Statistics and Data Analysis
Unit 01 – Lecture 03 Notes
Preprocessing Pipelines and Exploratory Data Analysis (EDA)

Tofik Ali

February 17, 2026

What You Will Learn

In this lecture we move from individual cleaning techniques to a full workflow:

- how to organize preprocessing as a **pipeline**,
- how to run a basic **EDA** (exploratory data analysis),
- how to choose plots based on variable types,
- and how to avoid **data leakage**.

1. Preprocessing Pipelines

1.1 What is a pipeline?

A pipeline is an ordered set of steps that you apply consistently to data. For example:

1. read raw data,
2. clean and validate,
3. create summaries and plots,
4. save cleaned data and reports.

1.2 Why pipelines matter

Pipelines are important because they:

- make your work **reproducible** (you can re-run on new data),
- reduce errors (clear steps, less manual editing),
- make collaboration easier (others can follow the same steps),
- help avoid **data leakage** in machine learning workflows.

Exercise 1 (solution)

One reasonable order is:

1. Load raw data
2. Check missingness
3. Fix data types + invalid ranges
4. EDA plots
5. Save cleaned dataset

2. Exploratory Data Analysis (EDA)

2.1 What is EDA?

EDA is the first structured look at the dataset to understand:

- what the data contains,
- how clean it is,
- what the distributions look like,
- and what relationships might exist.

EDA is not only about plots. It also includes summary tables, missingness reports, and sanity checks.

2.2 Minimum EDA checklist

1. **Shape and columns**: number of rows and columns; column meanings.
2. **Data types**: numeric/categorical/datetime.
3. **Missingness**: missing % per column.
4. **Ranges**: are values possible? (0–100%, CGPA 0–10, etc.)
5. **Univariate summaries**: `describe()` for numeric; counts for categories.
6. **Relationships**: scatter plots and correlations for numeric features.
7. **Group comparisons**: program-wise or gender-wise summaries.

2.3 Plot selection

- Numeric (one variable): histogram / boxplot
- Categorical (one variable): bar chart (counts)
- Numeric vs numeric: scatter plot
- Numeric vs categorical: boxplot grouped by category
- Many numeric features: correlation matrix/heatmap

Exercise 2 (solution)

- Distribution of `final_marks`: histogram or boxplot.
- Compare `final_marks` across `program`: boxplot grouped by program.
- Relationship between `study_hours_week` and `final_marks`: scatter plot.

3. Data Leakage (Very Important)

3.1 What is leakage?

Leakage happens when we accidentally use information from the test set (future/unseen data) during training. This makes the results look better than reality.

Common leakage examples:

- computing scaling parameters using the full dataset before train/test split,
- imputing missing values using the full dataset mean/median before split,
- using future data to predict the past (time series leakage).

Exercise 3 (solution)

Not correct. The fix is:

- split into train/test first,
- compute scaling/imputation rules on **train only**,
- apply them to test.

4. Mini Demo (Python)

Run from the lecture folder:

```
python demo/pipeline_eda_demo.py
```

The script does:

- reads `data/case_study.csv`,
- cleans and validates (range checks + median imputation),
- saves `data/case_study_clean.csv`,
- saves group summary and correlation matrix as CSV,
- saves four plots into `images/`.

References

- McKinney, W. *Python for Data Analysis.* O'Reilly, 2022.
- Tukey, J. W. *Exploratory Data Analysis.* Addison-Wesley, 1977.
- Montgomery, D. C., & Runger, G. C. *Applied Statistics and Probability for Engineers.* Wiley, 7th ed., 2020.

Appendix: Slide Deck Content (Reference)

The material below is a reference copy of the slide deck content. Exercise solutions are explained in the main notes where applicable.

Title Slide

https://github.com/tali7c/Statistics-and-Data-Analysis

Quick Links

Workflow EDA Checklist Plots Demo Summary

Agenda

- Overview
- Workflow and Pipelines
- EDA Checklist
- Plots
- Demo
- Summary

Learning Outcomes

- Explain what a preprocessing pipeline is and why it matters
- Apply a simple end-to-end workflow: load → clean → validate → summarize
- Perform basic EDA: missingness, summary stats, group summaries, correlations
- Choose appropriate plots for numeric and categorical variables

What is a Pipeline?

A pipeline is an ordered set of steps applied consistently to data.

- Makes analysis **reproducible** (same input ⇒ same output)
- Reduces mistakes (steps are documented and repeatable)
- Helps avoid **data leakage** (train/test separation)

Typical End-to-End Workflow (Practical)

1. Understand the question (what do you want to learn/decide?)
2. Acquire data (files, DB, API)
3. Inspect: shape, dtypes, missingness
4. Clean: duplicates, invalid ranges, inconsistent categories
5. Validate: check constraints (0–100%, 0–10 CGPA, etc.)
6. EDA: summary tables + plots + simple relationships
7. Save outputs (cleaned dataset, plots, summary tables)

Example: "Pipeline" in Code (Concept)

```
df = read_raw()
df = clean_strings(df)
df = coerce_types(df)
df = range_check(df)
df = impute_missing(df)
save_clean(df)
eda_report(df)
```

This is a simple pipeline: each step has a clear purpose.

