

# Package ‘BETS’

October 12, 2017

**Type** Package

**Title** Brazilian Economic Time Series

**Version** 0.3.3

**Date** 2017-10-10

**Maintainer** Pedro Costa Ferreira <pedro.guilherme@fgv.br>

**Depends** R (>= 3.4.1)

**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, DBI, rjson, rvest, xml2, lubridate, htmltools, httr

**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 | file LICENSE

**BugReports** <https://github.com/nmecsyst/BETS/issues>

**URL** <https://github.com/nmecsyst/BETS>

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**VignetteBuilder** knitr

**Repository** CRAN

**Author** Pedro Costa Ferreira [aut, cre],  
Talitha Speranza [aut],  
Jonatha Azevedo [aut],  
Fernando Teixeira [aut],  
Daiane Marcolino [aut]

Date/Publication 2017-10-12 14:03:31 UTC

## R topics documented:

BETS	3
BETS.addin_en	4
BETS.addin_pt	4
BETS.arch_test	4
BETS.chart	5
BETS.corrgram	7
BETS.dashboard	8
BETS.deflate	9
BETS.dummy	10
BETS.get	11
BETS.grnn.test	13
BETS.grnn.train	14
BETS.normalize	15
BETS.predict	15
BETS.report	16
BETS.save	19
BETS.save.sas	20
BETS.save.spss	21
BETS.save.stata	21
BETS.search	22
BETS.sidra.get	24
BETS.sidra.search	25
BETS.std_resid	26
BETS.t_test	27
BETS.ur_test	28
chart.add_basic	28
chart.add_extra	29
chart.add_notes	30
check.series	30
connection	31
draw.cap_utl	31
draw.cdb	32
draw.cei	32
draw.conf_lvl	33
draw.eap	33
draw.ei_comps	34
draw.ei_vars	34
draw.gdp_comps	35
draw.gdp_mon	35
draw.gdp_unemp	36
draw.gdp_vars	36
draw.generic	37
draw.iie_br	37
draw.indprod	38

draw.ipca . . . . .	38
draw.lab_coin . . . . .	39
draw.lab_lead . . . . .	39
draw.lei . . . . .	40
draw.misery_index . . . . .	40
draw.selic . . . . .	41
draw.sent_ind . . . . .	41
draw.survey . . . . .	42
draw.ulc . . . . .	42
draw.unemp . . . . .	43
draw.vargdp . . . . .	43
get.series . . . . .	43
get.series.bacen . . . . .	44
msg . . . . .	45
sidra.aux . . . . .	45

<b>Index</b>	<b>47</b>
--------------	-----------

---

BETS	<i>BETS: A package for obtaining and analysing thousands of Brazilian economic time series.</i>
------	---

---

## 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>

---

BETS.addin_en	<i>BETS search</i>
---------------	--------------------

---

**Description**

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

**Usage**

```
BETS.addin_en()
```

---

BETS.addin_pt	<i>BETS search</i>
---------------	--------------------

---

**Description**

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

**Usage**

```
BETS.addin_pt()
```

---

BETS.arch_test	<i>Perform an ARCH test</i>
----------------	-----------------------------

---

**Description**

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

**Usage**

```
BETS.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>

---

 BETS.chart

*Create a chart with BETS aesthetics*


---

**Description**

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

**Usage**

```
BETS.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)**

VALUE	DESCRIPTION	CODE
'iie_br'	Uncertainty Index	ST_100.0
'sent_ind'	Economic Sentiment Index (average between several confidence indexes)	(*)
'gdp_mon'	GDP Monthly and Interannual Variation (last values) - GDP Monitor (FGV/IBRE)	(*)
'ei_vars'	Economic Indicators (Leading and Coincident) monthly variation	(*)
'ei_comps'	Economic Indicators (Leading and Coincident) components variation	(*)
'lei'	Leading Economic Indicator (LEI - FGV/IBRE with The Conference Board)	(*)
'cei'	Coincident Economic Indicator (CEI - FGV/IBRE with the Conference Board)	(*)
'gdp_vars'	GDP components variation (whole series) - GDP Monitor (FGV/IBRE)	(*)
'misery_index'	Misery Index	13522 plus 24369
'gdp_comps'	GDP components variation (last values) - GDP Monitor (FGV/IBRE)	(*)

