# JDemetra+ online documentation

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

12/22/22

# Table of contents

# Preface

Welcome to the JDemetra+ online documentation.

JDemetra+ is an open-source software for seasonal adjustment and time series analysis, 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 and Insee.

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

This website is under construction, in the meantime you can fill a large number of the gaps by referring to the old online documentation available here.

# JDemetra+

## Introduction

JDemetra+ is an open source software for seasonal adjustment and time series analysis. It has been officially recommended by Eurostat to the European Statistical System members since 2015. It is unique in its combination of very fast Java routines, a graphical user interface and a family of R packages. The graphical interface offers a structured and visual feedback, suitable for refined analysis and training, whereas R tools allow the user to mix the capabilities of JDemetra+ with the versatility of the R world, be it for mathematical functions or data wrangling.

.

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

## Version 2.x and version 3

3.0 to be released in December 2022, fills several critical gaps in the tool box of a time series analyst providing extended features for seasonal adjustment and trend estimation, including high frequency data and production tools.

## Main functions

Core JDemetra (java) algorithms can be accessied via several tools: - Graphical User Interface GUI - …enhanced with additionnal plug-ins - `R` packages

### Seasonal adjustment algorithms

All are available for low and high frequency data.

| Algorithms | Access | Key features |
|---|---|---|

X13-Arima

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

### 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)

add 1 or 2 chpters of quick launch

## 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 periodic 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)

### Starting 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)

### Starting 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 span in GUI

Use the specification window for a given series (link to details) and expand the nodes (add access of specification window)

Setting span 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
```

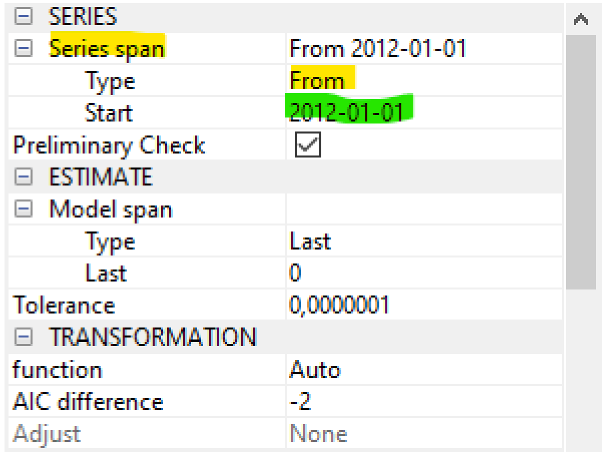



Figure 1: Setting series span

```
#
# 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)
```

#### Setting the model span

Setting works the same way as settinf series (estimation) span described above. But here, the user can specify the span (data interval) of the time series to be used for the estimation of the RegARIMA model coefficients. It allows to impede a chosen part of the data from influencing the regression estimates.

Additionnal (vs series span setting) parameters are described below:

| | |
|---|---|
| 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. |

Setting in GUI:

Use the specification window

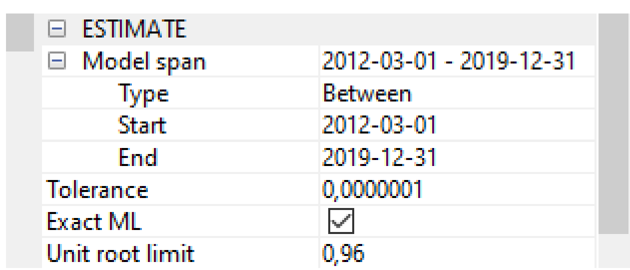



Figure 2: Model span setting

Setting in R

Specific settingcnot possible (to confirm), estimation span and model span must be equal. So additional Tramo parameters for model span are set on the estimation span.

Tramo example: version 2

```
#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)
```

version 3

### 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 in GUI

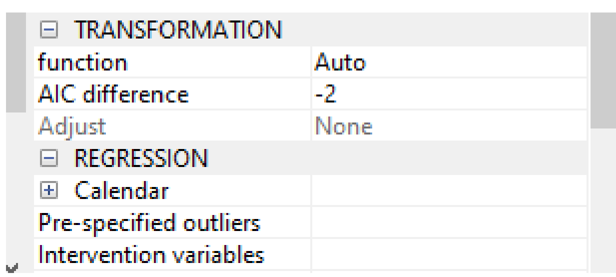



Figure 3: Model span setting

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", # choose from: 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 described above (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 Retrieving parameters which are in use

Automatically chosen or user-defined calendar options (as well as other pre-adjustment options) are displayed at th top of the MAIN Results NODE

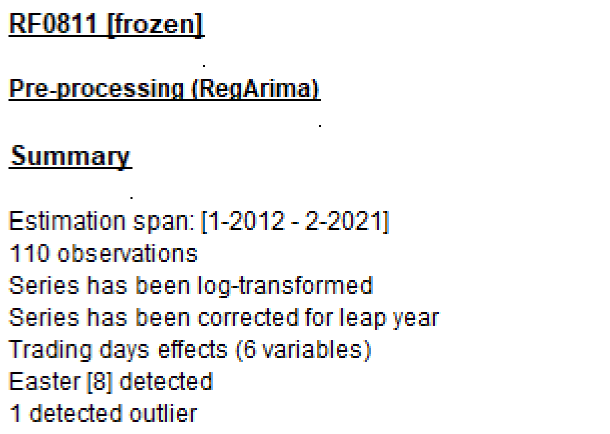
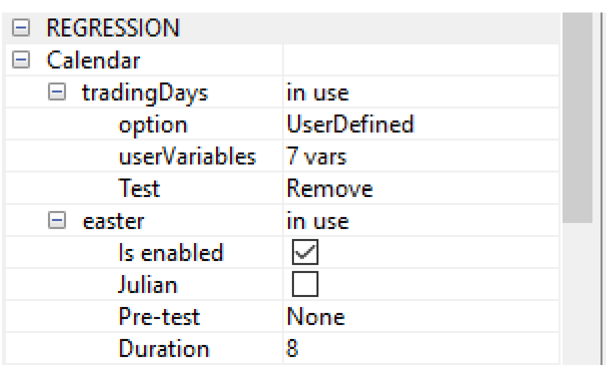



Figure 4: Text

Details of regression results are displayed in the pre processing panel

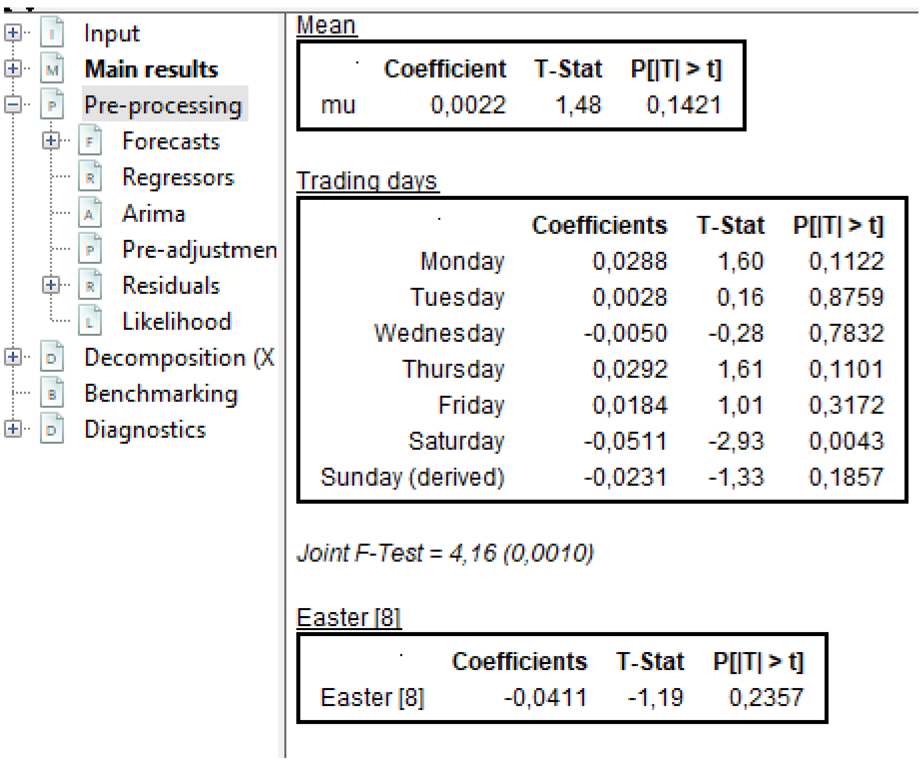
##### 0.0.0.0.2 * GUI Setting options

Use the specification window




Detils on options; default an user defined importing user defined regressors

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




Mean

| | Coefficient | T-Stat | P[\|T\| > t] |
|---|---|---|---|
| mu | 0,0022 | 1,48 | 0,1421 |

Trading days

| | Coefficients | T-Stat | P[\|T\| > t] |
|---|---|---|---|
| Monday | 0,0288 | 1,60 | 0,1122 |
| Tuesday | 0,0028 | 0,16 | 0,8759 |
| Wednesday | -0,0050 | -0,28 | 0,7832 |
| Thursday | 0,0292 | 1,61 | 0,1101 |
| Friday | 0,0184 | 1,01 | 0,3172 |
| Saturday | -0,0511 | -2,93 | 0,0043 |
| Sunday (derived) | -0,0231 | -1,33 | 0,1857 |

*Joint F-Test = 4,16 (0,0010)*

Easter [8]

| | Coefficients | T-Stat | P[\|T\| > t] |
|---|---|---|---|
| Easter [8] | -0,0411 | -1,19 | 0,2357 |

Figure 5: Text

### 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 / holidays / 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: RJDemetra

```
# 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 explains how to **use** X-11 decomposition algorithm, via R as well as via GUI. The algorithm itself is explained in more details here

In a nutshell, X-11 will de deompose the **linearized series** using iteratively different moving averages iteratively. The effects of pre-treatment will be reallocated at the end.

## Default specifications

The default specifications for X-11 must be chosen at the starting of the SA processing. They are detailed in the RegArima part Note: X-11 can be run without pre-treatment

### Quick Launch

#### From GUI

With - a workspace open - an SA pocessing created an open data - choose a default specification - drop your data and press green arrow (see old doc?, gif demo?)

#### In R

in version 2

```
library(RJDemetra)
model_sa<-x13(serie_brute, spec ="RSA5c")
```

The model_sa object contains all results and will be progressively detai below. Its global structure is outlined here.

in version 3

### Output 1: series

(X11 just final series or intermediate as well ? homogenise with seats ?)

| Series | Final X11 name | Final Results | Reallocation of pre-adjustment effects |
|---|---|---|---|
| Raw series (forecasts) | | y (y_f) | |
| Linearized series | B1 | | none |
| 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)

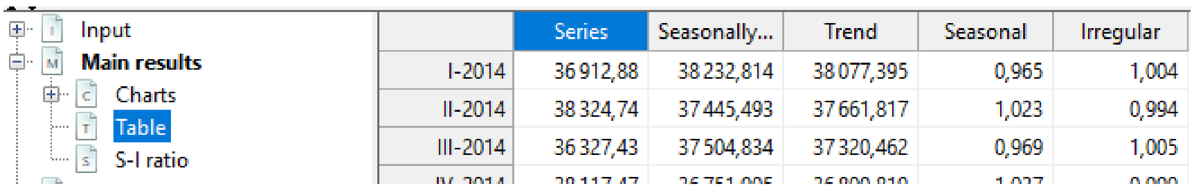
Here we focus on the final components main series (= 2nd part of D tables, outliers effects added)

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

### Display in GUI

homogeneiser avec seats

### Retrieve in GUI

Main results 


(forecasts glued, values in *italic*)

X-11 Tables

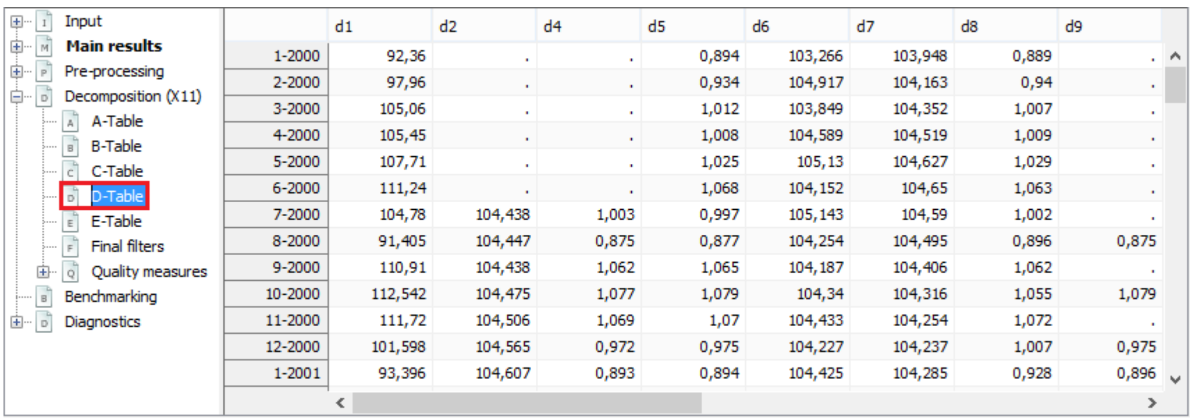



Figure 6: Text

Output series can be exported out if GUI by two means: - generating output files from GUI as explained here - running the cruncher to genereate those files as described here

### Retrieve in R

in version 2

```
# final components
model_sa$final$series
```

```
# their forecasts y_f sa_f s_f t_f i_f
model_sa$final$forecasts
# from user defined output
```

in version 3

## Output 2: diagnostics

X11 produces the following type diagnostics or quality measures

### SI-ratios

It is a plot, see below

Description, and computation details

#### 0.0.0.0.1 * Retrieve in GUI

NODE Main Results > SI-Ratios SA_MainResults_SI_ratios.png

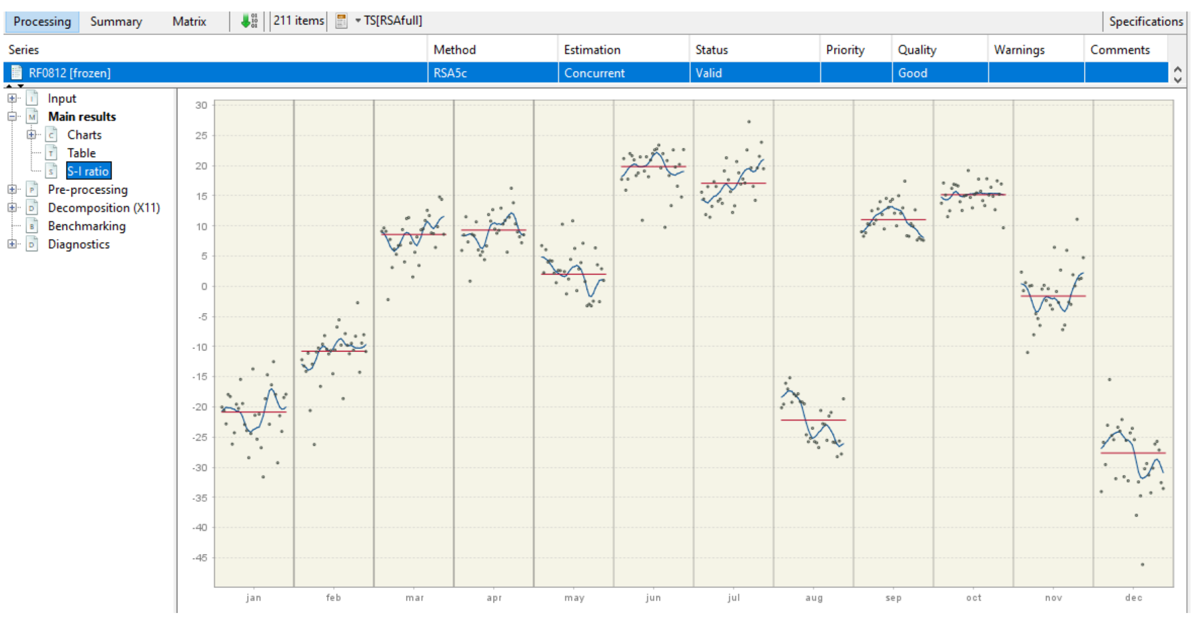



Figure 7: Text

In GUI all values cannot be retrieved

#### 0.0.0.0.2 * Retrieve in R

in version 2

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

in version 3

### 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

add composite indicators \(Q\) and \(Q - M2.\)

The $Q$ statistic is a composite indicator calculated from the $M$ statistics.

$$Q = \frac{10M1 + 11M2 + 10M3 + 8M4 + 11M5 + 10M6 + 18M7 + 7M8 + 7M9 + 4M10 + 4M11}{100}$$

$Q = Q - M2$ (also called $Q2$) is the $Q$ statistic for which the $M2$ statistics was excluded from the formula, i.e.:

$$Q - M2 = \frac{10M1 + 10M3 + 8M4 + 11M5 + 10M6 + 18M7 + 7M8 + 7M9 + 4M10 + 4M11}{89}$$

If a time series does not cover at least 6 years, the $M8$, $M9$, $M10$ and $M11$ statistics cannot be calculated. In this case the $Q$ statistic is computed as:

$$Q = \frac{14M1 + 15M2 + 10M3 + 8M4 + 11M5 + 10M6 + 32M7}{100}$$

The model has a satisfactory quality if the $Q$ statistic is lower than 1.

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

##### 0.0.0.0.1 * Retrieve in GUI

To display results in GUI, expande NODE Decompostion(X-11) > Quality Measures > Summary

Results displayed in red indicate that the test failed.

**Monitoring and Quality Assessment Statistics**

| | | |
|---|---|---|
| M-1 | 0.547 | The relative contribution of the irregular over three months span |
| M-2 | 0.020 | The relative contribution of the irregular component to the stationary portion of the variance |
| M-3 | 0.000 | The amount of period to period change in the irregular component as compared to the amount of period to period change in the trend-cycle |
| M-4 | 1.744 | The amount of autocorrelation in the irregular as described by the average duration of run |
| M-5 | 0.152 | The number of periods it takes the change in the trend- cycle to surpass the amount of change in the irregular |
| M-6 | 0.209 | The amount of year to year change in the irregular as compared to the amount of year to year change in the seasonal |
| M-7 | 1.379 | The amount of moving seasonality present relative to the amount of stable seasonality |
| M-8 | 1.743 | The size of the fluctuations in the seasonal component throughout the whole series |
| M-9 | 0.339 | The average linear movement in the seasonal component throughout the whole series |
| M-10 | 2.986 | The size of the fluctuations in the seasonal component in the recent years |
| M-11 | 2.891 | The average linear movement in the seasonal component in the recent years |
| Q | 0.863 | |
| Q-m2 | 0.967 | |

Figure 8: Text

##### 0.0.0.0.2 * Retrieve in R

in version 2

```
# this code snippet is not self-sufficient
# shpould it be
model_sa$decomposition$mstats
```

in version 3

### Detailed Quality measures

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

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

If a stat can be retrieved in R, the code example is in the corresponding paragraph, otherwise it means that these stats are not directly available and must be computed by the user if needed.

**0.0.0.0.1 *** Average percent change (or Average differences) without regard to sign over the indicated span

(here clarify notations) (link to tables details in meth chapter or appendix) (here A, D, E are needed) (link to Ladiray-Quenneville)

The first table presents the average percent change without regard to sign of the percent changes (multiplicative model) or average differences (additive model) over several periods (from 1 to 12 for a monthly series, from 1 to 4 for a quarterly series) for the following series:

- $O$ – Original series (Table A1);
- CI – Final seasonally adjusted series (Table D11);
- $I$ – Final irregular component (Table D13);
- $C$ – Final trend (Table D12);
- $S$ – Final seasonal factors (Table D10);
- $P$ – Preliminary adjustment coefficients, i.e. regressors estimated by the RegARIMA model (Table A2);
- $TD\&H$ – Final calendar component (Tables A6 and A7);
- Mod.O – Original series adjusted for extreme values (Table E1);
- Mod.CI – Final seasonally adjusted series corrected for extreme values (Table E2);
- Mod.I – Final irregular component adjusted for extreme values (Table E3).

In the case of an additive decomposition, for each component the average absolute changes over several periods are calculated as:

$$\text{Component}_d = \frac{1}{n-d}\sum_{t=d+1}^{n} |Table_t - Table_{t-d}|$$

where:

$d$ – time lag in periods (from a monthly time series $d$ varies from to 4 or from 1 to 12);

$n$ – total number of observations per period;

Component – the name of the component;

Table – the name of the table that corresponds to the component.

Average percent change without regard to sign over the indicated span

| Span | O | CI | I | C | S | P | TD&H | Mod.O | Mod.CI | Mod.I |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 7,50 | 3,81 | 3,49 | 1,42 | 6,99 | 0,00 | 0,00 | 7,75 | 3,57 | 3,29 |
| 2 | 5,33 | 4,88 | 3,90 | 2,88 | 3,57 | 0,00 | 0,00 | 5,40 | 4,61 | 3,55 |
| 3 | 8,23 | 5,75 | 3,74 | 4,39 | 7,16 | 0,00 | 0,00 | 8,53 | 5,50 | 3,39 |
| 4 | 6,36 | 6,75 | 3,76 | 5,94 | 0,00 | 0,00 | 0,00 | 6,74 | 6,74 | 3,56 |

Figure 9: Text

For the multiplicative decomposition the following formula is used:

$$\text{Component}_d = \frac{1}{n-d} \sum_{t=d+1}^{n} \left|\frac{\text{Table}_t}{\text{Table}_{t-d}} - 1\right|$$

.

(add image of result table for multiplictive model)

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

(homogenize titles: “variance”) (one formula is enough)

Next, Table F2B of relative contributions of the different components to the differences (additive model) or percent changes (multiplicative model) in the original series is displayed. They express the relative importance of the changes in each component. Assuming that the components are independent, the following relation is valid:

$$O_d^2 \approx C_d^2 + S_d^2 + I_d^2 + P_d^2 + TD\&H_d^2$$

In order to simplify the analysis, the approximation can be replaced by the following equation:

$$O_d^{*2} = C_d^2 + S_d^2 + I_d^2 + P_d^2 + TD\&H_d^2$$

The notation is the same as for Table F2A. The column Total denotes total changes in the raw time series.

Data presented in Table F2B indicate the relative contribution of each component to the percent changes (differences) in the original series over each span, and are calculated as: $\frac{I_d^2}{O_d^{*2}}$, $\frac{C_d^2}{O_d^{*2}}$, $\frac{S_d^2}{O_d^{*2}}$, $\frac{P_d^2}{O_d^{*2}}$ and $\frac{TD\&H_d^2}{O_d^{*2}}$ where: $O_d^{*2} = I_d^2 + C_d^2 + S_d^2 + P_d^2 + TD\&H_d^2$.

The last column presents the *Ratio* calculated as: $100 \times \frac{O_d^{*2}}{O_d^2}$, which is an indicator of how well the approximation $\left(O_d^*\right)^2 \approx O_d^2$ holds.

Relative contributions to the variance of the percent change in the components of the original series

| Span | I | C | S | P | TD&H | Total | Ratio |
|---|---|---|---|---|---|---|---|
| 1 | 17,53 | 3,27 | 79,20 | 0,00 | 0,00 | 100,00 | 102,79 |
| 2 | 37,38 | 24,71 | 37,91 | 0,00 | 0,00 | 100,00 | 115,35 |
| 3 | 13,97 | 23,47 | 62,56 | 0,00 | 0,00 | 100,00 | 112,79 |
| 4 | 26,47 | 73,53 | 0,00 | 0,00 | 0,00 | 100,00 | 105,49 |

Figure 10: Text

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

When an additive decomposition is used, Table F2C presents the average and standard deviation of changes calculated for each time lag $d$, taking into consideration the sign of the changes of the raw series and its components. In case of a multiplicative decomposition the respective table shows the average percent differences and related standard deviations.

Average percent change with regard to sign and standard deviation over indicated span

| | O | | I | | C | | S | | CI | |
|---|---|---|---|---|---|---|---|---|---|---|
| Span | Avg | S.D. | Avg | S.D. | Avg | S.D. | Avg | S.D. | Avg | S.D. |
| 1 | 1,97 | 8,67 | 0,05 | 3,73 | 1,41 | 0,48 | 0,53 | 8,01 | 1,46 | 3,81 |
| 2 | 3,19 | 5,72 | 0,15 | 4,48 | 2,86 | 0,97 | 0,24 | 4,42 | 3,02 | 4,72 |
| 3 | 4,97 | 9,47 | 0,09 | 4,52 | 4,36 | 1,44 | 0,55 | 8,30 | 4,46 | 4,97 |
| 4 | 5,93 | 3,81 | 0,10 | 4,32 | 5,90 | 1,90 | 0,00 | 0,00 | 6,01 | 5,06 |

Figure 11: Text

**0.0.0.0.4 *** Average duration of run

Average duration of run is an average number of consecutive monthly (or quarterly) changes in the same direction (no change is counted as a change in the same direction as the preceding change). JDemetra+ displays this indicator for the seasonally adjusted series, for the trend and for the irregular component.

