JDemetra+ online documentation

Anna Smyk, Alain Quartier-la-Tente, Tanguy Barthelemy, Karsten Webel

12/23/2022

Table of contents

Preface

Welcome to the JDemetra+ online documentation.

JDemetra+ is a software for seasonal adjustment and other time series functions, developed in the framework of Eurostat's "Centre of Excellence on Statistical Methods and Tools" by the National Bank of Belgium with the support of the Bundesbank.

To learn more about this project https://ec.europa.eu/eurostat/cros/content/centre-excellence-statistical-methods-and-tools.

JDemetra+

Introduction

JD+ is a library of algorithms for seasonal adjustment and time series econometrics. You can learn more about the history of the project here (link to below)

link to key references - handbooks (3) - sets of guidelines (2 or 3 ?)

Version 2.2.3 and version 3

approach

as v2 still widely used …

Main functions

Seasonal adjustment algorithms

All are available for low and high frequency data.

	Algorithms	Access	Key features
X13-Arima			
Tramo-Seats			
STL			
State Space Models (STS)			

Algorithms	Access	Key features

Temporal Disaggregation and benchmarking

Algorithms	Access	Key features
Chow-lin		
Fernandez		

Trend-cycle estimation

Algorithms	Access	Key features

Nowcasting

Algorithms	Access	Key features

Structure of this book

This book is divided in four parts, allowing the user to access the resources from different perspectives.

Algorithms

This part provides a step by step description of all the algorithms featured in JD+, grouped by purpose - seasonal adjustment - benchmarking - temporal disaggregation - … links

Tools

JDemetra+ offers 3 kinds of tools - A Graphical User Interface (GUI) which can be enhanced with plug-ins - A set of R packages - A Cruncher for mass production in seasonal adjustment

Underlying Statistical Methods

This part gives details about the underlying statistical methods to foster a more in-depth understanding of the algorithms.Those methods are described in the light and spirit of their use as building blocks of the algorithms presented above, not aiming at all at their comprehensive coverage.

How to use this book

Audience

This book targets the beginner as well as seasoned methodologist interested in using JDemetra+ software for any the purposes listed below.

The documentation is built in layers allowing to skip details and complexity in the first steps

Part I

Algorithms

Seasonal Adjustment

Hint at GUI chap for global principles - ws structure (new SAP) - global specification handling - file structure explanation

Hint at tool selection (link to chapter)

Motivation

The primary aim of the seasonal adjustment process is to remove seasonal fluctuations from the time series. Seasonal fluctuations are quasi-periodic infra-annual movements. They can mask…

[insert def SA from b_ov]

- infra annual peridodic numbers
- no unamigous definititon
- depends on its practial estimation

Data frequencies : high and low

Until recently seasonal where designed to tackly monthly frequencies or lower in JD+, X13-Seats Arima : 2,4,12 Seats : 2,3,4,6,12

For infra monthly data

Unobserved Components (UC)

The main components, each representing the impact of certain types of phenomena on the time series (X_t), are:

- The trend (T_t) that captures long-term and medium-term behaviour;
- The seasonal component (S_t) representing intra-year fluctuations, monthly or quarterly, that are repeated more or less regularly year after year;

- The irregular component (I_t) combining all the other more or less erratic fluctuations not covered by the previous components.

In general, the trend consists of 2 sub-components:

- The long-term evolution of the series;
- The cycle, that represents the smooth, almost periodic movement around the long-term evolution of the series. It reveals a succession of phases of growth and recession. Trend and cycle are not separated in SA alogrithms.

Two decomposition models:

- The additive model: $X_t = T_t + S_t + I_t$;
- The multiplicatve model: $X_t = T_t \times S_t \times I_t$.

In the Reg-Arima (link) or Tramo(link) modelling the the multplicative model is estimated in logs; $log(X_t) = log(T_t) + log(S_t) + log(I_t)$; Seats decompostion also relies on logs wheres X-11 decomp operates…

Detecting seasonal patterns

Seasonality tests (chap M_Tests)

Available algorithms

(link to table)

The two most popular are..

X-13ARIMA is a seasonal adjustment program developed and supported by the U.S. Census Bureau. It is based on the U.S. Census Bureau's earlier X-11 program, the X-11-ARIMA program developed at Statistics Canada, the X-12-ARIMA program developed by the U.S. Census Bureau.

Tramo-Seats

Those will be the most detailed in this doc, but there alternative solutions.

STL

BSM

X13 and TS are two step algos, pretreatment +dcomp (no pre ptreatment in STL + all at once in BSM)

- pre-treatment objectives and methods
- decomposition objective and methods

Pre-treatment : Reg-ARIMA (or TRAMO)

rationale

NOTE: Modelling

this should just be a "note" not a part

(small addendum) the reg arima part can be run without decompostion slight differences in parameters

ref to additionnal chapter ?

Global outline

let's call it regArima modelling (and tramo only when TRamo differences are outlined)

GOAL : remove deterministic effects to improve decompostion (details on modelling: link)

global structure regression with ARIMA structure for residuals

$$Y_t = \sum \alpha_i O_{it} + \sum \beta_j C_{jt} + \sum \gamma_i U_{it} + Y_{lin,t}$$

where - O_{it} are the i final outliers (Ao, LS, TC) from an automatic detection pocedure (link to outliers chap)

- C_{it} are the calendar regressors (automatic or user-defined) (link to calendar chap)
- U_{it} are all the other user-defined regressors (link to outliers and regressors chap)
- $Y_{lin,t} \sim ARIMA(p, d, q)(P, D, Q)$

In the following sections we cover in more details: not the tight structure - Default params: (quick launch) INPUT (?) - Setting User-defined parameters (and not at the end post diagnostics ?) - Output1: series - Output2: Final Params - Output3: Diagnostics for each of these steps - calendar (auto and user) - outliers (auto and user) - adding intervention variables (here: just how to feed the model, details on regressor and calendar specif chapter) - aima models parameters annd diagno

RegArima modelling part can be of an SA process or run on its own - GUI - R (sligthly different specs,...)

WARNING : set params + retrieve their values: global principles in GUI

Retrieve: in the specification window and panels for main (to detail ?)

in R

Default Specifications

here general explanations

default specification are set for the whole sa procedure: pre-treatment end decomposition

they are slightly different for X13-ARIMA and tramo seats

the user chooses the algo + one default spec

this default spec can be modified with user defined params for most of its parts, this is tackeled in details dor every sub part of the reg arima (tramo) modelling and corresponding decompostion (x11 and seats)

Strating point for X13-ARIMA

Spec identifier	Log/level detection	Outliers detection	Calendar effects	ARIMA
RSA0	*NA*	*NA*	*NA*	Airline(+mean)
RSA1	automatic	AO/LS/TC	*NA*	Airline(+mean)
RSA2c	automatic	AO/LS/TC	2 TD vars+Easter	Airline(+mean)
RSA3	automatic	AO/LS/TC	*NA*	automatic
RSA4c	automatic	AO/LS/TC	2 TD vars+Easter	automatic
RSA5	automatic	AO/LS/TC	7 TD vars+Easter	automatic
X-11	*NA*	*NA*	*NA*	*NA*

explanations :

NA : non applied

X11

RSA0

RSA3: no calendar effect correction

automatic: test is performed,

outliers : AO/LS/TC type of outliers automatically detected under a critical T-Stat value (defautl value=4)

calendar

- 2 regresssors: weekdays vs week-ends + LY
- 7 regressors: each week day vs sunadys + LY
- always tested
- easter tested (length = 6 days in tramo, 8 days in X13 arima)

arima model automatic detection

TEST : RSA0 : mult or add

TD vars include LY (link to calendar)

Strating point for Tramo-Seats

Spec identifier	Log/level detection	Outliers detection	Calendar effects	ARIMA
RSA0	*NA*	*NA*	*NA*	Airline(+mean)
RSA1	automatic	AO/LS/TC	*NA*	Airline(+mean)
RSA2	automatic	AO/LS/TC	2 TD vars+Easter	Airline(+mean)
RSA3	automatic	AO/LS/TC	*NA*	automatic
RSA5	automatic	AO/LS/TC	6 TD vars+Easter	automatic
RSAfull	automatic	AO/LS/TC	automatic	automatic

Principle of user setting parameters: can be done from one of the default specs or any spec in “save” as mode very similar inGUI and R

Spans

Estimation span

principle + some differences between tramo and reg-arima

Specifies the span (data interval) of the time series to be used in the seasonal adjustment process (no computations outside) The user can restrcit the span

Table Common settings

Option	Description (expected format)
All	default
From	first observation included (yyyy-mm-dd)
To	last observation included (yyyy-mm-dd)
Between	interval [from ; to] inculded (yyyy-mm-dd to yyyy-mm-dd)
First	number of obs from the begining of the series included (dynamic) (integer)
Last	number of obs from the end of the series (dynamic)(integer)
Excluding	excluding N first obs and P last obs from the computation,dynamic) (integer)
Preliminary check	chek to exclude highly problematic series e.g. the series with a number of identical observations and/or missing values above pre-specified threshold values. (True/False)

Setting in GUI image1: specification window and expanded nodes

Setting in R, part of modifying a specification (link)

```
library(RJDemetra)
# estimation interval: option with static dates
user_spec_1<-x13_spec(spec = c("RSA5c", "RSA0", "RSA1", "RSA2c", "RSA3", "RSA4c", "X11"),
preliminary.check = TRUE,
estimate.from = "2012-06-01",
estimate.to = "2019-12-01")

# estimation interval: option with dynamic numbers of observations

#
# spec can be applied on different series and therefore exclude different dates
user_spec_2<-x13_spec(spec = c("RSA5c", "RSA0", "RSA1", "RSA2c", "RSA3", "RSA4c", "X11"),
estimate.first = 12)

# eestimation on the last 120 obs
user_spec_3<-x13_spec(spec = c("RSA5c", "RSA0", "RSA1", "RSA2c", "RSA3", "RSA4c", "X11"),
estimate.last = 120)

#excluding first 24 and last 36 observations
user_spec_4<-x13_spec(spec = c("RSA5c", "RSA0", "RSA1", "RSA2c", "RSA3", "RSA4c", "X11"),
estimate.exclFirst = 24,
```

```
estimate.exclLast = 36)

# Retrieve settings
```

For comprehensive details about x13_spec function see {RJDemetra} documentation for tramo seats : the same ..one example outlining differences

[V3(JDemetra+ V3) Example in X13]

Tramo-seats

(JD+ version 2) With RJDemetra, example in Tramo-Seats:

```
#excluding first 24 and last 36 observations
user_spec_1<-tramoseats_spec( spec = c("RSAfull", "RSA0", "RSA1", "RSA2", "RSA3", "RSA4",
estimate.exclFirst = 24,
estimate.exclLast = 36)
```

Model span

Principle: same as estimation span (link) above

Specifies the span (data interval) of the time series to be used for the estimation of the RegARIMA model coefficients. The RegARIMA model is then applied to the whole series. With this argument the data early in the series can be prevented from affecting the forecasts, or, alternatively, data later in the series are excluded from the modelling span, so that they do not influence the regression estimates. The available parameter values are

Additionnal parameters

Model span specific parameters

Tolerance	Convergence tolerance for the nonlinear estimation. The absolute changes in the log-likelihood are compared to Tolerance to check the convergence of the estimation iterations. The default setting is 0.0000001.
Tramo specifc parameters	
Exact ML	When this option is marked, an exact maximum likelihood estimation is performed. Alternatively, the Unconditional Least Squares method is used. However, in the current version of JDemetra+ it is not recommended to change this parameter's value

Unit Root Limit	Limit for the autoregressive roots. If the inverse of a real root of the autoregressive polynomial of the ARIMA model is higher than this limit, the root is set equal to 1. The default parameter value is 0.96.

Retrieve in GUI:

image specification window and nodes

Retrieve in R: not possible (to confirm), estimation span and model span must be equal

so additional parameters for model span are set on the estimation span

Tramo seats exemple

```
#excluding first 24 and last 36 observations
user_spec_1<-tramoseats_spec( spec = c("RSAfull", "RSA0", "RSA1", "RSA2", "RSA3", "RSA4",
estimate.tol = 0.0000001,
estimate.eml = FALSE,
estimate.urfinal = 0.98)
```

[V3 model span specif parameters]

Decomposition Scheme

Principle

ecriture add

ecriture multi

if multi : log

ecriture log

consequences on X11 decomp and seats decomp

Transformation test – a test is performed to choose between an additive decomposition (no transformation) (link to reg A chap to detail this)

Settings

Function

transform {function=}

Transformation of data. 2 The user can choose between:

None – no transformation of the data;

Log – takes logs of the data;

Auto – the program tests for the log-level specification. This option is recommended for automatic modelling of many series.

