), int64(5), object(3)
memory usage: 781.4+ KB
```

```
[ ] print("Missing values:\n", csp.isnull().sum())
```

```
Missing values:
 College_ID                0
IQ                         0
Prev_Sem_Result            0
CGPA                       0
Academic_Performance       0
Internship_Experience      0
Extra_Curricular_Score     0
Communication_Skills       0
Projects_Completed         0
Placement                  0
dtype: int64
```

```
csp['CGPA'].value_counts()
```

CGPA	count
9.41	35
7.29	32
6.72	32
6.09	31
9.47	31
...	...
10.44	1
4.57	1
4.67	1
4.56	1
4.54	1

590 rows × 1 columns

dtype: int64

```
[ ]  print("Total student=",csp['CGPA'].value_counts())
```

```
Total student= CGPA
9.41     35
7.29     32
6.72     32
6.09     31
9.47     31
         ..
10.44     1
4.57      1
4.67      1
4.56      1
4.54      1
Name: count, Length: 590, dtype: int64
```

```
[ ]  csp['CGPA'].value_counts()
```

CGPA	count
9.41	35
7.29	32
6.72	32
6.09	31
9.47	31
...	...
10.44	1
4.57	1
4.67	1
4.56	1
4.54	1

590 rows × 1 columns

dtype: int64

```
csp['College_ID'].str.lower()
```

	College_ID
0	clg0030
1	clg0061
2	clg0036
3	clg0055
4	clg0004
...	...
9995	clg0021
9996	clg0098
9997	clg0066
9998	clg0045
9999	clg0060

10000 rows × 1 columns

dtype: object

```
[ ] csp['Placement'].value_counts()
```

Placement	count
No	8341
Yes	1659

dtype: int64

```
[ ] print("Total placed = ",csp['Placement'].value_counts()['Yes'])
```

```
Total placed =  1659
```

```
print("First 5 records:")
print(csp.head())

print("\nLast 5 records:")
print(csp.tail())
```

```
First 5 records:
  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

  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

   Projects_Completed Placement
0                   4        No
1                   0        No
2                   1        No
3                   1        No
4                   2        No

Last 5 records:
     College_ID   IQ  Prev_Sem_Result  CGPA  Academic_Performance  \
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  \
9995                    No                       1                     8
9996                    No                       0                     7
9997                   Yes                       3                     9
9998                    No                       7                     5
9999                    No                       3                     5

      Projects_Completed Placement
9995                   0       Yes
9996                   2        No
9997                   5        No
9998                   1        No
9999                   5        No
```

```
print("Dataset Shape:", csp.shape)
print("\nColumn Names:", csp.columns)
print("\nData Types:\n", csp.dtypes)
```

```
Dataset Shape: (10000, 10)

Column Names: Index(['College_ID', 'IQ', 'Prev_Sem_Result', 'CGPA', 'Academic_Performance',
       'Internship_Experience', 'Extra_Curricular_Score',
       'Communication_Skills', 'Projects_Completed', 'Placement'],
      dtype='object')

Data Types:
 College_ID                object
IQ                         int64
Prev_Sem_Result          float64
CGPA                     float64
Academic_Performance       int64
Internship_Experience     object
Extra_Curricular_Score     int64
Communication_Skills       int64
Projects_Completed         int64
Placement                 object
dtype: object
```

```
csp.dropna(inplace=True)
print("\nMissing values after handling:\n", csp.isnull().sum())

print("\nNumber of duplicate rows:", csp.duplicated().sum())

csp.drop_duplicates(inplace=True)
print("\nDataset shape after removing duplicates:", csp.shape)
```

```

Missing values after handling:
 College_ID                0
IQ                         0
Prev_Sem_Result            0
CGPA                       0
Academic_Performance       0
Internship_Experience      0
Extra_Curricular_Score     0
Communication_Skills       0
Projects_Completed         0
Placement                  0
dtype: int64

Number of duplicate rows: 0

Dataset shape after removing duplicates: (10000, 10)
```

```
high = csp[csp['CGPA'] > 8.0]
print("\nStudents who scored more than 8 CGPA:\n", high)
```

```
Students who scored more than 8 CGPA:
      College_ID   IQ  Prev_Sem_Result  CGPA  Academic_Performance  \
7       CLG0096  111             8.77  8.76                     7
9       CLG0057  108             8.82  8.60                     4
10      CLG0063   93             8.73  8.90                     2
12      CLG0064  103             8.64  9.01                     7
13      CLG0017   71             8.74  8.40                     6
...         ...  ...              ...   ...                   ...
9994    CLG0064  117             8.71  8.44                     6
9995    CLG0021  119             8.41  8.29                     4
9996    CLG0098   70             9.25  9.34                     7
9998    CLG0045  107             8.77  8.92                     3
9999    CLG0060  109             9.41  9.77                     8

     Internship_Experience  Extra_Curricular_Score  Communication_Skills  \
7                       No                       3                     1
9                       No                       5                     9
10                     Yes                       5                     6
12                     Yes                       8                     6
13                      No                       0                     5
...                    ...                     ...                   ...
9994                    No                       9                     4
9995                    No                       1                     8
9996                    No                       0                     7
9998                    No                       7                     5
9999                    No                       3                     5

      Projects_Completed Placement
7                      2       Yes
9                      1        No
10                     0        No
12                     1        No
13                     2        No
...                  ...       ...
9994                   4       Yes
9995                   0       Yes
9996                   2        No
9998                   1        No
9999                   5        No