**Average duration of run.**

| | |
|---|---|
| CI | 8.44 |
| I | 1.31 |
| C | 15.20 |

Figure 12: Text

#### 0.0.0.0.5 * I/C ratio over indicated span and global

The $\frac{I}{C}$ ratios for each value of time lag $d$, presented in Table F2E, are computed on a basis of the data in Table F2A. Global IC is displayed below the table

**I/C Ratio for indicated span.**

| | |
|---|---|
| **1** | **0.150** |
| 2 | 0.052 |
| 3 | 0.039 |
| 4 | 0.031 |

**I/C Ratio:** 0.314

Figure 13: Text

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

The relative contribution of components to the variance of the stationary part of the original series is calculated for the irregular component ($I$), trend made stationary ($C$), seasonal component ($S$) and calendar effects (TD&H).

The trend is made staionnary by by extracting a linear trend from the trend component presented in Table D12.

**Relative contribution of the components to the stationary portion of the variance in the original series.**

| | |
|---|---|
| I | 0.01 |
| C | 99.56 |
| S | 0.15 |
| P | 0.00 |
| TD&H | 0.00 |
| Total | 99.72 |

Figure 14: Text

#### 0.0.0.0.7 * Autocorrelations in the irregular

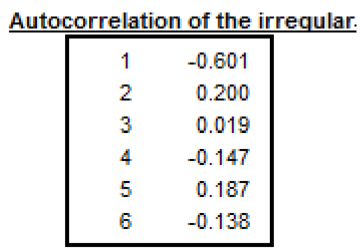
The last table shows the autocorrelogram of the irregular component from Table D13. In the case of multiplicative decomposition it is calculated for time lags between 1 and the number of periods per year +2 using the formula:

$$\mathrm{Corr}_k I = \frac{\sum_{t=k+1}^{N} (I_t - 1)(I_{t-k} - 1)}{\sum_{t=1}^{N} (I_t - 1)^2}$$

where $N$ is number of observations in the time series and $k$ the lag.

For the additive decomposition the formula is:

$$Corr_k I_t = \frac{\sum_{t=k+1}^{N} (I_t \times I_{t-k})}{\sum_{t=1}^{N} (I_t)^2}$$



Autocorrelation of the irregular.

| | |
|---|---|
| 1 | -0.601 |
| 2 | 0.200 |
| 3 | 0.019 |
| 4 | -0.147 |
| 5 | 0.187 |
| 6 | -0.138 |

Figure 15: Text

#### 0.0.0.0.8 * Heteroskedasticity

Cochran test on equal variances within each period

The Cochran test is design to identify the heterogeneity of a series of variances. X-13-ARIMA-SEATS uses this test in the extreme value detection procedure to check if the irregular component is heteroskedastic. In this procedure the standard errors of the irregular component are used for an identification of extreme values. If the null hypothesis that for all the periods (months, quarters) the variances of the irregular component are identical is rejected, the standard errors will be computed separately for each period (in case the option *Calendarsigma*=**signif** has been selected).

#### 0.0.0.0.9 * Moving seasonality ratios (MSR)

For each $i^{\text{th}}$ month we will be looking at the mean annual changes for each component by calculating:

**Heteroskedasticity (Cochran test on equal variances within each period)**

| Test statistic | Critical value (5% level) | Decision |
|---|---|---|
| 0.1303 | 0.15 | Null hypothesis is not rejected. |

Figure 16: Text

$$\overline{S}_i = \frac{1}{N_i - 1} \sum_{t=2}^{N_i} |S_{i,t} - S_{i,t-1}|$$

and

$$\overline{I}_i = \frac{1}{N_i - 1} \sum_{t=2}^{N_i} |I_{i,t} - I_{i,t-1}|$$

,

where $N_i$ refers to the number of months i in the data, and the moving seasonality ratio of month $i$:

$$MSR_i = \frac{\overline{I}_i}{\overline{S}_i}$$

The Moving Seasonality Ratio (MSR) is used to measure the amount of noise in the Seasonal-Irregular component. By studying these values, the user can select for each period the seasonal filter that is the most suitable given the noisiness of the series.

**Moving Seasonality Ratios (MSR)**

| Period | I | S | MSR |
|---|---|---|---|
| 1 | 0.0597 | 0.0211 | 2.8292 |
| 2 | 0.0808 | 0.0135 | 5.9850 |
| 3 | 0.0767 | 0.0139 | 5.5038 |
| 4 | 0.0777 | 0.0262 | 2.9640 |

Figure 17: Text

### Output 3: 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 those 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 in GUI

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

#### Retrieve in R

in version 2

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

in version 3

### 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 $3 \times 3$
- choices : $3 \times 1$, $3 \times 3$, $3 \times 5$, $3 \times 9$, $3 \times 15$ 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, adding links to relevant method chapter)

SEATS (Signal Extraction for Arima Time Series) algorithm will decompose the linearized series, in level or in logarithm, using the Arima model fitted in the pre-treatment phase (link). The sections below describe - the specifications needed to run SEATS - the generated output - series - diagnostics - final parameters - the available user-defined parameters

### Default specifications

The default specifications for SEATS must be chosen at the starting of the SA processing. They are detailed in the RegArima part Starting point for Tramo-Seats

### Quick Launch

link from introduction 3 steps - import data - chose spec - retrieve results (GUi, R , cruncher) overall procedure

in GUI

in R in version 2 in version 3

### Global outline in GUI

2 places for each series - decomp node for output, diagnostics and (some) final parameters

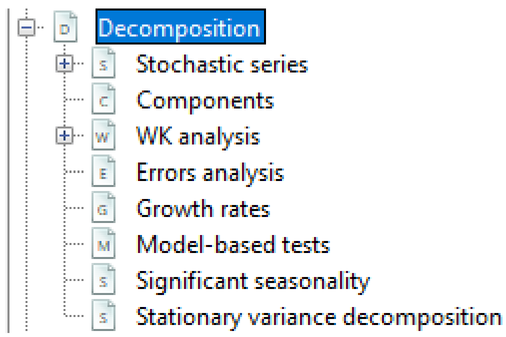



Figure 18: Text

This node will be detailed in the sections below.

- specification window for customizable finl parameters access

### Global outline in R

??

## Output 1: series

### Stochastic series

Decompostion of the linearized series or of its logarithm in case of a multiplicative model, in this case the series are displayed in log.

y_lin is split into components: t_lin, s_lin, i_lin

suffixes: - _f stands for forecast - _e stands for - _ef stands for

#### 0.0.0.0.1 * Display in GUI

NODE Decompostion>Stochastic series - Table with series and its standard error image

- Plot of Trend with confidence interval image
- Plot of Seasonal component with confidence interval image

#### 0.0.0.0.2 * Retrieve from GUI

Generating outout from GUI (link) or from Cruncher (link), stochastic series, their standard errors, forecasts and forecasts errors can be acessed with the following names

| 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 | |

#### 0.0.0.0.3 * Retrieve in R

version 2 version 3

### Components (Level)

Decompostion of the linearized series, back to level in case of a multiplicative model.

y_lin is split into components: t_lin, s_lin, i_lin

suffixes: - _f stands for forecast - _e stands for - _ef stands for

#### 0.0.0.0.1 * Displayed in GUI

NODE Decompostion>Components - Table with series and its standard error image

#### 0.0.0.0.2 * Retrieve from GUI

Generating outout from GUI (link) or from Cruncher (link), component series, their standard errors, forecasts and forecasts errors can be acessed with the following names

| Series Name | Meaning |
|---|---|
| decomposition.y_cmp | |
| decomposition.y_cmp_f | |
| decomposition.y_cmp_ef | |
| decomposition.t_cmp | |
| decomposition.t_cmp_f | |
| decomposition.t_cmp_e | |
| decomposition.t_cmp_f | |
| decomposition.sa_cmp | |
| decomposition.sa_cmp_f | |
| decomposition.sa_cmp_e | |
| decomposition.sa_cmp_ef | |
| decomposition.s_cmp | |
| decomposition.s_cmp_f | |
| decomposition.s_cmp_e | |
| decomposition.s_cmp_ef | |
| decomposition.i_cmp | |
| decomposition.i_cmp_f | |
| decomposition.i_cmp_e | |
| decomposition.i_cmp_ef | |

#### 0.0.0.0.3 * Retrieve in R

version 2 version 3

#### 0.0.0.0.4 * Bias correction

### Final series

pb : qu'est ce que la sa de de cmp vs la SA finale ?

| Series | Final SEATS components | Final Results | Reallocation of pre-adjustment effects |
|---|---|---|---|
| Raw series (forecasts) | | y (y_f) | |
| Linearized series | B1 | | none |
| 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)

#### 0.0.0.0.1 * Display in GUI

Final results are displayed for each series in the NODE MAIN>Table

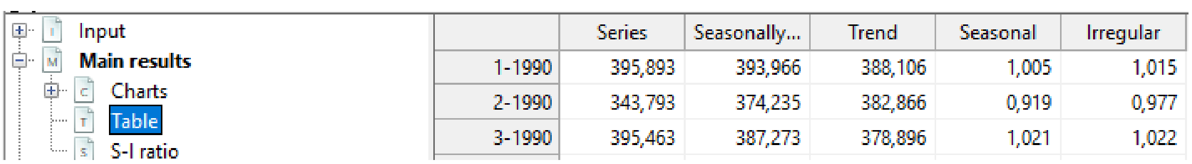



Figure 19: Text

Forecasts are glued at the end it *italic*

What can be done with this table (exapandable note ?) - charts container (link) - single series contextual menu (link)

##### 0.0.0.0.2 * Retrieve from GUI

Generating outout from GUI (link) or from Cruncher (link), component series, their standard errors, forecasts and forecasts errors can be acessed with the following names

| Series Name | Meaning |
|---|---|
| y | |
| y_f | |
| t | |
| t_f | |
| sa | |
| sa_f | |
| s | |
| s_f | |
| i | |
| i_f | |

##### 0.0.0.0.3 * Retrieve in R

version 2 the names are the same as for retrieving from GUI

version 3

### Output2: Final Params

This section describes 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 ?

#### Arima Models for components

brief description

##### 0.0.0.0.1 * Display in GUI

Click on the decomposition NODE

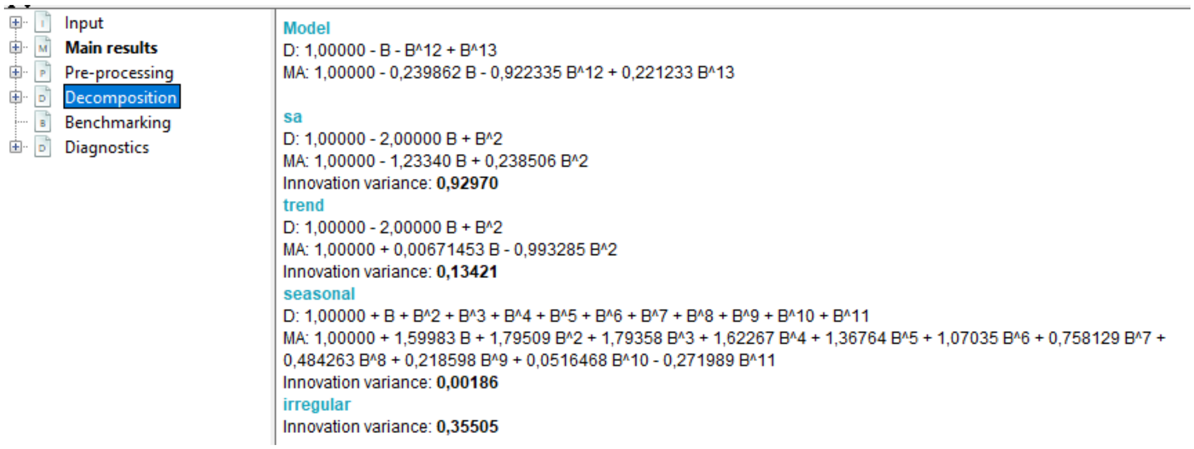



Figure 20: Text

##### 0.0.0.0.2 * Retrieve from GUI

?

##### 0.0.0.0.3 * Retrieve in R

version 2 version 3

#### Other final parameters

Final parameters which can be fine tuned be the user are described in User-defined specifications section below

### 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

### Setting user-defined parameters

The section below explains how the user can fine-tune some seats parameters, which are put in context in the corresponding method chapter.the default value is indicated ().

- 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

#### 0.0.0.0.1 * Seting parameters in GUI

In specification window (link) corresponding to a given series:

| SEATS | |
|---|---|
| Prediction length | -1 |
| Approximation mode | Legacy |
| MA unit root boundary | 0,95 |
| Trend boundary | 0,5 |
| Seasonal tolerance | 2 |
| Seasonal boundary | 0,8 |
| Seas. boundary (unique) | 0,8 |
| Method | Burman |

Figure 21: Text

#### 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, have more built-in available features, like pre-defined quality assessment)

- display in GUI
- retrieve from GUI (or cruncher)
- retrieve from R

(see excel table)

### 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 (what 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 algorithms. 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 ($y_{\text{lin}}$). 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 $(\text{No.}\,(\text{Mondays}) - No.\,(\text{Sundays})\,, \ldots No.\,(\text{Saturdays}) - No.(Sundays))$ 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.

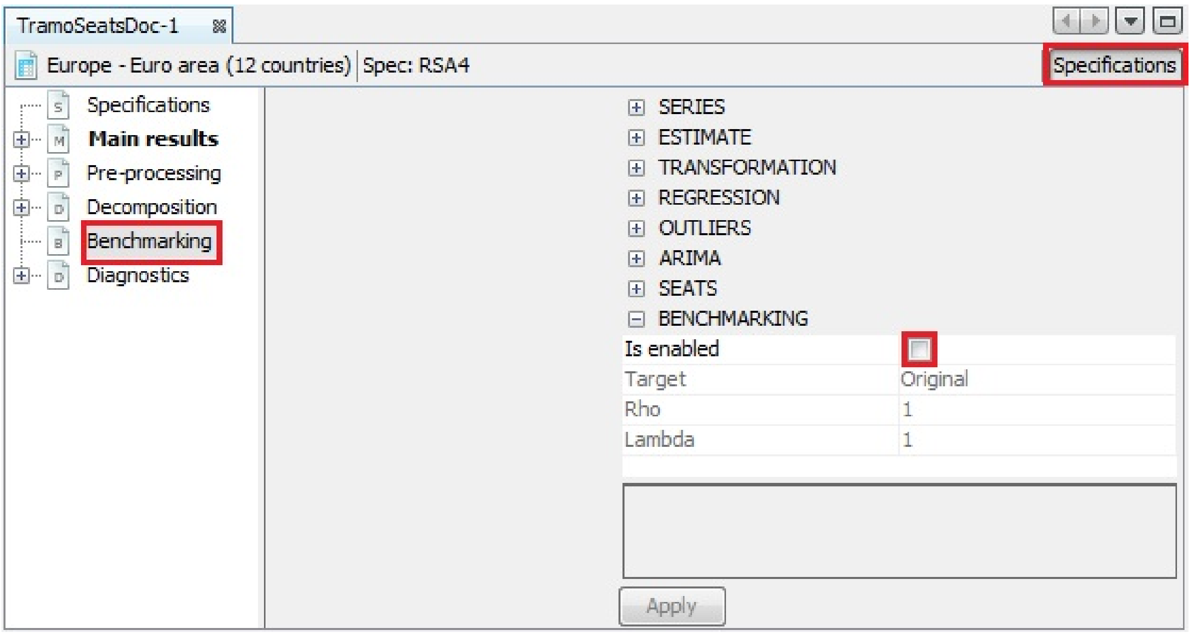



Figure 22: Text

**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.

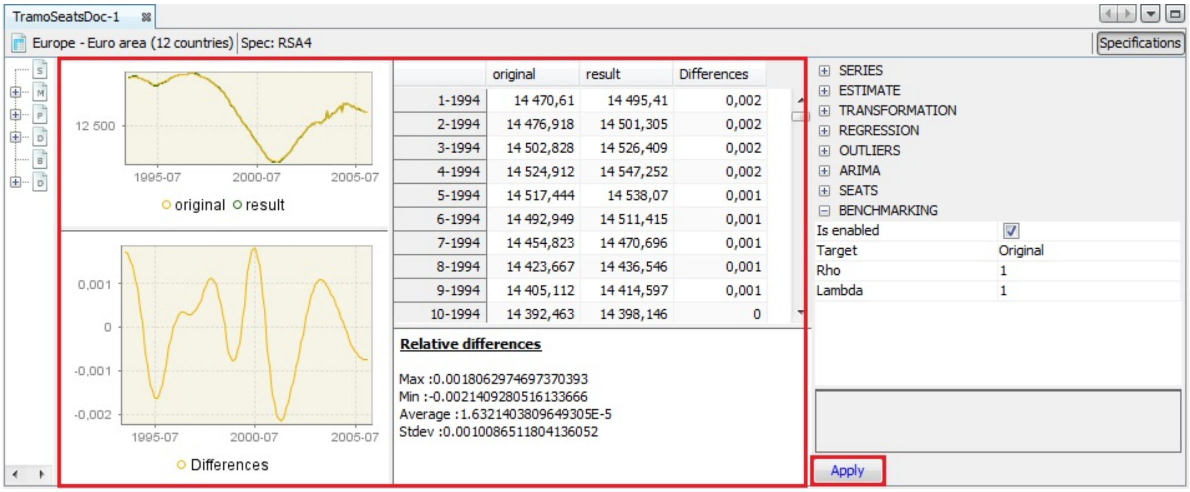



Figure 23: Text

**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**

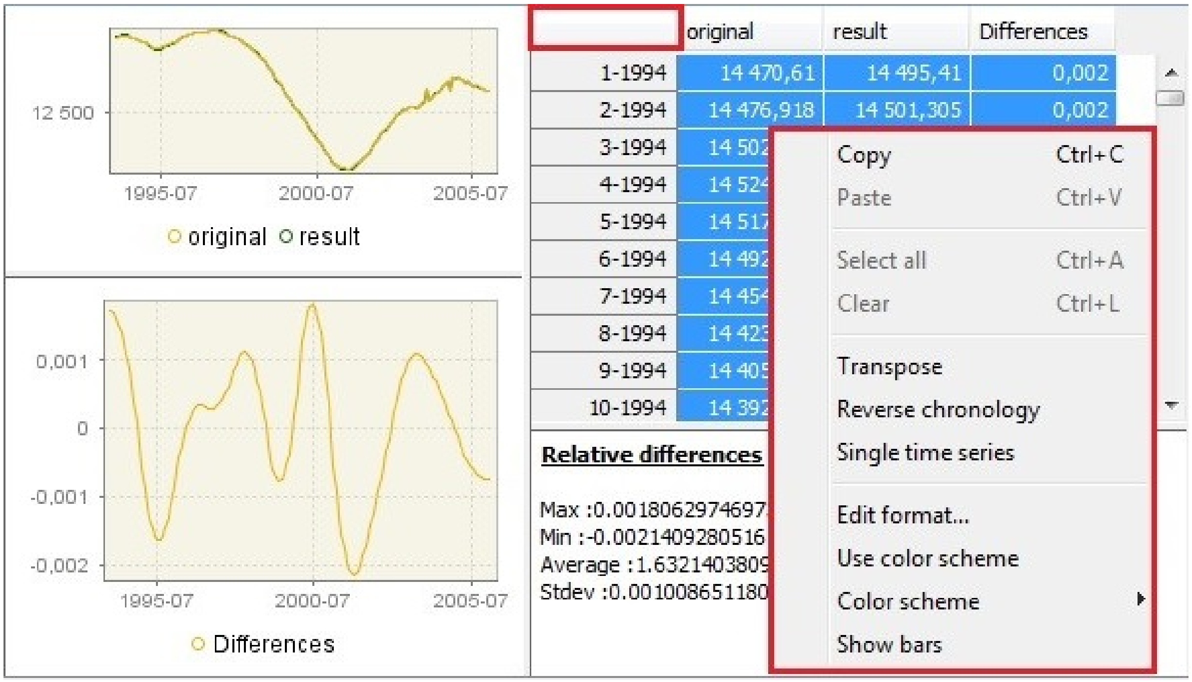



Figure 24: Text

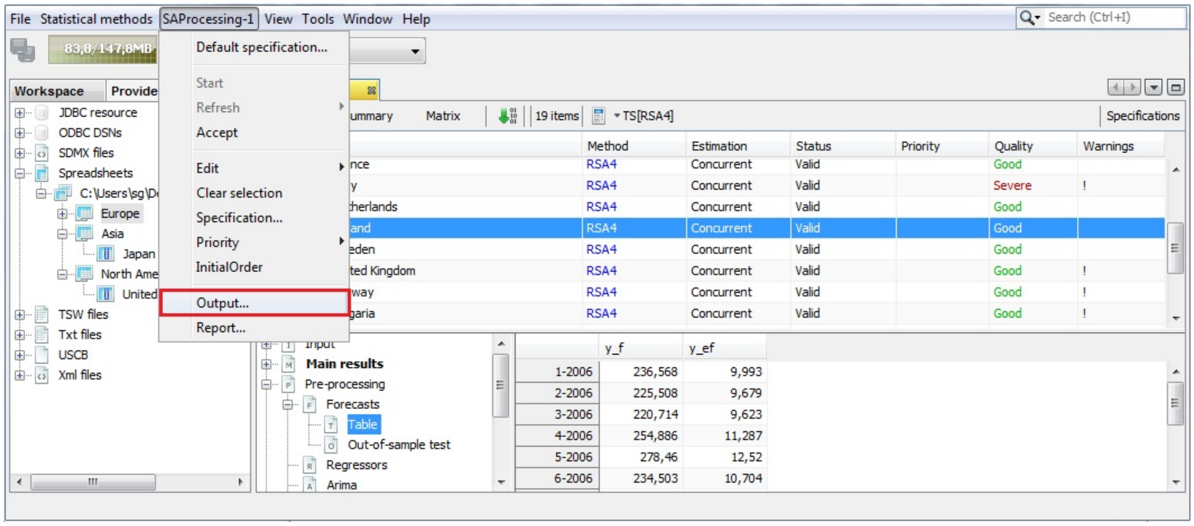



Figure 25: Text

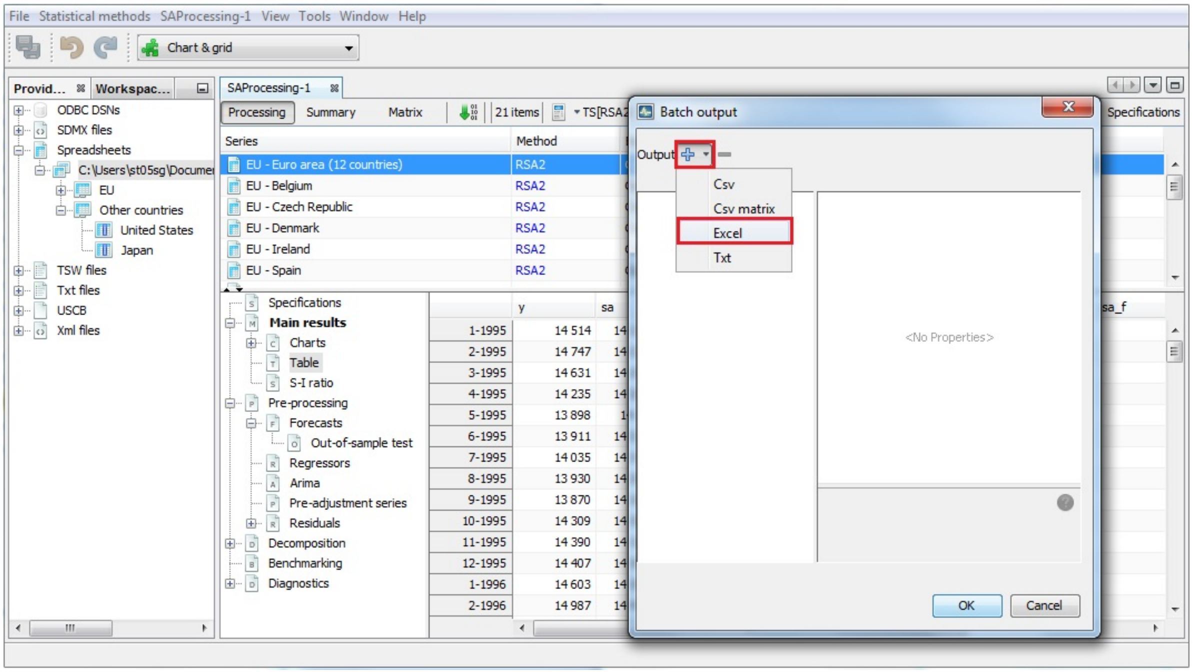



Figure 26: Text

## Benchmarking in R

package rjd3bench orga doc - here - in package - example

## References

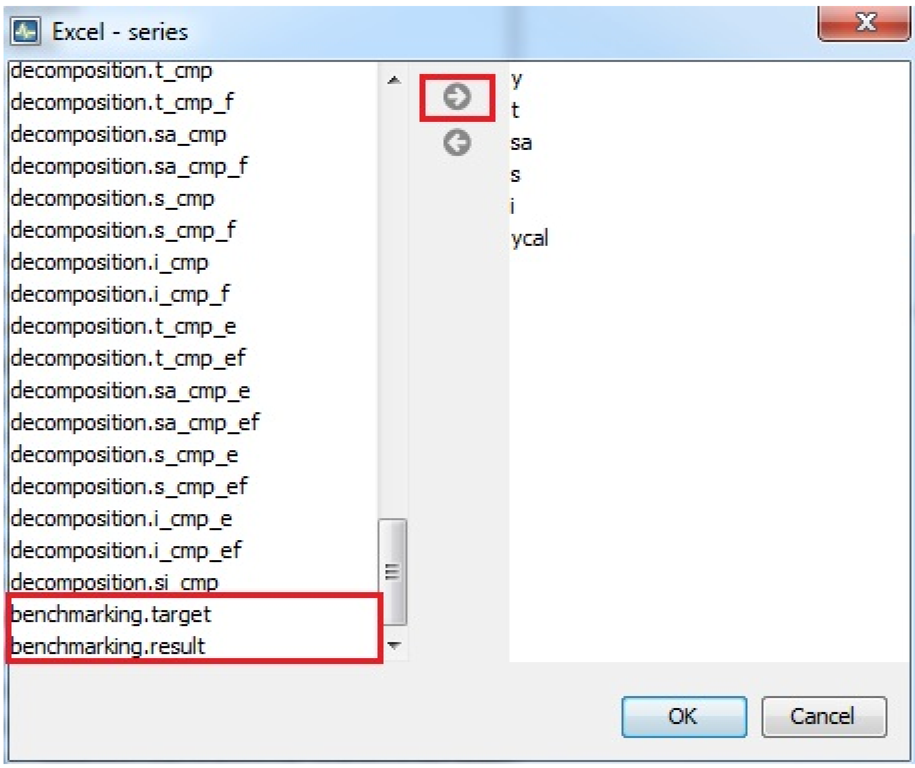



Figure 27: Text

# 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

This chapter provides general information about using the graphical interface (GUI). Specific indications related to a given algorithm (X13-Arima, Tramo-seats, Benchmarking...) are displayed in the relevant chapters.