The default setting is Auto.

Reg-Arima specific settings

AIC difference

transform {aicdiff=}

Defines the difference in AICC needed to accept no transformation over a log transformation when the automatic transformation

```
selection option is invoked. The option is disabled when Function is not set to Auto. The de
```

Adjust

transform {adjust=}

Options for proportional adjustment for the leap year effect. The option is available when Function is set to Log. Adjust can be set to:

LeapYear – performs a leap year adjustment of monthly or quarterly data;

LengthofPeriod – performs a length-of-month adjustment on monthly data or length-of-quarter adjustment on quarterly data;

None – does not include a correction for the length of the period.

The default setting is None

Tramo specific settings

Fct
Transformation; fct

Controls the bias in the log/level pre-test (the function is active when **Function** is set to *Auto*); **Fct** $>$ 1 favours levels, **Fct** $<$ 1 favours logs. The default setting is 0.95.

Set and Retrieve in GUI

Image specification window + expanded nodes : RegA+ Tramo

Retrieve: in the specification window and panels for main (to detail ?)

Set and in R

X13

```
#excluding first 24 and last 36 observations
user_spec <-x13_spec(spec = c("RSA5c", "RSA0", "RSA1", "RSA2c", "RSA3", "RSA4c", "X11"),
transform.function ="Log", # c(NA, "Auto", "None", "Log"),
transform.adjust = "LeapYear", #c(NA, "None", "LeapYear", "LengthOfPeriod"),
transform.aicdiff = -3)

#Retrieve settings
```

Tramo seats settings

```
#transfo
user_spec_1<-tramoseats_spec( spec = c("RSAfull", "RSA0", "RSA1", "RSA2", "RSA3", "RSA4",
transform.function = "Auto", #c(NA, "Auto", "None", "Log"),
transform.fct = 0.5)


# Retrieve settings
```

Calendar correction

Rationale : why correct (in a nutshell)

Method regression as deterministic effects, details of regressor building in calendar chapter building

Default specifications for calendar correction

Options related to calendar regressors choice are embedded into default specifications (link)

National calendars not taken into accout.

- working days
- trading days
- Leap year regressor
- Test: Remove/ Add / None
- Easter
- Test add / remove / non

- duration (enabled when testing removed)
- Julian
- pre-test

0.0.0.0.1 * GUI: Setting this options and retrieveing parameters which are in use

0.0.0.0.2 * R: Setting this options and retrieveing parameters which are in use

Customizing calendars

How to correct this ?

solution 1: if working with GUI build a new calendar in GUI (building details in chap C) set this option import it : here

GUI : how to use it or customize HTML file structure explanation

Question: can rjd3modelling generate an HTML calendar for GUI

set this option in GUI

image: spec window calendar / holdays / choics of calendar

set this option in R ?

version 2:

version 3:

solution 2: import external regressors can be built with rjd3modelling (link) which can then be used in via are or imported via GUI

set this option in GUI how to import vaiables into JD+ / set utility (in interface chapter) calssical user defined

set this option in R version 2:

```
spec_4 <- x13_spec(spec = spec_1,
                   tradingdays.option = "UserDefined",
                   tradingdays.test = "None",
                   usrdef.varEnabled = TRUE,
                   usrdef.varType = "Calendar",
                   usrdef.var = reg3) # set of regressors in TS format
```

version 3

(generic sentence to quasi repeat at the end of each part) Once regressors set, the RegArima (tramo) model will be estimated gloablly with all the other regression variables and taking into account arima model specificities as well. That is why diagnostics are all jointly displayed at the end of the process. (link)

link to: worked example: french calendar in R

Outliers and intervention variables

quick intro: rational = improve decomposition (cf guidelines) (determinsitic= effects estimated by regression 2 ways - (customizable) automatic detection - user prespecifified outliers

Automatic detection

overview: one sentence

0.0.0.0.1 * what can be set (list)

(what do we look for) : AO, LS, TC (detailed in outliers chapter link), how cf. RegArima method chapter types can be set and TC rate specified SO can be added to detection detection - detection enabled - detection span - types of outliers (+ rate for TC) - critical value ("method" = is an incative parameter, always add one)

all is identic for TRamo

0.0.0.0.1.1 * how to set this in GUI

image specification window, outliers box commented

0.0.0.0.2 * how ro set this in R

version 2: RJDEemetra

```
# idem for x13 or tramo
# spec_4 <- x13_spec(spec = spec_1,
# outlier.enabled = NA,
#   outlier.from = NA_character_,
#   outlier.to = NA_character_,
#   outlier.first = NA_integer_,
#   outlier.last = NA_integer_,
#   outlier.exclFirst = NA_integer_,
```

```
#   outlier.exclLast = NA_integer_,
#   outlier.ao = NA,
#   outlier.tc = NA,
#   outlier.ls = NA,
#   outlier.so = NA,
#   outlier.usedefcv = NA,
#   outlier.cv = NA_integer_,
#   outlier.method = c(NA, "AddOne", "AddAll"),
#   outlier.tcrate = NA_integer_,
```

(how to retrieve these settings? in regression results cf end))

Pre-specified Outliers

the user can set outliers (in the past) will be complemented by automatic detection (unless disabled)

0.0.0.0.1 * in GUI

image specif window open the calendar image of calendar

0.0.0.0.2 * how to set this in R

version 2: RJDemetra

```
# idem for x13 or tramo
# spec_4 <- x13_spec(spec = spec_1,
```

Other User-defined regressors

rjd3modelling

some principles - allocate to a component - out of calendars and outliers warning: calendars won't be allocated to calendars, if not…

0.0.0.0.1 * Import regressors in GUI

- import
- additionnal operations

0.0.0.0.2 * Import regressors R

```
spec_5 <- x13_spec(spec = spec_1,
                   usrdef.varEnabled = TRUE,
                   # On choisit d'attribuer l'outlier à la composante tendance
                   usrdef.varType = "Trend",
                   # Les régresseurs doivent être au format ts
                   usrdef.var = reg_externe_ts)
```

Arima Model

Once regressors have been defined, the software looks for a structure of the autocorrelations of the lneariezd series

The arima model will be evaluated lookng at its residuals supposed to be a whiet noise

Default specifications

Key specifications on arima model are embedded in default specifications: airline (daufault model) or full automatic reserach.

automdl.enabled a logical. If TRUE, the automatic modelling of the ARIMA model is enabled. If FALSE, the parameters of the ARIMA model can be specified.

A word on restrictions; can some setting be mixed, ex user model and different u roots limits

Forecast horizon is in X-11..

Modifing automatic detection

Control variables for the automatic modelling of the ARIMA model (when automdl.enabled is set to TRUE):

automdl.acceptdefault
a logical. If TRUE, the default model (ARIMA(0,1,1)(0,1,1)) may be chosen in the first step of the automatic model identification. If the Ljung-Box Q statistics for the residuals is acceptable, the default model is accepted and no further attempt will be made to identify another model.

automdl.cancel
the cancelation limit (numeric). If the difference in moduli of an AR and an MA roots (when estimating ARIMA(1,0,1)(1,0,1) models in the second step of the automatic identification of

the differencing orders) is smaller than the cancelation limit, the two roots are assumed equal and cancel out.

automdl.ub1 the first unit root limit (numeric). It is the threshold value for the initial unit root test in the automatic differencing procedure. When one of the roots in the estimation of the ARIMA(2,0,0)(1,0,0) plus mean model, performed in the first step of the automatic model identification procedure, is larger than the first unit root limit in modulus, it is set equal to unity.

automdl.ub2 the second unit root limit (numeric). When one of the roots in the estimation of the ARIMA(1,0,1)(1,0,1) plus mean model, which is performed in the second step of the automatic model identification procedure, is larger than second unit root limit in modulus, it is checked if there is a common factor in the corresponding AR and MA polynomials of the ARMA model that can be cancelled (see automdl.cancel). If there is no cancellation, the AR root is set equal to unity (i.e. the differencing order changes).

automdl.mixed
a logical. This variable controls whether ARIMA models with non-seasonal AR and MA terms or seasonal AR and MA terms will be considered in the automatic model identification procedure. If FALSE, a model with AR and MA terms in both the seasonal and non-seasonal parts of the model can be acceptable, provided there are no AR or MA terms in either the seasonal or non-seasonal terms.

automdl.balanced
a logical. If TRUE, the automatic model identification procedure will have a preference for balanced models (i.e. models for which the order of the combined AR and differencing operator is equal to the order of the combined MA operator).

automdl.armalimit
the ARMA limit (numeric). It is the threshold value for t-statistics of ARMA coefficients and constant term used for the final test of model parsimony. If the highest order ARMA coefficient has a t-value smaller than this value in magnitude, the order of the model is reduced. If the constant term t-value is smaller than the ARMA limit in magnitude, it is removed from the set of regressors.

automdl.reducecv
numeric, ReduceCV. The percentage by which the outlier's critical value will be reduced when an identified model is found to have a Ljung-Box statistic with an unacceptable confidence coefficient. The parameter should be between 0 and 1, and will only be active when automatic outlier identification is enabled. The reduced critical value will be set to (1-ReduceCV)*CV, where CV is the original critical value.

automdl.ljungboxlimit
the Ljung Box limit (numeric). Acceptance criterion for the confidence intervals of the Ljung-Box Q statistic. If the LjungBox Q statistics for the residuals of a final model is greater than

the Ljung Box limit, then the model is rejected, the outlier critical value is reduced and model and outlier identification (if specified) is redone with a reduced value.

automdl.ubfinal numeric, final unit root limit. The threshold value for the final unit root test. If the magnitude of an AR root for the final model is smaller than the final unit root limit, then a unit root is assumed, the order of the AR polynomial is reduced by one and the appropriate order of the differencing (non-seasonal, seasonal) is increased. The parameter value should be greater than one.

setting in GUI

image specification window, arima with auto box ticked

setting in R (see if Tramo specificities)

```
# spec_5 <- x13_spec(spec = spec_1,
# automdl.enabled = NA,
#   automdl.acceptdefault = NA,
#   automdl.cancel = NA_integer_,
#   automdl.ub1 = NA_integer_,
#   automdl.ub2 = NA_integer_,
#   automdl.mixed = NA,
#   automdl.balanced = NA,
#   automdl.armalimit = NA_integer_,
#   automdl.reducecv = NA_integer_,
#   automdl.ljungboxlimit = NA_integer_,
#   automdl.ubfinal = NA_integer_,
```

User-defined Arima model

Control variables for the non-automatic modelling of the ARIMA model (when automdl.enabled is set to FALSE):

arima.mu
logical. If TRUE, the mean is considered as part of the ARIMA model.

arima.p numeric. The order of the non-seasonal autoregressive (AR) polynomial.

arima.d numeric. The regular differencing order.

arima.q numeric. The order of the non-seasonal moving average (MA) polynomial.

arima.bp
numeric. The order of the seasonal autoregressive (AR) polynomial.

arima.bd
numeric. The seasonal differencing order.

arima.bq
numeric. The order of the seasonal moving average (MA) polynomial.

Control variables for the user-defined ARMA coefficients. Coefficients can be defined for the regular and seasonal autoregressive (AR) polynomials and moving average (MA) polynomials. The model considers the coefficients only if the procedure for their estimation (arima.coefType) is provided, and the number of provided coefficients matches the sum of (regular and seasonal) AR and MA orders (p,q,bp,bq).

arima.coefEnabled
logical. If TRUE, the program uses the user-defined ARMA coefficients.

arima.coef
a vector providing the coefficients for the regular and seasonal AR and MA polynominals. The vector length must be equal to the sum of the regular and seasonal AR and MA orders. The coefficients shall be provided in the following order: regular AR (Phi; p elements), regular MA (Theta; q elements), seasonal AR (BPhi; bp elements) and seasonal MA (BTheta; bq elements). E.g.: arima.coef=c(0.6,0.7) with arima.p=1, arima.q=0,arima.bp=1 and arima.bq=0.

arima.coefType
a vector defining the ARMA coefficients estimation procedure. Possible procedures are: "Undefined" = no use of any user-defined input (i.e. coefficients are estimated), "Fixed" = the coefficients are fixed at the value provided by the user, "Initial" = the value defined by the user is used as the initial condition. For orders for which the coefficients shall not be defined, the arima.coef can be set to NA or 0, or the arima.coefType can be set to "Undefined". E.g.: arima.coef = c(-0.8,-0.6,NA), arima.coefType = c("Fixed","Fixed","Undefined").

