wbdata Documentation

Release 0.2.7

Oliver Sherouse

January 20, 2017

Contents

CHAPTER 1

What is wbdata?

Wbdata is a simple python interface to find and request information from the World Bank's various databases, either as a dictionary containing full metadata or as a pandas DataFrame. Currently, wbdata wraps most of the World Bank API, and also adds some convenience functions for searching and retrieving information.

Wbdata was designed to be used either in a script or in a shell. In a shell, wbdata assumes that the user will use most functions to look up the codes necessary to retrieve the information he wants. To this end, the default in shell mode for most functions is to simply print the id and human-readable name of each item in question. In a script, the default is to return the entire response from the World Bank converted to python objects.

All the functions that you need to get started are in the wbdata library reference module.

Finally, it should be pointed out that wbdata is in the "release early" portion of the "release early, release often" cycle, and the current test suite is pretty perfunctory. You won't end up with the wrong data, but any irregularities I haven't specifically encountered in the World Bank database have not been dealt with.

CHAPTER 2

Full Package Documentation

2.1 wbdata library reference

Wbdata provides a set of functions that are used to interface with the World Bank's databases. For any function involving pandas capabilities, pandas must (obviously) be installed.

2.1.1 Convenience Functions

wbdata.**search_indicators**(*query*, *source=None*, *topic=None*, *display=None*)

Search indicators for a certain term. Very simple. Only one of source or topic can be specified. In interactive mode, will return None and print ids and names unless suppress_printing is True.

Query the term to match against indicator names

Source if present, id of desired source

Topic if present, id of desired topic

Display if True,print ids and names instead of returning results. Defaults to True if in interactive prompt, or False otherwise

Returns a list of dictionaries representing indicators if display is False

wbdata.**search_countries**(*query*, *incomelevel=None*, *lendingtype=None*, *display=None*)

Search countries by name. Very simple search.

Query the string to match against country names

Incomelevel if present, search only the matching incomelevel

Lendingtype if present, search only the matching lendingtype

Display if True,print ids and names instead of returning results. Defaults to True if in interactive prompt, or False otherwise

Returns a list of dictionaries representing countries if display is False

wbdata.**get_dataframe**(*indicators*, *country=u'all'*, *data_date=None*, *convert_date=False*, *keep_levels=False*)

Convenience function to download a set of indicators and merge them into a pandas DataFrame. The index will be the same as if calls were made to get_data separately.

Indicators An dictionary where the keys are desired indicators and the values are the desired column names

Country a country code, sequence of country codes, or "all" (default)

Data_date the desired date as a datetime object or a 2-sequence with start and end dates

Convert_date if True, convert date field to a datetime.datetime object.

Keep_levels if True and pandas is True, don't reduce the number of index levels returned if only getting one date or country

Returns a pandas dataframe

wbdata.**get_panel**(*indicators*, *country=u'all'*, *data_date=None*, *convert_date=False*, *items=u'indicators'*, *major_axis=u'dates'*)

Convenience function to download a set of indicators and merge them into a pandas Panel.

Indicators An dictionary where the keys are desired indicators and the values are the desired column names

Country a country code, sequence of country codes, or "all" (default)

Data_date a 2-sequence with start and end dates

Convert_date if True, convert date field to a datetime.datetime object.

Items which values to use as the Panel items. One of "indicators", "countries", "dates"

Major_axis which values to use as major axis for each item. One of "indicators", "countries", "dates"

Returns a pandas panel

2.1.2 Finding the data you want

wbdata.**get_source**(*source_id=None*, *display=None*)
Retrieve information on a source

Source_id a source id or sequence thereof. None returns all sources

Display if True,print ids and names instead of returning results. Defaults to True if in interactive prompt, or False otherwise

Returns if display is False, a dictionary describing a source

wbdata.**get_topic**(*topic_id=None*, *display=None*)
Retrieve information on a topic

