Package ‘BETS’
September 28, 2018

Type Package
Title Brazilian Economic Time Series
Version 0.4.9
Date 2018-08-25
Maintainer Talitha Speranza <talitha.speranza@fgv.br>
Depends R (>= 3.4)
Imports grnn, ggplot2, plotly, urca, forecast, zoo, rmarkdown,
foreign, seasonal, stringr, dygraphs, shiny (>= 0.13), miniUI
(>= 0.1.1), rstudioapi (>= 0.4), DT, webshot, RMySQL,
digest,DBI, rjson, rvest, xml2, lubridate, htmltools, httr,
dplyr, sqldf
Suggests mFilter, devtools, xts, knitr
Description It provides access to and information about the most important
Brazilian economic time series - from the Getulio Vargas Foundation <http://portal.fgv.br/en>,
the Central Bank of Brazil <http://www.bcb.gov.br> and the Brazilian Institute of Geography
and Statistics <http://www.ibge.gov.br>. It also presents tools for managing, analysing (e.g.
generating dynamic reports with a complete analysis of a series) and exporting
these time series.
License GPL-3
BugReports https://github.com/nmecsys/BETS/issues
URL https://github.com/nmecsys/BETS
Encoding UTF-8
LazyData true
RoxygenNote 6.0.1
NeedsCompilation no
Repository CRAN
VignetteBuilder knitr
Author Pedro Costa Ferreira [aut],
Talitha Speranza [aut, cre],
Jonatha Costa [aut].
Fernando Teixeira [ctb],
Daiane Marcolino [ctb]

Date/Publication  2018-09-28 17:10:03 UTC

R topics documented:

arch_test ..................................................  3
bcbExpectA ..................................................  4
bcbExpectATop5 ..............................................  5
bcbExpectInf12 ..............................................  6
bcbExpectM ..................................................  7
bcbExpectMTop5 ..............................................  8
bcbExpectT ..................................................  9
BETS ......................................................... 10
BETS.addin_en .............................................. 10
BETS.addin_pt .............................................. 11
BETSget .................................................... 11
BETSsearch .................................................. 12
BETSsources ............................................... 15
chart ........................................................ 15
chart.add_basic ............................................ 17
chart.add_extra ............................................ 18
chart.add_notes ........................................... 19
check.series .............................................. 20
connection ................................................. 20
corrgram ................................................... 20
dashboard ................................................... 21
deflate ...................................................... 22
draw.cap_utl .................................................. 23
draw.cdb .................................................... 24
draw.cei ..................................................... 24
draw.conf_lvl ............................................... 25
draw.eap .................................................... 25
draw.ei_comps ............................................. 26
draw.ei_vars ............................................... 26
draw.gdp_comps ............................................. 27
draw.gdp_mon ............................................... 27
draw.gdp_unemp .......................................... 28
draw.gdp_vars ............................................. 28
draw.generic ............................................... 29
draw.iie_br ................................................. 29
draw.indprod .............................................. 30
draw.ipca ................................................. 30
draw.lei .................................................... 30
draw.misery_index ....................................... 31
draw.lab_coin .............................................. 31
draw.lab_lead .............................................. 31
draw.lab_lead .............................................. 32
**arch_test**

Perform an ARCH test

**Description**

Performs an ARCH test and show the results. Formerly, this function was part of FinTS, now an obsoleted package.

**Usage**

```r
arch_test(x, lags = 12, demean = FALSE, alpha = 0.5)
```

**Arguments**

- **x**: A ts object. The time series
- **lags**: An integer. Maximum number of lags
- **demean**: A boolean. Should the series be demeaned?
- **alpha**: A numeric value. Significance level
Value

A list with the results of the ARCH test

Author(s)

Spencer Graves <spencer.graves@prodsyse.com>, Talitha Speranza <talitha.speranza@fgv.br>

---

**bcbExpectA**

<table>
<thead>
<tr>
<th>bcbExpectA</th>
<th>bcbExpectA</th>
</tr>
</thead>
</table>

**Description**

Market Expectations with annual reference.

**Usage**

```r
bcbExpectA(indicator = "IPCA", limit = 100, variables = c("Media", "Mediana", "DesvioPadrao", "CoeficienteVariacao", "Minimo", "Maximo", "numeroRespondentes", "baseCalculo"), start, end)
```

**Arguments**

- **indicator**: A string. Available indicator.
- **limit**: A integer. A limit of data in request, top is 10000.
- **start**: Initial date at which the data was projected, in ISO format.
- **end**: Final date at which the data was projected, in ISO format.

**Value**

A data.frame.

**Note**

The available indicators are: Balanca comercial, Balanco de pagamentos, Fiscal, IGP-DI, IGP-M, INPC, IPA-DI, IPA-M, IPCA, IPCA-15, IPC-FIPE, Precos administrados por contrato e monitorado, Producao industrial, PIB Industrial, PIB Servicos, PIB Total, Meta para taxa over-selic e Taxa de cambio.

In collaboration with Angelo Salton <https://github.com/angelosalton>.

**Examples**

```r
# bcbExpectA()
```
bcbExpectATop5

Description

Annual Market Expectations Top5.

Usage

bcbExpectATop5(indicator = "IGP-DI", limit = 100,
variables = c("tipoCalculo", "Media", "Mediana", "DesvioPadrao",
"CoeficienteVariacao", "Minimo", "Maximo"), start, end)

Arguments

indicator      A string. Available indicator.
limit         A integer. A limit of data in request, top is 10000.
variables    Possible options: "Media", "Mediana", "DesvioPadrao", "CoeficienteVariacao",
              "Minimo", "Maximo".
start          Initial date at which the data was projected, in ISO format.
end            Final date at which the data was projected, in ISO format.

Value

A data.frame.

Note

The available indicators are: IGP-DI, IGP-M, IPCA, Meta para taxa over-selic, Taxa de cambio.

Examples

# bcbExpectATop5()
Description

Market expectations for inflation in the next 12 months

Usage

bcbExpectInf12(indicator = "IPCA", limit = 100, variables = c("Media", "Mediana", "DesvioPadrao", "CoeficienteVariacao", "Minimo", "Maximo", "numeroRespondentes", "baseCalculo"), start, end)

Arguments

- limit: A integer. A limint of data in request, top is 10000.
- start: Initial date at which the data was projected, in ISO format.
- end: Final date at which the data was projected, in ISO format.

Value

A data.frame.

Note

The available indicators are: IGP-DI, IGP-M, INPC, IPA-DI, IPA-M, IPCA, IPCA-15, IPC-FIPE.

Examples

# bcbExpectInf12()
bcbExpectM

Description

Market Expectations with mensal reference.

Usage


Arguments

indicator A string. Available indicator.
limit A integer. A limint of data in request, top is 10000.
start Initial date at which the data was projected, in ISO format.
end Final date at which the data was projected, in ISO format.

Value

A data.frame.

Note


Examples

# bcbExpectM()
**bcbExpectMTop5**

**Description**

Monthly Market Expectations Top5.

**Usage**

```r
bcbExpectMTop5(indicator = "IGP-DI", limit = 100,
variables = c("tipoCalculo", "Media", "Mediana", "DesvioPadrao",
"CoeficienteVariacao", "Minimo", "Maximo"), start, end)
```

**Arguments**

- `indicator` A string. Available indicator.
- `limit` A integer. A limint of data in request, top is 10000.
- `start` Initial date at which the data was projected, in ISO format.
- `end` Final date at which the data was projected, in ISO format.

**Value**

A data.frame.

**Note**

The available indicators are: IGP-DI, IGP-M, IPCA, Meta para taxa over-selic, Taxa de cambio.

**Examples**

```r
# bcbExpectMTop5()
```
**Description**

Quarterly Market Expectations.

