Package: SAvalidation (via r-universe)

September 17, 2024

Title Validation checks on seasonally adjusted time series

Version 0.0.1.7

Description Functions for running validation checks on a pair of time series, an unadjsuted (NSA) and seasonally adjusted (SA) series.

License EUPL+ file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Depends R (>= 4.1.0)

LazyData true

SystemRequirements Java (>= 17)

Imports dplyr, ggplot2, lubridate, quarto, rjd3toolkit (>= 3.2.0), RJDemetra, stats, tidyr, zoo, DT, knitr, kableExtra

Suggests testthat (>= 3.0.0)

Remotes github::rjdverse/rjd3toolkit

Config/testthat/edition 3

Repository https://seasadjwg.r-universe.dev

RemoteUrl https://github.com/SeasAdjwG/SAvalidation

RemoteRef HEAD

RemoteSha c22c8ecff4eb4daa4e60cbc4b79236d04595d33a

Contents

adjust_fact_plot 2
annual_totals_plot 3
cal_effect_plot 4
check_for_calendar_vars 5
check_identical 5
check_negatives 6

check_nsa_sa_ts 6
check_over_adjustment 7
data_to_check 7
get_combined_seasonality_test 8
level1_validation 8
level2_validation 9
nsa_sa_plot 10

Index **12**

`adjust_fact_plot` *Plot adjustment factor*

Description

Plot adjustment factor

Usage

```
adjust_fact_plot(
  nsa,
  sa,
  title = NULL,
  easter_lag = 6,
  julian_easter = FALSE,
  default_type = "X13",
  default_spec_nsa = "RSA2c",
  add_mean = TRUE
)
```

Arguments

`nsa`	ts object (usually unadjusted time series)
`sa`	ts object (usually seasonally adjusted time series)
`title`	optional title
`easter_lag`	integer defining the number of days before Easter Sunday to create the Easter window
`julian_easter`	logical, should EAster be a Julian Easter, default FALSE
`default_type`	values should be "X13" or "TS" to define whether X13 or TRAMO-SEATS is used to test for decomposition mode
`default_spec_nsa`	name of a default JDemetra+ specification to use for determining decomposition mode (default is "RSA2c")
`add_mean`	boolean indicating if the mean by quarter should be added

Value

A plot of derived adjustment factors

Examples

```
data(data_to_check, package = "SAvalidation")
adjust_fact_plot(data_to_check$nsa,data_to_check$sa)
```

---

annual_totals_plot *Plot of relative difference of annual totals*

Description

Plot of relative difference of annual totals

Usage

```
annual_totals_plot(nsa, sa, title = NULL)
```

Arguments

nsa	ts object (usually unadjusted time series)
sa	ts object (usually seasonally adjusted time series)
title	optional title

Value

A plot of the relative difference of annual totals between nsa and sa

Examples

```
data(data_to_check, package = "SAvalidation")
annual_totals_plot(data_to_check$nsa,data_to_check$sa)
```

`cal_effect_plot` *Plot of calendar effects*

Description

Plot of calendar effects

Usage

```
cal_effect_plot(
  nsa,
  sa,
  sa_mod,
  title = NULL,
  default_type = "X13",
  default_spec_nsa = "RSA2c"
)
```

Arguments

`nsa`	ts object (usually unadjusted time series)
`sa`	ts object (usually seasonally adjusted time series)
`sa_mod`	An SA object from RJDemetra
`title`	optional title
`default_type`	values should be "X13" or "TS" to define whether X13 or TRAMO-SEATS is used to test for decomposition mode
`default_spec_nsa`	name of a default JDemetra+ specification to use for determining decomposition mode (default is "RSA2c")

Value

A plot of calendar effects and derived adjusted effects

Examples

```
test_sa_mod <- RJDemetra::x13(data_to_check$sa,
spec="RSA2c",
userdefined = RJDemetra::user_defined_variables("X13-ARIMA"))
cal_effect_plot(data_to_check$nsa,data_to_check$sa,test_sa_mod)
```

check_for_calendar_vars *Extract test for calendar effects*

Description

Extract test for calendar effects

Usage

```
check_for_calendar_vars(sa_mod)
```

Arguments

sa_mod	An SA object from RJDemetra

Value

A logical indicating whether calendar regressors found in the SA model object

Examples

```
data(data_to_check)
nsa <- data_to_check$nsa
jd_mod <- RJDemetra::x13(nsa)
check_for_calendar_vars(jd_mod)
```

check_identical *Check if two series are identical*

Description

Check if two series are identical

Usage

```
check_identical(nsa, sa)
```

Arguments

nsa	A ts object
sa	A ts object

Value

Logical TRUE if all(nsa==sa)

Examples

```
data(data_to_check)
check_identical(data_to_check$nsa,data_to_check$sa)
```

