Data Provenance in the Age of Automation: Lessons from Fifteen Years of Programmatic Access to World Bank Open Data

João Pedro Azevedo
jpazvd.github.io

Abstract. This article examines data provenance in the age of automation, drawing on fifteen years of development and sustained use of `wbopendata`, the first Stata command to provide programmatic access to an international development data repository. Today, `wbopendata` provides access to over 29,000 indicators from 51 databases spanning 296 countries and aggregates, with features including five download modes, multilingual metadata, and publication-ready graph formatting. Its broader contribution lies in treating data acquisition as code: indicator selections, country filters, and time ranges become explicit parameters in analysis scripts rather than undocumented manual downloads—addressing the data provenance dimension of the reproducibility crisis that statistical reforms alone cannot fix. Yet the command has never been more relevant. As AI tools accelerate analytical workflows while enabling plausible fabrication of statistics and citations, anchoring research to authoritative, version-controlled data sources becomes essential infrastructure—not legacy convenience. I document the command's latest syntax and stored results, demonstrate workflows from basic queries to choropleth mapping, and present a 44-scenario test suite. I also discuss risks that frictionless data access can obscure—provenance opacity, coverage gaps masked by convenient defaults, and sustainability pressures—alongside mitigation strategies. I distill three design principles from fifteen years of sustained use: backward compatibility builds trust, domain-specific syntax lowers barriers, and scripted data access makes reproducibility the default rather than the exception.

Keywords: Stata, Open Data, World Bank, APIs, reproducible research, development indicators, metadata

1 Introduction

In April 2010, the World Bank launched its Open Data Initiative (**?**), transforming decades of development statistics into a freely accessible global public good. By coupling an open data portal with a programmatic Application Programming Interface (API), the initiative reframed official statistics from downloadable artifacts into queryable services. This shift altered not only how data are disseminated, but how empirical research can be conducted.

Within less than a year, `wbopendata` (**?**) was released as a Stata command translating this new API into a domain-specific interface for applied researchers. Over the subsequent fifteen years, both the World Bank's data infrastructure and `wbopendata`

evolved substantially: the number of databases expanded, legacy endpoints were retired, metadata standards matured, and new requirements emerged around multilingual documentation, versioning, and reproducibility. Throughout these changes, `wbopendata` maintained backward compatibility while embedding increasingly sophisticated mechanisms for scripted data access, metadata management, and automated documentation.

This article advances three claims. First, a substantial share of the reproducibility crisis in the social sciences stems not from statistical methodology, but from opaque data acquisition practices. Manual downloads, undocumented filters, and ad hoc preprocessing sever the link between published results and their underlying data sources. Second, treating data acquisition as code—where indicator selection, country coverage, and time ranges are explicit, parameterized, and executable—offers a practical response to this problem. `wbopendata` operationalizes this principle by making data provenance an integral component of the analytical script rather than an external narrative. Third, in an era of rapidly expanding AI-assisted research, such constraints are becoming more, not less, important. As generative tools lower the cost of producing plausible analyses and narratives, anchoring empirical work to authoritative and verifiable data sources becomes essential infrastructure for scientific credibility.

Against this backdrop, `wbopendata` is best understood not as a convenience tool, but as a constraint mechanism. By binding analysis to a single authoritative API and exposing all data selection decisions as executable code, it limits the space of admissible data inputs in ways that complement—but also discipline—AI-assisted workflows. The command ensures that acceleration downstream does not come at the cost of unverifiable or opaque data upstream.

The remainder of the article documents the current capabilities of `wbopendata`, demonstrates reproducible analytical workflows, and reflects on lessons learned from fifteen years of sustained use. Section **??** introduces the design principles and scope of the command. Section **??** documents its syntax and stored results. Section **??** presents canonical workflows for reproducible analysis and visualization. Section **??** describes the technical implementation and test suite. Section **??** discusses broader implications for reproducibility in the context of AI-assisted research, and Section **??** concludes.

2 Design principles and scope

The fifteen-year evolution of `wbopendata` reflects design choices shaped by sustained use in applied research environments rather than abstract software engineering goals. These choices respond to practical constraints common in work with official statistics: heterogeneous upstream producers, revisions to published series, institutional computing restrictions, and the need to document provenance transparently. This section distills three design principles embedded in the command and clarifies its intended scope.

2.1 Data acquisition as code

The foundational principle of `wbopendata` is that data acquisition should be treated as part of the analytical codebase rather than as an external preparatory step. Indicator selection, country coverage, time ranges, and filters are expressed as explicit command parameters that can be executed, inspected, and version-controlled alongside the analysis itself.

This approach contrasts with common workflows in which researchers manually download spreadsheets from web portals, rename files, apply undocumented filters, and import the results into statistical software. Such workflows often leave no complete record of how the analytical dataset was constructed, making replication difficult even when code and data files are shared. By encoding acquisition decisions directly in Stata syntax, `wbopendata` produces an executable specification of provenance: a command line documents what data were requested, from which source, and under what constraints.

This principle does not guarantee identical data across time. Instead, it guarantees traceability. If upstream data are revised, the same command yields systematically updated results rather than silently diverging datasets. Reproducibility, in this sense, includes both exact replication (when sources are unchanged) and transparent updating (when new observations or revisions are introduced).

2.2 Backward compatibility as trust infrastructure

A second guiding principle is the prioritization of backward compatibility. Over the past decade, the World Bank API has undergone substantial changes, including endpoint deprecations, catalog restructuring, and metadata revisions. Throughout these transitions, `wbopendata` has aimed to preserve stable syntax and predictable behavior wherever possible.

Backward compatibility serves a methodological function beyond user convenience. Researchers build analytical pipelines that may be rerun years after initial publication, often by different teams. When data access commands change semantics or fail without warning, reproducibility is compromised even if the original code is preserved. By absorbing API changes within the command's internal logic—rather than propagating them to the user interface—`wbopendata` reduces the risk that historical analyses become irreproducible due to infrastructure drift.

This commitment imposes constraints. New features are introduced cautiously, defaults are chosen conservatively, and deprecated behavior is handled through informative diagnostics. The result is a command that evolves incrementally while maintaining continuity, fostering long-term trust among users who rely on it as part of production workflows.

2.3 Domain-specific syntax as error prevention

The third design principle is the use of domain-specific syntax tailored to applied development research. Rather than exposing users directly to HTTP requests, JSON parsing, pagination logic, or API schemas, `wbopendata` presents an interface organized around concepts familiar to its audience: indicators, countries, years, topics, and metadata.

This choice is not merely ergonomic. By constraining user input to semantically meaningful parameters, the command reduces opportunities for error and misinterpretation. Indicator codes must be valid; country identifiers must conform to documented standards; time ranges are explicit. These constraints limit the space of admissible queries and make incorrect or ambiguous data requests easier to detect.

In this sense, domain-specific syntax functions as a guardrail. While generic API clients offer maximal flexibility, they also place the burden of correctness entirely on the user. `wbopendata` trades some generality for a higher likelihood that queries correspond to interpretable, well-documented data products.

2.4 Scope, boundaries, and diffusion

Scope and non-goals

Clarifying scope is essential to understanding the role of `wbopendata` in reproducible research. The command is designed to provide programmatic access to *aggregated* development indicators disseminated through the World Bank API, together with their associated metadata. It is not a tool for accessing confidential or licensed microdata, nor does it perform statistical harmonization, imputation, or methodological adjustments beyond those already embodied in the source databases.

Similarly, `wbopendata` does not aim to abstract away substantive judgment. Choices about indicator suitability, methodological breaks, coverage gaps, and interpretation remain the responsibility of the researcher. The command supports these judgments by surfacing metadata and coverage diagnostics, but it does not substitute for domain expertise.

Within these boundaries, `wbopendata` is best understood as infrastructure rather than analysis software. Its contribution lies in constraining and documenting the earliest stage of the empirical workflow—data acquisition—so that subsequent analysis, whether conducted manually or assisted by automated tools, rests on verifiable and reproducible foundations.

Diffusion beyond `wbopendata`

The design principles outlined above are not unique to `wbopendata`. They reflect a broader pattern of tooling that has emerged within applied development research to address reproducibility, provenance, and scale. In particular, similar principles have guided the development of other Stata-based data access tools, both within and beyond

the World Bank.

Within the World Bank, these ideas informed the evolution of internal data infrastructure such as `datalib`, `datalib2`, and `datalibweb` (**?**), which provide structured, version-controlled access to raw and harmonized household survey microdata. While differing substantially in scope and access restrictions from `wbopendata`, these tools apply the same core logic: data retrieval is scripted, parameterized, and embedded directly in analytical workflows. This approach has enabled large-scale, reproducible analytical systems, most notably the Poverty and Inequality Platform (PIP) (**?**), a publicly accessible global platform that disseminates official poverty and inequality estimates produced through transparent and replicable pipelines.

Outside the World Bank, related principles can be observed in user-written Stata tools such as `datazoom` (**?**), which systematizes the preparation of Brazilian household survey microdata produced by the national statistical agency. Although `datazoom` operates on locally obtained files rather than remote APIs, it similarly emphasizes standardized structure, transparent preprocessing, and reusable code as prerequisites for credible empirical analysis.