Exercise 1: Put Steps in Order

Arrange these steps in a reasonable order:

1. EDA plots
2. Load raw data
3. Fix data types + invalid ranges
4. Save cleaned dataset
5. Check missingness

Solution 1

One reasonable order:

1. Load raw data
2. Check missingness
3. Fix data types + invalid ranges
4. EDA plots
5. Save cleaned dataset

What is EDA?

Exploratory Data Analysis (EDA) is the first structured look at your data.

- Understand distribution (shape, spread, outliers)
- Understand relationships (scatter plots, correlation)
- Compare groups (e.g., program-wise summaries)
- Identify issues early (missingness, strange values)

EDA Checklist (Minimum)

- **Data quality:** missingness %, duplicates, invalid ranges
- **Univariate:** histograms/boxplots for numeric; bar charts for categorical
- **Bivariate:** scatter plot for numeric–numeric; boxplot for numeric by category
- **Multivariate (basic):** correlation matrix/heatmap for numeric columns
- **Group summaries:** mean/median/std by program or gender

Plot Selection (Quick Guide)

- Numeric (one variable): histogram, boxplot
- Categorical (one variable): bar chart (counts)
- Numeric vs numeric: scatter plot
- Numeric vs categorical: boxplot (numeric grouped by category)
- Many numeric features: correlation heatmap

Exercise 2: Choose the Plot

Pick a good plot for each:

1. Distribution of `final_marks` (numeric)
2. Compare `final_marks` across `program` (categorical)
3. Relationship between `study_hours_week` and `final_marks`

Solution 2

- (1) Histogram or boxplot
- (2) Boxplot of marks grouped by program
- (3) Scatter plot (hours vs marks)

Exercise 3: Spot Data Leakage

A student computes mean/std for scaling using the **entire dataset**, then splits into train/test and trains a model.

Question: Is this correct? If not, what should be done instead?

Solution 3

Not correct: it uses test information during training (**leakage**).
Correct approach:

- split into train/test first
- compute scaling parameters on **train only**
- apply the same parameters to test

Mini Demo (Python)

Run from the lecture folder:

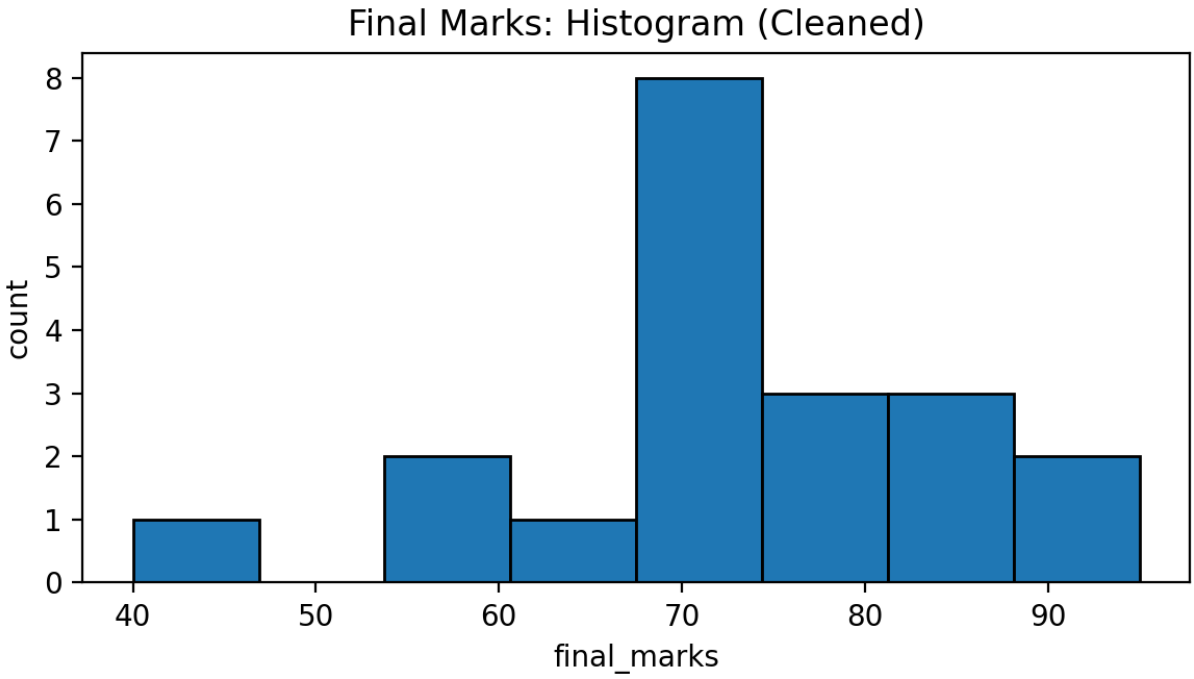
```
python demo/pipeline_eda_demo.py
```