(for both options) fcst.horizon
the forecasting horizon (numeric). The forecast length generated by the RegARIMA model in periods (positive values) or years (negative values). By default, the program generates a two-year forecast (fcst.horizon set to -2).

setting in GUI

image specification window, arima with auto box not tickec

setting in R (see if Tramo specificities)

```
spec_5 <- x13_spec(spec = spec_1,
automdl.enabled = FALSE,

  arima.mu = NA,
```

```
    arima.p = NA_integer_,
    arima.d = NA_integer_,
    arima.q = NA_integer_,
    arima.bp = NA_integer_,
    arima.bd = NA_integer_,
    arima.bq = NA_integer_,
    arima.coefEnabled = NA,
    arima.coef = NA,
    arima.coefType = NA,
    fcst.horizon = NA_integer_,
```

Tramo and Reg-Arima are very similar…details in M chapter, parameters and output are the same, treated as one here

Context : here in put is spcified (auto or user) focus on global model estimation

Quick launch (= automatic run)

Output1: series (if relevant) Retrieve regressors GUI (cruncher) R

Retrieve Pre-adjustment series Table

in GUI (cruncher)

in R

Output2: Final Params what is selected and estimated by the software (not in the specifs above)

arima orders (if relevant) and coefficients

GUI

R

RegArima model Results and Diagnostics

Regression Results and diagnostics

tables with results for all variables (auto and user-defined) student, F test - tests

Arima model results

0.0.0.0.1 * Final parameters

table with coefficients, student, correlation

in GUI additonnal graphs

in R

0.0.0.0.2 * Goodness of fit Test on Arima model residuals

list

retrieve in GUI - main - prep proc main node datails peut etre pas de capture pour tout: pb de volume la lacture des chapitres doit rester fluide appndix, gui chapter ? - diagnostics node here just residuals and full residuals (what is the difference)

(global post decomp results explained in QA part)

retireve in R v2 rjdemetra (comprehensive inR)

Residual Calendar effects

Residual calendar effects what kinf of test is performed ? td spectral peaks residual effect in SA or I serie

retrieve in GUI

retrieve in R (v2 and v3)

Allocation of pre-treatment effects

(in the decomp part for x11 and seats, table with allocation post decomp)

X-11 Decompostion

this part should allow to use x-11 via R as well as via GUI

Default specifications

link to table from reg-arima model

if identifiers slightly different in GUI, make it clear + pb of names in V3

in calendar effects : what is tested, link to calendar chapter

typos : RJDemetra doc (pull req)

- 7 TD vars

Quick Launch

(linked from intro JDemetra+ intro chapter)

Output 1: series

Series	Final X11 name	Final Results	Reallocation of pre-adjustment effects
Raw series (forecasts)		y (y_f)	
Final seasonal component	D16	s (s_f)	
Final trend	D12	t (t_f)	
Final irregular	D13	i (i_f)	
Calendar component			
Seasonal without calendar	D10		

make clear reallocation of outliers effects (frequent questions)

reallocation of intervention variables effects

att : diff D10 and D16 ?

X-11 gives access to a great part of it's intermediate computations (link to M chapter)

Here we focus on the final components main series (= 2nd part of D tables, outliers effects added)

List of series (edit : table with name and meaning)

Retrieve in GUI (forecasts glued)

link to chap GUI : generate output in GUI (generic) link to chap cruncher : generate output with cruncher

Retrieve in R (forecast in a specific variable)

Link to appendix: final output overview

Output 2: final parameters

Relevant if parameters not set manually, or any parameters automatically selected by the software without having a fixed default value. (The rest of the parameters is set in the spec) To manually set hose parameters and see all the fixed default values see Specifications / parameters section

Final trend filter : length of Henderson filter applied for final estimation (in the second part of the D step).

Final seasonal filer: length of Henderson filter applied for final estimation (in the second part of the D step).

Retrieve via GUI image ? Node Decomposition(X11) > Final Filters (make node clickable, link to GUI chapter, handling of nodes)

Retrieve in R From the model_sa object (link to R package chap global explanation)

```
model_sa$decomposition$s_filter
model_sa$decomposition$t_filter

# add version 3 equivalent
```

Output 3: diagnostics

X11 produces the following type diagnostics or quality measures Table with link to detail - SI ratios

SI-ratios

It is a plot (paste example)

Description, and computation details

Retrieve in GUI NODE Main Results > SI-Ratios

in GUI values cannot be retrieved Retrieve in R

```
# data frame with values
model_sa$decomposition$si_ratio
# customizable plot
plot(model_sa, type= "cal-seas-irr",first_date = c(2015, 1))

# add version 3 equivalent
```

M-statistics

At the end of the decomposition, X-11 algorithm provides quality mesures of the decomposition called "M statistics": 11 statistics (M1 to M11) and 2 summary indicators (Q et Q-M2). By design (clickable) $0 < M_x < 3$ and acceptance region is $M_x \leq 1$ Formula for Q (They are based on F-tables)

- **M1** The relative contribution of the irregular over three months span
- **M2** The relative contribution of the irregular component to the stationary portion of the variance
- **M3** The amount of month to month change in the irregular component as compared to the amount of month to month change in the trend-cycle (I/C-ratio)
- **M5** MCD (Months for Cyclical Dominance): The number of months it takes the change in the trend-cycle to surpass the amount of change in the irregular
- **M6** The amount of year to year change in the irregular as compared to the amount of year to year change in the seasonal (only valid for 3x5 seasonal filter)
- **M7** The amount of moving seasonality present relative to the amount of stable seasonality
- **M8** The size of the fluctuations in the seasonal component throughout the whole series
- **M9** The average linear movement in the seasonal component throughout the whole series
- **M10** Same as 8, calculated for recent years only (4 years, N-2 to N-5)
- **M11** Same as 9, calculated for recent years only

For futher details link to Lothian-Mory (quid in M_meth X11)

Retrieve in GUI NODE Decompostion(X-11) > Quality Measures > Summary

Retrieve in R

```
model_sa$decomposition$mstats
# add version 3 equivalent
```

Detailed Quality measures

List (+ link to chap M: add descp from Lothian Mory/ current doc) make clear: on whic x11 series is the measure based

in GUI all the diagnostics below can be retrieved in NODE Decomposition(X-11) > Quality Measures > Details

If a stat can be retrievedin R, the code example is in the corresponding paragraph, otherwise it is not possible

0.0.0.0.1 * Average differences without regard to sign over the indicated span

0.0.0.0.2 * Relative contribution to the variance of the differences in the components of the original series

0.0.0.0.3 * Average differences with regard to sign and standard deviation over indicated span

0.0.0.0.4 * Average duration of run

0.0.0.0.5 * I/C ratio over indicated span

0.0.0.0.6 * I/C ratio (global)

0.0.0.0.7 * Relative contribution to the stationnary part of the variance in the original series

0.0.0.0.8 * Autocorrelations in the irregular

0.0.0.0.9 * Heteroskedasticity

Cochran test on equal variances within each period

0.0.0.0.10 * Moving seasonality ratios (MSR)

User-defined parameters

Here is described how to change default values or automatic choices.

0.0.0.0.1 * General settings

- **Mode**
  - check if this option still works, if so add and edit instructions from old page)
  - if not but button present : explain that the mode is determined in pre-adjustment (function)
- **Seasonal component**
  - check if still relevant, idem as above
  - in v.2.3 if not ticked, S estimated but options on seasonal filter not available
- **Forecasts horizon**

Length of the forecasts generated by the RegARIMA model - in months (positive values) - years (negative values) - if set to is set to 0, the X-11 procedure does not use any model-based forecasts but the original X-11 type forecasts for one year. - default value: -1, thus one year from the Arima model

- **Backcasts horizon**

Length of the backcasts generated by the RegARIMA model - in months (positive values) - years (negative values) - default value: 0

0.0.0.0.2 * Irregular correction

- **LSigma**
  - sets lower sigma (standard deviation) limit used to down-weight the extreme irregular values in the internal seasonal adjustment iterations, learn more here (LINK to M_ chapter)
  - values in $[0, Usigma]$
  - default value is 1.5
- **USigma**
  - sets upper sigma (standard deviation)
  - values in $[Lsigma, +\infty]$
  - default value is 2.5
- **Calendarsigma**

  - allows to set different **LSigma** and **USigma** for each period
  - values
    - None (default)
    - All: standard errors used for the extreme values detection and adjustment computed separately for each calendar month/quarter
    - Signif: groups determined by cochran test (check)
    - Sigmavec: set two customized groups of periods

- **Excludeforecasts**

  - ticked : forecasts and backcasts from the RegARIMA model not used in Irregular Correction
  - unticked (default): forecasts and backcasts used

0.0.0.0.3 * Seasonality extraction filters choice

Specifies which be used to estimate the seasonal factors for the entire series (link to relevant part in M chapter):

- **Seasonal filter**

- default value: *MSR* (Moving seasonality ratio), automatic choice of final seasonal filter (cf msr defintion and decison table: link), initial filters are computed with 3×3

- choices : 3×1, 3×3, 3×5, 3×9, 3×15 or Stable

- "Stable" : constat factor for each calendar period (simple moving average of all $S + I$ values for each period)

User choices will be applied to all steps or only to final phase D step.

The seasonal filters can be selected for the entire series, or for a particular month or quarter.

- **Details on seasonal filters**

Sets different seasonal filters by period in order to account for seasonal heteroskedasticity (link to M chapter)

- default value: empty

0.0.0.0.4 * Trend estimation filters

- **Automatic Henderson filter** our user-defined
  - default: length 13
  - unticked: user defined length choice
- **Henderson filter** length choice
  - values: odd number in $[3, 101]$
  - default value: 13

Check: will user choice be applied to all steps or only to final phase D step

Parameter setting in GUI

Specification (general or particular to one series) image

All the parameters above can be set with the specification (link to general explanations) box

Parameter setting in R packages

extensive help on functions available in package help pages Rcode snippets

In R, to implement any param change, it is required to retrieve current spec, modify it and apply it again (see T R packages chapter for details). (specific link)

here example changing all the settings (just remove irrelevant changes)

Rjdemetra (v2) Edit : here static R code link to a "worked example" wih dynamic code ticked box in GUI corresponds to …? in R

```
#Creating a modified specification, customizing all available X11 parameters
modified_spec<- x13_spec(current_sa_model,
    #x11.mode="?",
    #x11.seasonalComp = "?",
    x11.fcasts = -2,
    x11.bcasts = -1,
    x11.lsigma = 1.2,
    x11.usigma = 2.8,
    x11.calendarSigma = NA, # EDIT with example
      x11.sigmaVector = NA,
    x11.excludeFcasts = NA
    # filters
```

```
    x11.trendAuto = NA, # needed inf value ?
    x11.trendma = 23,
    x11.seasonalma = "S3X9
    # details on seasonal filters ?)

#New SA estimation : apply modified_spec
modified_sa_model<-x13(raw_series,modified_spec)
```

EDIT : link to package help page v2 +v3

SEATS Decomposition

policy: what is readable in GUI should be explained (even if shortly with link to details here) intermediate series ?

decomp of lineraized series in level or log

Default specifications

link to Tramo-Seats part

if identifiers slightly different in GUI, make it clear + pb of names in V3

in calendar effects : what is tested, link to calendar chapter

typos : RJDemetra doc (pull req)

- 7 TD vars

Quick Launch

overall procedure

in GUI

in R

Output 1: series

global indications: _f: forecasts - _e - -ef

Stochastic series

decompostion of the linearisezed series or of its log if multiplicative model

components: t_lin, s_lin, i_lin

if log, everything is dispalayed in log

Series Name	Meaning
decomposition.y_lin	
decomposition.y_lin_f	
decomposition.y_lin_ef	
decomposition.t_lin	
decomposition.t_lin_f	
decomposition.t_lin_e	
decomposition.t_lin_f	
decomposition.sa_lin	
decomposition.sa_lin_f	
decomposition.sa_lin_e	
decomposition.sa_lin_ef	
decomposition.s_lin	
decomposition.s_lin_f	
decomposition.s_lin_e	
decomposition.s_lin_ef	
decomposition.i_lin	
decomposition.i_lin_f	
decomposition.i_lin_e	
decomposition.i_lin_ef	

Retrieve in GUI

NODE : Decompostion > Stocastic series

forecast f series are pasted at the end of the correspoding series

- _e series are labelled with (stde)

_ef appear in graphs (but values cannot be retrieved)

IMAGE1 GUI

- image 1 : decompostion node and series
- image 2 trend graph
- image 3 seasonal graph

Retrieve in R

Components (Level)

if additive model : same as

decompostion of the linearisezed series or of its log if multiplicative model

components: t_lin, s_lin, i_lin

if log, everything is dispalayed in log

Series Name	Meaning
decomposition.y_lin	
decomposition.y_lin_f	
decomposition.y_lin_ef	
decomposition.t_lin	
decomposition.t_lin_f	
decomposition.t_lin_e	
decomposition.t_lin_f	
decomposition.sa_lin	
decomposition.sa_lin_f	
decomposition.sa_lin_e	
decomposition.sa_lin_ef	
decomposition.s_lin	
decomposition.s_lin_f	
decomposition.s_lin_e	
decomposition.s_lin_ef	
decomposition.i_lin	
decomposition.i_lin_f	
decomposition.i_lin_e	
decomposition.i_lin_ef	

Retrieve in GUI

NODE : Decompostion > Stocastic series

forecast f series are pasted at the end of the correspoding series

- _e series are labelled with (stde)

_ef appear in graphs (but values cannot be retrieved)

IMAGE1 GUI

- image 1 : decompostion node and series
- image 2 trend graph
- image 3 seasonal graph

Retrieve in R

Biais correction voir JP (done between lin and and comp ?)