More recently, the same design logic has been extended to multi-language data access through `unicefData` (**?**), which provides a triangulated suite of R, Python, and Stata interfaces to UNICEF's SDMX-based data warehouse. While implemented in different programming environments, these tools share a common emphasis on explicit data acquisition, metadata exposure, and reproducible defaults.

These examples are not intended as an exhaustive survey of data infrastructure tools, nor do they imply architectural equivalence. Rather, they illustrate that the principles embedded in `wbopendata` have proven portable across institutions, data modalities, and governance contexts. Their recurrence suggests that treating data access as constrained, executable infrastructure—rather than an informal preprocessing step—is a generalizable response to the reproducibility challenges facing empirical research with official statistics.

3 The wbopendata command

The databases accessible through `wbopendata` include World Development Indicators (WDI), Doing Business, Worldwide Governance Indicators, International Debt Statistics, Africa Development Indicators, Education Statistics, Enterprise Surveys, Gender Statistics, Health Nutrition and Population Statistics, Global Financial Inclusion (Findex), Poverty and Equity, Human Capital Index, Sustainable Development Goals, and many more. Table **??** summarizes the current scope.

Table 1: World Bank Open Data coverage

Dimension	Coverage
Indicators	29,000+
Data sources	51 databases
Topic categories	21
Countries & regions	296
Country attributes	17
Time coverage	1960–present
Languages	3 (English, Spanish, French)

`wbopendata` talks directly to the World Bank API (JSON over HTTP), returns tidy Stata datasets, and caches results to minimize repeat downloads. Five pull modes cover country, topic, single-indicator (all countries), single-indicator (selected countries), and multi-indicator requests. Output may be wide or long; `latest` works in long mode and returns ready-to-use scalars for titles and subtitles. Metadata is always fetched; v17.7.1 adds basic country context (region, admin region, income level, lending type) by default.

3.1 Syntax

`wbopendata, { indicator(`*string*`) | country(`*string*`) | topics(`*string*`) }`
`[ options ]`

Data selection

Exactly one of the following is required:

`indicator(`*string*`)` specifies one or more World Bank indicator codes. Multiple indicators can be requested by separating codes with semicolons, for example, `indicator(SP.POP.TOTL;NY.GDP.PCAP`

`country(`*string*`)` specifies ISO3 country codes or World Bank region codes to retrieve all available indicators for selected countries. Multiple codes can be separated by semicolons.

`topics(#)` specifies a topic ID (1–21) to retrieve all indicators within a thematic category such as education, health, or environment.

Time and language

`year(`*string*`)` restricts the time interval. For example, `year(2000:2020)` returns data only for years 2000 through 2020.

`language(`*string*`)` sets the language for metadata display. Valid codes are `en` (English, default), `es` (Spanish), and `fr` (French).

`projection` accesses population estimates and projections from the Health Nutrition and Population Statistics database rather than actual census data.

Output format

`long` returns data in long format with one row per country-year. The default is wide format with year-specific columns (`yr1960`, `yr1961`, ...).

`clear` replaces any data currently in memory. Required if data are already loaded.

`latest` keeps only the most recent non-missing observation per country. Requires the `long` option. When multiple indicators are requested, retains only observations where *all* indicators have non-missing values in the same year.

`describe` displays indicator metadata without downloading data. Useful for exploring indicator definitions and sources before committing to a full download.

`nometadata` suppresses the metadata display that normally appears after data retrieval.

Country attributes

`basic` adds region, administrative region, income level, and lending type variables to the downloaded data. This is the default behavior in v17.7.1+.

`nobasic` suppresses the default country attribute variables.

`full` adds all 17 country attributes including geographic coordinates and capital city. See Table **??** for the complete list.

`geo`, `capital`, `latitude`, `longitude` add specific geographic fields without the full set of attributes.

`match(`*varname*`)` merges country attributes into an existing dataset. The variable *varname* must contain World Bank country codes (ISO3 format).

Metadata management

`update query` displays the vintage dates of locally cached indicator and country metadata.

`update check` compares local metadata against the remote repository and reports whether updates are available.

`update all` downloads fresh metadata from the repository, replacing the local cache.

Graph metadata (v17.7.1+)

`linewrap(`*string*`)` wraps metadata text for use in graph titles. The argument specifies which metadata to wrap: `name`, `description`, `note`, `source`, `topic`, or `all`.

maxlength(#) sets the maximum characters per line for wrapped text. The default is 50.

linewrapformat(*string*) controls the output format: `stack` (stacked lines), `newline` (newline-separated), `nlines` (returns line count), `lines` (returns individual lines), or `all` (returns all formats).

3.2 Stored results

`wbopendata` is an r-class command that stores results in `r()`. These stored results are critical for automation: they allow downstream code to programmatically access indicator metadata, construct dynamic graph titles, and build reproducible pipelines without manual intervention.

Indicator codes and variable names. World Bank indicator codes like `SI.POV.DDAY` contain periods, which Stata does not allow in variable names. The command automatically converts indicator codes to Stata-safe variable names by replacing periods with underscores and converting to lowercase: `SI.POV.DDAY` becomes `si_pov_dday`. Both forms are stored: `r(indicator#)` preserves the original API code for documentation and re-querying, while `r(varname#)` provides the Stata variable name for use in analysis commands.

Indexed versus aggregate returns. Results come in two forms. Indexed returns (`r(varname1)`, `r(varname2)`, ...) store metadata for each indicator separately, enabling indicator-specific labeling and citation. Aggregate returns store combined information: `r(indicator)` contains the full semicolon-separated query string as entered, while `r(name)` contains all variable names as a space-separated list suitable for `foreach` loops or variable lists.

For each requested indicator (indexed by # = 1, 2, ...), the command returns:

Table 2: Stored results

Result	Type	Description
Aggregate returns (always)		
`r(indicator)`	local	Full query string (semicolon-separated)
`r(name)`	local	All Stata variable names (space-separated)
Indexed returns (per indicator, always)		
`r(indicator#)`	local	Original API indicator code (e.g., SI.POV.DDAY)
`r(varname#)`	local	Stata-safe variable name (e.g., si_pov_dday)
`r(varlabel#)`	local	Indicator label from API
`r(source#)`	local	Source database identifier
`r(time#)`	local	Time dimension name
`r(sourcecite#)`	local	Clean organization name (when Note is non-empty)
With `year()` option		
`r(year#)`	local	Year or year range requested
With `latest` option		
`r(latest)`	local	Formatted subtitle string for graphs
`r(latest_ncountries)`	local	Number of countries with data
`r(latest_avgyear)`	local	Average year of observations
`r(latest_year)`	local	Maximum year retained
With `linewrap()` option		
`r(name#_stack)`	local	Wrapped name for `title()`
`r(description#_stack)`	local	Wrapped description for captions
`r(note#_stack)`	local	Wrapped methodological notes
`r(source#_stack)`	local	Wrapped source text
`r(topic#_stack)`	local	Wrapped topic name
`r(*#_nlines)`	scalar	Line count for each field
`r(*#_line1)`, ...	local	Individual wrapped lines

These returns enable fully automated workflows: a script can download data, extract `r(name1_stack)` for the graph title, `r(sourcecite1)` for the source note, and `r(latest)` for a coverage subtitle—all without hardcoding any metadata.

4 Reproducible analytical workflows

This section presents three canonical workflows that illustrate how `wbopendata` supports reproducible empirical research. Rather than exhaustively documenting every option, the workflows are selected to demonstrate the command's core design principles: scripted data acquisition, metadata-driven annotation, and integration into larger analytical pipelines. Extended examples and visualizations appear in Appendix **??**, with diagnostics and QA artifacts in Appendices **??** and **??**.

4.1 Scripted data acquisition and data shape

The most fundamental workflow enabled by `wbopendata` is the scripted retrieval of development indicators with explicit control over data shape. The following command retrieves a single indicator for all countries and returns the data in long format:

```
. wbopendata, indicator(NY.GDP.MKTP.CD) long clear
```

This single line constitutes a complete, executable specification of data provenance. The indicator definition, source database, country coverage, and time dimension are all determined by the API query and documented through the command's stored results. Unlike manual downloads, no implicit filtering or preprocessing occurs outside the script.

The choice between wide and long formats is not cosmetic. Long format aligns naturally with Stata's panel data commands, regression routines, and `bysort` operations, while wide format may be preferable for descriptive or tabular summaries. By making this choice explicit at the point of data acquisition, `wbopendata` ensures that downstream analysis remains transparent and reproducible. Full metadata output associated with this query is reported in Appendix **??**.

4.2 Metadata-driven visualization

A second canonical workflow demonstrates how metadata retrieved programmatically can be used directly to produce publication-ready visualizations. The `latest` and `linewrap()` options enable the command to return machine-readable metadata suitable for graph titles, captions, and source annotations.

