Briyani

Data Curation

Collect

Pull raw sources into one staging area

Clean

De-duplicate, validate and standardise

Curate

Label, enrich and publish trusted sets

Round 1 — you just collected data!

You looked at the biryani picture and called out what you saw:

Rice	Chicken	Onion	Spices
Plate	Egg	Coriander	Mutton?

Each answer is a raw data point — that was raw Data Collection: unstructured, unlabelled, and everyone noticed different things!

But raw answers are messy!

What you told me

- “chicken… or is it mutton?”
- “onion” … and “Onion” again
- “plate”
- “biryani, briyani, biriyani”

Why that’s a problem

- Unsure labels — nobody can verify from the picture alone
- Duplicates — same thing counted twice
- Not part of the dish at all
- Same word, different spellings

Raw data can’t be used as it is — it needs curation.

Round 2 — now recreate this biryani!

Suddenly the picture isn't enough. What else would you need?

- The recipe — steps, quantities, order?
- Where did the ingredients come from?
- Who cooked it — and when?
- Is the chicken fresh or spoiled?
- How do we store the leftovers?
- Can everyone eat it?

Everything you just asked for IS data curation — let's look at each one.

Step 1 — Clean: wash the rice, pick out the stones

KEEP

Rice • Chicken • Egg

Onion • Spices • Coriander

The real ingredients of the dish

REMOVE / FIX

Plate — not an ingredient

Second “onion” — duplicate

“Briyani” — fix the spelling

Cleaning = fixing errors, removing duplicates and things that don’t belong.

Step 2 — Organise: sort your ingredients

The base

- Basmati rice

The protein

- Chicken
- Egg

Flavour & garnish

- Spices
- Fried onion
- Coriander

Organising = grouping data into categories so it's easy to find and use.

Step 3 — Label: name it right

Vague answer		Clear label
“meat”	→	**Chicken — leg piece**
“onion”	→	**Fried onion — garnish**
“white grains”	→	**Basmati rice — the base**

Labelling (annotation) = giving every data point a clear, precise name.

Step 4 — Check and enrich: taste it!

- **Is it really chicken?** — ask the cook (verify with the source)
- **How much rice?** — add the quantities (fill in missing details)
- **Is the chicken fresh or spoiled?** — a perfect recipe with bad ingredients gives a bad result — just like clean analysis on bad data
- **Add helpful notes** — "spicy", "serves 4" (extra context makes data richer)

Validation = making sure data is correct · Enrichment = adding useful detail.

Step 5 — Document: every biryani has a story

The question about the biryani	The data curation idea
The recipe — steps, quantities, order	**Metadata — data about the data**
Which rice? Which farm? Which shop?	**Provenance — where the data came from**
Ambur or Hyderabadi? By whom, and when?	**Version and history — which one is this?**

Documentation = recording the data's story so others can verify and trust it.

Step 6 — Preserve: store it, and know when to bin it

BIRYANI RECIPE — v1.0

Basmati rice — 2 cups

Chicken (leg piece) — 500 g

Fried onion — garnish

Spices — 2 tbsp · serves 4

Storage: fridge · eat within 2 days

Checked by the cook — August 2026

Why write it down — and store it well?

- Stored safely — the recipe isn't lost; anyone can recreate the dish
- Fridge = cold storage · reheating = format migration
- Kept too long? Throw it away — that's a retention policy!

Preservation = storing, sharing and managing data over its whole life.

Step 7 — Govern: can everyone eat it?

In the kitchen	In the data world
Allergies? Veg or non-veg? Halal?	**Access and consent — not all data is for everyone**
The cook decides what goes in	**The data steward — someone is responsible**
Hygiene rules of the kitchen	**Compliance — follow the policies and laws**

Governance = deciding who may use the data — and who answers for it.

Two masala racks — which kitchen would you trust?

Same contents, different curation — labels are metadata, expiry dates are lifecycle management.

Round 3 — close the loop

The photo is the data. The recipe, source, date, cook and storage instructions are the curation.

Without them the biryani can't be trusted, reproduced or reused — and neither can a dataset.

What is Data Curation?

Data curation is the process of **organizing, cleaning, enriching** and **managing** data so it is **accurate, reliable** and **useful** over its entire lifecycle.

1. COLLECT

Gather data from various sources. (Raw ingredients)

2. CLEAN

Remove errors, duplicates and inconsistencies. (Wash & clean)

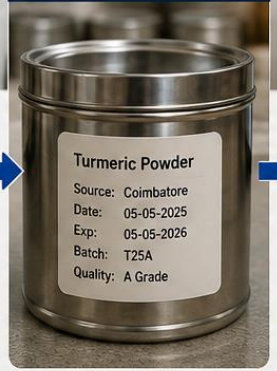
3. ORGANIZE

Structure and store data in a consistent, easy-to-find way. (Organize & store)

4. ENRICH

Add context and metadata to bring more meaning. (Add masala!)

5. MANAGE



Maintain quality, track lifecycle and ensure data is up-to-date. (Store safely & update)

6. USE & SHARE

Deliver trusted data that leads to better decisions. (Cook & serve!)

In short:

Raw Data → Curation → Trusted Data = Better Insights Better Decisions

Data Curation is...

- More than just cleaning data
- A lifecycle process
- The bridge between data and decision-making
- The foundation of reliable analytics and AI

Good data is not by chance, it's by curation.

Challenge — find that photo in 30 seconds!

Everyone, open your phone gallery. Find one specific photo — your school farewell, or your first day at IIT Tirupati. **You have 30 seconds!**

Found it in seconds?

- Albums — ‘Farewell 2024’
- Search by date, place or face
- **Your gallery is curated**

Still scrolling…?

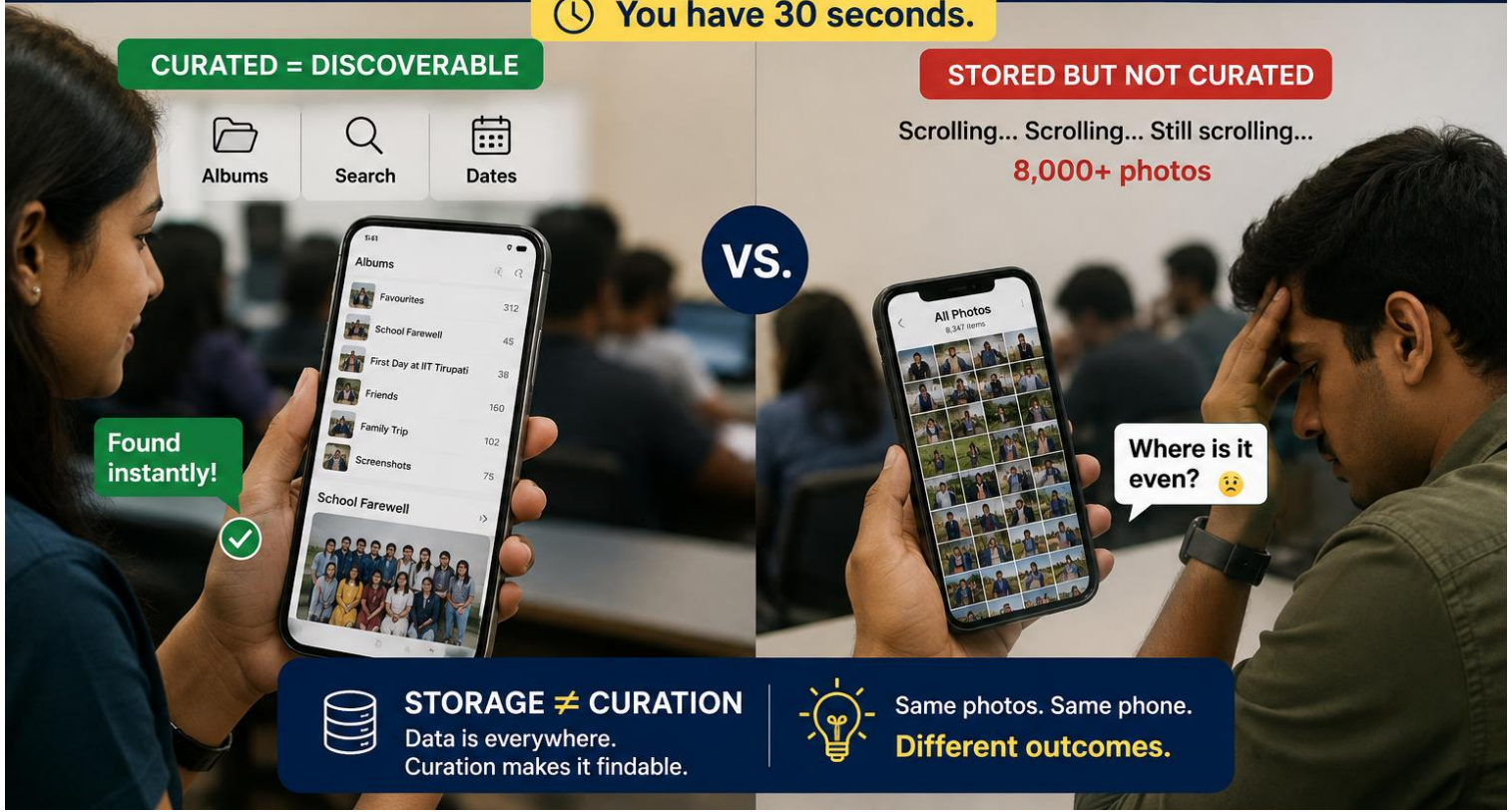
- 8,000 photos, one endless scroll
- No albums, no tags, no order
- **Stored — but not curated**

Storage ≠ curation — and you just discovered it on your own phone.

THE PHONE GALLERY CHALLENGE

Everyone open your gallery. Find one specific photo —
your school farewell / first day at IIT Tirupati.

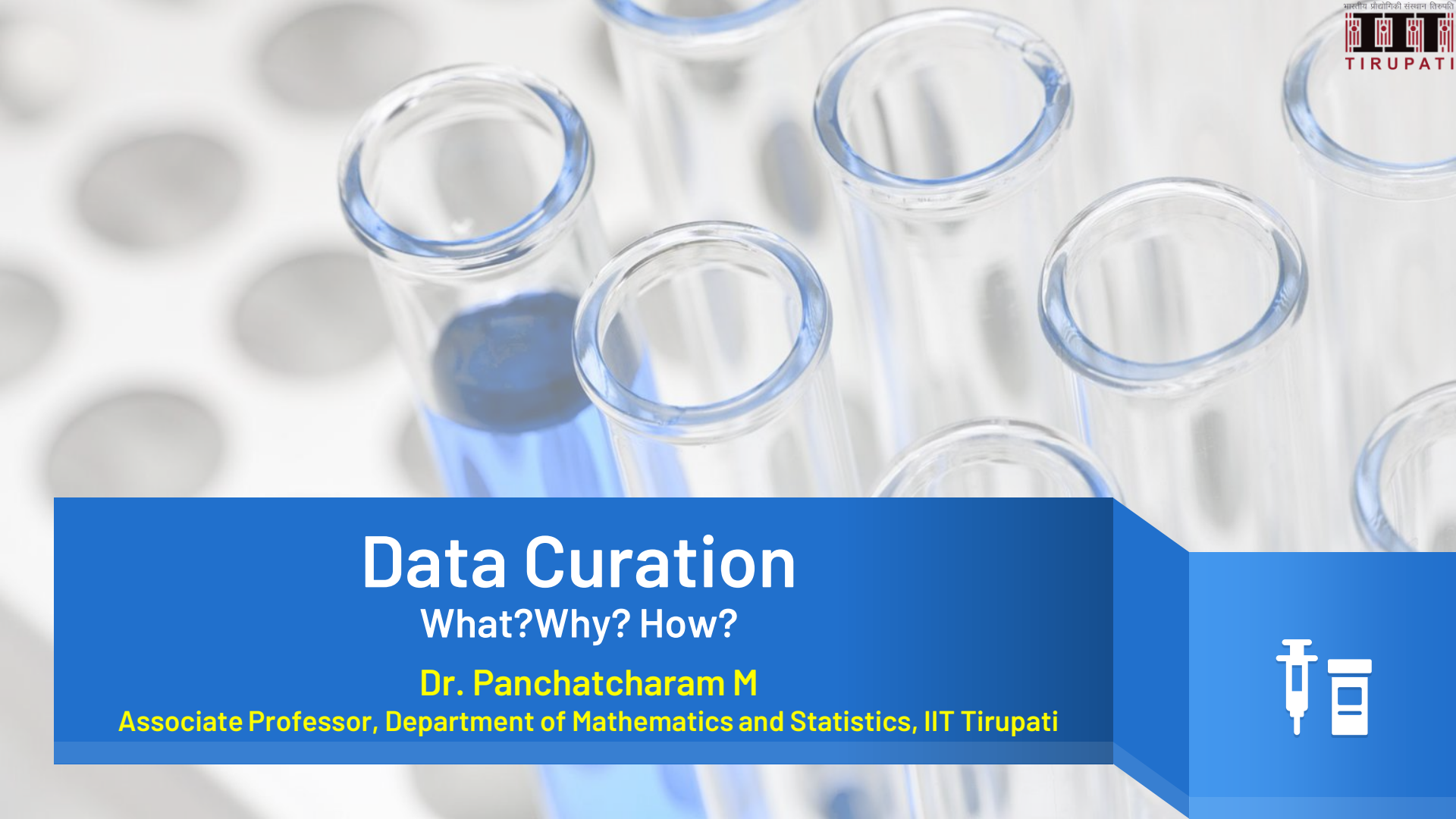
You have 30 seconds.