Final series

decompostion of the linearisezed series or of its log if multiplicative model

components: t_lin, s_lin, i_lin

if log, everything is dispalayed in log

Series Name	Meaning
decomposition.y_lin	
decomposition.y_lin_f	
decomposition.y_lin_ef	
decomposition.t_lin	
decomposition.t_lin_f	
decomposition.t_lin_e	
decomposition.t_lin_f	
decomposition.sa_lin	
decomposition.sa_lin_f	
decomposition.sa_lin_e	
decomposition.sa_lin_ef	
decomposition.s_lin	
decomposition.s_lin_f	
decomposition.s_lin_e	
decomposition.s_lin_ef	
decomposition.i_lin	
decomposition.i_lin_f	
decomposition.i_lin_e	
decomposition.i_lin_ef	

Retrieve in GUI

NODE : Decompostion > Stocastic series

forecast f series are pasted at the end of the corresponding series

- _e series are labelled with (stde)

_ef appear in graphs (but values cannot be retrieved)

IMAGE1 GUI

- image 1 : decompostion node and series
- image 2 trend graph
- image 3 seasonal graph

Retrieve in R

make clear reallocation of outliers effects (frequent questions)

Table

GUI

R

Output2: Final Params

what is chosen automatically or default (cf estp) Relevant if parameters not set manually, or any parameters automatically selected by the software without having a fixed default value. (The rest of the parameters is set in the spec) To manually set hose parameters and see all the fixed default values see Specifications / parameters section

here all have a default value, nothing to be automatically selected

Output3: Diagnostics

SEATS (model based) specific diagnostics check GUI (more plots, values) vs R links to methods chapter, here just be able to "read" (knowledge assumed) (here direct base on gui plots, and the add what can be retrieved in R, if cannot be retrieved in R : could it be rebuild from available elements) Obj : simple description res : current doc, jpslides, doc esp, Q JP

- WK analysis

components final estimators

- Error analysis autocorrelation of the errors (sa, trend) revisons of the errors
- Growth rates
- Model based tests

- Significant seasonality
- Stationnary variance decompostion

Fine-tuning : user defined parameters

- Prediction length

Forecast span used in the decomposition default: one year (-1) (years are set in negative values, positive values indicate nimber of periods)

- Approximation Mode

Modification type for inadmissible models None (default) Legacy Noisy

- MA unit root boundary

Modulus threshold for resetteing MA “near-unit” roots [0,1] default (0.95)

- Trend Boundary Modulus thershold for assingning positive real AR Roots [0,1] default (0.5)
- Seasonal Tolerance Degree threshold for assigning complex AR roots [0,10] default (2)
- Seasonal Boundary (unique) Modulus threshold for assigning negative real AR roots [0,1] default (0.8)
- Seasonal Boundary (unique) Same modulus threshold unique seasonal AR roots [0,1] default (0.8)
- Method

Algorithm used for estimation of unobserved componements

Burman (default)

KalmanSmoother

McEllroyMatrix

Further explanation (cf slides JP)

link to M chapter for details on root allocation

0.0.0.0.1 * Set in GUI

In specification window (link to GUI global exp)

IMAGE

0.0.0.0.2 * Set in R

version 2 (RJDemetra)

```
tramoseats_spec(
 spec = c("RSAfull", "RSA0", "RSA1", "RSA2", "RSA3", "RSA4", "RSA5"),
  fcst.horizon = NA_integer_,
  seats.predictionLength = NA_integer_,
  seats.approx = c(NA, "None", "Legacy", "Noisy"),
  seats.trendBoundary = NA_integer_,
  seats.seasdBoundary = NA_integer_,
  seats.seasdBoundary1 = NA_integer_,
  seats.seasTol = NA_integer_,
  seats.maBoundary = NA_integer_,
  seats.method = c(NA, "Burman", "KalmanSmoother", "McElroyMatrix")
)
```

in version 3 with {rjd3tramoseats}

Quality assessment for X13-Arima and Tramo-Seats

(add at begininng of chapter ? X13 an d TS as historical and more popular, ghave more built-in available features, like predifined quality assessment)

(all GUI based analysis) NODES below bechmarking

Matrix

comparaing all predefined specs available only in a formatted state in GUI (in R has to be generated by the user

Residual seasonality

list (chat is available)

visu in GUI (can be larger than what is pre-defined in R)

Residual calendar effects

Add other diagnostics node

STL

use on linearized data data, no pre-adjustment (2nd layer)

SSF

(2nd layer)

Seasonal adjustment of high-frequency data

Motivation: data specificities

Infra monthly data displays multiple and non integer periodicities which cannot be dealt with classical versions of SA algorithlms. JD+ provides tailored versions of these algorithms.

Link to worked example (Daily French births)

Available algorithms

code here and/or link to R packages chapter

Col1	Algorithm	GUI v 3.0	R package / function
Pre-treatment	Fractionnal Airline Model	yes	
Decompsition	Extended SEATS Fractionnal Airline Model	yes	
	Extended X-11	yes	
	Extended STL	no	
One-Step	SSF Framework	no	

Unobserved Components

Identifying seasonal patterns

Spectral analysis

Seasonality tests

Pre-adjustment

Calendar correction

generate

Outliers and intervention variables

add

Linearization

Decomposition

Extended X-11

Arima Model Based (AMB) Decomposition

STL decomposition

State Space framework

one step for pre-treatment and decomp

Quality assessment

Residual seasonality

Residual Calendar effects

Outlier detection

(in or outside a seasonal adjustment process)

Motivation

With Reg Arima models

Specific TERROR tool

With structural models (BSM)

Calendar and user-defined corrections

This chapter describes the generating process of calendar regressors, outliers and other input variables. The use of this variables inside a seasonal adjustment process is described in the relevant chapter on SA or on SA of HF data.

Overview of Calendar effects in JDemetra

edit : this has a evolved a lot with v3 definition possibilities via GUI and R have to be re described

The following description of the calendar effects in JDemetra+ is strictly based on PALATE, J. (2014).

A natural way for modelling calendar effects consists of distributing the days of each period into different groups. The regression variable corresponding to a type of day (a group) is simply defined by the number of days it contains for each period. Usual classifications are:

- Trading days (7 groups): each day of the week defines a group (Mondays,...,Sundays);
- Working days (2 groups): week days and weekends.

The definition of a group could involve partial days. For instance, we could consider that one half of Saturdays belong to week days and the second half to weekends.

Usually, specific holidays are handled as Sundays and they are included in the group corresponding to "non-working days". This approach assumes that the economic activity on national holidays is the same (or very close to) the level of activity that is typical for Sundays. Alternatively, specific holidays can be considered separately, e.g. by the specification that divided days into three groups:

- Working days (Mondays to Fridays, except for specific holidays),
- Non-working days (Saturdays and Sundays, except for specific holidays),
- Specific holidays.

Summary of the method used in JDemetra+ to compute trading day and working day effects

The computation of trading day and working days effects is performed in four steps:

1. Computation of the number of each weekday performed for all periods.
2. Calculation of the usual contrast variables for trading day and working day.
3. Correction of the contrast variables with specific holidays (for each holiday add +1 to the number of Sundays and subtract 1 from the number of days of the holiday). The correction is not performed if the holiday falls on a Sunday, taking into account the validity period of the holiday.
4. Correction of the constant variables for long term mean effects, > taking into account the validity period of the holiday; see below > for the different cases.

The corrections of the constant variables may receive a weight corresponding to the part of the holiday considered as a Sunday.

An example below illustrates the application of the above algorithm for the hypothetical country in which three holidays are celebrated:

- New Year (a fixed holiday, celebrated on 01 January);
- Shrove Tuesday (a moving holiday, which falls 47 days before Easter Sunday, celebrated until the end of 2012);
- Freedom day (a fixed holiday, celebrated on 25 April).

The consecutive steps in calculation of the calendar for 2012 and 2013 years are explained below.

First, the number of each day of the week in the given month is calculated as it is shown in table below.

Number of each weekday in different months

Month	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Jan-12	5	5	4	4	4	4	5
Feb-12	4	4	5	4	4	4	4
Mar-12	4	4	4	5	5	5	4
Apr-12	5	4	4	4	4	4	5
May-12	4	5	5	5	4	4	4
Jun-12	4	4	4	4	5	5	4
Jul-12	5	5	4	4	4	4	5
Aug-12	4	4	5	5	5	4	4

Month	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Sep-12	4	4	4	4	4	5	5
Oct-12	5	5	5	4	4	4	4
Nov-12	4	4	4	5	5	4	4
Dec-12	5	4	4	4	4	5	5
Jan-13	4	5	5	5	4	4	4
Feb-13	4	4	4	4	4	4	4
Mar-13	4	4	4	4	5	5	5
Apr-13	5	5	4	4	4	4	4
May-13	4	4	5	5	5	4	4
Jun-13	4	4	4	4	4	5	5
Jul-13	5	5	5	4	4	4	4
Aug-13	4	4	4	5	5	5	4
Sep-13	5	4	4	4	4	4	5
Oct-13	4	5	5	5	4	4	4
Nov-13	4	4	4	4	5	5	4
Dec-13	5	5	4	4	4	4	5

Next, the contrast variables are calculated (table below) as a result of the linear transformation applied to the variables presented in table below.

Contrast variables (series corrected for leap year effects)

Month	Mon	Tue	Wed	Thu	Fri	Sat	Length
Jan-12	0	0	-1	-1	-1	-1	0
Feb-12	0	0	1	0	0	0	0.75
Mar-12	0	0	0	1	1	1	0
Apr-12	0	-1	-1	-1	-1	-1	0
May-12	0	1	1	1	0	0	0
Jun-12	0	0	0	0	1	1	0
Jul-12	0	0	-1	-1	-1	-1	0
Aug-12	0	0	1	1	1	0	0
Sep-12	-1	-1	-1	-1	-1	0	0
Oct-12	1	1	1	0	0	0	0
Nov-12	0	0	0	1	1	0	0
Dec-12	0	-1	-1	-1	-1	0	0
Jan-13	0	1	1	1	0	0	0
Feb-13	0	0	0	0	0	0	-0.25
Mar-13	-1	-1	-1	-1	0	0	0
Apr-13	1	1	0	0	0	0	0
May-13	0	0	1	1	1	0	0

Month	Mon	Tue	Wed	Thu	Fri	Sat	Length
Jun-13	-1	-1	-1	-1	-1	0	0
Jul-13	1	1	1	0	0	0	0
Aug-13	0	0	0	1	1	1	0
Sep-13	0	-1	-1	-1	-1	-1	0
Oct-13	0	1	1	1	0	0	0
Nov-13	0	0	0	0	1	1	0
Dec-13	5	5	4	4	4	4	0

In the next step the corrections for holidays is done in the following way:

- New Year: In 2012 it falls on a Sunday. Therefore no correction is applied. In 2013 it falls on a Tuesday. Consequently, the following corrections are applied to the number of each weekday in January: Tuesday -1, Sunday +1, so the following corrections are applied to the contrast variables: -2 for Tuesday and -1 for the other contrast variables.
- Shrove Tuesday: It is a fixed day of the week holiday that always falls on Tuesday. For this reason in 2012 the following corrections are applied to the number of each weekday in February: Tuesday -1, Sunday +1, so the following corrections are applied to the contrast variables: -2 for the contrast variable associated with Tuesday, and -1 for the other contrast variables. The holiday expires at the end of 2012. Therefore no corrections are made for 2013.
- Freedom Day: In 2012 it falls on a Wednesday. Consequently, the following corrections are applied to the number of each weekday in April: Wednesday -1, Sunday +1, so the following corrections are applied to the contrast variables: -2 for Wednesday and -1 for the other contrast variables. In 2013 it falls on Thursday. Therefore, the following corrections are applied to the number of each weekday in April: Thursday -1, Sunday +1, so the following corrections are applied to the contrast variables: -2 for Thursday, and -1 for the other contrast variables.

