Statistics and Data Analysis

Unit 01 – Lecture 01: Data Types, Sources, and Cleaning Basics

Tofik Ali

School of Computer Science, UPES Dehradun

February 17, 2026

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

Quick Links

Types & Formats Sources Cleaning Demo Summary

Agenda

1. Overview
2. Data Types and Formats
3. Data Sources and Acquisition
4. Data Cleaning
5. Demo
6. Summary

Learning Outcomes

- Identify common data types and formats used in analytics

Learning Outcomes

- Identify common data types and formats used in analytics
- List common data sources and acquisition methods

Learning Outcomes

- Identify common data types and formats used in analytics
- List common data sources and acquisition methods
- Detect typical data quality issues (missing values, duplicates, outliers)

Learning Outcomes

- Identify common data types and formats used in analytics
- List common data sources and acquisition methods
- Detect typical data quality issues (missing values, duplicates, outliers)
- Apply basic cleaning steps in Python and save a cleaned dataset

Dataset, Observation, Variable

- **Dataset:** a collection of observations (rows) and variables (columns)

Goal: convert raw data into a form suitable for analysis and modeling.

Dataset, Observation, Variable

- **Dataset:** a collection of observations (rows) and variables (columns)
- **Observation:** one record (e.g., one student)

Goal: convert raw data into a form suitable for analysis and modeling.

Dataset, Observation, Variable

- **Dataset:** a collection of observations (rows) and variables (columns)
- **Observation:** one record (e.g., one student)
- **Variable/Feature:** one attribute (e.g., attendance, CGPA)

Goal: convert raw data into a form suitable for analysis and modeling.

Common Data Types (Practical View)

- **Numeric:** integers (count), real values (measurements)

Common Data Types (Practical View)

- **Numeric:** integers (count), real values (measurements)
- **Categorical:** nominal (branch), ordinal (rating: low/med/high)

Common Data Types (Practical View)

- **Numeric:** integers (count), real values (measurements)
- **Categorical:** nominal (branch), ordinal (rating: low/med/high)
- **Binary:** yes/no, pass/fail

Common Data Types (Practical View)

- **Numeric:** integers (count), real values (measurements)
- **Categorical:** nominal (branch), ordinal (rating: low/med/high)
- **Binary:** yes/no, pass/fail
- **Date/Time:** join date, timestamp

Common Data Types (Practical View)

- **Numeric:** integers (count), real values (measurements)
- **Categorical:** nominal (branch), ordinal (rating: low/med/high)
- **Binary:** yes/no, pass/fail
- **Date/Time:** join date, timestamp
- **Text:** feedback, comments

Exercise 1: Classify Variable Types

For each variable, write the type (numeric/categorical/binary/datetime/text):

1. Age
2. Program/Branch (CSE, ECE, . . .)
3. Attendance (%)
4. Join date
5. Feedback comment

Solution 1

- Age: numeric (integer)
- Program/Branch: categorical (nominal)
- Attendance (%): numeric (real)
- Join date: datetime
- Feedback: text (unstructured)

Data Formats

- **Structured:** fixed schema (tables)
  Examples: CSV, SQL tables

Data Formats

- **Structured:** fixed schema (tables)
  Examples: CSV, SQL tables
- **Semi-structured:** flexible schema with tags/keys
  Examples: JSON, XML

Data Formats

- **Structured:** fixed schema (tables)
  Examples: CSV, SQL tables
- **Semi-structured:** flexible schema with tags/keys
  Examples: JSON, XML
- **Unstructured:** free-form content
  Examples: text documents, images, audio

Structured Example (Table)

student_id	program	attendance_pct	cgpa
1001	CSE	92	8.2
1002	CSE	85	7.5
1003	ECE	105	8.9

Note: 105% attendance is an example of an out-of-range value.

Semi-structured Example (JSON)

```
{
  "student_id": 1001,
  "program": "CSE",
  "attendance_pct": 92,
  "courses": ["Math", "DSA", "Stats"]
}
```

Keys may vary from record to record (flexible schema).

Unstructured Example (Text/Log)

```
2026-02-08 10:02:11 INFO login user=1007 device=android city=Del
```

Useful information exists, but it requires parsing and feature extraction.