STORAGE ≠ CURATION
Data is everywhere.
Curation makes it findable.

Same photos. Same phone.
Different outcomes.

भारतीय प्रौद्योगिकी संस्थान तिरुपति
TIRUPATI



Data Curation

What?Why? How?

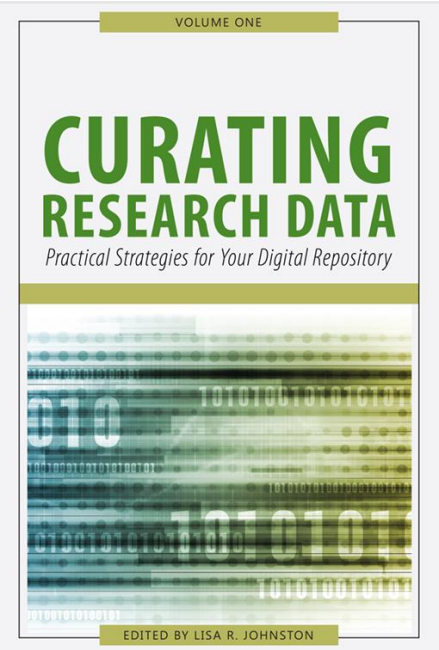
Dr. Panchatcharam M

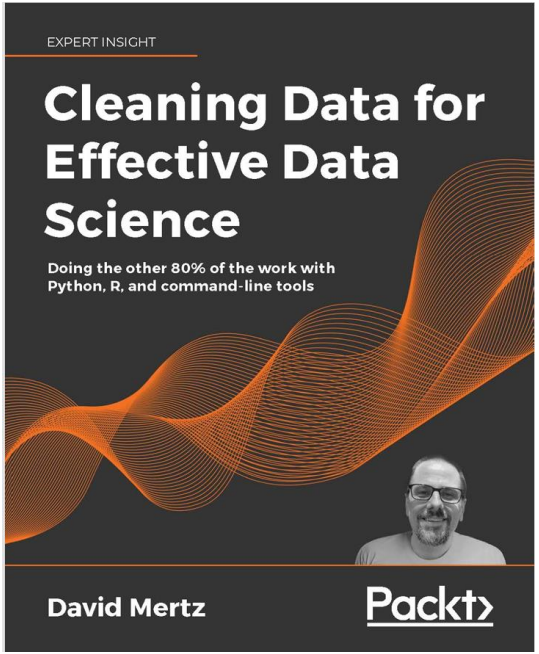
Associate Professor, Department of Mathematics and Statistics, IIT Tirupati

References

- Digital Curation Centre (DCC) Curation Lifecycle Model
- Wilkinson et al. (2016), "The FAIR Guiding Principles for scientific data management and stewardship", *Scientific Data*
- DAMA-DMBOK (Data Management Body of Knowledge) — chapters on governance and quality
- Digital Personal Data Protection Act, 2023 (India) — official text for the compliance section
- Borgman, *Big Data, Little Data, No Data* (MIT Press) — for motivation and selection/appraisal discussion

References


VOLUME ONE
CURATING
RESEARCH DATA
Practical Strategies for Your Digital Repository
EDITED BY LISA R. JOHNSTON


EXPERT INSIGHT
Cleaning Data for Effective Data Science
Doing the other 80% of the work with Python, R, and command-line tools
David Mertz
Packt

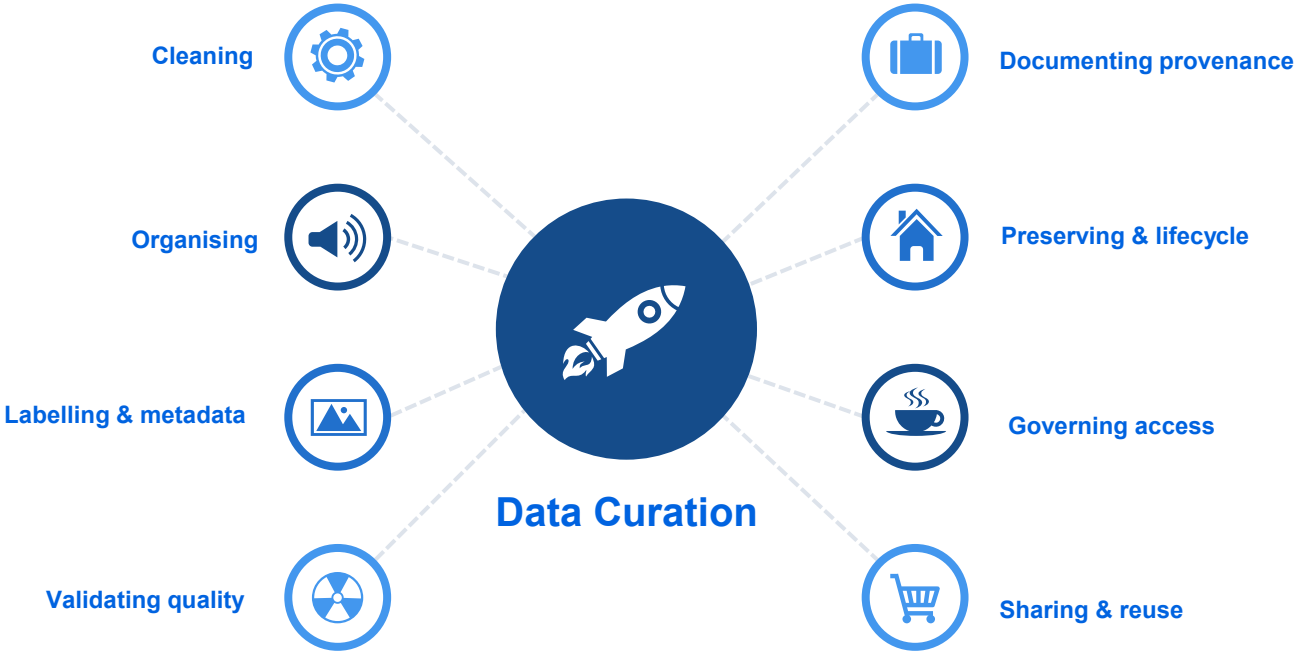

Mapping the World of Data Problems
Bad Data
Handbook
O'REILLY
Q. Ethan McCallum

Data Curation

What is Data Curation?

1

What is Data Curation?



Why bother with curation?

Motivation

Act 1 — congratulations, your paper is published!

Journal of Important Results

A Landmark Finding in Our Field

A. Postdoc, B. Student, C. Advisor

Vol. 42 · 2023 · pp. 1–12

ACCEPTED — 2023

The results are exciting. The figures look great. The reviews are glowing.

Everyone celebrates — and everyone moves on.

The postdoc graduates. The student who ran the analysis takes a job in industry.

Fast-forward two years…

Act 2 — two years later, an email arrives

From: Dr. R. Reviewer <collaborator@other-uni.edu>
To: C. Advisor
Subject: Request — dataset and processing steps for your 2023 paper

Dear colleague,

We loved your 2023 paper and want to build on it. Could you share the dataset behind Figure 3, and the steps you used to remove outliers, so we can reproduce the result?

Many thanks!

A perfectly normal request — science runs on it. So… where is the data?

Act 3 — where is the data?

`final_data_v3_REALLY_final.xlsx`

…found only on the laptop of a student who left last year

No units

Is temperature in °C or °F? Rainfall in mm or inches?

No column definitions

What does “col_7_adj_v2” even mean?

No outlier log

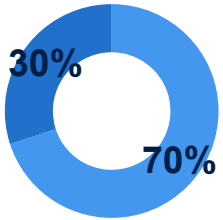
Which points were removed — and why?

The data exists — but it can’t be understood, trusted or reused. It is effectively lost.

The industry price tag

Where a data project's time really goes

- Preparing data (finding, cleaning, organising)
- Actual analysis and insight



60–80%

of a typical project's time is spent preparing data — not analysing it

Sources: Dasu & Johnson (2003); CrowdFlower survey (2016)

Better-curated data = less time cleaning, more time discovering.

The scientific price tag — the reproducibility crisis

70%

of researchers surveyed had tried and failed to reproduce another group's experiment

> 50%

had failed to reproduce their OWN experiment

Source: Baker, M. (2016), Nature 533, 452–454 — survey of 1,576 researchers

A major culprit: data and processing steps that were never curated — if no one can rerun it, no one can trust it.

What poor curation costs

Time
Experiments redone, analyses rebuilt from scratch — months lost.

Money
Salaries and instrument hours spent recreating what already existed.

Trust
Reviewers, collaborators and funders lose confidence in the group.

Careers
Corrections and retractions follow authors for years.

None of these are analysis problems — every one is a curation problem.

Rewind — what the curated version looks like

The group's data folder

- Data dictionary — every column defined, with units
- Versioned files — v1.0, v1.1, with a change log
- Outlier log — what was removed, and why
- Shared repository — not one person's laptop

Two years later…

The reviewer gets everything in 10 minutes.

Figure 3 reproduces exactly.

The paper — and the group — are trusted.

A little curation every week beats two years of regret.

"

Curation isn't overhead — it's insurance.

You are always collaborating with at least one person: your future self.

Next: how we curate — step by step.

Definition and scope

What data curation is — and what it isn't

2

Data curation, formally

Data curation is the **active and ongoing** management of data through its **lifecycle** — so that it stays **discoverable, understandable, trustworthy and reusable**.

Discoverable	Understandable	Trustworthy	Reusable

Keep this definition in mind — the whole lecture lives inside it.

Four promises to a future user

Discoverable

Can a stranger find it? Catalogued, indexed, searchable — like albums in your phone gallery.

Understandable

Can they make sense of it? Units, column definitions, context — the label on the jar.

Trustworthy

Can they believe it? Provenance, quality checks, an outlier log — the expiry date.

Reusable

Can they build on it? Open formats, licences, documentation — the recipe card.

If any one promise is broken, the data's value collapses for everyone who comes after you.

“Active and ongoing” — not a one-off cleanup

Even perfectly curated data decays. Left alone…

Links rot

The cloud folder moves, the URL dies, the drive is wiped.

Formats age

Nobody can open that 2009 file format any more.

People leave

The student who knew what ‘col_7b’ meant is gone.

So curation loops: check → refresh → re-document → check again — for as long as the data matters.

Curation is a habit, not a spring-clean — “active and ongoing” is in the definition for a reason.

Three terms people mix up

Each level keeps everything below it — and adds a new promise.

Data storage
Keeping bytes safe
disks · cloud · backups

Data management
Keeping operations running
databases · access · backups

Data curation
Keeping data reusable
+ selection · metadata · docs · QA · preservation

Storage is necessary. Management is necessary. Only curation makes data reusable.

Data storage: keeping bytes safe

What it is

- Disks, cloud buckets, tapes
- Copies and backups of the bytes
- Keeps files from being lost

In the kitchen…

The fridge.

The biryani is kept cold and safe — but nothing tells you which box it's in, how old it is, or whether it's still good.

Necessary — but not sufficient. Stored data can still be unfindable, unreadable and untrustworthy.

Data management

What it is

- Databases and file systems that organise records
- Backups and recovery when things fail
- Access control — who can read or write

Operational, day-to-day handling.

In the kitchen…

- Shelves and containers in known places
- Spare gas cylinder for when one runs out
- Only the cooks may enter the store room

A well-run kitchen — but still no recipe card.

Management keeps data running day to day — still not enough to make it reusable.

Data curation: what it adds

On top of storage and management, curation adds five deliberate activities:

Selection
What is worth keeping? Not everything is.

Enrichment
Metadata: units, labels, context added.

Documentation
READMEs, data dictionaries, method notes.

Quality assurance
Checks, validation, outlier logs.

Preservation
Formats and archives that outlive laptops.

All five exist for one purpose: so the data can be reused — by someone who wasn't there.

Storage vs management vs curation — side by side

	Storage	Management	Curation
Core job	Keep bytes safe	Keep operations running	Make reuse possible
Typical work	Disks, cloud, file backups	Databases, access control, recovery	Metadata, documentation, QA, archiving
Biryani analogy	The fridge	The well-run kitchen	Labelled rack + recipe card
Enough on its own?	No	No	This is the goal

"

Curation is about future users — and the first future user is you.

The five-year stranger test: will someone who has never met you understand this data in five years?

If yes — it's curated.

