Questionable analytical observations and what to do about them

Charles T. Gray & Hannah Fraser

September 22, 2025

Abstract

A critical challenge in data management is balancing complex data transformation with the need for analytic transparency for domain-knowledge validation. FAIR (findable, accessible, interoperable, reusable) data principles as *compliance* fall short when viewed as a checkbox exercise at publication. Living analyses evolve over time, and all analyses are iterative, requiring ongoing validation of data processing decisions. Contemporary data practices on the modern data stack provide robust tooling to validate data transformation, but in practice render data observability opaque to non-technical stakeholders. This creates a fundamental tension between engineering efficiency and analytic interpretability that undermines the core principles of FAIR data.

This document presents a novel tool-agnostic semantic layering methodology for analytic data pipeline design, demonstrated through analysis of UN Sustainable Development Goal (SDG) priorities from dam-affected communities in North India.

The central methodological contribution is a semantically-structured approach to data transformation that extends dbt's engineering-focused patterns to prioritize research interpretability and stakeholder communication. Unlike standard dbt implementations that emphasize technical efficiency, this semantic layering methodology explicitly preserves research context, makes harmonization decisions transparent, and maintains conceptual clarity throughout the data pipeline.

The semantic layer architecture consists of four conceptually distinct stages: source base (raw data cleaning), source entities (region-specific preservation), semantic models (cross-source harmonization), and analytic models (research-ready datasets). This approach addresses the critical gap between technical data engineering practices and research methodology requirements, enabling automated observability while maintaining scholarly rigor.

This semantic layering methodology offers a replicable framework for research data management that bridges the divide between engineering efficiency and research transparency, with applications across all domains requiring multi-source data integration and stakeholder communication.

Contents

List of Tables

List of Figures

1 Questionable analytical observations

Underestimating the complexity of data transformation is a well-recognised driver of garbage in, garbage out data analysis. For example, gene name errors caused by Excel formatting issues are widespread—detected in 700 scientific publications (approximately 20 per cent of 3,500 publications analysed) [19]. More recently, the UK government's COVID-19 tracking lost about 16,000 cases due to Excel row limits [13]. These examples illustrate how technical limitations and data handling errors can lead to significant analytical inaccuracies.

1.1 Questionable research practices

Broadly, the purpose of research is to discover something useful about the world. However, in reality there are so many kinks in the pipeline between the research question and the answer, that the usability and usefulness of the answer that is produced by that pipeline is uncertain. Though, research that passes peer review is often seen as authoritative and accurate. The 'replication crisis' movement challenged this, finding that only 1/3 – 1/2 of research, when repeated, finds the same results. This low replication rate has been attributed in part to the use of Questionable Research Practices which make research findings appear more impactful and reliable than warranted by the methods used (Gould et al. in prep). Examples of Questionable Research Practices include HARKing (Hypothesising after results are known), p-hacking (which includes everything from choosing to drop outliers after examining the

impact on the results, to rounding p-values down to reach a threshold e.g. p¡0.05), and cherry picking (conducting multiple analyses but only presenting the ones with the significant/most publishable results) [11]. These practices are prevalent across the fields of science [**?**] are indicative of a huge problem in the research pipeline. However, the scope of QRP research has been limited to the analysis and write-up parts of the research pipeline. We argue that the focus on problematic research practices needs to include the earlier parts of the pipeline as well. In this paper we describe "Questionable Analytic Observations" which occur between the point of data collection and the point of analysis, with flow on effects through the rest of the research pipeline. Questionable analytical observations arise when data transformation processes are not transparently validatable or interpretable to domain experts, leading to invisible garbage in, garbage out analytics. This is particularly problematic in multi-source data integration contexts, where harmonization decisions can significantly impact analytical outcomes.

Sharing code, materials and data is one proposed counter to Questionable Research Practices. This is in line with FAIR principles (see below), facilitates true critical analysis of the analysis methods, and should allow someone to test the method by using the code and data to reproduce the original results. Currently, we are falling well short when it comes to code and data sharing estimates suggesting that across fields 0.5-27

1.2 What to do about questionable analytical observations

To address questionable analytical observations, we propose a methodological bridge (Section 3) between modern data engineering practices and analytical validation. This involves:

- **Semantic layering of data transformations** to preserve research context and make harmonization decisions explicit (Section 3.3);
- **Automated observability frameworks** that generate transparent documentation and validation reports from pipeline artifacts (Section 6.3);
- **Stakeholder-centric communication** using conceptual rather than technical terminology to facilitate domain expert validation (Figures 4 and 2);
- **Iterative validation processes** that allow ongoing scrutiny of data processing decisions as analyses evolve over time.

1.3 Case study: giveadam project

This manuscript and accompanying GitHub repository `giveadam` present a case study implementing the proposed methodology on open science data (Section 2). The project analyzes UN Sustainable Development Goal (SDG) priorities from dam-affected communities in North India, using survey data collected by Garima Gupta from residents of Tehri (where a dam was constructed 20 years ago) and Arunachal Pradesh (where a dam is under development).

This case study demonstrates the methodology on small, analytic data where validation is tractable, but the principles are applicable to large-scale data contexts. To illustrate scalability, the methodology uses open source modern data stack tools standard in enterprise environments, such as Data Build Tool (dbt). The following sections detail the semantic layering methodology, repository architecture, and adherence to FAIR principles.

1.3.1 Minimal representation

This case study was chosen as it provides a minimal representation of two key challenges in data managemen:

- **Multi-source data integration:** The project integrates two regional survey datasets with differing formats and labeling conventions, requiring explicit harmonization decisions that impact analytical outcomes.
- **Stakeholder alignment:** Interoperation of three people with different roles and expertise: Garima Gupta (data collection and domain expertise); Charles T. Gray (data engineering and pipeline design); and Hannah Fraser (code review and validation). Crucially, it is Garima Gupta who possessses the domain knowledge to validate the data processing decisions, but she is not a data engineer. This necessitates a methodology that facilitates transparent communication and validation across technical and domain expertise boundaries. Similarly, Hannah Fraser is not a data engineer, but she is a domain expert and can validate the data processing decisions are interpretable.

