

17 Capstone: Case Studies and Framework for Independent Practice

Purpose of this Chapter

This chapter draws together the full pipeline introduced in Chapter 1 through a set of condensed case studies, each illustrating how several of the book's methods combine in a single applied Nigerian health data science project. It closes with a practical checklist that a reader can apply to a new project of their own, before the book's final chapter turns to a dedicated, hands-on treatment of Python for disease epidemiology.

Case Study A

Malaria Diagnostics Across Global Fund-Supported States

A comparative analysis of RDT, microscopy, and PCR-confirmed malaria testing across thirteen Global Fund-supported states draws on: Chapter 2's understanding of HMIS versus archive-verified data provenance; Chapter 4's cleaning workflow for facility-month records; Chapter 5's exploratory comparison of TPR across diagnostic modalities and states; Chapter 7's hypothesis testing (and multiple-comparison correction) for state-by-state differences; and Chapter 8's logistic regression modelling of confirmed positivity as a function of diagnostic method, season, and state. The key finding that HMIS-derived TPR runs consistently higher than archive-verified TPR across all thirteen states is a

direct, policy-relevant product of the provenance-aware approach this book has emphasised throughout, with implications for how confidently routine data alone can support commodity forecasting decisions.

Case Study B

Leishmaniasis Baseline Mapping and Trend Modelling

A national neglected tropical disease baseline analysis combines Chapter 10's Poisson GLM trend modelling (quantifying the average annual percentage change in reported cases) with Chapter 14's choropleth mapping and small-area smoothing (visualising the spatial distribution of case rates by state, and stabilising estimates for low-population LGAs) and an explicit attribution stamp on every output, reflecting this book's insistence (Chapter 4) that every analytic output should be traceable to its exact data source and generation date — essential when the resulting maps and trend estimates feed directly into an elimination programme's implementation planning and donor reporting.

Case Study C

Adolescent Health Cohort Analysis

A two-wave longitudinal analysis of an adolescent health cohort draws on: Chapter 4's handling of missingness across survey waves (distinguishing MAR attrition from MNAR attrition tied to worsening symptoms, and applying multiple imputation); Chapter 5's exploratory comparison of symptom severity by household food security and school attendance; and Chapter 8's logistic and ordinal regression modelling of depressive symptom severity as a function of individual and household covariates, with careful attention (Chapter 4) to the correlated-observation structure induced by repeated measurement of the same adolescents across waves.

Case Study D

Respondent-Driven Sampling of a Key Population

An RDS study of female sex workers draws directly on Chapter 7's specialised RDS estimators (comparing Volz-Heckathorn and successor-of-successor estimates for robustness), combined with Chapter 8's regression modelling of correlates of a health outcome of interest within the recruited sample, illustrating that even a technically sophisticated sampling design does not exempt the analyst from the confounding and model-diagnostic discipline introduced earlier in the book, nor from the privacy safeguards of Chapter 16 given the sensitivity of the population involved.

Case Study E

A Predictive Triage Model, From Data to Deployment Decision

A synthetic but realistic capstone example: a research team builds a severe-malaria risk model at triage, following the full arc of this book. Chapter 4 cleans and audits the raw triage register; Chapter 5's EDA reveals a right-skewed parasite density variable and a suspiciously high missingness rate for respiratory rate at one facility (traced, on investigation, to a broken pulse oximeter during a specific month); Chapter 8 and Chapter 11 together compare a LASSO logistic regression against a random forest, using facility-grouped cross-validation (Section 11.7) to avoid an optimistic performance estimate; Chapter 16's subgroup audit (Section 16.6) reveals the model under-performs at primary health centres; and the project concludes not with deployment, but with a recommendation to collect a modest amount of additional primary-health-centre training data before proceeding — a genuinely realistic, and appropriately cautious, capstone outcome that models the kind of judgement this book has tried to cultivate throughout, rather than a premature declaration of success.