How we curate

Eight steps to rescue one messy file

3

Our patient: the file from the horror story

ADMITTED FOR CURATION

`final_data_v3_REALLY_final.xlsx`

- No units, no column definitions
- Mixed date formats, stray “N/A” and blanks
- Outliers removed — nobody remembers how
- Lives on one laptop, no backup

THE TREATMENT PLAN — 8 STEPS

1. **Plan**
2. **Select**
3. **Clean**
4. **Validate**
5. **Annotate**
6. **Version**
7. **Preserve**
8. **Publish**

By step 8, a stranger can find, understand, trust and reuse this data.

Step 1 — Plan: decide before you collect

WHAT YOU DO

- Decide before collecting: what data, what format, how much
- Assign responsibility — who curates it, where it lives
- Write it down as a short Data Management Plan (DMP)

ON OUR FILE

A one-page DMP would have named a shared store, an owner, and a format —

not one laptop and 14 guesses at 'final'.

Ten minutes of planning beats two years of archaeology.

Step 2 — Select: keep what is worth curating

WHAT YOU DO

- Not everything is worth keeping — curation costs time and money
- Keep raw data and the scripts; derived tables can be regenerated
- Weigh the cost of curating against the value of reuse

ON OUR FILE

Keep: the raw readings and the analysis script.

Drop: 14 near-identical 'final' copies.

Curate the few things that matter — not everything that exists.

Step 3 — Clean: one meaning per value

WHAT YOU DO

- One convention for missing values — not ‘N/A’, ‘na’ and blanks
- Find and remove duplicate records
- Standardise encodings, units and date formats (UTF-8, ISO dates)

ON OUR FILE

37 blanks marked as missing;
12 duplicate rows dropped;

every date → YYYY-MM-DD.

Cleaning makes the data say what it actually means.

Step 4 — Validate: turn ‘probably fine’ into ‘checked’

WHAT YOU DO

- Range checks — can a temperature really be 999°?
- Type checks — numbers are numbers, dates are dates
- Cross-checks — against a second instrument or trusted source

ON OUR FILE

The rule ‘temperature between −10 and 55 °C’ flags 3 impossible readings — sensor glitches, not science.

Each flag is logged, not silently deleted.

Validation is how ‘probably fine’ becomes ‘checked’.

Step 5 — Annotate: write the letter to a stranger

WHAT YOU DO

- Data dictionary — every column: name, meaning, type, units
- README — who collected it, what, when, where, how
- Codebook — what coded values mean (1 = pass, 2 = fail…)

ON OUR FILE

Cryptic column ‘t2m’ becomes: ‘air temperature at 2 m height, °C, hourly mean’.

Metadata = data about data.

Annotation is the letter you write to a stranger — and the stranger is future you.

Step 6 — Version & provenance: never overwrite raw

WHAT YOU DO

- Raw data is read-only — cleaning creates v1.1, it never overwrites
- Semantic versions: v1.1 = corrections, v2.0 = schema change
- Record lineage: source → transformation → output. Scripts are the provenance — Excel clicks are not

ON OUR FILE

raw/ folder frozen and checksummed. The outlier rule is now one script line — anyone can rerun it and get Figure 3.

No more REALLY_final — just v1.0, v1.1, v2.0 with a change log.

If you can’t say where a number came from, it isn’t evidence.

Step 7 — Preserve: planned survival

WHAT YOU DO

- Durable open formats: CSV, Parquet, HDF5/NetCDF — not a proprietary format that needs one vendor's licence
- Fixity: checksums (SHA-256) catch silent corruption
- 3-2-1 rule: 3 copies, 2 kinds of media, 1 offsite

ON OUR FILE

The .xlsx becomes data.csv + dictionary.csv, each with a checksum, held in the repo, the institute server and cloud storage.

Formats will be migrated as standards age — that's planned, too.

Preservation is planned survival — not luck.

Step 8 — Publish: give it an address

WHAT YOU DO

- Deposit in a repository — Zenodo, IEEE DataPort, data.gov.in, or your institute's archive
- Get a DOI — a permanent, citable address
- Attach a licence (CC-BY, CC0) so reuse is legal, not just possible

ON OUR FILE

doi:10.5281/zenodo.1234567 — data, dictionary, README and scripts, CC-BY 4.0.

The reviewer's email is now answered with a single link — in ten minutes, not two years.

Curated: findable, understandable, checked, versioned, preserved — and citable.

Core activities

Six things curators actually do

4

The six core activities

1. Selection & appraisal
What is worth keeping? Cost of curation vs value of reuse.

2. Cleaning & validation
Missing values, duplicates, encodings, unit errors.

3. Annotation & metadata
Data dictionaries, codebooks, README — data about data.

4. Provenance & versioning
What transformations produced this, from what sources?

5. Preservation
Durable open formats, checksums, format migration.

6. Access & publication
Repositories, DOIs, licences — making reuse possible.

You already did all six — that was Steps 1–8. Now the professional names.

Selection & appraisal

THE TRADE-OFF

- Curation costs real money and labour — every dataset kept must earn its keep
- Value of reuse: who might need this, and how soon?
- Keeping everything = curating nothing well

THE TWO-QUESTION TEST

1. Could this be recreated? If yes — keep the recipe, not the dish.
2. Would anyone pay the cost of reuse? If no one — why curate it?

Appraisal = spending curation effort where reuse value justifies it.

Raw vs derived — what do you actually keep?

RAW DATA — KEEP, READ-ONLY

- Cannot be recreated — the experiment happened once
- Instrument output, survey responses, sensor logs
- One immutable copy, checksummed

DERIVED DATA — KEEP THE RECIPE

- Cleaned tables, aggregates, model outputs
- Can be regenerated from raw + scripts
- Curate the script; regenerate the rest on demand

Raw data is irreplaceable — derived data is reproducible. Spend accordingly.

Cleaning & validation — the four classic diseases

Missing values
Blanks, 'N/A', 'na', −999 all meaning 'unknown'

Fix: Pick ONE convention and document it

Duplicates
Same record entered twice, or merged from two files

Fix: De-duplicate on a declared key

Inconsistent encodings
UTF-8 vs Latin-1: 'Tirupati' becomes 'TirupatiÂ'

Fix: Standardise to UTF-8 everywhere

Unit errors
kg vs lb, °C vs °F — the Mars orbiter died of this

Fix: state units; validate ranges

Cleaning fixes the values; validation proves they can be trusted.

Annotation & metadata — data about data

Data dictionary

One row per column: name, meaning, type, units, allowed values

't2m = air temperature at 2 m, °C, hourly mean'

Codebook

What coded values mean, and how they were assigned

'1 = urban, 2 = rural — assigned from 2021 census'

README

The letter to a stranger: who collected what, when, where, how

'Weather station IIT Tirupati roof, Jan–Dec 2025, operator KR'

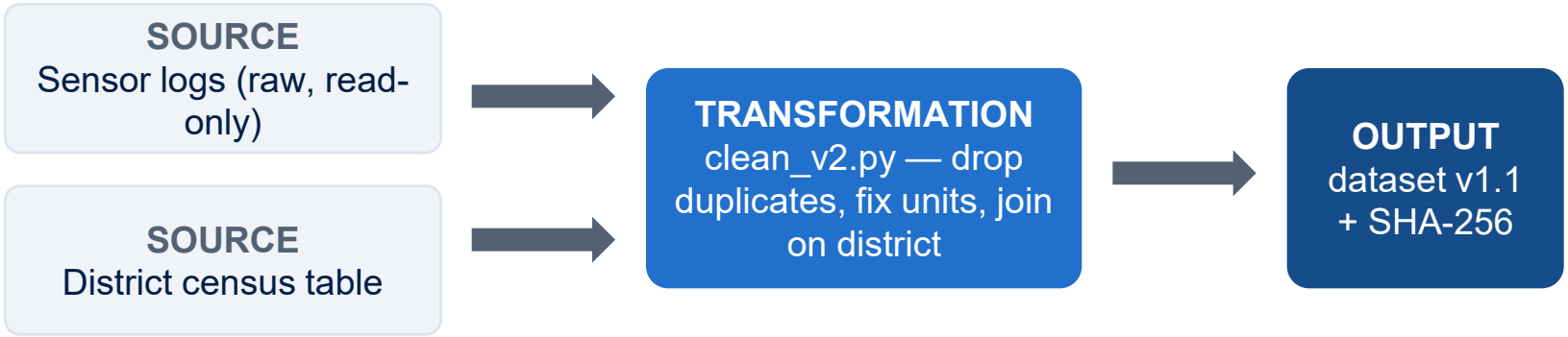
If the data is the biryani, metadata is the recipe card taped to the lid.

A data dictionary in practice

Column	Meaning	Type / units	Allowed values
station_id	Weather station code	text	TPT01–TPT04
ts	Reading timestamp (IST)	ISO 8601 datetime	2025-01-01 onwards
t2m	Air temperature at 2 m	number, °C	−10 to 55
rh	Relative humidity	number, %	0 to 100

Five rows of table save five weeks of email — write it while you still remember.

Provenance & versioning — the lineage of a dataset



v1.0 → v1.1 for corrections · v2.0 for schema changes · raw is never overwritten

The script is the provenance — clicks in Excel leave no trail.

Preservation — formats that outlive their software

CHOOSE — durable & open

- CSV — simple tables, readable anywhere
- Parquet — large columnar analytics
- HDF5 / NetCDF — big scientific arrays

Any tool can open these in 20 years.

AVOID — proprietary & fragile

- Formats that need one vendor's software
- Untracked binary blobs — no diff, no check

Media dies too: floppy → CD → cloud. Plan the migration.

Checksums catch silent corruption; open formats survive their software.

Access & publication — making reuse possible

Repository

A managed home — Zenodo, IEEE DataPort, data.gov.in, or your institute's archive

Not a laptop, not a WhatsApp link

DOI

A persistent identifier — the citation address that never moves

Papers cite your data like they cite you

Licence

CC0 / CC-BY / ODbL — says what reusers may legally do

No licence = nobody can safely reuse it

Published, identified, licensed — that is when curation pays off for everyone.

Six activities, one habit

1 · Select

2 · Clean and validate

3 · Annotate

4 · Provenance and version

5 · Preserve

6 · Publish

Not a heroic one-off clean-up — a weekly habit alongside the research itself.

Next: how the world checks you did all six — the FAIR principles.

The FAIR principles

Four letters the world checks

5

FAIR — the four promises, made checkable

F	A	I	R
Findable	**Accessible**	**Interoperable**	**Reusable**
Someone who needs it can locate it — identifiers + rich metadata	Once found, it can be retrieved by standard, open protocols	It plugs into other data and tools — standard formats and vocabularies	A stranger can reuse it — licence + provenance

FAIR (Wilkinson et al., 2016) turns ‘well-curated’ into a checklist the world agrees on.

F — Findable: can anyone locate it?

F

Findable

- Persistent identifier — a DOI that never breaks, unlike a URL
- Rich metadata — title, creators, keywords, methods, dates
- Indexed — deposited where search engines and catalogues look (Zenodo, data.gov.in)

Test: can a stranger find it with a search — without emailing you?

Findable = the labelled jar on a public shelf — not a dabba in someone's drawer.

A — Accessible: a standard way in

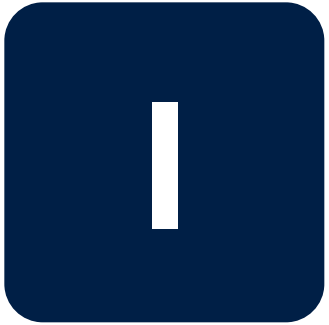
Accessible

- Retrievable by standard, open protocols — https, not a bespoke tool
- Metadata stays available even when the data itself is restricted
- Clear conditions: who may access, and how to request it

“As open as possible, as closed as necessary.”

Accessible ≠ public — it means there is a standard, documented way in.

I — Interoperable: speaks a shared language



Interoperable

- Standard formats — CSV, NetCDF, DICOM — not a home-made layout
- Controlled vocabularies and ontologies — ‘temp_c’ means one agreed thing
- Plays well with other datasets — join on shared codes (district, station ID)

Test: can someone else’s software read it without asking you?

Interoperable = your data plugs into tools and datasets you’ve never met.

R — Reusable: ready for the next project

Reusable

- A clear licence — CC-BY, CC0 — so reuse is legal, not a favour
- Provenance attached — where it came from, what was done to it
- Community standards — documented the way your field expects

Test: could a stranger build on this without emailing you?

Reusable is the goal — F, A and I exist so that R can happen.

FAIR ≠ open

FAIR, but access-controlled

- Patient MRI dataset: DOI, rich metadata, DICOM standard, clear licence
- Anyone can FIND and cite it
- Only approved researchers, after ethics review, can OPEN it

Open, but not FAIR

- A public zip on someone's homepage: no DOI, no metadata, cryptic columns
- Anyone can download it
- Nobody can understand, trust or cite it

"As open as possible, as closed as necessary" — sensitive data can be FAIR.