The result of these corrections is presented in table below.

Contrast variables corrected for holidays

Month	Mon	Tue	Wed	Thu	Fri	Sat	Length
Jan-12	0	0	-1	-1	-1	-1	0
Feb-12	-1	-2	0	-1	-1	-1	0.75
Mar-12	0	0	0	1	1	1	0
Apr-12	-1	-2	-3	-2	-2	-2	0
May-12	0	1	1	1	0	0	0
Jun-12	0	0	0	0	1	1	0
Jul-12	0	0	-1	-1	-1	-1	0

Month	Mon	Tue	Wed	Thu	Fri	Sat	Length
Aug-12	0	0	1	1	1	0	0
Sep-12	-1	-1	-1	-1	-1	0	0
Oct-12	1	1	1	0	0	0	0
Nov-12	0	0	0	1	1	0	0
Dec-12	0	-1	-1	-1	-1	0	0
Jan-13	-1	-1	0	0	-1	-1	0
Feb-13	0	0	0	0	0	0	-0.25
Mar-13	-1	-1	-1	-1	0	0	0
Apr-13	0	0	-1	-2	-1	-1	0
May-13	0	0	1	1	1	0	0
Jun-13	-1	-1	-1	-1	-1	0	0
Jul-13	1	1	1	0	0	0	0
Aug-13	0	0	0	1	1	1	0
Sep-13	0	-1	-1	-1	-1	-1	0
Oct-13	0	1	1	1	0	0	0
Nov-13	0	0	0	0	1	1	0
Dec-13	0	0	-1	-1	-1	-1	0

Finally, the long term corrections are applied on each year of the validity period of the holiday.

- New Year: Correction on the contrasts: +1, to be applied to January of 2012 and 2013.
- Shrove Tuesday: It may fall either in February or in March. It will fall in March if Easter is on or after 17 April. Taking into account the theoretical distribution of Easter, it gives: prob(March) = +0.22147, prob(February) = +0.77853. The correction of the contrasts will be +1.55707 for Tuesday in February 2012 and +0.77853 for the other contrast variables. The correction of the contrasts will be +0.44293 for Tuesday in March 2012, +0.22147 for the other contrast variables.
- Freedom Day: Correction on the contrasts: +1, to be applied to April of 2012 and 2013.

The modifications due to the corrections described above are presented in table below.

Trading day variables corrected for the long term effects

Month	Mon	Tue	Wed	Thu	Fri	Sat	Length
Jan-12	1	1	0	0	0	0	0
Feb-12	-0.22115	-0.44229	0.778853	-0.22115	-0.22115	-0.22115	0.75
Mar-12	0.221147	0.442293	0.221147	1.221147	1.221147	1.221147	0
Apr-12	0	-1	-2	-1	-1	-1	0
May-12	0	1	1	1	0	0	0

Month	Mon	Tue	Wed	Thu	Fri	Sat	Length
Jun-12	0	0	0	0	1	1	0
Jul-12	0	0	-1	-1	-1	-1	0
Aug-12	0	0	1	1	1	0	0
Sep-12	-1	-1	-1	-1	-1	0	0
Oct-12	1	1	1	0	0	0	0
Nov-12	0	0	0	1	1	0	0
Dec-12	0	-1	-1	-1	-1	0	0
Jan-13	0	0	1	1	0	0	0
Feb-13	0	0	0	0	0	0	-0.25
Mar-13	-1	-1	-1	-1	0	0	0
Apr-13	1	1	0	-1	0	0	0
May-13	0	0	1	1	1	0	0
Jun-13	-1	-1	-1	-1	-1	0	0
Jul-13	1	1	1	0	0	0	0
Aug-13	0	0	0	1	1	1	0
Sep-13	0	-1	-1	-1	-1	-1	0
Oct-13	0	1	1	1	0	0	0
Nov-13	0	0	0	0	1	1	0
Dec-13	0	0	-1	-1	-1	-1	0

Mean and seasonal effects of calendar variables

The calendar effects produced by the regression variables that fulfil the definition presented above include a mean effect (i.e. an effect that is independent of the period) and a seasonal effect (i.e. an effect that is dependent of the period and on average it is equal to 0). Such an outcome is inappropriate, as in the usual decomposition of a series the mean effect should be allocated to the trend component and the fixed seasonal effect should be affected to the corresponding component. Therefore, the actual calendar effect should only contain effects that don't belong to the other components.

In the context of JDemetra+ the mean effect and the seasonal effect are long term theoretical effects rather than the effects computed on the time span of the considered series (which should be continuously revised).

The mean effect of a calendar variable is the average number of days in its group. Taking into account that one year has on average 365.25 days, the monthly mean effects for a working days are, as shown in the table below, 21.7411 for week days and 8.696 for weekends.

Monthly mean effects for the Working day variable

Groups of Working day effect	Mean effect
Week days	365.25/12*5/7 = **21.7411**
Weekends	365.25/12*2/7 = **8.696**
Total	365.25/12 = **30.4375**

The number of days by period is highly seasonal, as apart from February, the length of each month is the same every year. For this reason, any set of calendar variables will contain, at least in some variables, a significant seasonal effect, which is defined as the average number of days by period (Januaries..., first quarters...) outside the mean effect. Removing that fixed seasonal effects consists of removing for each period the long term average of days that belong to it. The calculation of a seasonal effect for the working days classification is presented in the table below.

The mean effect and the seasonal effect for the calendar periods

Period	Average number of days	Average number of week days	Mean effect	Seasonal effect
January	31	31*5/7=22.1429	21.7411	0.4018
February	28.25	28.25*5/7=20.1786	21.7411	-1.5625
March	31	31*5/7=22.1429	21.7411	0.4018
April	30	30*5/7=21.4286	21.7411	-0.3125
May	31	31*5/7=22.1429	21.7411	0.4018
June	30	30*5/7=21.4286	21.7411	-0.3125
July	31	31*5/7=22.1429	21.7411	0.4018
August	31	31*5/7=22.1429	21.7411	0.4018
September	30	30*5/7=21.4286	21.7411	-0.3125
October	31	31*5/7=22.1429	21.7411	0.4018
November	30	30*5/7=21.4286	21.7411	-0.3125
December	31	31*5/7=22.1429	21.7411	0.4018
Total	365.25	260.8929	260.8929	0

For a given time span, the actual calendar effect for week days can be easily calculated as the difference between the number of week days in a specific period and the sum of the mean effect and the seasonal effect assigned to this period, as it is shown in the table below for the period 01.2013 – 06.2013.

The calendar effect for the period 01.2013 - 06.2013

Time period (t)	Week days	Mean effect	Seasonal effect	Calendar effect
Jan-2013	23	21.7411	0.4018	0.8571

Time period (t)	Week days	Mean effect	Seasonal effect	Calendar effect
Feb-2013	20	21.7411	-1.5625	-0.1786
Mar-2013	21	21.7411	0.4018	-1.1429
Apr-2013	22	21.7411	-0.3125	0.5714
May-2013	23	21.7411	0.4018	0.8571
Jun-2013	20	21.7411	-0.3125	-1.4286
Jul-2013	23	21.7411	0.4018	0.8571

The distinction between the mean effect and the seasonal effect is usually unnecessary. Those effects can be considered together (simply called mean effects) and be computed by removing from each calendar variable its average number of days by period. These global means effect are considered in the next section.

Impact of the mean effects on the decomposition

When the ARIMA model contains a seasonal difference – something that should always happen with calendar variables – the mean effects contained in the calendar variables are automatically eliminated, so that they don't modify the estimation. The model is indeed estimated on the series/regression variables after differencing. However, they lead to a different linearised series (y_{lin}). The impact of other corrections (mean and/or fixed seasonal) on the decomposition is presented in the next paragraph. Such corrections could be obtained, for instance, by applying other solutions for the long term corrections or by computing them on the time span of the series.

Now the model with "correct" calendar effects (denoted as C), i.e. effects without mean and fixed seasonal effects, can be considered. To simplify the problem, the model has no other regression effects.

For such a model the following relations hold:

$$y_{\text{lin}} = \ y - C$$

$$T = \ F_T\left(y_{\text{lin}}\right)$$

$$S = \ F_S\left(y_{\text{lin}}\right) + C$$

$$I = \ F_I\left(y_{\text{lin}}\right)$$

where:

T - the trend;

S - the seasonal component;

I - the irregular component;

F_X - the linear filter for the component X.

Consider next other calendar effects ($\widetilde{C}$) that contain some mean (cm, integrated to the final trend) and fixed seasonal effects (cs, integrated to the final seasonal). The modified equations are now:

$$\widetilde{C} = C + cm + cs$$

$$\tilde{y}_{\text{lin}} = \ y - \widetilde{C} = \ y_{\text{lin}} - cm - cs$$

$$\widetilde{T} = \ F_T\left(\tilde{y}_{\text{lin}}\right) + cm$$

$$\widetilde{S} = \ F_S\left(\tilde{y}_{\text{lin}}\right) + C + cs$$

$$\tilde{I} = \ F_I\left(\tilde{y}_{\text{lin}}\right)$$

Taking into account that F_X is a linear transformation and that[1]

$$F_T\left(\text{cm}\right) = cm$$

$$F_T\left(\text{cs}\right) = 0$$

$$F_S\left(\text{cm}\right) = 0$$

[1] In case of SEATS the properties can be trivially derived from the matrix formulation of signal extraction. They are also valid for X-11 (additive).

$$F_S\left(\text{cs}\right) = cs$$

$$F_I\left(\text{cm}\right) = 0$$

$$F_I\left(\text{cs}\right) = 0$$

The following relationships hold:

$$\widetilde{T} = \ F_T\left(\tilde{y}_{\text{lin}}\right) + cm = F_T\left(y_{\text{lin}}\right) - cm + cm = T$$

$$\widetilde{S} = \ F_S\left(\tilde{y}_{\text{lin}}\right) + C + cs = F_S\left(y_{\text{lin}}\right) - cs + C + cs = S$$

$$\tilde{I} = \ I$$

If we don't take into account the effects and apply the same approach as in the "correct" calendar effects, we will get:

$$\breve{T} = \ F_T\left(\tilde{y}_{\text{lin}}\right) = T - cm$$

$$\breve{S} = \ F_S\left(\tilde{y}_{\text{lin}}\right) + \widetilde{C} = S + cm$$

$$\breve{I} = \ F_I\left(\tilde{y}_{\text{lin}}\right) = I$$

The trend, seasonal and seasonally adjusted series will only differ by a (usually small) constant.

In summary, the decomposition does not depend on the mean and fixed seasonal effects used for the calendar effects, provided that those effects are integrated in the corresponding final components. If these corrections are not taken into account, the main series of the decomposition will only differ by a constant.

Linear transformations of the calendar variables

As far as the RegARIMA and the TRAMO models are considered, any non-degenerated linear transformation of the calendar variables can be used. It will produce the same results (likelihood, residuals, parameters, joint effect of the calendar variables, joint F-test on the coefficients of the calendar variables…). The linearised series that will be further decomposed is invariant to any linear transformation of the calendar variables.

However, it should be mentioned that choices of calendar corrections based on the tests on the individual t statistics are dependent on the transformation, which is rather arbitrary. This is the case in old versions of TRAMO-SEATS. That is why the joint F-test (as in the version of TRAMO-SEATS implemented in TSW+) should be preferred.

An example of a linear transformation is the calculation of the contrast variables. In the case of the usual trading day variables, they are defined by the following transformation: the 6 contrast variables $(\text{No.}\,(\text{Mondays}) - No.\,(\text{Sundays})\,, \dots No.\,(\text{Saturdays}) - No.(Sundays))$ used with the length of period.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \text{Mon} \\ \text{Tue} \\ \text{Wed} \\ \text{Thu} \\ \text{Fri} \\ \text{Sat} \\ \text{Sun} \end{bmatrix} = \begin{bmatrix} Mon - Sun \\ Tue - Sun \\ Wed - Sun \\ Thu - Sun \\ Fri - Sun \\ Sat - Sun \\ \text{Length of period} \end{bmatrix}$$

For the usual working day variables, two variables are used: one contrast variable and the length of period

$$\begin{bmatrix} 1 & -\frac{5}{2} \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \text{Week} \\ \text{Weekend} \end{bmatrix} = \begin{bmatrix} \text{Contrast week} \\ \text{Length of period} \end{bmatrix}$$

The Length of period variable is defined as a deviation from the length of the month (in days) and the average month length, which is equal to 30.4375. Instead, the leap-year variable can be used here (see Regression sections in RegARIMA or Tramo)[2].

