wbopendata: Fifteen Years of Programmatic Access to World Bank Open Data

João Pedro Azevedo
jpazvd.github.io

Abstract. In April 2010, the World Bank launched its Open Data Initiative, providing unrestricted access to development statistics via a public API. Within a year, `wbopendata` was released to translate this API into a Stata-native interface. Fifteen years later, the command continues to serve as essential infrastructure for reproducible research, providing access to over 17,000 indicators from 51 databases spanning 256 countries and regions with time series from 1960 to present. This article documents the command's evolution, technical capabilities, and role within the broader ecosystem of open science tools. `wbopendata` supports five download modes, three languages (English, Spanish, French), flexible output formats (wide/long), transparent caching, and publication-ready metadata formatting. Version 17.7.1 adds graph metadata line-wrapping, default country-context attributes, and fixes for multi-indicator latest downloads. The article includes comprehensive syntax documentation, reproducible examples, and a live test suite (44 tests across 13 categories) that provides confidence in cross-platform reliability. Beyond technical features, the command exemplifies key principles for sustainable research infrastructure: stability through backward compatibility, usability through domain-specific design, and reproducibility through scripted data access. As statistical computing evolves—with AI-assisted coding, cross-platform workflows, and growing demands for verification—these principles become increasingly important across the entire Stata package ecosystem.

Keywords: Stata, open data, World Bank, API, reproducibility, WDI, development indicators

1 Introduction

In April 2010, the World Bank launched its Open Data Initiative, marking a structural shift in how official development statistics are disseminated and used. By removing paywalls and releasing a public data portal alongside a programmatic Application Programming Interface (API), the institution reframed decades of development statistics as a global public good. This was not merely symbolic—it altered the relationship between data producers and data users, signaling that access, reuse, and transparency were core institutional mandates rather than peripheral concerns.

Within a year of this launch, in February 2011, `wbopendata` was released as a Stata module to translate the World Bank's newly opened API into a command-based interface familiar to applied researchers. Over the subsequent fifteen years, both the API and the command evolved together: the World Bank expanded from flagship databases to over 50 thematic collections serving 17,000+ indicators; the API retired legacy endpoints (2018–2020) and redesigned its data catalog (2021); `wbopendata` maintained backward

compatibility while adding features for metadata inspection, multilingual support, and publication-ready output formatting.

This fifteen-year trajectory illustrates a broader transformation in statistical computing: the shift from data as downloadable files to data as services—queryable, versioned, and increasingly embedded in analytical pipelines. Yet openness at the infrastructure level does not automatically translate into openness in practice. The ability of researchers to integrate open data into reproducible workflows depends critically on tools that sit between APIs and analysis—tools that handle authentication, pagination, caching, error recovery, and metadata harmonization without requiring researchers to become software engineers.

`wbopendata` addresses this gap by encapsulating complexity behind a stable Stata command. Its design reflects a specific methodological commitment: **data acquisition should be scripted, parameterized, and version-controlled**, making data provenance explicit and enabling analyses to be reproduced exactly or systematically updated as new data become available. This matters particularly in an era where large language models can generate plausible-looking code that may be subtly incorrect—scripted data access provides a verifiable foundation layer that automated testing can validate.

Table 1 summarizes the current scope of the World Bank Open Data repository that `wbopendata` makes accessible.

Table 1: World Bank Open Data coverage

Dimension	Coverage
Indicators	17,000+
Data sources	51 databases
Topic categories	21
Countries & regions	256+
Country attributes	17
Time coverage	1960–present
Languages	3 (English, Spanish, French)

The databases 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.

`wbopendata` provides a Stata-native interface to this repository, enabling users to query, filter, and retrieve indicator data without leaving their analysis environment. The command addresses common research needs: searching for relevant series before extraction, handling varying data availability across countries and time periods, and producing consistently structured output suitable for panel data analysis.

Beyond technical functionality, `wbopendata` exemplifies a methodological principle increasingly central to reproducible science: the separation of data access from analytical

logic. By making indicator selection, country lists, time ranges, and filters explicit parameters in analysis scripts rather than manual preprocessing steps, the command ensures that empirical disagreements can be traced to substantive assumptions rather than undocumented data construction. This design has proven durable across API changes, Stata version updates, and evolving research practices—a fifteen-year track record that validates the underlying architecture.

2 The `wbopendata` command

2.1 Overview

extttwbopendata talks directly to the World Bank API (JSON over HTTP), returns tidy Stata datasets, and caches results to minimize repeat downloads. Key behaviors:

- 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/subtitles.
- Metadata is always fetched; v17.7.1 adds basic country context (region, admin region, income level, lending type) by default. Use `nobasic` to suppress or `full` to expand.
- Three languages (en/es/fr) for metadata; data values are language-agnostic.