FAIR in India — closer to home than you think

NATIONAL REPOSITORIES

- data.gov.in — India's open government data portal: 600,000+ datasets
- Domain repositories — ICMR, ISRO, IMD publish curated datasets
- Institutional repositories at IITs and universities

FUNDER EXPECTATIONS

- DST, DBT, SERB grant guidelines increasingly ask for a Data Management Plan (DMP)
- A DMP answers: what data, what formats, who curates, where it lives, who may access it
- Journals too: many now require a data availability statement

FAIR is no longer optional — your future funder and journal will ask for the plan.

Quick check — is your pen drive FAIR?

Your project data sits on a pen drive in your bag. Is it FAIR? Which letters fail — and why?

Quick check — is your pen drive FAIR?

Your project data sits on a pen drive in your bag. Is it FAIR? Which letters fail — and why?

F

Findable?

No DOI, no metadata, not indexed — only you know it exists.

A

Accessible?

No protocol — access requires finding you and your bag.

I

Interoperable?

Maybe — depends on formats; .xlsx with merged cells says no.

R

Reusable?

No licence, README or provenance — no reuse.

Verdict: stored — but failing all four letters. Discuss: what single fix helps most?

Data Acquisition

Where data comes from, and how to capture it well

6

Sources of data

Primary · secondary · operational

The biryani test: primary vs secondary

PRIMARY — you cook it

Four hours in your own kitchen

You pick the ingredients, spice level and quantity → you control the instrument, the sample and the variables

It takes four hours → primary data is slow and expensive

Full control, high cost

VS

SECONDARY — Swiggy brings it

Thirty minutes to your door

Cheap and fast

You never saw their kitchen → you don't know the methodology

Their recipe, not yours → collected for someone else's purpose

Low cost, imperfect fit

The trade-off never changes:
control and cost, versus speed and fit.

One wrinkle: the restaurant cooks to your order

PRIMARY	STILL PRIMARY	SECONDARY
You cook it yourself Your question, your recipe, your kitchen. *Run your own survey or experiment*	**You order it "less spicy"** Cooked on your order, to your instructions — outsourced, but the purpose is yours. *Agency-run survey*	**Leftovers from the wedding** Cooked for someone else's reason; you just reuse it. Also stale, and unverifiable. *Published datasets*

The test is purpose, not effort: was it collected for your question?

Behind the counter: operational vs analytical

OPERATIONAL — tonight's order slips

Table 7 wants two biryanis and one raita

A by-product, not a research effort — like POS swipes and click logs.

About now: settle the bill and the table flips to “free”, overwriting the old value.

One row per event; speed beats depth.

ANALYTICAL — the month-end register

Which dish sells best on rainy Fridays?

The owner sits down with a copy of all those slips. Historical, append-only, aggregated.

Nobody's dinner depends on it — a slow query is inconvenient, not an outage.

Reads millions of rows, not one table.

Run that query on the live billing system at 8:30 pm Saturday and service crawls — which is why we extract into a warehouse.

Where does data come from? Three families

PRIMARY	SECONDARY	OPERATIONAL
You collect it yourself	**Someone else collected it**	**Your organisation generates it anyway**
Experiments · sensors & IoT · surveys · field measurements · clinical instruments	Open-data portals · published datasets · APIs · web scraping	System logs · transaction records · internal databases
You cook the biryani	*You order it in*	*The kitchen's daily register*

Most real projects mix all three — each family needs different curation care.

Primary — experiments and field measurements

WHAT IT LOOKS LIKE

- Lab experiments — controlled conditions, repeated runs
- Field measurements — river gauges, soil samples, traffic counts
- You design the collection — maximum control, maximum effort

CURATION ANGLE

Record instrument settings, calibration constants, date, location and operator at collection time — they cannot be reconstructed later.

Lab notebook = the first metadata.

Primary data: you control quality — and you carry the full curation burden.

Primary — sensors and IoT: data that never sleeps

WHAT IT LOOKS LIKE

- Weather stations, GPS trackers, smart meters, lab loggers
- Continuous streams — readings every second or minute
- Cheap to deploy, easy to drown in — volume grows while you sleep

CURATION ANGLE

Capture at source: sensor ID, calibration date, sampling rate, clock sync, units.

A reading without its timestamp convention is just a number.

Sensors are tireless — and tirelessly wrong if never recalibrated.

Primary — surveys and clinical instruments

WHAT IT LOOKS LIKE

- Questionnaires, interviews, clinical trials, medical instruments
- Humans on both sides — asking and answering
- Question wording and instrument calibration shape every value

CURATION ANGLE

Keep the questionnaire version, consent forms, instrument model and calibration certificate with the data.

“Q7 on form v2” means nothing without form v2.

People-data carries people-context — lose the form, lose the meaning.

Secondary — government open data portals

WHAT IT LOOKS LIKE

- data.gov.in — India's national open data platform
- Census, IMD weather, agricultural and economic statistics
- Free, official, already collected — someone else paid for it

CURATION ANGLE

Record the exact download date, portal URL, dataset version and licence — portals update and remove files.

"I got it from the portal" is not provenance.

Official ≠ analysis-ready — open data still needs your curation.

Secondary — published datasets: someone already curated

WHAT IT LOOKS LIKE

- Kaggle, UCI ML Repository — teaching & practice sets
- Zenodo, IEEE DataPort — research data with DOIs
- Benchmark datasets your field already trusts

Someone curated these once — that's why they're usable.

CURATION ANGLE

Record the version and download date — datasets get revised.

Check the licence before you build on it.

Read the codebook — don't guess what columns mean.

A published dataset is someone else's curation — cite it, version it, respect its licence.

Secondary — APIs: data on request

WHAT IT LOOKS LIKE

- REST endpoints returning JSON — weather, finance, transport
- Authentication keys, rate limits, pagination
- You write code once, fetch fresh data forever

Details of keys, limits and payloads -- next.

CURATION ANGLE

Log the query, endpoint and timestamp of every pull.

APIs change and disappear — archive the raw responses.

Same query tomorrow ≠ same answer.

An API is a moving target — archive what you fetched, when, and with what query.

Secondary — web scraping: the last resort

WHAT IT LOOKS LIKE

- Parsing HTML pages when no API or download exists
- Fragile — breaks whenever the site changes layout
- Legality & ethics: robots.txt, terms of service, server load

Full ethics discussion coming soon.

CURATION ANGLE

Save the raw HTML with a timestamp — pages vanish.

Record the scraper version that parsed each page.

Ask first: is there an API or a download instead?

Scrape politely and only when you must — and archive exactly what the page said, when.

Operational data — the organisation's daily exhaust

WHAT IT LOOKS LIKE

- Server and application logs — every click, every request
- Transactional databases — orders, payments, admissions
- Collected for operations, not for your analysis

Biryani: the kitchen's daily register — written for billing, not for research.

CURATION ANGLE

The schema was designed for operations — document what each field really means.

Watch for silent changes when systems are upgraded.

Privacy alert: logs are full of personal data.

Operational data was never designed for analysis — re-curate it before you trust it.

Cook it yourself, or order in? — the trade-off

PRIMARY — COOK IT YOURSELF

- Full control — you choose every ingredient
- Fits your question exactly (fitness-for-purpose)
- Slow and expensive — and you curate everything

Your biryani, your recipe, your responsibility.

SECONDARY — ORDER IT IN

- Fast and cheap — someone else cooked
- May not fit — their recipe, their choices
- You inherit their curation — good or bad

Check the kitchen before you eat: licence, metadata, provenance.

Control and fit (primary) vs cost and speed (secondary) — most projects blend both, so curate the joins.

Acquisition methods

Six ways data arrives — and their failure modes

7

Six ways data arrives

1 · Manual entry
Typed by a human — slowest, most error-prone

2 · Files
CSV, Excel, JSON, XML handed over or downloaded

3 · APIs
Programmatic requests to a service — JSON in, data out

4 · Web scraping
Parsing pages never meant to be data

5 · Streaming / sensors
Continuous readings, timestamps, packets

6 · Surveys / instruments
Questionnaires and calibrated devices

Each method has its own failure modes — know them before the data arrives.

Method 1 — manual entry: humans make typos

WHAT GOES WRONG

- Typos: 71.2 becomes 712 — a 10× error
- Column drift — value typed one cell over
- Silent unit mix-ups: kg vs lb, °C vs °F
- Fatigue — error rate climbs with hour of day

MITIGATE

- Validation rules at entry — ranges, types, pick-lists
- Double entry for critical fields — two people, compare

Cheapest fix: don't let the error in.

Manual entry is a data source with a built-in error generator — design the form, not just the sheet.

Method 2 — files: CSV, Excel, JSON, XML

THE WORKHORSE OF DATA EXCHANGE

- CSV — simple, universal, but no types or schema
- Excel — convenient, but formulas, merged cells, hidden sheets
- JSON / XML — nested structure, schema optional

Rule: ingest a copy, never edit the original file.

CAPTURE ON ARRIVAL

- Who sent it, when, which version
- Checksum the file immediately
- Record the encoding and delimiter you assumed

A file is a promise with no contract — verify structure before trusting content.

File traps — encoding, delimiters and locales

ENCODING ‘Tirupati’ arrives as ‘Tirupatâ€”’	**FIX** Convert to UTF-8 on arrival
DELIMITERS A comma inside “Reddy, A.” splits one column into two	**FIX** Quoted CSV or tab; check column counts
LOCALES 3,14 vs 3.14 · 02/03 = 2 March or 3 Feb?	**FIX** ISO dates: YYYY-MM-DD; fix decimals

Files lie quietly — decide encoding, delimiter and date format before the first byte arrives.

APIs — data through a formal doorway

HOW IT WORKS

- REST endpoints return JSON — weather, finance, social media
- Authentication keys identify you — keep them out of shared code
- Rate limits cap requests — plan the collection window
- Pagination — results arrive page by page; missing a page = silent gaps

CAPTURE NOW

Archive the raw JSON response, the query, the endpoint version and the date — the API will change under you.

Yesterday's query can return different data today.

APIs are convenient but unstable — archive raw responses as your immutable record.

API discipline — a reproducible pull

1. Script the pull — no clicking; the script is the provenance
2. Log query, endpoint, key owner (not the key!), timestamp
3. Save the raw JSON exactly as received — read-only
4. Parse into tables in a separate, versioned step

A pull you can re-run and defend — that is curated acquisition.

Web scraping — when there is no doorway

HOW IT WORKS

- Parse the HTML a website sends to your browser
- Fragile — a site redesign silently breaks your parser
- No schema, no guarantees — you infer structure yourself
- Use when no API or download exists — last resort, not first

CAPTURE NOW

Save the raw HTML with URL and timestamp before parsing — pages change and disappear.

The page you scraped today may not exist tomorrow.

Scraped data is a snapshot of a moving target — archive the page, not just the numbers.

Scraping — legal is not the same as polite

THREE CHECKS BEFORE YOU SCRAPE

- robots.txt — does the site say bots are welcome here?
- Terms of service — does scraping breach the contract?
- Server load — throttle requests; you share the site with real users

BEYOND LEGALITY

Public ≠ fair game: profiles scraped ‘legally’ may still expose people who never consented to research use.

Ask: would the people behind this data object?

Scrape like a guest: announce yourself, take little, and leave the kitchen as you found it.

Streaming and sensors — data with a heartbeat

DESIGN DECISIONS UP FRONT

- Sampling rate — every second? minute? hour?
- Timestamps — which timezone? UTC or IST?
- Clock sync — two sensors, two clocks, one truth (NTP)
- Buffering — what happens when the network drops?

CAPTURE NOW

Sensor ID, firmware version, calibration constants, location, units — none of these can be reconstructed next month.

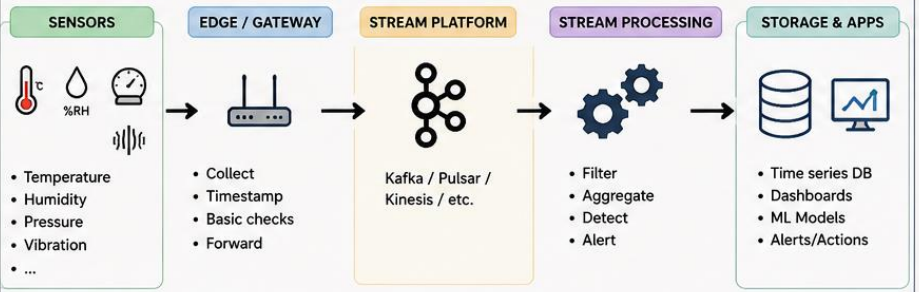
Weather station 'T-03, v2.1, ±0.2 °C, roof of Block B'.

A stream never pauses for you — decide rate, clocks and buffers before switching it on.