Topic_id a topic id or sequence thereof. None returns all topics

Display if True,print ids and names instead of returning results. Defaults to True if in interactive prompt, or False otherwise

Returns if display is False, a dictionary describing an income level aggregate

wbdata.**get_lendingtype**(*type_id=None*, *display=None*)
Retrieve information on an income level aggregate

Level_id lending type id or sequence thereof. None returns all lending type aggregates

Display if True,print ids and names instead of returning results. Defaults to True if in interactive prompt, or False otherwise

Returns if display is False, a dictionary describing an lending type aggregate

wbdata.**get_incomelevel**(*level_id=None*, *display=None*)
Retrieve information on an income level aggregate

Level_id a level id or sequence thereof. None returns all income level aggregates

Display if True,print ids and names instead of returning results. Defaults to True if in interactive prompt, or False otherwise

Returns if display is False a dictionary describing an income level aggregate

wbdata.**get_country**(*country_id=None*, *incomelevel=None*, *lendingtype=None*, *display=None*)
Retrieve information on a country or regional aggregate. Can specify either country_id, or the aggregates, but not both

Country_id a country id or sequence thereof. None returns all countries and aggregates.

Incomelevel desired incomelevel id or ids.

Lendingtype desired lendingtype id or ids.

Display if True,print ids and names instead of returning results. Defaults to True if in interactive prompt, or False otherwise.

Returns if display is False, a dictionary describing an lending type aggregate.

wbdata.**get_indicator**(*indicator=None*, *source=None*, *topic=None*, *display=None*)
Retrieve information about an indicator or indicators. Only one of indicator, source, and topic can be specified. Specifying none of the three will return all indicators.

Indicator an indicator code or sequence thereof

Source a source id or sequence thereof

Topic a topic id or sequence thereof

Display if True,print ids and names instead of returning results. Defaults to True if in interactive prompt, or False otherwise

Returns if display is False, a list of dictionary objects representing indicators

2.1.3 Retrieving your data

wbdata.**get_data**(*indicator*, *country=u'all'*, *data_date=None*, *convert_date=False*, *pandas=False*, *column_name=u'value'*, *keep_levels=False*)
Retrieve indicators for given countries and years

Indicator the desired indicator code

Country a country code, sequence of country codes, or "all" (default)

Date the desired date as a datetime object or a 2-tuple with start and end dates

Convert_date if True, convert date field to a datetime.datetime object.

Pandas if True, return results as a pandas Series. The index will be the date of the data if only one country is specified, the countries if only one date is specified, or a multi-index of country and date otherwise.

Column_name the desired name for the pandas column

Keep_levels if True and pandas is True, don't reduce the number of index levels returned if only getting one date or country

Returns list of dictionaries or pandas Series

2.2 wbdata.fetcher library reference

Fetcher is the mechanism used by wbdata for reading, paging, caching, and converting responses from the World Bank. Luckily, most users will have no need to deal with this library, which is why it is separate. wbdata.fetcher: retrieve and cache queries

class wbdata.fetcher.**Cache**
Docstring for Cache

sync()
Sync cache to disk

wbdata.fetcher.**fetch**(*query_url*, *args=None*, *cached=True*)
fetch data from the World Bank API or from cache

Query_url the base url to be queried

Args a dictionary of GET arguments

Cached use the cache

Returns a list of dictionaries containing the response to the query

wbdata.fetcher.**fetch_url**(*url*)
Fetch a url directly from the World Bank, up to TRIES tries