1.4 Enterprise applications for data governance at scale

The case study described above was chosen so that the methodology is interpretable on data that can be validated by eye. However, at scale, in large data platforms simply opening datasets is not an option. Platforms at scale for enterprise data management and public good typically involve:

- hundreds, if not **thousands, of sources** many of which are provisioned in legacy formats (e.g. Excel, CSV, XML, JSON, SQL dumps);

- hundreds, if not **thousands, of databases** requiring aggregation with integrity;
- **billions of rows of living data** per table in each database equiring ongoing validation specific to `snapshot` or `incremental` data processing.

When working across legacy sources in the volatility of the technology landscape, data engineers typically have little more to work with than inscrutable table names and column headers. Only after many complex steps requiring advanced software engineering codebases is the data available in a way that is accessible to stakeholders with enough domain knowledge to validate that what has been extracted by, say, ETL (extract, transform, load) pipelines is the desired output.

1.4.1 Modern data stack

Advances in the modern data stack (a term to loosely define the steps of cloud-oriented computational tools to manage data [1]), such as dbt [5] and dagster [**?**] have unpredented potential to address these challenges. However, there is a prevailing tendency for these tools to be oriented to interpretability by engineers, rendering the complexity of data processin — crucially the assumptions made — opaque to domain experts.

This creates a fundamental tension between engineering efficiency and analytic interpretability that undermines the core principles of FAIR data. The semantic layering methodology presented in Section 3 addresses this tension by reorienting data transformation pipelines around research concepts rather than technical convenience, enabling both computational efficiency and methodological rigor.

2 About the giveadam data

The data in this project aggregates two region-based surveys of UN SDG priorities from participants in Tehri and Arunachal Pradesh. Respondents ranked their top three UN SDG priorities, providing insights into development preferences in dam-affected communities. The data is stored in the `data/` directory of the giveadam repository (see Section 3.2 for complete repository structure). This project aggregates survey responses across regions using the semantic layering methodology (detailed in Section 3) to investigate the following research question:

> *What are the differences in SDG priorities between residents of Tehri, where a dam was constructed 20 years ago for hydroelectric power, and Arunachal Pradesh, where a dam is currently being developed?*

Figure 1 shows a treemap of the top 3 UN SDG priorities from survey respondents in Tehri (dam constructed 20 years ago) and Arunachal Pradesh (dam under development), North India.

By structuring the data in tidy format (one row per ranking by one respondent), the dataset enables flexible analysis across multiple dimensions of respondent characteristics and preferences.

3 Semantic data pipeline methodology

The core methodological innovation of the giveadam project is a semantic layering approach that reconceptualizes data transformation pipelines for research contexts. This methodology addresses a fundamental limitation in current data engineering practices: the tension between technical efficiency and research interpretability within the modern data stack paradigm [1].

3.1 The semantic layering paradigm

Traditional data engineering prioritizes performance optimization and code maintainability, often at the expense of conceptual clarity for non-technical stakeholders. The semantic layering methodology inverts this priority structure, organizing transformations around research concepts rather than technical convenience. This approach ensures that:

1. **Research context is preserved** throughout the transformation pipeline
2. **Harmonization decisions are explicit** and auditable by domain experts
3. **Stakeholder communication** uses conceptual rather than technical terminology
4. **Methodological transparency** is maintained without sacrificing automation