A Practical Checklist for an Independent Project

- Have I written down the precise question before touching any data (Chapter 1)?
- Do I know exactly which data source(s) I am using, their unit of observation, and their known limitations (Chapter 2)?
- Have I documented my cleaning decisions in a reproducible script, not a manually edited spreadsheet (Chapter 4)?
- Have I explored the data (Chapter 5) before committing to a specific formal model?
- Have I chosen a method appropriate to my data structure and my goal — inference versus prediction (Chapters 7-13)?
- If I used machine learning, have I validated out-of-sample with a split that respects any grouping/clustering in the data (Chapter 11)?
- If my data or results have any spatial or temporal structure, have I accounted for it rather than assuming independence (Chapters 10 and 14)?
- Have I checked my model's assumptions and, where relevant, its performance across policy-relevant subgroups (Chapters 8, 11, and 16)?
- Have I obtained the necessary ethical approval and handled identifiable data appropriately, including a k-anonymity check where relevant (Chapter 16)?
- Can a colleague re-run my entire analysis from raw data to final output and get the same answer (Chapter 6)?

Common Reasons Independent Projects Stall

- Underestimating data access lead time (Section 2.9), leaving too little time for the actual analysis once approvals finally come through.

- Skipping the exploratory analysis stage (Chapter 5) and moving directly to a formal model, only to discover a basic data quality problem much later that requires restarting the analysis.
- Choosing an unnecessarily complex method (a deep learning model for a 300-patient dataset, per Chapter 15's caution) that consumes disproportionate time relative to its benefit over a simpler, well-validated alternative.
- Failing to set up a reproducible project structure (Chapter 6) from the start, leading to confusion later about which script or dataset version produced a given result.

Extended Case Study

When a Capstone Analysis Contradicted the Initial Hypothesis

A final, instructive capstone example, chosen specifically because its outcome contradicted what the research team initially expected: a student project set out to test the hypothesis that a state's community health worker programme, which had been rapidly scaled up over an 18-month period, was responsible for an observed improvement in early ANC (antenatal care) attendance rates in the areas where the programme was deployed. Following this book's full pipeline — careful data provenance work distinguishing programme-deployment LGAs from comparison LGAs (Chapter 2), thorough exploratory analysis (Chapter 5) revealing that programme-deployment LGAs already had somewhat higher baseline ANC attendance before the programme began (a warning sign for confounding by pre-existing area characteristics, per Chapter 8), and a difference-in-differences regression model explicitly designed to account for this pre-existing gap rather than a simple before-after or treatment-vs-comparison comparison alone — the fully adjusted analysis found no statistically significant additional improvement in early ANC

attendance attributable to the programme beyond what would have been expected from the areas' pre-existing trajectory and other confounding trends.

This finding was, initially, unwelcome to a research team that had expected (and to some extent hoped) to demonstrate the programme's effectiveness. Rather than searching for an alternative specification or subgroup that would produce the originally-expected positive result (a temptation this book has warned against throughout, particularly in Chapter 7's discussion of multiple comparisons and Section 7.10's discussion of honestly reported null results), the team instead reported the null finding plainly, together with a careful discussion of its limitations (a relatively short 18-month follow-up window, and the possibility that the programme's benefits, if any, might operate through pathways other than early ANC attendance specifically, such as improved attendance at later ANC visits or improved postnatal care uptake, which the available data could not directly test). This honestly-reported, methodologically rigorous null result was, in the assessment of the student's supervisory committee, ultimately a stronger piece of work than a less careful analysis that had uncritically confirmed the initially expected positive finding would have been — precisely because it demonstrated the full analytical discipline this book has argued for throughout, applied faithfully even when its conclusion was not the one the research team had hoped to reach.

A Note on Working with Co-Authors and Supervisors

Nearly every substantial project following this book's pipeline will involve at least one co-author or supervisor, and the reproducibility and documentation practices emphasised throughout this book (Chapter 6 especially) pay a second dividend specifically in this collaborative context: a supervisor or co-author who can read a clearly commented, version-controlled script and

its accompanying data provenance notes can review and contribute to a project far more efficiently, and with far more genuine understanding of what was actually done, than one who receives only a final results table with no visibility into the underlying process. Establishing early in a collaboration who is responsible for which stage of the pipeline (Section 1.2), and agreeing on a shared, version-controlled repository structure from the outset, prevents the common and avoidable friction of two collaborators independently producing slightly different versions of ‘the’ cleaned dataset or ‘the’ final model, each unaware that the other has made different, undocumented choices somewhere along the way.