Url the url to retrieve

Returns a string with the url contents

2.3 wbdata.api library reference

The api library is where all the nuts-and-bolts functions are defined, as well as those which are imported into the main namespace. wbdata.api: Where all the functions go

wbdata.api.**cast_float**(*value*)
Return a floated value or none

wbdata.api.**convert_dates_to_datetime**(*data*)
Return a datetime.datetime object from a date string as provided by the World Bank

wbdata.api.**convert_month_to_datetime**(*monthstr*)
return datetime.datetime object from %YM%m formatted string

wbdata.api.**convert_quarter_to_datetime**(*quarterstr*)
return datetime.datetime object from %YQ%# formatted string, where # is the desired quarter

wbdata.api.**convert_to_dataframe**(*data*, *column_name*)
Convert a set of values to a dataframe with columns for country and date

wbdata.api.**convert_year_to_datetime**(*yearstr*)
return datetime.datetime object from %Y formatted string

wbdata.api.**get_country**(*country_id=None*, *incomelevel=None*, *lendingtype=None*, *display=None*)
Retrieve information on a country or regional aggregate. Can specify either country_id, or the aggregates, but not both

Country_id a country id or sequence thereof. None returns all countries and aggregates.

Incomelevel desired incomelevel id or ids.

Lendingtype desired lendingtype id or ids.

Display if True,print ids and names instead of returning results. Defaults to True if in interactive prompt, or False otherwise.

Returns if display is False, a dictionary describing an lending type aggregate.

wbdata.api.**get_data**(*indicator*, *country=u'all'*, *data_date=None*, *convert_date=False*, *pandas=False*, *column_name=u'value'*, *keep_levels=False*)

Retrieve indicators for given countries and years

Indicator the desired indicator code

Country a country code, sequence of country codes, or "all" (default)

Date the desired date as a datetime object or a 2-tuple with start and end dates

Convert_date if True, convert date field to a datetime.datetime object.

Pandas if True, return results as a pandas Series. The index will be the date of the data if only one country is specified, the countries if only one date is specified, or a multi-index of country and date otherwise.

Column_name the desired name for the pandas column

Keep_levels if True and pandas is True, don't reduce the number of index levels returned if only getting one date or country

Returns list of dictionaries or pandas Series

wbdata.api.**get_dataframe**(*indicators*, *country=u'all'*, *data_date=None*, *convert_date=False*, *keep_levels=False*)

Convenience function to download a set of indicators and merge them into a pandas DataFrame. The index will be the same as if calls were made to get_data separately.

Indicators An dictionary where the keys are desired indicators and the values are the desired column names

Country a country code, sequence of country codes, or "all" (default)

Data_date the desired date as a datetime object or a 2-sequence with start and end dates

Convert_date if True, convert date field to a datetime.datetime object.

Keep_levels if True and pandas is True, don't reduce the number of index levels returned if only getting one date or country

Returns a pandas dataframe

wbdata.api.**get_incomelevel**(*level_id=None*, *display=None*)

Retrieve information on an income level aggregate

Level_id a level id or sequence thereof. None returns all income level aggregates

Display if True,print ids and names instead of returning results. Defaults to True if in interactive prompt, or False otherwise

Returns if display is False a dictionary describing an income level aggregate

wbdata.api.**get_indicator**(*indicator=None*, *source=None*, *topic=None*, *display=None*)

Retrieve information about an indicator or indicators. Only one of indicator, source, and topic can be specified. Specifying none of the three will return all indicators.

Indicator an indicator code or sequence thereof

Source a source id or sequence thereof

Topic a topic id or sequence thereof

Display if True,print ids and names instead of returning results. Defaults to True if in interactive prompt, or False otherwise

Returns if display is False, a list of dictionary objects representing indicators

wbdata.api.**get_lendingtype**(*type_id=None*, *display=None*)
Retrieve information on an income level aggregate

Level_id lending type id or sequence thereof. None returns all lending type aggregates

Display if True,print ids and names instead of returning results. Defaults to True if in interactive prompt, or False otherwise

Returns if display is False, a dictionary describing an lending type aggregate

wbdata.api.**get_panel**(*indicators*, *country=u'all'*, *data_date=None*, *convert_date=False*, *items=u'indicators'*, *major_axis=u'dates'*)

Convenience function to download a set of indicators and merge them into a pandas Panel.