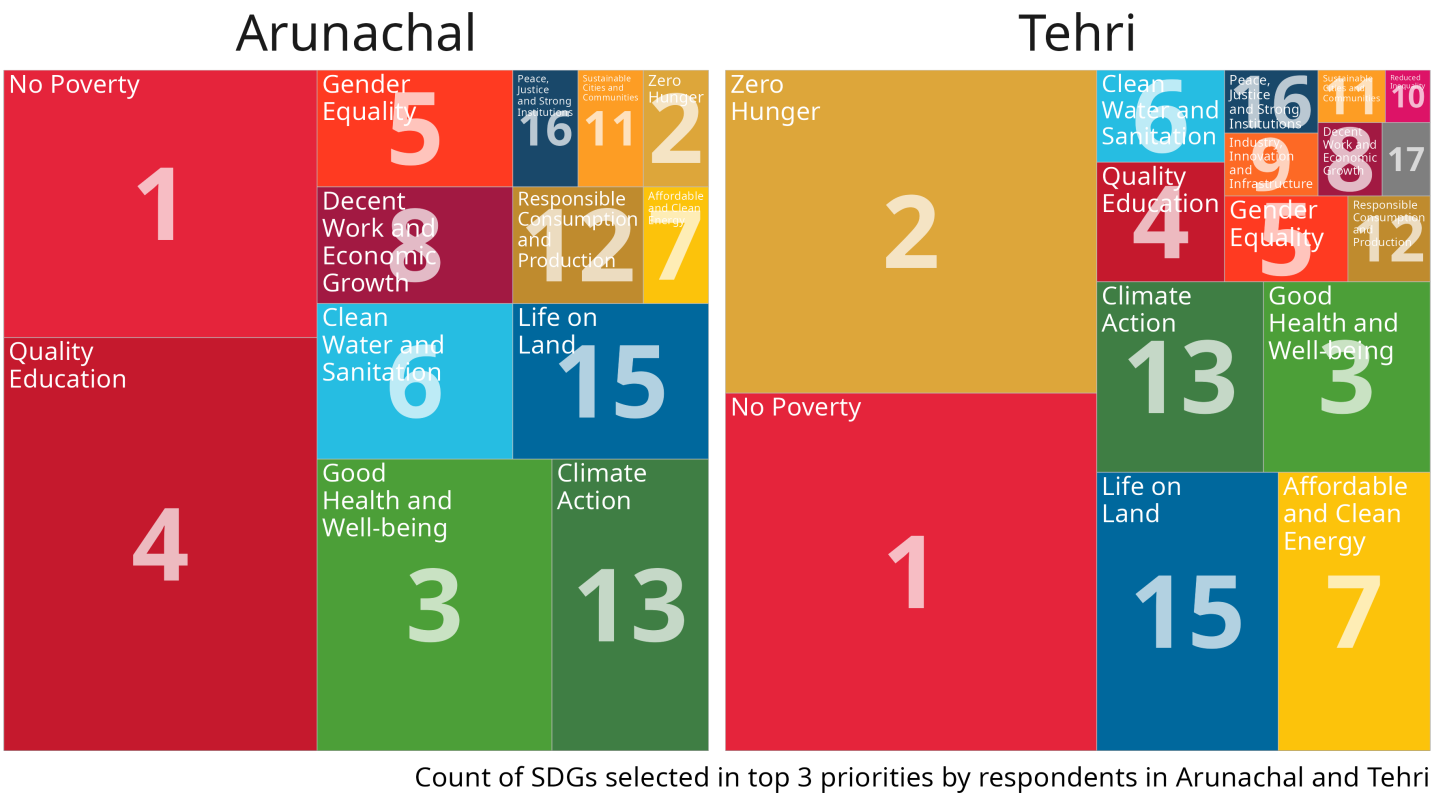

Figure 1: Treemap of top 3 UN SDG priorities from survey respondents in Tehri (dam constructed 20 years ago) and Arunachal Pradesh (dam under development), North India. Each rectangle represents a specific UN SDG, with its size proportional to the number of respondents who ranked it among their top three priorities. The treemap visually highlights the most and least prioritized SDGs in each region, providing insights into regional differences in development priorities.

3.2 Repository architecture supporting semantic design

The giveadam repository is organized into distinct functional areas:

- **raw_data/** — Original survey data files as provided by Garima Gupta
- **data/** — Published, analysis-ready datasets (pipeline outputs)
- **dbt_project/** — Data transformation pipeline with semantic layering
- **observability/** — Automated methodology documentation and quality reports
- **scripts/** — Data preparation and analysis scripts (R, Python)
- **vis-scripts/** — Visualization generation scripts

3.3 Core innovation: Semantic transformation architecture

The central methodological contribution lies in reimagining data transformation layers as conceptual research stages rather than technical processing steps. Standard dbt implementations use staging → intermediate → marts layers optimized for engineering workflows [6]. The semantic layering methodology introduces a fundamentally different paradigm:

- **source_base/** — Raw data integrity and initial cleaning
- **source_entities/** — Context-preserving regional data entities
- **semantic/** — Explicit cross-source harmonization with documented assumptions
- **analytic/** — Research-question-specific datasets ready for analysis

3.3.1 Why semantic layering transforms research data management

Each semantic layer addresses specific research methodology requirements:

1. **Context preservation** — Source entities maintain regional survey characteristics, preventing premature harmonization that obscures methodological differences
2. **Transparent harmonization** — Semantic models explicitly document how different data sources are reconciled (e.g., SDG labeling differences between regions)
3. **Assumption visibility** — Every transformation decision is captured and testable, enabling methodological scrutiny
4. **Stakeholder accessibility** — Layer terminology reflects research concepts (semantic, analytic) rather than engineering jargon (staging, marts)
5. **Iterative extensibility** — New research questions can be addressed by extending the analytic layer without disrupting upstream logic

This approach resolves the fundamental tension between automated data processing and research transparency, enabling both computational efficiency and methodological rigor.

3.4 Automated observability for research transparency

The observability framework generates this methodology documentation automatically from pipeline artifacts, ensuring that:

- Documentation remains synchronized with actual transformations
- All data quality assumptions are explicitly validated and reported
- Research methodology is fully reproducible from code
- Stakeholders can verify data processing decisions without technical expertise

The complete automated documentation process includes both methodology generation and data publication. The validation tables in this document are generated through the automated process detailed in Section 6.3. Additionally, `scripts/publish_data.R` extracts the final analytic models from the dbt pipeline and exports them as CSV files to the `data/` directory for publication and external use.

The validation tables presented in this document are high-level summaries. Engineers or researchers requiring deeper pipeline investigation can use `dbt docs generate` and `dbt docs serve` for an interactive exploration (e.g., Figure 2) of the complete data lineage (Figure 4), including detailed model specifications, column-level lineage, and test results.

4 FAIR principles implementation

The giveadam project implements all four FAIR principles [18] through its semantic layering methodology and automated observability framework detailed in Section 3. This section demonstrates how the semantic approach enhances traditional FAIR compliance by making research processes themselves findable, accessible, interoperable, and reusable.

4.1 Findable: Discovery and identification

Rich metadata and identifiers: The datasets are published with comprehensive metadata in `data/README.md`, including detailed column descriptions, data provenance, and research context. Each dataset has unique identifiers (`respondents.csv`, `SDG_rankings.csv`) and is version-controlled in the GitHub repository `giveadam` at `https://github.com/softloud/giveadam`.

Semantic documentation: Unlike standard data repositories, the semantic layering approach provides findable documentation at multiple conceptual levels. Researchers can locate relevant transformations by research concept (semantic models in Table 1) rather than technical implementation details. The four-layer architecture shown in Figure 4 enables discovery through conceptual navigation.

Automated cataloging: The observability framework generates searchable metadata from pipeline artifacts (Table 1), ensuring that all data transformations and quality checks are discoverable through the automated documentation system detailed in Section 6.3.