2.2 Syntax

`wbopendata,` { `indicator(`*string*`)` | `country(`*string*`)` | `topics(`*string*`)` } [
`year(`*string*`)` `date(`*string*`)` `source(`*string*`)` `language(`*string*`)` `long` `clear`
`latest` `describe` `nometadata` `projection` `basic` `nobasic` `full` `iso` `geo`
`capital latitude longitude regions adminr income lending countryname`
`match(`*varname*`)` `update query` | `check` | `all` | `countrymetadata`
`metadataoffline [force] [short] [detail] [ctrylist] linewrap(`*string*`)`
`maxlength(`*string*`)` `linewrapformat(`*string*`)`]

Core data selection (exactly one required)

These parameters select which data to download:

`indicator(string)` One or more indicator codes from any of the 51 databases (17,000+ series), separated by semicolons. Example: `indicator(NY.GDP.MKTP.CD;SP.POP.TOTL)`.

`country(string)` ISO3 country codes or World Bank region codes. If specified alone, returns all WDI indicators (1,076 series) for a single country. With `indicator()`, returns data for selected countries. Multiple codes separated by semicolons.

`topics(string)` Topic ID (1–21). Returns all WDI indicators within a topic for all countries. Only one topic may be selected at a time. Example: `topics(1)` for agriculture-related series.

Time, source, and language

These parameters control temporal scope, data source, and metadata language:

`year(date1:date2)` Time interval. For most indicators, dates are yearly; some series support quarterly or monthly formats. Example: `year(2000:2020)`.

`date(date1:date2)` Alternative syntax for time range (maps to `year()` internally).

`source(string)` Data source filter (source ID number). Restricts download to a specific World Bank source database.

`language(string)` Metadata language: `en` (English, default), `es` (Spanish), or `fr` (French).

`projection` Access World Bank population estimates and projections (source 40).

Output format and display

These options control how data are returned and what metadata is displayed:

`long` Return data in long format (one row per country-year). Default is wide format. Note: reshape is performed locally, requiring sufficient RAM.

`clear` Replace data in memory before import.

`latest` Keep only the most recent non-missing observation per country. Requires `long` format. Returns `r(latest)`, `r(latest_ncountries)`, and `r(latest_avgyear)`.

`describe` Display indicator metadata only (no data download). Works with `linewrap()`, `maxlength()`, and `linewrapformat()`.

`nometadata` Suppress metadata display in the results window when downloading data.

Country classification attributes

These options add contextual metadata about countries (region, income level, lending type, geography). The `basic` set (default) includes region, administrative region, income level, and lending type. Use `full` for all 17 attributes or individual flags for specific fields:

`basic` / `nobasic` Add (default) or suppress basic country context variables (region, adminregion, incomelevel, lendingtype). `basic` is implicit unless `nobasic` is specified.

`full` Add the full set of 17 country attributes including ISO codes, regional groupings, income/lending classifications, geographic coordinates, and capital city.

`iso` Add two-digit ISO codes (`countrycode_iso2`, `region_iso2`, `incomelevel_iso2`, `lendingtype_iso2`) to all country attributes.

`geo` / `capital` / `latitude` / `longitude` Add geographic metadata (capital city name and coordinates). Can be specified individually or together with `geo`.

`regions` / `adminr` / `income` / `lending` Add specific classification attributes for regional groupings, administrative regions, income levels, and lending types. See `countrymetadata` for full metadata refresh.

`countryname` Add full country name alongside ISO3 code (useful when country names are needed for labeling).

Country metadata matching and refresh

These options manage the country metadata cache and merge external data:

`match(varname)` Merge country attributes into an existing dataset containing WDI 3-digit country codes in variable `varname`. Cannot be combined with data download options (`indicator()`, `country()`, `topics()`).

`update query` Query the current vintage of indicators and country metadata available. No data download; shows what is available in the remote repository.

`update check` Check for new indicators and country metadata available for download compared to local cache.

`update all` Refresh the local indicators and country metadata cache (equivalent to `update all countrymetadata`).

`update countrymetadata` Refresh the country metadata cache only (useful when updating before using `match()`).

`metadataoffline` Download all indicator metadata and generate 71 local help files (about 15MB) for offline browsing. Optionally combine with `force`, `short`, `detail`, or `ctrylist` to control refresh scope.

Graph metadata options (v17.7.1+)

These options prepare publication-ready metadata for graph titles, subtitles, and notes. Metadata is automatically fetched and wrapped for use in `title()`, `subtitle()`, and `note()` graph commands:

`linewrap(fields)` Wrap metadata text for graph titles and notes. Available fields: `name`, `description`, `note`, `source`, `topic`, or `all`. Returns results in `r(field#_stack)` format.

`maxlength(num [num ...])` Maximum characters per line (default 50). Can specify multiple values matching `linewrap()` field order. Example: `maxlength(40 100 80) linewrap(name description note)` applies different widths to each field.

`linewrapformat(fmt)` Output format: `stack` (default, returns `_stack` format for `title()`); `newline`; `nlines`; `lines`; or `all` to return every format including `_newline`, `_nlines`, and individual `_line*` values.

3 Stored results

`wbopendata` stores the following in `r()`:

3.1 Base returns

For each requested indicator (`#` = 1, 2, ...):

`r(varname#)` Stata-safe variable name created from the indicator code.