Such transformations have several advantages. They suppress from the contrast variables the mean and the seasonal effects, which are concentrated in the last variable. So, they lead to fewer correlated variables, which are more appropriate to be included in the regression model. The sum of the effects of each day of the week estimated with the trading (working) day contrast variables cancel out.

[2] GÓMEZ, V., and MARAVALL, A (2001b).

Handling of specific holidays

check vs GUI (v3) and rjd3 modelling

Three types of holidays are implemented in JDemetra+:

- Fixed days, corresponding to the fixed dates in the year (e.g. New Year, Christmas).
- Easter related days, corresponding to the days that are defined in relation to Easter (e.g. Easter +/- n days; example: Ascension, Pentecost).
- Fixed week days, corresponding to the fixed days in a given week of a given month (e.g. Labor Day celebrated in the USA on the first Monday of September).

From a conceptual point of view, specific holidays are handled in exactly the same way as the other days. It should be decided, however, to which group of days they belong. Usually they are handled as Sundays. This convention is also used in JDemetra+. Therefore, except if the holiday falls on a Sunday, the appearance of a holiday leads to correction in two groups, i.e. in the group that contains the weekday, in which holiday falls, and the group that contains the Sundays.

Country specific holidays have an impact on the mean and the seasonal effects of calendar effects. Therefore, the appropriate corrections to the number of particular days (which are usually the basis for the definition of other calendar variables) should be applied, following the kind of holidays. These corrections are applied to the period(s) that may contain the holiday. The long term corrections in JDemetra+ don't take into account the fact that some moving holidays could fall on the same day (for instance the May Day and the Ascension). However, those events are exceptional, and their impact on the final result is usually not significant.

Fixed day

The probability that the holiday falls on a given day of the week is 1/7. Therefore, the probability to have 1 day more that is treated like Sunday is 6/7. The effect on the means for the period that contains the fixed day is presented in the table below (the correction on the calendar effect has the opposite sign).

The effect of the fixed holiday on the period, in which it occurred

Sundays	Others days	Contrast variables
+ 6/7	- 1/7	1/7 - (+ 6/7)= -1

Easter related days

Easter related days always fall the same week day (denoted as Y in the table below: The effects of the Easter Sunday on the seasonal means). However, they can fall during different periods (months or quarters). Suppose that, taking into account the distribution of the dates for Easter and the fact that this holiday falls in one of two periods, the probability that Easter falls during the period m is p, which implies that the probability that it falls in the period $m+1$ is $1-p$. The effects of Easter on the seasonal means are presented in the table below.

The effects of the Easter Sunday on the seasonal means

|**Period** | **Sundays Days X Others days Contrast Y Other contrasts** |————| ———- ———— ————— —————- ——————— |m |+ p - p 0 - 2p - p |m+1 | + (1-p) - (1-p) 0 - 2×(1-p) - (1-p)

The distribution of the dates for Easter may be approximated in different ways. One of the solutions consists of using some well-known algorithms for computing Easter on a very long period. JDemetra+ provides the Meeus/Jones/Butcher's and the Ron Mallen's algorithms (they are identical till year 4100, but they slightly differ after that date). Another approach consists in deriving a raw theoretical distribution based on the definition of Easter. It is the solution used for Easter related days. It is shortly explained below.

The date of Easter in the given year is the first Sunday after the full moon (the Paschal Full Moon) following the northern hemisphere's vernal equinox. The definition is influenced by the Christian tradition, according to which the equinox is reckoned to be on 21 March[3] and the full moon is not necessarily the astronomically correct date. However, when the full moon falls on Sunday, then Easter is delayed by one week. With this definition, the date of Easter Sunday varies between 22 March and 25 April. Taking into account that an average lunar month is 29.530595 days the approximated distribution of Easter can be derived. These calculations do not take into account the actual ecclesiastical moon calendar.

For example, the probability that Easter Sunday falls on 25 March is 0.004838 and results from the facts that the probability that 25 March falls on a Sunday is 1/7 and the probability that the full moon is on 21 March, 22 March, 23 March or 24 March is 5/29.53059. The probability that Easter falls on 24 April is 0.01708 and results from the fact that the probability that 24 April is Sunday is 1/7 and takes into account that 18 April is the last acceptable date for the full moon. Therefore the probability that the full moon is on 16 April or 17 April is 1/29.53059 and the probability that the full moon is on 18 April is 1.53059/29.53059.

The approximated distribution of Easter dates

	Day	Probability
22 March	1/7 * 1/29.53059	
23 March	1/7 * 2/29.53059	

[3] In fact, astronomical observations show that the equinox occurs on 20 March in most years.

	Day	Probability
24 March	1/7 * 3/29.53059
25 March	1/7 * 4/29.53059
26 March	1/7 * 5/29.53059
27 March	1/7 * 6/29.53059
28 March	1/29.53059
29 March	1/29.53059
...	...
18 April	1/29.53059
19 April	1/7 * (6 + 1.53059)/29.53059
20 April	1/7 * (5 + 1.53059)/29.53059
21 April	1/7 * (4 + 1.53059)/29.53059
22 April	1/7 * (3 + 1.53059)/29.53059
23 April	1/7 * (2 + 1.53059)/29.53059
24 April	1/7 * (1 + 1.53059)/29.53059
25 April	1/7 * 1.53059/29.53059

Fixed week days

Fixed week days always fall on the same week day (denoted as Y in the table below) and in the same period. Their effect on the seasonal means is presented in the table below.

The effect of the fixed week holiday on the period, in which it occurred

Sundays	Day Y	Others days
+ 1	- 1	0

The impact of fixed week days on the regression variables is zero because the effect itself is compensated by the correction for the mean effect.

Holidays with a validity period

When a holiday is valid only for a given time span, JDemetra+ applies the long term mean corrections only on the corresponding period. However, those corrections are computed in the same way as in the general case.

It is important to note that using or not using mean corrections will impact in the estimation of the RegARIMA and TRAMO models. Indeed, the mean corrections do not disappear after differencing. The differences between the SA series computed with or without mean corrections will no longer be constant.

Different Kinds of calendars

see link with GUI

This scenario presents how to define different kinds of calendars. These calendars can be applied to the specifications that take into account country-specific holidays and can be used for detecting and estimating the calendar effects.

The calendar effects are those parts of the movements in the time series that are caused by different number of weekdays in calendar months (or quarters). They arise as the number of occurrences of each day of the week in a month (or a quarter) differs from year to year. These differences cause regular effects in some series. In particular, such variation is caused by a leap year effect because of an extra day inserted into February every four years. As with seasonal effects, it is desirable to estimate and remove calendar effects from the time series.

The calendar effects can be divided into a mean effect, a seasonal part and a structural part. The mean effect is independent from the period and therefore should be allocated to the trend-cycle. The seasonal part arises from the properties of the calendar that recur each year. For one thing, the number of working days of months with 31 calendar days is on average larger than that of months with 30 calendar days. This effect is part of the seasonal pattern captured by the seasonal component (with the exception of leap year effects). The structural part of the calendar effect remains to be determined by the calendar adjustment. For example, the number of working days of the same month in different years varies from year to year.

Both X-12-ARIMA/X-13ARIMA-SEATS and TRAMO/SEATS estimate calendar effects by adding some regressors to the equation estimated in the pre-processing part (RegARIMA or TRAMO, respectively). Regressors mentioned above are generated from the default calendar or the user defined calendar.

The calendars of JDemetra+ simply correspond to the usual trading days contrast variables based on the Gregorian calendar, modified to take into account some specific holidays. Those holidays are handled as "Sundays" and the variables are properly adjusted to take into account the long term mean effects.

Tests for residual trading days

We consider below tests on the seasonally adjusted series (sa_t) or on the irregular component (irr_t). When the reasoning applies on both components, we will use y_t. The functions *stdev* stands for "standard deviation" and *rms* for "root mean squares"

The tests are computed on the log-transformed components in the case of multiplicative decomposition.

TD are the usual contrasts of trading days, 6 variables (no specific calendar).

Non significant irregular

When irr_t is not significant, we don't compute the test on it, to avoid irrelevant results. We consider that irr_t is significant if $stdev(irr_t) > 0.01$ (multiplicative case) or if $stdev(irr_t)/rms(sa_t) > 0.01$ (additive case).

F test

The test is the usual joint F-test on the TD coefficients, computed on the following models:

0.0.0.0.1 * Autoregressive model (AR modelling option)

We compute by OLS:

$$y_t = \mu + \alpha y_{t-1} + \beta TD_t + \epsilon_t$$

0.0.0.0.2 * Difference model

We compute by OLS:

$$\Delta y_t - \overline{\Delta y_t} = \beta TD_t + \epsilon_t$$

So, the latter model is a restriction of the first one ($\alpha = 1, \mu = \mu = \overline{\Delta y_t}$)

The tests are the usual joint F-tests on $\beta \quad (H_0 : \beta = 0)$.

By default, we compute the tests on the 8 last years of the components, so that they might highlight moving calendar effects.

Remark:

In Tramo, a similar test is computed on the residuals of the Arima model. More exactly, the F-test is computed on $e_t = \beta TD_t + \epsilon_t$, where e_t are the one-step-ahead forecast errors.

Benchmarking and temporal disaggregation

In this chapter we describe the practical implementation, the underlying theory in a dedicated chapter.(link)

Benchmarking overview

Often one has two (or multiple) datasets of different frequency for the same target variable. Sometimes, however, these data sets are not coherent in the sense that they don't match up. Benchmarking[^1] is a method to deal with this situation. An aggregate of a higher-frequency measurement variables is not necessarily equal to the corresponding lower-frequency less-aggregated measurement. Moreover, the sources of data may have different reliability levels. Usually, less frequent data are considered more trustworthy as they are based on larger samples and compiled more precisely. The more reliable measurements, hence often the less frequent, will serve as benchmark.

In seasonal adjustment methods benchmarking is the procedure that ensures the consistency over the year between adjusted and non-seasonally adjusted data. It should be noted that the [ESS Guidelines on Seasonal Adjustment (2015)] (https://ec.europa.eu/eurostat/documents/3859598/6830795/K GQ-15-001-EN-N.pdf/d8f1e5f5-251b-4a69-93e3-079031b74bd3), do not recommend benchmarking as it introduces a bias in the seasonally adjusted data. The U.S. Census Bureau also points out that "*forcing the seasonal adjustment totals to be the same as the original series annual totals can degrade the quality of the seasonal adjustment, especially when the seasonal pattern is undergoing change. It is not natural if trading day adjustment is performed because the aggregate trading day effect over a year is variable and moderately different from zero*"[^2]. Nevertheless, some users may need that the annual totals of the seasonally adjusted series match the annual totals of the original, non-seasonally adjusted series[^3].

According to the [ESS Guidelines on Seasonal Adjustment (2015)] (https://ec.europa.eu/eurostat/documents/38 GQ-15-001-EN-N.pdf/d8f1e5f5-251b-4a69-93e3-079031b74bd3), the only benefit of this approach is that there is consistency over the year between adjusted and the non-seasonally adjusted data; this can be of particular interest when low-frequency (e.g. annual) benchmarking figures officially exist (e.g. National Accounts, Balance of Payments, External Trade, etc.) and where users' needs for time consistency are stronger.

Tools

Benchmarking with GUI

1. With the pre-defined specifcations the benchmarking functionality is not applied by default following the *ESS Guidelines on Seasonal Adjustment* (2015) recommendations. It means that once the user has seasonally adjustd the series with a pre-defined specifcation the *Benchmarking* node is empty. To execute benchmarking click on the *Specifications* button and activate the checkbox in the *Benchmarking* section.