`check_negatives` *Check for negatives*

Description

Check for negatives

Usage

```
check_negatives(series_to_check)
```

Arguments

`series_to_check`	A numeric vector (or matrix) to check for any negative values

Value

Logical TRUE if any negative values exist

Examples

```
data(data_to_check)
check_negatives(data_to_check$nsa)
```

`check_nsa_sa_ts` *Check nsa and sa are univariate time series of same period*

Description

Check nsa and sa are univariate time series of same period

Usage

```
check_nsa_sa_ts(nsa, sa)
```

Arguments

`nsa`	ts object
`sa`	ts object

Value

stops if problem with nsa and sa time series

Examples

```
data(data_to_check)
check_nsa_sa_ts(data_to_check$nsa,data_to_check$sa)
```

---

check_over_adjustment *Check for over adjustment*

Description

Tests null hypothesis that acf at lag 4 is greater or equal to zero

Usage

```
check_over_adjustment(sa_mod, pval = 0.05)
```

Arguments

sa_mod	An SA object from RJDemetra generated with userdefined="preprocessing.model.y_lin"
pval	numeric value used to define a p-value to use as a threshold for accepting/rejecting null hypothesis (default 0.05)

Value

Logical evidence of

Examples

```
data(data_to_check)
sa <- data_to_check$sa
jd_mod <- RJDemetra::x13(sa,userdefined="preprocessing.model.y_lin")
check_over_adjustment(jd_mod)
```

---

data_to_check *Example data for validation*

Description

Example data for validation

Format

list object with name, nsa and sa time series

get_combined_seasonality_test *Extract combined test for seasonality on linearised series*

Description

Extract combined test for seasonality on linearised series

Usage

```
get_combined_seasonality_test(sa_mod)
```

Arguments

sa_mod	An SA object from RJDemetra

Value

A character vector stating the final result of the combined test for seasonality

Examples

```
data(data_to_check)
nsa <- data_to_check$nsa
jd_mod <- RJDemetra::x13(nsa)
get_combined_seasonality_test(jd_mod)
```

level1_validation *Level 1 validation check*

Description

Level 1 validation check

Usage

```
level1_validation(
  nsa,
  sa,
  default_type = "X13",
  default_spec_nsa = "RSA1",
  default_spec_sa = "RSA2c"
)
```

Arguments

nsa	A ts object
sa	A ts object
default_type	Character must be either "X13" (default) or "TS" determining whether X13 or TRAMO-SEATS is used for testing
default_spec_nsa	Character defining the JDemetra+ specification for tests on the nsa series (default="RSA1")
default_spec_sa	Character defining the JDemetra+ specification for tests on the sa series (default="RSA2c")

Value

Message about the level 1 validation (pass, pass with warnings or fail)

Examples

```
data(data_to_check)
level1_validation(data_to_check$nsa,data_to_check$sa)
```

---

level2_validation *Title*

Description

Title

Usage

```
level2_validation(
  nsa,
  sa,
  series_name,
  dataset_name = "National Accounts Main Aggregates",
  title = series_name,
  output_directory = NULL,
  dashboard_template = "skeleton.qmd",
  start_date = "1999-01-01",
  default_type = "X13",
  default_spec_nsa = "RSA2c",
  default_spec_sa = "RSA2c",
  java_home = Sys.getenv("JAVA_HOME")
)
```

Arguments

nsa	ts object (usually unadjusted time series)
sa	ts object (usually seasonally adjusted time series)
series_name	a name for the time series to be analysed
dataset_name	a name for the dataset
title	title
output_directory	optional output directory for dashboard (default uses getwd())
dashboard_template	name of dashboard template to use
start_date	Character defining start date in format "YYYY-MM-DD"
default_type	values should be "X13" or "TS" to define whether X13 or TRAMO-SEATS is used to test for decomposition mode
default_spec_nsa	name of a default JDemetra+ specification to use for tests on NSA series (default is "RSA2c")
default_spec_sa	name of a default JDemetra+ specification to use for tests on SA series (default is "RSA2c")
java_home	JAVA_HOME environment variable

Value

creates an html dashboard with series name in given output directory

Examples

```
data(data_to_check, package = "SAvalidation")
level2_validation(data_to_check$nsa,data_to_check$sa,data_to_check$name)
```

---

nsa_sa_plot *Plot nsa and sa*

Description

Plot nsa and sa

Usage

```
nsa_sa_plot(nsa, sa, title = NULL)
```

Arguments

nsa	ts object (usually unadjusted time series)
sa	ts object (usually seasonally adjusted time series)
title	Optional title

Value

ggplot of nsa and sa series

Examples

```
data(data_to_check)
nsa_sa_plot(data_to_check$nsa,data_to_check$sa)
```

Index

∗ **data**
data_to_check, 7

adjust_fact_plot, 2
annual_totals_plot, 3

cal_effect_plot, 4
check_for_calendar_vars, 5
check_identical, 5
check_negatives, 6
check_nsa_sa_ts, 6
check_over_adjustment, 7

data_to_check, 7

get_combined_seasonality_test, 8

level1_validation, 8
level2_validation, 9

nsa_sa_plot, 10