Presenting Findings to a Non-Technical Audience

Every case study and mini-project in this book has emphasised technically rigorous analysis; the final, and in many ways most consequential, step of the pipeline introduced in Chapter 1 is communicating that analysis’s findings to an audience — an LGA health officer, a State Ministry of Health technical committee, a community leader — who may have little to no formal statistical training but who holds the actual authority or influence to act on the findings. Effective communication in this setting means leading with the practical implication rather than the statistical method (‘this LGA needs additional net distribution before the rainy season’ rather than ‘the Poisson regression coefficient for LGA was significant at $p < 0.05$ ’), using a small number of carefully chosen visualisations rather than a dense slide of tables, being explicit and honest about the confidence level and limitations of a finding without using so much technical hedging language that the message becomes unclear, and being prepared to answer ‘so what should we actually do about it’ as the natural and expected follow-up question to any presented finding — a skill this book’s case studies (Section 17.2 through 17.6) have modelled throughout, but which ultimately develops most through direct

practice presenting real findings to real, non-technical stakeholders.

Building a Reusable Personal Code Library Across Projects

A natural progression for a reader who has completed several of this book’s mini-projects is to notice recurring code patterns — a cleaning function, a reporting template, a Kaplan-Meier plotting helper — and consolidate them into a personal, reusable code package rather than re-writing similar code from scratch for each new project. Both R (via a lightweight personal package built with ‘devtools’/‘usethis’) and Python (via a simple installable local package) support this pattern well, and maintaining such a library, with basic documentation and version control (Chapter 6), compounds in value over a research career: each new project both benefits from and contributes back to an increasingly capable, personally-tailored analytic toolkit, which is one of the most practically valuable habits an independent Nigerian health data scientist can build over time.

Writing the Methods Section

Translating Code into Prose

A final practical skill this book has implicitly built toward throughout its many worked examples is translating a completed analysis — code, output, and all — into the concise, precise prose of a methods section suitable for a thesis chapter or journal submission. A strong methods section states the data source and study design, describes cleaning and exclusion decisions with their rationale (Chapter 4), names the exact statistical or machine learning method used with enough detail for replication (Chapters 7-15), and states how missing data, multiple comparisons, or model validation were handled — essentially a prose summary of the same provenance-aware, transparent decisions this book has

asked readers to make and document throughout every mini-project, now compressed into the formal register expected of a scientific publication.

A Script to Scaffold a New Reproducible Project

The checklist in Section 17.7 is easiest to follow when a project's directory structure enforces good habits from the very first commit, rather than being retrofitted later. A short scaffolding script that creates the standard folders recommended throughout Chapter 6 (raw data, cleaned data, scripts, outputs), an empty data dictionary template (Section 4.8), and an initial Git repository in a single command removes any excuse for skipping these steps under time pressure at the start of a new project.

PYTHON example

```
# Python: a one-command project scaffolding script
import os, subprocess

def scaffold_project(project_name):
    folders = ["data/raw", "data/clean", "scripts", "outputs", "docs"]
    for folder in folders:
        os.makedirs(os.path.join(project_name, folder), exist_ok=True)
    with open(os.path.join(project_name, "docs/data_dictionary.csv"), "w") as f:
        f.write("variable,definition,type,allowed_range,source\n")
    subprocess.run(["git", "init", project_name], check=True)
    print(f"Scaffolded reproducible project at ./{project_name}")

scaffold_project("malaria_tpr_reconciliation_2026")
```

Chapter Summary

- Real Nigerian health data science projects combine methods from across this book's chapters, not a single technique in isolation.
- Provenance-aware analysis (Chapter 2's core lesson) recurs as the key to interpreting each case study correctly.
- A capstone project's most valuable outcome is sometimes an appropriately cautious recommendation not to deploy yet, rather than a premature declaration of success.

- A short, disciplined checklist — question, data, cleaning, exploration, method choice, validation, structure, assumptions, ethics, reproducibility — is a practical summary of the entire book.

Mini-Project

Mini-Project 17: A Full Capstone Project Proposal

Design (and, as far as feasible, execute) a complete capstone project of your own, applying the checklist in Section 17.7 end to end.

Step 1. Choose a Nigerian health data question of genuine interest to you, distinct from the five case studies in this chapter.

Step 2. Draft a one-page project proposal identifying your data source(s), cleaning plan, exploratory questions, and intended formal analysis method(s), citing the relevant chapters.

Step 3. Execute as much of the pipeline as your data and time allow, from cleaning through to a formal analysis or model.

Step 4. Apply the ten-point checklist (Section 17.7) explicitly to your own project, noting any items you could not fully satisfy and why.

Step 5. Conclude with a short, honest statement of your project's current limitations and what would be needed before its findings could inform a real decision.

