Package 'africamonitor'

July 4, 2022

Title Africa Macroeconomic Monitor Database API

Version 0.2.0

Description An R API providing access to a relational database with macroeconomic data for Africa.
The database contains >700 macroeconomic time series from mostly international sources,
grouped into 50 macroeconomic and development-related topics. Series are carefully selected
on the basis of data coverage for Africa, frequency, and relevance to the macro-development context.
The project is part of the 'Kiel Institute Africa Initiative'
<https://www.ifw-kiel.de/institute/initiatives/kielinstituteafricainitiative/>,
which, amongst other things, aims to develop a parsimonious database with highly relevant indicators
to monitor macroeconomic developments in Africa, accessible through a fast API and a web-based platform
at <https://africamonitor.ifw-kiel.de/>.
The database is maintained at the Kiel Institute for the World Economy <https://www.ifw-kiel.de/>.

URL https://africamonitor.ifw-kiel.de/

License GPL-3

Encoding UTF-8

Imports DBI, RMySQL, data.table, collapse

RoxygenNote 7.1.2

Depends R (>= 3.3.0)

LazyData true

NeedsCompilation no

Author Sebastian Krantz [aut, cre]

Maintainer Sebastian Krantz <sebastian.krantz@graduateinstitute.ch>

Repository CRAN

Date/Publication 2022-07-04 12:10:05 UTC

R topics documented:

---

africamonitor-package *Africa Macroeconomic Monitor Database API*

Description

An R API providing access to a relational database with macroeconomic data for Africa. The database is maintained at the Kiel Institute for the World Economy.

Functions

Functions and data providing information about the available data

am_sources()
am_series()
am_countries
am_entities

Function to retrieve the data from the database

am_data()

Functions to reshape data and add temporal identifiers

am_pivot_wider()
am_pivot_longer()
am_expand_date()

Helper functions to convert inputs to R dates

am_as_date()

Global Macros with core ID variables in the database

.AMID
.AMT

.AMID *Global Identifier Macros*

Description

The macro .AMID contains the string c("ISO3", "Series") denoting the names of ID variables that identify the cross-sectional dimension in the database.

The macro .AMT contains the string c("Date", "Year", "Quarter", "FY", "QFY", "Month", "Day") denoting temporal identifiers generated by am_expand_date. The "Date" variable is sufficient to uniquely identify a point in time in the database.

Each value in the database is uniquely identified by ISO3, Series and Date.

Usage

```
.AMID
.AMT
```

See Also

africamonitor

Examples

```
.AMID
.AMT
```

am_as_date *Coerce Vectors to Dates*

Description

This function coerces date strings i.e. "YYYY-MM-DD" or "YYYY-MM", years e.g. 2015 (numeric or character), year-quarters e.g. "2015Q1" or "2015-Q1", year-months e.g. "2015M01" or "2015-M01", fiscal years e.g. "1997/98" or numeric values representing dates (e.g. previously imported Excel date) to a regular R date.

Usage

```
am_as_date(x, end = FALSE, origin = "1899-12-30")
```

Arguments

`x` a character date string `"YYYY-MM-DD"` or `"YYYY-MM"`, year-quarter `"YYYYQN"` or `"YYYY-QN"`, , year-month `"YYYYMNN"` or `"YYYY-MNN"`, fiscal year `"YYYY/YY"` or calendar year `YYYY` (numeric or character), or a numeric value corresponding to a date that can be passed to `as.Date.numeric`.

`end` logical. `TRUE` replaces missing time information with a period end-date which is the last day of the period. `FALSE` replaces missing month and day information with `"-01"`, so the year date is the 1st of January, the fiscal year date the 1st of July, and for months / quarters the 1st day of the month / quarter.

`origin` a date or date-string that can be used as reference for converting numeric values to dates. The default corresponds to dates generated in Excel for Windows. See `as.Date.numeric`.

Value

A `Date` vector.

