Package 'GDPuc'

November 19, 2025

Title Easily Convert GDP Data

Version 1.6.0

Date 2025-11-19

Description Convert GDP time series data from one unit to another. All common GDP units are included, i.e. current and constant local currency units, US$ via market exchange rates and international dollars via purchasing power parities.

License GPL (>= 3)

URL https://github.com/pik-piam/GDPuc, https://pik-piam.github.io/GDPuc/

BugReports https://github.com/pik-piam/GDPuc/issues

Depends R (>= 2.10)

Imports cli (>= 2.4.0), crayon, dplyr, glue, magrittr, rlang (>= 1.0.0), tibble, tidyr, tidyselect, withr

Suggests countrycode, covr, knitr, magclass, madrat (>= 3.6.4), purrr, rmarkdown, stringr, testthat (>= 3.0.0), usethis, WDI, zoo

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.3

NeedsCompilation no

Author Johannes Koch [aut, cre]

Maintainer Johannes Koch <jokoch@pik-potsdam.de>

Repository CRAN

Date/Publication 2025-11-19 17:10:02 UTC

Contents

---

convertGDP *Convert GDP data*

Description

convertGDP() converts GDP time series data from one unit to another, using GDP deflators, market exchange rates (MERs) and purchasing power parity conversion factors (PPPs).

Usage

```
convertGDP(
  gdp,
  unit_in,
  unit_out,
  source = "wb_wdi",
  use_USA_cf_for_all = FALSE,
  with_regions = NULL,
  replace_NAs = NULL,
  verbose = getOption("GDPuc.verbose", default = FALSE),
  return_cfs = FALSE,
  iso3c_column = "iso3c",
  year_column = "year"
)

convertCPI(...)

convertSingle(x, iso3c, year = NULL, unit_in, unit_out, ...)

toolConvertGDP(
  gdp,
  unit_in,
  unit_out,
  source = "wb_wdi",
  use_USA_cf_for_all = FALSE,
  with_regions = NULL,
  replace_NAs = NULL,
  verbose = getOption("GDPuc.verbose", default = FALSE),
  return_cfs = FALSE,
  iso3c_column = "iso3c",
  year_column = "year"
)

toolConvertSingle(x, iso3c, year = NULL, unit_in, unit_out, ...)

toolConvertCPI(...)
```

Arguments

`gdp` A tibble, data frame or magpie object, the latter of which requires the magclass package to be installed. The data-frame needs to have at least 2 columns, in some cases 3:

- a character column with iso3c (wikipedia) country codes,
- a numeric column with years (only required when converting from or to current currencies),
- a numeric column named "value" with GDP values.

`unit_in` A string with the incoming GDP unit, one of:

- "current LCU"
- "current Int$PPP"
- "current US$MER"
- "constant YYYY LCU"
- "constant YYYY Int$PPP"
- "constant YYYY US$MER"
- "constant YYYY €" or "constant YYYY EUR"
- "constant YYYY xxx_CU"

where YYYY should be replaced with a year e.g. "2010" or "2017".

`unit_out` A string with the outgoing GDP unit, one of:

- "current LCU"
- "current Int$PPP"
- "current US$MER"
- "constant YYYY LCU"
- "constant YYYY Int$PPP"
- "constant YYYY US$MER"
- "constant YYYY €" or "constant YYYY EUR"
- "constant YYYY xxx_CU"

where YYYY should be replaced with a year e.g. "2010" or "2017", and xxx with a valid iso3c country code, e.g. "JPN_CU" to pick the currency unit of Japan.

`source` A string referring to a package internal data frame containing the conversion factors, or a data-frame that exists in the calling environment. Use print_source_info() to learn about the available sources.

`use_USA_cf_for_all` TRUE or FALSE (default). If TRUE, then the USA conversion factors are used for all countries.

`with_regions` NULL by default, meaning no regional codons are recognized. To convert regional data, a "country to region mapping" must be passed to the function. Any regions will then be disaggregated according to the region mapping and weighed by the GDP share of countries in that region in the year of the unit (only constant units are compatible with with_regions not equal NULL), converted on a country level, and re-aggregated before being returned. Can be set to one of the following:

- A character string referring to a madrat regionmapping. Requires madrat to be installed, and the mapping to be accessible via `madrat::toolGetMapping()`.
- A data-frame with a country to region mapping: one column named "iso3c" with iso3c country codes, and one column named "region" with region codes to which the countries belong.