## Available algorithms

The graphical user interface in the 2.x family gives access to:

- Seasonal adjustment (SA) algorithms
  - X13-Arima
  - Tramo-Seats
  - Direct-indirect SA comparisons
- Outlier detection (TERROR)
- Benchmarking

The graphical user interface in the 3.x family gives access **in addition** to extended SA algorithms for high-frequency data (HF) high-frequency data (HF).

## Available Time Series tools

The graphical user interface in the 2.x and 3.x family give access to generic time series tools:

- Graphics
  - time domain
  - spectral analysis
- Tests
  - seasonality tests
  - autocorrelation, normality, randomness tests

## 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).

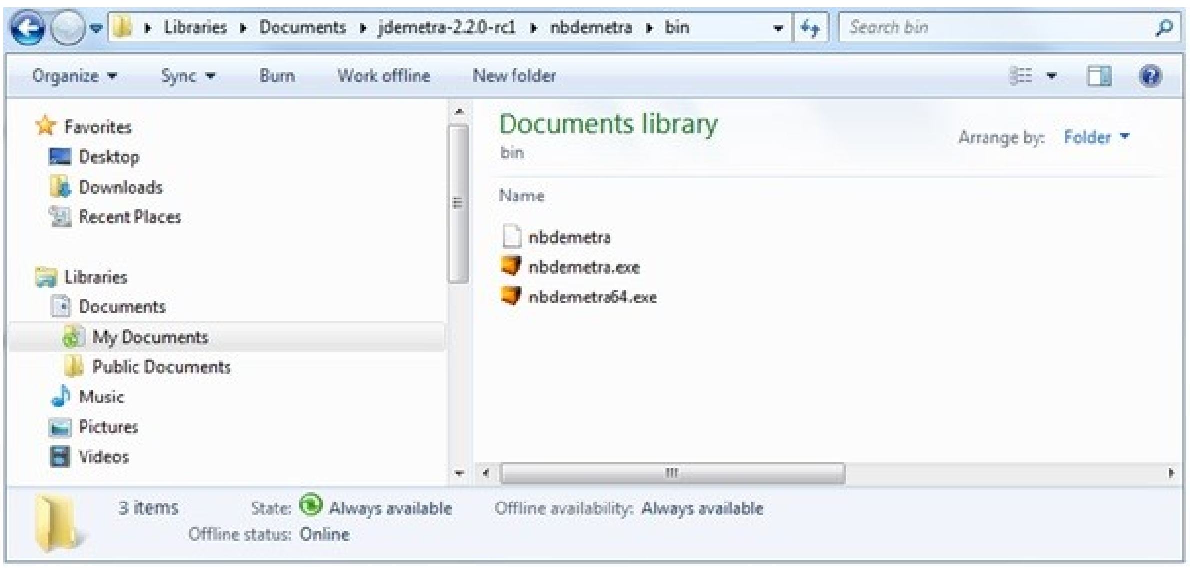



Figure 28: Text

**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).

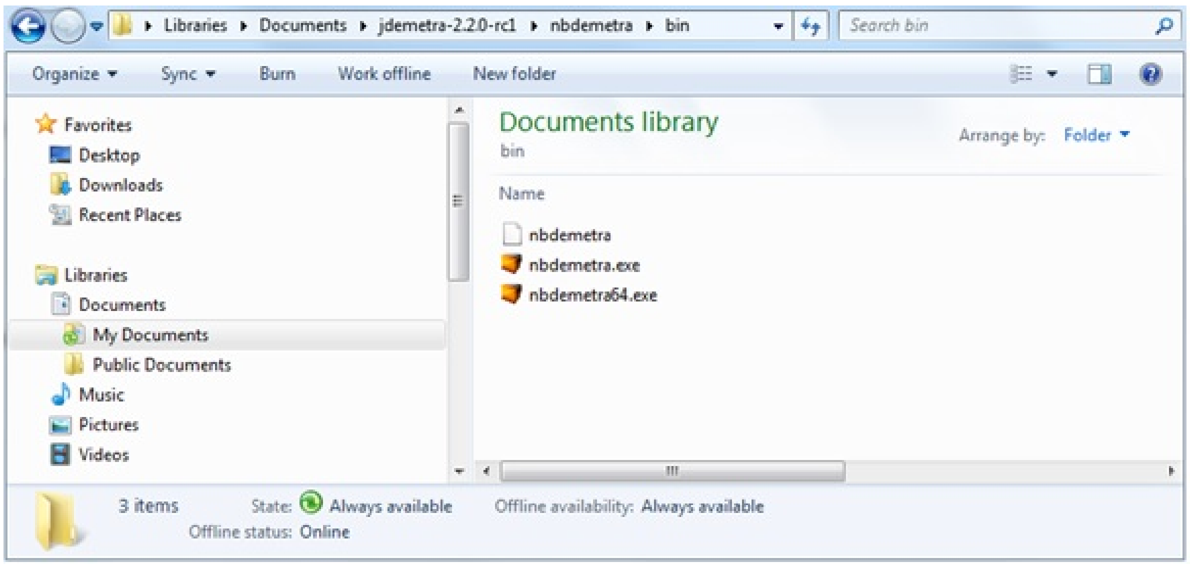



Figure 29: Text

**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**

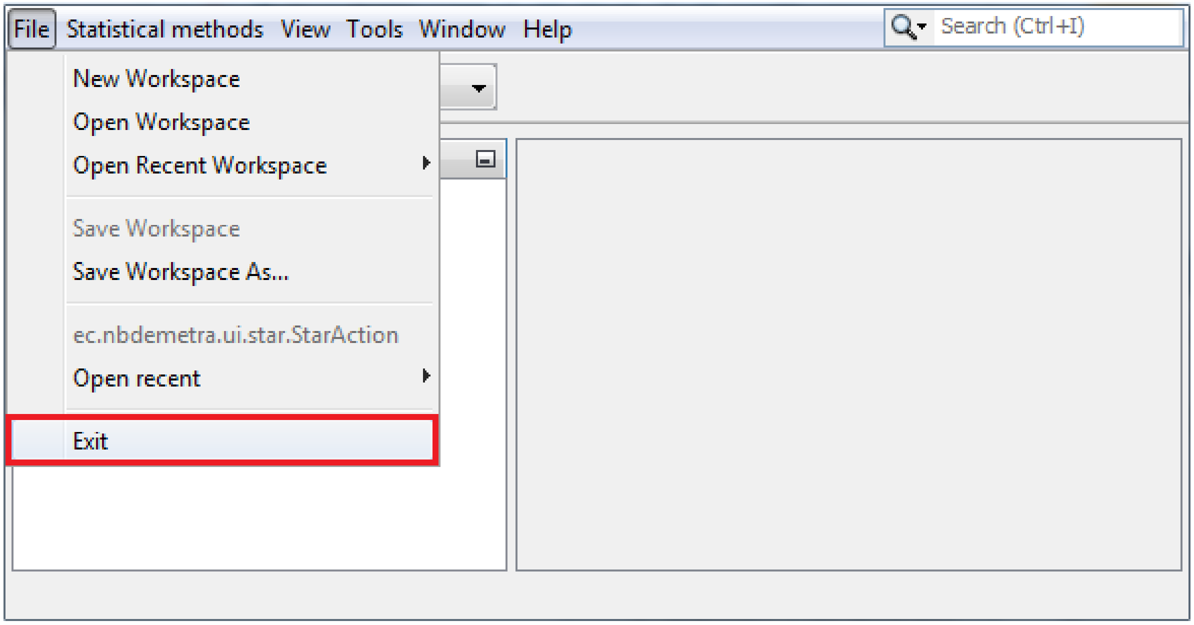



Figure 30: Text

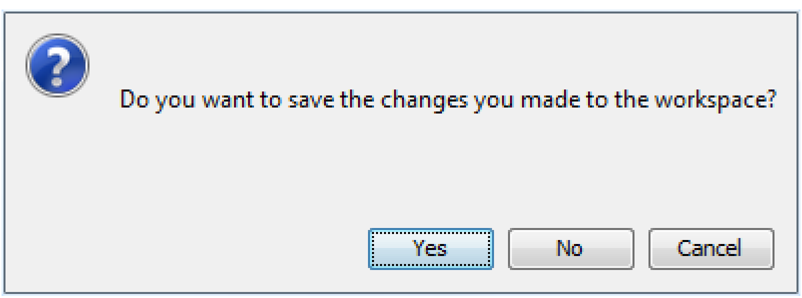



Figure 31: Text

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.

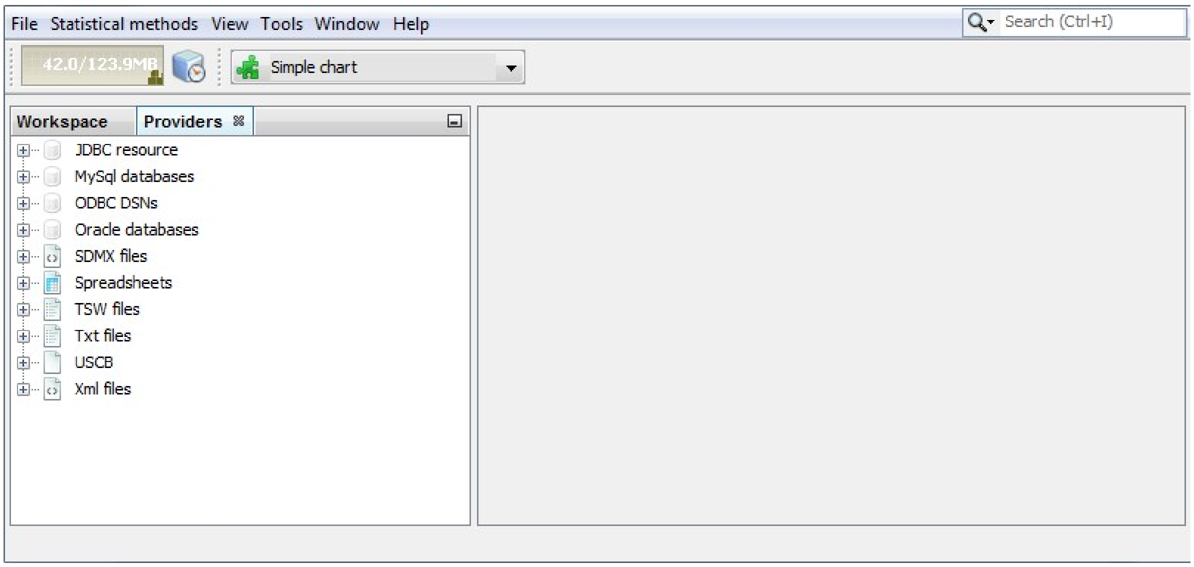



Figure 32: Text

### JDemetra+ default window

By default, on the left hand side of the window two panels are visible: the *Workspace* panel and the *Providers* panel. The *Workspace* panel stores the work performed by the user in a coherent and structured way. The *Providers* panel presents the list of the data sources and organizes the imported series within each data provider. By default, JDemetra+ supports the following data sources: * JDBC; * ODBC; * SDMX; * Excel spreadsheets; * TSW (input files for the Tramo-Seats-Windows application by the Bank of Spain); * TXT; * USCB (input files for the X-13-ARIMA-SEATS application by the U.S. Census Bureau); * XML.

All standard databases (Oracle, SQLServer, DB2, MySQL) are supported by JDemetra+ via JDBC, which is a generic interface to many relational databases. Other providers can be added by users by creating plugins. We will now focus on the Spreadsheets data source, which corresponds to the series prepared in an Excel file. The file should have dates in Excel date format. Dates should be placed in the first column (or in the first row) and titles of the series in the corresponding cell of the first row (or in the first column). The top-left cell [A1] can include text or it can be left empty. The empty cells are interpreted by JDemetra+ as missing values and they can appear at the beginning, in the middle and at the end of the time series. The example is shown below.

**Example of an Excel spreadsheet that can be imported to JDemetra+**

Once the spreadsheet is prepared and saved, it can be imported to JDemetra+ as it is shown by the tutorial below.

**An example of importing process for the Excel file**

The default JDemetra+ window, which is displayed after launching the program, is clearly divided into several panels.

**JDemetra+ default view**

The key parts of the user interface are: * The application menu. * The Providers window, which organises time series; * The Workspace window, which stores results generated by the software as well as settings used to create them; * A central empty zone for presenting the actual analyses further called the Results panel.

## Widgets

### All TS&view

### Search option

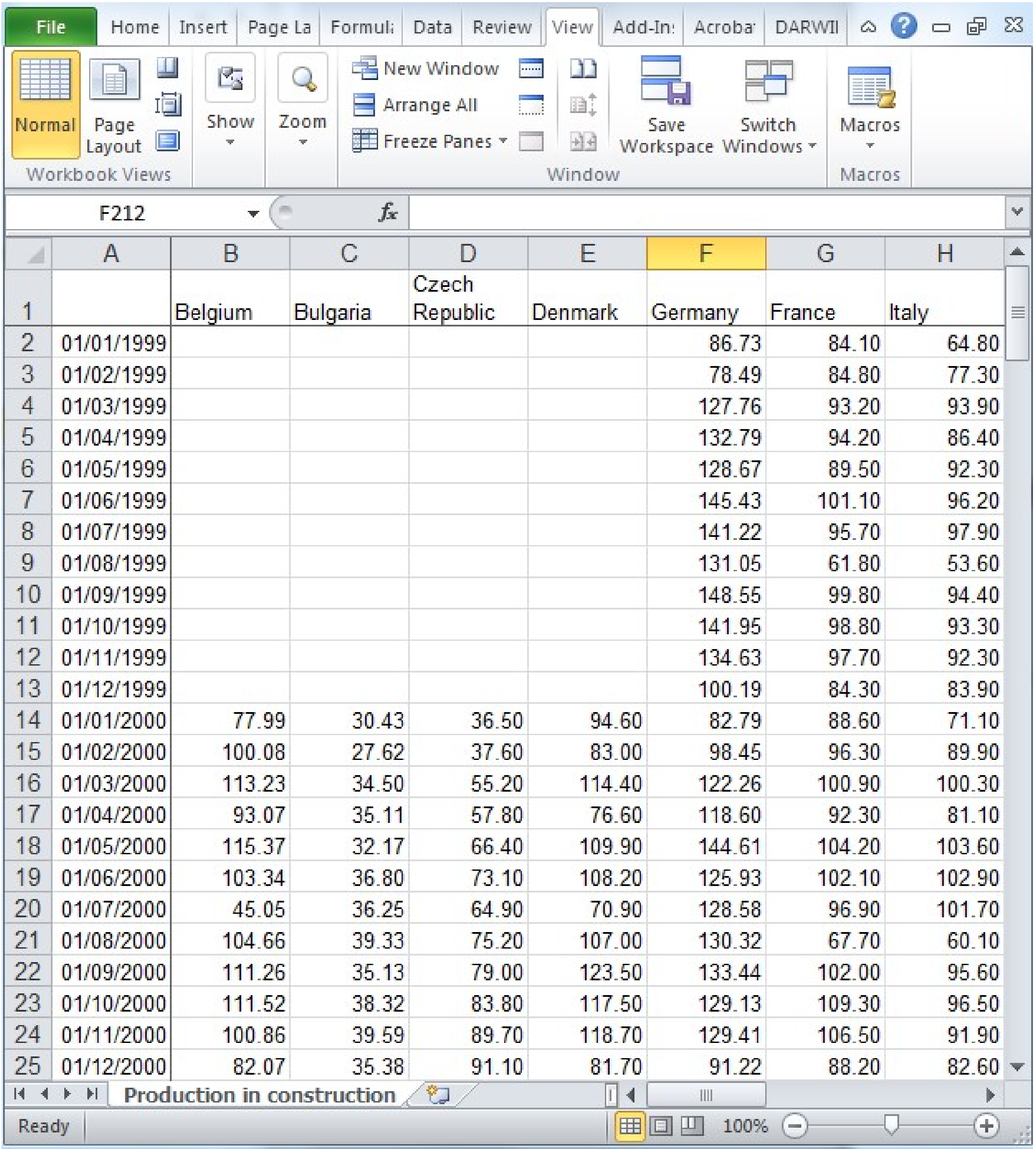
## Top bar menus

(Application menu: all that is available from the top bar)

The majority of functionalities are available from the main application menu, which is situated at the very top of the main window. If the user moves the cursor to an entry in the main menu and clicks on the left mouse button, a drop-down menu will appear. Clicking on an entry in the drop-down menu selects the highlighted item.

**The main menu with selected drop-down menu**

The functions available in the main application menu are: * File * Statistical methods * X-13Doc * RegArimaDoc * TramoDoc * TramoSeatsDoc * View * Tools * Window * Help



| A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|
|  | Belgium | Bulgaria | Czech Republic | Denmark | Germany | France | Italy |
| 01/01/1999 |  |  |  |  | 86.73 | 84.10 | 64.80 |
| 01/02/1999 |  |  |  |  | 78.49 | 84.80 | 77.30 |
| 01/03/1999 |  |  |  |  | 127.76 | 93.20 | 93.90 |
| 01/04/1999 |  |  |  |  | 132.79 | 94.20 | 86.40 |
| 01/05/1999 |  |  |  |  | 128.67 | 89.50 | 92.30 |
| 01/06/1999 |  |  |  |  | 145.43 | 101.10 | 96.20 |
| 01/07/1999 |  |  |  |  | 141.22 | 95.70 | 97.90 |
| 01/08/1999 |  |  |  |  | 131.05 | 61.80 | 53.60 |
| 01/09/1999 |  |  |  |  | 148.55 | 99.80 | 94.40 |
| 01/10/1999 |  |  |  |  | 141.95 | 98.80 | 93.30 |
| 01/11/1999 |  |  |  |  | 134.63 | 97.70 | 92.30 |
| 01/12/1999 |  |  |  |  | 100.19 | 84.30 | 83.90 |
| 01/01/2000 | 77.99 | 30.43 | 36.50 | 94.60 | 82.79 | 88.60 | 71.10 |
| 01/02/2000 | 100.08 | 27.62 | 37.60 | 83.00 | 98.45 | 96.30 | 89.90 |
| 01/03/2000 | 113.23 | 34.50 | 55.20 | 114.40 | 122.26 | 100.90 | 100.30 |
| 01/04/2000 | 93.07 | 35.11 | 57.80 | 76.60 | 118.60 | 92.30 | 81.10 |
| 01/05/2000 | 115.37 | 32.17 | 66.40 | 109.90 | 144.61 | 104.20 | 103.60 |
| 01/06/2000 | 103.34 | 36.80 | 73.10 | 108.20 | 125.93 | 102.10 | 102.90 |
| 01/07/2000 | 45.05 | 36.25 | 64.90 | 70.90 | 128.58 | 96.90 | 101.70 |
| 01/08/2000 | 104.66 | 39.33 | 75.20 | 107.00 | 130.32 | 67.70 | 60.10 |
| 01/09/2000 | 111.26 | 35.13 | 79.00 | 123.50 | 133.44 | 102.00 | 95.60 |
| 01/10/2000 | 111.52 | 38.32 | 83.80 | 117.50 | 129.13 | 109.30 | 96.50 |
| 01/11/2000 | 100.86 | 39.59 | 89.70 | 118.70 | 129.41 | 106.50 | 91.90 |
| 01/12/2000 | 82.07 | 35.38 | 91.10 | 81.70 | 91.22 | 88.20 | 82.60 |

Figure 33: Text

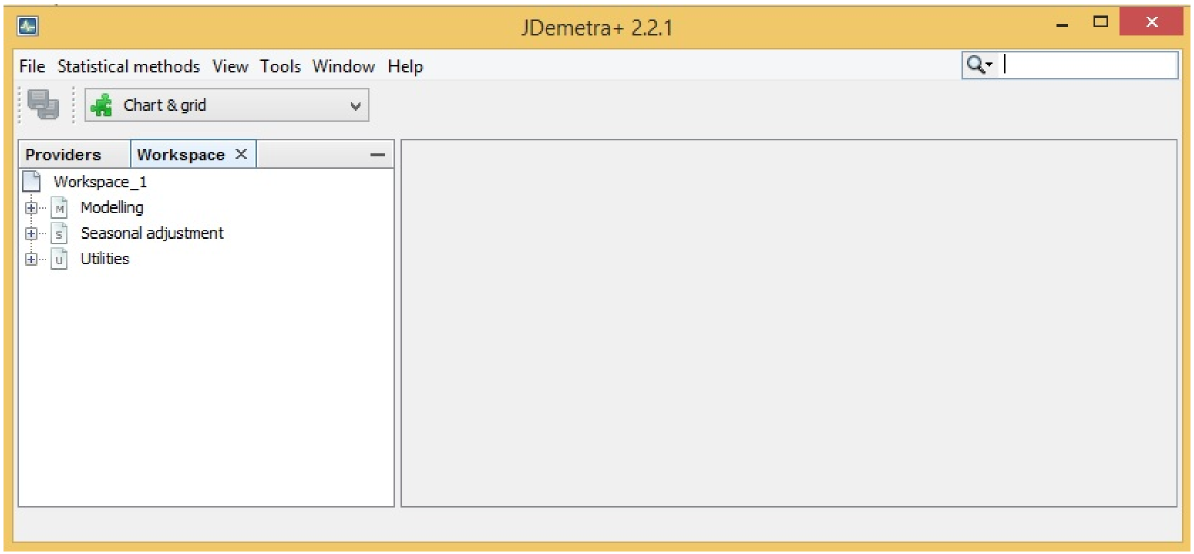



Figure 34: Text

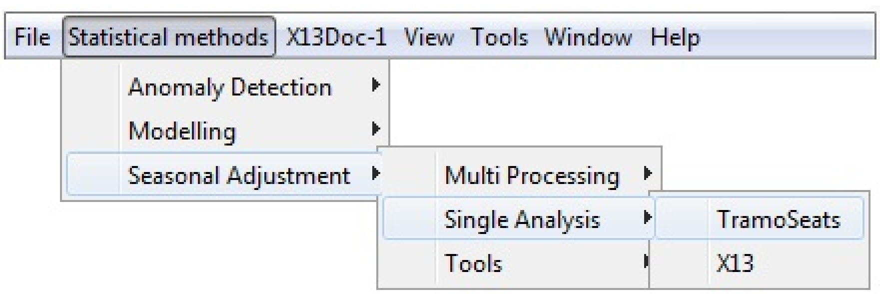



Figure 35: Text

### File

The *File* menu is intended for working with workspaces and data sources. It offers the following functions:
* **New Workspace** – creates a new workspace and displays it in the *Workspace* window with a default name (*Workspace_#number*); * **Open Workspace** – opens a dialog window, which enables the user to select and open an existing workspace; * **Open Recent Workspace** – presents a list of workspaces recently created by the user and enables the user to open one of them; * **Save Workspace** – saves the project file named by the system under the default name (*Workspace_#number*) and in a default location. The workspace can be re-opened at a later time; * **Save Workspace As...** – saves the current workspace under the name chosen by the user in the chosen location. The workspace can be re-opened at a later time; * **Open Recent** – presents a list of datasets recently used and enables the user to open one of them; * **Exit** – closes an application.

