

# Package ‘GDPuc’

July 20, 2023

**Title** Easily Convert GDP Data

**Version** 0.11.1

**Date** 2023-07-13

**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, lifecycle, magrittr,  
rlang (>= 1.0.0), tibble, tidyr, tidyselect, withr

**Suggests** covr, knitr, magclass, purrr, rmarkdown, stringr, testthat  
(>= 3.0.0), usethis, WDL, zoo

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**NeedsCompilation** no

**Author** Johannes Koch [aut, cre]

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

**Repository** CRAN

**Date/Publication** 2023-07-20 11:50:02 UTC

## R topics documented:

convertGDP . . . . .	2
print_source_info . . . . .	4

---

convertGDP	<i>Convert GDP data</i>
------------	-------------------------

---

### Description

#### [Stable]

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",
  with_regions = NULL,
  replace_NAs = NULL,
  verbose = getOption("GDPuc.verbose", default = FALSE),
  return_cfs = FALSE
)
```

### Arguments

gdp	<p>A tibble, data frame or magpie object, the latter of which requires the <a href="#">magclass</a> package to be installed. The data-frame needs to have at least 3 columns:</p> <ul style="list-style-type: none"> <li>• a character column with iso3c (<a href="#">wikipedia</a>) country codes,</li> <li>• a numeric column with years,</li> <li>• a numeric column named "value" with GDP values.</li> </ul>
unit_in	<p>A string with the incoming GDP unit, one of:</p> <ul style="list-style-type: none"> <li>• "current LCU"</li> <li>• "current Int\$PPP"</li> <li>• "current US\$MER"</li> <li>• "constant YYYY LCU"</li> <li>• "constant YYYY Int\$PPP"</li> <li>• "constant YYYY US\$MER"</li> </ul> <p>where YYYY should be replaced with a year e.g. "2010" or "2017".</p>
unit_out	<p>A string with the outgoing GDP unit, one of:</p> <ul style="list-style-type: none"> <li>• "current LCU"</li> <li>• "current Int\$PPP"</li> <li>• "current US\$MER"</li> <li>• "constant YYYY LCU"</li> </ul>

- "constant YYYY Int\$PPP"
- "constant YYYY US\$MER"

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

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 <code>print_source_info()</code> to learn about the available sources.
with_regions	NULL or a data-frame. The data-frame should be "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. Any regions in the gdp object 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, converted on a country level, and re-aggregated before being returned.
replace_NAs	<p>NULL by default, meaning no NA replacement. Can be set to one of the following:</p> <ul style="list-style-type: none"> <li>• 0: resulting NAs are simply replaced with 0.</li> <li>• NA: resulting NAs are explicitly kept as NA.</li> <li>• "no_conversion": resulting NAs are simply replaced with the values from the gdp argument.</li> <li>• "linear": missing conversion factors in the source object are inter- and extrapolated linearly. For the extrapolation, the closest 5 data points are used.</li> <li>• "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.</li> </ul> <p>Can also be a vector with "linear" as first element, e.g. <code>c("linear", 0)</code> or <code>c("linear", "no_conversion")</code>, in which case, the operations are done in sequence.</p>
verbose	TRUE or FALSE. A flag to turn verbosity on or off. By default it is equal to the <code>GDPuc.verbose</code> 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".

## 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 \$)" with 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".

**See Also**

The `countrycode` package to convert country codes.

---

`print_source_info`      *Print information on sources*

---

**Description****[Stable]**

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`            The name of one of the internal sources:

1. "wb\_wdi"
2. "wb\_wdi\_linked"

**Value**

No return value, called for side effects.

# Index

`convertGDP`, 2

`print_source_info`, 4