STREAMING AND SENSOR FAILURES: HOW AND ITS IMPACT

Sensors feed data → Streaming system processes it → Failures happen → Data quality and decisions suffer

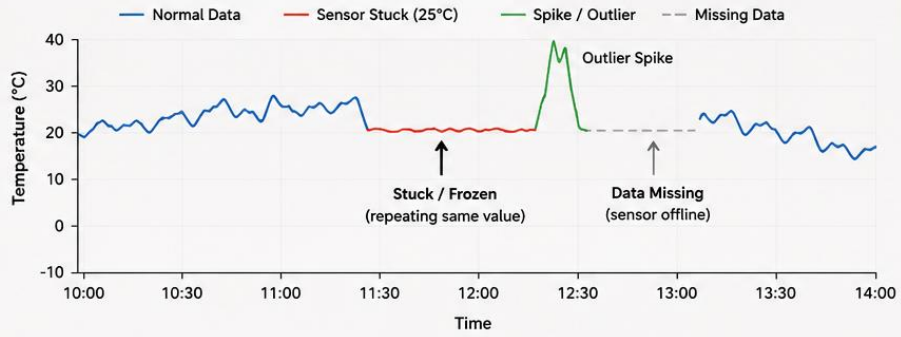
1 THE STREAMING PIPELINE



2 HOW FAILURES HAPPEN

Failure Type	How it Happens	Examples
Sensor Offline	Power loss, network issue, battery drain, hardware crash	Sensor stops sending data
Degraded / Noisy	Aging sensor, environmental interference, calibration drift	Values become noisy or inaccurate
Stuck / Frozen	Firmware hang, memory leak, buffer full	Same value repeats for too long
Spike / Outlier	Electrical noise, loose connection, glitch	Sudden unrealistic values
Timestamp / Order	Clock drift, network delay, reordering in transit	Late, early or out-of-order events
Data Loss	Packet loss, broker failure, retention misconfig	Missing events / gaps in the stream

3 EXAMPLE: TEMPERATURE SENSOR FAILURE



4 IMPACT ON SYSTEM AND BUSINESS

Wrong Insights	Bad data → wrong aggregates, trends and predictions	Impact: Incorrect dashboards and reports
Late / Missed Alerts	Failures hide real problems or create false alarms	Impact: Delayed response, higher risk
Poor Automation	Control systems act on bad data (e.g., HVAC, pumps, robots)	Impact: Inefficient operations or equipment damage
Model Degradation	ML models trained on bad data perform poorly	Impact: Lower accuracy, wrong predictions
Business Impact	Downtime, quality issues, customer dissatisfaction	Impact: Revenue loss, higher costs, reputation loss

5 HOW WE HANDLE FAILURES

Detect Early
- Heartbeats
- Range checks
- Anomaly detection

Validate & Clean
- Remove spikes
- Handle duplicates
- Time sync

Handle Missing Data
- Interpolation
- Last known value
- Gap alerts

Make System Resilient
- Redundant sensors
- Buffer + retry
- Graceful degradation

Monitor & Alert
- Health dashboards
- Smart alerts
- Escalation

Good streaming starts with good data. Detect, handle, and learn from failures.

REAL-WORLD EXAMPLE: WHEN BAD SENSOR DATA CONTROLS A PLANE

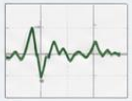
Boeing 737 MAX – MCAS and *Angle-of-Attack Sensor Data*

Ethiopian Airlines Flight 302 • 10 March 2019

1 THE DATA AND CONTROL FLOW

ANGLE-OF-ATTACK (AoA) SENSOR
Measures the aircraft's orientation relative to the airflow

→ **SENSOR DATA (Streaming)**
Continuous data sent to the flight computer



→ **MCAS SOFTWARE**
Monitors AoA. If AoA is high, MCAS activates.

→ **AUTOMATIC STABILIZER COMMAND**
MCAS commands the horizontal stabilizer to move nose-down.

→ **AIRCRAFT RESPONSE**
Nose of the aircraft moves down.

2 WHAT HAPPENED IN ETHIOPIAN AIRLINES FLIGHT 302?

Erroneous Angle-of-Attack (AoA) sensor input contributed to repeated activation of MCAS, which commanded nose-down stabilizer movement.

1. AoA sensor gave a very high (erroneous) reading.
2. MCAS interpreted it as a high AoA condition.
3. MCAS automatically commanded the stabilizer nose-down.
4. The nose of the aircraft pitched down.
5. MCAS reactivated multiple times.
6. The crew struggled to control the aircraft.

Outcome: Loss of control and a fatal crash.

3 THE FAILURE CHAIN

SENSOR FAILURE

AoA sensor provided erroneous data (possible faulty sensor).

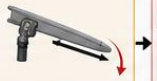
→ **BAD DATA**

Incorrect measurement entered the system.

→ **SOFTWARE INTERPRETATION**

MCAS treated the input as indicating a high angle of attack.

→ **AUTOMATED ACTION**



Nose-down stabilizer commands were triggered.

→ **PHYSICAL RESPONSE**

The aircraft responded to those commands.

→ **IMPACT**

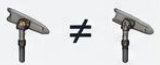
Loss of control and fatal crash.

4 KEY LESSON

In cyber-physical systems, data quality is not just an IT problem—
it can directly affect the physical world.

5 WHAT COULD HAVE PREVENTED IT?

Cross-check sensors



Compare multiple sensors. Detect disagreement.

Validate data

Use range checks, rate checks, and plausibility tests.

Graceful degradation

If data is unreliable, do not automate.

Better alerts to pilots

Clear, actionable alerts in real time.

Transparent automation

Pilots should know when and why automation acts.

6 TAKEAWAY FOR DATA SYSTEMS

- Sensors can fail. Data can be wrong.
- Always validate, cross-check, and monitor data streams.
- Design software to be robust to bad data.
- Automation must be explainable and overrideable.
- Because in the real world, a bad data point can cost lives.

> "Bad data doesn't just give you the wrong answer—
> It can make the wrong decision for you."

When streams break — gaps, drift and silent loss

Missing packets

Network drops silently — the file looks complete but hours are gone.

Log expected vs received; flag gaps loudly.

Clock drift

A sensor clock loses 2 s/day — events misalign across sensors.

Sync to NTP; log clock offsets.

Buffer overflow

Bursts overflow the buffer — oldest readings overwritten.

Size for worst case; alert on overflow.

Streams fail quietly — count what you expected, count what arrived, and log the difference.

Surveys and instruments — bias enters early

SURVEYS — QUESTION DESIGN

- Leading wording nudges answers
- Option order and scale direction bias responses
- Pilot the questionnaire; version every change

INSTRUMENTS — CALIBRATION

- Un-calibrated sensor = consistent, confident, wrong
- Record calibration date, constants, operator
- Re-check after transport, repair or firmware updates

A biased question and an un-calibrated probe make the same mistake — measuring something other than the truth.

The golden rule — capture it now or lose it forever

Some metadata can never be reconstructed afterwards. Write it down at the moment of collection:

Instrument settings
Gain, mode, firmware version

Calibration constants
Last calibration date + values

Date, time & location
With timezone and coordinates

Operator
Who collected it — ask them now

Units
°C or °F? mg or µg? Say it

Why now?
In six months, nobody will remember.

If it isn't captured at acquisition, it is gone — no amount of curation can invent it later.

Recap — method × what to capture

Method	Biggest risk	Capture at acquisition
Manual entry	Typos, drifting formats	Validation rules; who entered
Files	Encoding & locale traps	Checksum, encoding, source, date
APIs	Rate limits, changing schema	Raw JSON, query, key version
Scraping	Fragile HTML, ethics	Raw HTML, URL, timestamp, robots.txt
Streaming	Gaps, clock drift	Sampling rate, sync log, gap log
Surveys	Question bias, calibration	Questionnaire version, instrument ID

Sampling and bias

Five ways data lies before you even open it

8

Population vs sample — the spoonful test

POPULATION — the whole pot

- Everyone or everything you want conclusions about
- Usually too big, too slow or too costly to measure fully

The full handi of biryani.

SAMPLE — the spoonful you taste

- The part you actually measure
- Only works if the spoonful is representative — stir first!

One spoon from the top = all rice, no chicken — a bad sample.

Every dataset is a spoonful. The question is always: how was the spoon filled?

The sampling frame — the list you draw from

POPULATION — all farmers in Andhra Pradesh

FRAME — farmers with a registered mobile number

SAMPLE — the 400 who answered the call

Every arrow loses people — and rarely at random. The frame is never the population.

Before trusting a dataset, ask: who could never have appeared in it?

Five biases that arrive with the data

1 · Selection
Who got in was never random

2 · Survivorship
You only see what survived

3 · Non-response
The silent are different

4 · Measurement
The instrument tilts the answer

5 · Temporal drift
The world changed mid-collection

All five happen before analysis begins — they ride in with the data itself.

Bias 1 — selection: who got in was never random

WHAT IT IS

- The way cases enter the dataset favours some kinds over others
- Convenience samples: your friends, your ward, your campus
- The sample looks fine — the bias is in the door, not the data

EXAMPLE

A campus food survey run only in the hostel mess — day scholars never had a chance to answer.

Ask: who could never appear here?

Selection bias is invisible in the file — you must ask how the door was opened.

Bias 2 — survivorship: you only see what survived

WHAT IT IS

- Failures leave the dataset before you ever see it
- What remains looks stronger, healthier, luckier than reality
- The missing rows are the message

THE CLASSIC

WWII: armour the bullet holes on returning planes? No — armour where returners were NOT hit. Those hits downed the planes that never came back.

(Abraham Wald, 1943)

Ask what never made it into the file — dropped sensors, deleted rows, dead startups.

Bias 3 — non-response: the silent are different

WHAT IT IS

- The people who don't answer differ from those who do
- A 20% response rate is not a small sample — it's a different population
- Silence is data you don't have

EXAMPLE

Course feedback: the very happy and very angry reply; the busy middle never does. The average is a fiction.

Record the response rate — always.

Report who answered AND who didn't — the gap is part of the dataset.

Bias 4 — measurement: the instrument tilts the answer

WHAT IT IS

- The measuring process itself skews the values
- Uncalibrated sensors, leading survey questions, observer effects
- Systematic — it doesn't average out with more data

EXAMPLES

A thermometer mounted above hot asphalt reads +3 °C every single day.

"Don't you agree the mess food is good?" — the question answers itself.

More data does not fix a tilted instrument — calibrate and pre-test instead.

Bias 5 — temporal drift: the world moves on

WHAT IT IS

- The world changes while you collect — or after you stop
- Sensors age, populations shift, behaviour moves online
- Old training data quietly stops representing today

The question: when was this collected — and is that still true?

CLASSIC CASE

Sensor data collected only during working hours — the model has never seen a night or a Sunday.

Cover the full cycle, and stamp every record with its time.

Data is a photograph of a moment — label the moment, and re-shoot before the scene changes.

OLD TRAINING DATA QUIETLY STOPS REPRESENTING TODAY

*The question: **when was this collected — and is that still true?***

1. MODEL TRAINED ON OLD DATA (2019)

Learned Pattern
Monday, 9:00 AM
→ Heavy traffic

Predicted Travel Time
45 minutes

Model trained on historical data believes this pattern is normal.

TIME PASSES
(2019 → 2026)

- New 6-lane road built (2022–2023)
- Better infrastructure & connectivity
- Changed commuting behavior
- Traffic patterns have changed

2. TODAY'S REALITY (2026)

Actual Pattern Today
Monday, 9:00 AM
→ Smooth traffic

Actual Travel Time
15–20 minutes

The old model still predicts 45 minutes because it uses outdated data.

Wrong prediction.
Poor user experience.

3. WHAT HAPPENED?

Old Data (2019)
Single lane road, heavy traffic

→ **Model Trains on Old Data**
Learns old patterns

→ **World Changes (2020–2026)**
Road expanded to 6 lanes, new flyover, new developments

→ **Predictions Drift Over Time**
No longer matches reality

4. THE QUESTION

When was this collected — and is that still true today?

Data has a timestamp.
Relevance has an expiry.

5. TAKEAWAY

- Check when the training data was collected.
- Monitor for changes in the real world.
- Retrain or update the model with recent data.
- Data quality is not just about accuracy, it's about relevance over time.

REAL-WORLD EXAMPLE: GOOGLE MAPS TRAVEL-TIME PREDICTION

A model trained in 2019 on a single lane road predicted 45 minutes travel time on Monday 9:00 AM.

But after the road was expanded to 6 lanes and traffic patterns changed, the actual time is now **15–20 minutes**.

Old training data quietly stopped representing today.

2019 (Training Data)

2026 (Today)

- Same road.
- Same location.
- Different reality.

Update your data.
Update your models.
Make better decisions.

Case study — the hospital model that never met you

A diagnostic ML model is trained on records from one urban private hospital — mostly patients from a single region, age band and income group.

Selection
Only patients who could reach and afford this hospital

Non-response
Those who never sought care are invisible

Temporal
Records reflect that hospital's equipment and era

Deployed in a rural clinic, the model underperforms — on exactly the patients who need it most.

The model isn't 'wrong' — the sample never contained the people it now serves.

> No amount of downstream cleaning fixes a biased collection design.

You can scrub every cell and validate every unit — the people who were never sampled are still not there.