[4088 rows x 10 columns]
```

```
csp_sorted = csp.sort_values(by='CGPA')
print("\nDataset sorted by CGPA:")
print(csp_sorted.head())
```

```
Dataset sorted by CGPA:
     College_ID   IQ  Prev_Sem_Result  CGPA  Academic_Performance  \
9333    CLG0044  113             5.03  4.54                     6
8835    CLG0017   99             5.01  4.56                     5
8574    CLG0025   97             5.06  4.57                     6
3332    CLG0005  123             5.02  4.58                     3
319     CLG0029   98             5.03  4.59                    10

     Internship_Experience  Extra_Curricular_Score  Communication_Skills  \
9333                   Yes                       9                     5
8835                   Yes                       0                     9
8574                    No                       7                     5
3332                   Yes                       2                     9
319                     No                      10                     4

      Projects_Completed Placement
9333                   4        No
8835                   4        No
8574                   2        No
3332                   5       Yes
319                    0        No
```

```
csp['Knowledge'] = csp['IQ'] + csp['CGPA'] + csp['Academic_Performance']

csp.rename(columns={'College_ID': 'University_ID', 'CGPA': 'Cumulative_Grade_Point_Average'}, inplace=True)

print("\nDataset with new column and renamed columns:")
print(csp.head())
```

```
Dataset with new column and renamed columns:
  University_ID   IQ  Prev_Sem_Result  Cumulative_Grade_Point_Average  \
0       CLG0030  107             6.61                            6.28
1       CLG0061   97             5.52                            5.37
2       CLG0036  109             5.36                            5.83
3       CLG0055  122             5.47                            5.75
4       CLG0004   96             7.91                            7.69

   Academic_Performance Internship_Experience  Extra_Curricular_Score  \
0                     8                    No                       8
1                     8                    No                       7
2                     9                    No                       3
3                     6                   Yes                       1
4                     7                    No                       8

   Communication_Skills  Projects_Completed Placement  Knowledge
0                     8                   4        No     121.28
1                     8                   0        No     110.37
2                     1                   1        No     123.83
3                     6                   1        No     133.75
4                    10                   2        No     110.69
```

```
print(csp.groupby('Internship_Experience')['Cumulative_Grade_Point_Average'].mean())

print("\nCount of students by Internship Experience:")
print(csp.groupby('Internship_Experience').size())

print("\nSum of Projects Completed by Internship Experience:")
print(csp.groupby('Internship_Experience')['Projects_Completed'].sum())
```

```
Internship_Experience
No     7.547344
Yes    7.509591
Name: Cumulative_Grade_Point_Average, dtype: float64

Count of students by Internship Experience:
Internship_Experience
No     6036
Yes    3964
dtype: int64

Sum of Projects Completed by Internship Experience:
Internship_Experience
No     15257
Yes     9877
Name: Projects_Completed, dtype: int64
```

```
pivot_table_result = csp.pivot_table(
    values='Academic_Performance',
    index='IQ',
)
pivot_table_result
```

IQ	Academic_Performance
41	2.0
42	2.0
44	8.0
45	2.0
51	4.0
...	...
148	1.0
150	3.0
152	5.0
157	4.0
158	8.0

104 rows × 1 columns

Post Lab Subjective/Objective type Questions:

1. **What is the difference between .info() and .describe() in pandas?**
   The .info() and .describe() functions in pandas serve different purposes when exploring a dataset. The .info() method provides a concise summary of the DataFrame, including the number of non-null entries, column names, data types, and memory usage. It is particularly useful for getting a quick overview of the dataset's structure and identifying missing values. On the other hand, the .describe() method generates descriptive statistics for numerical columns by default, such as count, mean, standard deviation, minimum and maximum. It helps in understanding the distribution and spread of the data. While .info() focuses on the structure and completeness of the data, .describe() is more concerned with the statistical properties and variability of the dataset.

2. **How does pandas handle missing data? Mention at least two functions used for this purpose.**
   Pandas handles missing data using NaN (Not a Number) values. df.isnull() / df.notnull(): Identify missing data df.dropna(): Removes rows or columns with missing values.

3. **What is a pivot table in pandas, and how is it useful in summarizing data?**
   A pivot table in pandas is a powerful tool that allows you to summarize and aggregate data using multiple dimensions which is created using pd.pivot_table(). It is Useful for quickly reshaping data, comparing subgroups, and generating report. Ex: Group sales data by Region and Product and calculate average sales.

4. **What were the key insights you discovered from the dataset during your analysis?**
   During the analysis of the College Student Placement Factors dataset, several key insights emerged. Firstly, students with higher CGPAs and consistent academic performance across 10th and 12th grades were more likely to get placed. Secondly, participation in extracurricular activities and internships showed a positive correlation with placement outcomes, indicating the importance of soft skills and practical exposure. Additionally, students with strong communication and technical skills, as reflected in the dataset, had a higher placement rate. It was also observed that students from specific degree programs and colleges had better placement statistics, highlighting institutional influence. These insights can help institutions and students focus on the most impactful factors to improve employability.

Conclusion:

I have successfully used pandas, which involved exploring and analyzing data in Python. Through this experiment, I learned how to load real-world datasets, perform basic data exploration using functions like .info() and .describe(), handle missing and duplicate data, apply filtering, sorting. This experiment helped me understand the practical importance of data analysis in identifying trends and making informed decisions.