**Usage**

```r
bcbExpectT(indicator = "PIB Total", limit = 100, variables = c("Media", "Mediana", "DesvioPadrao", "CoeficienteVariacao", "Minimo", "Maximo", "numeroRespondentes"), start, end)
```

**Arguments**

- `indicator` A string. Available indicator.
- `limit` A integer. A limint of data in request, top is 10000.
- `start` Initial date at which the data was projected, in ISO format.
- `end` Final date at which the data was projected, in ISO format.

**Value**

A data.frame.

**Note**

The available indicators are: PIB Agropecuario, PIB Industrial, PIB Serviços e PIB Total.

**Examples**

```r
# bcbExpectT()
```
Description

The Brazilian Economic Time Series (BETS) package provides access and information about the most important Brazilian economic time series.

These series are created by three influential centers: the Central Bank of Brazil (BCB), the Brazilian Institute of Geography and Statistics (IBGE) and the Brazilian Institute of Economics, from the Getulio Vargas Foundation (FVG-IBRE). Currently, there are more than 18,640 available time series, most of them free of charge. Besides providing access to this vast database, the package allows the user to interact with data in an easy and friendly way.

For instance, the user can search for a time series using keywords. More importantly, it installs several consecrated packages for time series analysis, giving the user the option to perform a complete analysis without having to worry about installing and loading other packages. In a near future, the authors will publish a series of R exercises to be solved with BETS and its statistical/econometrical tools, therefore helping the user to understand the behavior of brazilian time series.

Note

The authors would like to thank the support by the Getulio Vargas Foundation (FGV) and make it clear that all data in the package is in public domain. The rights of all centers from which the series are taken are maintained. We reaffirm that BETS is mainly intended for academic usage.

Author(s)

Pedro Costa Ferreira <pedro.guilherme@fgv.br>, Jonatha Costa <jonatha.costa@fgv.br>, Talitha Speranza <talitha.speranza@fgv.br>, Fernando Teixeira <fernando.teixeira@fgv.br>

Description

An interface for searching time series with possibility to extract the data in different extensions.

Usage

BETS.addin_en()
**Description**

An interface for searching time series with possibility to extract the data in different extensions.

**Usage**

```r
BETS.addin_pt()
```

---

**BETSget**  
*Get a complete time series from a BETS database*

**Description**

Extracts a complete time series from either the Central Bank of Brazil (BCB), the Brazilian Institute of Geography and Statistics (IBGE) or the Brazilian Institute of Economics (FGV/IBRE).

**Usage**

```r
BETSget(code, from = "", to = "", data.frame = FALSE, frequency = NULL)
```

**Arguments**

- **code**: A character or an integer. The unique code that references the time series. This code can be obtained by using the `search` function. More than one code can be provided at once, through a vector. In this case, be careful with the dates, i.e, parameters `from` and `to`. They must either be the same length as code, containing the date limits in order, or an isolated date, but nothing in between. See the examples section.

- **from**: A character or a `data` object. Starting date of the time series (format YYYY-MM-DD). Can be a vector of dates/characters if the length of the parameter `code` is greater than 1.

- **to**: A character or a `data` object. Ending date of the time series (format YYYY-MM-DD). Can be a vector of dates/characters if the length of the parameter `code` is greater than 1.

- **data.frame**: A boolean. True if you want the output to be a data frame. True to `ts` output.

- **frequency**: An integer. The frequency of the time series. It is not needed. It is going to be used only if the metadata for the series is corrupted.

**Value**

A `ts` (time series) object containing the desired series.
Note

Due to the significant size of the databases, it could take a while to retrieve the values. However, it shouldn’t take more than 90 seconds.

See Also

ts, BETSsearch and seas

Examples

# Annual series: GDP at constant prices, in R$ (brazilian reais)
#BETSget(1208)

# International reserves - Cash concept
#int.reserves <- get("3543")
#plot(int.reserves)

# Exchange rate - Free - United States dollar (purchase)
#us.brl <- get(3691)

# Multiple requests
# BETSget(code = c(10777,4447),from = "2001-01-01", to = "2016-10-31")
# BETSget(code = c(10777,4447),from = c("2001-10-31",""),to = c("2016-10-31",""))

# f <- c("2001-10-31","1998-09-01")
# t <- c("2014-10-31","2015-01-01")
# BETSget(code = c(10777,4447), from = f, to = t)

# BETSget(code = c(10777,4447),from = "2001-10-31", to = c("2014-10-31","2015-01-01"))
# BETSget(code = c(10777,4447),from = c("2002-10-31","1997-01-01"), to = "2015-01-01")

BETSsearch

Search for a Brazilian Economic Time Series

Description

Searches the BETS databases for a time series by its description, source, periodicity, code, data, unit of measurement and database name.

Usage

BETSsearch(description = "+", src, periodicity, unit, code, start, view = FALSE, lang = "en")
Arguments

- **description**: A character. A search string to look for matching series descriptions. Check the syntax rules under the 'Details' section for better performance.
- **src**: A character. The source of the series. See the 'Details' section for a list of the available sources.
- **periodicity**: A character. The periodicity of the series. See the 'Details' section for a list of possible values.
- **unit**: A character. The unit of measurement of the data. See the 'Details' section for a list of possible values.
- **code**: An integer. The index of the series within the database.
- **start**: A date. Starting date of the series.
- **view**: A boolean. The default is TRUE. If set to FALSE, the output's head will be printed in your console as a data.frame.
- **lang**: A character. The search language. The default is "en" for english, but "pt" for portuguese is also possible.

Details

- Syntax rules for the parameter description, the search string to look for matching series descriptions:
  1. To search for alternative words, separate them by white spaces. Example: `description = "ipca core"` means that the series description must contain 'ipca' AND 'core'
  2. To search for whole expressions, surround them with ' '. Example: `description = 'core ipca index'` means that the series description must contain 'core ipca' AND 'index'
  3. To exclude words from the search, insert a ~ before each of them. Example: `description = "ipca ~ core"` means that the series description must contain 'ipca' AND must NOT contain 'core'
  4. To exclude whole expressions from the search, surround them with code ' ' and insert a ~ before each of them. Example: `description = "~ 'ipca core' index"` means that the series description must contain 'index' AND must NOT contain 'core ipca'
  5. It is possible to search for multiple words or expressions and to negate multiple words or expressions, as long as the preceeding rules are observed.
  6. The white space after the negation sign (~) is not required. But the white spaces AFTER expressions or words ARE required.

- Possible values for the parameter src:
  - **IBGE**: Brazilian Institute of Geography and Statistics
  - **BCB**: Central Bank of Brazil
  - **FGV**: Getulio Vargas Foundation
  - **FGV-IBRE**: Getulio Vargas Foundation - Brazilian Institute of Economics
  - **BCB e FGV**: Central Bank of Brazil and Getulio Vargas Foundation
  - **BCB-Deban**: Central Bank of Brazil - Department of Banking and Payments
  - **BCB-Depin**: Central Bank of Brazil - Department of International Reserves
  - **BCB-Derin**: Central Bank of Brazil - Department of International Affairs
  - **BCB-Desig**: Central Bank of Brazil - Department of Financial Monitoring
  - **BCB-Secre**: Central Bank of Brazil - Executive Secretariat
• Possible values for the parameter *periodicity*:

A anual data  
M monthly data  
Q quaterly data  
W weekly data  
D daily data

• Possible values for the parameter *unit*:

R$ brazilian reais  
$ US dollars  
% percentage

**Value**

A list that can be interpreted as a *data.frame*. The fields are described below.

