Group Submission — AI Interaction Log (example)

Group Topic: Data Importing & Hidden Assumptions
Dataset: insideairbnb_chicago_detailed_listings_messy_600_2025_09_22.csv

Prompt 1 (Goal: robust import plan)

Prompt:
"We have a CSV with messy missing values (blank/NA/N/A/period/space), mixed date formats, percent strings like 99%, currency like $1,027.00, and boolean flags t/f. What is a robust import strategy in R that avoids silent errors?"

AI response excerpt (illustrative):

- Use readr::read_csv()
- Specify na = c("", "NA", "N/A")
- Keep risky columns as character first, then parse deliberately
- Use problems() and post-import checks (glimpse, summary, count(is.na))

Our annotation:

- **Right:** Advises explicit missing tokens, delaying parsing for risky columns, and validating post-import.
- **Hidden assumption:** It listed missing tokens but **did not include "." or a single space** as missing.
- **Evidence from our file:** In host_response_rate, missing tokens include **blank=28, NA=12, N/A=8, '.'=11, space=11**. If "." and " " aren't treated as missing, they persist as valid strings.

Prompt 2 (Goal: detect silent type-guessing problems)

Prompt:
"What are the most common silent import errors caused by type guessing, and how do I detect them?"

AI response excerpt (illustrative):

- Leading zeros can be lost if IDs/zip codes are parsed numeric
- Percent/currency stored as text prevents numeric ops
- Dates can parse partially
- Detect by checking column classes, unique values, and parsing warnings

Our annotation:

- **Right:** Identifies the correct risk categories (types, dates, percent/currency).
- **Hidden assumption:** It assumes "checking column class" is enough. Many failures require **quantifying** parse success/failure or inspecting "high-risk" rows.
- **Evidence:** host_since is **mixed format**: ISO **480** rows, US-style **84** rows (non-missing **564**). If we parse with a single-format parser, some values will become NA without obvious alarms.

Prompt 3 (Goal: mixed-date parsing strategy)

Prompt:
"Our date columns mix YYYY-MM-DD and M/D/YYYY. What's the safest way to parse and how do we report failures?"

AI response excerpt (illustrative):

- Use lubridate::parse_date_time(x, orders=c("ymd","mdy"))
- After parsing, report how many dates became NA compared to non-missing inputs
- Inspect a few rows that failed parsing

Our annotation:

- **Right:** Multi-format parsing plus explicit failure reporting is a good practice.
- **Assumption:** It assumes only two formats exist; that must be verified.
- **Evidence:** We checked two columns:
  - host_since: ISO=480, US=84, other=0 (non-missing 564)
  - last_scraped: ISO=510, US=90, other=0 (non-missing 600)

Prompt 4 (Goal: safe conversion for percent and currency)

Prompt:
"How should we convert 99% and $1,027.00 into numeric safely and verify nothing broke?"

AI response excerpt (illustrative):

- Use readr::parse_number() for currency
- For percent, strip %, convert to numeric, and decide scale (0–1 vs 0–100)
- Verify with summaries and spot checks

Our annotation:

- **Right:** parse_number() is appropriate for $ and commas; percent needs deliberate scaling decision.
- **Hidden assumption:** It assumes no extreme outliers.

- **Evidence:** Our price column contains commas in **9** values, including **$40,000.00**. A careless conversion can overflow intuition (and distort later modeling/plots) unless outliers are flagged.

Prompt 5 (Goal: encoding/non-ASCII audit)

Prompt:
“How do we check for encoding problems (e.g., names like Zoë, María) during import?”

AI response excerpt (illustrative):

- Ensure UTF-8 locale when reading
- After import, search for replacement characters or garbled sequences
- Spot-check known non-ASCII strings

Our annotation:

- **Right:** Emphasizes post-import checks rather than blind trust.
- **Hidden assumption:** It assumes we know what to look for; students need concrete checks (e.g., search for “Ã” patterns).
- **Evidence:** We found non-ASCII host names present, including **Zoë, María, Ömer, Tuğba, Le Méridien, The Swissôtel Chicago**. These must display correctly to avoid join/deduplication errors.