The example below retrieves two indicators, retains the most recent comparable observation for each country, and prepares wrapped metadata for use in figures:

```
. wbopendata, indicator(SI.POV.DDAY; SH.DYN.MORT) ///
    long latest linewrap(name description note)
```

The command stores formatted strings in `r()` that can be passed directly to Stata's graphing commands. Figure **??** illustrates this workflow by constructing a scatter plot of poverty headcount ratios against under-five mortality rates, with axis titles, definitions, and source notes populated automatically from the returned metadata.

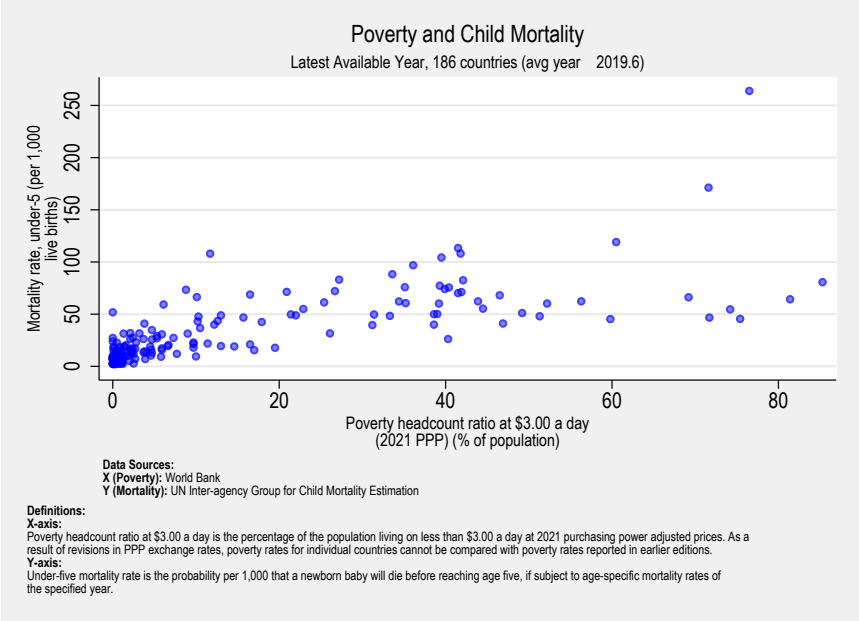

Figure 1: Poverty and child mortality scatter plot with automatic metadata annotation. Axis titles, definitions, and source citations are populated directly from `wbopendata`'s stored results using the `linewrap()` option.

Note: See Appendix **??** for the script used to generate this figure.

By treating metadata as structured output rather than narrative text, `wbopendata` reduces transcription errors and ensures consistency between data and documentation. In particular, the `latest` option records coverage diagnostics—including the number of countries and the average observation year—that can be displayed transparently in figure subtitles. Alternative formatting options and full metadata returns are documented in Appendix **??**.

4.3 Integration into analytical pipelines

Beyond single-command use, `wbopendata` is designed to function as infrastructure within larger analytical pipelines. A prominent example is the World Bank's Learning Poverty project (**?**), which computes global and regional estimates of the share of children unable to read and understand a short text by age ten.

In this pipeline, `wbopendata` is used to retrieve population weights, enrollment rates, and auxiliary indicators from the World Development Indicators database. These inputs are combined with harmonized learning assessment data in a fully scripted workflow released under an open-source license with multiple versioned releases. Because all external data dependencies are accessed programmatically, the resulting estimates can

be independently replicated and systematically updated as new data become available.

This use case illustrates how the design principles described in Section **??** scale beyond exploratory analysis. By constraining data acquisition to explicit, auditable commands, `wbopendata` enables complex, multi-step analytical systems to remain reproducible over time, even as underlying data sources are revised. Additional pipeline examples and related visualizations are provided in Appendix **??**.

5 Technical implementation and reliability

This section documents the technical implementation of `wbopendata` and the mechanisms used to ensure reliability over time. The focus is deliberately operational: how the command is installed, how it interacts with upstream data infrastructure, and how stability is maintained as external APIs evolve.

5.1 Installation

The recommended installation method is via the Statistical Software Components (SSC) archive:

```
. ssc install wbopendata, replace
```

For users who require the latest development version, including recently added metadata-handling features, the package can be installed directly from the public repository:

```
. net install wbopendata, ///
  from("https://raw.githubusercontent.com/jpazvd/wbopendata/main/src") replace
```

Both methods rely exclusively on Stata's native package management system and require no external dependencies.

5.2 Architecture

`wbopendata` is implemented entirely in Stata's ado-file language and communicates with the World Bank's REST API endpoints (**?**). Key architectural features include automatic pagination (retrieval and assembly of multi-page API responses), deterministic caching using hashed request parameters, and systematic normalization of indicator codes into legal Stata variable names.

Metadata parsing and formatting build on established community utilities, including `tknz` (**?**), `linewrap` (**?**), and `_pecats` (**?**). These components support string tokenization, metadata wrapping, and category handling without introducing external binaries.

The implementation avoids third-party executables, ensuring compatibility with institutional computing environments that restrict software installation. Network requests respect Stata's proxy configuration options (e.g., `set httpproxy on`), allowing the com-

mand to operate behind corporate firewalls.

5.3 Country attributes and classification

When invoked with the `full` option, `wbopendata` appends a set of 17 country attributes that support classification, merging, and stratified analysis. Table **??** summarizes these variables.

Table 3: Country attributes returned with `full` option

Variable	Description
`countrycode` / `countryname`	ISO3 code and name
`region` / `regionname`	Region code and name
`adminregion` / `adminregionname`	Administrative region
`incomelevel` / `incomelevelname`	Income classification
`lendingtype` / `lendingtypename`	Lending type (IBRD, IDA, Blend)
`capital`	Capital city name
`latitude` / `longitude`	Capital coordinates

These attributes can also be merged into existing datasets using the `match(varname)` option, where `varname` contains World Bank country codes. This enables enrichment without repeated data downloads.

Regional aggregates are identified explicitly via the `region` variable, allowing users to include or exclude aggregate observations using transparent filters (e.g., `keep if region != "NA"` for individual countries).

5.4 Testing and error handling

To ensure reliability, `wbopendata` ships with a live integration test harness that exercises the same workflows encountered by users in practice. Tests are executed end-to-end against the live World Bank API, covering indicator retrieval, pagination, caching behavior, option handling, and fixes for historical issues.

Because Stata lacks the mature continuous integration and mocking frameworks available in other statistical environments, the test design prioritizes detection of upstream regressions such as schema changes, pagination failures, or API endpoint deprecations. The test suite comprises 44 integrated tests distributed across 13 categories, summarized in Table **??**.

Table 4: Test suite composition (44 tests across 7 categories)

Abbr.	Category	Tests	Focus
ENV	Environment	2	Network connectivity, API availability
DL	Download modes	8	Indicator, country, topic, multi-indicator
FMT	Output format	5	Wide, long, reshape, latest
CTRY	Country options	4	Basic, full, geo, ISO
LW	Linewrap	6	Metadata wrapping, formats
REG	Regression	7	Historical bug fixes
ADV	Advanced	12	Edge cases, error handling

Note: See Appendix **??** for full test definitions and execution logic.

When an indicator code is invalid or deprecated, `wbopendata` returns informative diagnostics that guide users toward corrective action. Verbatim examples of missing indicators, archived series, and update operations are reported in Appendix **??**.

5.5 Metadata management

Local metadata caches are managed through the `update` options. `update query` reports the current vintage, `update check` compares local metadata against the remote repository, and `update all` refreshes cached definitions and country classifications.

Regular metadata updates are particularly important when using the `match()` option, as income groups, regional classifications, and even country names may be revised over time. By making metadata management explicit, `wbopendata` reduces the risk of silently propagating outdated classifications into downstream analysis.

In addition to country classifications and descriptive fields, the set of available indicator codes itself evolves over time. Indicators may be deprecated, archived, or reclassified as source databases are revised, merged, or retired, and metadata associated with existing series may be corrected or updated. `wbopendata` treats these changes as first-class events rather than silent failures. Deprecated indicators trigger explicit diagnostic messages that identify the archival location, while revised metadata is propagated through the local cache when updates are applied.

By making indicator availability and metadata revisions explicit, the command reduces the risk that analytical pipelines continue to rely unknowingly on obsolete series or outdated definitions. Verbatim examples illustrating deprecated indicators and metadata updates are reported in Appendix **??**.

6 Discussion: reproducibility and constraints in the age of AI

Fifteen years after its initial release, `wbopendata` remains relevant not because of any single technical feature, but because of the methodological principles embedded in its design. As discussed in Section **??**, these principles treat data acquisition as executable infrastructure rather than an informal preprocessing step. This section reflects on the implications of that design in the context of contemporary empirical research, particularly as analytical workflows are increasingly mediated by AI-assisted tools.

6.1 Reproducibility beyond statistical methodology

The reproducibility crisis in the social sciences has been widely documented (**??**). Much of the resulting reform agenda has focused on statistical practice: pre-registration, replication studies, robustness checks, and disclosure of code. While these efforts are essential, they address only part of the problem. In many empirical workflows, the construction of the analytical dataset itself remains opaque, relying on manual downloads, undocumented filters, and ad hoc preprocessing that cannot be fully reconstructed from code alone.