Outputs:

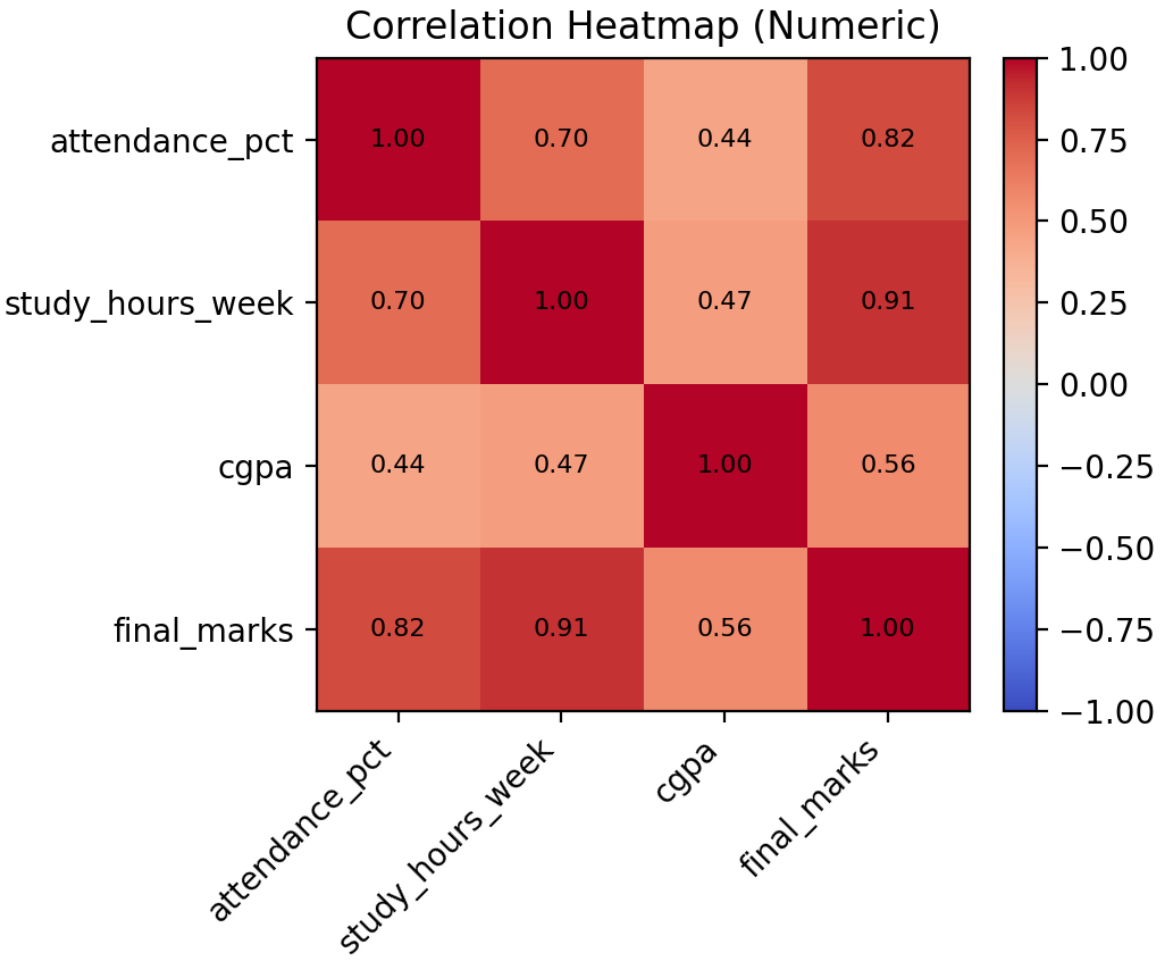
- `data/case_study_clean.csv`
- `data/summary_by_program.csv`
- `data/corr_matrix.csv`
- plots in `images/` (histogram, boxplot, scatter, heatmap)

Demo Output (Example)

Histogram



Correlation Heatmap



Summary

- Pipelines make preprocessing repeatable and reduce mistakes
- EDA is about understanding quality, distributions, and relationships
- Pick plots based on variable types (numeric vs categorical)
- Save cleaned data, plots, and summary tables as reusable artifacts

Exit question: Name two checks you must do before trusting a dataset for analysis.