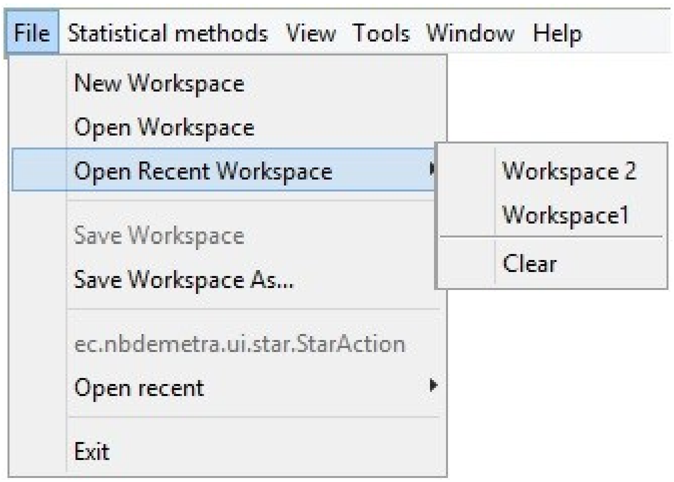



Figure 36: Text

**The content of the *File* menu**

### Statistical Methods

The Statistical methods menu includes functionalities for modelling, analysis and the seasonal adjustment of a time series. They are divided into three groups: * Anomaly Detection – allows for a purely automatic identification of regression effects; * Modelling – enables time series modelling using the TRAMO and RegARIMA models; * Seasonal adjustment – intended for the seasonal adjustment of a time series with the TRAMO-SEATS and X-13ARIMA-SEATS methods.

**The *Statistical methods* menu.**

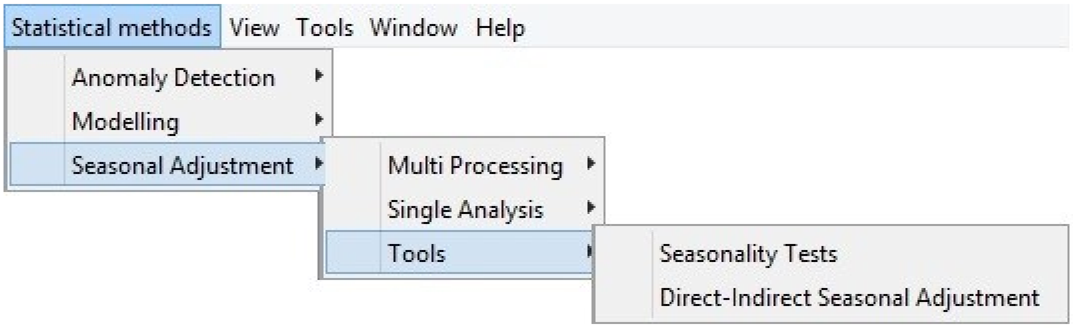



Figure 37: Text

## View

The View menu contains functionalities that enable the user to modify how JDemetra+ is viewed. It offers the following items: * **Split** – the function is not operational in the current version of the software. * **Toolbars** – displays selected toolbars under the main menu. The *File* toolbar contains the *Save all* icon. The *Performance* toolbar includes two icons: one to show the performance of the application, the other to stop the application profiling and taking a snapshot. The *Other* toolbar determines the default behaviour of the program when the user double clicks on the data. It may be useful to plot the data, visualise it on a grid, or to perform any pre-specified action, e.g. execute a seasonal adjustment procedure. * **Show Only Editor** – displays only the *Results* panel and hides other windows (e.g. *Workspace* and *Providers*). * **Full Screen** – displays the current JDemetra+ view in full screen.

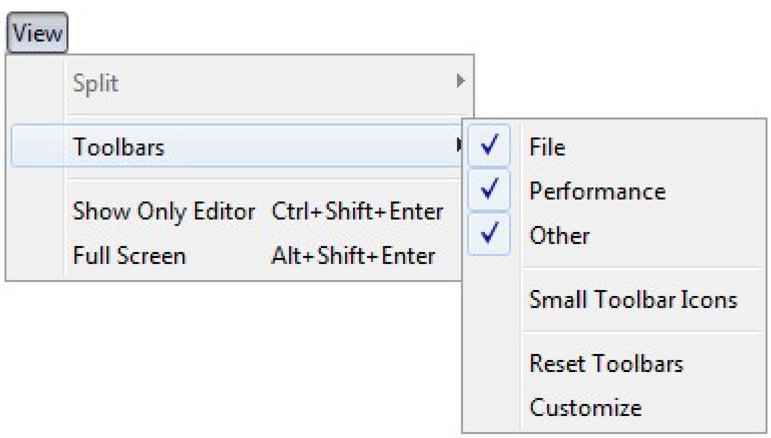



Figure 38: Text

**The *View* menu**

## Tools menu

The following functionalities are available from the *Tools* menu:

- *Container* – includes several tools for displaying data in a time domain;
- *Spectral analysis* – contains tools for the analysis of a time series in a frequency domain;
- *Aggregation* – enables the user to investigate a graph of the sum of multiple time series;
- *Differencing* – allows for the inspection of the first regular differences of the time series;
- *Spreadsheet profiler* – offers an Excel-type view of the XLS file imported to JDemetra+.
- *Plugins* – allows for the installation and activation of plugins, which extend JDemetra+ functionalities.
- *Options* – presents the default interface settings and allows for their modification.

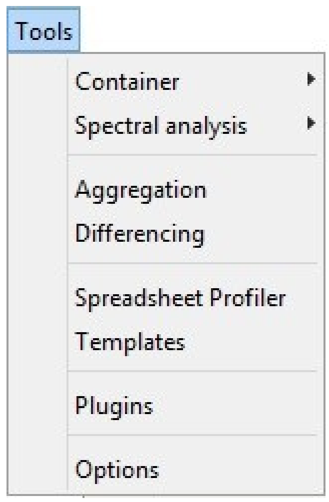



Figure 39: Text

**The *Tools* menu**

### Container

*Container* includes basic tools to display the data. The following items are available: *Chart*, *Grid*, *Growth Chart* and *List*.

**The *Container* menu**

Several containers can be opened at the same time. Each of them may include multiple time series.

*Chart* plots the time series as a graph. This function opens an empty window. To display a given series drag and drop the series from the *Providers* window into the empty window.

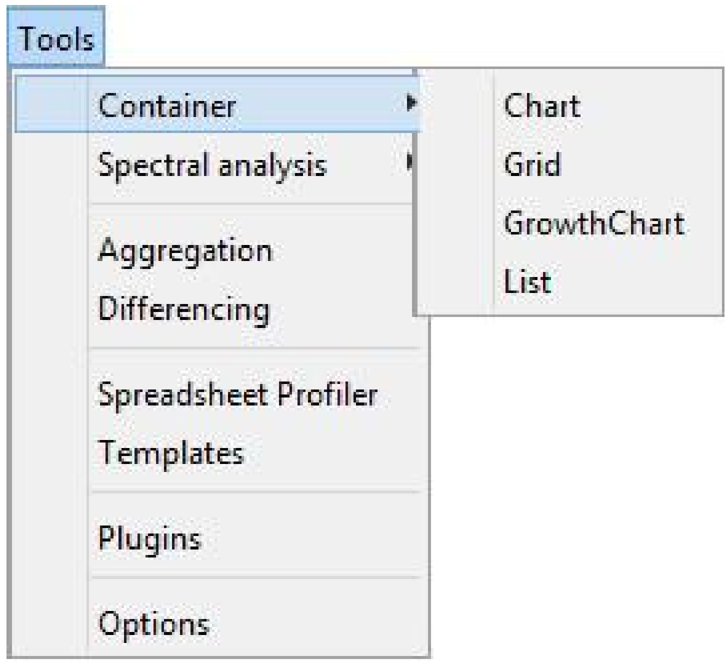



Figure 40: Text

More than one series can be displayed on one graph. The chart is automatically rescaled after adding a new series.

**Launching the *Chart* functionality**

The series to be viewed can be also dragged from the other windows (e.g. from the *Variables* window) or directly from the windows that display the results of the estimation procedure.

**Displaying the seasonally adjusted series on a separate chart**

To adjust the view of the chart and save it to a given location use the local menu, which is displayed after right-clicking on the chart. The explanation of the functions available for the local menu is given below.

**Local menu basic options for the time series graph**

To display the time series value at a given date, hover over it with the cursor. Once the time series is selected by clicking on it with the right mouse button, the options dedicated to this series are available.

**Local menu options for chart**

A list of possible actions includes:

- **Open** – opens selected time series in a new window that contains *Chart* and *Grid* panels.


Figure 41: Text

- **Open with** – opens the time series in a separate window according to the user choice (*Chart & grid* or *Simple chart*). The *All ts views* option is not currently available.
- **Save** – saves the marked series in a spreadsheet file or in a text file.
- **Rename** – enables the user to change the time series name.
- **Freeze** – disables modifications of the chart.
- **Copy** – copies the series and allows it to be pasted to another application e.g. into Excel.
- **Paste** – pastes the time series previously marked.
- **Split into yearly components** – opens a window that presents the analysed series data split by year. This chart is useful to investigate the differences in time series values caused by the seasonal factors as it gives some information on the existence and size of the deterministic and stochastic seasonality in data.
- **Remove** – removes a time series from the chart.
- **Select all** – selects all the time series presented in the graph.
- **Show title** – option is not currently available.
- **Show legend** – displays the names of all the time series presented on the graph.
- **Edit format** – enables the user to change the data format.
- **Color scheme** – allows the colour scheme used in the graph to be changed.

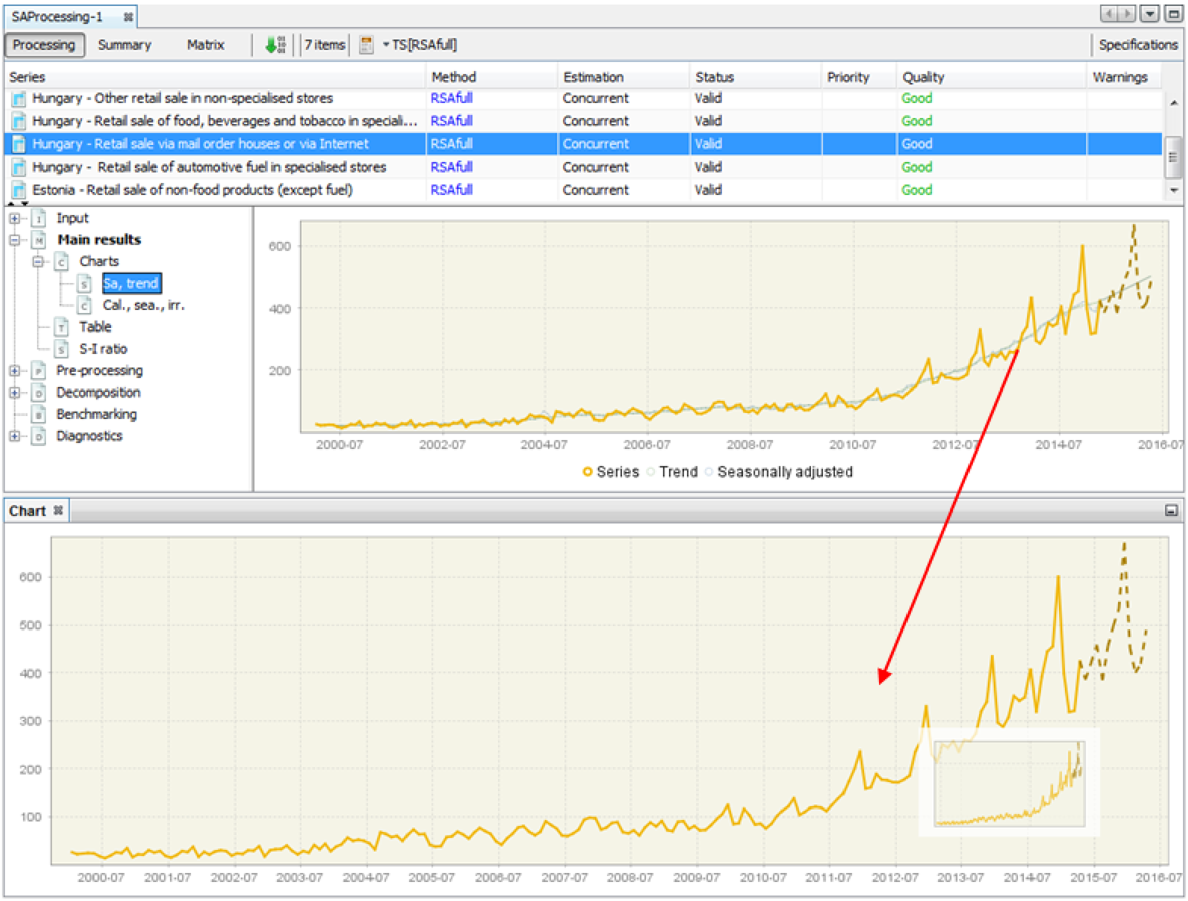



Figure 42: Text

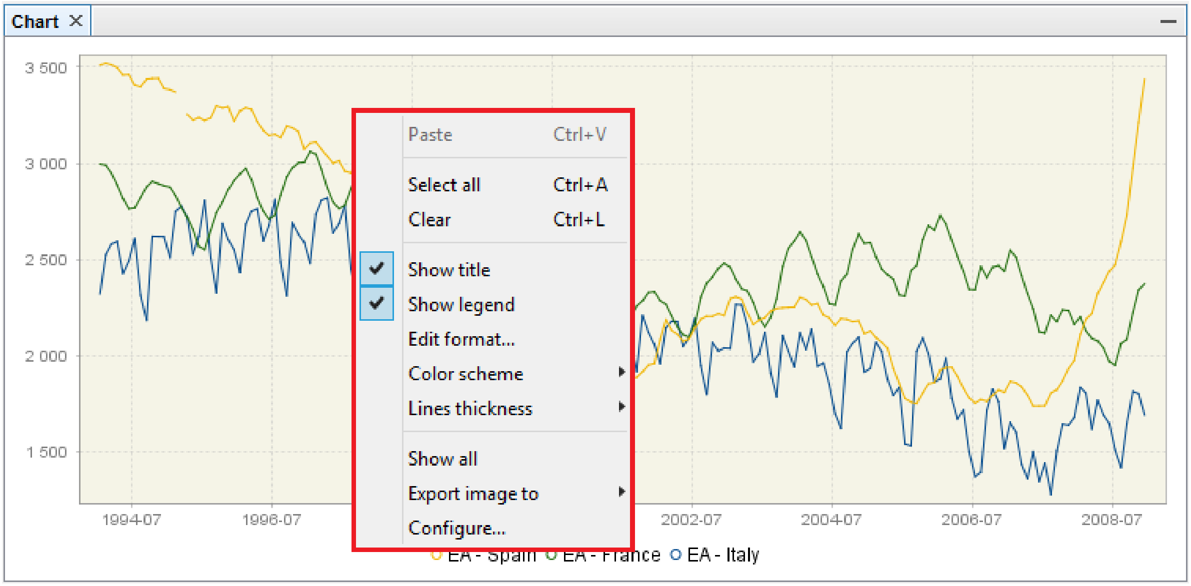



Figure 43: Text

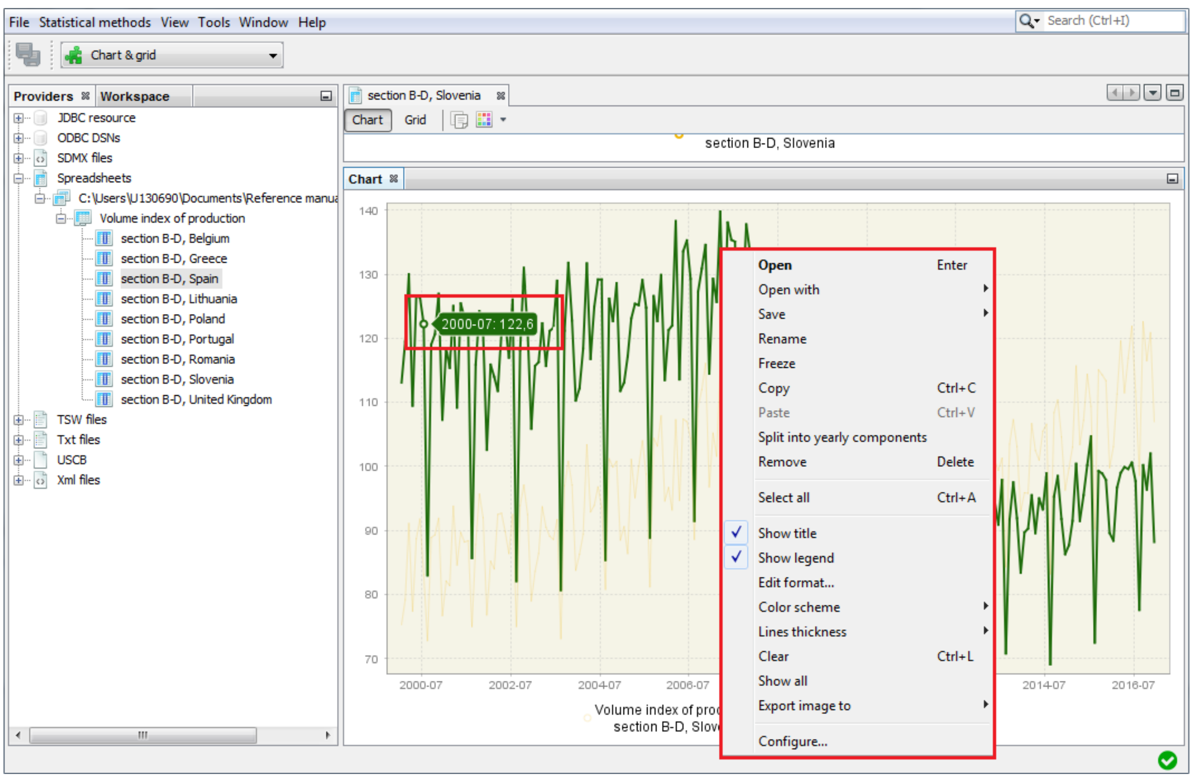



Figure 44: Text

- **Lines thickness** – allows the user to choose between thin and thick lines to be used for a graph.
- **Clear** – removes all the time series from the chart.
- **Show all** – this option is not currently available.
- **Export image to** – allows the graph to be sent to the printer and saved in the clipboard or as a file in a jpg format.
- **Configure** – enables the user to customize the chart and series display.

*Grid* enables the user to display the selected time series as a table. This function opens an empty window. To display a given series drag and drop the series from the *Providers* window into the empty window. More than one series can be displayed in one table.

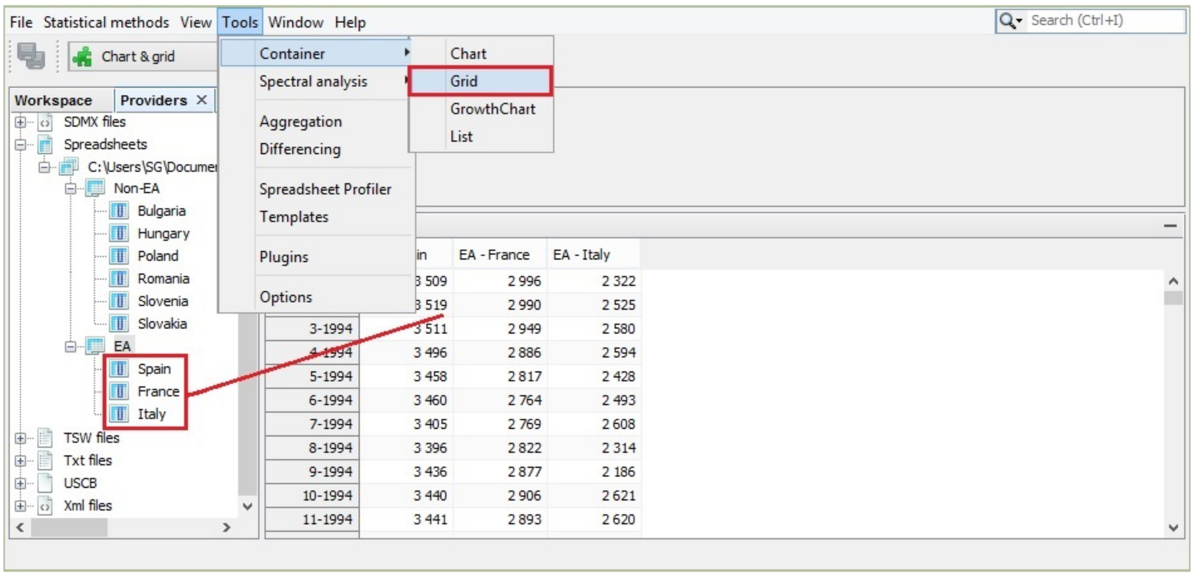



Figure 45: Text

### Launching the *Grid* functionality

To display options available for a given time series, left click on any time series' observation.

### Local menu options for the *Grid* view

The options available in *Grid* are:

- **Transpose** – changes the orientation of the table from horizontal to vertical.
- **Reverse chronology** – displays the series from the last to the first observation.
- **Single time series** – removes from the table all time series apart from the selected one.

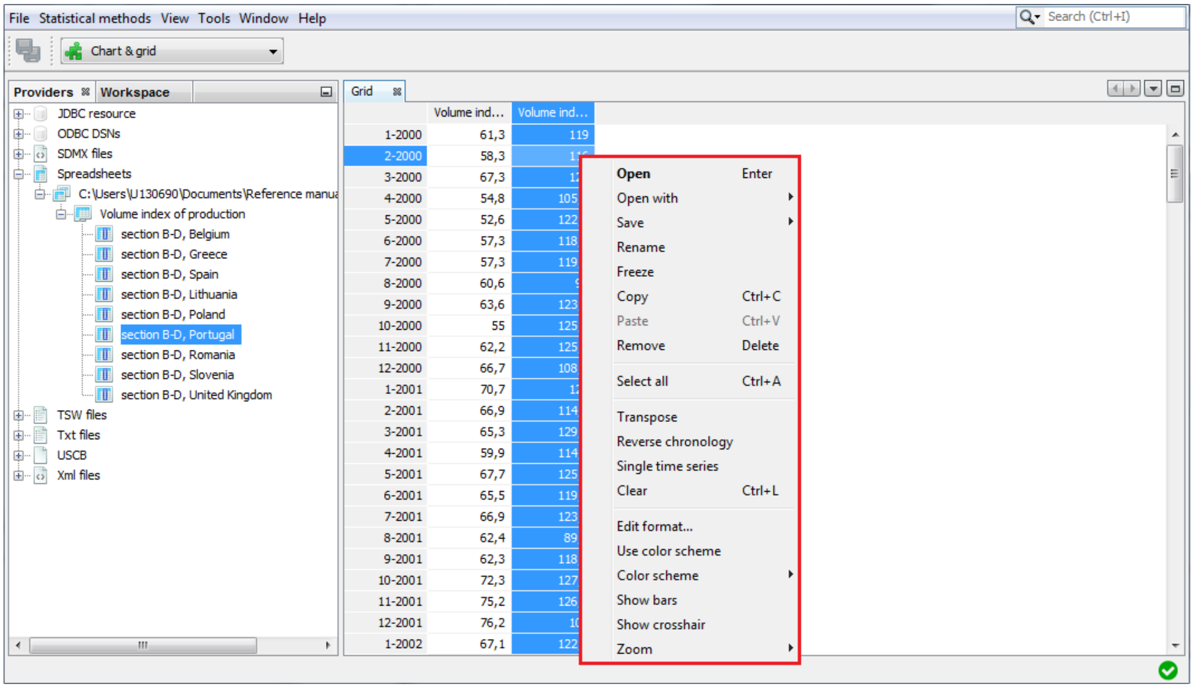



Figure 46: Text

- **Use color scheme** – allows the series to be displayed in colour.
- **Show bars** – presents values in a table as horizontal bars.
- **Show crosshair** – highlights an active cell.
- **Zoom** – option for modifying the chart size.