Design the collection; don't repair the collapse.

Documentation

Write it down while it’s happening

9

The collection protocol — your SOP

WHAT THE SOP RECORDS

- What is measured, with which instrument and settings
- How often, by whom, in what order
- Units, formats and the missing-value convention
- What to do when something goes wrong

WHY IT MATTERS

Three students collect rainfall for one project. Without an SOP: three formats, two units, and one argument.

With an SOP, the dataset looks like one careful person made it.

A one-page protocol, written before day one, outlives every team member.

One protocol, many hands

WITHOUT A PROTOCOL

Student A: “14/3, 23.5”

Student B: “Mar 14, 74.3 F”

Student C: “…I forgot to note the date”

Three private dialects — one useless merge.

WITH A PROTOCOL

Everyone: “2026-03-14, 23.5 °C”

Same instrument, same time of day, same missing-value convention.

Rows merge as if one careful person collected them all.

Comparability is designed, not cleaned into existence afterwards.

Raw data is sacred — look, don't touch

THE THREE RULES

- One immutable, read-only copy of raw data — from day one
- Every cleaning step writes a NEW file (v1.1, v1.2…)
- Scripts + logs record how each version was made

You can always regenerate clean from raw — never the reverse.

WHY SO STRICT?

The moment someone 'fixes' a value directly in the raw file, the original observation is gone — forever, silently.

Remember Act 3: no record of how outliers were removed.

Treat raw data like a film negative — print from it, never paint on it.

Name files so they explain themselves

THE HAUNTED FOLDER

```
final_data.xlsx
final_data_v3_REALLY_final.xlsx
new data (copy) (2).xlsx
```

Which one goes in the paper? Nobody knows.

SELF-EXPLAINING NAMES

```
2026-03-14_weather_tpt_raw_v01.csv
2026-03-14_weather_tpt_clean_v02.csv
```

Date · what · where · stage · version — sorts itself, explains itself, no spaces.

A file name is metadata you get for free — spend it wisely, from day one.

One folder structure, from day one

```
project/
└ raw/          ← originals only
└ scripts/      ← all transformations
└ clean/        ← versioned outputs
└ docs/         ← README, SOP
```

Anyone can navigate it on day one — including you, two years later.

THE HABITS

- Set the structure before the first byte arrives
- Same tree for every project — no reinventing
- A README in the root explains the tree itself
- Desktop and Downloads are not project folders

Structure is cheapest at the start — and impossibly expensive two years in.

Mini-activity — diagnose this CSV (3 min)

date	temp	humidity	notes
14/03/26	23.5	62	clear
03-15-2026	74 F	N/A	sensor moved?
March 16	23,8	na	
17/3/26	24.1 C		RAIN
2026-03-18	?	58%	ok

YOUR TASK

Spot the horrors.

List five acquisition-stage decisions that would have prevented them.

Every mess in this table was preventable before the first row was typed.

Debrief — five decisions that would have prevented it

1. One date format — ISO 8601 (YYYY-MM-DD), written into the SOP
2. One unit, declared once — °C in the data dictionary, never inside cells
3. One missing-value convention — a single agreed code, not N/A, na and blanks
4. Validation at entry — reject ‘74 F’ and ‘?’ the moment they are typed
5. A data dictionary from day one — so ‘humidity’ has a type, range and unit

Three minutes of class time, one lesson: prevention is a design act, not a repair job.

Acquisition done right

Know your source
Primary, secondary or operational — each has its own risks

Know your method
Files, APIs, streams, surveys — and their failure modes

Design against bias
The sample decides what the data can ever say

Capture metadata now
Settings, units, operator — or lose them forever

Protect the raw
Read-only originals; every change is a new version

Write it down
SOP, names, folders — documentation starts on day one

Acquisition decisions are irreversible — next: managing data through its whole life.

Lifecycle management

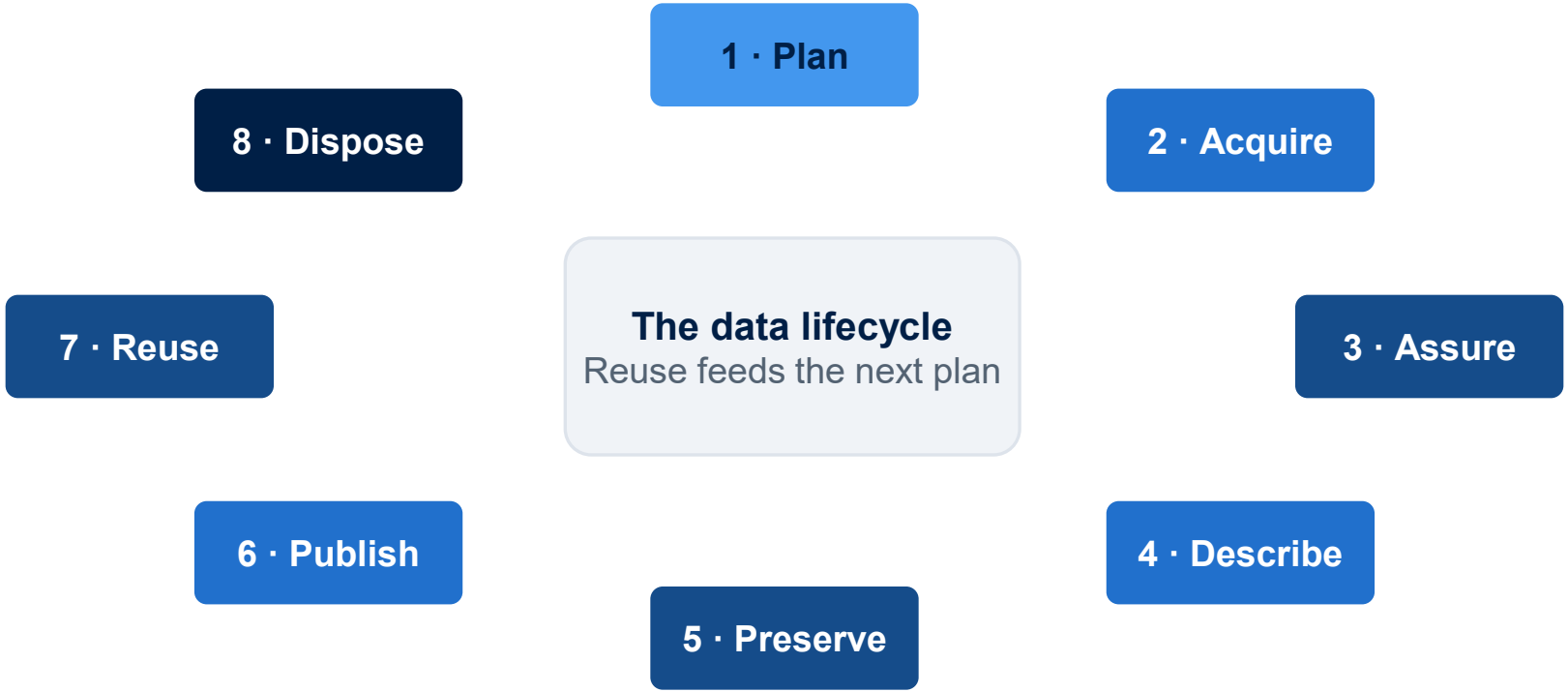
From plan to disposal — and back again

10

The lifecycle model

Eight stages, drawn as a circle

The data lifecycle — one loop



Stage 1 — Plan: the Data Management Plan

FIVE QUESTIONS A DMP ANSWERS

- What data — types, sources, expected volume
- What formats — open, durable, documented
- Who is responsible — named curator, not ‘the group’
- Where it lives — storage, backup, repository
- What it costs — curation time and storage budget

ONE PAGE IS ENOUGH

Funders increasingly require a DMP with every proposal — but the real customer is your own project, two years on.

Homework: you will write one for a dataset of your choice.

Plan the data before the first byte arrives — everything downstream inherits this page.

Stage 2 — Acquire: you already know this one

Choose the source
Primary, secondary or operational — and their trade-offs

Watch the traps
Encodings, locales, rate limits, packet gaps, question bias

Capture it now
Settings, calibration, time, place, operator, units

All above discussion lives inside this one stage — collection done right is curation half-done.

Stage 3 — Process & assure: earn the word ‘clean’

WHAT HAPPENS HERE

- Clean — missing values, duplicates, encodings, units
- Validate — range, type and cross-checks
- Transform — derive analysis-ready tables from raw
- Assure — record what was checked, by whom, when

QUALITY ASSURANCE ≠ CLEANING

Cleaning changes the data. Assurance is the evidence trail — the checklist that says which tests this version passed.

Steps 3–4 of ‘How we curate’, now as a lifecycle stage.

Every dataset is ‘clean’ until someone checks — assurance is the checking, written down.

Stage 4 — Describe: metadata with a standard accent

YOU ALREADY HAVE THE HABITS

- Data dictionary, README, codebook — Step 5
- Now add: a shared standard, so machines can read it too
- Standards fix the field names — you fill the values

Same letter, different accents — one per discipline.

WHY A STANDARD?

'temp' in your README means nothing to a search engine. 'dc:coverage' or a CF standard name is findable by every tool that speaks the standard.

Next slide: three you should recognise.

Describe in a shared vocabulary — metadata is only FAIR if machines can read it.

Metadata standards — speak a known language

Don't invent your own metadata fields — communities already agreed on schemas:

Dublin Core
The generic one

15 basic fields — title, creator, date, format… works for any resource

DICOM
Medical imaging

Scanner settings, patient context, acquisition parameters baked into every file

NetCDF-CF
Climate & earth science

Self-describing arrays: units, coordinates, conventions inside the file

Generic core + your domain's standard — that's how strangers' machines read your data.

Stage 5 — Store: hot, warm and cold

HOT — Working data — SSD, local disk, live database
Accessed daily · fastest · most expensive

WARM — Recent projects — institute server, cloud bucket
Accessed monthly · balanced cost

COLD — Finished projects — archive, tape, deep storage
Accessed rarely · cheapest · slow retrieval

Data moves DOWN as projects age

— and back up when reuse begins

Match the tier to the access pattern — paying hot prices for cold data wastes the budget.

The 3-2-1 rule — and checksums that stand guard

3

copies of your data

original + two backups

2

different media

disk + cloud, not two folders on one laptop

1

copy offsite

fire, theft and floods happen to buildings

Fixity check: store the SHA-256 checksum with every archived file — re-compute on a schedule.

If a single bit rots on disk, the mismatch raises the alarm early.

Backups protect against loss — checksums protect against silent corruption. You need both.

Stage 6 — Publish: repository, DOI, licence

Publishing = repository + DOI + licence — so ‘may I use this?’ needs no email.

CC-BY

Reuse allowed, credit required

The default for research data

CC0

No rights reserved — public domain

Maximum reuse, zero friction

ODbL

Share-alike for databases

Derivatives must stay open

No licence stated = all rights reserved = legally unusable, however FAIR the rest of it is.

Data without a licence is a gift nobody is allowed to open.

Stage 7 — Use & reuse: where value is created

WHAT HAPPENS HERE

- Analysis, models, papers — the reason the data exists
- Integration: joining your data with other datasets
- New users arrive with questions you never imagined

Every earlier stage was for this moment.

THE LOOP CLOSES

Reuse produces new derived data — which needs its own plan, metadata and preservation.

A reused dataset re-enters the cycle at Stage 1.

Reuse is the return on every curation investment you made upstream.

Stage 8 — Archive or dispose: a deliberate end

ARCHIVE — keep, coldly

- Retention schedule says how long (funders: often 5–10 yrs)
- Cold storage + checksums + minimal metadata

Still findable — just sleeping.

DISPOSE — delete, properly

- Deleting a file ≠ destroying data
- Overwrite or cryptographic erasure; all copies, all backups

Sometimes deletion is the law — Part 4.

Every dataset deserves a deliberate end — archived with care, or destroyed with proof.

"

The lifecycle is a circle, not a corridor.

Reuse generates new data, new questions and new plans — yesterday's published dataset is tomorrow's Stage 1.

Curation never finishes — it hands over.

Versioning & provenance

Where did this number come from?

11

Why version data — the cure for REALLY_final

WITHOUT VERSIONS

`final.xlsx · final_v2.xlsx · final_REALLY_final.xlsx`

- Which one made Figure 3? Nobody knows
- Each save overwrites history — mistakes are permanent
- Two collaborators edit two different ‘finals’

THE TWO RULES

1. Raw data is never overwritten — ever.

2. Every change creates a new, numbered version with a one-line change log.

Versions are cheap. Lost history is not.

A version number is a promise: this exact file made that exact figure.

Semantic versioning — numbers that mean something

v1.0

First public release

The dataset as first published — the reviewer's baseline.

v1.1

Correction

Fixed values, filled gaps — same columns, same meaning.

v2.0

Schema change

Columns added, renamed or re-unitised — old scripts may break.

The rule of thumb: change the decimal for fixes, the integer when downstream code must change.

v1.1 says 'trust me, nothing moved' — v2.0 says 'stop and re-read the dictionary'.