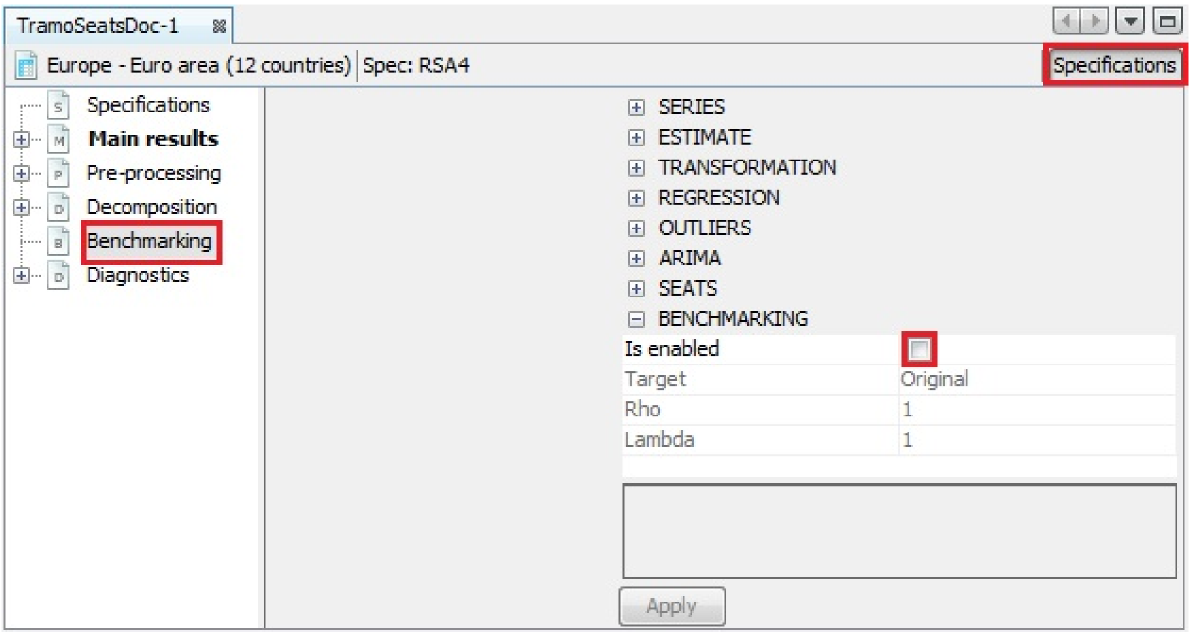

Figure 1: Text

Benchmarking option – a default view

2. Three parameters can be set here. *Target* specifies the target variable for the benchmarking procedure. It can be either the *Original* (the raw time series) or the *Calendar Adjusted* (the time series adjusted for calendar effects). *Rho* is a value of the AR(1) parameter (set between 0 and 1). By default it is set to 1. Finally, *Lambda* is a parameter that relates to the weights in the regression equation. It is typically equal to 0 (for an additive decomposition), 0.5 (for a proportional decomposition) or 1 (for a multiplicative decomposition). The default value is 1.

3. To launch the benchmarking procedure click on the **Apply** button. The results are displayed in four panels. The top-left one compares the original output from the seasonal adjustment procedure with the result from applying a benchmarking to the seasonal

adjustment. The bottom-left panel highlights the differences between these two results. The outcomes are also presented in a table in the top-right panel. The relevant statistics concerning relative differences are presented in the bottom-right panel.

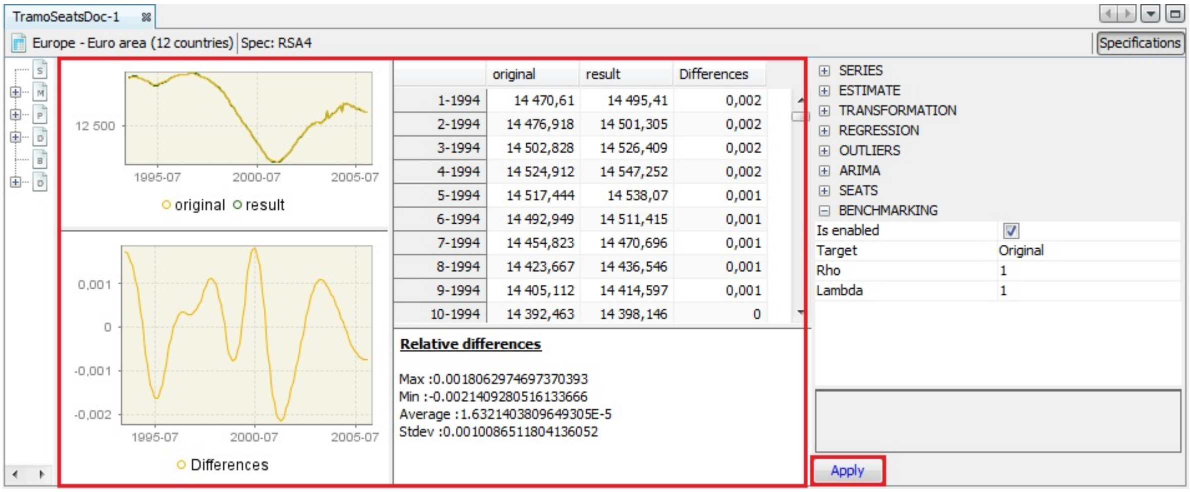

Figure 2: Text

The results of the benchmarking procedure

4. Both pictures and the table can be copied the usual way (see the *Simple seasonal adjustment of a single time series* scenario).

Options for benchmarking results

5. To export the result of the benchmarking procedure (*benchmarking.result*) and the target data (*benchmarking.target*) one needs to once execute the seasonal adjustment with benchmarking using the muli-processing option (see the *Simple seasonal adjustment of multiple time series* scenario. Once the muli-processing is executed, select the *Output* item from the *SAProcessing* menu.

The *SAProcessing* menu

6. Expand the "+" menu and choose an appropriate data format (here Excel has been chosen). It is possible to save the results in TXT, XLS, CSV, and CSV matrix formats. Note that the available content of the output depends on the output type.

Exporting data to an Excel file

7. Chose the output items that refer to the results from the benchmarking procedure, move them to the window on the right and click **OK**.