When none of the series is selected, the local menu offers a reduced list of options. The explanation of the other options can be found below in the '*Local menu options for chart*' figure in the *Container* section.

### A reduced list of options for *Grid*

The *Growth chart* tab opens an empty window. Once a given series is dropped into it, *Growth chart* presents the year-over-year or period-over-period growth rates for the selected time series. More than one series can be displayed in a table. The growth chart is automatically rescaled after adding a new series.

### The *Growth chart* view with a local menu

A left click displays a local menu with the available options. Those that are characteristic for the *Growth chart* are:

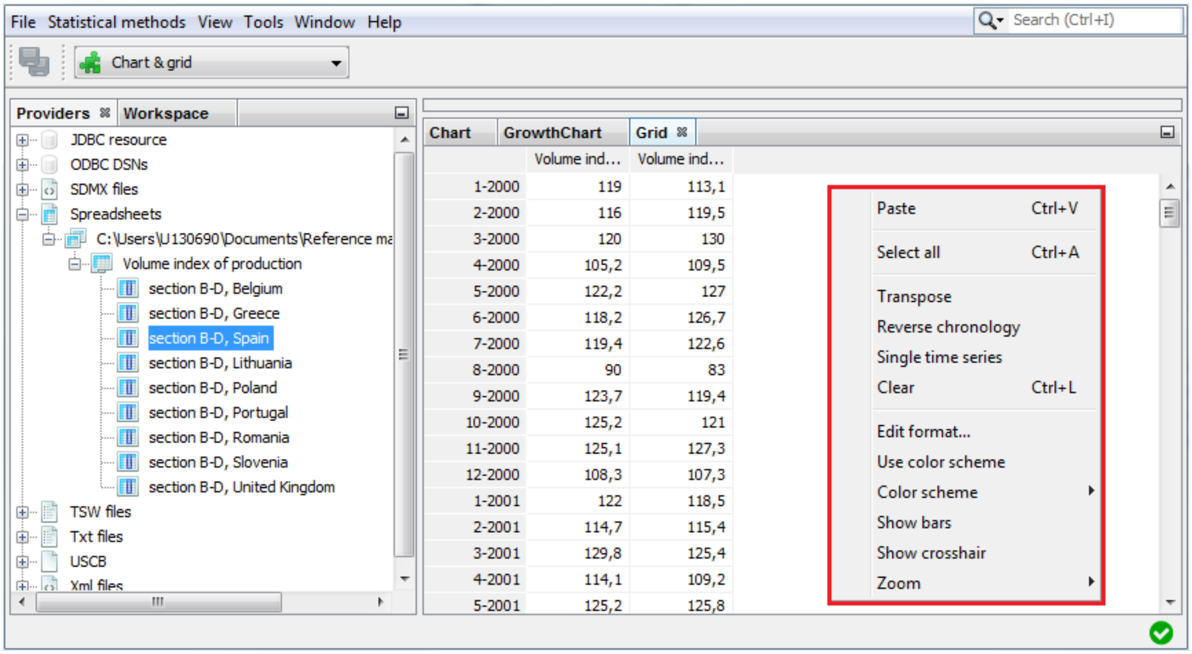



Figure 47: Text

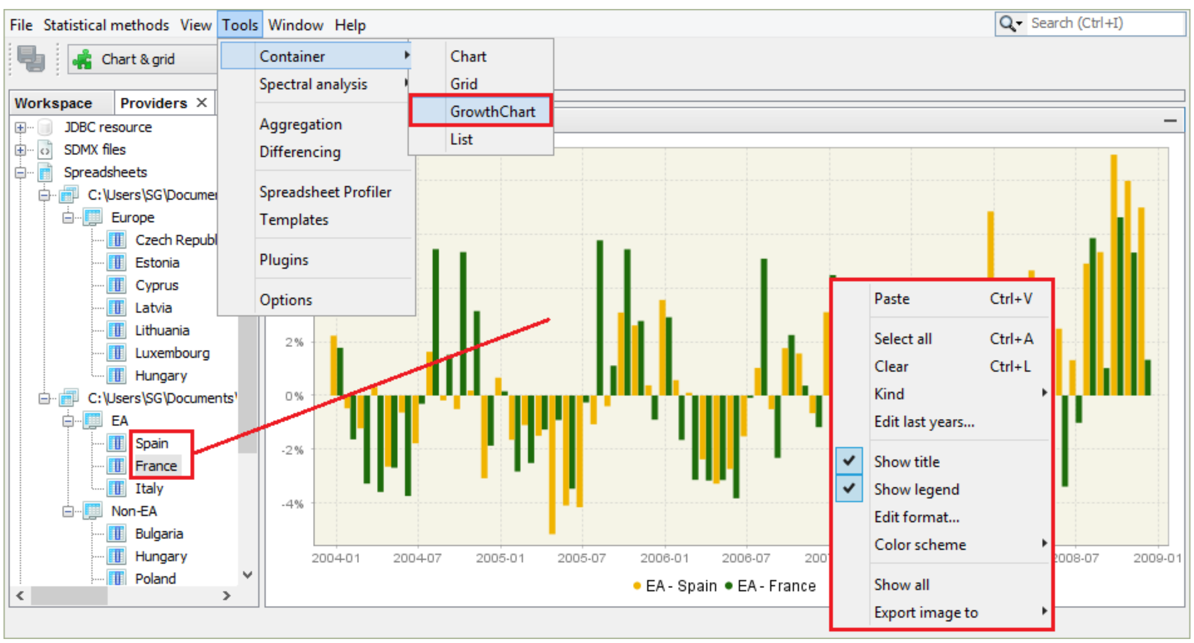



Figure 48: Text

- **Kind** – displays m/m (or q/q) and y/y growth rates for all time series in the chart (previous period and previous year options respectively). By default, the period-over-period growth rates are shown.
- **Edit last year** – for clarity and readability purposes, only five of the last years of observations are shown by default. This setting can be adjusted in the *Options* section, if required.

The explanation of other options can be found below in the '*Local menu options for chart*' figure in the *Container* section.

The *List* tab provides basic information about the chosen time series, such as; the start and end date, the number of observations and a sketch of the data graph. This function opens an empty window. To display information, drag and drop the series from the *Providers* window into the *List* window. A right click displays the local menu with all available options. Apart from the standard options, the local menu for *List* enables marking the series that match the selected frequency (yearly, half-yearly, quarterly, monthly) by using the *Select by frequency* option. An explanation of other options can be found below in the '*Local menu options for chart*' figure in the *Container* section.

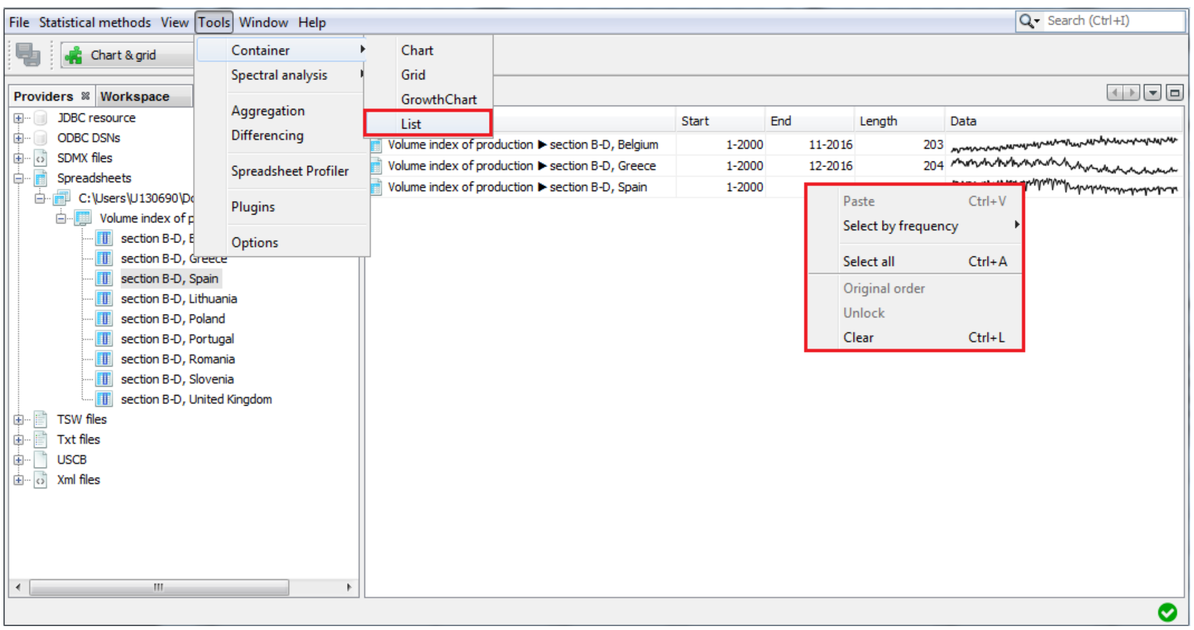



Figure 49: Text

### A view of a list of series

For a selected series a local menu offers an extended list of options. The explanation of the functions available for the local menu is given below in the '*Local menu options for chart*' figure in the *Container* section.

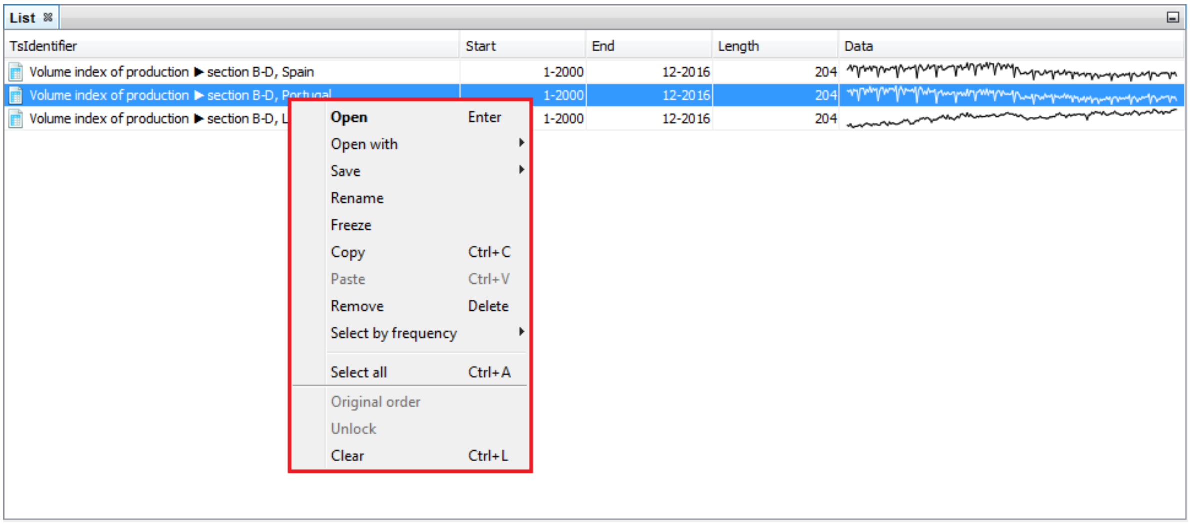



Figure 50: Text

### Options available for a selected series from the list

#### Spectral analysis

The *Spectral analysis* section provides three spectral graphs that allows an in-depth analysis of a time series in the frequency domain. These graphs are the *Auto-regressive Spectrum*, the *Periodogram* and the *Tukey Spectrum*. For more information the user may study a basic desription of spectral analysis and a detailed presentation of the abovementioned tools.

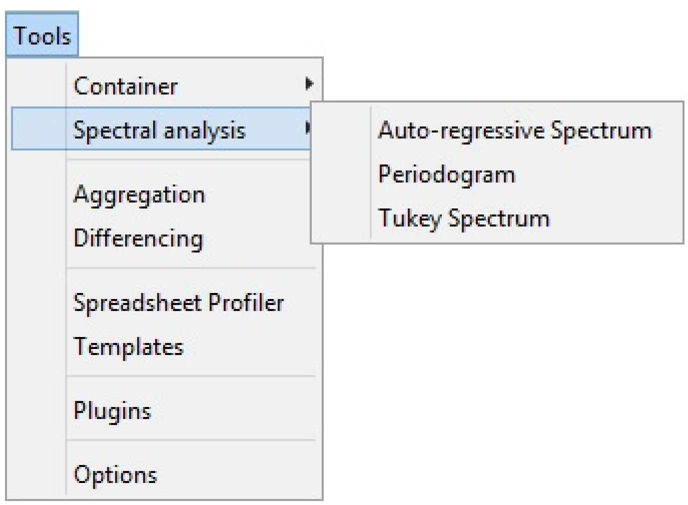



Figure 51: Text

### Tools for spectral analysis

## Aggregation

*Aggregation* calculates the sum of the selected series and provides basic information about the selected time series, including the start and end date, the number of observations and a sketch of the data graph, in the same way as in the *List* functionality. *Aggregation* opens an empty window. To sum the selected series, drag and drop them from the *Providers* window into the *Aggregation* window. Right click displays the local menu with the available options. The content of the local menu depends on the panel chosen (the panel on the left that contains the list of the series and the panel on the right that presents the graph of an aggregate). The local menu for the list of series offers the option *Select by frequency*, which marks all the series on the list that are yearly, half-yearly, quarterly or monthly (depending on the user's choice). The explanation of the other options can be found below in the '*Local menu options for chart*' figure in the *Container* section. The local menu for the panel on the left offers functionalities that are analogous to the ones that are available for the *List* functionalities, while the options available for the local menu in the panel on the left are the same as the ones available in *Chart* (see *Container*).

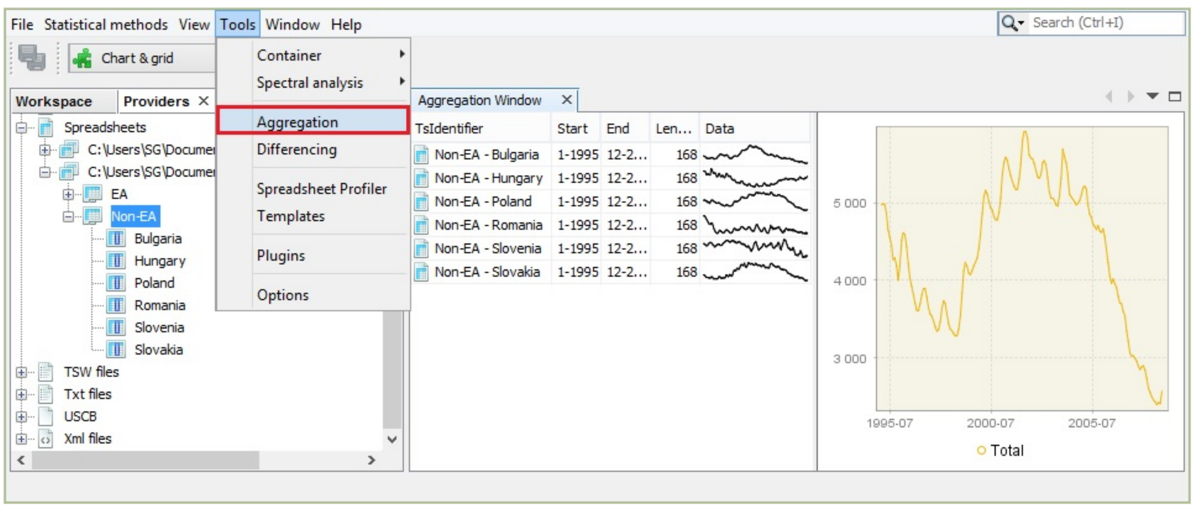



Figure 52: Text

### The *Aggregation* tool

## Differencing

The *Differencing* window displays the first regular differences for the selected time series together with the corresponding periodogram and the PACF function. By default, the window presents the results for non-seasonally and seasonally differenced series (( $d = 1, D = 1$)). These settings can be changed through the *Properties* window (*Tools* $\rightarrow$ *Properties*). A description of a periodogram and the PACF function can be found here.

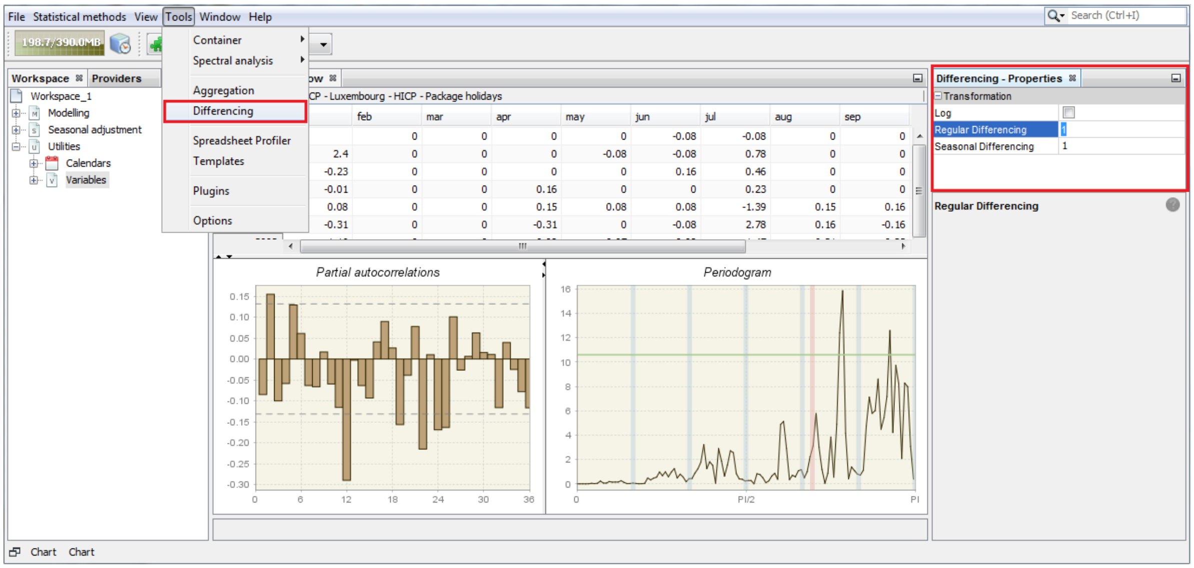


Figure 53: Text

**The properties of the *Differencing* tool**

The typical results are shown below. The bottom left graph presents the partial autocorrelation coefficients (vertical bars) and the confidence intervals. The right-click local menu offers several functionalities for a differenced series. An explanation of the available options can be found below in the "*Local menu options for chart*" figure in the *Container* section.

**The *Differencing* tool**

For the *Partial autocorrelation* and the *Periodogram* panels the right-button menu offers "a copy series" option that allows data to be exported to another application and a graph to be printed and saved to a clipboard or as a jpg file. Information about the partial autocorrelation function is given here. The description of a periodogram is presented in the Spectral graphs scenario.

### Spreadsheet profiler

The *Spreadsheet profiler* offers an Excel-type view of the XLS file imported to JDemetra+. To use this functionality drag the file name from the *Providers* window and drop it to the empty *Spreadsheet profiler* window.

**The *Spreadsheet Profiler* window**

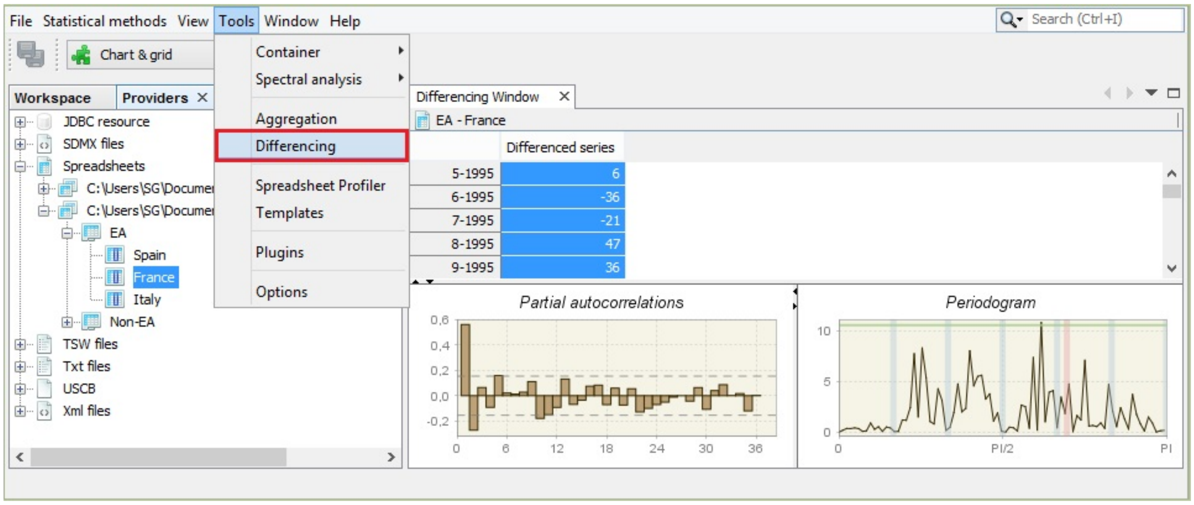



Figure 54: Text

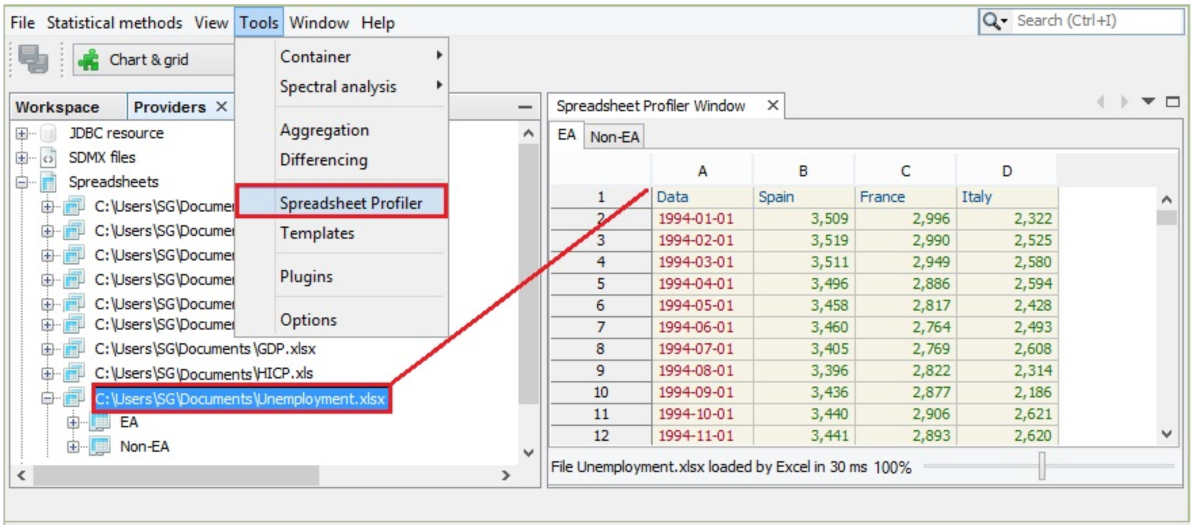



Figure 55: Text

### Plugins

Installation an functionalities of plugins are described in the related chapter

### Options

The *Options* window includes five main panels: *Demetra*, *General*, *Keymap*, *Appearance* and *Miscellaneous*. They are visible in the very top of the *Options* window.