Indicators An dictionary where the keys are desired indicators and the values are the desired column names

Country a country code, sequence of country codes, or "all" (default)

Data_date a 2-sequence with start and end dates

Convert_date if True, convert date field to a datetime.datetime object.

Items which values to use as the Panel items. One of "indicators", "countries", "dates"

Major_axis which values to use as major axis for each item. One of "indicators", "countries", "dates"

Returns a pandas panel

wbdata.api.**get_source**(*source_id=None*, *display=None*)
Retrieve information on a source

Source_id a source id or sequence thereof. None returns all sources

Display if True,print ids and names instead of returning results. Defaults to True if in interactive prompt, or False otherwise

Returns if display is False, a dictionary describing a source

wbdata.api.**get_topic**(*topic_id=None*, *display=None*)
Retrieve information on a topic

Topic_id a topic id or sequence thereof. None returns all topics

Display if True,print ids and names instead of returning results. Defaults to True if in interactive prompt, or False otherwise

Returns if display is False, a dictionary describing an income level aggregate

wbdata.api.**id_only_query**(*query_url*, *query_id*, *display*)
Retrieve information when ids are the only arguments

Query_url the base url to use for the query

Query_id an id number. None returns all sources

Display if True,print ids and names instead of returning results. Defaults to True if in interactive prompt, or False otherwise

Returns if display is False, a dictionary describing a source

wbdata.api.**parse_value_or_iterable**(*arg*)
If arg is a single value, return it as a string; if an iterable, return a ;-joined string of all values

wbdata.api.**print_ids_and_names**(*objs*)
Courtesy function to display ids and names from lists returned by wbdata. This will mostly be useful in interactive mode.

Objs a list of dictionary objects as returned by wbdata

wbdata.api.**search_countries**(*query*, *incomelevel=None*, *lendingtype=None*, *display=None*)
Search countries by name. Very simple search.

Query the string to match against country names

Incomelevel if present, search only the matching incomelevel

Lendingtype if present, search only the matching lendingtype

Display if True,print ids and names instead of returning results. Defaults to True if in interactive prompt, or False otherwise

Returns a list of dictionaries representing countries if display is False

wbdata.api.**search_indicators**(*query*, *source=None*, *topic=None*, *display=None*)
Search indicators for a certain term. Very simple. Only one of source or topic can be specified. In interactive mode, will return None and print ids and names unless suppress_printing is True.

Query the term to match against indicator names

Source if present, id of desired source

Topic if present, id of desired topic

Display if True,print ids and names instead of returning results. Defaults to True if in interactive prompt, or False otherwise

Returns a list of dictionaries representing indicators if display is False

wbdata.api.**uses_pandas**(*f*)
Raise ValueError if pandas is not loaded

CHAPTER 3

Installation

Wbdata is available on PyPi which means you can download it from there or install using easy_install or pip:

```
easy_install wbdata
```

```
pip install wbdata
```

You can also download or get the source from GitHub.

CHAPTER 4

Suggested Citation

If you use wbdata in a paper, it would be great to cite it. It lets me know to keep working on it, and it lets other researchers know how to make their lives easier. Suggested citation would be something like:

> Sherouse, Oliver (2014). Wbdata. Arlington, VA. Available from http://github.com/OliverSherouse/wbdata.