Exporting the results of the benchmarking procedure

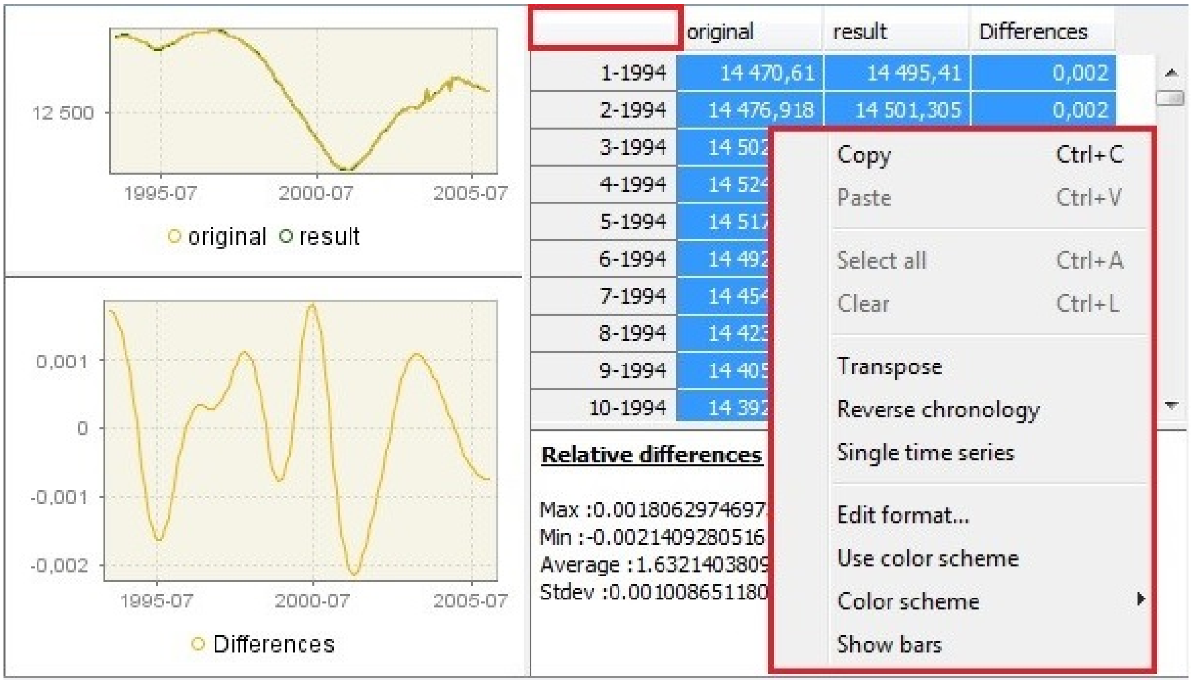

Figure 3: Text

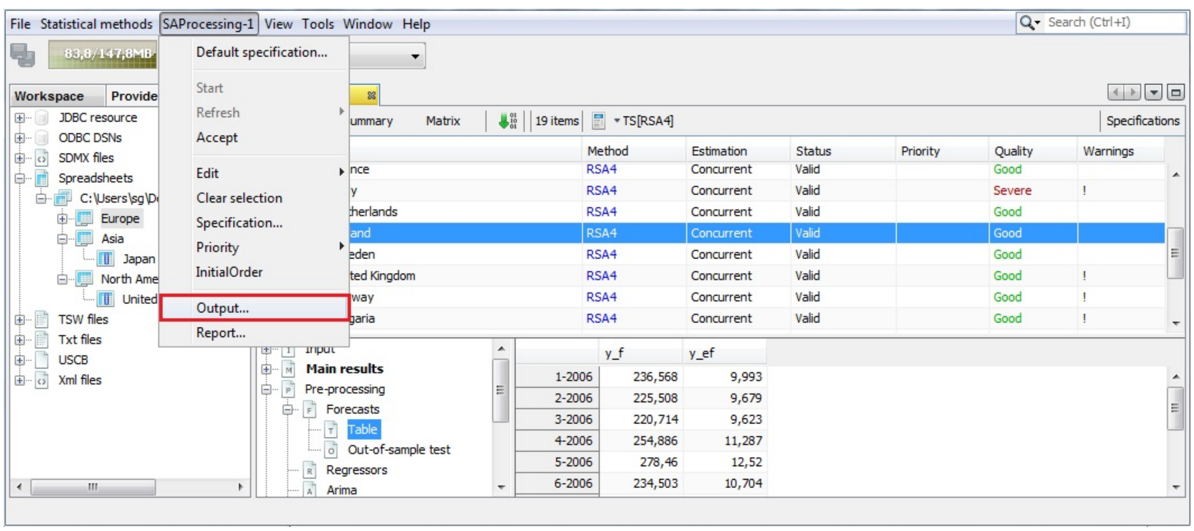

Figure 4: Text

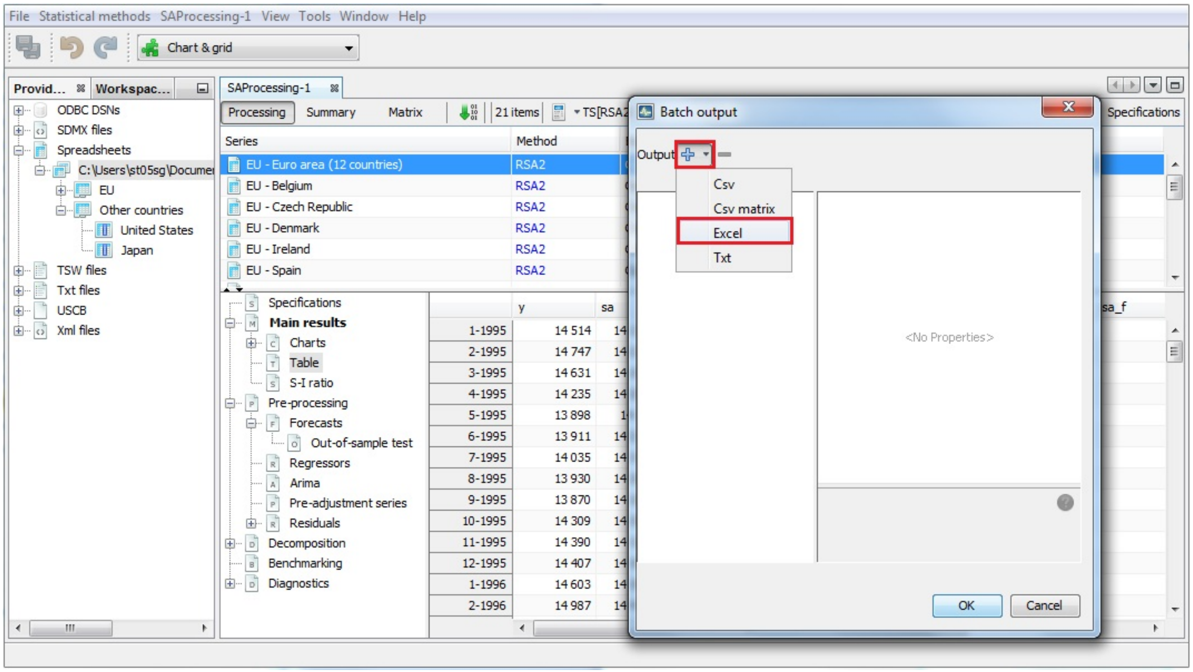

Figure 5: Text

Benchmarking in R

package rjd3bench orga doc - here - in package - example

References

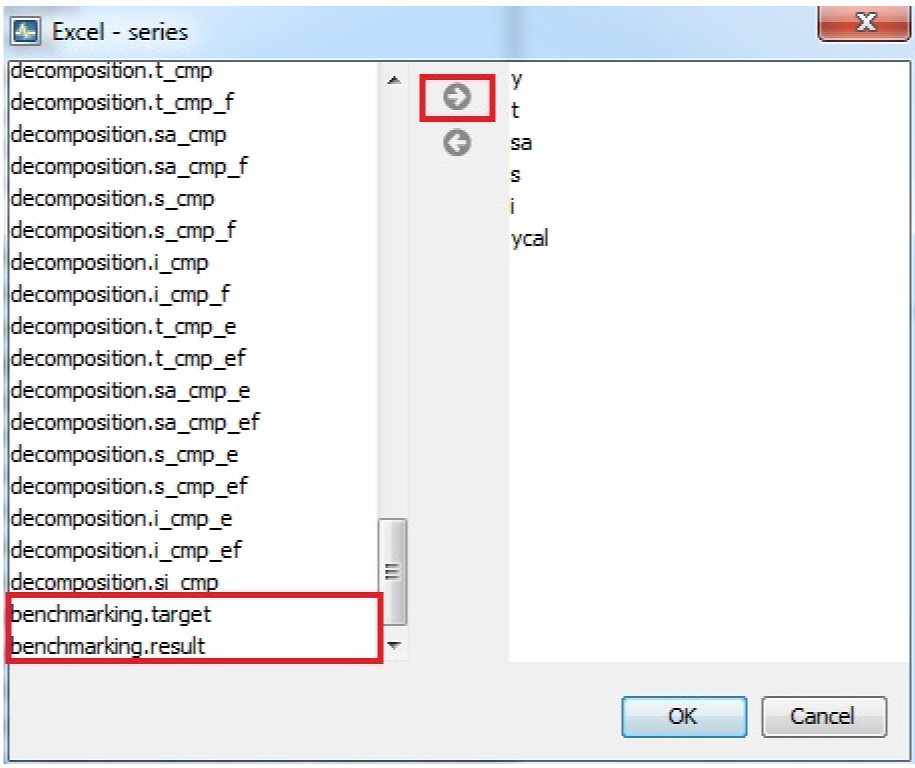

Figure 6: Text

Trend-cycle estimation

Motivation

Estimation Methods

Tools

rjd3 highfeq package

rjdfilters package

Nowcasting

Motivation

Underlying Theory: references ?

Tools

- plug in ?
- R package ?

Part II

Tools

Graphical User Interface

Overview

General info about using the interface, specific info related to an algorithm (X13-Arima, Tramo-seats, bechmarking) is displayed in the relevant chapters.

Warning: info from v2 family, will be progressively switched to v3 Internal links might not all be active

Missing:

- NEXT : Tools menu (galbal descp + disptach to specific parts by function)
- workspace window details
- output items

Add: all generic operations (like in data v, single time series, context menus)

Quick links (in appendicies)

- import data
- run automatic sa (cf current doc)

Installation Procedure

JDemetra+ is a stand-alone application packed in a zip package. To run JDemetra+ the Java RE 8 or higher is needed. Java RE can be downloaded from Oracle website.

The official release of JDemetra+ is accessible at a dedicated Github page. The site presents all available releases - both official releases (labelled in green as latest releases) and pre-releases (labelled in red) - packed in zip packages. From the *Latest release* section either choose the installer appropriate for your operating system (Windows, Linux, Mac OS, Solaris) or take the portable zip-file. The installation process is straightforward and intuitive. For example, when the zip-file is chosen and downloaded, then under Windows OS the application can be found in the “bin”-folder of the installation/unpacked zip. To open an application, double click on nbdemetra.exe or nbdemetra64.exe depending on the system version (nbdemetra.exe for the 32-bit system version and nbdemetra64.exe for the 64-bit system version).

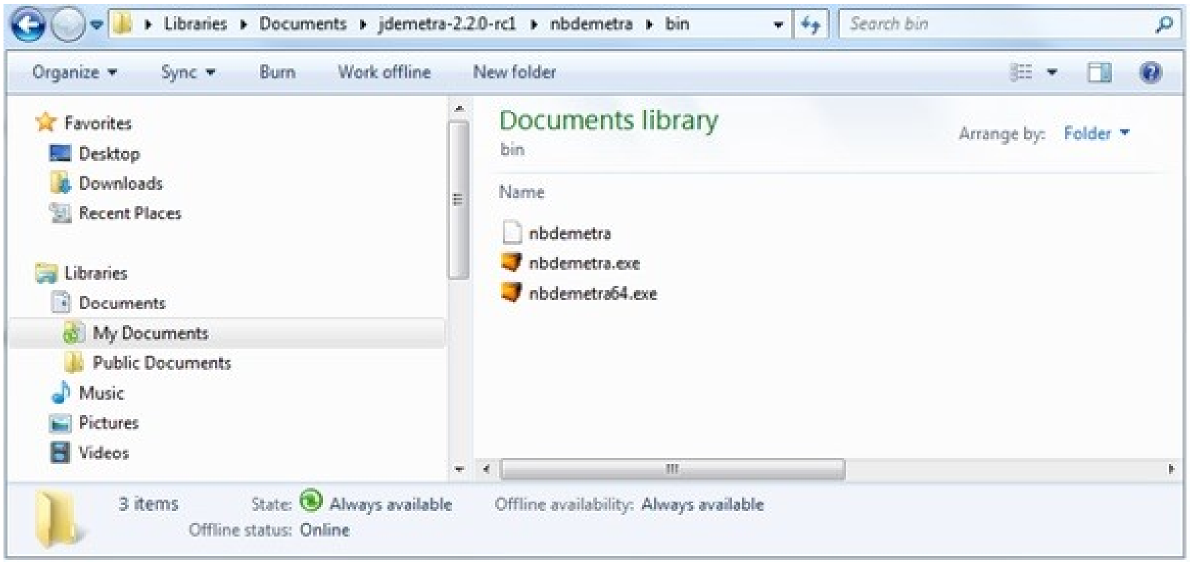

Figure 7: Text

Launching JDemetra+

If the launching of JDemetra+ fails, you can try the following operations:

- Check if Java SE Runtime Environment (JRE) is properly installed by typing in the following command in a terminal: *java –version*
- Check the logs in your home directory:
  - %appdata%/.nbdemetra/dev/var/log/ for Windows;
  - ~/.nbdemetra/dev/var/log/ for Linux and Solaris;
  - ~/Library/Application Support/.nbdemetra/dev/var/log/ for Mac OS X.

In order to remove a previously installed JDemetra+ version, the user should delete an appropriate JDemetra+ folder.

Running JDemetra+

To open an application, navigate to the destination folder and double click on *nbdemetra.exe* or *nbdemetra64.exe* depending on the system version (*nbdemetra.exe* for the 32-bit system version and *nbdemetra64.exe* for the 64-bit system version).

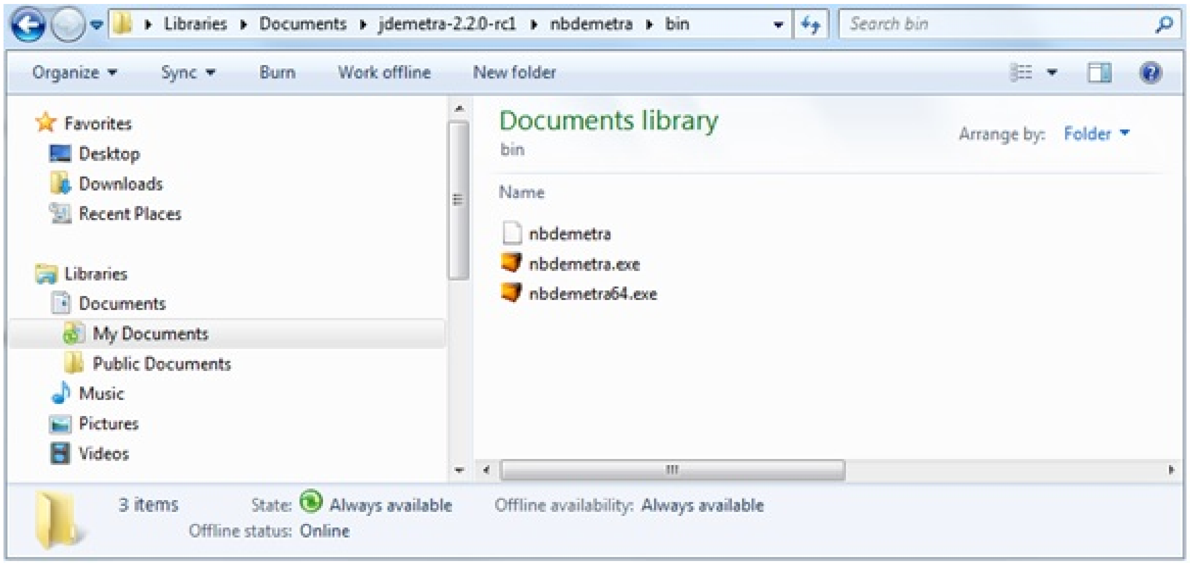

Figure 8: Text

Running JDemetra+

Closing JDemetra+

To close the application, select *File* → *Exit* from the File menu.

Closing JDemetra+

The other way is to click on the close box in the upper right-hand corner of the JDemetra+ window. If there is any unsaved work, JDemetra+ will display a warning and provide the user with the opportunity to save it. The message box is shown below.

The warning from leaving JDemetra+ without saving the workspace

BELOW: menus and functions common to all the algos

Interface Starting Winwow

The default view of the JDemetra+ window, which is displayed after launching the program, is shown below.

JDemetra+ default window

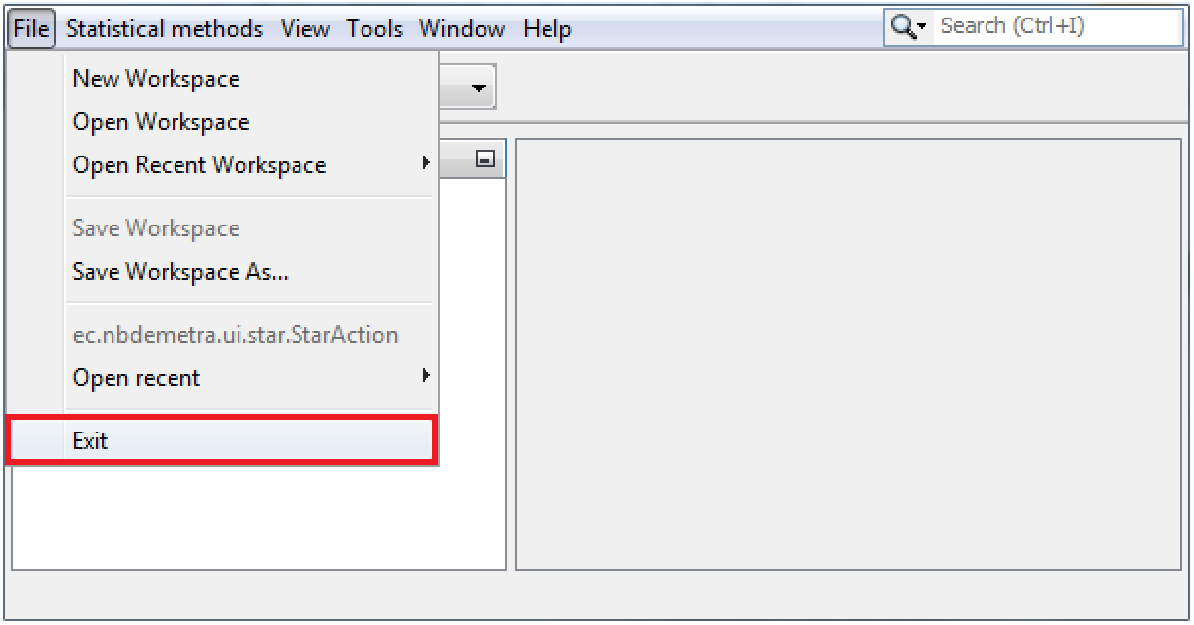

Figure 9: Text

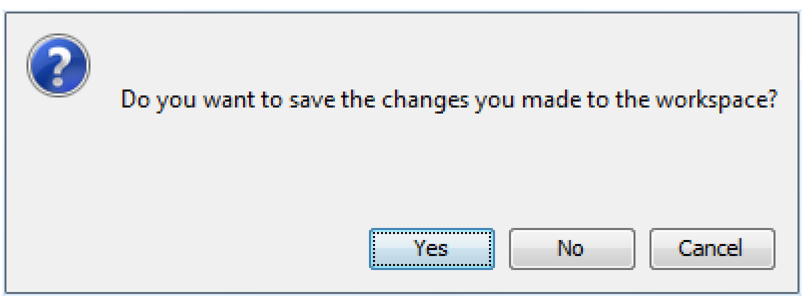

Figure 10: Text