See Also

`am_expand_date`, `africamonitor`

Examples

```
am_as_date("2011-05")
am_as_date(2011)
am_as_date("2011/12")
am_as_date("2011/12", end = TRUE)
am_as_date("2011Q1")
am_as_date("2011Q1", end = TRUE)
```

am_countries *Dataset of Countries in the Database*

Description

A dataset containing standardized codes of African countries for which data is provided according to various classifications and regional aggregations. Note that the API (`am_data`) only supports "ISO3" character codes.

Usage

```
am_countries
```

Format

A data frame with 55 rows and 9 variables:

Country Short Country Name

Country_ISO ISO Standardized Country Name

ISO2 ISO 2-Character Country Code

ISO3 ISO 3-Character Country Code

ISO3N ISO Numeric Country Code

IMF IMF Numeric Country Code

Region 2-Region Classification (UN except for Sudan)

Region_Detailed 5-Region Classification (Former World Bank)

Currency Main Official Currency

Source

countrycode R package (with some modification of regional aggregates).

See Also

`am_entities`, `am_sources`, `am_series`, `africamonitor`

Examples

```
head(am_countries)
```

`am_data` *Retrieve Data from the Database*

Description

This is the main function of the package to retrieve data from the database.

Usage

```
am_data(
  ctry = NULL,
  series = NULL,
  from = NULL,
  to = NULL,
  labels = TRUE,
  wide = TRUE,
  expand.date = FALSE,
  drop.1iso3c = TRUE,
  ordered = TRUE,
  return.query = FALSE,
  ...
)
```

Arguments

`ctry` character. (Optional) the ISO3 code of African countries (see `am_countries`).

`series` character. (Optional) codes of series matching the 'Series' column of the series table (retrieved using `am_series()`).

`from` set the start time of the data retrieved by either supplying a start date, a date-string of the form `"YYYY-MM-DD"` or `"YYYY-MM"`, year-quarters of the form `"YYYYQN"` or `"YYYY-QN"`, a numeric year `YYYY` (numeric or character), or a fiscal year of the form `"YYYY/YY"`. These expressions are converted to a regular date by `am_as_date`.

`to` same as `from`: to set the time period until which data is retrieved. For expressions that are not full "YYYY-MM-DD" dates, the last day of the period is chosen.

`labels` logical. `TRUE` will also return labels (series descriptions) along with the series codes.

`wide` logical. `TRUE` calls `am_pivot_wider` on the result. `FALSE` returns the data in a long format without missing values (suitable for `ggplot2`).

`expand.date` logical. `TRUE` will call `am_expand_date` on the result.

`drop.1iso3c` logical. If only one country is selected through `ctry`, `TRUE` will drop the 'ISO3' column in the output.

`ordered` logical. `TRUE` orders the result by 'Date' and, if `labels = TRUE`, by series, maintaining a fixed column-order of series. `FALSE` returns the result in a random order, to the benefit of faster query execution.

`return.query` logical. `TRUE` will not query the database but instead return the constructed SQL query as a character string (for debugging purposes).

`...` further arguments passed to `am_pivot_wider` (if `wide = TRUE`) or `am_expand_date` (if `expand.date = TRUE`), no conflicts between these two.

Details

If `labels = FALSE`, the series table is not joined to the data table, and `ordered = TRUE` will order series retrieved in alphabetic order. If `labels = TRUE` data is ordered by series and date, preserving the order of columns in the dataset. If multiple countries are received they are ordered alphabetically according to the 'ISO3' column.

Series at different frequencies can be queried, but, if `wide = TRUE`, this will result in missing values for all but the first observations per period in the lower frequency series.

Value

A `data.table` with the result of the query.