Common Data Sources

- Surveys and forms (Google Forms, LMS exports)

Common Data Sources

- Surveys and forms (Google Forms, LMS exports)
- Databases (student records, attendance systems)

Common Data Sources

- Surveys and forms (Google Forms, LMS exports)
- Databases (student records, attendance systems)
- Web and app logs (clickstream)

Common Data Sources

- Surveys and forms (Google Forms, LMS exports)
- Databases (student records, attendance systems)
- Web and app logs (clickstream)
- Sensors/IoT (temperature, GPS)

Common Data Sources

- Surveys and forms (Google Forms, LMS exports)
- Databases (student records, attendance systems)
- Web and app logs (clickstream)
- Sensors/IoT (temperature, GPS)
- Public datasets (government portals, research repositories)

Acquisition Methods

- Files: CSV/Excel export → `read_csv`, `read_excel`

Acquisition Methods

- Files: CSV/Excel export → `read_csv`, `read_excel`
- Database query: SQL → extract tables

Acquisition Methods

- Files: CSV/Excel export → `read_csv`, `read_excel`
- Database query: SQL → extract tables
- API calls: JSON responses → parse and store

Acquisition Methods

- Files: CSV/Excel export → `read_csv`, `read_excel`
- Database query: SQL → extract tables
- API calls: JSON responses → parse and store
- Manual entry: small datasets (careful with errors)

Exercise 2: Choose a Source

For each case, suggest a likely source (survey/database/log/API):

1. Daily attendance of students
2. Online learning platform clicks
3. Student feedback comments
4. Weather readings every minute

Solution 2

- Attendance: database export (or CSV from attendance system)
- Clicks: logs (web/app logs)
- Feedback: survey + text field (unstructured text)
- Weather readings: sensors/IoT or API

Why Cleaning Matters

- Models and statistics assume data is meaningful and consistent

Why Cleaning Matters

- Models and statistics assume data is meaningful and consistent
- “Garbage in, garbage out” → wrong conclusions

Why Cleaning Matters

- Models and statistics assume data is meaningful and consistent
- “Garbage in, garbage out” → wrong conclusions
- Cleaning improves: accuracy, fairness, and reproducibility

Common Data Quality Issues

- Missing values (blank, NaN, NULL)

Common Data Quality Issues

- Missing values (blank, NaN, NULL)
- Duplicates (same record repeated)

Common Data Quality Issues

- Missing values (blank, NaN, NULL)
- Duplicates (same record repeated)
- Inconsistent categories (`cse`, `CSE`, ` CSE`)

Common Data Quality Issues

- Missing values (blank, NaN, NULL)
- Duplicates (same record repeated)
- Inconsistent categories (`cse`, `CSE`, ` CSE`)
- Out-of-range values (attendance 105%, CGPA 12)

Common Data Quality Issues

- Missing values (blank, NaN, NULL)
- Duplicates (same record repeated)
- Inconsistent categories (`cse`, `CSE`, ` CSE`)
- Out-of-range values (attendance 105%, CGPA 12)
- Wrong data type ("nine" instead of 9.0)

Handling Missing Values (Basic Options)

- **Drop:** remove rows/columns (only if few missing and safe)

Handling Missing Values (Basic Options)

- **Drop:** remove rows/columns (only if few missing and safe)
- **Impute:** fill with mean/median/mode (simple baseline)

Handling Missing Values (Basic Options)

- **Drop:** remove rows/columns (only if few missing and safe)
- **Impute:** fill with mean/median/mode (simple baseline)
- **Domain rule:** fill with a meaningful default (carefully)

Handling Missing Values (Basic Options)

- **Drop:** remove rows/columns (only if few missing and safe)
- **Impute:** fill with mean/median/mode (simple baseline)
- **Domain rule:** fill with a meaningful default (carefully)
- **Flag:** create an indicator feature “was_missing”

Exercise 3: Missingness Decision

In a dataset of 20 students, the column cgpa has 2 missing values.

- What is the missingness percentage?
- Suggest one reasonable action for this column.