Which change bumps which number?

The change	New version
Fixed 12 duplicate rows	**v1.1 — correction**
Corrected °F → °C in one column	**v1.2 — correction**
Renamed columns, split address field	**v2.0 — schema change**
Added a whole new year of records	**v2.0 — shape change**
Re-ran with a fixed random seed	**v1.3 — correction**

Same meaning, better data → v1.x · different shape or meaning → v2.0

Tools that do this for you — names to remember

Git
Versions code and small text files — the standard everywhere

Git-LFS
Git's extension for large files — data alongside code

DVC
Data Version Control — Git-style workflow built for datasets

DB snapshots
Point-in-time copies of a whole database

Names only for today — you'll use them hands-on later in the course.

Don't version by hand what a tool can version for you.

Provenance — the biography of a dataset

SOURCE
station logger, v1.0, 2026-03-14

→

TRANSFORMATION
clean_v2.py · drop dupes, fix units

→

OUTPUT
v1.1 + SHA-256

Provenance answers: which sources fed this? · which script, which settings? · who ran it, when? · can I rerun it today and get the same file?

Every dataset should introduce itself: ‘I came from X, via Y, on date Z.’

The code IS the provenance

clean_v2.py

```
drop_duplicates(subset='station_id')
df = df[df.t2m.between(-10, 55)]
df.to_csv('clean_v1.1.csv')
```

Every decision, readable and re-runnable, forever.

WHY SCRIPTS WIN

A script answers 'what exactly did you do?' by existing.

Re-run it on the raw file — you get the same clean file, byte for byte.

Documentation that cannot lie.

Write the cleaning as code — the script is a lab notebook that re-runs itself.

Why Excel clicks can’t testify

POINT-AND-CLICK CLEANING

- Sort, delete rows, fix cells by hand — no record survives
- ‘I think I removed some outliers… somewhere’
- Undo history dies when the file closes

Fine for looking — fatal for cleaning.

SCRIPTED CLEANING

- Every step written down, in order, forever
- Re-runnable on raw data — same result every time
- Reviewable: a colleague can read the decisions

Excel for exploring; scripts for transforming.

If the cleaning lives in your mouse hand, the provenance is already lost.

Checksums — a fingerprint for every file

HOW IT WORKS

- A hash (MD5, SHA-256) condenses the whole file into one short string
- Change one bit anywhere → completely different fingerprint

```
sha256 clean_v1.1.csv
7f3a09b2…e41c
```

Store this beside the file.

WHEN TO CHECK

After every copy or download · before every analysis run · on a schedule in the archive.

A checksum turns ‘I hope the file is intact’ into a yes/no question.

Silent corruption — the quietest data loss

HOW IT HAPPENS

- A disk sector fades; a copy drops a block; a bit flips
- No error message — the file opens fine
- One temperature reading is now 3275 °C — and nobody knows

The scariest loss is the one that looks like data.

THE DEFENCE

Checksum at ingest → store it → re-verify on a schedule.

Mismatch? Restore from a backup that does match.

This is fixity — preservation's heartbeat.

Backups protect against deletion — only fixity checks protect against decay.

The workflow — all three habits together

VERSION
raw/ is read-only; every release gets a number (v1.0 → v1.1 → v2.0)

RECORD
every transformation is a script, linked source → output

VERIFY
every file carries a SHA-256; re-checked on schedule

RESULT
Any number in any figure can introduce itself.

Version it, script it, checksum it — three habits, total traceability.

"

If you can't say how it was made, it isn't data — it's a rumour.

Versioning says which one. Provenance says how. Checksums say it's still the same.

Next: choosing formats that survive decades.

Storage formats

Will it still open in ten years?

12

Open vs proprietary formats

OPEN — the spec is public

- CSV, JSON, Parquet, HDF5, NetCDF, PNG
- Any tool — today's or 2040's — can implement a reader

The format outlives any vendor.

PROPRIETARY — one vendor holds the key

- Old lab-instrument binaries, niche CAD, legacy stats files
- If the vendor dies or the licence lapses, the data is hostage

The lock belongs to someone else.

Archive in open formats — keep a proprietary copy only if it preserves extra detail.

Text vs binary — readable vs efficient

TEXT — CSV, JSON, XML

- A human can open it in any editor and read it
- Survives decades — worst case, it's still legible bytes
- But: bulky, slow to parse, no types

Great for archives and small data.

BINARY — Parquet, HDF5, NetCDF

- Compact, fast, typed — built for big data
- Needs a library to read — choose OPEN binary formats
- But: unreadable without the right tool

Great for working data at scale.

Readable-forever vs fast-and-compact — open formats let you have both.

Row vs columnar — CSV and Parquet

ROW-ORIENTED — CSV

- Stores record after record, like a diary
- Great for reading whole records, appending, streaming
- Reads every column even if you need one

One row = one line of text.

COLUMN-ORIENTED — PARQUET

- Stores column after column, like a ledger
- Reads only the columns you ask for — often 10×+ faster for analytics
- Compresses far better — similar values sit together

Built for scan-and-aggregate workloads.

Same table, two shapes on disk — pick the shape that matches how it will be read.

So... CSV or Parquet? Both.

Sharing with anyone, any tool	CSV — universal, human-readable
Archival master copy	CSV (open, simple) + checksum
Analytics on millions of rows	Parquet — columnar speed + compression
Pipeline intermediate files	Parquet — types preserved, compact

Publish CSV for people, keep Parquet for pipelines — and never trust types to CSV alone.

Big scientific data — HDF5 and NetCDF

WHEN TABLES AREN'T ENOUGH

- Satellite grids, climate runs, microscope stacks — arrays, not rows
- HDF5: hierarchical containers — many arrays + metadata in one file
- NetCDF (CF): the climate community's standard on top of HDF5

Self-describing: units and axes travel inside the file.

TWO TRICKS FOR SCALE

Compression — shrink each block as it's written.

Chunking — split arrays into tiles, so you can read one region without loading the whole planet.

Big science data ships in self-describing containers — compressed, chunked, and labelled inside.

Format migration — a race you re-run every decade

WHY FILES OUTLIVE SOFTWARE

- Retention is 5–10+ years — software versions last 2–3
- A format nobody can open is a polite form of data loss
- Plan periodic migration: open the archive, re-save, re-checksum

Migration is a scheduled task, not an emergency.

A FAMILIAR STORY

A 2005 thesis dataset in a .mdb Access database — today it needs a software archaeologist.

Saved as CSV + a schema note, it would still open anywhere.

Every decade, walk your archive forward — before the software walks away.

Media obsolescence — the shelf keeps moving

Floppy
1.4 MB · dead by 2005

CD-R
700 MB · rots in a decade

DVD
4.7 GB · drives vanished

Hard disk
TBs · fails mechanically

Cloud
who pays in 2040?

Each hop preserved the bits only because someone copied them forward in time.

Media die faster than data — preservation means copying, not keeping.

The cost model — disks are cheap, people are not

STORAGE — CHEAP

A terabyte costs less than a textbook, and falls every year.

Keeping bytes is the easy part.

CURATION — EXPENSIVE

Metadata, QA, documentation and migration are skilled human hours.

Understanding bytes is the hard part.

That is why Selection & appraisal (activity 1) exists: curate the few datasets that earn it.

“Store everything” is affordable — “curate everything” is not. Choose deliberately.

Before you save — the five-question format check

1. Is the specification open and documented?
2. Will it open with no special software in 10 years?
3. Does it keep precision, units and encoding intact?
4. Does the community I share with already use it?
5. Can I pair it with a checksum and a schema note?

Five yes answers = a format your successors will thank you for.

"

Choose formats for your successors, not for your software.

The tool you love today is the .mdb file of 2040.

Next: retention and end-of-life — when data must die.

Retention & end-of-life

How long to keep it — and how to truly let go

13

How long must data live?

COMMON RETENTION NORMS

- Funders: keep research data 5–10 years after publication
- Institutions: often set their own minimums per policy
- Clinical & regulated data: longer, sometimes decades

Check your funder's policy — it is a condition of the grant.

WHY SO LONG?

Challenges to a paper can arrive years later — the data is your evidence.

Slide 25's reviewer email came two years after publication.

Retention is not hoarding — it is a scheduled promise with an end date.

The retention clock

Year 0
Paper published — dataset archived, clock starts

Year 5
Funder minimum reached — review, don't auto-delete

Year 10
Institutional norm — archive deeper or schedule disposal

Review
Keep (still cited?) or dispose — deliberately

Put the disposal date in the DMP on day one — end-of-life is planned, not discovered.

Must keep — and must delete: the tension

MUST KEEP

- Funder: retain 10 years
- Journal: data behind figures stays available
- Legal hold during a dispute

Deleting too early = misconduct risk.

MUST DELETE

- A participant withdraws consent
- Privacy law: erase when purpose is served
- Contract: destroy partner data at project end

Keeping too long = liability.

Both sides can bind the same dataset — governance resolves who wins. That's next.

Deleting a file ≠ destroying data

WHAT ‘DELETE’ ACTUALLY DOES

- Recycle bin: the file just moves house
- Emptying it: only the index entry is removed
- The bytes stay on disk until overwritten — recovery tools read them back

‘Deleted’ usually means ‘hidden from you’.

LIBRARY ANALOGY

Deleting a file is tearing the card out of the library catalogue — the book is still on the shelf.

Anyone who walks the aisles can still read it.

If the data must be gone, removing the catalogue card is not enough — destroy the book.

Secure disposal — three ways to truly destroy

Overwrite

Write new data over every sector — several passes for magnetic media.

Best for: Old drives, local disks

Cryptographic erasure

Encrypt from day one; destroying the key destroys the data — instantly, everywhere.

Best for: Cloud storage, backups

Physical destruction

Shred, degauss or melt the medium itself.

Best for: Failed drives, top-sensitivity data

Cryptographic erasure is the only method that also reaches the copies you can't touch.

Disposal is an engineering task with a proof at the end — not a drag to the recycle bin.

The copies you forgot about

3-2-1 backups
The offsite copy outlives the original — by design.

Cloud replicas
Object stores keep shadow copies and version history.

Shared drives
That ‘temporary’ copy in the group folder from 2023.

Colleagues’ laptops
Everyone who ever said ‘just email it to me’.

Email attachments
Sent once, stored forever — in two mailboxes.

Old media
The pen drive in the desk drawer.

A retention schedule that forgets the copies is theatre — inventory them on day one.

End-of-life checklist — a deliberate goodbye

1. Check the retention schedule — is the clock actually up?
2. Check legal holds and consent terms — any reason to keep or to delete?
3. Inventory every copy — backups, cloud, collaborators
4. Destroy by method with proof — or archive cold with checksum
5. Record the disposal — what, when, how, authorised by whom

Data ends well when its ending was planned — the last entry in the provenance log.

"

Plan its life. Version its changes. Choose its formats. Schedule its ending.

That is lifecycle management — Part 3 complete.

Next — who decides, who answers? Governance.

Governance

Who may, who must, who answers

14

What governance is

Policies, roles and accountability

Governance — the rules of the kitchen

Data governance is the framework of policies, roles and processes that decides

how data is managed, who can do what with it, and who is accountable when it goes wrong.

Who may?
Access rules — who can read, edit, share

Who must?
Duties — who cleans, documents, reviews

Who answers?
Accountability — whose name is on it

Curation does the work — governance decides whose job it is and by what rules.

Back to the kitchen — who runs this place?

Data owner = The restaurant owner

Accountable for the kitchen — sets rules, carries the risk

Data steward = The head chef

Responsible for quality — recipes followed, standards kept

Data custodian = The kitchen manager

Runs the infrastructure — fridges, locks, stock, backups

Data users = The diners

Use the dish — within the house rules, allergies declared

Four hats, one kitchen — same data, four different responsibilities.

The Four Roles — Responsibilities at a glance

Role	Decides / does	In our research group
Owner	Sets policy, accountable for risk	The PI / lab head
Steward	Quality, standards, metadata	The senior PhD student
Custodian	Storage, access, backups	IT / the server admin
User	Analyses within the rules	Everyone using the data

Owner vs Steward — Accountable vs Responsible

OWNER — ACCOUNTABLE

- Answers when things go wrong — even if others did the work
- Signs the policy, approves access, owns the risk

One name, not a committee.

STEWARD — RESPONSIBLE

- Does the day-to-day: quality checks, metadata, standards
- Knows the data best — the person you actually call

The head chef, not the owner.

Accountable ≠ responsible — governance names both, so nothing falls between.

Governance artifacts — the paperwork that works

Data policy
The house rules — what is allowed, required, forbidden

Data catalogue
The menu — what datasets exist, who owns them, how to ask

Access control matrix
Who can read, write, share — role by role, dataset by dataset

Quality SLAs
Agreed thresholds — e.g. ≥95% complete, updated monthly

If it isn't written down, it isn't governance — it's folklore.

Governance Decides — Management Does

GOVERNANCE — decision rights

- Who may access patient records?
- What quality threshold is acceptable?
- How long do we retain? Who signs off?