<table>
<thead>
<tr>
<th>code</th>
<th>The code/index of the series within the database</th>
</tr>
</thead>
<tbody>
<tr>
<td>description</td>
<td>The description of the series</td>
</tr>
<tr>
<td>periodicity</td>
<td>The periodicity of the series</td>
</tr>
<tr>
<td>start</td>
<td>Starting date of the series</td>
</tr>
<tr>
<td>source</td>
<td>The source of the series</td>
</tr>
<tr>
<td>unit</td>
<td>The unit of measurement of the data</td>
</tr>
</tbody>
</table>

**References**

Central Bank of Brazil

**Examples**

```r
#not run
#BETSsearch(description="sales",view = FALSE)

#BETSsearch(src="Denor", view = FALSE)

#BETSsearch(periodicity="A", view = FALSE)
```
**BETSSources**

*Display a list of sources available at BETS package*

---

**Description**

Display a list of sources available at BETS package in console. The numbers of sources will increase with new versions of the package.

**Usage**

```r
BETSSources()
```

---

**chart**

*Create a chart with BETS aesthetics*

---

**Description**

Create a professional looking chart, using a pre-defined BETS series or a custom series.

**Usage**

```r
chart(ts, style = "normal", file = NULL, open = TRUE, lang = "en",
       params = NULL)
```

**Arguments**

- **ts**: A character or a `ts` object. A custom time series or the name of a pre-defined series. A complete list of names is under the 'Details' section.
- **style**: A character. Should the chart be made with Plotly (style = "plotly") or with R standard library (style = "normal")?
- **file**: A character. The whole path, including a custom name, for the output (an image file). The default value is NULL. If left to NULL, the chart will be rendered in the standard R plotting area.
- **open**: A boolean. Whether to open the file containing the chart.
- **lang**: A character. The language. For now, only 'en' (english) is available.
- **params**: A list. Parameters for drawing custom charts. See the 'details' section.
Details

Names of pre-defined charts:

1. Business Cycle Dashboard ('plotly' style)

<table>
<thead>
<tr>
<th>VALUE</th>
<th>DESCRIPTION</th>
<th>CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>'iie_br'</td>
<td>Uncertainty Index</td>
<td>ST_100.0</td>
</tr>
<tr>
<td>'sent_ind'</td>
<td>Economic Sentiment Index (average between several confidence indexes)</td>
<td>(*)</td>
</tr>
<tr>
<td>'gdp_mon'</td>
<td>GDP Monthly and Interannual Variation (last values) - GDP Monitor (FGV/IBRE)</td>
<td>(*)</td>
</tr>
<tr>
<td>'ei_vars'</td>
<td>Economic Indicators (Leading and Coincident) monthly variation</td>
<td>(*)</td>
</tr>
<tr>
<td>'ei_comps'</td>
<td>Economic Indicators (Leading and Coincident) components variation</td>
<td>(*)</td>
</tr>
<tr>
<td>'lei'</td>
<td>Leading Economic Indicator (LEI - FGV/IBRE with The Conference Board)</td>
<td>(*)</td>
</tr>
<tr>
<td>'cei'</td>
<td>Coincident Economic Indicator (CEI - FGV/IBRE with the Conference Board)</td>
<td>(*)</td>
</tr>
<tr>
<td>'gdp_vars'</td>
<td>GDP components variation (whole series) - GDP Monitor (FGV/IBRE)</td>
<td>(*)</td>
</tr>
<tr>
<td>misery_index</td>
<td>Misery Index</td>
<td>13522 plus 24369</td>
</tr>
<tr>
<td>'gdp_comps'</td>
<td>GDP components variation (last values) - GDP Monitor (FGV/IBRE)</td>
<td>(*)</td>
</tr>
<tr>
<td>'gdp_unemp'</td>
<td>GDP monthly levels versus Unemployment Rate</td>
<td>22109 and 24369</td>
</tr>
<tr>
<td>'conf_lvl'</td>
<td>Enterprises Confidence Index versus Consumers Confidence Index</td>
<td>(*)</td>
</tr>
<tr>
<td>'inst_cap'</td>
<td>Installed Capacity Index</td>
<td>(*)</td>
</tr>
<tr>
<td>'lab_lead'</td>
<td>Labor Leading Indicator</td>
<td>(*)</td>
</tr>
<tr>
<td>'lab_coin'</td>
<td>Labor Coincident Indicator</td>
<td>(*)</td>
</tr>
<tr>
<td>'transf_ind'</td>
<td>Transformation Industry Confidence Index (Expectations versus Present Situation)</td>
<td>(*)</td>
</tr>
<tr>
<td>'serve'</td>
<td>Services Confidence Index (Expectations versus Present Situation)</td>
<td>(*)</td>
</tr>
<tr>
<td>'constr'</td>
<td>Construction Confidence Index (Expectations versus Present Situation)</td>
<td>(*)</td>
</tr>
<tr>
<td>'retail'</td>
<td>Retail Sellers Confidence Index (Expectations versus Present Situation)</td>
<td>(*)</td>
</tr>
<tr>
<td>'consum'</td>
<td>Consumer Confidence Index (Expectations versus Present Situation)</td>
<td>(*)</td>
</tr>
</tbody>
</table>

2. Macro Situation Dashboard ('normal' style)

<table>
<thead>
<tr>
<th>VALUE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>'ipca_with_core'</td>
<td>National consumer price index (IPCA) - in 12 months and Broad national consumer price index - Core IPCA</td>
</tr>
<tr>
<td>'ulc'</td>
<td>Unit labor cost - ULC-US$ - June/1994=100</td>
</tr>
<tr>
<td>'eap'</td>
<td>Economically active population</td>
</tr>
<tr>
<td>'cdb'</td>
<td>Time deposits (CDB/RDB-preset) - Daily return (percentage)</td>
</tr>
<tr>
<td>'indprod'</td>
<td>Production Indicators (2012=100) - General</td>
</tr>
<tr>
<td>'selic'</td>
<td>Interest rate - Selic accumulated in the month in annual terms (basis 252)</td>
</tr>
<tr>
<td>'unemp'</td>
<td>Unemployment rate - by metropolitan region (PNAD-C)</td>
</tr>
<tr>
<td>'vargdp'</td>
<td>GDP - real percentage change in the year</td>
</tr>
</tbody>
</table>

(*) Not available on BETS databases yet. But you can find it in .csv files saved under your BETS installation directory.

3. Custom Charts

None of these parameters is required. Please note that some parameters only work for a certain type of chart.

<p>| PARAMETER | DESCRIPTION                           |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>A character. Either 'bar' or 'lines'. Whether to plot bars or lines. Works for main series, only.</td>
</tr>
<tr>
<td>trend</td>
<td>A boolean. Default is FALSE. Set it to TRUE if the trend of the main series (parameter ts) is to be drawn.</td>
</tr>
<tr>
<td>title</td>
<td>A character. Plot's title.</td>
</tr>
<tr>
<td>subtitle</td>
<td>A character. Plot's subtitle.</td>
</tr>
<tr>
<td>xlim</td>
<td>A numeric vector. X axis limits</td>
</tr>
<tr>
<td>ylim</td>
<td>A numeric vector. Y axis limits</td>
</tr>
<tr>
<td>arr.ort</td>
<td>A character. Orientation of the arrow pointing to the last value of the main series. Valid values are 'h' (horizontal) and 'v' (vertical).</td>
</tr>
<tr>
<td>arr.len</td>
<td>A numeric value. Length of the arrow pointing to the last value of the main series.</td>
</tr>
<tr>
<td>extra</td>
<td>A ts object. A second series to be plotted.</td>
</tr>
<tr>
<td>extra.y2</td>
<td>A boolean. Default is FALSE. Does the extra series require a second y axis?</td>
</tr>
<tr>
<td>extra.arr.ort</td>
<td>A character. Orientation of the arrow pointing to the last value of the extra series. Valid values are 'h' (horizontal) and 'v' (vertical).</td>
</tr>
<tr>
<td>extra.arr.len</td>
<td>A numeric value. Length of the arrow pointing to the last value of the extra series.</td>
</tr>
<tr>
<td>colors</td>
<td>A character or integer vector. A vector of colors, one for each series. Trends will always be drawn in gray.</td>
</tr>
<tr>
<td>legend</td>
<td>A character vector. Names of the series. Default is NULL (no legends).</td>
</tr>
<tr>
<td>legend.pos</td>
<td>A character. Legend position. If type is set to 'normal', possible values are 'top' and 'bottom'; if type is 'plotly', either 'h' (horizontal) and 'v' (vertical).</td>
</tr>
<tr>
<td>codace</td>
<td>A boolean. Default is FALSE. Include shaded areas for recessions, as dated by CODACE(**)?</td>
</tr>
</tbody>
</table>

(**) Business Cycle Dating Committee (FGV/IBRE)

Value

If parameter file is not set by the user, the chart will be shown at the standard R plotting area. Otherwise, it is going to be saved on your computer.