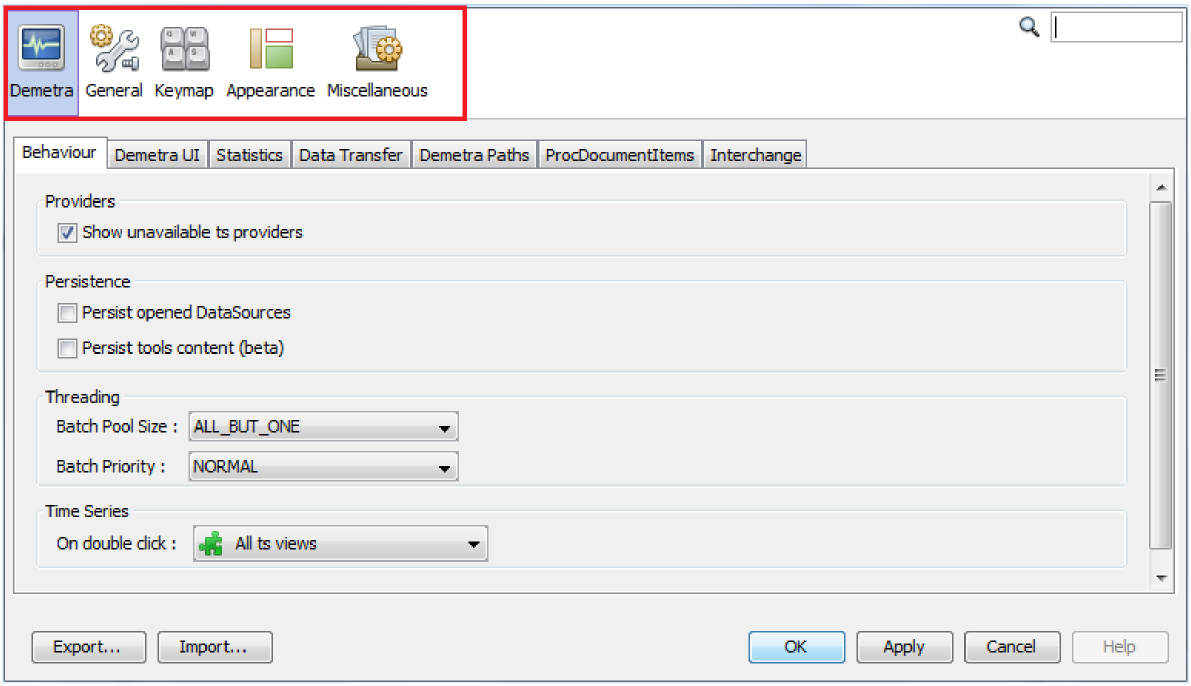



Figure 56: Text

#### The main sections of the *Options* window

By default, the *Demetra* tab is shown. It is divided into seven panels: *Behaviour*, *Demetra UI*, *Statistics*, *Data transfer*, *Demetra Paths*, *ProcDocumentItems*, and *Interchange*.

*Behaviour* defines the default reaction of JDemetra+ to some of the actions performed by the user.

- **Providers** – an option to show only the data providers that are currently available.
- **Persistence** – an option to restore the data sources after re-starting the application so that there is no need to fetch them again (**Persist opened DataSources**) and an option to restore all the content of the chart and grid tools (**Persist tools content**).

- **Threading** – defines how resources are allocated to the computation (**Batch Pool Size** controls the number of cores used in parallel computation and **Batch Priority** defines the priority of computation over other processes). Changing these values might improve computation speed but also reduce user interface responsiveness.

- **Time Series** – determines the default behaviour of the program when the user double clicks on the data. It may be useful to plot the data, visualise it on a grid, or to perform any pre-specified action, e.g. execute a seasonal adjustment procedure.

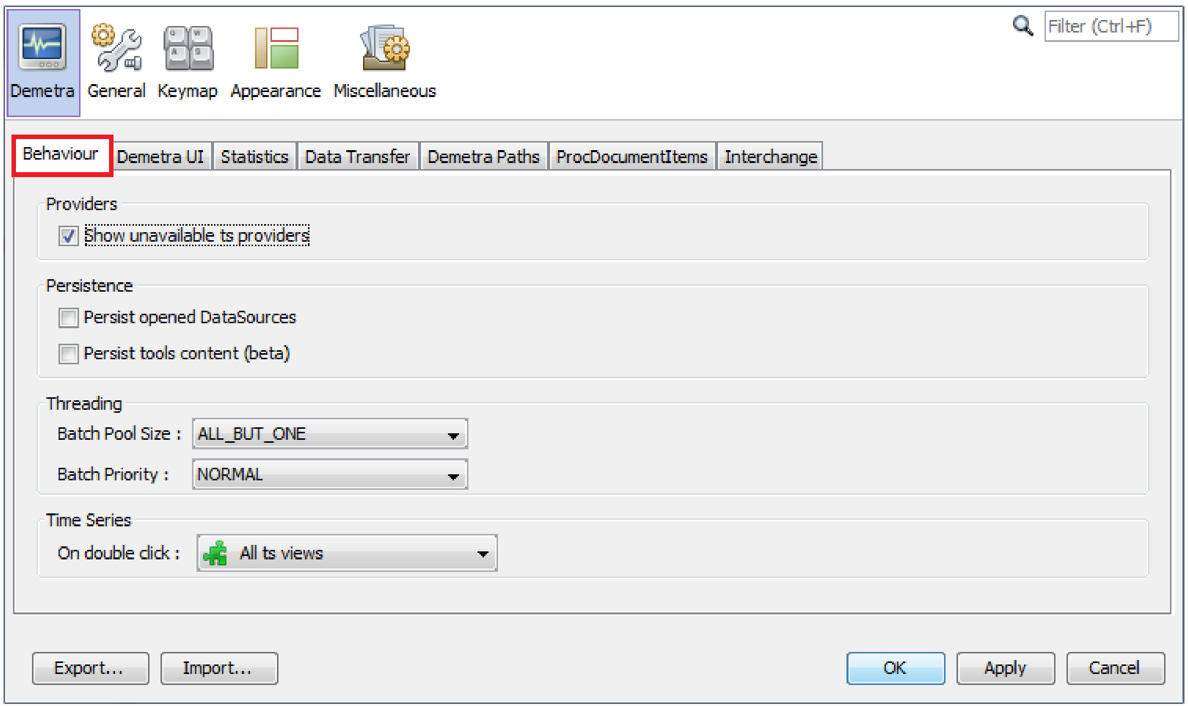



Figure 57: Text

### The content of the *Behavior* tab

The *Demetra UI* tab enables the setting of:

- A default colour scheme for the graphs (**Color scheme**).

- The data format (uses MS Excel conventions). For example, ###,###.#### implies the numbers in the tables and the y-axis of the graphs will be rounded up to four decimals after the decimal point (**Data format**).

- The default number of last years of the time series displayed in charts representing growth rates (**Growth rates**).

- The control of the view of the window for adding pre-specified outliers. (**Pre-specified Outliers**).
- The visibility of the icons in the context menus (**Context Menus**).

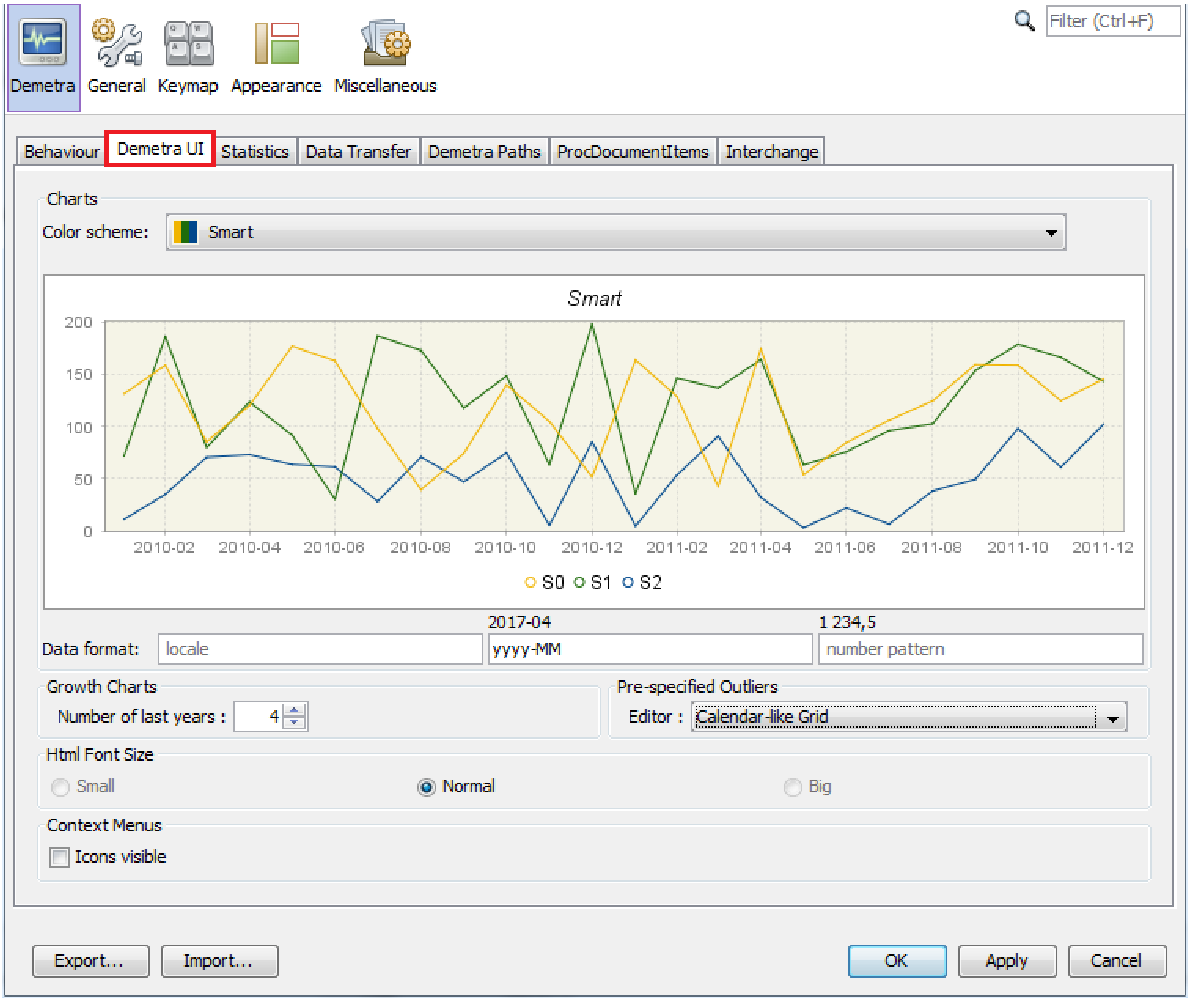



Figure 58: Text

### The content of the *Demetra UI* tab

The *Statistics* tab includes options to control:

- The number of years used for spectral analysis and for model stability (**Default Number of Last Years**);
- The default pre-defined specification for seasonal adjustment (**Seasonal Adjustment**);
- The type of the analysis of revision history (**Revision History**):

- *FreeParameters* – the RegARIMA model parameters and regression coefficients of the RegARIMA model will be re-estimated each time the end point of the data is changed. This argument is ignored if no RegARIMA model is fit to the series.
- *Complete* – the whole RegARIMA model together with regressors will be re-identified and re-estimated each time the end point of the data is changed. This argument is ignored if no RegARIMA model is fitted to the series.
- *None* – the ARIMA parameters and regression coefficients of the RegARIMA model will be fixed throughout the analysis at the values estimated from the entire series (or model span).

- The settings for the quality measures and tests used in a diagnostic procedure:
  - **Default components** – a list of series and diagnostics that are displayed in the **SAProcessing** \(→\) **Output** window. The list of default items can be modified with the respective **Select** button (see figure below)
  - **Diagnostics** – a list of diagnostics tests, where the user can modify the default settings (see figure "*The panel for modification of the settings for the tests in the Basic checks section*" below).

**The *Default components* section on the *Statistics* tab**

An explanation of the list of the series and diagnostics components that are displayed in the *Default components* section can be found here.

To modify the settings for a particular measure, double click on a selected row (select the test's name from the list and click on the working tools button), introduce changes in the pop-up window and click the **OK** button.

To reset the default settings for a given test, select this test from the list and click on the backspace button situated below the working tools button. The description of the parameters for each quality measure and test used in a diagnostic procedure can be found in the output from modelling and the output from seasonal adjustement nodes.

**The panel for modification of the settings for the tests in the *Basic checks* section**

The users can customize the diagnostics and they can specify the default settings for different outputs. Their preferences are saved between different sessions of JDemetra+. This new feature is accessible in the *Statistics* tab of the *Options* panel.

**The settings of the output files**

The *Data Transfer* tab contains multiple options that define the behaviour of the drag and drop and copy-paste actions. To change the default settings, double click on the selected item. Once the modifications are introduced, confirm them with the **OK** button.

**The content of the *Data Transfer* tab**

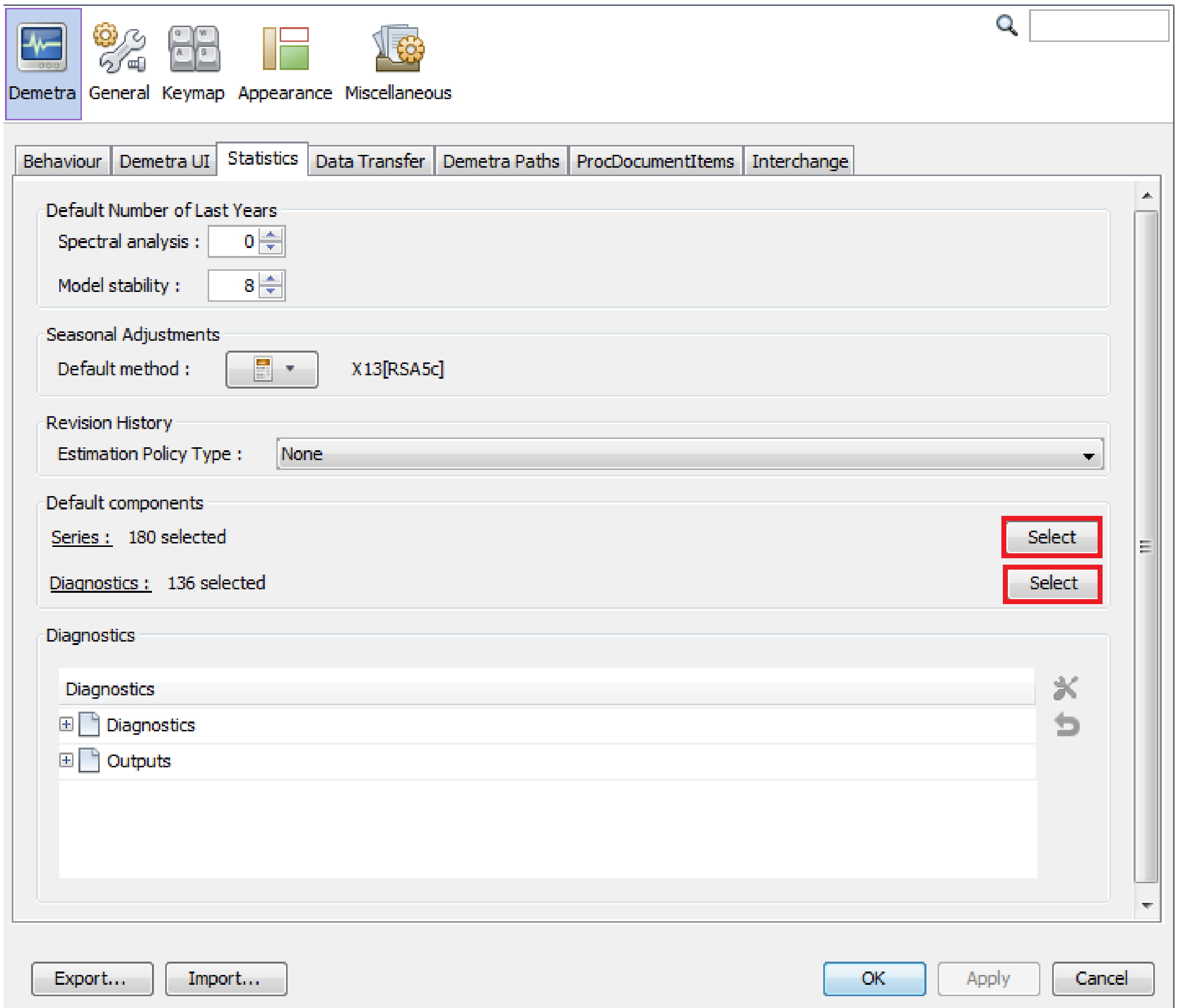



Figure 59: Text

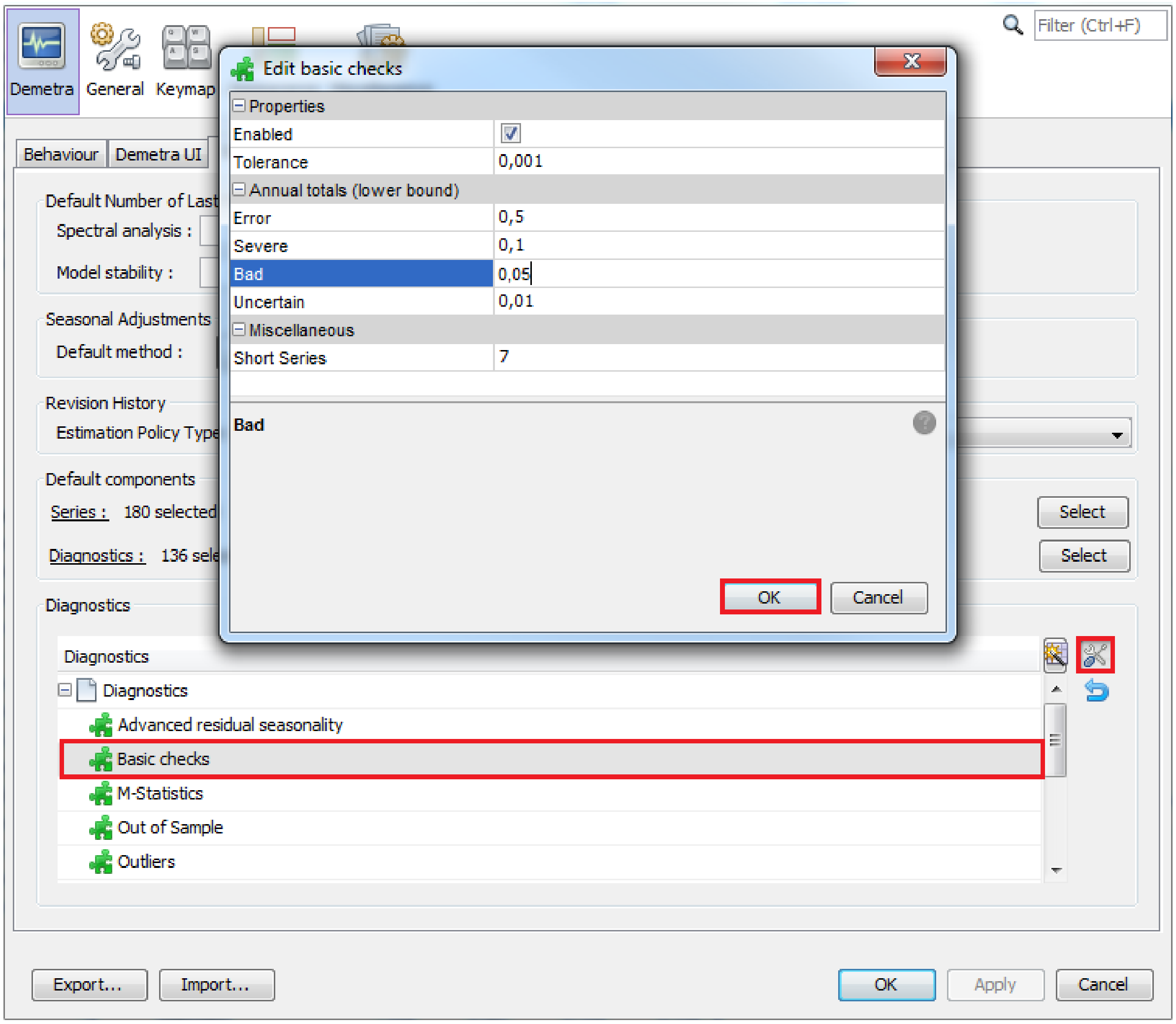



Figure 60: Text

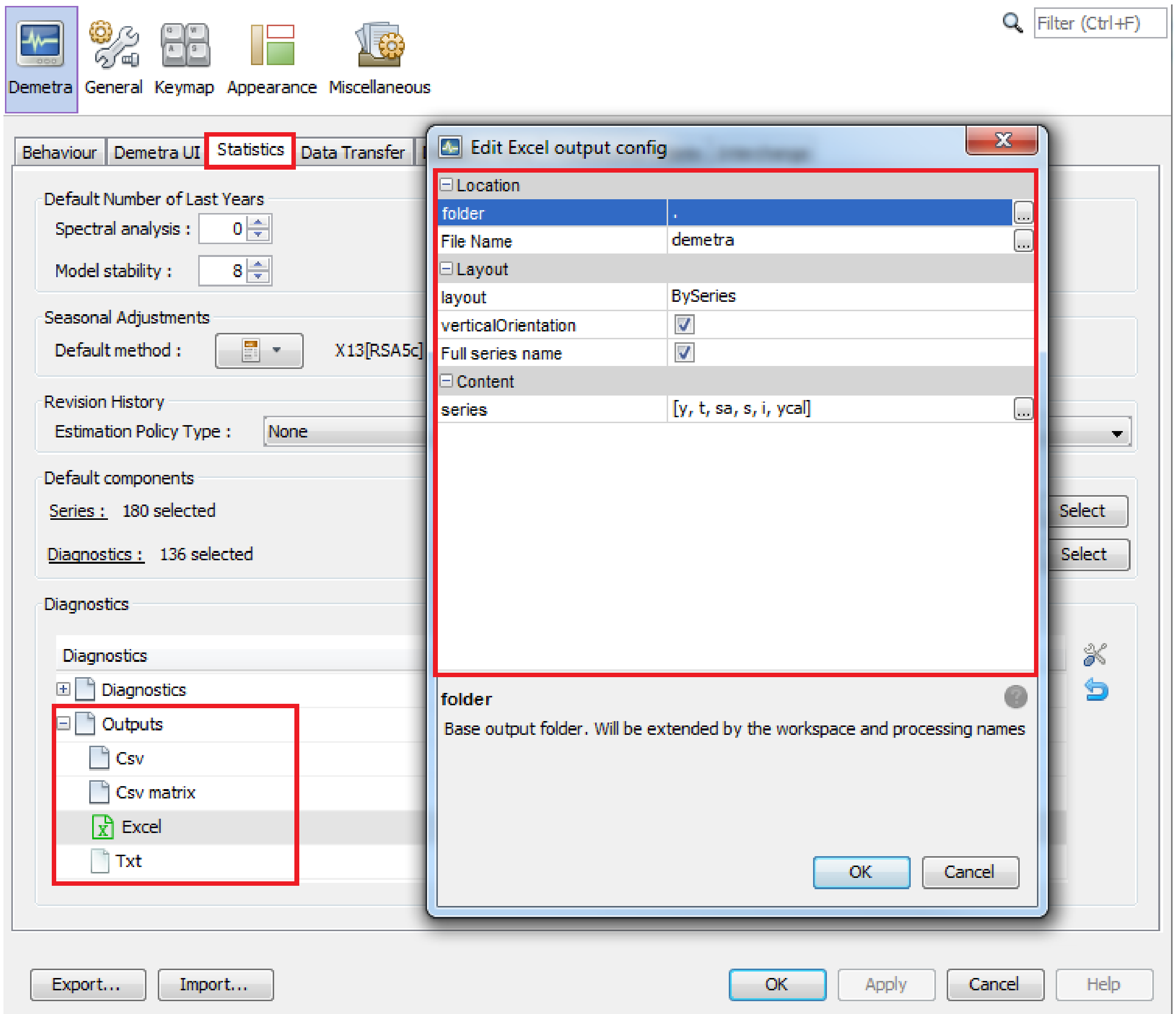



Figure 61: Text

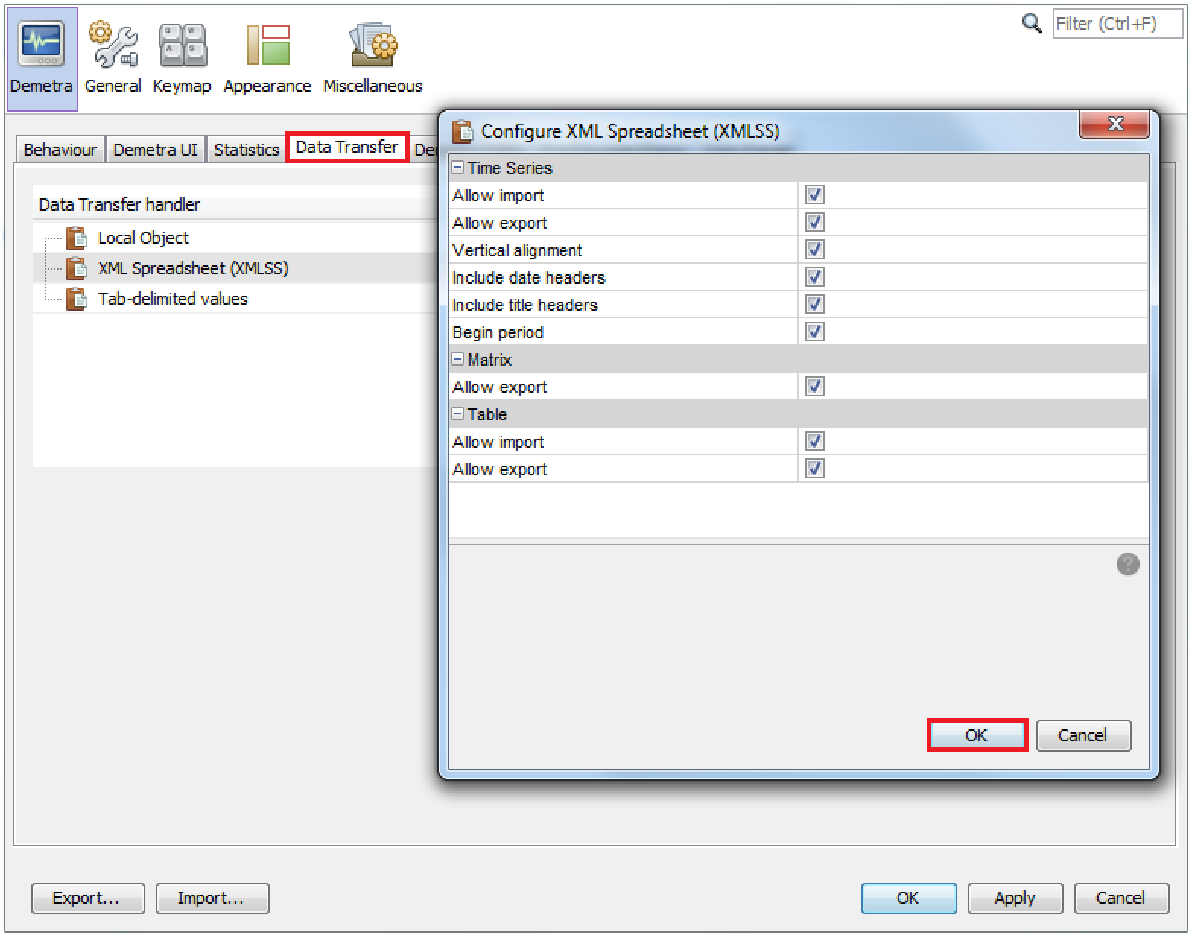



Figure 62: Text

*Demetra Paths* allows the user to specify the relative location of the folders where the data can be found. In this way, the application can access data from different computers. Otherwise, the user would need to have access to the exact path where the data is located. To add a location, select the data provider, click the "+" button and specify the location.