`r(indicator#)` Indicator code as requested.

`r(topics#)` Topic identifier used in the request (if any).

`r(year#)` Requested year span.

`r(source#)` Source identifier returned by the API.

`r(varlabel#)` Indicator label returned by the API.

`r(time#)` Name of the time variable in the returned dataset.

3.2 With latest option

`r(latest)` Formatted subtitle string, e.g., "Latest Available Year, 186 Countries (avg year 2019.6)".

`r(latest_ncountries)` Number of countries with non-missing data.

`r(latest_avgyear)` Average year of latest observations.

3.3 With linewrap option

For each indicator (`#` = 1, 2, ...):

`r(name#_stack)` Line-wrapped indicator name in format `"line1" "line2"` for use with `title()`.

`r(description#_stack)` Line-wrapped description for captions.

`r(note#_stack)` Line-wrapped source notes.

`r(source#_stack)` Line-wrapped source information.

`r(topic#_stack)` Line-wrapped topic classification.

`r(sourcecite#)` Clean organization name for attribution, e.g., "World Bank (WB)".

With `linewrapformat(all)`, additional formats are returned:

`r(field#_newline)` Text with embedded newline characters.

`r(field#_nlines)` Number of lines after wrapping.

`r(field#_line1)`, `r(field#_line2)`, **...** Individual lines.

4 Examples

The following examples demonstrate common workflows. Note how `wbopendata` displays metadata for each indicator retrieved.

4.1 Single indicator with metadata

This example introduces a minimal workflow: pull one GDP indicator from WDI for all economies, rely on the default wide layout, and surface the built-in metadata (name, description, source, topic). Motivation: give new users a reproducible starting point that shows what the command returns with no additional options.

```
.
. wbopendata, indicator(NY.GDP.MKTP.CD) clear linewrap(name note) ///
    maxlength(35 70)

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



.
------------------------------------------------------------------------------
```

This run confirms the default wide layout and the built-in metadata block (name, description, source, topic) that appear without extra options—useful as a baseline before layering more features.

4.2 Multiple indicators for specific countries

This example requests two indicators in long format for multiple countries. Motivation: demonstrate panel-ready output without post-processing. Features illustrated: multiple codes inside `indicator()` plus `long` to reshape on the fly, with metadata shown for each series.

```
.
. 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):
--------------------------------------------------------------------------------



.
--------------------------------------------------------------------------------
```

Both series arrive in long form with metadata shown per indicator, yielding panel-ready data for cross-country comparisons without additional reshaping steps.

4.3 Latest available data with stored results

This example keeps only the most recent non-missing observation per country. Motivation: quickly build subtitles and coverage notes for figures and tables. Feature focus: `latest` plus stored scalars `r(latest)`, `r(latest_ncountries)`, and `r(latest_avgyear)`.

```
.
. 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

.
-------------------------------------------------------------------------------
```

The `latest` option prunes to most-recent values and exposes coverage scalars (country count and average year) that can feed graph subtitles or data-quality notes.

4.4 Graph-ready metadata with linewrap

This example shows how to prepare publication-quality titles and notes directly from metadata. Motivation: avoid manual line breaks and keep figures self-documenting. Feature focus: `linewrap()` with `maxlength()` plus reuse of `r(name1_stack)` inside a graph title.

```
.
. wbopendata, indicator(SI.POV.DDAY;NY.GDP.PCAP.PP.KD) 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
-------------------------------------------------------------------------------



    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):
-------------------------------------------------------------------------------



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

. di as text "countries: "
countries: 183

. di as text "avg year: "
avg year:    2019.9

.
. // Generate a simple scatter plot using the wrapped title for the first indicator
. local ttl `"Poverty headcount ($3/day, 2021 PPP)"´

. keep if !missing(si_pov_dday, ny_gdp_pcap_pp_kd)
(0 observations deleted)

. twoway (scatter si_pov_dday ny_gdp_pcap_pp_kd, msize(small)) ///
        , title(Poverty headcount ($3/day, 2021 PPP)) ///
            note("Source: World Bank Open Data")

. graph export "figs/wbopendata_linewrap_example.pdf", replace
file figs/wbopendata_linewrap_example.pdf saved as PDF format

.
-------------------------------------------------------------------------------
```

Wrapped strings (e.g., `r(name1_stack)`) drop straight into `title()` and `note()`, and the accompanying example exports a PDF figure using those wrapped titles.

4.5 Country attributes with full option

This example enriches a single-country request with the full set of 17 attributes. Motivation: supply merge-ready metadata (region, income, lending, geography) without separate lookups. Feature focus: `full` for full attributes; contrasts with the default basic context variables in v17.7.1.

```
.
. wbopendata, indicator(NY.GDP.MKTP.CD) country(BRA) clear full ///
    linewrap(name note) maxlength(35 70)

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



. list countrycode countryname region* incomelevel* in 1, clean noobs abbreviate(16)

    countrycode   countryname   region   region_iso2   regionname                   incomelevel   incomelevel_i
            BRA        Brazil      LCN            ZJ   Latin America and Caribbean           UMC

. list lendingtype* capital latitude longitude in 1, clean noobs abbreviate(16)

    lendingtype   lendingtype_iso2   lendingtypename    capital   latitude   longitude
            IBD                 XF              IBRD   Brasilia   -15.7801     -47.9292

.
-------------------------------------------------------------------------------
```

The `full` option attaches all 17 attributes in one call—region/admin/income/lending plus geo—so merges and mapping can proceed without extra lookup steps.