Author(s)

Talitha Speranza <talitha.speranza@fgv.br>

Examples

```r
# chart(ts = "sent_ind", file = "animal_spirits", open = T)
# chart(ts = "gdp_mon", file = "gdp_mon.png", open = F)
# chart(ts = "misery_index")
# chart(ts = "transf_ind", file = "transf_ind.png", open = F)
```

Description

Creates a plot of series 11777
Usage

chart.add_basic(ts, xlim = NULL, ylim = NULL, type = "lines",
    title = "", subtitle = "", col = "firebrick4", arr.size = NULL,
    arr.pos = "v", leg.pos = "top", trend = FALSE)

Arguments

ts A ts. the ts object.
xlim A numeric vector. x axis limits.
ylim A numeric vector. Y axis limits.
type A character. The type of plot (lines).
title A character. The plot title.
subtitle A character. The plot subtitle.
col A character. Color.
arr.size A vector.
arr.pos A vector.
leg.pos A vector.
trend A boolean.

Value

An image file is saved in the 'graphs' folder, under the BETS installation directory.

Author(s)

Talitha Speranza <talitha.speranza@fgv.br>

chart.add_extra Create a chart of the Unitary Labor Cost time series

Description

Creates a plot of series 11777

Usage

chart.add_extra(ts, ylim = NULL, xlim = NULL, col = "firebrick3",
    arr.size = NULL, arr.pos = "v", leg.pos = "top", leg.text = "",
    main.type = "lines")
Arguments

- **ts**: A `ts` object.
- **ylim**: A numeric vector. Y axis limits.
- **xlim**: A numeric vector. X axis limits.
- **col**: A character. Color.
- **arr.size**: A.
- **arr.pos**: A.
- **leg.pos**: A.
- **leg.text**: A.
- **main.type**: A.

Value

An image file is saved in the 'graphs' folder, under the BETS installation directory.

Author(s)

Talitha Speranza <talitha.speranza@fgv.br>

---

**chart.add_notes**

*Add notes*

Description

Add notes

Usage

```r
chart.add_notes(series.list, xlim, ylim, names = NULL, dec = 2)
```

Arguments

- **series.list**: A `ts` object
- **xlim**: A vector
- **ylim**: A vector
- **names**: A character
- **dec**: An integer

Author(s)

Talitha Speranza <talitha.speranza@fgv.br>
check.series  

Check series

Description

Check series in BETS dataset

Usage

check.series(ts, message = NULL)

Arguments

ts  A ts object
message  A character

Author(s)

Talitha Speranza <talitha.speranza@fgv.br>

connection  

Connection with the server

Description

Make the connection with the server

Usage

connection()

corrgram  

Plot the ACF or the PACF of a time series

Description

Plot correlograms using plot.ly and several other options that differ these plots from forecasts ACF and PACF.

Usage

corrgram(ts, lag.max = 12, type = "correlation", mode = "simple", ci = 0.95, style = "plotly", knit = f)
Arguments

ts
An object of type ts or xts. The time series for which the plot must be constructed.

lag.max
A numeric value. The number of lags to be shown in the plot.

type
A character. Can be either 'correlation' (for the ACF) or 'partial' (for the PACF).

mode
A character. Set this parameter to 'bartlett' if you want the variance to be calculated according to Bartlett's formula. Otherwise, it is going to be simply equal to $10\sqrt{n}$.

style
A numeric value. The confidence interval to be shown in the plot.

Value

A plot and a vector containing the correlations.

Author(s)

Talitha Speranza <talitha.speranza@fgv.br>
saveas  A character. A path and a name for the dashboard file (a .pdf file). If this parameter is not provided, the dashboard will be saved inside the 'dashboards' folder, under the BETS installation directory.

parameters  A list. A list of parameters. See the 'Details' section for a description of these parameters for each type of dashboard.

Details

Macro Situation and Custom Dashboard Parameters

text  The text to be printed in the dashboard. Separate paragraphs with two backslashes ‘n’ and pages with ‘##’. There are no other syntax rules.

author  The author’s name.

email  The author’s email.

url  The author’s webpage.

logo  The author’s business logo.

Additional Custom Dashboard Parameters

style  A character. The style of the charts. As in chart, can be either 'plotly' or 'normal'.

charts.opts  A list of parameters lists, one for each chart. Parameters are specified in chart

Value

A .pdf file (the dashboard)

Author(s)

Talitha Speranza <talitha.speranza@fgv.br>

Examples

# dashboard()
# dashboard(saveas = "survey.pdf")
# dashboard(type = "macro_situation")

deflate  Deflate a time series

Description

Deflate a time series using a deflator series. The deflator can be an index, a percentage or a point percentage series.
draw.cap_utl

Usage

deflate(ts, deflator, type = "index")

Arguments

ts A ts object. The time series to be deflated.
deflator A ts object. The deflator series.
type A character. Can be either 'index', 'point.perc' (for point percentage) or 'perc' (for percentage).

Value

The deflated series.

Author(s)

Talitha Speranza <talitha.speranza@fgv.br>

draw.cap_utl Create a chart of the Base Interest Rate (SELIC) time series

Description

Creates a plot of series 4189

Usage

draw.cap_utl()

Value

An image file is saved in the 'graphs' folder, under the BETS installation directory.

Author(s)

Talitha Speranza <talitha.speranza@fgv.br>
**draw.cdb**

Create a chart of the Time Deposits time series

**Description**

Creates a plot of series 14

**Usage**

draw.cdb()

**Value**

An image file is saved in the 'graphs' folder, under the BETS installation directory.

**Author(s)**

Talitha Speranza <talitha.speranza@fgv.br>

---

**draw.cei**

Create a chart of the Base Interest Rate (SELIC) time series

**Description**

Creates a plot of series 4189

**Usage**

draw.cei()

**Value**

An image file is saved in the 'graphs' folder, under the BETS installation directory.

**Author(s)**

Talitha Speranza <talitha.speranza@fgv.br>
**draw.conf_lvl**

**Create a chart of the Base Interest Rate (SELIC) time series**

**Description**

Creates a plot of series 4189

**Usage**

```r
draw.conf_lvl()
```

**Value**

An image file is saved in the 'graphs' folder, under the BETS installation directory.

**Author(s)**

Talitha Speranza <talitha.speranza@fgv.br>

---

**draw.eap**

**Create a chart of the Economically Active Population time series**

**Description**

Creates a plot of series 10810

**Usage**

```r
draw.eap()
```

**Value**

An image file is saved in the 'graphs' folder, under the BETS installation directory.

**Author(s)**

Talitha Speranza <talitha.speranza@fgv.br>
**draw.ei_comps**

Create a chart of the Base Interest Rate (SELIC) time series

**Description**

Creates a plot of series 4189

**Usage**

```r
draw.ei_comps()
```

**Value**

An image file is saved in the 'graphs' folder, under the BETS installation directory.