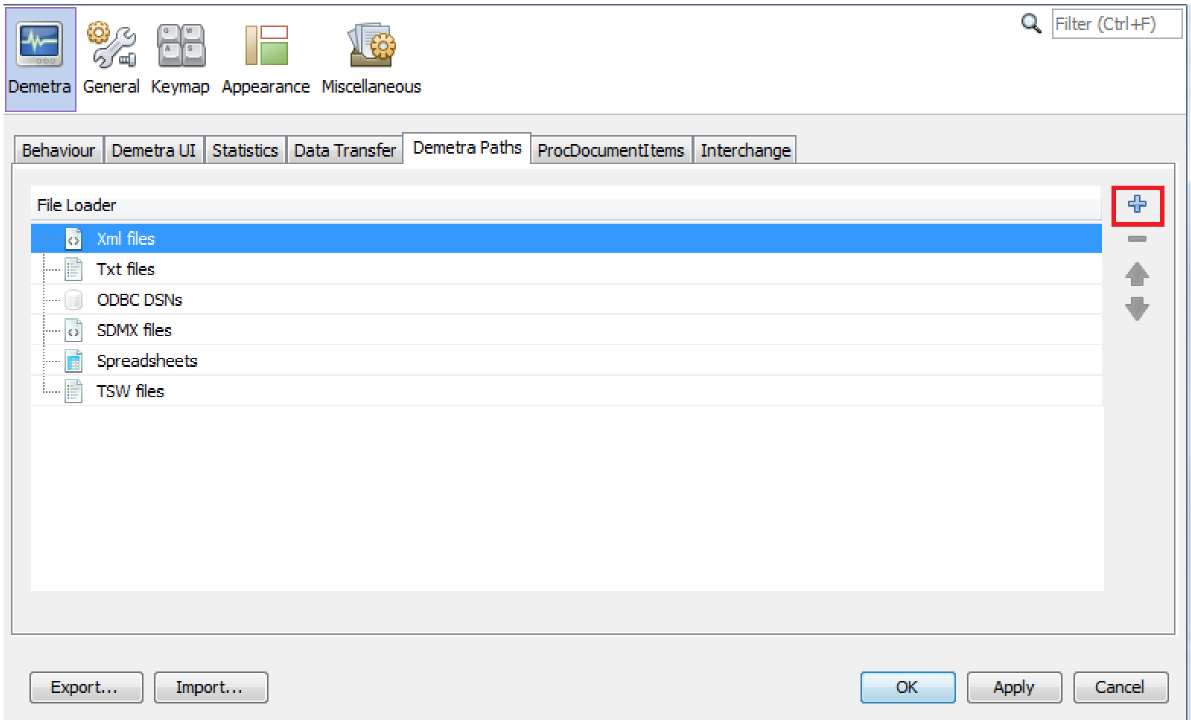



Figure 63: Text

### The content of the *Demetra Paths* tab

*ProcDocumentItems* includes a list of all reports available for processed documents like seasonal adjustment. The *Interchange* tab lists the protocols that can be used to export/import information like calendars, specifications, etc. For the time being, the user cannot customize the way the standard exchanges are done. However, such features could be implemented in plug-ins.

The next section, *General*, allows for the customisation of the proxy settings. A proxy is an intermediate server that allows an application to access the Internet. It is typically used inside a corporate network where Internet access is restricted. In JDemetra+, the proxy is used to get time series from remote servers like .Stat.

### The *General* tab

*Keymap* provides a list of default key shortcuts to access some of the functionalities and it allows the user to edit them and to define additional shortcuts.

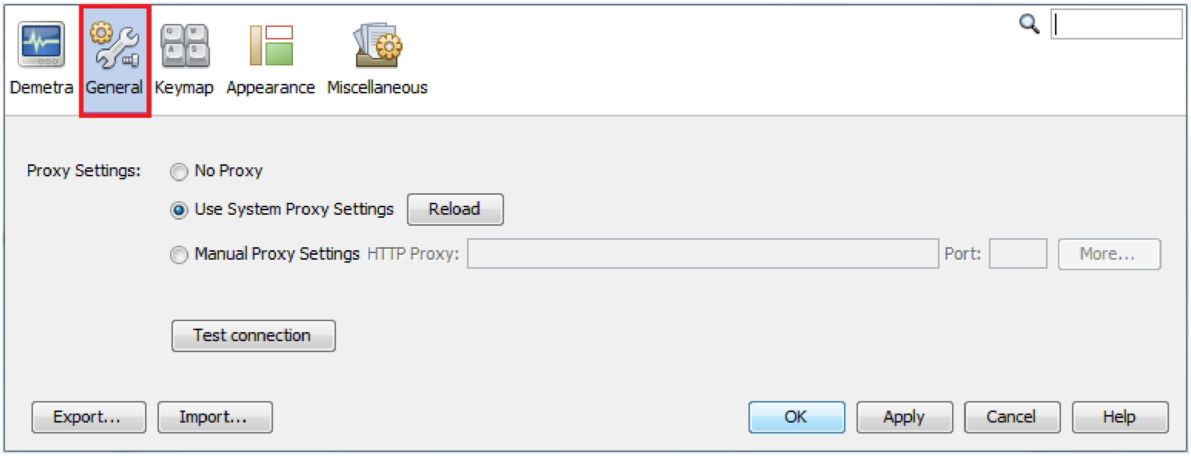



Figure 64: Text

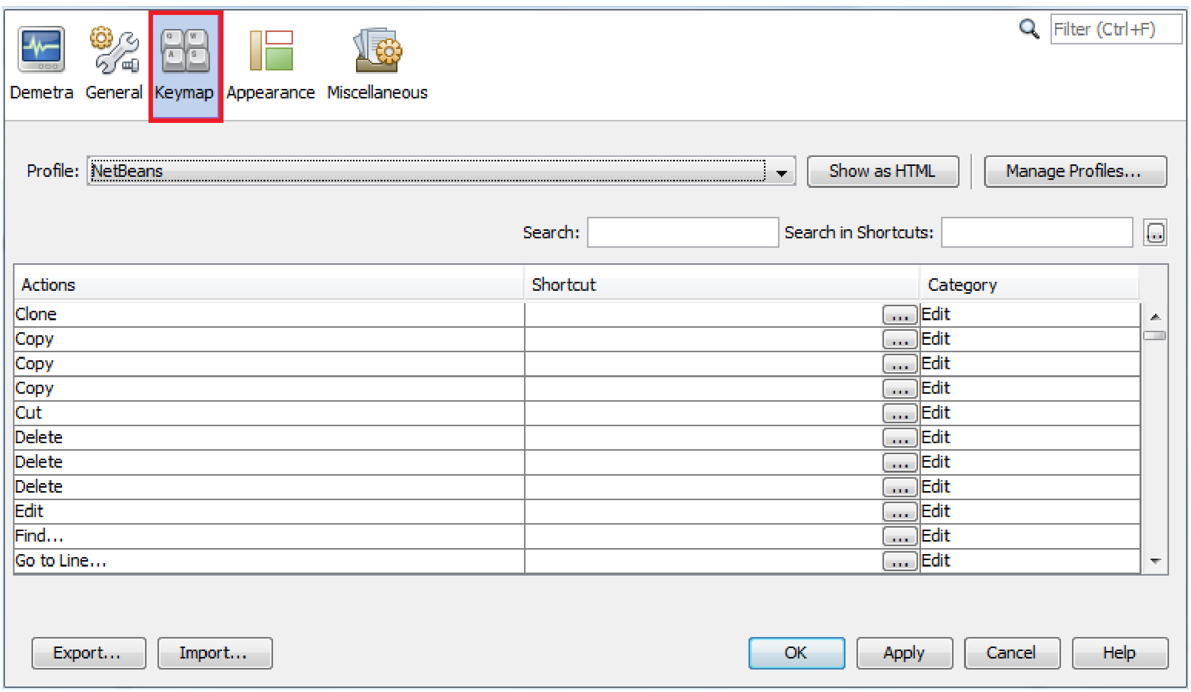



Figure 65: Text

#### The *Keymap* tab

The *Appearance* and *Miscellaneous* tabs are tabs automatically provided by the Netbeans platform. They are not used by JDemetra+.

### Window menu

The *Window* menu offers several functions that facilitate the analysis of data and enables the user to adjust the interface view to the user's needs.

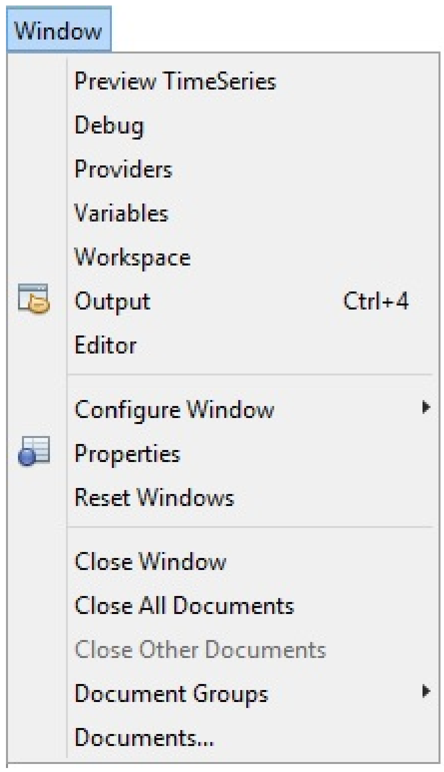



Figure 66: Text

#### The *Window* menu

- **Preview Time Series** – opens a window that plots any of the series the user selects from *Providers*.
- **Debug** – opens a *Preview Time Series* window that enables a fast display of the graphs for time series from a large dataset. To display the graph click on the series in the *Providers* window.
- **Providers** – opens (if closed) and activates the *Providers* window.

- **Variables** – opens (if closed) and activates the *Variable* window.
- **Workspace** – opens (if closed) and activates the *Workspace* window.
- **Output** – a generic window to display outputs in the form of text; useful with certain plug-ins (e.g. tutorial descriptive statistics).
- **Editor** – activates the editor panel (and update the main menu consequently).
- **Configure Window** – enables the user to change the way that the window is displayed (maximise, float, float group, minimise, minimise group). This option is active when some window is displayed in the JD+ interface.
- **Properties** – opens the *Properties* window and displays the properties of the marked item (e.g. time series, data source).
- **Reset Windows** – restores the default JDemetra+ view.
- **Close Window** – closes all windows that are open.
- **Close All Documents** – closes all documents that are open.
- **Close Other Documents** – closes all documents that are open except for the one that is active (which is the last activated one).
- **Document Groups** – enables the user to create and manage the document groups.
- **Documents** – lists all documents that are active.

### Help menu

## Providers window

The *Providers* window presents the list of the data sources and organises the imported series within each data provider.

**The *Providers* window**

The allowed data sources include: * JDBC; * ODBC; * SDMX; * Spreadsheets; * TSW; * TXT; * USCB; * XML.

All standard databases (Oracle, SQLServer, DB2, MySQL) are supported by JDemetra+ via JDBC, which is a generic interface to many relational databases. Other providers can be added by users by creating plugins (see *Plugins* section in the Tools menu). To import data, right-click on the appropriate provider from the *Providers* panel and specify the required parameters. For all providers the procedure follows the same logic. An example is provided here.

The *Providers* window organises data in a tree structure reflecting the manner in which data are presented in the original source. The picture below presents how JDemetra+ visualises

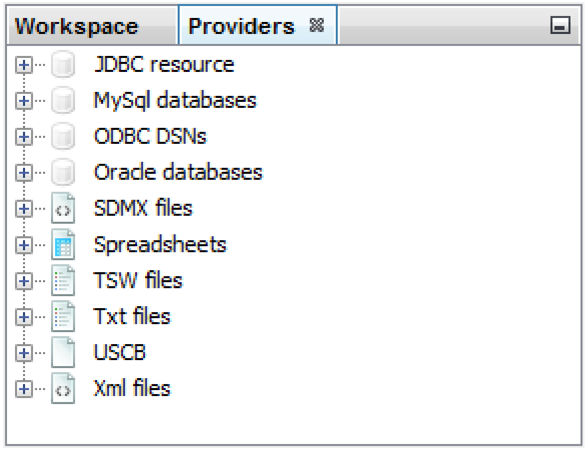



Figure 67: Text

the imported spreadsheet file. If the user expands all the pluses under the spreadsheet all the series within each sheet that has been loaded are visible. Here two time series are visible: *Japan* (under the *Asia* branch) and *United States* (under the *North America* branch) while the *Europe* branch is still folded. The names of the time series have been taken from the column headings of the spreadsheet while the names of the branches come from sheets' names.

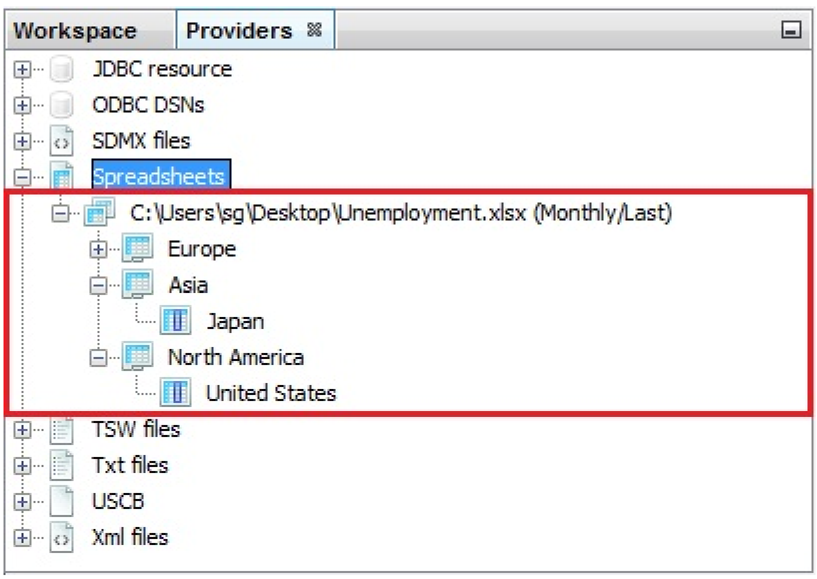



Figure 68: Text

## A structure of a dataset

Series uploaded to the *Providers* window can be displayed, modified and tested for seasonality and used in estimation routines (see Modelling and Seasonal adjustment). The data sources can be restored after re-starting the application so that there is no need to get them again. This functionality can be set in the *Behaviour* tab available at the *Option* item from the *Tools* menu.

## Spreadsheets

The Spreadsheets data source corresponds to the series prepared in the Excel file. The file should have true dates in the first column (or in the first row) and titles of the series in the corresponding cell of the first row (or in the first column). The top-left cell

$$A1$$

can include a text or it can be left empty. The empty cells are interpreted by JDemetra+ as missing values and they can appear in the beginning, in the middle and in the end of time series.

An example is presented below:

| | A | B | C | D |
|---|---|---|---|---|
| 1 | | Currency | M1 | M3 |
| 2 | 31-Dec-96 | 67865,98 | 140428,8 | 23563,88 |
| 3 | 31-Jan-97 | 63680,53 | 139384,1 | 22852,73 |
| 4 | 28-Feb-97 | 63625,74 | 141692,2 | 23513,54 |
| 5 | 31-Mar-97 | 65497,84 | 144931,6 | 24591,24 |
| 6 | 30-Apr-97 | 66635,33 | 148012,8 | 25873,43 |
| 7 | 31-May-97 | 69033,24 | 151700,6 | 25934,04 |
| 8 | 30-Jun-97 | 71672,27 | 154747,6 | 26835,17 |
| 9 | 31-Jul-97 | 74386,61 | 160454,2 | 27841,01 |
| 10 | 31-Aug-97 | 74328,59 | 162408,5 | 27907,99 |
| 11 | 30-Sep-97 | 74658,3 | 165037 | 27630,6 |
| 12 | 31-Oct-97 | 74854,52 | 170176,5 | 27663,32 |
| 13 | 30-Nov-97 | 75283,86 | 173413,4 | 27694,36 |
| 14 | 31-Dec-97 | 79239,77 | 179602,4 | 27255,87 |
| 15 | 31-Jan-98 | 73597,48 | 178239,7 | 26487,78 |
| 16 | 28-Feb-98 | 74417,05 | 180850,5 | 27403,85 |
| 17 | 31-Mar-98 | 75621,69 | 183236,3 | 27286,09 |
| 18 | 30-Apr-98 | 75681,04 | 185907,5 | 28834,44 |
| 19 | 31-May-98 | 78728,63 | 191080,2 | 28860,73 |

Figure 69: Text

**Example of an Excel spreadsheet that can be imported to JDemetra+**

Time series are identified by their names. JDemetra+ derives some information (like data periodicity, starting and ending period) directly from the first column (or from the first row, depending on the chosen data orientation (vertical or horizontal)).

### Import data

To import data from a given data source, click on this data source in the *Providers* window shown below, choose *Open* option and specify the import details, such as a path to a data file. These details vary according to data providers. The example below show how to import the data from an Excel file.

1. From the *Providers* window right-click on the *Spreadsheets* branch and choose *Open* option.

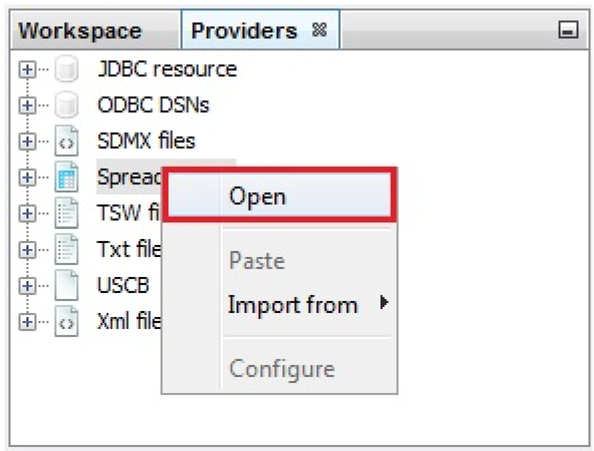



Figure 70: Text

**Data provider available by default**

2. The *Open data source* window contains the following options:

   - **Spreadsheet file** – a path to access the Excel file.
   - **Data format** – the data format used to read dates and values. It includes three fields: *locale* (country), *date pattern* (data format, e.g. *yyyy-mm-dd*), *number pattern* (a metaformat of numeric value, e.g. 0.## represents two digit number).
   - **Frequency** – time series frequency. This can be undefined, yearly, half-yearly, four-monthly, quarterly, bi-monthly, or monthly. When the frequency is set to undefined, JDemetra+ determines the time series frequency by analysing the sequence of dates in the file.
   - **Aggregation type** – the type of aggregation (over time for each time series in the dataset) for the imported time series. This can be *None*, *Sum*, *Average*, *First*, *Last*, *Min* or *Max*. The aggregation can be performed only if the *frequency* parameter is specified. For example, when frequency is set to *Quarterly* and aggregation type is set to *Average*, a monthly time series is transformed to quarterly one with values

that are equal to the one third of the sum of the monthly values that belong to the corresponding calendar quarter.

- **Clean missing** – erases the missing values of the series.

Next, in the *Source* section click the grey "....” button (see below) to open the file.

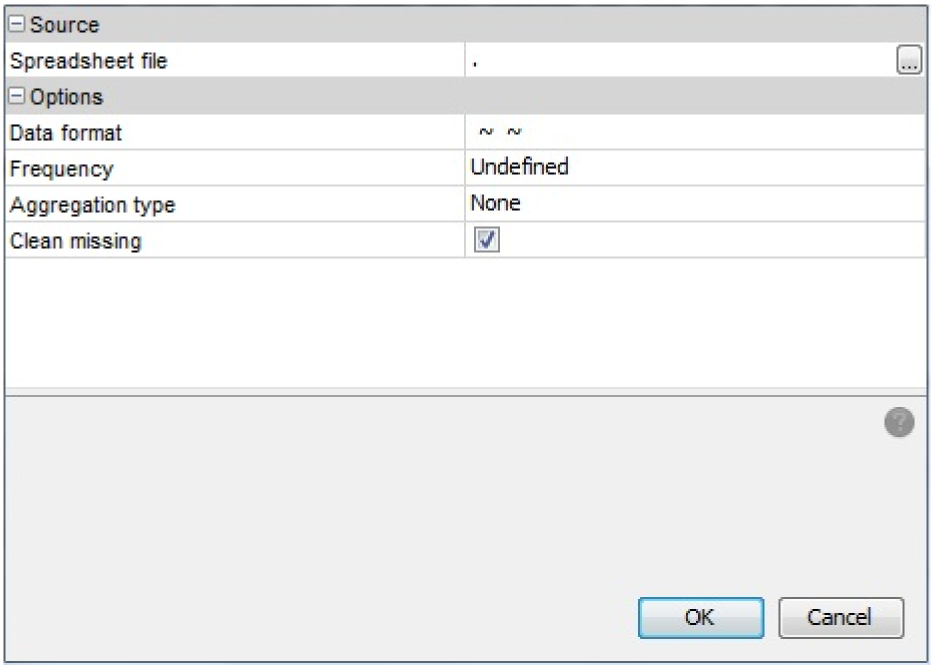



Figure 71: Text

**Data source window**

3. Choose a file and click *OK*.

**Choice of an Excel spreadsheet**

4. The user may specify *Data format*, *Frequency* and *Aggregation type*, however this step is not compulsory. When these options are specified JDemetra+ is able to convert the time series frequency. Otherwise, the functionality that enables the time series frequency to be converted will not be available.

**Options for importing data**

5. The data are organized in a tree structure.

**Dataset structure**

Once the data has been successfully imported, it is available to the user for various analyses (e.g. visualization, modelling, seasonal adjustment, etc.)