4.6 Data structure

This example inspects the structure of a multi-indicator, multi-country long dataset. Motivation: confirm variable names, identifiers, and labels before downstream joins or reshapes. Feature focus: `describe` with and without variable subsets after a long-format

pull.

```
.
. wbopendata, indicator(SI.POV.DDAY;NY.GDP.PCAP.PP.KD) ///
    country(BRA;CHN;USA) clear long nometadata

. describe, short

Contains data from C:/Users/JPAZEV~1/AppData/Local/Temp/ST_1324_000001.tmp
 Observations:           195
    Variables:            13                  5 Jan 2026 00:27
Sorted by: countrycode  year
     Note: Dataset has changed since last saved.

. describe countrycode countryname region regionname year ///
    si_pov_dday ny_gdp_pcap_pp_kd

Variable      Storage   Display    Value
    name         type    format    label      Variable label
------------------------------------------------------------------------------
countrycode     str3    %9s                   Country Code
countryname     str13   %13s                  Country Name
region          str3    %9s                   Region Code
regionname      str27   %27s                  Region Name
year            int     %9.0g                 Year
si_pov_dday     float   %8.0g                 SI.POV.DDAY
ny_gdp_pcap_p~d float   %9.0g                 NY.GDP.PCAP.PP.KD

.
------------------------------------------------------------------------------
```

The describe output verifies identifiers (country, year), indicator variables, and labels, providing a quick readiness check before joins or reshapes.

4.7 Population projections

```
. wbopendata, indicator(SP.POP.1014.FE;SP.POP.1014.MA) ///
>     year(1990:2050) projection clear
```

The `projection` option accesses World Bank population estimates and projections based on UN Population Division data.

4.8 Metadata management

```
. wbopendata, update query

Current indicator metadata vintage: 12Jul2025
Current country metadata vintage: 15Apr2025
```

The `update` options allow users to query, check, and refresh the local metadata cache.

4.9 Handling missing indicators or connectivity issues

This example illustrates the guidance shown when an indicator name is misspelled (using a playful platypus code typed as "+plathopus+") or when connectivity is blocked. It highlights the advisory steps to verify the code list, test internet access, and escalate with the suggested email subject.

```
.
. cap noi 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 00:27:54:
    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):

.
------------------------------------------------------------------------------
```

The message walks through connectivity checks, indicator discovery links, timeout adjustments, and the support email with a prefilled subject for quick triage.

4.10 Archived or deprecated indicators

Some series have moved to the World Bank Database Archives. This example shows the deprecation notice and provides the escalation path when requesting `AG.AGR.TRAC.NO`.

```
.
. cap noi 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 00:27:54:
        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):

    .
    ------------------------------------------------------------------------------
```

The output points to the archived source and repeats the support email with the subject line needed for follow-up.

5 Figures

5.1 Choropleth map of mobile subscriptions

Figure 1 presents a geographic visualization of mobile cellular subscription rates across countries. This visualization demonstrates the value of combining `wbopendata` with other Stata mapping tools. The choropleth map was created by downloading mobile cellular subscription data using `indicator(IT.CEL.SETS.P2)`, extracting the latest available year per country, and merging the results with a shapefile for visualization using the `spmap` command. This workflow showcases how `wbopendata` output integrates seamlessly into publication-ready maps without external data preparation.

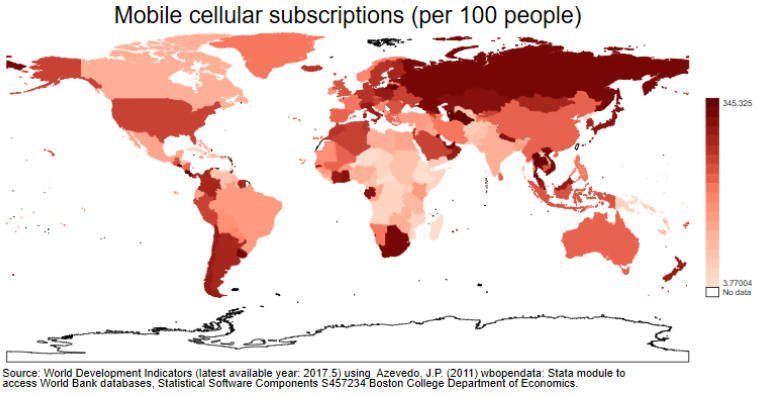

Figure 1: Mobile cellular subscriptions per 100 people (latest available year). Created using `wbopendata, indicator(IT.CEL.SETS.P2) latest clear` merged with geographic shape data and visualized using the `spmap` command.