By treating data acquisition as code, `wbopendata` addresses this dimension directly. A command specifying indicators, countries, and time ranges constitutes a complete and executable description of how the analytical dataset was assembled. When rerun, the same command either reproduces the original data exactly or yields systematically updated results if upstream sources have been revised. In both cases, the provenance of the data is explicit and auditable. Reproducibility is therefore strengthened not by constraining statistical analysis, but by constraining how data enter the analytical pipeline.

6.2 Constraints as analytical infrastructure

The increasing adoption of AI-assisted tools has altered the balance of effort in empirical research. Tasks that were once time-consuming—coding, visualization, and narrative drafting—can now be performed rapidly. This acceleration, however, amplifies the importance of upstream constraints. When tools can generate plausible-looking statistics, code, and citations, the credibility of empirical work depends increasingly on the verifiability of its inputs.

In this environment, `wbopendata` functions less as a productivity complement and more as a constraint mechanism. By binding analysis to a single, authoritative API and requiring explicit specification of indicators, coverage, and time periods, it limits the space of admissible data inputs. These constraints do not inhibit analysis; rather, they discipline it by ensuring that downstream automation operates on verifiable and well-documented sources. Acceleration is thereby decoupled from fabrication: analytical workflows can scale without relaxing standards of provenance.

This role is consistent with the design principles outlined in Section **??**. Backward compatibility preserves trust across time, domain-specific syntax reduces opportunities for error, and scripted data access makes provenance the default rather than an afterthought. Together, these features position `wbopendata` as infrastructure that shapes how analysis is conducted, rather than merely facilitating it.

6.3 Generalizability and limits

The principles embodied in `wbopendata` are not unique to development indicators or to Stata. As discussed earlier, similar design logic underpins other data-access tools operating on aggregated indicators and microdata, both within and outside the World Bank. Their recurrence suggests that constraining data access through executable specifications is a generalizable response to reproducibility challenges in applied research.

At the same time, important limits remain. Programmatic access does not resolve substantive issues of data quality, coverage gaps, or methodological breaks. Indicators may be outdated, revised, or unavailable for political or technical reasons. Moreover, secure access to confidential microdata continues to require institution-specific governance arrangements that cannot be fully standardized. `wbopendata` does not eliminate these challenges; it makes them visible by exposing metadata, coverage diagnostics, and revision histories directly to the user.

6.4 Implications for open science

The experience of `wbopendata` over fifteen years underscores a broader lesson for open science: principles alone are insufficient without infrastructure that embeds them into routine practice. Transparency and reproducibility are widely endorsed norms, yet they are often undermined by tooling that leaves critical steps implicit. By constraining data acquisition through executable commands, tools like `wbopendata` shift reproducibility from aspiration to default behavior.

In this sense, the contribution of `wbopendata` is methodological rather than technological. It demonstrates that disciplined data access can coexist with flexible analysis, and that constraining the earliest stages of the workflow can enhance, rather than limit, analytical innovation. As empirical research continues to evolve in the presence of increasingly powerful automation, such constraint-based infrastructure will remain essential to maintaining credibility and trust.

7 Conclusion

Fifteen years after the World Bank Open Data Initiative transformed development statistics into a global public good, `wbopendata` continues to function as core infrastructure for reproducible empirical research in Stata. The command provides stable, programmatic access to over 29,000 development indicators from 51 databases, encap-

sulating API communication, pagination, caching, and metadata management within a single, backward-compatible interface.

This longevity reflects deliberate design choices rather than technical inertia. By maintaining backward compatibility across upstream changes, encapsulating complexity behind domain-specific syntax, and treating data acquisition as code rather than manual preprocessing, `wbopendata` shifts reproducibility from an aspirational norm to a default behavior. Version 17.7.1 extends this approach with publication-ready metadata handling, default country-context attributes, and strengthened support for multi-indicator workflows.

The command's pure-Stata implementation ensures compatibility with institutional computing environments, while the bundled live test suite and QA framework demonstrate how reliability can be sustained even when upstream data systems evolve. More broadly, the experience documented here illustrates that reproducible analytics depends not only on statistical methods, but on disciplined data-access infrastructure that makes provenance explicit, testable, and auditable.

In the age of AI-assisted research, this constraint-based approach is increasingly important. As analytical workflows accelerate and the cost of producing plausible statistics and narratives falls, credibility hinges on limiting the space of admissible inputs to authoritative, verifiable sources. `wbopendata` illustrates how such constraints can be embedded directly into routine research practice—supporting scale and automation without relaxing standards of trust.

While developed in Stata and applied to World Bank data, the principles distilled from fifteen years of use—scripted data acquisition, conservative evolution, and operational validation—are not tool-specific. They point toward a broader lesson for empirical research and open science: reproducibility is sustained not by documentation alone, but by infrastructure that enforces it upstream.

Acknowledgments

The author thanks the World Bank Open Data Initiative for making development data freely accessible, and the many users who have contributed bug reports and feature suggestions through GitHub.

8 References

Azevedo, J. P. 2011. WBOPENDATA: Stata module to access World Bank databases. Statistical Software Components S457234, Boston College Department of Economics. Accessed: 2026-01-05. https://ideas.repec.org/c/boc/bocode/s457234.html.

———. 2024. unicefData: Programmatic access to UNICEF SDMX data warehouse. GitHub repository. Interfaces for R, Python, and Stata. https://github.com/unicef-drp/unicefData.

Baker, M. 2016. 1,500 scientists lift the lid on reproducibility. *Nature* 533(7604): 452–454.

Clarke, D. 2012. WORLDSTAT: Stata module to produce World Bank visualizations. *Statistical Software Components* (S457565). https://ideas.repec.org/c/boc/bocode/s457540.html.

Elliott, D. C. 2002. TKNZ: Stata module to tokenize string into named macros. *Statistical Software Components* (S426302). Revised 17 Oct 2006. https://ideas.repec.org/c/boc/bocode/s426302.html.

Long, J. S., and J. Freese. 2001. _PECATS: Utility to determine names and values of categories of dependent variable. *Stata Technical Bulletin* 58(sg155). Part of MLOGTEST package. https://www.stata.com/stb/stb58/sg155/_pecats.hlp.

Open Science Collaboration. 2015. Estimating the reproducibility of psychological science. *Science* 349(6251): aac4716.

Over, M., and J. a. P. Azevedo. 2000. LINEWRAP: Split a long string into shorter strings and, optionally, display them (Version 2.1 4Jun2023) Version 2.1. http://digital.cgdev.org/doc/stata/MO/Misc/linewrap/linewrap.html.

Pisati, M. 2007. SPMAP: Stata module to visualize spatial data. *Statistical Software Components* (S456812). https://ideas.repec.org/c/boc/bocode/s456812.html.

PUC-Rio Department of Economics. 2020. datazoom_social: Stata package for Brazilian household surveys. GitHub repository. Accessed: 2026-01-05. https://github.com/datazoompuc/datazoom_social_Stata.

World Bank. 2010. World Bank Open Data Initiative. Press release. Launched April 2010. https://www.worldbank.org/en/news/press-release/2010/04/20/world-bank-group-opens-data-to-all.

———. 2018. datalibweb: Stata package for accessing harmonized microdata. Internal World Bank infrastructure. Provides version-controlled retrieval of household survey datasets. https://github.com/worldbank/datalibweb.

———. 2019. Learning Poverty: Reproducible Stata codebase for global learning poverty indicators. GitHub repository. MIT License, versions 1.0–4.0. https://github.com/worldbank/LearningPoverty.

———. 2024a. Poverty and Inequality Platform (PIP). https://pip.worldbank.org. Publicly accessible platform providing official global poverty and inequality estimates produced through transparent and replicable analytical pipelines. Accessed January 2026.

———. 2024b. World Bank API Documentation. Developer documentation. https://datahelpdesk.worldbank.org/knowledgebase/topics/125589-developer-information.

About the authors

João Pedro Azevedo is Deputy Director and Chief Statistician in UNICEF's Division of Data, Analytics, Planning and Monitoring. Previously, he worked for 16 years at the World Bank as Lead Economist. His research focuses on poverty measurement, education statistics, and reproducible and scalable analytical pipelines for global, regional, and national monitoring systems to inform policy.

The software repository (https://github.com/jpazvd/wbopendata) includes example do-files, test scripts, and complete documentation.

A Extended examples and visualizations

This appendix provides extended examples, alternative visualizations, and verbatim output referenced in the main text. These materials are included to support full transparency and reproducibility while keeping the exposition in Section **??** focused on canonical analytical workflows.

A.1 Extended data retrieval examples

This subsection reports additional data retrieval examples illustrating variations in indicator selection, output shape, and filtering options. These examples expand on the scripted data acquisition workflow described in Section **??**.