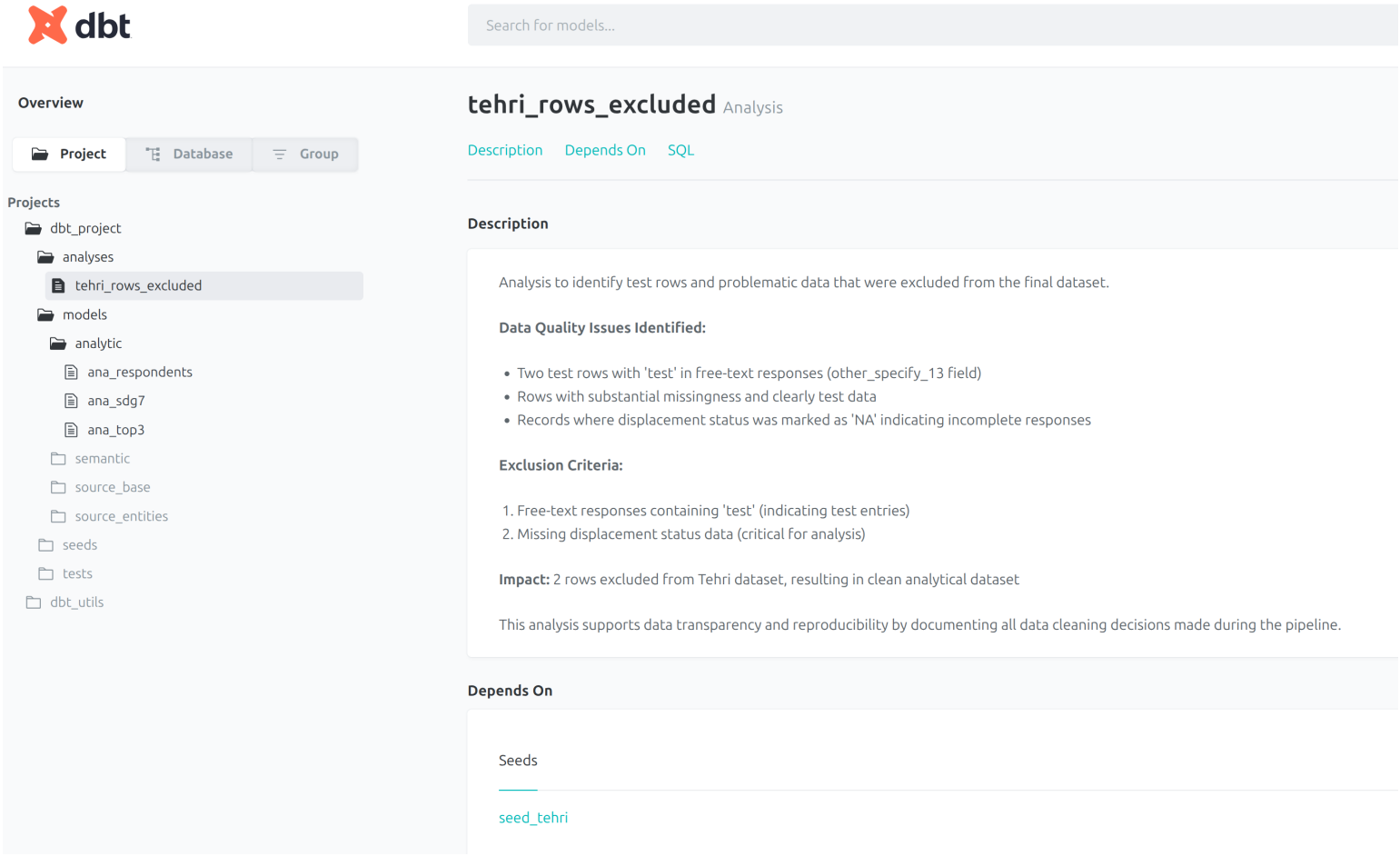

Figure 2: Screenshot from interactive dbt documentation showing the SQL logic excluding two test rows from the Tehri dataset. This exclusion is fully documented in `dbt_project/analyses/tehri_rows_excluded.sql` and reflects the methodological transparency principles outlined in Section 3. DBT documentation provides a user-friendly interface for engineers and stakeholders to verify data processing decisions without requiring technical expertise.

4.2 Accessible: Retrieval and usability

Open access and standard protocols: Data are published in universally accessible CSV format [16] without authentication barriers. The complete repository is openly available via HTTPS and Git protocols [2], supporting both web browser access and programmatic retrieval through GitHub [8].

Human and machine readable: Column names follow interpretable conventions prioritizing domain understanding over technical convenience. The semantic layer architecture (Figure 4) ensures that data structure reflects research logic rather than processing efficiency.

Multiple access modalities: Researchers can access data through multiple pathways: direct CSV download, Git repository cloning [2], or programmatic URL access in R [15] or Python [14]. The interactive dbt documentation (`dbt docs serve`) [5] provides a web-based interface for exploring complete data lineage.

4.2.1 SDG rankings dataset

This dataset contains respondent rankings enriched with demographic metadata:

- **id_respondent:** Unique identifier for each respondent
- **rank:** Priority ranking (1=highest, 2=medium, 3=lowest priority)
- **sdg_number:** UN SDG number (1-17)
- **sdg_label:** Full name of the Sustainable Development Goal
- **age:** Age of respondent in years
- **gender:** Self-reported gender

- **displacement_status:** Dam impact classification
- **region:** Survey location (tehri, arunachal)

By enriching the responses with respondent metadata, we can analyse responses in the context of respondent demographics and characteristics. For example, in Figure 3, we can see the distribution of top 3 UN SDG priorities by gender.

4.2.2 Respondents dataset

- **id_respondent:** Unique identifier for each respondent
- **age:** Age of respondent in years
- **gender:** Self-reported gender (Male, Female, Prefer not to say)
- **displacement_status:** Impact classification related to dam construction
- **region:** Survey location (tehri, arunachal)

4.3 Interoperable: Integration and exchange

Standard data formats and vocabularies: Data are published in CSV format [16] using UTF-8 encoding with standardized missing value representation (NA). Column naming follows consistent conventions across datasets, enabling seamless joining and integration across the semantic layers shown in Figure 4.

Semantic harmonization documentation: The semantic layer explicitly addresses interoperability challenges by documenting how disparate data sources are harmonized. For example, SDG labeling differences between regional surveys are reconciled with full documentation of mapping decisions in the semantic models (Table 1), enabling other researchers to understand and adapt the harmonization logic.