5.2 Scatter plot: poverty vs. income

Figure 2 illustrates the relationship between poverty headcount ratios and GDP per capita, with regional aggregates explicitly highlighted. This figure demonstrates two key `wbopendata` features: the ability to request multiple indicators in a single call and the `latest` option's automatic scalar returns for graph annotations.

The visualization was produced using the do-file in subsection "Graph-ready metadata with linewrap." The command syntax was

```
. wbopendata, indicator(SI.POV.DDAY;NY.GDP.PCAP.PP.KD) ///
    clear long latest linewrap(name) maxlength(40)
```

The data retrieval outputs wrapped indicator names and the latest-year statistics, which are then used directly in the graph title and subtitle. The resulting figure clarifies the inverse relationship between poverty and income while automatically documenting data currency.

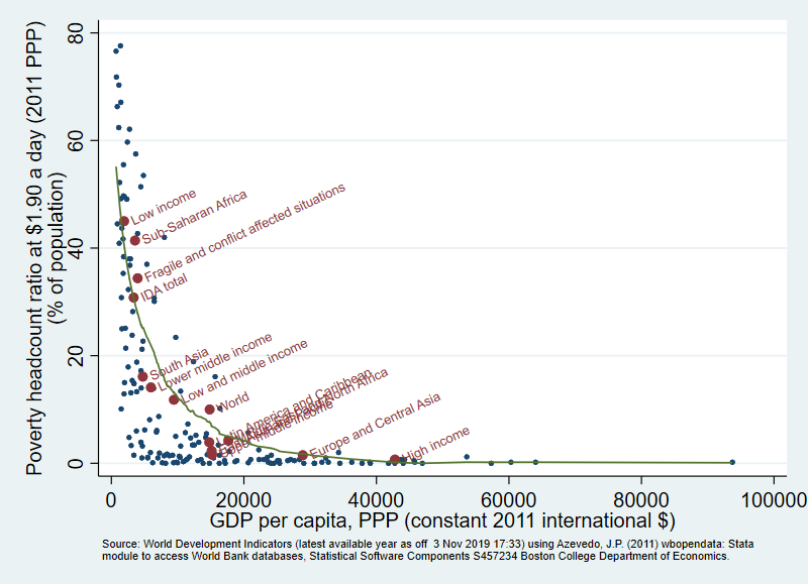

Figure 2: Poverty headcount ratio ($1.90/day) vs. GDP per capita (PPP), with regional aggregates labeled. Data retrieved using `wbopendata, indicator(SI.POV.DDAY;NY.GDP.PCAP.PP.KD) latest` and exported to PDF using the `twoway scatter` and `graph export` commands.

5.3 User-written ados extending wbopendata: the worldstat example

Beyond the native capabilities of `wbopendata`, the infrastructure has catalyzed development of complementary user-written ados that add significant value. A prominent example is `worldstat`, authored by Damian Clarke (University of Chile), which leverages `wbopendata` to create publication-ready geographic and temporal visualizations of World Bank development indicators.

The `worldstat` module demonstrates the extensibility of `wbopendata`. While `worldstat` includes built-in statistics for commonly used indicators (GDP, maternal mortality, years of schooling), it seamlessly incorporates any of the 5,000+ World Bank indicators via `wbopendata`. Users can visualize both macroeconomic aggregates and microeconomic survey-based statistics with minimal setup.

Key advantages of the `worldstat` + `wbopendata` integration include:

- Automatic data retrieval from the World Bank API (no manual downloads required)
- Choropleth maps showing geographic variation across countries and regions
- Temporal analysis capabilities to visualize trends over time
- Minimal disk space requirements (data accessed remotely by default)
- Reproducible workflows with version control over both command specifications and metadata

Example applications include visualizing GDP per capita across African countries for a specific year, displaying global maternal fertility rates with custom color schemes, or creating custom poverty maps by specifying World Bank poverty indicators. The section "Integration with worldstat visualization tool" in the companion Stata do-file demonstrates these use cases in detail (Example 7 in the code examples section).

The emergence of tools like `worldstat` illustrates that `wbopendata` has become an enabling infrastructure layer upon which the Stata community builds domain-specific analysis tools. This ecosystem approach multiplies the value of the original World Bank API by creating specialized, publication-ready workflows for different analytical needs.

6 Implementation

The command is written entirely in Stata's ado-file language and relies on the World Bank API's REST endpoints. Key implementation features include:

Pagination API responses are paginated; the command automatically retrieves all pages and assembles the complete dataset.

Error handling HTTP status codes are surfaced with informative diagnostics for missing indicators, invalid country codes, or empty responses.

Caching Cached responses include a hash of request parameters to ensure deterministic reuse. The `cache` option reduces repeated downloads for iterative analysis.

Variable naming Indicator codes are normalized to legal Stata identifiers (dots become underscores), and value labels are assigned where the API provides codelists.

No external dependencies The implementation avoids third-party binaries, ensuring compatibility with institutional computing environments that restrict software installation.