CHAPTER 5

A Typical User Session

Let's say we want to find some data for the ease of doing business in some well-off countries. I might start off by seeing what sources are available and look promising:

```
>>> wbdata.get_source()
11  Africa Development Indicators
31  Country Policy and Institutional Assessment (CPIA)
26  Corporate Scorecard
1   Doing Business
30  Exporter Dynamics Database: Country-Year
12  Education Statistics
13  Enterprise Surveys
28  Global Findex ( Global Financial Inclusion database)
33  G20 Basic Set of Financial Inclusion Indicators
14  Gender Statistics
15  Global Economic Monitor
27  GEP Economic Prospects
32  Global Financial Development
21  Global Economic Monitor (GEM) Commodities
29  Global Social Protection
16  Health Nutrition and Population Statistics
18  International Development Association - Results Measurement System
6   International Debt Statistics
25  Jobs for Knowledge Platform
19  Millennium Development Goals
24  Povstats
20  Public Sector Debt
23  Quarterly External Debt Statistics (QEDS) - General Data Dissemination System (GDDS)
22  Quarterly External Debt Statistics (QEDS) - Special Data Dissemination Standard (SDDS)
2   World Development Indicators
3   Worldwide Governance Indicators
```

Well, that "Doing Business"—source 1—looks like a winner. Let's see what we've got available to us there.

```
>>> wbdata.get_indicator(source=1)
IC.BUS.EASE.XQ      Ease of doing business index (1=most business-friendly regulations)
IC.CRD.INFO.XQ      Credit depth of information index (0=low to 6=high)
IC.CRD.PRVT.ZS      Private credit bureau coverage (% of adults)
IC.CRD.PUBL.ZS      Public credit registry coverage (% of adults)
IC.DCP.COST         Cost to build a warehouse (% of income per capita)
IC.DCP.PROC         Procedures required to build a warehouse (number)
IC.DCP.TIME         Time required to build a warehouse (days)
IC.EC.COST          Cost to enforce a contract (% of claim)
IC.EC.PROC          Procedures required to enforce a contract (number)
```

```
IC.EC.TIME          Time required to enforce a contract (days)
IC.EXP.COST.EXP     Trade: Cost to export (US$ per container)
IC.EXP.COST.IMP     Trade: Cost to import (US$ per container)
IC.EXP.DOCS         Documents to export (number)
IC.EXP.DOCS.IMP     Trade: Documents to import (number)
IC.EXP.TIME.EXP     Trade: Time to export (day)
IC.EXP.TIME.IMP     Trade: Time to import (days)
IC.GE.COST          Cost to get electricity(% of income per capita)
IC.GE.NUM           Procedures required to connect to electricity (number)
IC.GE.TIME          Time required to connect to electricity (days)
IC.IMP.DOCS         Documents to import (number)
IC.ISV.COST         Resolving insolvency: cost (% of estate)
IC.ISV.DURS         Time to resolve insolvency (years)
IC.ISV.RECRT        Resolving insolvency: recovery rate (cents on the dollar)
IC.LGL.CRED.XQ      Strength of legal rights index (0=weak to 10=strong)
IC.LIC.NUM          Procedures required to build a warehouse (number)
IC.LIC.TIME         Time required to build a warehouse (days)
IC.PI.DIR           Extent of director liability index (0 to 10)
IC.PI.DISCL         Extent of disclosure index (0 to 10)
IC.PI.INV           Strength of investor protection index (0 to 10)
IC.PI.SHAR          Ease of shareholder suits index (0 to 10)
IC.REG.CAP          Minimum paid-in capital required to start a business (% of income per capita)
IC.REG.COST         Cost to start a business (% of income per capita)
IC.REG.DURS         Time required to start a business (days)
IC.REG.PROC         Start-up procedures to register a business (number)
IC.RP.COST          Cost to register property (% of property value)
IC.RP.PROC          Procedures required to register property (number)
IC.RP.TIME          Time required to register property (days)
IC.TAX.DURS         Time to prepare and pay taxes (hours)
IC.TAX.PAYM         Tax payments (number)
IC.TAX.TOTL.CP.ZS   Total tax rate (% of commercial profits)
```

Alrighty. There's a lot there. But let's say I'm in the early stages of developing a question and go for the most general measure, which is the first one, the "Ease of Doing Business Index", which has the id "IC.BUS.EASE.XQ".