'gdp_unemp'	GDP monthly levels versus Unemployment Rate	22109 and 24369
'conf_lvl'	Enterprises Confidence Index versus Consumers Confidence Index	(*)
'inst_cap'	Installed Capacity Index	(*)
'lab_lead'	Labor Leading Indicator	(*)
'lab_coin'	Labor Coincident Indicator	(*)
'transf_ind'	Transformation Industry Confidence Index (Expectations versus Present Situation)	(*)
'servc'	Services Confidence Index (Expectations versus Present Situation)	(*)
'constr'	Construction Confidence Index (Expectations versus Present Situation)	(*)
'retail'	Retail Sellers Confidence Index (Expectations versus Present Situation)	(*)
'consm'	Consumer Confidence Index (Expectations versus Present Situation)	(*)

## 2. Macro Situation Dashboard ('normal' style)

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

(\*) 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.

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

(\*\*) 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

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

---

BETS.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

```
BETS.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 <a href="#">Bartlett's formula</a> . Otherwise, it is going to be simply equal to $1/\sqrt{N}$ .
ci	A numeric value. The confidence interval to be shown in the plot.

style	A character. Set this parameter to 'normal' if you want it made with ggplot2 or to 'plotly' if you want to be a <a href="#">plotly</a> object.
knit	A boolean. If you're using this function to exhibit correlograms on a R dynamic report, set this parameter to true.

**Value**

A plot and a vector containing the correlations.

**Author(s)**

Talitha Speranza <talitha.speranza@fgv.br>

---

BETS.dashboard	<i>Create a BETS custom dashboard</i>
----------------	---------------------------------------

---

**Description**

Generate thematic dashboards using a selection of BETS time series and charts. For now, themes and charts are pre-defined.

**Usage**

```
BETS.dashboard(type = "business_cycle", charts = "all", saveas = NA,
  parameters = NULL)
```

**Arguments**

type	A character. The theme of the dashboard. The only three options, for the time being, is 'business_cycle', 'macro_situation' and 'custom'. Custom dashboards can be rendered with any given set of charts.
charts	A character and/or ts object list. The charts to be added to a custom dashboard. Up to 16 charts are allowed, including pre-defined charts, identified by their codes (see <a href="#">BETS.chart</a> ). This will only work if parameter 'type' is set to 'custom'.
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  
**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 `BETS.chart`, can be either 'plotly' or 'normal'.  
**charts.opts** A list of parameters lists, one for each chart. Parameters are specified in [BETS.chart](#)

**Value**

A .pdf file (the dashboard)

**Author(s)**

Talitha Speranza <talitha.speranza@fgv.br>

**Examples**

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

---

BETS.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.

**Usage**

```
BETS.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>

---

BETS.dummy

*Create a monthly or quarterly dummy*

---

**Description**

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

**Usage**

```
BETS.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](#), [BETS.dummy](#)

**Examples**

```
#1 from a specific date to another specific date
BETS.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:

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

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

#Months equal to 1 only for some years
BETS.dummy(start = c(2000,1), end = c(2012,5), frequency = 12, month = 5, year = 2005:2007)
BETS.dummy(start = c(2000,1), end = c(2012,5), frequency = 12, month = 3, year = c(2005,2007))
BETS.dummy(start = c(2000,1), end = c(2012,5), frequency = 12, month = 5:6, year = c(2005,2007))

#specific dates
BETS.dummy(start = c(2000,1), end = c(2012,5), frequency = 12, date = list(c(2010,1)))
BETS.dummy(start = c(2000,1), end = c(2012,5),
  freq = 12, date = list(c(2010,9), c(2011,1), c(2000,1)) )
```

---

BETS.get

*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**

```
BETS.get(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 <a href="#">BETS.search</a> 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](#), [BETS.search](#) and [seas](#)

**Examples**

```
# Annual series: GDP at constant prices, in R$ (brazilian reais)
#BETS.get(1208)

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

# Exchange rate - Free - United States dollar (purchase)
#us.br1 <- BETS.get(3691)

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

```
# BETS.get(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")  
# BETS.get(code = c(10777,4447), from = f, to = t)  
  