---

**draw.ei_vars**

Create a chart of the Base Interest Rate (SELIC) time series

**Description**

Creates a plot of series 4189

**Usage**

```r
draw.ei_vars()
```

**Value**

An image file is saved in the 'graphs' folder, under the BETS installation directory.

**Author(s)**

Talitha Speranza <talitha.speranza@fgv.br>
**draw.gdp_comps**  
*Create a chart of the Base Interest Rate (SELIC) time series*

**Description**  
Creates a plot of series 4189

**Usage**  
```r  
draw.gdp_comps()  
```

**Value**  
An image file is saved in the 'graphs' folder, under the BETS installation directory.

**Author(s)**  
Talitha Speranza <talis77ha.speranza@fgv.br>

---

**draw.gdp_mon**  
*Create a chart of the Base Interest Rate (SELIC) time series*

**Description**  
Creates a plot of series 4189

**Usage**  
```r  
draw.gdp_mon()  
```

**Value**  
An image file is saved in the 'graphs' folder, under the BETS installation directory.

**Author(s)**  
Talitha Speranza <talis77ha.speranza@fgv.br>
**draw.gdp_unemp**

Create a chart of the Base Interest Rate (SELIC) time series

**Description**

Creates a plot of series 4189

**Usage**

```
draw.gdp_unemp()
```

**Value**

An image file is saved in the `graphs` folder, under the BETS installation directory.

**Author(s)**

Talitha Speranza <talitha.speranza@fgv.br>

---

**draw.gdp_vars**

Create a chart of the Base Interest Rate (SELIC) time series

**Description**

Creates a plot of series 4189

**Usage**

```
draw.gdp_vars()
```

**Value**

An image file is saved in the `graphs` folder, under the BETS installation directory.

**Author(s)**

Talitha Speranza <talitha.speranza@fgv.br>
**draw.generic**

Create a chart of the Base Interest Rate (SELIC) time series

**Description**

Creates a plot of series 4189

**Usage**

draw.generic(ts, style, params)

**Arguments**

- **ts**: aaaa
- **style**: aaa
- **params**: aaa

**Value**

An image file is saved in the 'graphs' folder, under the BETS installation directory.

**Author(s)**

Talitha Speranza <talitha.speranza@fgv.br>

---

**draw.iie_br**

Create a chart of the Base Interest Rate (SELIC) time series

**Description**

Creates a plot of series 4189

**Usage**

draw.iie_br()

**Value**

An image file is saved in the 'graphs' folder, under the BETS installation directory.

**Author(s)**

Talitha Speranza <talitha.speranza@fgv.br>
**draw.indprod**

*Create a chart of the Production Indicators time series*

**Description**

Creates a plot of series 21859

**Usage**

```
draw.indprod()
```

**Value**

An image file is saved in the 'graphs' folder, under the BETS installation directory.

**Author(s)**

Talitha Speranza <talitha.speranza@fgv.br>

---

**draw.ipca**

*Create a chart of the National Consumer Price Index time series*

**Description**

Creates a plot of series 13522 (NCPI), along with series 4466 (NCPI core)

**Usage**

```
draw.ipca()
```

**Value**

An image file is saved in the 'graphs' folder, under the BETS installation directory.

**Author(s)**

Talitha Speranza <talitha.speranza@fgv.br>
**draw.lab_coin**

Create a chart of the Base Interest Rate (SELIC) time series

**Description**

Creates a plot of series 4189

**Usage**

draw.lab_coin()

**Value**

An image file is saved in the ‘graphs’ folder, under the BETS installation directory.

**Author(s)**

Talitha Speranza <talitha.speranza@fgv.br>

---

**draw.lab_lead**

Create a chart of the Base Interest Rate (SELIC) time series

**Description**

Creates a plot of series 4189

**Usage**

draw.lab_lead()

**Value**

An image file is saved in the ‘graphs’ folder, under the BETS installation directory.

**Author(s)**

Talitha Speranza <talitha.speranza@fgv.br>
Create a chart of the Base Interest Rate (SELIC) time series

**Description**

Creates a plot of series 4189

**Usage**

draw.lei()

**Value**

An image file is saved in the 'graphs' folder, under the BETS installation directory.

**Author(s)**

Talitha Speranza <talitha.speranza@fgv.br>

Create a chart of the Base Interest Rate (SELIC) time series

**Description**

Creates a plot of series 4189

**Usage**

draw.misery_index()

**Value**

An image file is saved in the 'graphs' folder, under the BETS installation directory.

**Author(s)**

Talitha Speranza <talitha.speranza@fgv.br>
**draw.selic**

Create a chart of the Base Interest Rate (SELIC) time series

**Description**

Creates a plot of series 4189

**Usage**

```
draw.selic()
```

**Value**

An image file is saved in the ‘graphs’ folder, under the BETS installation directory.

**Author(s)**

Talitha Speranza <talitha.speranza@fgv.br>

---

**draw.sent_ind**

Create a chart of the Base Interest Rate (SELIC) time series

**Description**

Creates a plot of series 4189

**Usage**

```
draw.sent_ind()
```

**Value**

An image file is saved in the ‘graphs’ folder, under the BETS installation directory.

**Author(s)**

Talitha Speranza <talitha.speranza@fgv.br>
**draw.survey**

*Create a chart of the Base Interest Rate (SELIC) time series*

**Description**

Creates a plot of series 4189

**Usage**

```r
draw.survey(survey)
```

**Arguments**

- `survey` xxx

**Value**

An image file is saved in the 'graphs' folder, under the BETS installation directory.

**Author(s)**

Talitha Speranza <talitha.speranza@fgv.br>

---

**draw.ulc**

*Create a chart of the Unitary Labor Cost time series*

**Description**

Creates a plot of series 11777

**Usage**

```r
draw.ulc()
```

**Value**

An image file is saved in the 'graphs' folder, under the BETS installation directory.
draw.unemp

Create a chart of the Open Unemployment Rate time series

Description

Creates a plot of series 10777

Usage

draw.unemp()

Value

An image file is saved in the ‘graphs’ folder, under the BETS installation directory.

draw.vargdp

Create a chart of the Real Percentage Change of GDP in the Year time series

Description

Creates a plot of series 7326

Usage

draw.vargdp()

Value

An image file is saved in the ‘graphs’ folder, under the BETS installation directory.

dummy

Create a monthly or quarterly dummy

Description

Returns a monthly or quarterly dummy (a time series with only 0s and 1s).

Usage

dummy(start = NULL, end = NULL, frequency = 12, year = NULL, month = NULL, quarter = NULL, date = NULL, from = NULL, to = NULL)
Arguments

**start**
An integer vector. The period of the first observation. The first element of the vector specifies the year of the first observation, whereas the second, the month (for monthly dummies) or quarter (for quarterly dummies).

**end**
An integer vector. The period of the last observation. The first element of the vector specifies the year of the last observation, whereas the second, the month (for monthly dummies) or quarter (for quarterly dummies).

**frequency**
An integer. The frequency of the dummy, that is, the number of observations per unit of time. The default is 12 (a monthly dummy).

**year**
An integer, a seq or a vector. The years for which the dummy must be set to 1. All periods of these years will be set to 1.

**month**
An integer, a seq or a vector. The months for which the dummy must be set to 1. These months will be set to 1 for all years.

**quarter**
An integer, a seq or a vector. The quarters for which the dummy must be set to 1. The quarters will be set to 1 for all years.

**date**
a list. The periods for which the dummy must be set to one. Periods must be represented as integer vectors, as described for `start` and `end`.