Modular pipeline architecture: The dbt project structure (Section 3.2) separates concerns across semantic layers, enabling selective reuse of transformation logic. Researchers can adopt the source entity patterns for context preservation while modifying semantic harmonization for different research domains. Each semantic layer (Table 1) implements specific interoperability functions that can be independently understood and modified.

Cross-tool compatibility: Tidy data principles ensure compatibility across analytical software. The semantic layer structure provides conceptual interoperability—researchers can understand and adapt the methodology regardless of their technical implementation preferences.

4.4 Reusable: Extension and adaptation

Comprehensive provenance and documentation: Complete methodology documentation enables confident reuse across research contexts. The automated observability system (Section 6.3) ensures that documentation evolves with implementation, preventing methodology drift that undermines reusability.

Extensible semantic framework: The semantic layering methodology provides a reusable framework beyond this specific dataset. The four-layer architecture (source base → source entities → semantic → analytic) demonstrated in Figure 4 can be adapted for any multi-source research data integration challenge.

Quality assurance infrastructure: The validation framework (Table 2) provides reusable patterns for data quality verification across research contexts. These automated tests ensure that reused components maintain data integrity standards.

Licensing and attribution: Data are licensed under Creative Commons Attribution 4.0 International (CC BY 4.0) [4], enabling broad reuse with appropriate attribution. The license covers both datasets and methodology, encouraging adaptation of the semantic layering approach.

Technical infrastructure reusability: The repository can be forked and the dbt pipeline extended for new data sources or research questions. The modular structure supports iterative development—researchers can extend the analytic layer for new questions without modifying upstream transformations.

Methodological transferability: The semantic layering approach addresses fundamental challenges in research data management that extend beyond this specific domain. The methodology's emphasis on context preservation, transparent harmonization, and stakeholder communication applies to any research requiring multi-source data integration and methodological transparency.

5 Data lineage

5.1 Source data and transformation overview

The giveadam project begins with two regional survey datasets stored in `raw_data/`. These datasets required integration across different formats: CSV files [16] from Arunachal Pradesh and Excel exports from the Kobo Toolbox [10] platform for the Tehri region.

To unify these disparate sources, we implemented a comprehensive data transformation pipeline using the Data Build Tool (dbt) [5, 12] within the `dbt_project/` directory. The complete repository architecture supporting this pipeline is detailed in Section 3.2.

5.2 Pipeline architecture and processing

The transformation pipeline consists of SQL scripts [3], automated tests, and comprehensive documentation that convert raw survey data into analysis-ready formats. A significant challenge involved extracting relevant columns from the complex Kobo export structure. We developed a dedicated R script [15, 17] (`scripts/tehri-cols.R`) to identify and map survey questions to data columns, with results documented in `figure_and_tables/tehri_columns.md` [9].

5.2.1 Data quality assurance

Our quality assurance process includes automated identification and exclusion of invalid responses. During processing, we identified and removed two test rows from the Tehri dataset that exhibited substantial missingness and clear indicators of being test data rather than genuine survey responses. This exclusion process is fully documented in `dbt_project/analyses/tehri_rows_excluded.sql` and reflects the methodological transparency principles outlined in Section 3.

5.3 Visual data flow representation

Figure 4 presents the directed acyclic graph (DAG) generated by dbt for the complete transformation pipeline. This visualization serves multiple purposes:

- **Dependency mapping:** Each node represents a data model or transformation step, while edges show the flow of dependencies between them
- **Layer visualization:** The graph clearly demonstrates the four-layer semantic architecture described in Section 6.1
- **Quality assurance:** The DAG shows how data validation and testing are integrated throughout the transformation process
- **Reproducibility:** The complete lineage enables full reproduction of any analytic dataset from source data

This architectural approach ensures data integrity and methodological transparency while enabling efficient processing and clear stakeholder communication.

5.4 Model catalog and validation

Table 1 provides a comprehensive catalog of all dbt models created within the `dbt_project/`, organized by semantic layer. These models implement the semantic extension of dbt's standard layering approach detailed in Section 3.3. Each model serves a specific role in the semantic transformation process, from preserving source context to enabling research-ready analysis.

The validation framework supporting these models is documented in Table 2, which details the automated quality checks applied at each transformation stage. This combination of systematic modeling and rigorous testing ensures both computational efficiency and methodological rigor.

6 Semantic layer implementation and observability

The observability framework operationalizes the semantic layering methodology through automated monitoring and validation at each conceptual stage. This implementation demonstrates how semantic design principles translate into technical infrastructure while maintaining research transparency.

6.1 Semantic layers as methodological framework

The four-layer semantic architecture serves as both a conceptual framework and technical implementation. Each layer embodies specific research methodology principles:

- **Source Base Models:** Embody the principle of data integrity preservation. These foundational models maintain fidelity to original data sources while implementing only essential cleaning operations. By materializing as seeds, they create an immutable record of processed raw data, enabling complete methodological auditing.
- **Source Entities:** Operationalize the context preservation principle. Each regional dataset maintains its original structure and labeling conventions, preventing premature harmonization. This layer enables validation of regional data characteristics and supports comparative analysis of data collection methodologies.

Table 1: Data Build Tool (DBT) models created by dbt_project, ordered by observability layer.

