

Getting Information from a CSV

Name		Date	
Class / Sec		Roll no.	

1 · AIM

To write a Python program that reads a **CSV file** and displays information about it — its **size**, its **column names**, its **data types** and a **statistical summary** — using Pandas.

2 · SOFTWARE AND LIBRARIES USED

TOOL	WHAT IT IS FOR
Python 3	The programming language the program is written in
Jupyter Notebook	The environment used to write and run the code (IDLE or Colab also acceptable)
Pandas	The data library. read_csv() loads a file into a DataFrame — a table Python can work with

3 · THEORY

Before analysing a dataset you need to know **what is in it**. Pandas provides four methods that answer that question, and each one answers a different part of it.

METHOD	WHAT IT RETURNS	THE QUESTION IT ANSWERS
df.shape	(rows, columns)	How big is this dataset?
df.columns	The list of column names	What fields does it contain?
df.info()	Types, non-null counts, memory	Is anything missing, and what type is each column?
df.describe()	count, mean, std, min, max, quartiles	What do the numbers look like?

This is Data Exploration: These four methods are the third stage of the **AI Project Cycle**. You are looking at the data before building anything with it — checking its size, its types, and whether anything is missing.

4 · PROGRAM

```
import pandas as pd

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

# 1. Size of the dataset
print("Shape (rows, columns):", df.shape)
print("Number of rows      :", df.shape[0])
print("Number of columns   :", df.shape[1])

# 2. Column names
print()
print("Column names:", list(df.columns))

# 3. Data types and missing values
print()
print("Information about the data:")
```

```
df.info()

# 4. Statistical summary of the numeric columns
print()
print("Statistical summary:")
print(df.describe())
```

Note on df.info(): `info()` prints its own output directly — it is written as `df.info()` on its own line, not inside a `print()`. Wrapping it in `print()` adds a stray **None** at the end.

5 • EXPECTED OUTPUT

Running the program prints four blocks of information.

```
Shape (rows, columns): (8, 5)
Number of rows      : 8
Number of columns   : 5

Column names: ['RollNo', 'Name', 'Class', 'Marks', 'Attendance']

Information about the data:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8 entries, 0 to 7
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   RollNo      8 non-null      int64
1   Name        8 non-null      object
2   Class       8 non-null      object
3   Marks       8 non-null      int64
4   Attendance  8 non-null      int64
dtypes: int64(3), object(2)

Statistical summary:
```

	RollNo	Marks	Attendance
count	8.000000	8.000000	8.000000
mean	4.500000	73.875000	87.875000
std	2.44949	13.431706	9.218576
min	1.000000	54.000000	71.000000
25%	2.750000	63.750000	83.500000
50%	4.500000	75.000000	90.000000
75%	6.250000	84.250000	94.500000
max	8.000000	91.000000	98.000000

If your output differs slightly: Very recent versions of Pandas print **str** instead of **object** for text columns, and the memory usage line may vary. Both are correct — do not worry if yours is not identical.

[PASTE SCREENSHOT HERE]

Paste a screenshot of YOUR program running, with the code and output visible together

6 • READING THE OUTPUT

Each block answers a different question about the dataset.

WHAT IT SHOWS	WHAT YOU LEARN FROM IT
(8, 5)	8 rows, 5 columns. The order is always (rows, columns) — never the other way round

8 non-null, every column	No missing data. If a column showed 6 non-null, two values would be missing
int64 and object	int64 is a whole number; object is text. Name and Class are text, the rest are numbers
describe() has 3 columns	Only the numeric columns appear. Name and Class are text, so there is nothing to average
mean Marks 73.875	Check it by hand: the eight marks total 591 , and $591 \div 8 = \mathbf{73.875}$

7 • OBSERVATION AND RESULT

Two or three sentences in your own words. Say what the four methods told you about the dataset.

Example observation: The dataset contains 8 rows and 5 columns, with no missing values — every column shows 8 non-null entries. Three columns are numeric and two hold text, so `describe()` summarised only the numeric ones. The average marks were 73.88 out of 100, with a lowest of 54 and a highest of 91.

Worth knowing: `RollNo` appears in `describe()` even though it is not really a measurement. Pandas summarises any numeric column, and it cannot tell an ID number from a score. Averaging a roll number is meaningless — noticing that shows real understanding.

8 • VIVA QUESTIONS FOR THIS PROGRAM

LIKELY QUESTION	SHAPE OF A STRONG ANSWER
What does <code>df.shape</code> return?	A pair — (rows, columns) . Here (8, 5): eight rows and five columns
How do you know nothing is missing?	info() shows 8 non-null for every column, and there are 8 rows in total
Why does <code>describe()</code> show only three columns?	It summarises numeric columns only. Name and Class hold text, so there is nothing to calculate
What does <code>int64</code> mean?	A whole number stored in 64 bits. object means text
Is the mean of <code>RollNo</code> useful?	No. It is an identifier , not a measurement. Pandas averages it because it cannot tell the difference

9 • HOW THIS ENTRY IS MARKED

FULL MARKS

All **four methods** used, not just one · the CSV submitted with the program · a screenshot of **your own** output · an observation naming **rows, columns and missing values** · you can explain why `describe()` shows only numeric columns

LOSES MARKS

Only **shape** used, ignoring the other three methods · no CSV submitted · an observation that just repeats the printed numbers · cannot say what **non-null** or **object** mean

Before you submit: Run all **four** methods, not just `shape`. And check one number by hand — add the marks and divide by the row count. Matching `describe()` proves you understood the output rather than pasted it.