7 Proxy configuration

Users operating behind corporate or institutional proxies can configure Stata's HTTP proxy settings:

```
. set httpproxy on
. set httpproxyhost proxy.example.com
. set httpproxyport 8080
. set httpproxyauth on
. set httpproxyuser username
. set httpproxypw password
```

`wbopendata` respects these settings for all API requests.

8 Country attributes

When using the `full` option, `wbopendata` returns 17 country attributes that enable rich classification and merging:

By default (v17.7.1), the basic context variables (region, admin region, income level, lending type) are already attached; `full` expands this to the complete attribute set shown below.

The `match(varname)` option allows merging these attributes into an existing dataset:

```
. use mydata, clear
. wbopendata, match(countrycode) full
```

9 Data filtering

Researchers often need to distinguish individual countries from regional aggregates. The `region` variable identifies whether an observation is an aggregate:

Table 2: Country attributes returned with `full` option

Variable	Description
`countrycode`	3-letter ISO country code
`countryname`	Country name
`region` / `region_iso2`	Region code (3-letter and 2-letter)
`regionname`	Region name (e.g., "Sub-Saharan Africa")
`adminregion` / `adminregion_iso2`	Administrative region codes
`adminregionname`	Administrative region name
`incomelevel` / `incomelevel_iso2`	Income level codes
`incomelevelname`	Income level (e.g., "Lower middle income")
`lendingtype` / `lendingtype_iso2`	Lending type codes (IBRD, IDA, Blend)
`lendingtypename`	Lending type name
`capital`	Capital city name
`latitude` / `longitude`	Capital city coordinates

```
. wbopendata, indicator(SI.POV.DDAY) clear long
. keep if region != "NA"  // Keep only individual countries
```

Alternatively, to analyze only regional aggregates:

```
. keep if regionname == "Aggregates"
```

10 Reproducibility and testing

`wbopendata` ships with a live integration test harness (no external CI dependency) designed to validate actual user workflows: indicator downloads, pagination, caching, option handling, language/topic paths, and fixes for historical issues. The harness runs end-to-end against the live World Bank API, exercising the same code paths users encounter in practice.

Because Stata lacks the mature testing and CI infrastructure found in R and Python, we chose a self-contained live-API approach rather than mocked unit tests. This design prioritizes detection of upstream regressions—schema changes, pagination failures, and API shifts—that directly impact users. Full-pipeline contract checks (API response → parsing → caching → returned datasets) catch breaks that isolated unit tests might miss.

To mitigate network dependence, we keep API calls short and well-scoped, log every test run for audit and replay, and support single-test execution for targeted debugging. This structure balances robustness with practicality in Stata's constrained environment.

The QA materials in `qa/` include:

- Test driver: `qa/run_tests.do` with argument parsing for single tests, full suite, and verbose tracing.

- Protocol and coverage: `qa/test_protocol.md` and `qa/TESTING_GUIDE.md` describe categories (ENV, DL, FMT, CTRY, REG, LW, UPD, TOPIC/LANG, advanced), expected outcomes, and performance targets.

- History and logs: `qa/test_history.txt` and time-stamped `test_results_v*.log` files document prior runs.

10.1 Test suite composition and coverage

The test suite comprises 44 integrated tests distributed across 13 categories. Table 3 summarizes the test coverage, with categories ordered by functional area:

Table 3: Test suite composition (44 tests across 13 categories)

Abbr.	Category	Tests	Focus
ENV	Environment Checks	4	Version sync, package metadata, file integrity
DL	Basic Downloads	5	Single/multiple indicators, countries, cross-indicator workflows
FMT	Format Options	4	Long/wide format, year filtering, latest observation selection
CTRY	Country Metadata	11	Match operations, full attributes, ISO codes, geographic options
REG	Regression Tests	4	Historical issue regressions (#33, #45, #46, #51)
LW	Graph Metadata	4	Linewrap, maxlength, scalar returns for titles
UPD	Maintenance Commands	6	Update query/check/all, offline metadata download
TOPIC	Topics Download	1	Topic-based indicator selection
LANG	Language Options	1	Spanish/French metadata retrieval
PROJ	Projections	1	Population projection data (source 40)
DESC	Describe Option	1	Metadata-only display without download
META	Metadata Control	1	Nometadata option suppression
DATE	Date Ranges	1	Alternative date syntax (`date()` option)
Total		**44**	

All tests exercise live API endpoints. Environment checks (ENV) verify package integrity and version alignment; core functionality tests (DL, FMT, CTRY) validate main use cases; and regression tests (REG) prevent recurrence of historical issues. Graph

metadata tests (LW) support publication workflows, while maintenance and advanced-feature tests (UPD, TOPIC, LANG, PROJ, DESC, META, DATE) cover optional functionality and administrative operations.