Solution 3

- Missingness $= 2/20 \times 100\% = 10\%$
- Action: impute using **median** CGPA (robust) and optionally add a flag

Outliers (Basic Idea)

An outlier is a value that is unusually far from typical values.

- Outliers can be **errors** (wrong entry) or **real extremes**

Outliers (Basic Idea)

An outlier is a value that is unusually far from typical values.

- Outliers can be **errors** (wrong entry) or **real extremes**
- They can strongly affect mean, variance, and some models

Outliers (Basic Idea)

An outlier is a value that is unusually far from typical values.

- Outliers can be **errors** (wrong entry) or **real extremes**
- They can strongly affect mean, variance, and some models
- Use rules like IQR fences as a **screening** step

IQR Rule (Fences)

$$\text{Lower fence} = Q_1 - 1.5 \times \mathrm{IQR}, \quad \text{Upper fence} = Q_3 + 1.5 \times \mathrm{IQR}$$

$$\mathrm{IQR} = Q_3 - Q_1$$

Values outside fences are *possible* outliers.

Exercise 4: IQR Outlier Check

Attendance (%): 70, 75, 80, 85, 90, 95, 150

Task: Compute Q_1, Q_3, IQR, fences, and decide if 150 is an outlier.

Solution 4

Sorted data: 70, 75, 80, 85, 90, 95, 150 (n=7). Median = 85.
Lower half: 70, 75, 80 $\Rightarrow Q_1 = 75$
Upper half: 90, 95, 150 $\Rightarrow Q_3 = 95$
IQR $= 95 - 75 = 20$
Fences: $75 - 30 = 45$ and $95 + 30 = 125$
Conclusion: $150 > 125 \Rightarrow$ outlier (by IQR rule).

Cleaning Checklist (Fast)

- Check shape, column names, and data types

Cleaning Checklist (Fast)

- Check shape, column names, and data types
- Check missingness and duplicates

Cleaning Checklist (Fast)

- Check shape, column names, and data types
- Check missingness and duplicates
- Standardize categories (trim whitespace, normalize case)

Cleaning Checklist (Fast)

- Check shape, column names, and data types
- Check missingness and duplicates
- Standardize categories (trim whitespace, normalize case)
- Check ranges and impossible values

Cleaning Checklist (Fast)

- Check shape, column names, and data types
- Check missingness and duplicates
- Standardize categories (trim whitespace, normalize case)
- Check ranges and impossible values
- Save a cleaned version (do not overwrite raw file)

Mini Demo (Python)

Run from the lecture folder:

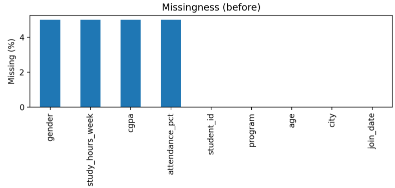
```
python demo/cleaning_demo.py
```

Outputs:

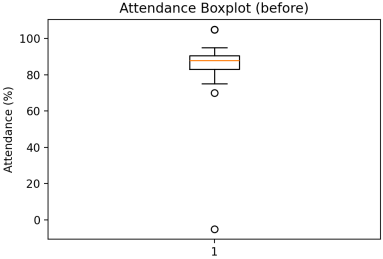
- `data/students_clean.csv`
- plots in `images/` (missingness and outlier visual)

Demo Output (Example)

Missingness



Attendance Outliers



Summary

- Data types and formats determine how we store and process data

Exit question: In one sentence, why can “attendance 105%” be dangerous in analysis?

Summary

- Data types and formats determine how we store and process data
- Different sources require different acquisition and validation steps

Exit question: In one sentence, why can “attendance 105%” be dangerous in analysis?

Summary

- Data types and formats determine how we store and process data
- Different sources require different acquisition and validation steps
- Cleaning deals with missing values, duplicates, inconsistencies, and outliers

Exit question: In one sentence, why can "attendance 105%" be dangerous in analysis?

Summary

- Data types and formats determine how we store and process data
- Different sources require different acquisition and validation steps
- Cleaning deals with missing values, duplicates, inconsistencies, and outliers
- Always save a cleaned dataset and document the rules you applied

Exit question: In one sentence, why can “attendance 105%” be dangerous in analysis?