**from**
An integer vector. The starting period of a sequence of periods for which the dummy must be set to one. Periods must be represented as integer vectors, as described for `start` and `end`.

**to**
The ending period of a sequence of periods for which the dummy must be set to one. Periods must be represented as integer vectors, as described for `start` and `end`.

Value

A monthly or a quarterly ts object.

See Also

ts, dummy

Examples

```r
# from a specific date to another specific date
dummy(start = c(2000,1), end = c(2012,5), frequency = 12, from = c(2005,1), to = c(2006,12))
```

# Other options that may be helpful:

```r
# over a month equal to 1
dummy(start = c(2000,1), end = c(2012,5), frequency = 12, month = c(5,12))
```

```r
# Months equal to 1 only for some year
dummy(start = c(2000,1), end = c(2012,5), frequency = 12, month = 5, year = 2010)
dummy(start = c(2000,1), end = c(2012,5), frequency = 12, month = 8, year = 2002)
```
get.series

Get a complete time series from a BETS database

Description

Extracts a complete time series from either the Central Bank of Brazil (BCB), the Brazilian Institute of Geography and Statistics (IBGE) or the Brazilian Institute of Economics (FGV/IBRE).

Usage

get.series(code, from = "", to = "", data.frame = FALSE, frequency = NULL)

Arguments

code A character. The unique code that references the time series. This code can be obtained by using the BETSsearch function.

from A character or a Data object. Starting date of the time series (format YYYY-MM-DD).

to A character or a Data object. Ending date of the time series (format YYYY-MM-DD).

data.frame A boolean. True if you want the output to be a data frame. True to ts output.

frequency An integer. The frequency of the time series. It is not needed. It is going to be used only if the metadata for the series is corrupted.
get.series.bacen A function to extract BACEN series using their API

Description
A function to extract BACEN series using their API

Usage
get.series.bacen(x, from = "", to = "", save = "")

Arguments
x Bacen series numbers. Either an integer or a numeric vector.
from A string specifying where the series shall start.
to A string specifying where the series shall end.
save A string specifying if data should be saved in csv or xlsx format. Defaults to not saving.

Author(s)
Fernando Teixeira <fernando.teixeira@fgv.br> and Jonatha Azevedo <jonatha.costa@fgv.br>

grnn.test Test a set of General Regression Neural Networks

Description
Given new values of the independent variables, tests a list of trained GRNNs and picks the best net, based on an accuracy measure between the forecasted and the actual values.

Usage
grnn.test(results, test.set)

Arguments
results The object returned by grnn.train.
test.set A ts list. The first element must be the actual values of the dependent variable. The others, the new values of the regressors.
Value

A list object representing the best network (according to forecasting MAPE). Its fields are:

- mape: The forecasting MAPE
- model: The network object
- sigma: The sigma parameter
- id: The id number of the network, as given by grnn.train
- mean: The predicted values
- x: The original series
- fitted: The fitted values
- actual: The actual values (to be compared with the predicted values)
- residuals: Difference between the fitted values and the series original values
- regressors: The regressors used to train the network

Author(s)

Talitha Speranza <talitha.speranza@fgv.br>
select A boolean. Must be set to FALSE if the regressors should not be chosen. The default is TRUE.

names A character vector. Optional. The names of the regressors. If not provided, indexes will be used and reported.

Value

A list of result objects, each representing a network. These objects are ordered by MAPE (the 20 best MAPEs) and its fields are:

- accuracy: A numeric value. Accuracy measure between the fitted and the actual series values. By default, the MAPE. In future versions, it will be possible to change it.
- fitted: The fitted values, that is, one step ahead predictions calculated by the trained net.
- net: An object returned by the grnn function. Represents a trained net.
- sigma: A numeric. The sigma that was chosen, either by the user or by the function itself (in case select was set to TRUE)
- regressors: A character vector. Regressors that were chosen, either by the user or by the function itself (in case select was set to TRUE)
- sigma.accuracy: A data.frame. Sigma versus accuracy value of the corresponding trained network. Those networks were trained using the best set of regressors.
- residuals: A numeric vector. Fitted values subtracted from the actual values.

grnn.train also returns a diagnostic of training rounds and a sigma versus accuracy plot.

Author(s)

Talitha Speranza <talitha.speranza@fgv.br>

---

**msg**

Format and show a console message.

**Description**

Customizes a message and shows it in the console.

**Usage**

```r
msg(..., skip_before = TRUE, skip_after = FALSE, warn = FALSE)
```

**Arguments**

- ... Arguments to be passed to `message`
- skip_before A boolean. Indicates if a line should be skipped before the message.
- skip_after A boolean. Indicates if a line should be skipped after the message.
- warn A boolean. Indicates whether a warning should be thrown.
**normalize**

*Value*

None

*Author(s)*

Talitha Speranza <talitha.speranza@fgv.br>, Jonatha Azevedo <jonatha.azevedo@fgv.br>

---

**normalize**

*Normalize a time series*

**Description**

Normalizes a time series, either by standardization or by mapping to values between 0 and 1.

**Usage**

normalize(series, mode = "scale")

**Arguments**

- series: A ts object or a ts list. The series to be normalized.
- mode: A character. The normalization method. Set this parameter to 'maxmin' to map series values to values between 0 and 1. Alternatively, set this parameter to 'scale' to standardize (subtract the mean and divide by the standard deviation).

**Value**

A ts object or a ts list. The normalized series.

*Author(s)*

Talitha Speranza <talitha.speranza@fgv.br>

---

**predict**

*Get the predicted values of a model and visualize it*

**Description**

This function is built upon forecast. Besides the model predictions, it returns an accuracy measure table (calculated by the accuracy function) and a graph showing the original series, the predicted values and the actual values.

**Usage**

predict(..., actual = NULL, main = "", ylab = "", xlim = NULL, style = "dygraphs", unnorm = NULL, legend.pos = "topright", knit = F)
Arguments

Arguments passed on to forecast. If the model is a neural network, these arguments will be passed on to grnn.test.

actual
A numeric vector. The actual values (to be compared with predicted values).

main
A character. The name of the prediction plot.

ylab
A character. The Y axis label.

xlim
A numeric vector. The limits of the X axis.

style
A character. Can be either 'dygraphs' (the dygraph function will be used to make the plot, which is going to be HTML based) or 'normal' (standard R functions will be used to make the plot).

unnorm
A numeric vector. If predictions must be unnormalized, set the first element of this vector to the mean and the second, to the standard deviation.

legend.pos
A character. The position of the legend. Possible values are standard R plot values, i.e., 'topright', 'bottomleft', etc.

knit
A boolean. Set this parameter to TRUE if

Value

Besides the prediction plot, this function returns an object whose fields are:

- accuracy: An object returned by accuracy. It is a table containing several accuracy measures
- predictions: A numeric vector containing the predicted values.

Author(s)

Talitha Speranza <talitha.speranza@fgv.br>

Description

Generate automatic reports with a complete analysis of a set of time series. For now, SARIMA (Box & Jenkins approach), Holt-Winters and GRNN analysis are possible. Soon, Multilayer Perceptron, Fuzzy Logic and Box-Cox analysis will become available.

Usage

report(mode = "SARIMA", ts = 21864, parameters = NULL, report.file = NA, series.saveas = "none")
Arguments

mode
A character. The type of the analysis. So far, 'SARIMA', 'GRNN' and 'HOLT-WINTERS' are available.

ts
A integer, a ts object or a list of integers and ts objects. Either the ID of the series in the BETS database or a time series object (any series, not just BETS's). If a list is provided, a report is generated for each series in this list, which can be mixed with IDs and time series objects.

parameters
A list. The parameters of the report. See the 'details' section for more information.

report.file
A character. A path and a name for the report file (an .html file). If there is more than one series, this name will be used as a prefix. If this parameter is not provided, the report will be saved inside the 'reports' folder, under the BETS installation directory.

series.saveas
A character. The format of the file on which the series and the predictions should be written. Possible values are 'none' (default), 'sas', 'dta', 'spss', 'csv', 'csv2'. It is saved under the same directory as the report file.

