

Package ‘BETS’

April 30, 2017

Type Package

Title Brazilian Economic Time Series

Version 0.2.1

Date 2017-04-29s

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

Depends R (>= 3.2.0)

Imports grnn, ggplot2, plotly, urca, forecast, sqldf, zoo, rmarkdown,
foreign, seasonal, stringr, dygraphs, RCurl, shiny (>= 0.13),
miniUI (>= 0.1.1), rstudioapi (>= 0.4), DT, webshot, RMySQL,
DBI

Suggests mFilter, lubridate, xts

Description It provides access to and information about the most important
Brazilian economic time series - from the Getulio Vargas Foundation, the Central
Bank of Brazil and the Brazilian Institute of Geography and Statistics. 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/pedrocostaferreira/BETS/issues>

URL <https://github.com/pedrocostaferreira/BETS>

Encoding UTF-8

LazyData true

RoxygenNote 5.0.1

NeedsCompilation no

Author Pedro Costa Ferreira [aut, cre],
Talitha Speranza [aut],
Jonatha Azevedo [aut]

Repository CRAN

Date/Publication 2017-04-30 19:28:27 UTC

R topics documented:

BETS	3
BETS.addin_en	3
BETS.addin_pt	4
BETS.chart	4
BETS.corrrgram	5
BETS.dashboard	6
BETS.deflate	7
BETS.dummy	8
BETS.get	9
BETS.grnn.test	10
BETS.grnn.train	11
BETS.normalize	12
BETS.predict	13
BETS.report	14
BETS.save	16
BETS.save.sas	17
BETS.save.spss	18
BETS.save.stata	18
BETS.search	19
BETS.std_resid	21
BETS.t_test	22
BETS.ur_test	23
check.series	24
draw.animal_spirits	24
draw.cap_utl	25
draw.conf_lvl	25
draw.ei_comps	25
draw.ei_vars	26
draw.gdp_comps	26
draw.gdp_unemp	26
draw.gdp_vars	27
draw.iie_br	27
draw.lab_mrkt	27
draw.misery_index	28
draw.survey	28
get.data.frame	29
get.series.bacen	29
msg	30

BETS

BETS: A package for obtaining and analysing thousands of Brazilian economic time series.

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 given by the Getulio Vargas Foundation (FGV) and make it clear that all data in the package is public domain. We repeat all the rights of all centers from which the data are taken and reaffirm that the major focus of the package is on the academic side.

Author(s)

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

BETS.addin_en

BETS search

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.chart	<i>Create a chart with BETS aesthetics</i>
------------	--

Description

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

Usage

```
BETS.chart(ts, 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.
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:

VALUE	DESCRIPTION	CODE
'animal_spirits'	Expectations index minus Present Situation index	(*)
'iie_br'	Uncertainty Index	ST_100.0
'ei_vars'	Economic Indicators (Leading and Coincident) monthly variation	(*)
'ei_comps'	Economic Indicators (Leading and Coincident) components variation	(*)
'gdp_vars'	GDP components variation (whole series)	(*)
'misery_index'	Misery Index	13522 plus 24369
'gdp_comps'	GDP components variation (last values)	(*)

'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_mrkt'	Labor Indicators (Leading and Coincident)	(*)
'transf_ind'	Transformation Industry Confidence Index (Expectations versus Present Situation)	(*)
'serve'	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)	(*)

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

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.

Note

It is not yet possible to make charts of custom series.

Author(s)

Talitha Speranza <talitha.speranza@fgv.br>

Examples

```
# BETS.chart(ts = "animal_spirits", file = "graphs/animal_spirits", open = T)
# BETS.chart(ts = "ei_vars", file = "graphs/ei_vars.png", open = F)
# BETS.chart(ts = "misery_index")
# BETS.chart(ts = "transf_ind", file = "graphs/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 theses 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 Bartlett's formula . 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 plotly 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

Create a BETS custom dashboard

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 = "sentiment", saveas = NA)
```

Arguments

type	A character. The theme of the dashboard. The only option, for the time being, is 'sentiment'.
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.

Value

A .pdf file (the dashboard)

Author(s)

Talitha Speranza <talitha.speranza@fgv.br>

Examples

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

BETS.deflate	<i>Deflate a time series</i>
--------------	------------------------------

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. The unique code that references the time series. This code can be obtained by using the BETS.search function.
from	A character or a Data object. Starting date of the time series (format YYYY-DD-MM).
to	A character or a Data object. Ending date of the time series (format YYYY-DD-MM).
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)
```

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 [BETS.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 (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

- | | |
|------------------------|--|
| <code>train.set</code> | 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. |
| <code>sigma</code> | 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). |
| <code>step</code> | 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 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.