`replace_NAs` NULL by default, meaning no NA replacement. Can be set to one of the following:

- 0: resulting NAs are simply replaced with 0.
- NA: resulting NAs are explicitly kept as NA.
- "no_conversion": resulting NAs are simply replaced with the values from the gdp argument.
- "linear": missing conversion factors in the source object are inter- and extrapolated linearly. For the extrapolation, the closest 5 data points are used.
- "regional_average": missing conversion factors in the source object are replaced with the regional average of the region to which the country belongs. This requires a region-mapping to be passed to the function, see the with_regions argument.
- "with_USA": missing conversion factors in the source object are extended using US growth rates. If that is not possible (for instance if the conversion factor is missing entirely) the conversion factors are replaced with US ones. For example, if the conversion requires PPPs and deflators, but the PPPs are missing entirely, then even though there is deflator data, it is the the US deflator that is used.

Can also be a vector with "linear" as first element, e.g. c("linear", 0) or c("linear", "no_conversion"), in which case, the operations are done in sequence.

`verbose` TRUE or FALSE. A flag to turn verbosity on or off. Be default it is equal to the GDPuc.verbose option, which is FALSE if not set to TRUE by the user.

`return_cfs` TRUE or FALSE. Set to TRUE to additionally return a tibble with the conversion factors used. In that case a list is returned with the converted GDP under "result", and the conversion factors used under "cfs".

`iso3c_column` String designating the name of the column containing the iso3c codes. Defaults to "iso3c".

`year_column` String designating the name of the column containing the years. Defaults to "year".

`...` Arguments passed on to `convertGDP()`

`x` Number to convert

`iso3c` Country code

`year` NULL, or year of value. Only plays a role when converting from or to current currencies.

Details

When providing a custom source to the function, a certain format is required. The source object must be a data frame or tibble with at least the following columns:

- a character column named "iso3c" with iso3c (wikipedia) country codes,
- a numeric column named "year" with years,
- a numeric column named "GDP deflator" with values of the GDP deflator divided by 100 (so that in the base year the GDP deflator is equal to 1, not 100). The base year of the deflator can be any year, and can be country-specific.
- a numeric column named "MER (LCU per US$)" with MER values,
- a numeric column named "PPP conversion factor, GDP (LCU per international $)" wit PPP exchange rate values.

Value

The gdp argument, with the values in the "value" column, converted to unit_out. If the argument return_cfs is TRUE, then a list is returned with the converted GDP under "result", and the conversion factors used under "cfs".

Functions

- `convertCPI()`: Short cut for `convertGDP(..., source = "wb_wdi_cpi")`
- `convertSingle()`: Convert a single value, while specifying iso3c code and year. Simpler than creating a single row tibble.
- `toolConvertGDP()`: Madrat wrapper around `convertGDP()`
- `toolConvertSingle()`: Madrat wrapper around `convertSingle()`
- `toolConvertCPI()`: Madrat wrapper around `convertCPI(...)`

See Also

The countrycode package to convert country codes.

Examples

```
my_tbble <- tibble::tibble(iso3c = "FRA",
                           year = 2013,
                           value = 100)

convertGDP(gdp = my_tbble,
           unit_in = "current LCU",
           unit_out = "constant 2015 Int$PPP")

# Convert using the CPI as deflator.
convertGDP(gdp = my_tbble,
           unit_in = "current LCU",
           unit_out = "constant 2015 Int$PPP",
           source = "wb_wdi_cpi")
# Or using the shortcut `convertCPI()`
convertCPI(gdp = my_tbble,
           unit_in = "current LCU",
           unit_out = "constant 2015 Int$PPP")
```

```
# Convert a single value quickly
convertSingle(x = 100,
              iso3c = "FRA",
              year = 2013,
              unit_in = "current LCU",
              unit_out = "constant 2015 Int$PPP")
```

`print_source_info` *Print information on sources*

Description

Print detailed information on conversion factor sources to the screen. Information includes the name, origin, date, html-link and an associated note. Calling the function without any argument will print information on all available sources.

Usage

```
print_source_info(source)
```

Arguments

`source`	Empty, or the name of one of the internal sources: 1. "wb_wdi" 2. "wb_wdi_linked" 3. "wb_wdi_cpi"

Value

No return value, called for side effects.

Examples

```
print_source_info()
```

Index