Details

SARIMA Report Parameters

- cf.lags: An integer. Maximum number of lags to show on the ACFs e PACFs
- n.ahead: An integer. Prevision horizon (number of steps ahead)
- inf.crit: A character. Information criterion to be used in model selection.
- dummy: A ts object. A dummy regressor. Must also cover the forecasting period.
- ur.test: A list. Parameters of `ur_test`
- arch.test: A list. Parameters of `arch_test`
- box.test: A list. Parameters of `Box.test`

GRNN Report Parameters

- auto.reg: A boolean. Is the dependant variable auto-regressive?
- present.regs: A boolean. Include non-lagged series among regressors?
- lag.max: A integer. Regressors' maximum lag
- regs: A list. Regressors codes or time series
- start.train: Training set starting period
- end.train: Training set ending period
- start.test: Testing set starting period
- end.test: Testing set ending period
- sigma.interval: A numeric vector. Sigma interval
- sigma.step: A numeric value. Sigma step
- var.names: A character vector. Variable names
HOLT-WINTERS Report Parameters

- **alpha**: Smooth factor of the level component. If numeric, it must be within the half-open unit interval (0, 1]. A small value means that older values in x are weighted more heavily. Values near 1.0 mean that the latest value has more weight. NULL means that the HoltWinters function should find the optimal value of alpha. It must not be FALSE or 0.

- **beta**: Smooth factor of the trend component. If numeric, it must be within the unit interval [0, 1]. A small value means that older values in x are weighted more heavily. Values near 1.0 mean that the latest value has more weight. NULL means that the HoltWinters function should find the optimal value of beta. The trend component is omitted if beta is FALSE or 0.

- **gamma**: Smooth factors of the seasonal component. If numeric, it must be within the unit interval [0, 1]. A small value means that older values in x are weighted more heavily. Values near 1.0 mean that the latest value has more weight. NULL means that the HoltWinters function should find the optimal value of gamma. The seasonal component will be omitted if gamma is FALSE or 0. This must be specified as FALSE if frequency(x) is not an integer greater than 1.

- **additive**: A single character string specifying how the seasonal component interacts with the other components. "additive", the default, means that x is modeled as level + trend + seasonal and "multiplicative" means the model is (level + trend) * seasonal. Abbreviations of "additive" and "multiplicative" are accepted.

- **l.start**: The starting value of the level component.
- **b.start**: The starting value of the trend component
- **s.start**: The starting values of seasonal component, a vector of length frequency(x)
- **n.ahead**: Prevision horizon (number of steps ahead)

For more information about these parameters, see also `HoltWinters`. Most parameters are the same and we just reproduced their documentation here.

**Value**

One or more .html files (the reports) and, optionally, data files (series plus predictions).

**Author(s)**

Talitha Speranza <talitha.speranza@fgv.br>

**Examples**

```r
#-- SARIMA

# parameters = list(lag.max = 48, n.ahead = 12 )
# report(ts = 21864, parameters = parameters)

# report(ts = 4447, series.saveas = "csv")

# series = list(BETSget(4447), BETSget(21864))
# parameters = list(lag.max = 20, n.ahead = 15 )
# report(ts = series, parameters = parameters)
```
save

Prepare a time series to be exported

Description

To be used with saveSpss, saveSas and others.

Usage

```
save(code = NULL, data = NULL, file.name = "series", type = "")
```

```r
# series = list(4447, 21864)
# report(ts = series, parameters = parameters)

# parameters = list(
#  cf.lags = 25,
#  n.ahead = 15,
#  dummy = dum,
#  arch.test = list(lags = 12, alpha = 0.01),
#  box.test = list(type = "Box-Pierce")
# )
# report(ts = window(BETSget(21864), start = c(2002,1), end = c(2015,10)),
# parameters = parameters)

# dum <- dummy(start = c(2002,1), end = c(2017,1),
# from = c(2008,9), to = c(2008,11))

# parameters = list(
#  cf.lags = 25,
#  n.ahead = 15,
#  dummy = dum
# )

# report(ts = window(BETSget(21864), start = c(2002,1), end = c(2015,10)),
# parameters = parameters)

###-- GRNN

# params = list(regs = 4382)
# report(mode = "GRNN", ts = 13522, parameters = params)

###-- HOLT-WINTERS

# params = list(alpha = 0.5, gamma = TRUE)
# report(mode = "HOLT-WINTERS", ts = 21864, series.saveas = "csv", parameters = params)

# params = list(gamma = T, beta = TRUE)
# report(mode = "HOLT-WINTERS", ts = 21864, series.saveas = "csv", parameters = params)
```
Arguments

- **code**: An integer. The unique identifier of the series within the BETS database.
- **data**: A data.frame or a ts. Contains the data to be written. If data is supplied, the BETS database will not be searched.
- **file.name**: A character. The name of the output file. The default is 'series.spss'.
- **type**: A character. The type of the file (e.g. 'spss' or 'sas').

Value

A list with the data frame to be saved and the file name

---

**saveSas**

*Export a time series to SAS*

Description

Writes a time series to a .sas (SAS) file.

Usage

```
saveSas(code = NULL, data = NULL, file.name = "series")
```

Arguments

- **code**: An integer. The unique identifier of the series within the BETS database.
- **data**: A data.frame or a ts. Contains the data to be written. If data is supplied, the BETS database will not be searched.
- **file.name**: A character. The name of the output file. The default is 'series.sas'.

Value

None

Examples

```
#Exchange rate - Free - United States dollar (purchase)
#us.brl <- get(3691)
#require(seasonal)
#us.brl.seasonally_adjusted <- seas(us.brl)
#saveSas(data = us.brl.seasonally_adjusted, file.name="us.brl.seasonally_adjusted")
# Or
#saveSas(code=3691, file.name="us.brl")
```
saveSpss

Export a time series to SPSS

Description
Writes a time series to a .spss (SPSS) file.

Usage
saveSpss(code = NULL, data = NULL, file.name = "series")

Arguments
- code: An integer. The unique identifier of the series within the BETS database.
- data: A data.frame or a ts. Contains the data to be written. If data is supplied, the BETS database will not be searched.
- file.name: A character. The name of the output file. The default is 'series.spss'.

Examples

```r
#Exchange rate - Free - United States dollar (purchase)
#us.brl <- get(3691)
#requires(seasonal)
#us.brl.seasonally_adjusted <- seas(us.brl)
#saveSpss(data = us.brl.seasonally_adjusted, file.name="us.brl.seasonally_adjusted")
# Or
#saveSpss(code=3691, file.name="us.brl")
```

saveStata

Export a time series to STATA

Description
Writes a time series to a .dta (STATA) file.

Usage
saveStata(code = NULL, data = NULL, file.name = "series")

Arguments
- code: An integer. The unique identifier of the series within the BETS database.
- data: A data.frame or a ts. Contains the data to be written. If data is supplied, the BETS database will not be searched.
- file.name: A character. The name of the output file. The default is 'series.dta'.

Examples

```r
```
Value
None

Examples

#Exchange rate - Free - United States dollar (purchase)
#us.brl <- get(3691)
#requires(seasonal)
#us.brl.seasonally_adjusted <- seas(us.brl)
#saveStata(data = us.brl.seasonally_adjusted,file.name="us.brl.seasonally_adjusted")
# Or
#saveStata(code=3691,file.name="us.brl")

---

Description
Searches the Sidra databases for a series by its description or a given table descriptions.