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	<i>Normalize a time series</i>
----------------	--------------------------------

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 forecast . If the model is a neural network, these arguments will be passed on to BETS.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 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, only 'SARIMA' is 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**

- lag.max: An integer Maximum number of lags to show on the ACFs e PACFs
- n.ahead: An integer Prevision horizon (number of steps ahead)

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)

##-- 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	<i>Export a time series to SPSS</i>
----------------	-------------------------------------

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	<i>Export a time series to STATA</i>
-----------------	--------------------------------------

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.brl <- BETS.get(3691)
#requires(seasonal)
#us.brl.seasonally_adjusted <- seas(us.brl)
#BETS.save.stata(data = us.brl.seasonally_adjusted, file.name="us.brl.seasonally_adjusted")
# Or
#BETS.save.stata(code=3691, file.name="us.brl")
```

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 results are NOT going to be shown.
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-Derim	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	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.

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("RMySQL", version = "0.10.9", repos = "http://cran.us.r-project.org")
```

References

Central Bank of Brazil

Examples

```
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.std_resid	<i>Plot standardized residuals</i>
----------------	------------------------------------

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 Arima or an arima 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	<i>Test the significance of the parameters of an ARIMA model</i>
-------------	--

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 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>

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)
```

BETS.ur_test	<i>Perform unit root tests</i>
--------------	--------------------------------

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 = "DF", level = "5pct")
```

Arguments

...	Arguments passed on to urca functions
mode	A character. The type of the test. For now, only the Augmented Dickey-Fuller test is available.
level	A character. The confidence level. Can be either '1pct', '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>

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>

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

Description

Creates a plot of series 4189

Usage

```
draw.animal_spirits()
```

Value

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

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.

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.

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

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

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.

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.

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.

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.

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.

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

Description

Creates a plot of series 4189

Usage

```
draw.lab_mrkt()
```

Value

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

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.

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.

get.data.frame	<i>Get a BETS series as a data.frame.</i>
----------------	---

Description

By default, `BETS.get` returns a `ts` object. However, there are many situations in which is more convenient to work with a `data.frame`. So, `get.data.frame` receives the code of a BETS series and returns a `data.frame` containing the data of the corresponding series. Alternatively, a `ts` can be supplied, in which case the BETS databases will not be searched.

Usage

```
get.data.frame(code, ts = NULL)
```

Arguments

code	An integer. The unique identifier of the series within the BETS database.
ts	An <code>ts</code> object. A time series to be formatted as a <code>data.frame</code> .

Value

A `data.frame`. The first column contains the dates. The second, its values.

get.series.bacen	<i>A function to extract BACEN series using their API</i>
------------------	---

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>

msg

Format and show a console message.

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 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.

Value

None

Index

- *Topic **bacen**
 - get.series.bacen, 29
- *Topic **get**
 - BETS.get, 9
- *Topic **search**
 - BETS.search, 19

- accuracy, 13
- Arima, 21, 22
- arma, 21, 22

- BETS, 3
- BETS-package (BETS), 3
- BETS.addin_en, 3
- BETS.addin_pt, 4
- BETS.chart, 4
- BETS.corrgram, 5
- BETS.dashboard, 6
- BETS.deflate, 7
- BETS.dummy, 8, 8
- BETS.get, 9, 29
- BETS.grnn.test, 10, 13
- BETS.grnn.train, 10, 11, 11, 21
- BETS.normalize, 12
- BETS.predict, 13
- BETS.report, 14
- BETS.save, 16
- BETS.save.sas, 17
- BETS.save.spss, 18
- BETS.save.stata, 18
- BETS.search, 9, 10, 19
- BETS.std_resid, 21
- BETS.t_test, 22
- BETS.ur_test, 23

- check.series, 24

- data.frame, 29
- draw.animal_spirits, 24
- draw.cap_utl, 25
- draw.conf_lvl, 25
- draw.ei_comps, 25
- draw.ei_vars, 26
- draw.gdp_comps, 26
- draw.gdp_unemp, 26
- draw.gdp_vars, 27
- draw.iie_br, 27
- draw.lab_mrkt, 27
- draw.misery_index, 28
- draw.survey, 28
- dygraph, 13

- forecast, 5, 13

- get.data.frame, 29
- get.series.bacen, 29
- grnn, 11, 12
- grnn-package, 11

- HoltWinters, 15

- message, 30
- msg, 30

- plotly, 6

- RMySQL, 21

- seas, 10

- ts, 8, 10, 29