Model	Description
Analytic Models	
ana_respondents	Cleaned and transformed respondents data for analysis.
ana_sdg	UN SDG labels and numbers as identifiers.
ana_sdg7	Dataset reporting if SDG 7 was prioritised in top 3 ranking by respondents, with respondent metadata.
ana_top3	Top 3 analytical model for key metrics
Semantic Models	
sem_respondents	Each row represents one respondent in either the survey undertaken in Tehri or the survey undertaken in Arunachale Pradesh.
sem_sdg_labels	This is a transformation table to integrate SDG labelling across the two survey datasets.
sem_top3	Each row a ranking of top 3 SDG priorities, by one respondent from either the Tehri or Arunachal survey. In particular this step ensures that the SDG labels are harmonised across the two surveys.
sem_top3_arunachal_long	Long format of top 3 SDG rankings from Arunachal Pradesh respondents.
sem_top3_tehri_long	Each row represents a respondent's ranking of their top 3 UN Sustainable Development Goals (SDGs) for Tehri, India.
Source Entity Models	
se_respondents_arunachal	Each row represents a respondent from Arunachal Pradesh.
se_respondents_tehri	Each row represents a respondent from Tehri.
se_top3_arunachal	Each row represents a respondent's ranking top 3 UN Sustainable Development Goals (SDGs) for Arunachal Pradesh, India. These data are extracted in wide format from the original survey data preserving labelling for data validation.
se_top3_tehri	Each row represents a respondent's ranking of their top 3 UN Sustainable Development Goals (SDGs) for Tehri, India. These data are extracted in wide format from the original survey data preserving labelling for data validation.
Base Models	
base_arunachal	Raw data from survey in Arunachal Pradesh, collected by Dr Garima Gupta. Each row represents the survey responses of a single respondent. This step loads the data into the pipeline and establishes an identifier for each respondent. Only top 5 ranking and age and gender in these data, an extraction from the full arunachal dataset which is not available due to institutional restrictions. For this pipeline the top 3 rankings are extracted for analysis.
base_tehri	Raw data from survey in Tehri, collected by Dr Garima Gupta. Each row represents the survey responses of a single respondent. This step loads the data into the pipeline and establishes an identifier for each respondent. See 'raw_data/tehri_cols.md' for details on all questions included in this dataset.

- **Semantic Models:** Implement the transparent harmonization principle. This layer explicitly documents how cross-source differences are resolved, such as reconciling disparate SDG labeling systems. All harmonization logic is encoded in SQL with accompanying documentation, making methodological decisions auditable and modifiable.
- **Analytic Models:** Realize the research-readiness principle. These models combine harmonized data with enriched metadata to directly support specific research questions. By materializing as tables, they optimize performance for iterative analysis while maintaining full lineage to upstream decisions.

6.1.1 Methodological advantages of semantic layering

This semantic architecture delivers specific methodological benefits that standard data engineering approaches cannot provide:

1. **Assumption explicitness:** Every harmonization decision is documented and testable
2. **Context preservation:** Regional and methodological differences are maintained until explicitly resolved
3. **Stakeholder communication:** Layer names and logic reflect research concepts rather than technical implementation
4. **Methodological auditability:** Complete transformation lineage enables scholarly review and replication
5. **Iterative extensibility:** New research directions can be accommodated without fundamental restructuring

6.2 Interactive data exploration

The validation tables presented in this document provide high-level summaries of data quality and pipeline structure. For comprehensive pipeline investigation, stakeholders can access the full interactive documentation using:

```
cd dbt_project/
dbt docs generate
dbt docs serve
```

This provides an interactive web interface showing detailed model specifications, column-level lineage, test results, and complete data flow visualization.

6.3 Observability table generation

The validation tables presented in this document (Tables 1 and 2) are automatically generated from dbt artifacts to ensure methodology documentation remains synchronized with the actual pipeline implementation.

6.3.1 Automated metadata extraction

The observability tables are generated through the following automated process:

1. **dbt artifacts generation** — Running `dbt build` produces `manifest.json` and `run_results.json` containing complete pipeline metadata
2. **Python extraction** — `create-obs-tables/get-obs-dat.py` extracts model descriptions and test results from JSON artifacts
3. **R formatting** — `create-obs-tables/obs-table.R` formats extracted data into LaTeX tables
4. **Document compilation** — LaTeX tables are included in this methodology document via `\input` commands

This automation ensures that any changes to model descriptions, test specifications, or pipeline structure are immediately reflected in the methodology documentation, maintaining complete transparency between implementation and documentation.

6.3.2 Model materializations

The semantic layers employ different dbt materializations optimized for their function:

- **Source base models** — Materialized as `seeds` (CSV files loaded directly into DuckDB)
- **Source entity models** — Materialized as `views` to preserve disk space while maintaining fast access for downstream models
- **Semantic models** — Materialized as `views` to enable flexible harmonization logic without storage overhead
- **Analytic models** — Materialized as `tables` to optimize query performance for research analysis and data export

The progression from views to tables reflects the increasing stability and query frequency of models as they approach the analytical layer. Source and semantic layers prioritize flexibility and maintainability through views, while analytic models prioritize performance through table materialization for repeated research queries.

6.4 Data validation

Table 2 lists the dbt tests applied to `dbt_project/` transformations, ordered by semantic layer. These tests document the data quality assumptions validated at each transformation step in the lineage (Figure 4). The tests ensure data integrity across the semantic pipeline and provide automated quality assurance for research reproducibility.

7 Use of NLP tools

The giveadam project architecture, semantic data pipeline design, and all data transformations were conceived and implemented by Charles T. Gray. GitHub Copilot [7] was used for documentation editing and selected development operations tasks (supporting JSON manifest parsing scripts), but did not contribute to the fundamental design decisions, data modeling approach, or core analytical implementations. All code development, methodological innovations, and research insights represent the original work of the author.

8 Acknowledgements

The author gratefully thanks Garima Gupta for collecting and sharing the original survey data from Tehri and Arunachal Pradesh, North India. The author also acknowledges the open-source software community for developing the tools that made this project possible, including dbt, DuckDB, R, Python, and their associated libraries.

Table 2: Data Build Tool (DBT) tests applied to dbt_project models, ordered by observability layer.