See Also

`am_pivot_wider`, `am_expand_date`, `africamonitor`

Examples

```
# Return all indicators for Kenya from 2000
am_data("KEN", from = 2000)

# Return all indicators for Kenya from 2000 in long format
am_data("KEN", from = 2000, wide = FALSE)

# Return with date expanded
am_data("KEN", from = 2000, expand.date = TRUE)

# Same thing in multiple steps (with additional customization options):
am_data("KEN", from = 2000, wide = FALSE) |> am_pivot_wider() |> am_expand_date()

# Getting only GDP growth
am_data("KEN", "NGDP_RPCH", from = 2000)

# Getting GDP growth for all countries
am_data(series = "NGDP_RPCH", from = 2000)

# Reshaping to wider format
am_data(series = "NGDP_RPCH", from = 2000) |>
  am_pivot_wider(id_cols = "Date",
                 names_from = "ISO3",
                 values_from = "NGDP_RPCH")

# Getting growth and inflation for the EAC countries (all available years)
am_data(ctry = c("UGA", "KEN", "TZA", "RWA", "BDI", "SSD"),
        series = c("NGDP_RPCH", "PCPIPCH"))
```

am_entities *Dataset of African Economic and Regional Entities*

Description

A dataset mapping African countries to various economic and regional entities.

Usage

```
am_entities
```

Format

A data frame with 54 rows (one for each country, excluding Western Sahara) and 27 variables, of which 6 are country identifiers and the remaining 21 are logical variables indicating country membership to various economic and regional entities.

Source

Own compilation.

See Also

am_countries, am_sources, am_series, africamonitor

Examples

```
head(am_entities)
```

---

am_expand_date *Generate Temporal Identifiers from a Date Column*

Description

This function expands a date column and generates additional temporal identifiers from it (such as the year, month, quarter, fiscal year etc.).

Usage

```
am_expand_date(
  x,
  gen = c("Year", "Quarter", "Month"),
  origin = "1899-12-30",
  keep.date = TRUE,
  remove.missing.date = TRUE,
  sort = TRUE,
  as.factor = TRUE,
  ...
)
```

Arguments

x	either a vector of class 'Date', or coercible to date using as.Date, or a data frame / list containing with variable named 'Date'.
gen	character. A vector of identifiers to generate from x. The possible identifiers are found in .AMT.
origin	character / Date. Passed to as.Date: for converting numeric x to date.
keep.date	logical. TRUE will keep the date variable in the resulting dataset, FALSE will remove the date variable in favor of the generated identifiers.
remove.missing.date	logical. TRUE will remove missing values in x. If x is a dataset, rows missing the 'Date' variable will be removed.
sort	logical. TRUE will sort the data by the 'Date' column.

`as.factor` TRUE will generate quarters, fiscal years and months ('Quarter', 'FY', 'QFY', 'Month') as factor variables. It is also possible to use `as.factor = "ordered"` to generate ordered factors. FALSE will generate fiscal years as character and quarters and months as integer variables.

`...` not used.

Value

A `data.table` containing the computed identifiers as columns. See Examples.

See Also

`am_as_date`, `africamonitor`

Examples

```
# First a basic example
x <- seq.Date(as.Date("1999-01-01"), as.Date("2000-01-01"), by = "month")
am_expand_date(x)
am_expand_date(x, gen = .AMT[-1L], keep.date = FALSE)

# Now using the API
am_expand_date(am_data("KEN"))

# Same thing
am_data("KEN", expand.date = TRUE)
```

`am_pivot_longer` *Reshape Column-Based Data to Long Format*

Description

This function automatically reshapes wide (column-based) data into a long format akin to the format of the raw data coming from the database (`am_data(..., wide = FALSE)`). It can also be used as a general purpose reshaping command - with an additional capability to handle variable labels.

Usage

```
am_pivot_longer(
  data,
  id_cols = intersect(c("ISO3", .AMT), names(data)),
  to_value = setdiff(names(data), id_cols),
  variable_name = "Series",
  value_name = "Value",
  label_name = "Label",
  na.rm = TRUE,
  ...
)
```

Arguments

`data` a wide format data frame where all series have their own column.

`id_cols` character. Temporal identifiers of the data. By default all variables in `.AMT` and "ISO3" are selected.

`to_value` character. The names of all series to be stacked into the long format data frame.

`variable_name` character. The name of the variable to store the names of the series.