# BETS.get(code = c(10777,4447),from = "2001-10-31", to = c("2014-10-31","2015-01-01"))  
# BETS.get(code = c(10777,4447),from = c("2002-10-31","1997-01-01"), to = "2015-01-01")
```

---

BETS.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

```
BETS.grnn.test(results, test.set)
```

## Arguments

results	The object returned by <a href="#">BETS.grnn.train</a> .
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 (returned by [grnn](#))
- `sigma`: The sigma parameter
- `id`: The id number of the network, as given by [BETS.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>

---

BETS.grnn.train      *Train a General Regression Neural Network*

---

### Description

Creates a set of probabilistic neural networks as proposed by [Specht \[1991\]](#). The user provides a set of regressors and the function chooses which subset is the best, based on an accuracy measure (by default, the MAPE) between fitted and actual values. These networks have only one parameter, the sigma, which is the standard deviation of each activation function (gaussian) of the pattern layer. Sigma can also be automatically chosen. This function builds on [grnn-package](#).

### Usage

```
BETS.grnn.train(train.set, sigma, step = 0.1, select = TRUE, names = NA)
```

### Arguments

train.set	A <code>ts</code> list (a list of <code>ts</code> objects). The first element must be the dependent variable. The other elements, the regressors.
sigma	A numeric or a numeric vector. The sigma parameter, that is, the standard deviation of the activation functions (gaussians) of the pattern layer. Can be either a fixed value or a range (a vector containing the minimum and the maximum values).
step	A numeric value. If sigma is a range, the user must provide a step value to vary sigma. The function is going to select the best sigma based on MAPE.
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 fuction 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.

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

### Author(s)

Talitha Speranza <talitha.speranza@fgv.br>

---

BETS.normalize      *Normalize a time series*

---

### Description

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

### Usage

```
BETS.normalize(series, mode)
```

### 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>

---

BETS.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

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

**Arguments**

...	arguments passed on to <a href="#">forecast</a> . If the model is a neural network, these arguments will be passed on to <a href="#">BETS.grnn.test</a> .
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 <a href="#">dygraph</a> function will be use 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>

---

BETS.report

*Create dynamic reports with a full analysis of a set of time series*

---

**Description**

Generate automatic reports with a complete analysis of a set of time series. For now, only a SARIMA analysis (Box & Jenkins approach) is possible. In a near future, a GRNN (General Regression Neural Network) analysis will be released. Soon after, Holt-Winters, GRNN, Multilayer Perceptron, Fuzzy Logic and Box-Cox analysis will become available.

**Usage**

```
BETS.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'. 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 [BETS.ur\\_test](#)
- `arch.test`: A list. Parameters of [BETS.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

```
##-- SARIMA

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

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

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

```

# series = list(4447, 21864)
# BETS.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")
# )
# BETS.report(ts = window(BETS.get(21864), start= c(2002,1) , end = c(2015,10)),
# parameters = parameters)

# dum <- BETS.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
# )

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

##-- GRNN

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

##-- HOLT-WINTERS

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

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

```

---

BETS.save

*Prepare a time series to be exported*


---

### Description

To be used with BETS.save.spss, BETS.save.sas and others.

### Usage

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

**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

---

BETS.save.sas	<i>Export a time series to SAS</i>
---------------	------------------------------------

---

**Description**

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

**Usage**

```
BETS.save.sas(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.br1 <- BETS.get(3691)
#require(seasonal)
#us.br1.seasonally_adjusted <- seas(us.br1)
#BETS.save.sas(data = us.br1.seasonally_adjusted,file.name="us.br1.seasonally_adjusted")
# Or
#BETS.save.sas(code=3691,file.name="us.br1")
```

---

BETS.save.spss      *Export a time series to SPSS*

---

**Description**

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

**Usage**

```
BETS.save.spss(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**

```
#Exchange rate - Free - United States dollar (purchase)
#us.br1 <- BETS.get(3691)
#requires(seasonal)
#us.br1.seasonally_adjusted <- seas(us.br1)
#BETS.save.spss(data = us.br1.seasonally_adjusted, file.name="us.br1.seasonally_adjusted")
# Or
#BETS.save.spss(code=3691, file.name="us.br1")
```