Model	Test	Columns	Arguments	Result
Analytic Model Tests				
ana_respondents	unique_combination_of_columns	id_respondent	NA	pass
Semantic Model Tests				
Source Entity Tests				
se_respondents_arunachal	accepted_values	gender	Male__Female__Prefer_not_to_say	pass
se_respondents_arunachal	not_null	id_respondent	NA	pass
se_respondents_arunachal	unique	id_respondent	NA	pass
se_respondents_arunachal	unique_combination_of_columns	id_respondent	NA	pass
se_respondents_tehri	accepted_values	gender	Male__Female__Prefer_not_to_say	pass
se_respondents_tehri	not_null	id_respondent	NA	pass
se_respondents_tehri	unique	id_respondent	NA	pass
se_respondents_tehri	unique_combination_of_columns	id_respondent	NA	pass
se_top3_arunachal	accepted_values	rank	1__2__3	pass
se_top3_arunachal	not_null	id_respondent	NA	pass
se_top3_arunachal	unique_combination_of_columns	id_respondent	rank	pass
se_top3_tehri	accepted_values	rank	1__2__3	pass
se_top3_tehri	not_null	id_respondent	NA	pass
se_top3_tehri	unique_combination_of_columns	id_respondent	rank	pass
Base Model Tests				
base_arunachal	unique_combination_of_columns	id_respondent	NA	pass
base_arunachal	unique_combination_of_columns	respondents	NA	pass
base_tehri	not_null	id_respondent	NA	pass
base_tehri	unique	id_respondent	NA	pass
base_tehri	unique_combination_of_columns	id	NA	pass
base_tehri	unique_combination_of_columns	id_respondent	NA	pass

References

[1] Airbyte. What is the modern data stack and why should you care? https://airbyte.com/blog/modern-data-stack, 2023.

[2] Scott Chacon and Ben Straub. *Pro Git*. Apress, 2nd edition, 2014.

[3] Donald D Chamberlin and Raymond F Boyce. Sequel: A structured english query language. In *Proceedings of the 1974 ACM SIGFIDET (now SIGMOD) Workshop on Data Description, Access and Control*, pages 249–264, Ann Arbor, Michigan, 1974. ACM.

[4] Creative Commons. Creative commons attribution 4.0 international license. https://creativecommons.org/licenses/by/4.0/, 2013. CC BY 4.0.

[5] dbt Labs. dbt core: Transform data in your warehouse. https://github.com/dbt-labs/dbt-core, 2024.

[6] dbt Labs. How we structure our dbt projects. https://docs.getdbt.com/best-practices/how-we-structure/1-guide-overview, 2024.

[7] GitHub, Inc. Github copilot. https://github.com/features/copilot, 2024.

[8] GitHub, Inc. Github: Where the world builds software. https://github.com, 2024.

[9] John Gruber. Markdown. https://daringfireball.net/projects/markdown/, 2004.

[10] Harvard Humanitarian Initiative. Kobo toolbox: Data collection tools for challenging environments. https://www.kobotoolbox.org/, 2024. Cambridge, MA: Harvard Humanitarian Initiative.

[11] Leslie K John, George Loewenstein, and Drazen Prelec. Measuring the prevalence of questionable research practices with incentives for truth telling. *Psychological science*, 23(5):524–532, 2012.

[12] MotherDuck and dbt Labs. dbt-duckdb: dbt adapter for duckdb. https://github.com/duckdb/dbt-duckdb, 2024.

[13] BBC News. Covid: Test and trace data glitch caused by excel spreadsheet limitations, 2020.

[14] Python Software Foundation. *Python Language Reference*, 2024. Version 3.12.

[15] R Core Team. R: A language and environment for statistical computing. https://www.R-project.org/, 2024.

[16] Yakov Shafranovich. Common format and mime type for comma-separated values (csv) files. RFC 4180, 2005.

[17] Hadley Wickham, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, Alex Hayes, Lionel Henry, Jim Hester, et al. Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686, 2019.

[18] Mark D Wilkinson, Michel Dumontier, İjsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. The fair guiding principles for scientific data management and stewardship. *Scientific Data*, 3(1):160018, 2016.

[19] Mark Ziemann, Yotam Eren, and Assam El-Osta. Gene name errors are widespread in the scientific literature. *Genome Biology*, 17(1):177, 2016.