Deliverable: *A one-page project proposal, whatever analysis you were able to complete, and a completed checklist with honest self-assessment.*

End-of-Chapter Exercises

1. Choose a health data project from your own work. Apply the ten-point checklist in Section 17.7 and identify the two items you are currently weakest on.
2. For Case Study A, propose one additional analysis (drawing on any chapter of this book) that would strengthen the policy conclusions.
3. Design, in outline form only, a new capstone case study from a Nigerian health data source not covered in Sections 17.2-17.6, specifying which chapters' methods it would draw on.

4. In Case Study E (Section 17.6), explain why recommending ‘collect more data before deployment’ is a more scientifically honest outcome than deploying the model with a known subgroup performance gap.
5. For Case Study C, explain how you would decide whether the adolescent cohort’s follow-up attrition is more consistent with MAR or MNAR (Chapter 4), and what analysis you would run to investigate this.
6. For Case Study D, explain why privacy safeguards (Chapter 16) are especially important for an RDS study of a key population, compared to a general population household survey.
7. Write a short paragraph, in your own words, on which single habit from this entire book you consider most important to carry into your own future work, and why.
8. Using Section 17.8, identify which of the four listed reasons projects stall you consider most common in your own experience or observation, and explain why.
9. Explain how skipping Chapter 5’s exploratory analysis stage could lead to a costly restart later in a project.
10. Propose one practical habit (from anywhere in this book) that would directly address one of the four stall points listed in Section 17.8.
11. In the Section 17.9 case study, explain why the pre-existing difference in baseline ANC attendance between programme and comparison LGAs was a warning sign that needed to be addressed before drawing conclusions.
12. Why was a difference-in-differences approach more appropriate here than a simple before-after or treatment-vs-comparison comparison alone?

13. Explain why the research team's supervisory committee judged the honestly reported null result to be stronger work than an uncritical positive finding would have been.
14. Propose one additional outcome (beyond early ANC attendance) that this programme's benefits might plausibly show up in, and how you would test for it.
15. What is the risk, for the research team's own credibility, of searching for an alternative specification that would produce the originally hoped-for positive result?
16. Using Section 17.10, explain why a co-author who can read a well-documented, version-controlled script can contribute more effectively than one who only sees a final results table.
17. Propose one specific agreement a research team should make at the start of a collaboration to avoid the 'two different versions of the cleaned dataset' problem described in Section 17.10.
18. Why does establishing pipeline responsibilities early (Section 1.2) help prevent friction later in a collaborative project?
19. Describe a situation (real or plausible) where a lack of shared, version-controlled workflow caused confusion or wasted effort in a team project.
20. How would you raise this issue diplomatically with a co-author or supervisor who is used to working without version control?
21. Using Section 17.11, rewrite the example phrase 'the Poisson regression coefficient for LGA was significant at $p < 0.05$ ' into a plain-language, action-oriented statement suitable for a non-technical LGA health officer.

22. Why is being prepared to answer ‘so what should we do about it’ described as a natural and expected follow-up question?
23. Explain the risk of using too much technical hedging language when communicating a finding’s limitations to a non-technical audience.
24. Propose one visualisation choice (from Chapter 5) that would likely communicate a key finding more effectively to a community leader than a dense data table.
25. Why does this book argue that this communication skill develops mainly through direct practice, rather than through reading about it alone?
26. Summarise, in your own words, why this chapter’s checklist (Section 17.7) is presented as a practical summary of the entire book rather than a new set of ideas.
27. Explain the common thread linking the contradicted-hypothesis case study (Section 17.9) and the non-technical presentation discussion (Section 17.11).
28. Why does this chapter emphasise working constructively with co-authors and supervisors (Section 17.10) as a data science skill in its own right?
29. Propose a hypothetical capstone project from your own field, and describe how you would present its key finding to a non-technical audience (Section 17.11).
30. If your own capstone analysis contradicted your initial hypothesis, as in Section 17.9’s case study, how would you decide what to report?
31. Using Section 17.13, list the four key elements a strong methods section should include, according to this chapter.

32. Why is a methods section described as a ‘prose summary’ of the same decisions documented throughout this book’s mini-projects?
33. Choose one of your own mini-project write-ups from this book and draft a two-to-three sentence methods section based on it.
34. Why is stating how missing data and multiple comparisons were handled considered essential for a complete methods section?
35. What is the risk of omitting the rationale for a cleaning or exclusion decision from a published methods section?

Part 6

Python for Disease Epidemiology