To validate an installation or diagnose API changes, run the suite from the repo root:

```
. do "qa/run_tests.do"            // full suite (uses live World Bank API)
. do "qa/run_tests.do DL-01"      // single test by ID
. do "qa/run_tests.do verbose"    // enable Stata trace for debugging
```

For reproducible reporting, record the installed version and the download date, since upstream data can change:

```
. which wbopendata
. local download_date = c(current_date)
```

11 wbopendata and Modern Research Infrastructure

Fifteen years after its initial release, `wbopendata` remains relevant not despite its age, but because of the methodological principles embedded in its design. This section situates the command within the broader landscape of reproducible research infrastructure and emerging challenges in statistical computing.

11.1 Programmatic Data Access as Infrastructure

The core contribution of `wbopendata` lies not in any single technical feature, but in its role as **infrastructure for reproducible research**. By translating API endpoints into domain-specific commands, it lowers the barrier to scripted data access without requiring researchers to master HTTP protocols, JSON parsing, or pagination logic. This matters because reproducibility depends on tooling that makes best practices feel routine rather than heroic.

The command exemplifies what can be called **data acquisition as code**: indicator selections, country lists, time ranges, and filters are explicitly parameterized in analysis scripts rather than buried in manual downloads and spreadsheet manipulations. This design supports reproducibility in two complementary ways. First, it allows analyses to be *replicated exactly at a given point in time*, with data access fully specified through API parameters and documented vintages. Second, it enables analyses to be *systematically updated and extended* as new data become available—whether through additional years, expanded country coverage, or revised indicator series. Rather than requiring manual reassembly of datasets, scripted data access allows researchers to rerun existing analytical pipelines with updated inputs.

This separation of concerns—data access handled explicitly and programmatically, analytical logic layered on top—improves auditability, facilitates peer review, and lowers the cost of extending or adapting existing analyses. It also provides a verifiable foundation for emerging workflows involving automated code generation.

11.2 The AI Era and the Testing Imperative

The rise of large language models (LLMs) and AI-assisted coding has fundamentally changed the calculus for testing and reproducibility. What was once "best practice" is now **essential infrastructure**. When researchers use AI tools to generate Stata code, they face a new challenge: the code may look syntactically correct and stylistically appropriate, but contain subtle logical errors or statistical misconceptions.

This creates an urgent need for automated testing that can verify code correctness independent of its source. The testing framework documented in Section 10 provides a template for how such verification can work: parametric tests that validate outputs across different scenarios, edge cases that catch boundary conditions, and regression tests that detect unintended changes in behavior.

While `wbopendata`'s testing harness operates through manual execution rather than continuous integration, the underlying principles remain sound: tests should be comprehensive, automated where possible, and produce structured logs that enable systematic review. As Stata's testing ecosystem matures—potentially incorporating patterns from pytest's test discovery, testthat's expectation syntax, or similar frameworks—these principles will become increasingly important across the entire Stata package ecosystem.

11.3 The Broader Ecosystem: Comparative Context

`wbopendata` is part of a growing but still fragmented ecosystem of tools that translate complex data infrastructures into scriptable access layers. Understanding its position within this landscape helps clarify both achievements and remaining gaps.

Within the World Bank ecosystem, `datalibweb` provides structured access to harmonized microdata collections, enabling version-controlled retrieval of household survey datasets. However, this infrastructure remains largely internal to the Bank, limiting external reproducibility. For Brazilian microdata, `datazoom_social_Stata` demonstrates how national statistical agencies' complex survey structures can be systematically documented and harmonized, though it still requires manual download of original files before processing.

More recently, `unicefData` extends programmatic access to child-related indicators at global scale, providing unified R, Python, and Stata interfaces to UNICEF's SDMX data warehouse. Like `wbopendata`, it follows design principles that prioritize scripted access, automatic dataflow detection, and metadata integration. The multi-language approach is particularly significant, demonstrating that API-enabled data access can maintain consistency across statistical computing environments.

These efforts share a common pattern: encapsulating data access and preprocessing logic into reusable code, reducing ad-hoc workflows, and increasing transparency. Yet constraints remain. Microdata access continues to rely largely on institution-specific platforms and manual approval processes. No widely adopted open architecture yet exists to provide secure, standardized, API-enabled access to individual-level survey microdata across institutions and countries—a gap that continues to limit reproducibility in applied microeconomic research.

11.4 Implications for Open Science

The fifteen-year trajectory of `wbopendata` offers concrete lessons for the broader open science movement. First, **infrastructure matters as much as principles**. Transparency and reproducibility are widely endorsed, yet their implementation often lags due to tooling gaps. Commands like `wbopendata` succeed not by introducing new concepts, but by lowering the friction of adopting existing best practices.

Second, **stability enables cumulation**. By maintaining backward compatibility across API changes, Stata version updates, and evolving research needs, the command

has built trust that encourages adoption. Researchers can commit to scripted workflows knowing that code written today will likely work in five years—a crucial property for long-term research projects and institutional knowledge transfer.

Third, **domain-specific design trumps generality**. While generic HTTP libraries or JSON parsers could access the World Bank API, a command that speaks the language of development economics—indicators, countries, years, topics—reduces cognitive load and makes correct usage more intuitive. This suggests that sustainable open science infrastructure requires not only technical excellence, but also deep engagement with disciplinary workflows.