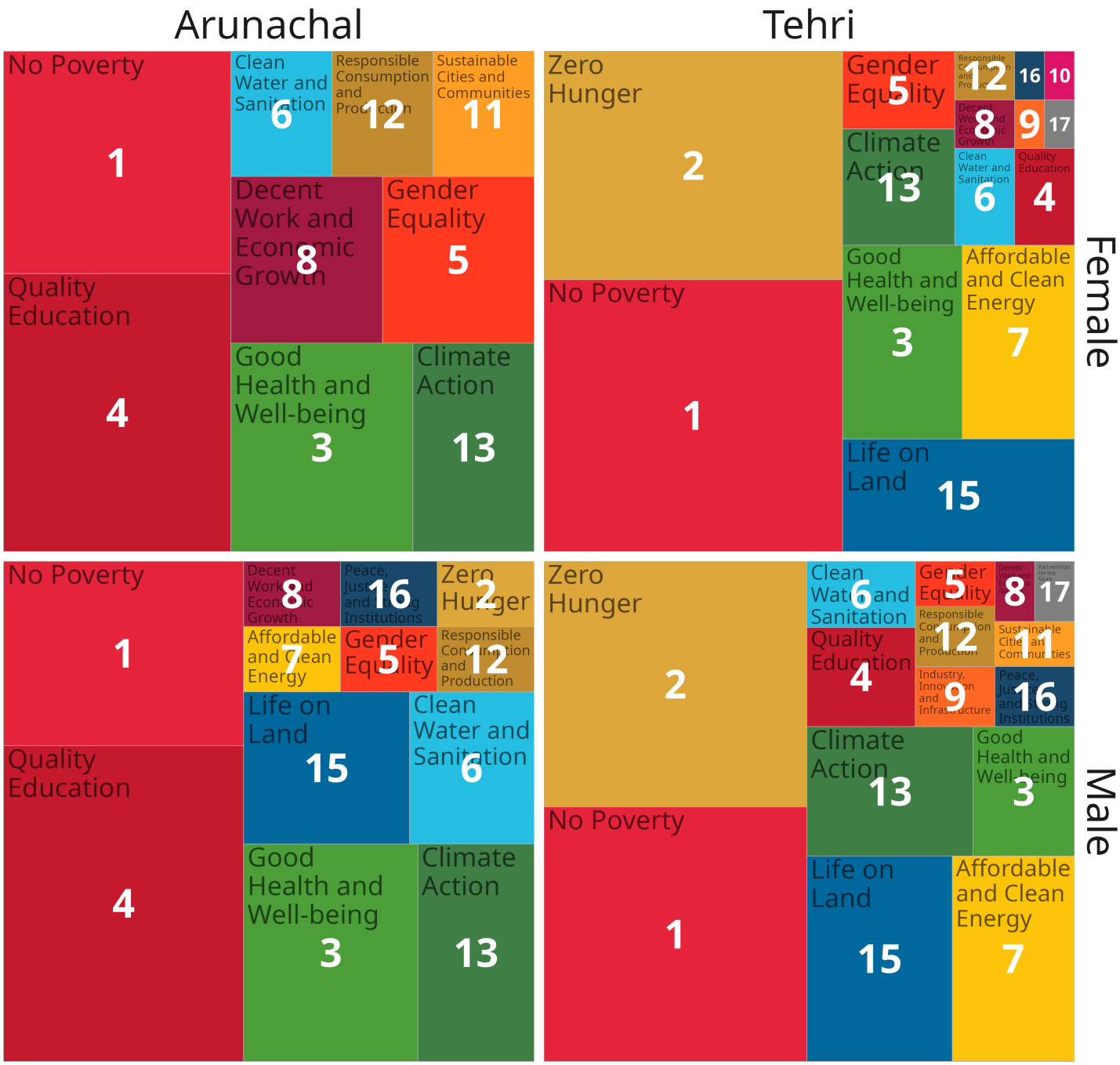

Figure 3: Treemap of top 3 UN SDG priorities from survey respondents in Tehri and Arunachal Pradesh, North India, by gender. Each rectangle represents a specific UN SDG, with its size proportional to the number of respondents of a particular gender who ranked it among their top three priorities. The treemap visually highlights the most and least prioritized SDGs in each region, providing insights into gender differences in development priorities.

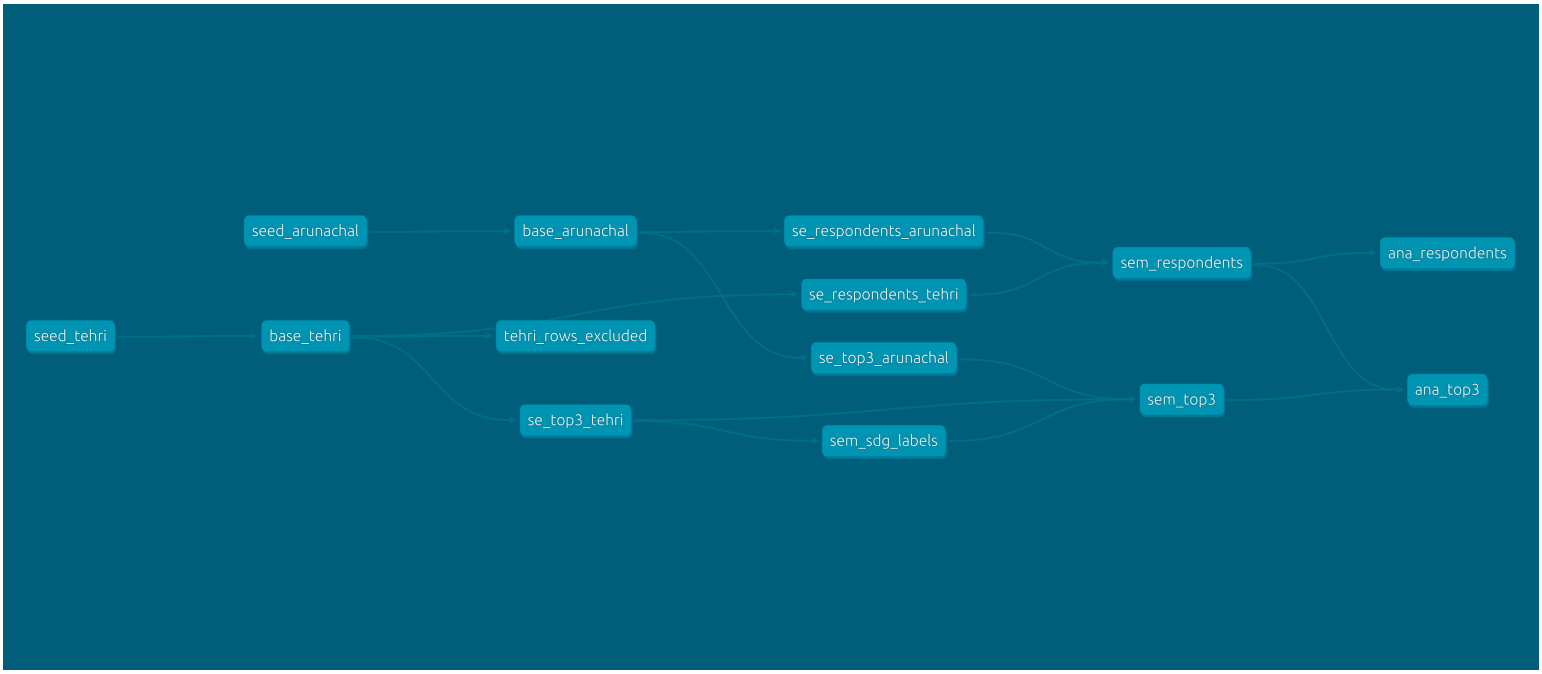

Figure 4: Data Build Tool (dbt) directed acyclic graph (DAG) showing the semantic layer structure from source entities through analytic models.