Single-indicator retrieval (verbatim output)

```
. wbopendata, indicator(NY.GDP.MKTP.CD) clear linewrap(name note) maxlength(35 7
      0)

    Metadata for indicator NY.GDP.MKTP.CD
--------------------------------------------------------------------------------
    Name: GDP (current US$)
--------------------------------------------------------------------------------
    Collection: 2 World Development Indicators
--------------------------------------------------------------------------------
    Description: Gross domestic product is the total income earned through
    the production of goods and services in an economic territory during an
    accounting period. It can be measured in three different ways: using
    either the expenditure approach, the income approach, or the production
    approach. This indicator is expressed in current prices, meaning no
    adjustment has been made to account for price changes over time. This
    indicator is expressed in United States dollars.
--------------------------------------------------------------------------------
    Note: Country official statistics, National Statistical Organizations
    and or Central Banks;
--------------------------------------------------------------------------------
    Topic(s):  ; 3 Economy and Growth
```

Multiple-indicator queries (verbatim output)

The following example retrieves multiple indicators simultaneously and returns the data in long format:

```
. wbopendata, indicator(SI.POV.DDAY;NY.GDP.PCAP.PP.KD) clear long ///
      linewrap(name note) maxlength(35 70)

    Metadata for indicator SI.POV.DDAY
--------------------------------------------------------------------------------
    Name: Poverty headcount ratio at $3.00 a day (2021 PPP) (% of
    population)
--------------------------------------------------------------------------------
    Collection: 2 World Development Indicators
--------------------------------------------------------------------------------
    Description: Poverty headcount ratio at $3.00 a day is the percentage of
    the population living on less than $3.00 a day at 2021 purchasing power
    adjusted prices. As a result of revisions in PPP exchange rates, poverty
    rates for individual countries cannot be compared with poverty rates
    reported in earlier editions.
--------------------------------------------------------------------------------
    Note: World Bank, Poverty and Inequality Platform. Data are based on
    primary household survey data obtained from government statistical
    agencies and World Bank country departments. Data for high-income
    economies are mostly from the Luxembourg Income Study database. For more
    information and methodology, please see http://pip.worldbank.org., World
    Bank (WB), uri: http://pip.worldbank.org, note: Data are based on
    primary household survey data obtained from government statistical
    agencies and World Bank country departments. Data for high-income
    economies are mostly from the Luxembourg Income Study database.
--------------------------------------------------------------------------------
    Topic(s): 11 Poverty ; 2 Aid Effectiveness ; 19 Climate Change
--------------------------------------------------------------------------------

    Metadata for indicator NY.GDP.PCAP.PP.KD
--------------------------------------------------------------------------------
    Name: GDP per capita, PPP (constant 2021 international $)
--------------------------------------------------------------------------------
    Collection: 2 World Development Indicators
--------------------------------------------------------------------------------
    Description: This indicator provides values for gross domestic product
    (GDP) expressed in constant international dollars, converted by
    purchasing power parities (PPPs). PPPs account for the different price
    levels across countries and thus PPP-based comparisons of economic
    output are more appropriate for comparing the output of economies and
    the average material well-being of their inhabitants than exchange-rate
    based comparisons.
--------------------------------------------------------------------------------
    Note: .
--------------------------------------------------------------------------------
    Topic(s):
--------------------------------------------------------------------------------

. describe

 Observations:        17,290
    Variables:            13                  5 Jan 2026 14:06
--------------------------------------------------------------------------------
Variable      Storage   Display    Value
    name         type    format    label      Variable label
```

```
------------------------------------------------------------------------------
countrycode     str3    %9s                 Country Code
countryname     str75   %75s                Country Name
region          str3    %9s                 Region Code
regionname      str51   %51s                Region Name
adminregion     str3    %9s                 Administrative Region Code
adminregionname str75   %75s                Administrative Region Name
incomelevel     str3    %9s                 Income Level Code
incomelevelname str19   %19s                Income Level Name
lendingtype     str3    %9s                 Lending Type Code
lendingtypename str14   %14s                Lending Type Name
year            int     %9.0g               Year
si_pov_dday     float   %9.0g               SI.POV.DDAY
ny_gdp_pcap_p~d float   %9.0g               NY.GDP.PCAP.PP.KD
------------------------------------------------------------------------------
Sorted by: countrycode  year
     Note: Dataset has changed since last saved.
```

This pattern is commonly used in multivariate analysis and panel regressions, where alignment across indicators and years is required.

Country- and topic-based queries

Additional examples demonstrate retrieval by country codes and topic identifiers, illustrating how `wbopendata` supports exploratory analysis and bulk downloads without manual file handling.

```
wbopendata, country(BRA; ARG; CHL) long clear
wbopendata, topics(11) clear
```

Full metadata output associated with these queries is reported in Appendix **??** and Appendix **??**.

A.2 Extended metadata output

This subsection reports verbatim metadata returned by `wbopendata`, including indicator names, definitions, sources, and topic classifications. These outputs correspond to examples discussed in Sections **??** and **??**.

Indicator metadata (illustrative excerpt)

```
Metadata for indicator SI.POV.DDAY
-----------------------------------------------------------
Name: Poverty headcount ratio at $3.00 a day (2021 PPP)
Description: ...
Source: World Bank, Poverty and Inequality Platform
Topic(s): Poverty
-----------------------------------------------------------
```

Metadata is retrieved directly from the World Bank API and cached locally to ensure consistency across repeated queries.

Coverage diagnostics with `latest` (verbatim output)

The following output illustrates coverage diagnostics returned by the `latest` option, including the number of countries and the average observation year:

```
. wbopendata, indicator(SI.POV.DDAY) clear long latest ///
      linewrap(name note) maxlength(35 70)

    Metadata for indicator SI.POV.DDAY
-------------------------------------------------------------------------------
    Name: Poverty headcount ratio at $3.00 a day (2021 PPP) (% of
    population)
-------------------------------------------------------------------------------
    Collection: 2 World Development Indicators
-------------------------------------------------------------------------------
    Description: Poverty headcount ratio at $3.00 a day is the percentage of
    the population living on less than $3.00 a day at 2021 purchasing power
    adjusted prices. As a result of revisions in PPP exchange rates, poverty
    rates for individual countries cannot be compared with poverty rates
    reported in earlier editions.
-------------------------------------------------------------------------------
    Note: World Bank, Poverty and Inequality Platform. Data are based on
    primary household survey data obtained from government statistical
    agencies and World Bank country departments. Data for high-income
    economies are mostly from the Luxembourg Income Study database. For more
    information and methodology, please see http://pip.worldbank.org., World
    Bank (WB), uri: http://pip.worldbank.org, note: Data are based on
    primary household survey data obtained from government statistical
    agencies and World Bank country departments. Data for high-income
    economies are mostly from the Luxembourg Income Study database.
-------------------------------------------------------------------------------
    Topic(s): 11 Poverty ; 2 Aid Effectiveness ; 19 Climate Change
-------------------------------------------------------------------------------

. di as text "latest year: "
latest year: 2024

. di as text "countries: "
countries: 186

. di as text "avg year: "
avg year:    2019.8
```

`linewrap()` stored results (verbatim output)

This output shows the wrapped metadata and stored results used for metadata-driven graph annotation:

```
. * Download indicators with linewrap option for graph-ready metadata
. * Returns r(name#_stack), r(description#_stack), r(sourcecite#), r(latest)
. wbopendata, indicator(SI.POV.DDAY; SH.DYN.MORT) clear long latest ///
      linewrap(name description note) maxlength(40 160)

    Metadata for indicator SI.POV.DDAY
```

```
-------------------------------------------------------------------------------
    Name: Poverty headcount ratio at $3.00 a day (2021 PPP) (% of
    population)
-------------------------------------------------------------------------------
    Collection: 2 World Development Indicators
-------------------------------------------------------------------------------
    Description: Poverty headcount ratio at $3.00 a day is the percentage of
    the population living on less than $3.00 a day at 2021 purchasing power
    adjusted prices. As a result of revisions in PPP exchange rates, poverty
    rates for individual countries cannot be compared with poverty rates
    reported in earlier editions.
-------------------------------------------------------------------------------
    Note: World Bank, Poverty and Inequality Platform. Data are based on
    primary household survey data obtained from government statistical
    agencies and World Bank country departments. Data for high-income
    economies are mostly from the Luxembourg Income Study database. For more
    information and methodology, please see http://pip.worldbank.org., World
    Bank (WB), uri: http://pip.worldbank.org, note: Data are based on
    primary household survey data obtained from government statistical
    agencies and World Bank country departments. Data for high-income
    economies are mostly from the Luxembourg Income Study database.
-------------------------------------------------------------------------------
    Topic(s): 11 Poverty ; 2 Aid Effectiveness ; 19 Climate Change
-------------------------------------------------------------------------------

    Metadata for indicator SH.DYN.MORT
-------------------------------------------------------------------------------
    Name: Mortality rate, under-5 (per 1,000 live births)
-------------------------------------------------------------------------------
    Collection: 2 World Development Indicators
-------------------------------------------------------------------------------
    Description: Under-five mortality rate is the probability per 1,000 that
    a newborn baby will die before reaching age five, if subject to
    age-specific mortality rates of the specified year.
-------------------------------------------------------------------------------
    Note: UN Inter-agency Group for Child Mortality Estimation, UN
    Children´s Fund (UNICEF), uri: http://www.childmortality.org, publisher:
    UNICEF, WHO, World Bank, United Nations Population Division;
-------------------------------------------------------------------------------
    Topic(s):
-------------------------------------------------------------------------------

. * Display wrapped metadata available for graph annotations
. return list

macros:
              r(name) : "si_pov_dday  sh_dyn_mort"
       r(latest_year) : "2023"
    r(latest_avgyear) : "   2019.6"
  r(latest_ncountries) : "186"
           r(latest) : "Latest Available Year, 186 countries (avg year    .."
         r(indicator) : "SI.POV.DDAY; SH.DYN.MORT"
             r(time2) : "year"
         r(varlabel2) : "Mortality rate, under-5 (per 1,000 live births)"
           r(source2) : "2 World Development Indicators"
        r(indicator2) : "SH.DYN.MORT"
          r(varname2) : "sh_dyn_mort"
       r(sourcecite2) : "UN Inter-agency Group for Child Mortality Estimati.."
       r(note2_stack) : ""UN Inter-agency Group for Child Mortality Estimat.."
  r(description2_stac
```

```
    k)                 : ""Under-five mortality rate is the probability per .."
        r(name2_stack) : ""Mortality rate, under-5 (per 1,000" "live births)""
              r(time1) : "year"
          r(varlabel1) : "Poverty headcount ratio at $3.00 a day (2021 PPP) .."
            r(source1) : "2 World Development Indicators"
         r(indicator1) : "SI.POV.DDAY"
           r(varname1) : "si_pov_dday"
        r(sourcecite1) : "World Bank"
        r(note1_stack) : ""World Bank, Poverty and Inequality Platform. Data.."
  r(description1_stac
    k)                 : ""Poverty headcount ratio at $3.00 a day is the per.."
        r(name1_stack) : ""Poverty headcount ratio at $3.00 a day" "(2021 PP.."
```