---

BETS.save.stata      *Export a time series to STATA*

---

**Description**

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

**Usage**

```
BETS.save.stata(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'.

**Value**

None

**Examples**

```
#Exchange rate - Free - United States dollar (purchase)
#us.br1 <- BETS.get(3691)
#requires(seasonal)
#us.br1.seasonally_adjusted <- seas(us.br1)
#BETS.save.stata(data = us.br1.seasonally_adjusted, file.name="us.br1.seasonally_adjusted")
# Or
#BETS.save.stata(code=3691, file.name="us.br1")
```

BETS.search

*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**

```
BETS.search(description = "*", src, periodicity, unit, code, start,
            view = TRUE, 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 preceding 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
BCB-Demab	Central Bank of Brazil - Department of Open Market Operations
BCB-Denor	Central Bank of Brazil - Department of Financial System Regulation
BCB-Depec	Central Bank of Brazil - Department of Economics
Sisbacen	Central Bank of Brazil Information System
Abecip	Brazilian Association of Real Estate Loans and Savings Companies

- Possible values for the parameter periodicity:

A	annual data
M	monthly data
Q	quarterly 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.

code	The code/index of the series within the database
description	The description of the series
periodicity	The periodicity of the series
start	Starting date of the series
source	The source of the series
unit	The unit of measurement of the data

**Note**

This function uses specific version of the [RMySQL](#) package. If it does not run correctly, try installing version 0.10.9 of the [RMySQL](#) package using:

```
> remove.packages("RMySQL") # If necessary
> install.packages("devtools") # If necessary
> devtools::install_version("DBI", version = "0.5", repos = "http://cran.us.r-project.org")
> devtools::install_version("RMySQL", version = "0.10.9", repos = "http://cran.us.r-project.org")
```

**References**

Central Bank of Brazil

**Examples**

```
#not run
#BETS.search(description="sales",view = FALSE)
# Output: BETS-package: Found 55 out of 12981 time series

#BETS.search(src="Denor", view = FALSE)
# Output: BETS-package: Found 1 out of 12981 time series

#BETS.search(periodicity="A", view = FALSE)
# Output: BETS-package: Found 2308 of 12981 time series
```

---

BETS.sidra.get

*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**

```
BETS.sidra.get(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**

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

---

BETS.sidra.search      *Search for Sidra Series*

---

**Description**

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

**Usage**

```
BETS.sidra.search(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**

```
## Not run:
BETS.sidra.search("pib")
BETS.sidra.search(1248)

## End(Not run)
```

---

BETS.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 [BETS.grnn.train](#).

**Usage**

```
BETS.std_resid(model, alpha = 0.05)
```

**Arguments**

model	An <a href="#">Arima</a> or an <a href="#">arima</a> object. The model.
alpha	A numeric between 0 and 1. The significance level.

**Value**

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

**Author(s)**

Talitha Speranza <talitha.speranza@fgv.br>

---

**BETS.t\_test***Test the significance of the parameters of an ARIMA model*

---

**Description**

Performs the t test on every parameter of an ARIMA model. This model can be an [Arima](#) or an [arima](#).

**Usage**

```
BETS.t_test(model, nx = 0, alpha = 0.05)
```

**Arguments**

model	An <a href="#">Arima</a> or an <a href="#">arima</a> 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 errors, 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]
BETS.t_test(model = fit.air)
```

---

BETS.ur\_test                      *Perform 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

```
BETS.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>

---

chart.add\_basic                      *Create a chart of the Unitary Labor Cost time series*

---

### 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 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. the 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	<i>Add notes</i>
-----------------	------------------

---

**Description**

Add notes

**Usage**

```
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	<i>Check series</i>
--------------	---------------------

---

**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	<i>Connection with the server</i>
------------	-----------------------------------

---

**Description**

Make the connection with the server

**Usage**

connection()

---

draw.cap_utl	<i>Create a chart of the Base Interest Rate (SELIC) time series</i>
--------------	---

---

**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**

```
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**

```
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	<i>Create a chart of the Base Interest Rate (SELIC) time series</i>
---------------	---

---

**Description**

Creates a plot of series 4189

**Usage**

```
draw.ei_comps()
```

**Value**

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

---

draw.ei_vars	<i>Create a chart of the Base Interest Rate (SELIC) time series</i>
--------------	---

---

**Description**

Creates a plot of series 4189

**Usage**

