Python Programming
Unit 06 – Lecture 04 Notes
Missing Data, Aggregation, Combining, CSV I/O

Tofik Ali

February 14, 2026

Contents

1 Lecture Overview

Real-world datasets are rarely perfect. Common issues:

- missing values,
- inconsistent categories,
- multiple tables that must be combined,
- and reading/writing data files (CSV).

This lecture teaches practical Pandas tools to handle these issues.

2 Core Concepts

2.1 Missing Values

Pandas represents missing numeric values as `NaN`. Useful functions:

- `df.isna()` and `df.notna()`
- `df.dropna()` remove rows/columns with missing data
- `df.fillna(value)` fill missing values

```
print(df.isna().sum())
df["marks"] = df["marks"].fillna(df["marks"].mean())
```

Important: filling strategy depends on context. Replacing missing marks with 0 may be wrong if the mark is unknown (not actually 0).

2.2 Aggregation with groupby

`groupby` groups rows by a category and then applies a function:

```
avg = df.groupby("city")["marks"].mean()
count = df.groupby("city")["sapid"].count()
```

2.3 Combining DataFrames

`concat` stacks data:

```
combined = pd.concat([df1, df2], ignore_index=True)
```

`merge` joins data like SQL:

```
merged = pd.merge(df, city_state, on="city", how="left")
```

2.4 CSV Import/Export

```
df = pd.read_csv("data/student_scores.csv")
df.to_csv("data/cleaned.csv", index=False)
```

3 Demo Walkthrough

Data: `data/student_scores.csv`
Script: `demo/pandas_missing_data_demo.py`

The demo:

- prints missing-value counts,
- fills missing `city` and `marks`,
- computes average marks per city,
- merges a city-to-state mapping,
- exports a cleaned CSV.

4 Interactive Checkpoints (with Solutions)

Checkpoint 1 Solution

Question: what does `fillna(0)` do?
Answer: it replaces missing values (`NaN`) with 0 in the selected Series/DataFrame.

Checkpoint 2 Solution

Question: when use `merge` instead of `concat`?
Answer:

- Use `merge` when you want to join tables using a key (like `city`).
- Use `concat` when you want to stack rows/columns.

5 Practice Exercises (with Solutions)

Exercise 1: Fill Missing Marks with Median

Solution:

```
median = df["marks"].median()
df["marks"] = df["marks"].fillna(median)
```

Exercise 2: City-wise Count

Solution:

```
print(df.groupby("city")["sapid"].count())
```

6 Exit Question (with Solution)

Question: read `"marks.csv"` into a DataFrame?
Answer:

```
df = pd.read_csv("marks.csv")
```