A.3 Extended visualizations

This subsection presents additional figures referenced but not reproduced in the main text. These include alternative mappings, color schemes, and regional breakdowns. Where relevant, the corresponding console output and commands are provided verbatim.

Metadata-driven graph annotation (supporting Figure ??)

```
. * Graph with wrapped axis titles, subtitle, definitions, and sources
. set scheme sj

. twoway (scatter sh_dyn_mort si_pov_dday, msize(small) mcolor(blue%50)), ///
      xtitle("Poverty headcount ratio at $3.00 a day" "(2021 PPP) (% of population)", size(small)) ///
      ytitle("Mortality rate, under-5 (per 1,000" "live births)", size(small)) ///
      title("Poverty and Child Mortality", size(medium)) ///
      subtitle("Latest Available Year, 186 countries (avg year    2019.6)", size(small)) ///
      caption("bf:Definitions:" ///
      "bf:X-axis: " "Poverty headcount ratio at $3.00 a day is the percentage of the population living on less
      "bf:Y-axis: " "Under-five mortality rate is the probability per 1,000 that a newborn baby will die before
      note("bf:Data Sources:" ///
      "bf:X (Poverty): World Bank" ///
      "bf:Y (Mortality): UN Inter-agency Group for Child Mortality Estimation", size(vsmall)) name(tmp1, replac
```

Choropleth mapping example (verbatim output)

```
. * Download indicator data
. tempfile wdi_data
. wbopendata, indicator(it.cel.sets.p2) long clear latest

    Metadata for indicator IT.CEL.SETS.P2
--------------------------------------------------------------------------------
    Name: Mobile cellular subscriptions (per 100 people)
--------------------------------------------------------------------------------
    Collection: 2 World Development Indicators
--------------------------------------------------------------------------------
    Description: Mobile cellular telephone subscriptions are subscriptions
    to a public mobile telephone service that provide access to the PSTN
    using cellular technology. The indicator includes (and is split into)
    the number of postpaid subscriptions, and the number of active prepaid
    accounts (i.e. that have been used during the last three months). The
    indicator applies to all mobile cellular subscriptions that offer voice
```

```
    communications. It excludes subscriptions via data cards or USB modems,
    subscriptions to public mobile data services, private trunked mobile
    radio, telepoint, radio paging and telemetry services.
-------------------------------------------------------------------------------
    Note: World Telecommunication ICT Indicators Database, International
    Telecommunication Union (ITU)
-------------------------------------------------------------------------------
    Topic(s): 9 Infrastructure
-------------------------------------------------------------------------------

. sort countrycode
. * Merge with shapefile coordinates
. use "C:/Users/jpazevedo/ado/plus/w/world-d.dta", clear
. * Create choropleth map
. set scheme sj
. sum year

    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
        year |        262    2022.584    2.400234       2004       2024
. local avg = string(<avg>, "%16.1f")
. spmap it_cel_sets_p2 using "C:/Users/jpazevedo/ado/plus/w/world-c.dta", id(_ID) ///
      clnumber(20) fcolor(Reds2) ocolor(none ..) ///
      title("Mobile cellular subscriptions (per 100 people)", size(*1.2)) ///
      legstyle(3) legend(ring(1) position(3)) ///
      note("Source: World Telecommunication ICT Indicators Database (latest: 2022.6)")
```

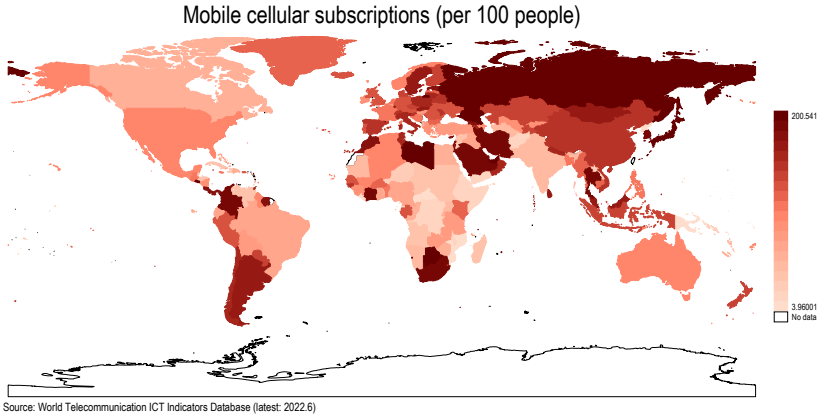

Figure A1: Mobile cellular subscriptions per 100 people (latest available year). Created by merging `wbopendata` output with geographic shape data and visualizing using the `spmap` command (**?**).

Alternative scatter plots (verbatim output)

```
. * Scatter plot: Poverty vs GDP per capita with lowess smoother
. set scheme sj

. graph twoway ///
      (scatter si_pov_dday ny_gdp_pcap_pp_kd, msize(*.3)) ///
```

```
(scatter si_pov_dday ny_gdp_pcap_pp_kd if regionname == "Aggregates", ///
msize(*.8) mlabel(countryname) mlabsize(*.8) mlabangle(25)) ///
(lowess si_pov_dday ny_gdp_pcap_pp_kd), ///
legend(off) ///
ytitle("Poverty headcount ratio at $2.15 a day", size(small)) ///
xtitle("GDP per capita, PPP (constant intl $)", size(small)) ///
note("Source: WDI (latest as of  5 Jan 2026 14:07)")
```

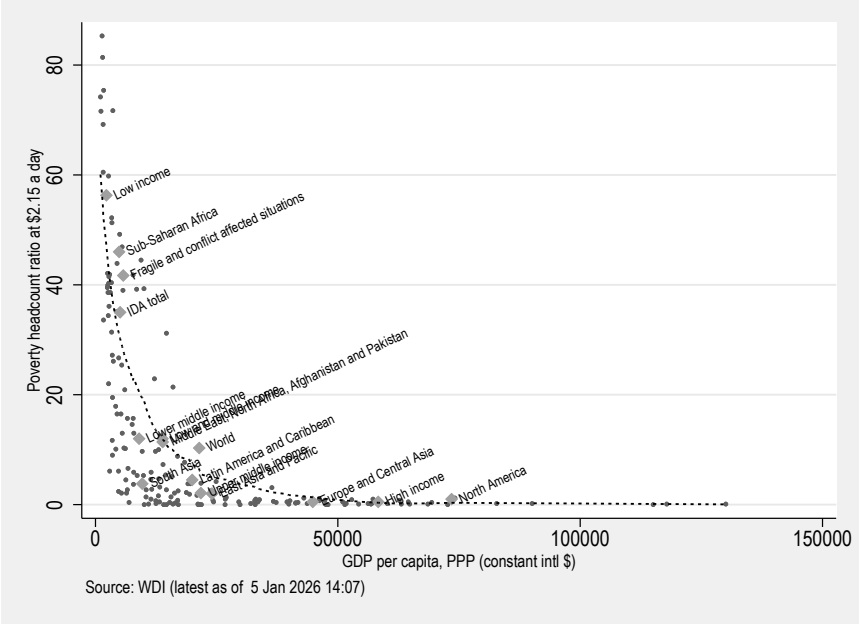

Figure A2: Poverty headcount ratio versus GDP per capita (PPP, constant international $). Regional aggregates are labeled; lowess smoother shows the cross-country relationship.

A.4 User-written extensions

This subsection documents examples using user-written commands that rely on `wbopendata` internally.