`value_name` character. The name of the variable to store the data values.

`label_name` character. The name of the variable to store the series labels.

`na.rm` logical. `TRUE` will remove all missing values from the long data frame.

`...` further arguments passed to `melt`.

Value

A `data.table` with the reshaped data.

See Also

`am_pivot_wider`, `africamonitor`

Examples

```
# Return all indicators for Kenya and Nigeria from the year 2000 onwards
data <- am_data(c("KEN", "NGA"), from = 2000)
am_pivot_longer(data)
```

---

am_pivot_wider *Reshape Long API Data to Column-Based Format*

Description

This function automatically reshapes long (stacked) raw data from the API (`am_data(..., wide = FALSE)`) to a wide format where each variable has its own column. It can also be used as a general purpose reshaping command - with an additional capability to handle variable labels.

Usage

```
am_pivot_wider(
  data,
  id_cols = intersect(c("ISO3", .AMT), names(data)),
  names_from = "Series",
  values_from = "Value",
  labels_from = if (any(names(data) == "Label")) "Label" else NULL,
  expand.date = FALSE,
  ...
)
```

Arguments

`data`	raw data from the API: A long format data frame where all values are stacked in a value column.
`id_cols`	character. Temporal identifiers of the data. By default all variables in `.AMT` and "ISO3" are selected.
`names_from`	character. The column containing the series codes. These will become the names of the new columns in the wider data format.
`values_from`	character. The column containing the data values.
`labels_from`	character. The column containing the labels describing the series.
`expand.date`	logical. `TRUE` will call `am_expand_date` on the data after reshaping.
`...`	further arguments passed to `dcast` or `am_expand_date`, no conflicts between these two.

Value

A `data.table` with the reshaped data.

See Also

`am_pivot_longer`, `africamonitor`

Examples

```
# Return all indicators for Kenya and Nigeria from the year 2000 onwards
am_pivot_wider(am_data(c("KEN", "NGA"), from = 2000, wide = FALSE))
```

`am_series` *Retrieve Series Table*

Description

This function pulls information about the data series available in the database.

Usage

```
am_series(
  dsid = NULL,
  source.info = TRUE,
  ordered = TRUE,
  return.query = FALSE
)
```

Arguments

dsid	character. (Optional) id's of datasources matching the 'DSID' column of the data sources table (retrieved using am_sources()) for which series information is to be returned.
source.info	logical. TRUE returns additional information from the data sources table (the source, the frequency of the data and when it was last updated).
ordered	logical. TRUE returns the series in a fixed order, while FALSE returns the result in a random order, to the benefit of faster query execution.
return.query	logical. TRUE will not query the database but instead return the constructed SQL query as a character string (for debugging purposes).

Details

The series table gives information about all of the time series in the database. Each series is given a unique code, and has a label describing the series. Further information recorded are the minimum and maximum time coverage, and (optionally) a separate series source and url. The default source.info = TRUE adds the source, the frequency of the data (homogeneous within source), and the date when the source was last updated.

Value

A data.table with information about the available series in the database.

See Also

am_countries, am_sources, am_data, africamonitor

Examples

```
# By default returns all series with additional information
am_series()

# Raw series table
am_series(source.info = FALSE)

# Only series in the WEO
am_series("IMF_WEO")
```

am_sources *Retrieve Data Sources Table*

Description

This function retrieves a table with information about the sources of data in the database, and when data from different sources was updated.

Usage

```
am_sources(ordered = TRUE)
```

Arguments

`ordered`	logical. `TRUE` orders the result in the order data was entered into the database, while `FALSE` returns the result in a random order, to the benefit of faster query execution.

Details

The data source table gives information about the various sources / providers of data in this database, including the source website, frequency and time coverage of data, a description of the source, when data from the source was updated and the way data is accessed from the source.

Value

A `data.table` with information about the sources of data in the database.

See Also

`am_countries`, `am_series`, `africamonitor`

Examples

```
am_sources()
```

Index