Now remember, we're only interested in high-income countries right now, because we're elitist. So let's use one of the convenience search functions to figure out the code for the United States so we don't have to wait for data from a bunch of other countries:

```
>>> wbdata.search_countries("united")
ARE United Arab Emirates
GBR United Kingdom
USA United States
```

"USA". Very creative. Thank you, World Bank. But in any case, let's get our data:

```
>>> wbdata.get_data("IC.BUS.EASE.XQ", country=USA)
```

And that will return a big long list of dictionaries with all the relevant data and metadata as organized by the World Bank. Now let's say we want to look at the United Kingdom as well ("GBR", see above), and only for the years 2010-2011. We can actually search using multiple countries and restrict the dates using datetime objects. Here's what that would look like:

```
>>> data_date = (datetime.datetime(2010, 1, 1), datetime.datetime(2011, 1, 1))
>>> wbdata.get_data("IC.BUS.EASE.XQ", country=("USA", "GBR"), data_date=data_date)
```

And we get another long list of dictionaries, which we can parse any which way we please.

So let's get a little bit more econometric. Let's say we want to fetch this same indicator, but also gdp per capita and for all OECD countries. Let's find the other indicator we want using another convenience search function:

```
>>> wbdata.search_indicators("gdp per capita")
GDPPCKD                      GDP per Capita, constant US$, millions
GDPPCKN                      Real GDP per Capita (real local currency units, various base years)
NV.AGR.PCAP.KD.ZG            Real agricultural GDP per capita growth rate (%)
NY.GDP.PCAP.CD               GDP per capita (current US$)
NY.GDP.PCAP.KD               GDP per capita (constant 2000 US$)
NY.GDP.PCAP.KD.ZG            GDP per capita growth (annual %)
NY.GDP.PCAP.KN               GDP per capita (constant LCU)
NY.GDP.PCAP.PP.CD            GDP per capita, PPP (current international $)
NY.GDP.PCAP.PP.KD            GDP per capita, PPP (constant 2005 international $)
NY.GDP.PCAP.PP.KD.ZG         GDP per capita, PPP annual growth (%)
SE.XPD.PRIM.PC.ZS            Expenditure per student, primary (% of GDP per capita)
SE.XPD.SECO.PC.ZS            Expenditure per student, secondary (% of GDP per capita)
SE.XPD.TERT.PC.ZS            Expenditure per student, tertiary (% of GDP per capita)
```

Like good economists, we'll use the one that seems most impressive: GDP per Capita at PPP in constant 2005 dollars, which has the id "NY.GDP.PCAP.PP.KD". But what about using OECD countries?

```
>>> wbdata.get_incomelevel()
HIC High income
HPC Heavily indebted poor countries (HIPC)
INX Not classified
LIC Low income
LMC Lower middle income
LMY Low & middle income
MIC Middle income
NOC High income: nonOECD
OEC High income: OECD
UMC Upper middle income
```

Funtastic. Finally, let's make sure we get our data into a lovely merged pandas DataFrame, suitable for analysis with that library, statsmodels, or whatever else we'd like.

```
>>> countries = [i['id'] for i in wbdata.get_country(incomelevel="OEC", display=False)]
>>> indicators = {"IC.BUS.EASE.XQ": "doing_business", "NY.GDP.PCAP.PP.KD": "gdppc"}
>>> df = wbdata.get_dataframe(indicators, country=countries, convert_date=True)
>>> df.describe()
              gdppc  doing_business
count    943.000000       62.000000
mean   25394.219689       29.322581
std     9495.732499       21.946861
min     5543.572247        3.000000
25%    19278.418595       11.000000
50%    24640.540261       27.500000
75%    30655.760574       41.000000
max    74021.456759       89.000000
>>> df = df.dropna()
>>> df.gdppc.corr(df.doing_business)
-0.2858022507189249
```

And, since lower scores on that indicator mean more business-friendly regulations, that's exactly what we would expect. It goes without saying that we can use our data now to do any other analysis required.

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

W

C

F

G

I

P

S

U

W