`worldstat`: Africa example (verbatim output)

```
. * Regional map: GDP per capita in Africa (2009)
. * Options: stat(GDP), year(2009), cname displays country names
. worldstat Africa, stat(GDP) year(2009) cname
worldstat is built using the functionality of the module wbopendata.
checking wbopendata consistency and verifying not already installed...
Accessing shape file for Africa to create geographical visualisation
Accessing shape files for map output remotely
(prefix now "http://damianclarke.net/stata/worldstat")
```

```
Importing GDP from World Bank database

    Metadata for indicator NY.GDP.PCAP.KD
-------------------------------------------------------------------------------
    Name: GDP per capita (constant 2015 US$)
-------------------------------------------------------------------------------
    Collection: 2 World Development Indicators
-------------------------------------------------------------------------------
    Description: Gross domestic product is the total income earned through
    the production of goods and services in an economic territory during an
    accounting period. It can be measured in three different ways: using
    either the expenditure approach, the income approach, or the production
    approach. The core indicator has been divided by the general population
    to achieve a per capita estimate.This indicator is expressed in constant
    prices, meaning the series has been adjusted to account for price
    changes over time. The reference year for this adjustment is 2015. This
    indicator is expressed in United States dollars.
-------------------------------------------------------------------------------
    Note: Country official statistics, National Statistical Organizations
    and or Central Banks;
-------------------------------------------------------------------------------
    Topic(s):  ; 3 Economy and Growth
-------------------------------------------------------------------------------

Visualising data
```

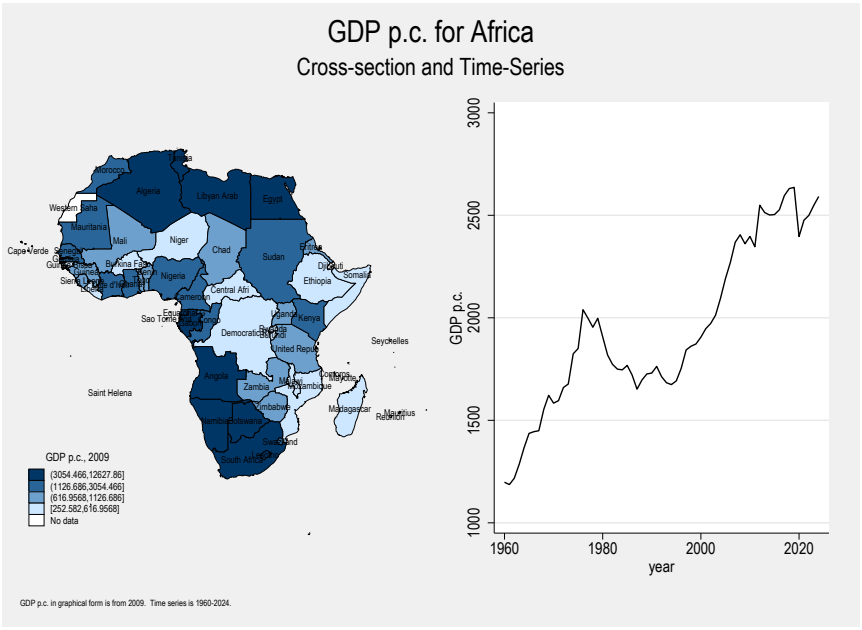

Figure A3: GDP per capita (constant 2015 US$) in Africa, 2009. Generated using `worldstat` (**?**), which calls `wbopendata` internally.

`worldstat`: **global example (verbatim output)**

```
. * Global map: Fertility rate with Pastel2 color scheme
. worldstat world, stat(FERT) fcolor(Pastel2)
worldstat is built using the functionality of the module wbopendata.
checking wbopendata consistency and verifying not already installed...
Accessing shape file for world to create geographical visualisation
Accessing shape files for map output remotely
(prefix now "http://damianclarke.net/stata/worldstat")
Importing FERT from World Bank database

    Metadata for indicator SP.DYN.TFRT.IN
-------------------------------------------------------------------------------
    Name: Fertility rate, total (births per woman)
-------------------------------------------------------------------------------
    Collection: 2 World Development Indicators
-------------------------------------------------------------------------------
    Description: Total fertility rate represents the number of children that
    would be born to a woman if she were to live to the end of her
    childbearing years and bear children in accordance with age-specific
    fertility rates of the specified year.
-------------------------------------------------------------------------------
    Note: World Population Prospects, United Nations (UN), publisher: UN
    Population Division;
-------------------------------------------------------------------------------
    Topic(s):  ; 8 Health
-------------------------------------------------------------------------------

Visualising data
```

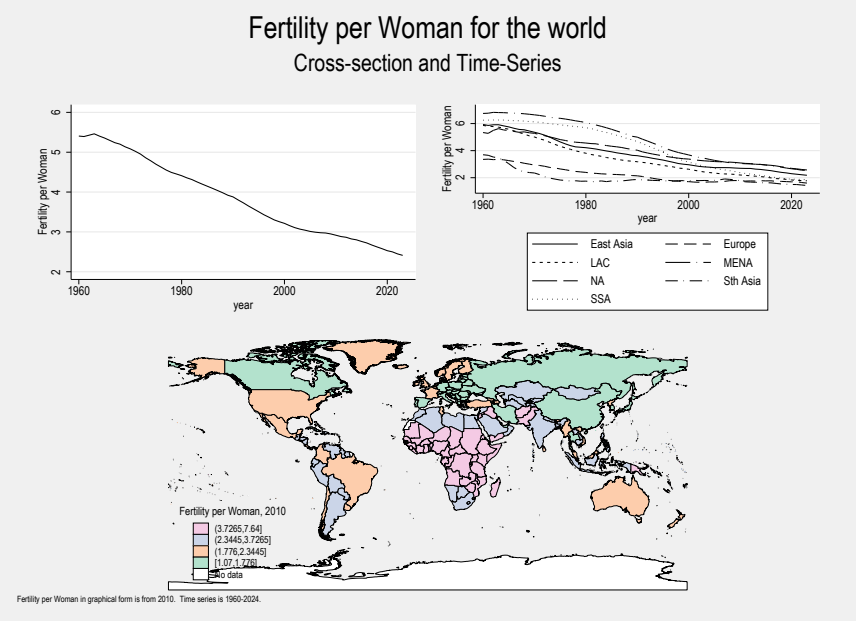

Figure A4: Fertility rate (births per woman) worldwide. The `worldstat` command uses `wbopendata` internally to retrieve World Bank indicators.

A.5 Notes on reproducibility

All examples in this appendix are fully reproducible using the version of `wbopendata` documented in the main text. Because outputs depend on live API responses, results may differ over time as underlying data are revised. Such differences reflect upstream updates rather than changes in analytical logic and can be identified by re-running the corresponding commands.

B Diagnostics and verbatim outputs

B.1 Missing indicator example

```
. wbopendata, language(en) indicator(platypus) long clear

   Sorry... No data was downloaded for indicator platypus.

   (1) Please check your internet connection by clicking here, if does not
   work please check with your internet provider or IT support, otherwise...
   (2) Please check your access to the World Bank API by clicking here, if
   does not work please check with your firewall settings or internet
   provider or IT support, otherwise...
   (3) Please check the availability of your indicator or topic by clicking
   here. If the paramater value is not valid...
```

```
    (4) Please check the list of available indictator(s) or topic(s) in the
    help wbopendata or by visiting the API query builder, if all the above
    seems fine...
    (5) Please consider ajusting your Stata timeout parameters. For more
    details see netio.
    (6) Please send us an email to report this error by clicking here or
    writing to:
            email: data@worldbank.org
            subject: wbopendata query error at 5 Jan 2026 14:07:20:
    https://api.worldbank.org/v2/en/countries/all/Indicators/platypus?download
    > format=CSV&HREQ=N&filetype=data

. di as text "Captured return code (expected nonzero): "
Captured return code (expected nonzero):
```

B.2 Deprecated / archived indicator example

```
. wbopendata, language(en) indicator(AG.AGR.TRAC.NO) clear

    Sorry... but indicator AG.AGR.TRAC.NO has been moved to 57 WDI Database
    Archives.

    Please send us an email to obtain more information clicking here or
    writing to:
            email: data@worldbank.org
            subject: wbopendata query error 23 [AG.AGR.TRAC.NO - Agricultural
    machinery, tractors] at 5 Jan 2026 14:07:20:
    https://api.worldbank.org/v2/Indicators/AG.AGR.TRAC.NO

. di as text "Captured return code (expected r(23) archive notice): "
Captured return code (expected r(23) archive notice):
```

B.3 Metadata cache update output

```
. wbopendata, update query

-------------------------------------------------------------------------------

Indicators update status

   Existing Number of Indicators:  29323
   Last check for updates:         4 Jan 2026 13:05:24
   New update available:           none      (as of 4 Jan 2026 13:05:24)
   Current update level:           4 Jan 2026 13:05:24

   Country metadata:               296
   Last country check:             4 Jan 2026 13:05:24
   Current country update level:   4 Jan 2026 13:06:46

Possible actions

   Check for available updates   (or type -wbopendata, update check detail -)

   See current documentation on indicators list, Regions,
   Administrative Regions, Income Levels, and Lending Types
```

C Quality assurance, testing, and operational validation

This appendix documents the quality assurance (QA) and testing framework used to maintain the reliability of `wbopendata` over time. These practices are a core component of the command's reproducibility guarantees, ensuring that changes in upstream data infrastructure, metadata, and indicator availability are detected, diagnosed, and addressed explicitly.