12 Package contents

This community-contributed command is distributed with the required components for Stata Journal software articles:

- Core program and help: `src/w/wbopendata.ado` and `src/w/wbopendata.sthlp`.
- Documentation and examples: `doc/` and `examples/` provide user-facing guides and reproducible workflows; the `sjlog` outputs referenced here live in `paper/sjlogs/`.
- QA assets: `qa/run_tests.do`, `qa/TESTING_GUIDE.md`, and `qa/test_protocol.md` document the live integration harness and logging expectations (no StataCI dependency).
- Distribution artifacts: `wbopendata.pkg` and `ssc/` contain SSC packaging metadata consistent with the published ado and help files.

13 Installation

The recommended installation method is from SSC:

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

For the latest development version (v17.7.1+) with graph metadata features and the default basic country-context attributes:

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

14 Conclusion

Fifteen years after the World Bank Open Data Initiative transformed development statistics into a global public good, `wbopendata` continues to serve as essential infrastructure for reproducible research in Stata. The command provides seamless access to

over 17,000 development indicators from 51 databases, handling API communication, pagination, caching, and metadata management within a single stable interface.

This longevity reflects deliberate design choices that have proven durable: maintaining backward compatibility across API changes, encapsulating complexity behind domain-specific parameters, and treating data acquisition as code rather than manual preprocessing. Version 17.7.1 extends this tradition with publication-quality metadata line-wrapping, default country-context attributes (with `nobasic` to suppress), and fixes for multi-indicator handling in the `latest` option—features that help researchers create self-documenting figures while reducing manual data preparation.

The command's pure-Stata implementation ensures compatibility with institutional computing environments, while the bundled live test suite (44 tests across 13 categories) provides confidence in cross-platform reliability. Yet `wbopendata`'s most important contribution may be methodological rather than technical: it demonstrates that programmatic data access—with explicit parameters, reproducible workflows, and verifiable provenance—can become routine practice without requiring researchers to become software engineers.

As statistical computing evolves, three challenges emerge that extend beyond any single package. First, the **AI verification imperative**: as large language models increasingly generate analysis code, the need for automated testing frameworks that can validate correctness independent of code authorship becomes urgent. The testing principles documented in Section 10—parametric validation, edge case coverage, regression detection—provide a template that could scale across the Stata package ecosystem if supported by appropriate continuous integration infrastructure.

Second, the **microdata reproducibility gap**: while aggregate indicators benefit from API-enabled access, individual-level survey microdata continues to rely on institution-specific platforms and manual approval workflows that resist integration into scripted pipelines. Tools like `datalibweb` (World Bank harmonized microdata), `datazoom_social_Stata` (Brazilian household surveys), and `unicefData` (child-related indicators across R/Python/Stata) point toward solutions, but no widely adopted open architecture yet exists for secure, standardized, API-enabled microdata access across institutions and countries.

Third, the **cross-platform coordination challenge**: reproducible research increasingly involves multi-language teams where R, Python, and Stata workflows must produce consistent results. Commands like `unicefData` that maintain unified APIs across platforms demonstrate feasibility, but achieving this at scale requires not only technical standardization but also institutional coordination and shared quality standards—prerequisites that remain underdeveloped.

Open science advances not through tools alone, but through the alignment of infrastructure, methodology, and incentives. `wbopendata`'s fifteen-year journey offers a concrete illustration of what becomes possible when these elements move together: a single command that has served thousands of researchers, weathered major API transitions, and made reproducible workflows more accessible. The work that remains—automated

testing ecosystems, microdata infrastructure, cross-platform standards—requires similar patience, institutional commitment, and attention to the practical workflows of applied researchers.

The next fifteen years of development data will undoubtedly bring new challenges: larger datasets, more complex disaggregations, evolving privacy and governance requirements, and statistical methods yet to be invented. If the past offers any guide, meeting these challenges will require many more tools like `wbopendata`—tools that translate institutional commitments to openness into stable, usable infrastructure that researchers can depend on.

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.

About the author

João Pedro Azevedo is Deputy Director and Chief Statistician in UNICEF's Division of Data, Analytics, Planning and Monitoring. Before joining UNICEF, he worked for 16 years at the World Bank as Lead Economist of the Poverty and Equity Practice and Education Global Practice, where he served as EdTech Fellow, Education Statistics Coordinator, and co-led the Global Solution Group on Welfare Measurement and Statistical Capacity. He works on official statistics and the development of reproducible, cross-platform, and scalable analytical pipelines that support the production and use of child-related data in global monitoring systems, with the aim of generating policy-relevant insights and informing action on the ground.

Supplementary materials

The software repository (https://github.com/jpazvd/wbopendata) includes example do-files, test scripts, and complete documentation. The 21 available topic codes for the `topics()` option are listed in the help file (`help wbopendata`).

Suggested citation: Azevedo, João Pedro. 2011. "WBOPENDATA: Stata module to access World Bank databases." Statistical Software Components S457234, Boston College Department of Economics.