Sets the rules of the game.

MANAGEMENT — execution

- Creates the accounts and permissions
- Runs the quality checks nightly
- Operates backups, storage, deletion jobs

Plays the game by those rules.

Governance is the rulebook; management is the referee and the players.

Scenario — ‘Can I have the patient dataset?’

A PhD student emails: “Could I get the hospital dataset for my model?” Who does what?

Owner — Decides: is this use within policy and consent scope?

Steward — Checks quality, documentation and de-identification first

Custodian — Grants access technically — account, encryption, audit log

User (student) — Signs the agreement; uses data only for the stated purpose

Four hats, one request — and every step leaves a record.

When governance works, access is a process — not a favour, not a fight.

"

Governance answers three questions: who may, who must, who answers.

Curation without governance is heroics; governance without curation is paperwork.

Next: the quality bar that governance sets.

Data quality dimensions

Six ways data can be good — or quietly bad

Six Dimensions of Data Quality

1 · Accuracy
Does it match reality?

2 · Completeness
Is anything missing?

3 · Consistency
Does it agree with itself?

4 · Timeliness
Is it fresh enough?

5 · Validity
Does it follow the rules?

6 · Uniqueness
Is each thing counted once?

Six questions to ask any dataset — governance decides how good is good enough.

Accuracy & Completeness

1 · ACCURACY

Values match reality — the patient really is 34, the temperature really was 31 °C.

Broken by: typos, wrong instruments, stale calibration.

Check: spot-verify against the source.

2 · COMPLETENESS

All expected records and fields are present — no silent gaps.

Broken by: sensor outages, skipped survey pages, lost files.

Check: count what arrived vs what was expected.

Accurate but half-missing, or complete but wrong — both fail the user.

Consistency & Timeliness

3 · CONSISTENCY

The data agrees with itself — the same patient isn't 34 in one table and 43 in another.

Broken by: parallel copies, manual edits, merge conflicts.

Check: cross-field and cross-table rules.

4 · TIMELINESS

Fresh enough for the decision at hand — yesterday's stock price is history, not data.

Broken by: slow pipelines, batch delays, forgotten updates.

Check: timestamp every record; measure the lag.

'Good' depends on the use — a climate study and a trading desk need different freshness.

Validity & Uniqueness

5 · VALIDITY

Values follow the declared rules — dates are dates, ages are 0–120, PIN codes have six digits.

Broken by: free-text fields, missing validation at entry.

Check: schema + rule validation on every load.

6 · UNIQUENESS

Each real-world thing appears exactly once — no double-counted patients or duplicate orders.

Broken by: re-imports, merges, spelling variants.

Check: key constraints and duplicate detection.

Valid means ‘legal’, unique means ‘once’ — databases can enforce both for free.

Governance sets the bar — in numbers

QUALITY SLA — weather_tirupati dataset

```
Completeness ≥ 98% of expected hourly rows
Duplicates < 0.1% of records
Timeliness: published within 24 h
Validity: 100% pass range checks
Review: steward reports monthly
```

WHO DOES WHAT

Governance writes these numbers and reviews the report.

Curation runs the checks and fixes what falls below the bar.

‘Good data’ is vague — ‘≥98% complete, <0.1% duplicate’ is a standard you can audit.

Monitoring — quality is watched, not assumed

HOW QUALITY IS WATCHED

- Automated checks run on every new batch
- A simple dashboard: % complete, % duplicate, last update
- Breaches alert the steward — not a lucky discovery

Same idea as the taste-test before serving.

WHO WATCHES

The steward monitors and reports; the owner reviews trends and decides when the bar itself must change.

Quality drifts — monitoring catches it early.

Unmonitored quality decays silently — a threshold without a check is a wish.

"

Governance sets the bar. Curation clears it.

Six dimensions, measurable thresholds, a steward who watches — that is quality by design, not by luck.

Next: privacy, security and the law.

Privacy & the law

Whose data is it — and what does the law demand?

Personal vs sensitive personal data

PERSONAL DATA

- Anything that identifies a person
- Name, phone, email, address, student ID
- Also combinations — ‘the left-handed girl in row 3’

Default: handle with consent and care.

SENSITIVE PERSONAL DATA

- Health records, biometrics, financial data
- Caste, religion, sexual orientation
- Harm from leaks is severe and irreversible

Higher bar: stricter consent, stronger security.

The more a leak could hurt someone, the higher the duty of care — the law grades it the same way.

Anonymise or pseudonymise — not the same thing

PSEUDONYMISATION

- Replace names with codes: P-1023
- A key file maps codes back to people
- Reversible — so it is still personal data in law

Guard the key like the data itself.

ANONYMISATION

- Remove or coarsen identifiers for good
- No key exists — re-identification should be impossible
- Done properly, it may leave the law's scope

Harder than it sounds — next slide shows why.

A code with a key is a disguise, not anonymity — the law can still see through it.

Three harmless columns can name you

Birth date
01-04-2003

+

Gender
F

+

PIN code
517501

=

YOU

Studies of census-style data show that a birth date, gender and postal code together uniquely identify most individuals — no name needed. ‘Anonymised’ tables with these quasi-identifiers can often be re-identified by joining public lists.

Dropping the name column is not anonymisation — think in combinations, not columns.

India’s law — the DPDP Act, 2023

Digital Personal Data Protection Act, 2023 — India’s framework for digital personal data.

Consent-based
Personal data is processed only with free, informed, specific consent — or narrow legitimate uses.

Purpose limitation
Data collected for X may be used for X — not quietly repurposed for Y.

Data fiduciary
The organisation deciding how data is used carries the obligations — like our data owner.

Real penalties
Non-compliance can cost up to ₹250 crore per breach category.

If your dataset contains people, DPDP is not optional reading — it is your checklist.

DPDP — what a data fiduciary must do

The data fiduciary decides why and how personal data is processed — with that decision comes duty:

- Ask first — Free, informed, specific consent — withdrawable any time
- Use only as stated — Purpose limitation: collected for X means used for X
- Keep it safe — Reasonable security safeguards; notify breaches
- Delete when done — Erase when purpose is served or consent withdrawn
- Answer for it — Grievance officer; accountable to the Data Protection Board

Under DPDP, personal data is a loan from the person — never a possession.

GDPR — the European benchmark

THREE IDEAS TO KNOW

- Lawful basis — every use of personal data needs a legal reason
- Right to erasure — people can demand deletion
- Data minimisation — collect only what the purpose needs

Sound familiar? DPDP shares this DNA.

WHY IT MATTERS TO YOU

Collaborate with an EU lab, host EU users, or reuse EU datasets — GDPR travels with the data.

Design to the strictest law you may meet.

GDPR made privacy a design requirement — the world's laws now rhyme with it.

Health data and human subjects

HIPAA — US HEALTH DATA

- Protects identifiable health records
- Relevant when using biomedical datasets from US sources

Health data is sensitive by default, everywhere.

RESEARCH ETHICS — YOUR LAB

- Institutional Ethics Committee approval before collecting
- Informed consent for any human-subject data

The IEC letter is part of the dataset's metadata — archive it.

For human data the question is never just 'can we?' — it's 'did they agree, and who approved?'

Security basics — compliance in practice

Access control
Least privilege — only those who need it, only what they need

Encryption
At rest and in transit — a stolen laptop stays sealed

Audit logs
Who touched what, when — the record that answers questions

Breach response
A plan before it happens: contain, assess, notify

Security is how compliance becomes real — policy on paper, locks on the door.

Which rules apply to your dataset?

Your data involves…	Rules in play	First call
Indian residents' personal data	DPDP Act 2023	Consent + purpose limitation
EU residents / EU collaboration	GDPR	Lawful basis + minimisation
US health records	HIPAA	De-identify before sharing
Human subjects in research	IEC / informed consent	Approval before collection

"

Privacy is not a feature you add — it is a promise you keep.

DPDP, GDPR and the ethics committee all ask the same thing: treat people's data the way you'd want yours treated.

Next: where the law ends and ethics begins.

Ethics beyond the law

Allowed is not the same as right

Legal ≠ ethical — the scraping example

WHAT THE LAW SAYS

- The posts are public — scraping may well be lawful
- No password bypassed, no contract signed

So: allowed.

WHAT ETHICS ASKS

- Did these people expect researchers to mine their posts?
- Could the analysis embarrass or endanger them?
- Would they consent if asked?

So: pause.

‘Public’ describes the data — not permission for every possible use of it.

Consent scope creep — collected for X, used for Y

2019: students share fitness-tracker data ‘to study exercise habits on campus’.

2021 — The same data feeds a stress-detection model — nobody re-asked.

2023 — An insurer partner wants it for ‘wellness scoring’ — quietly different purpose.

2025 — The original volunteers have graduated — and never knew.

Consent has a scope — a new purpose needs a new question, not a quiet reuse.

Dual use — the second life of a good tool

The same dataset — two futures:

Census data → intended: Plan schools and hospitals
→ second use: Target a minority community

Face dataset → intended: Unlock your own phone
→ second use: Mass surveillance

Mobility traces → intended: Optimise bus routes
→ second use: Track individual movements

You can't control every future use — but access rules and licences decide who gets the chance.

Algorithmic harms — bad governance, real victims

THE HARM CHAIN

- Ungoverned training data — biased, stale, unconsented
- Model learns the flaws and scales them
- Decisions hit real people: loans, jobs, diagnoses

Remember the hospital model from earlier?

WHERE CURATION HELPS

Documented provenance, bias checks at collection, and access rules make harmful shortcuts visible before deployment.

Governed data is the first line of defence.

Models inherit their diet — feed them governed, documented, consented data.

Licence your outputs — and respect what you borrowed

YOUR DATA, YOUR CHOICE

- CC-BY — use freely, credit me
- CC0 — no strings, maximum reuse
- Choose deliberately — silence helps nobody

An unlicensed dataset is unusable, not protected.

BORROWED DATA, THEIR RULES

- Read the licence before you build on it
- ODbL and share-alike terms follow your outputs
- Cite datasets like papers — DOI and version

Respecting licences is curation etiquette.

Ethical reuse is a two-way street — licence what you give, honour what you take.

"

The law is the floor, not the ceiling.

Before every reuse, ask the person in the data: would they nod?

Next: wrap-up, homework and exam hooks.

Wrap-up

Four things to remember, three things to practise

Take-home 1 — curation is active care

1

Curation = active, ongoing management of data through its lifecycle — for reuse and trust.

- Not a heroic one-off clean-up — a weekly habit
- The goal has a name: FAIR — findable, accessible, interoperable, reusable

If a stranger can't cook it from what you left behind, it isn't curated.

Take-home 2 — acquisition is irreversible

2

Acquisition decisions are irreversible — get them right at the source.

- Capture metadata at collection: settings, calibration, units, time, operator
- Design against bias — no downstream cleaning fixes a biased sample

Downstream can’t invent what upstream never recorded.

Take-home 3 — the lifecycle is a cycle

3

The lifecycle is a cycle, not a corridor.

- plan → collect → assure → describe → preserve → share → reuse → dispose
- Reuse generates new data and new plans — Stage 8 feeds Stage 1

Curation never finishes — it hands over.

Take-home 4 — governance makes it answerable

4

Governance assigns accountability — compliance is non-negotiable.

- Owner decides, steward tends, custodian guards, users respect the licence
- DPDP, GDPR and research ethics: the law is the floor, ethics is the ceiling

Who may, who must, who answers — every dataset should have an answer.

Homework — three things to practise

1 · Write a DMP
One page, for a dataset from your own research area: what, formats, who is responsible, where it lives, what it costs.

2 · Rescue a messy CSV
We give you the horror file. Produce (a) a cleaned version, (b) a data dictionary, (c) a provenance note listing every transformation.

3 · Judge a case
A hospital shares patient records with a startup. Identify the governance roles, the applicable law (DPDP), and what anonymisation would require.

Each task exercises a different muscle: planning, hands-on curation, and governance judgement.

Exam-style questions — could you answer these?

Q1. Explain the FAIR principles, with one concrete example per principle.

Q2. Differentiate data storage, data management and data curation.

Q3. List and explain four biases that arise at the data collection stage.

Q4. Describe the stages of the data curation lifecycle. Why is it drawn as a cycle?

Q5. What obligations does a data fiduciary have under the DPDP Act, 2023?

Every one of these was answered somewhere in this lecture notes— including by a plate of biryani.

The biryani, one last time

The photo → **Raw data**

Naming the ingredients → **Collection & labelling**

Washing the rice → **Cleaning & validation**

The recipe card → **Metadata & provenance**

The labelled spice rack → **Storage that curates**

The fridge & expiry dates → **Preservation & lifecycle**

Who may eat it → **Governance & consent**

Anyone can cook it again → **Reuse — the goal**

One plate of biryani carried the whole syllabus — now go curate like a cook who writes things down.

"

Thank you — and remember:

your future self is your first collaborator.

Curate for them — they already know where you hid the turmeric.