```
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**

draw.gdp\_comps()

**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\_mon

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

---

**Description**

Creates a plot of series 4189

**Usage**

draw.gdp\_mon()

**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_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	<i>Create a chart of the Base Interest Rate (SELIC) time series</i>
--------------	---

---

**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	<i>Create a chart of the Base Interest Rate (SELIC) time series</i>
-------------	---

---

**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>

`draw.lei`*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>

---

`draw.misery_index`*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**

```
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**

```
draw.ulc()
```

**Value**

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

---

draw.unemp	<i>Create a chart of the Open Unemployment Rate time series</i>
------------	---

---

**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	<i>Create a chart of the Real Percentage Change of GDP in the Year time series</i>
-------------	--

---

**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.

---

get.series	<i>Get a complete time series from a BETS database</i>
------------	--

---

**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 <a href="#">BETS.search</a> 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 xls format. Defaults to not saving.

**Author(s)**

Fernando Teixeira <fernando.teixeira@fgv.br> and Jonatha Azevedo <jonatha.costa@fgv.br>

---

msg	<i>Format and show a console message.</i>
-----	---

---

**Description**

Customizes a message and shows it in the console.

**Usage**

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

**Arguments**

...	Arguments to be passed to <a href="#">message</a>
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.

**Value**

None

**Author(s)**

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

---

sidra.aux	<i>Search for Sidra Series</i>
-----------	--------------------------------

---

**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 meta-data. If a numeric argument is provided the descriptions of the given table are seached .
len	A .
nova_req	A .
from	A .
to	A .
inputs	A .
territory	A .
variable	A .
header	A .
sections	A .

# Index

- \*Topic **IBGE**
  - BETS.sidra.search, 25
- \*Topic **bacen**
  - get.series.bacen, 44
- \*Topic **get**
  - BETS.get, 11
  - get.series, 43
- \*Topic **search**
  - BETS.search, 22
- \*Topic **sidra**
  - BETS.sidra.get, 24
  - BETS.sidra.search, 25

accuracy, 15, 16

Arima, 26, 27

arima, 26, 27

BETS, 3

BETS-package (BETS), 3

BETS.addin\_en, 4

BETS.addin\_pt, 4

BETS.arch\_test, 4, 17

BETS.chart, 5, 8, 9

BETS.corrgram, 7

BETS.dashboard, 8

BETS.deflate, 9

BETS.dummy, 10, 11

BETS.get, 11

BETS.grnn.test, 13, 16

BETS.grnn.train, 13, 14, 26

BETS.normalize, 15

BETS.predict, 15

BETS.report, 16

BETS.save, 19

BETS.save.sas, 20

BETS.save.spss, 21

BETS.save.stata, 21

BETS.search, 12, 22, 44

BETS.sidra.get, 24

BETS.sidra.search, 25

BETS.std\_resid, 26

BETS.t\_test, 27

BETS.ur\_test, 17, 28

Box.test, 17

chart.add\_basic, 28

chart.add\_extra, 29

chart.add\_notes, 30

check.series, 30

connection, 31

draw.cap\_utl, 31

draw.cdb, 32

draw.cei, 32

draw.conf\_lvl, 33

draw.eap, 33

draw.ei\_comps, 34

draw.ei\_vars, 34

draw.gdp\_comps, 35

draw.gdp\_mon, 35

draw.gdp\_unemp, 36

draw.gdp\_vars, 36

draw.generic, 37

draw.iie\_br, 37

draw.indprod, 38

draw.ipca, 38

draw.lab\_coin, 39

draw.lab\_lead, 39

draw.lei, 40

draw.misery\_index, 40

draw.selic, 41

draw.sent\_ind, 41

draw.survey, 42

draw.ulc, 42

draw.unemp, 43

draw.vargdp, 43

dygraph, 16

forecast, 7, 15, 16

get.series, 43

`get.series.bacen`, 44

`grnn`, 13, 14

`grnn-package`, 14

`HoltWinters`, 18

`message`, 45

`msg`, 45

`plotly`, 8

`RMySQL`, 24

`seas`, 12

`sidra.aux`, 45

`ts`, 11, 12