Unlike statistical reproducibility, which concerns the stability of estimates conditional on data, QA for data-access infrastructure must address a distinct set of risks: API schema changes, endpoint deprecations, evolving indicator vocabularies, pagination failures, and environment-specific constraints. The framework described here is designed to surface such issues under real-world operating conditions.

C.1 Testing philosophy: operational validation

The testing strategy for `wbopendata` is based on *live integration tests* that exercise the command end-to-end against the World Bank Data API. Tests intentionally rely on live network access and real data downloads, rather than mocked responses or static fixtures.

This approach differs from the release-gate testing paradigms used by CRAN or PyPI, where deterministic, offline tests are required to certify software correctness in sandboxed environments. In contrast, `wbopendata` operates in trusted, interactive Stata environments where network access is both available and essential. Because the core functionality of the command is to retrieve and process live data, operational validation against the actual API is the most informative way to assess reliability.

Under this paradigm, test failures do not necessarily imply software defects. They may reflect upstream API outages, schema changes, or metadata revisions. The objective of the test suite is therefore not to freeze outputs, but to make such changes explicit and diagnosable.

C.2 Test harness structure

Automated tests are implemented in a single driver script, `run_tests.do`, which supports full test runs, single-test execution, and verbose (trace) modes. Tests are organized into functional categories that correspond to core features of the command, including environment checks, download modes, output formatting, country metadata, graph metadata, update commands, language options, and advanced features.

Each test follows a standardized pattern:

- explicit execution of the command under test,
- immediate inspection of return codes,
- verification of expected variables, structures, or scalars,

- informative pass/fail messages tied to specific failure modes.

Helper routines manage test execution, result aggregation, and logging, ensuring that failures are attributed to specific test identifiers. A complete test run produces a timestamped log and updates a cumulative test history file.

C.3 Regression testing and cumulative reliability

A key feature of the QA framework is the systematic treatment of historical bugs as permanent regression tests. Issues such as failures in multi-indicator queries, metadata parsing errors, incorrect handling of deprecated indicators, or option incompatibilities are encoded as named test cases (e.g., `REG-33`, `REG-45`, `REG-46`, `REG-51`).

This approach ensures that fixes are not transient. Once a bug is resolved, the corresponding test remains in the suite and must pass in all subsequent runs. As a result, reliability is cumulative rather than episodic.

The longitudinal test history recorded in `test_history.txt` documents this process explicitly: early versions show multiple failures as new features were introduced and edge cases identified, followed by successive reductions in failures as fixes were applied, culminating in fully passing test runs for recent releases. This history provides an auditable record of stabilization over time rather than a single-point snapshot.

C.4 Diagnostics and failure modes

Tests are designed to capture and report informative diagnostics rather than binary outcomes. When a test fails, logs record the API endpoint accessed, the parameters supplied, and the return codes or error messages produced. This information is essential for distinguishing between upstream changes and genuine software regressions.

Illustrative examples of user-facing diagnostics—including missing indicators, deprecated series, and metadata update behavior—are reported in Appendix **??**. Appendix **??** documents *what users see*, while the present appendix documents *how those behaviors are systematically validated*.

C.5 Indicator lifecycle and metadata evolution

The QA framework explicitly accounts for the fact that the set of available indicator codes and their associated metadata evolve over time. Indicators may be deprecated, archived, or reclassified as source databases are revised, and metadata fields may be corrected or expanded.

Tests covering update commands (`update query`, `update check`, `update all`) verify that such changes are detected and propagated through the local cache. Deprecated indicators are expected to trigger informative diagnostics rather than silent failures. By treating indicator lifecycle events as testable behavior, the framework reduces the risk

that analytical pipelines continue to rely unknowingly on obsolete series or outdated definitions.

C.6 Reproducibility of the QA process

All QA artifacts—including test scripts, protocols, logs, and cumulative test histories—are version-controlled alongside the source code. This allows the QA process itself to be reproduced, inspected, and audited.

Because tests depend on live API responses, exact outputs may change as upstream data are revised. Such changes are treated as expected signals when they reflect documented updates. The purpose of the QA framework is therefore not to enforce static results, but to ensure that change is explicit, traceable, and consistent with upstream revisions.

C.7 Continuous validation under ecosystem constraints

Although `wbopendata` is developed within the Stata ecosystem, which lacks the automated release-gate infrastructure common to CRAN or PyPI, its QA framework implements the functional equivalent of continuous integration and continuous validation. Automated test execution, regression protection, and versioned test histories ensure that each code change is evaluated against both current functionality and historically fixed issues.

In this setting, continuous validation is achieved not through centralized CI/CD services, but through reproducible test scripts that can be executed consistently by developers prior to release. Each test run produces a timestamped log and updates a cumulative history, creating an auditable trail of stability across versions. This approach aligns with institutional constraints—such as the need for live API access and trusted execution environments—while preserving the core objective of CI/CD practices: early detection of regressions and disciplined release management.

By adapting CI/CD principles to the realities of statistical software distribution, `wbopendata` demonstrates that rigorous quality assurance is achievable even in environments without formal automation pipelines.

C.8 Illustrative QA artifacts

To support transparency, selected excerpts from the automated test harness and logs are included below. These excerpts illustrate the structure of the testing framework and the form of diagnostics produced during execution.

Test harness initialization (excerpt)

```
--------------------------------------------------------------------------------
```

```
WBOPENDATA TEST SUITE
Version: 17.7.1
Build:   04Jan2026
Date:    4 Jan 2026 21:12:38
Stata:   17
--------------------------------------------------------------------------------

. run_test "ENV-01" "wbopendata version matches repo"

--- TEST ENV-01: wbopendata version matches repo ---
Installed version: *! v 17.7.1  04Jan2026
Repo version:      *! v 17.7.1  04Jan2026
PASS
r; t=0.00 21:12:38

--------------------------------------------------------------------------------

. run_test "DL-01" "Single indicator download"

--- TEST DL-01: Single indicator download ---
. wbopendata, indicator(SP.POP.TOTL) clear long nometadata
Observations: 17,290
Variables:    12
PASS
r; t=3.19 21:12:42

--------------------------------------------------------------------------------

. run_test "DL-04" "Multiple indicators download"

--- TEST DL-04: Multiple indicators download ---
. wbopendata, indicator(SP.POP.TOTL;NY.GDP.MKTP.CD) clear long nometadata
PASS
r; t=4.96 21:12:49

--------------------------------------------------------------------------------

. run_test "REG-45" "Issue #45: URL in metadata"

--- TEST REG-45: Issue #45: URL in metadata ---
. wbopendata, indicator(SL.UEM.TOTL.ZS) clear
Metadata parsed successfully (URL preserved)
PASS
r; t=0.56 21:13:14

--------------------------------------------------------------------------------

. run_test "UPD-01" "Update query command"

--- TEST UPD-01: Update query command ---
. wbopendata, update query
Indicators update status: no updates available
PASS
r; t=0.02 21:13:21

--------------------------------------------------------------------------------

. run_test "UPD-05" "Update all"

--- TEST UPD-05: Update all ---
```

```
. wbopendata, update all
Metadata refreshed successfully
PASS
r; t=0.50 21:13:40

--------------------------------------------------------------------------------

TEST SUMMARY

Tests Run:    44
Tests Passed: 44
Tests Failed: 0

ALL TESTS PASSED!

--------------------------------------------------------------------------------
```

Summary of a complete test run (excerpt)

```
------------------------------------------------------------------------
. wbopendata Automated Test Suite - Summary Report
------------------------------------------------------------
  ENV:  Environment checks ................. 2/2 passed
  DL:   Download modes ..................... 8/8 passed
  FMT:  Output format ...................... 5/5 passed
  CTRY: Country attributes ................. 4/4 passed
  LW:   Linewrap metadata .................. 6/6 passed
  REG:  Regression fixes ................... 7/7 passed
  ADV:  Advanced options .................. 12/12 passed
------------------------------------------------------------------------
Total:    44 tests executed
Passed:   44 tests (100%)
Failed:   0 tests
Duration: 6 minutes 39 seconds
------------------------------------------------------------------------
Result: ALL TESTS PASSED
------------------------------------------------------------------------
```

C.9 Implications for reproducible analytics

The QA practices documented here reinforce the paper's central argument that reproducibility depends not only on statistical methods, but also on the reliability of data-access infrastructure. By embedding operational validation, regression testing, and explicit diagnostics into the development lifecycle, `wbopendata` ensures that scripted data acquisition remains a stable and auditable foundation for empirical analysis, even as upstream data systems evolve.

In this sense, quality assurance is not an ancillary engineering concern, but a core component of reproducible analytics. It provides the enforcement mechanism through which principles such as data acquisition as code and explicit data provenance are sustained in practice.