Usage

sidra.aux(x, len, nova_req, from, to, inputs, territory, variable, header, sections)

Arguments

- **x**: Either a character or a numeric. If character, function searches the Sidra metadata. If a numeric argument is provided the descriptions of the given table are searched.
- **len**: 
- **nova_req**: 
- **from**: 
- **to**: 
- **inputs**: 
- **territory**: 
- **variable**: 
- **header**: 
- **sections**: 
sidraGet

A function to extract Sidra series using their API

Description

The different parameters define the table and its dimensions (periods, variables, territorial units and classification) to be consulted. The parameters that define the sections may vary from table to table. Henceforth, the Sidra function ranges between 5 mandatory arguments to 7. You can only choose one variable per series per request, but multiple sections within the variable.

Usage

sidraGet(x, from, to, territory = c(n1 = "brazil", n2 = "region", n3 = "state", n6 = "city", n8 = "mesoregion", n9 = "microregion", n129 = "citizenship", n132 = "semiarid", n133 = "semiaridUF"), variable, cl = NULL, sections = NULL)

Arguments

- **x**: Sidra series number.
- **from**: A string or character vector specifying where the series shall start.
- **to**: A string or character vector specifying where the series shall end.
- **territory**: Specifies the desired territorial levels.
- **variable**: An integer describing what variable characteristics are to be returned. Defaults to all available.
- **cl**: A vector containing the classification codes in a vector.
- **sections**: A vector or a list of vectors if there are two or more classification codes containing the desired tables from the classification.

Examples

```r
## Not run: sidra = sidraGet(x = c(1612), from = 1990, to = 2015, territory = "brazil", variable =109)
## Not run: sidra = sidraGet(x = c(3653), from = c("200201"),
to = c("201703"), territory = "brazil",
variable = 3135, sections = c(129316,129330), cl = 544)
## Not run: sidra = sidraGet(x = c(3653), from = c("200201"),
to = c("201512"), territory = "brazil", variable = 3135,
sections = "all", cl = 544)
## Not run: sidra = sidraGet(x = c(1618), from = c("201703"), to = c("201703"),
territory = "brazil",
variable = 109, sections=list(c(39427), (39437,39441)), cl = c(49, 48))
## Not run: trim - x = 1620; from = 199001; to = 201701; territory = "brazil";
sections = list(c(90687)); cl =c(11255); variable = 583
## Not run: sidra = sidraGet(x = 1620, from = 199001, to = 201701,
territory = "brazil",
sections=list(c(90687)), cl =c(11255), variable = 583)
## End(Not run)
```
sidraSearch

*Search for Sidra Series*

**Description**

Searches the Sidra databases for a series by its description or a given table descriptions.

**Usage**

sidraSearch(description = NULL, code, view = TRUE, browse = FALSE)

**Arguments**

- **description**: A character argument. Function searches the Sidra metadata and prints results in a window.
- **code**: A numeric argument must be provided. The descriptions of the given table are returned.
- **view**: A boolean. The default is TRUE. If set to FALSE, the results are NOT going to be shown.
- **browse**: A boolean. If browse is set to TRUE, the description table opens in your browser for better visualization.

**Examples**

```r
## Not run:
sidraSearch(description = "pib")
sidraSearch(code = 1248)

## End(Not run)
```

std_resid

*Plot standardized residuals*

**Description**

Uses a model object to create a plot of standardized residuals. This model can be an Arima or an arima. In a near future, this function will also accept objects returned by grnn.train.

**Usage**

std_resid(model, alpha = 0.05)

**Arguments**

- **model**: An Arima or an arima object. The model.
- **alpha**: A numeric between 0 and 1. The significance level.
t_test

Value
Besides showing the plot, this function returns a numeric vector containing the standardized residuals.

Author(s)
Talitha Speranza <talitha.speranza@fgv.br>

Description
Performs the t test on every parameter of an ARIMA model. This model can be an Arima or an arima.

Usage
t_test(model, nx = 0, alpha = 0.05)

Arguments
model An Arima or an arima object. The model for which the parameters must be tested.
nx An integer. The number of exogenous variables
alpha A numeric value between 0 and 1. The significance level.

Value
A data.frame containing the standard erros, the t-statistic, the critical values and whether the null hypothesis should be rejected or not, for each model parameter.

Author(s)
Talitha Speranza <talitha.speranza@fgv.br>, Daiane Marcolino <daiane.mattos@fgv.br>

Examples
require(forecast)
data("AirPassengers")
fit.air<- Arima(AirPassengers,order = c(1,1,1), seasonal = c(1,1,1), method ="ML",lambda=0)
summary(fit.air)

# Significance test for the model SARIMA(1,1,1)(1,1,1)[12]
t_test(model = fit.air)
Performance unit root tests

**Description**

This function uses the package `urca` to perform unit root tests on a pre-defined time series. Unlike urca functions, it returns a meaningful table summarizing the results.

**Usage**

```r
ur_test(..., mode = "ADF", level = "5pct")
```

**Arguments**

- `...` Arguments passed on to `urca` functions
- `mode` A character. The type of the test. Set it to 'ADF' for Augmented Dickey-Fuller, 'KPSS' for KPSS or 'PP' for Phillips-Perron.
- `level` A character. The confidence level. Can be either '1pct' (not for KPSS), '2.5pct', '5pct' or '10pct'

**Value**

A list object. The first element is a data.frame with the test statistics, the critical values and the test results. The second, the model residuals.

**Author(s)**

Talitha Speranza <talitha.speranza@fgv.br>
Index

+Topic **IBGE**
  - sidraSearch, 50
+Topic **bacen**
  - get.series.bacen, 38
+Topic **get**
  - BETSget, 11
  - get.series, 37
+Topic **search**
  - BETSsearch, 12
+Topic **sidra**
  - sidraGet, 49
  - sidraSearch, 50

accuracy, 41, 42
arch_test, 3, 43
Arima, 50, 51
arima, 50, 51

bcbExpectA, 4
bcbExpectATop5, 5
bcbExpectInf12, 6
bcbExpectM, 7
bcbExpectMTop5, 8
bcbExpectT, 9
BETS, 10
BETS-package (BETS), 10
BETS.addin_en, 10
BETS.addin_pt, 11
BETSget, 11
BETSsearch, 12, 12, 37
BETSSources, 15
Box.test, 43

chart, 15, 27, 22
chart.add_basic, 17
chart.add_extra, 18
chart.add_notes, 19
check.series, 20
connection, 20
corrgram, 20

dashboard, 21
deflate, 22
draw.cap_utl, 23
draw.cdb, 24
draw.cei, 24
draw.conf_lvl, 25
draw.eap, 25
draw.ei_comps, 26
draw.ei_vars, 26
draw.gdp_comps, 27
draw.gdp_mon, 27
draw.gdp_unemp, 28
draw.gdp_vars, 28
draw.generic, 29
draw.iie_br, 29
draw.indprod, 30
draw.ipca, 30
draw.lab_coin, 31
draw.lab_lead, 31
draw.lei, 32
draw.misery_index, 32
draw.selic, 33
draw.sent_ind, 33
draw_survey, 34
draw.ulc, 34
draw.unemp, 35
draw.var_gdp, 35
dummy, 35, 36
dygraph, 42

forecast, 20, 41, 42
get.series, 37
get.series.bacen, 38
grnn-package, 39
grrn.test, 38, 42
grrn.train, 38, 39, 39, 50

HoltWinters, 44
message, 40
msg, 40
normalize, 41
plotly, 21
predict, 41
report, 42
save, 45
saveSas, 46
saveSpss, 47
saveStata, 47
search, 11
seas, 12
sidra.aux, 48
sidraGet, 49
sidraSearch, 50
std_resid, 50
t_test, 51
ts, 11, 12, 36
ur_test, 43, 52