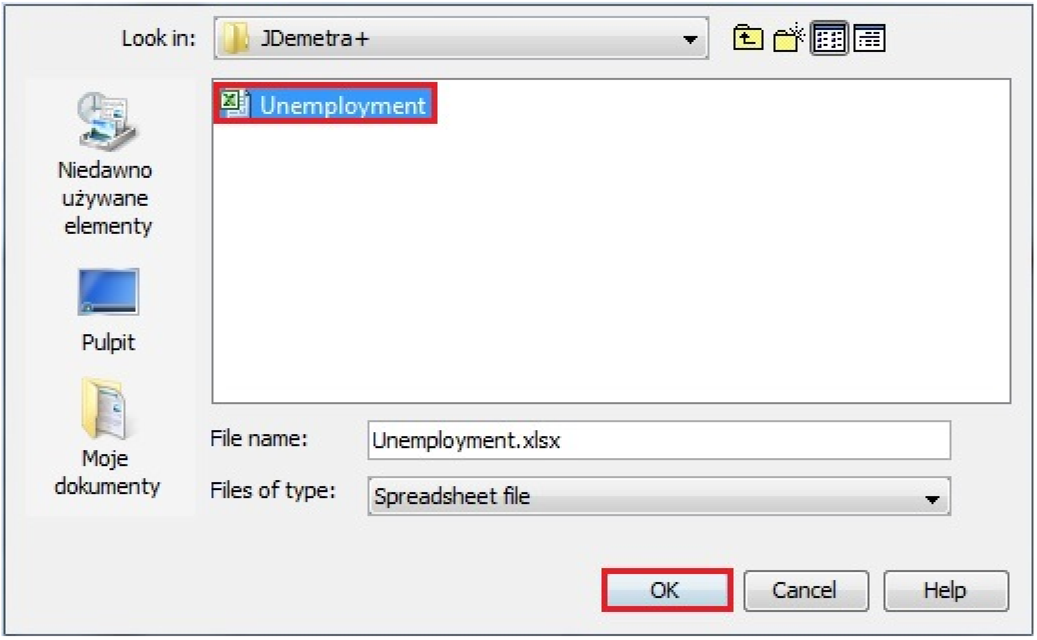



Figure 72: Text

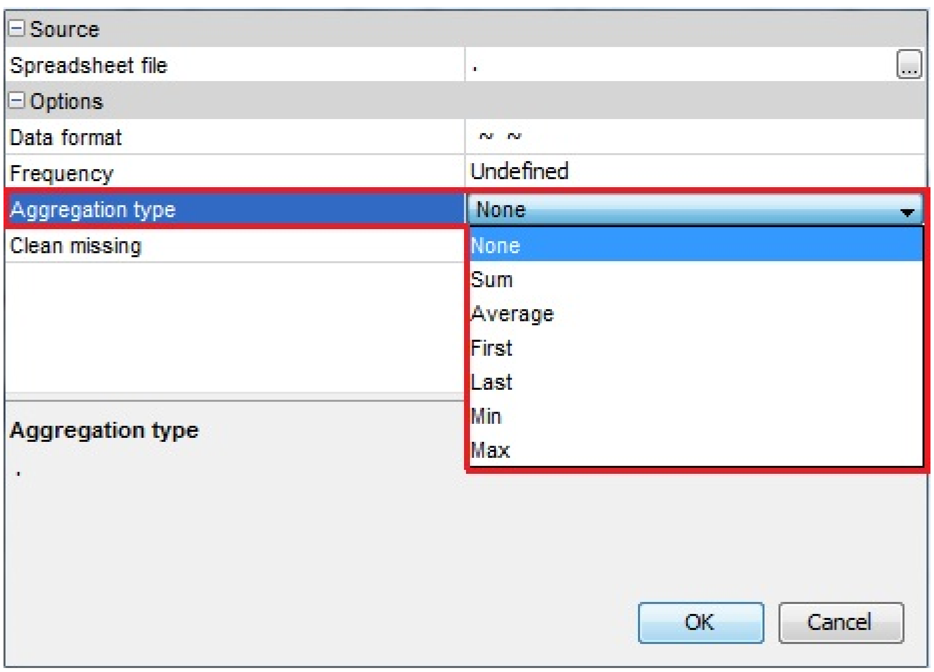



Figure 73: Text

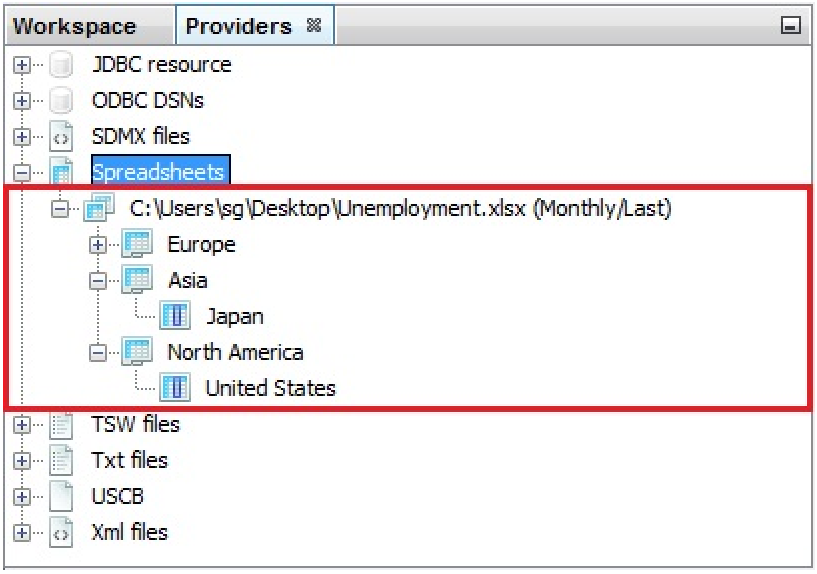



Figure 74: Text

The data are organized in a tree structure. If you expand all the plus-signs under the spreadsheet you will see all the series within each sheet that has been loaded. Here all time series are visible under the Production in construction branch. The names of time series have been taken from the columns' headings of the spreadsheet while the names of the branches come from sheets' names.

## Workspace window

Restructure: present all the nodes

Workspace is a JDemetra+ functionality that stores the work performed by the user in a coherent and structured way. By default, each workspace contains the pre-defined modelling and seasonal adjustment specifications and a basic calendar. A specification is a set of modelling and/or seasonal adjustment parameters. Within the workspace the following items can be saved: * User-defined modelling specifications and seasonal adjustment specifications; * Documents that contain results from time series modelling and output from the seasonal adjustment process; * User-defined calendars; * User-defined regression variables.

Together with the results from modelling and seasonal adjustment, the original data, paths to the input files and parameters of processes are all saved. These results can then be re-opened, updated, investigated and modified in further JDemetra+ sessions.

The workspace saved by JDemetra+ includes: * Main folder containing several folders that correspond to the different types of items created by the user and; * The xml file that enables the user to import the workspace to the application and to display its content.

An example of the workspace is shown in the figure below.

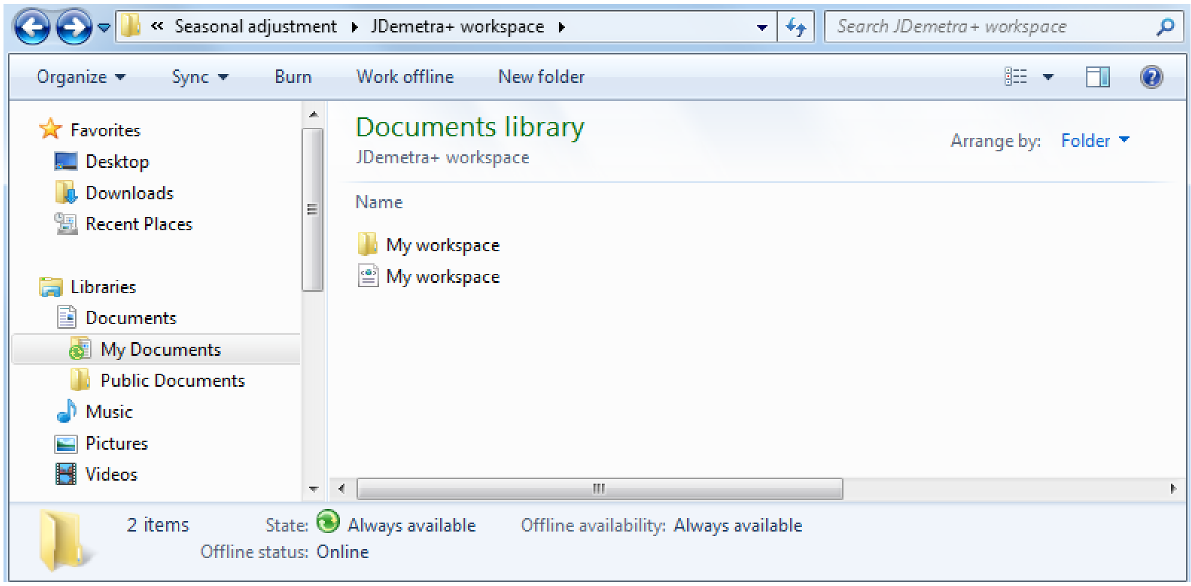



Figure 75: Text

**A workspace saved on PC**

The workspace can be shared with other users, which eases the burden of work with defining specifications, modelling and seasonal adjustment processes.

The content of the workspace is presented in the *Workspace* window. It is divided into three sections: * Modelling (contains the default and user-defined specifications for modelling; and the output from the modelling process) * Seasonal adjustment (contains the default and user-defined specifications for seasonal adjustment and the output from the seasonal adjustment process), * Utilities (calendars and user defined variables).

**The *Workspace* window**

## Results panel

The blank zone in the figure above (on the right of the view) is the location where JDemetra+ displays various windows. More than one window can be displayed at the same time. Windows can overlap with each other with the foremost window being the one in focus or active. The active window has a darkened title bar. The windows in the results panel can be arranged

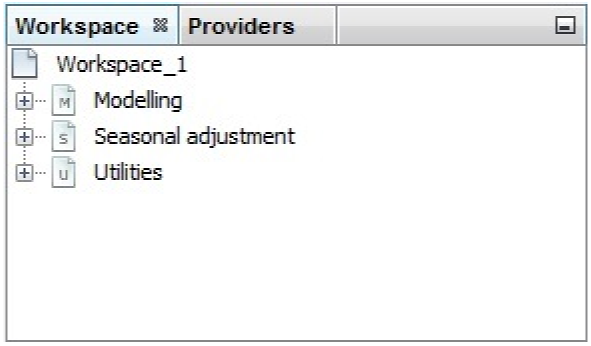



Figure 76: Text

in many different ways, depending on the user's needs. The example below shows one of the possible views of this panel. The results of the user's analysis are displayed in an accompanying window. The picture below shows two panels – a window containing seasonal adjustment results (upper panel) and another one containing an autoregressive spectrum (lower panel).

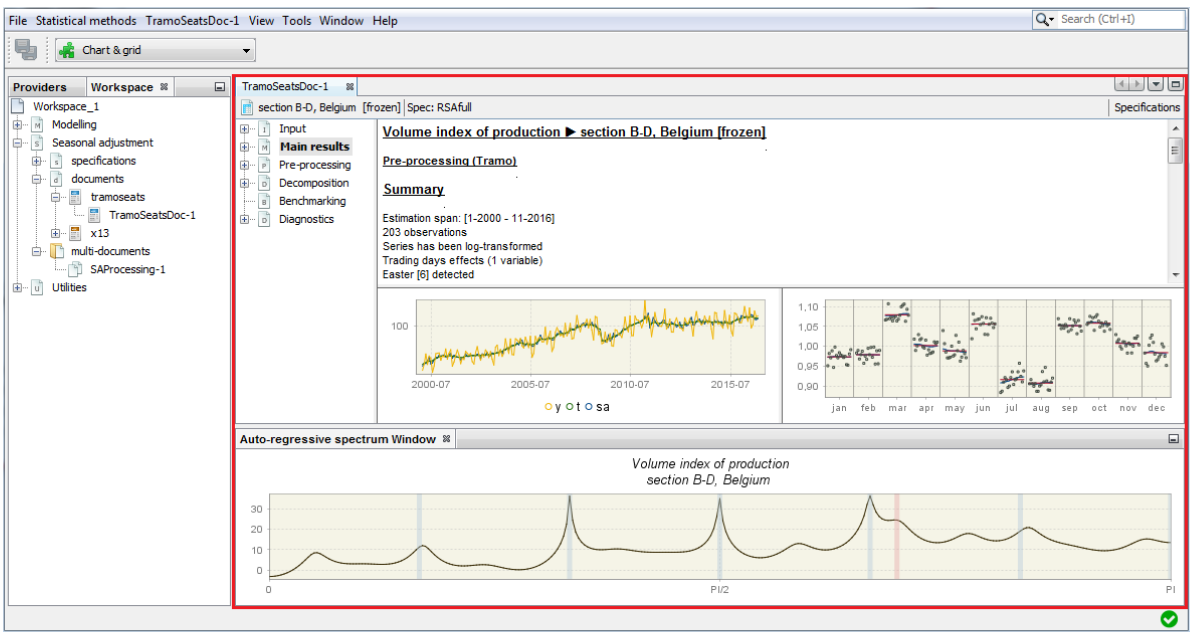



Figure 77: Text

**The *Results* panel filled with two windows**

## Data Visualisation

everything is in Tools > container

## Generating Outpt

add : some explanations + link to cruncher (production chapter)

### Steps

1. Once a seasonal adjustment process for the dataset is performed Go to the TOP menu bar and follow the path: *SAProcessing* → *Output...*
2. In the *Batch output* window the user can specify which output items will be saved and the folder in which JDemetra+ saves the results. It is possible to save the results in the *TXT*, *XLS*, *CSV*, and *CSV matrix* formats. In the first step the user should choose the output format from the list.

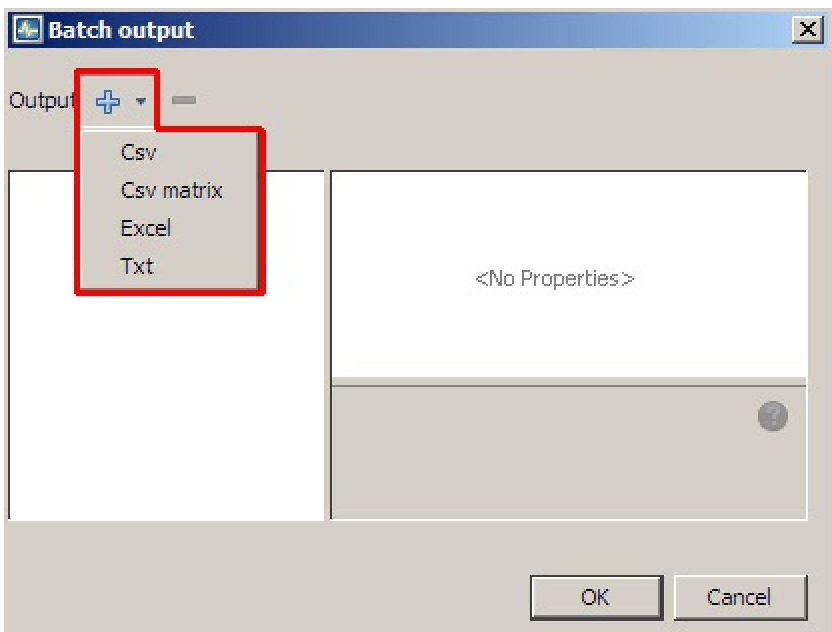



Figure 78: Text

#### Default output formats

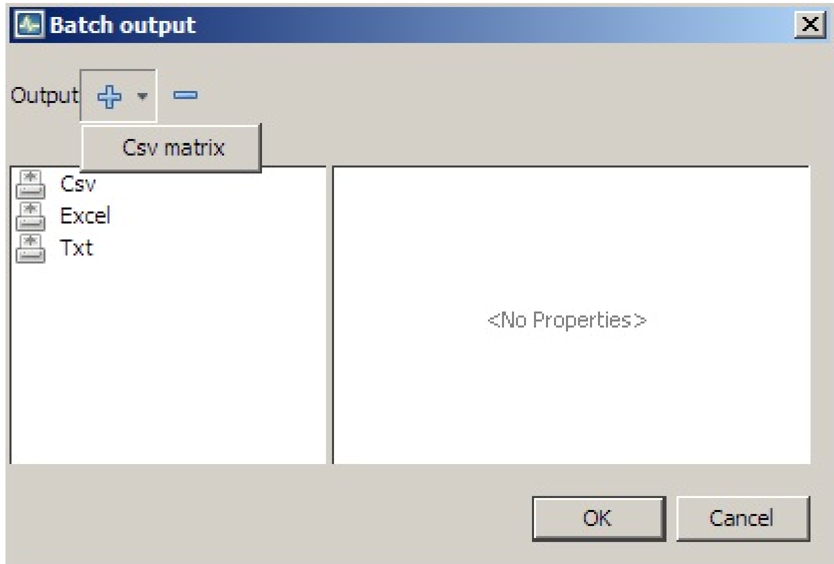



Figure 79: Text

3. The user may choose more than one format as the output can be generated in different formats at the same time.

   **Adding an output format to the list**

4. To display and modify the settings click on the given output format on the list. The available options depend on the output format.

5. For *Csv* format the following options are available: *folder* (location of the file), *file prefix* (name of the file), *presentation* (controls how the output is divided into separate files) and *series* (series included in the file). These options are presented in the next points of this case study.

   **Options for a *Csv* format**

6. The user can define the folder in which the selected results and components will be saved (click the *folder* item and choose the final destination).

   **Specifying a destination folder**

7. With the option *File Prefix* the user can modify the default name of the output saved in the CSV file.

   **Setting a *File Prefix* option**

8. *Layout* controls how the output is divided into separate files. Expand the list to display available options:

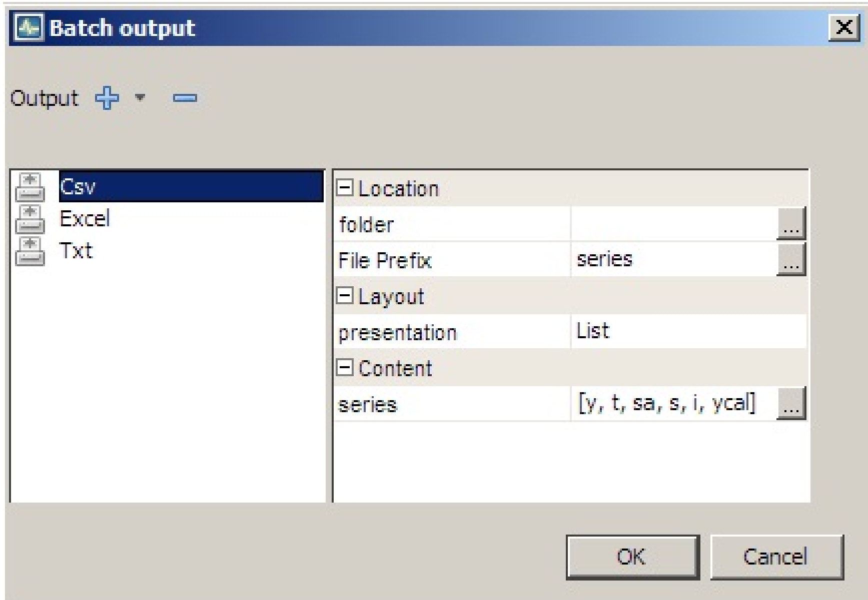



Figure 80: Text

- *HTable* – the output series will be presented in the form of horizontal tables (time series in rows).
- *VTable* – the output series will be presented in the form of vertical tables (time series in columns).
- *List* – the output series will be presented in the form of vertical tables (time series in rows). Apart from that, for each time series each file contains in separate columns: the data frequency, the first year and of estimation span, the first period (month or quarter) of observation span and the number of observations. The files do not include dates.

**Layout options for a *Csv* format**

9. The *Content* section presents a list of series that will be included into a set of output files. To modify the initial settings click on the grey button in the *Content* section. The *CVS-series* window presents two panels: the panel on the left includes a list of all valuable output items. The panel on the right presents the selected output items. Mark the series and use the arrows to change the settings. Confirm your choice with the *OK* button.

**Specifying a content of the output file**

10. Options available for the *XLS* format are the same as for the *TXT* format with an exception of the *Layout* section. The list of available codes in the *Content* section is

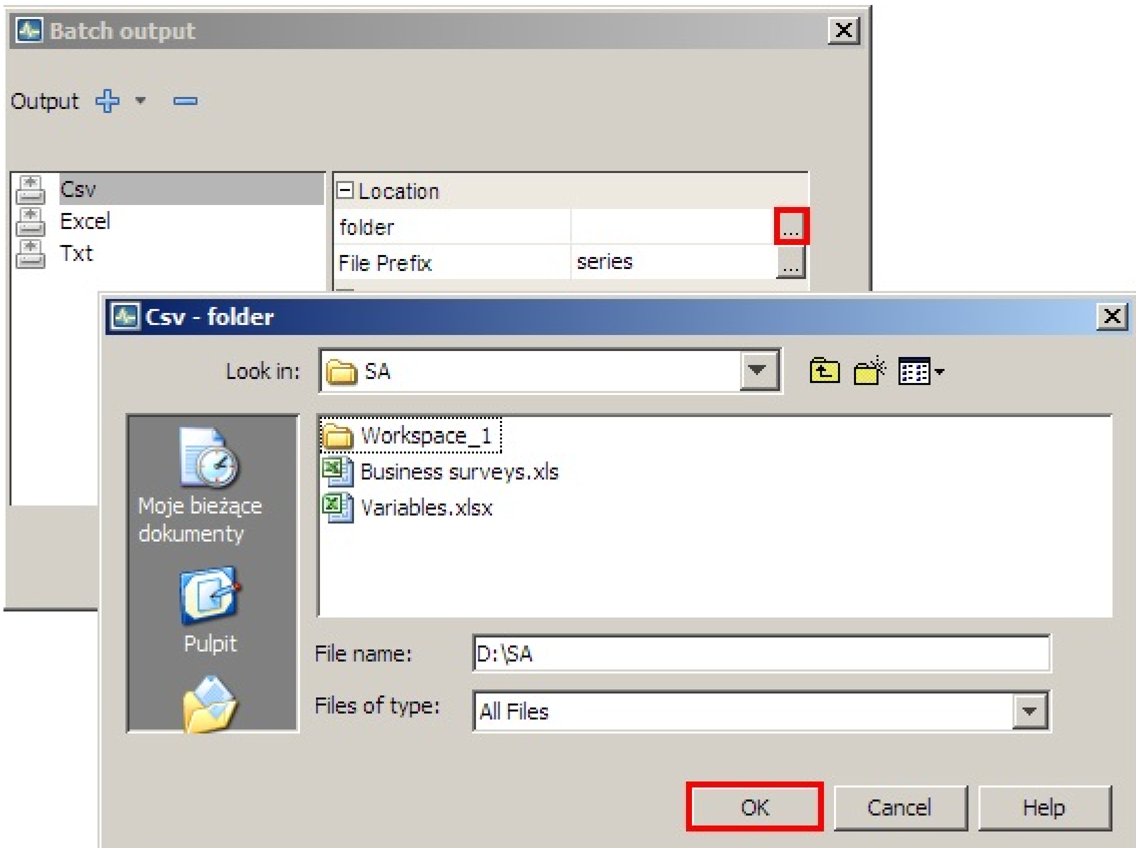



Figure 81: Text

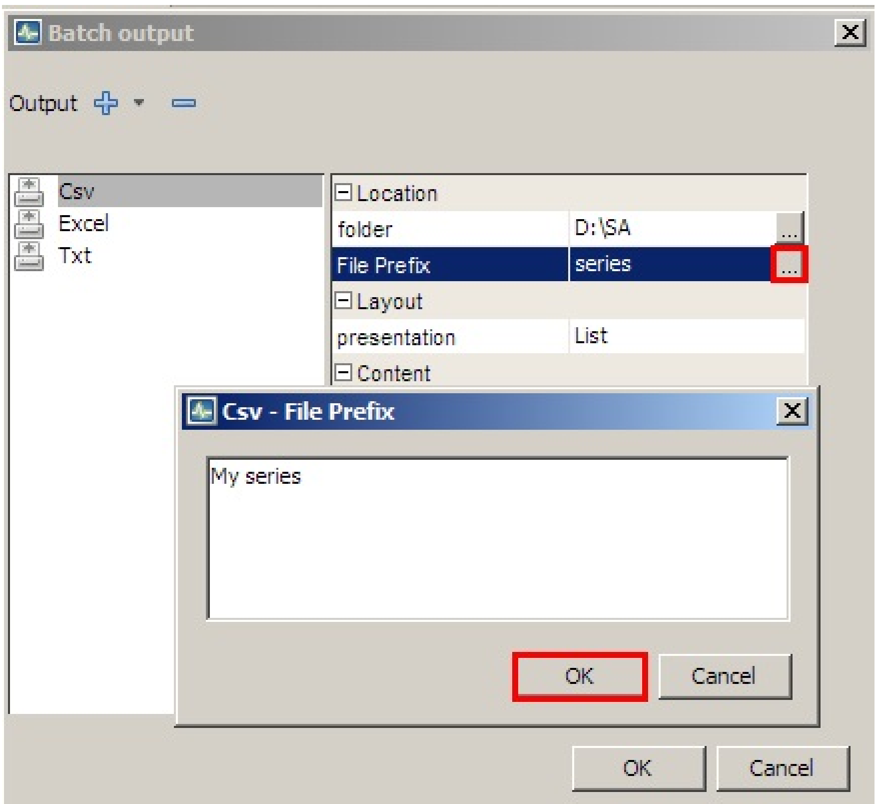



Figure 82: Text

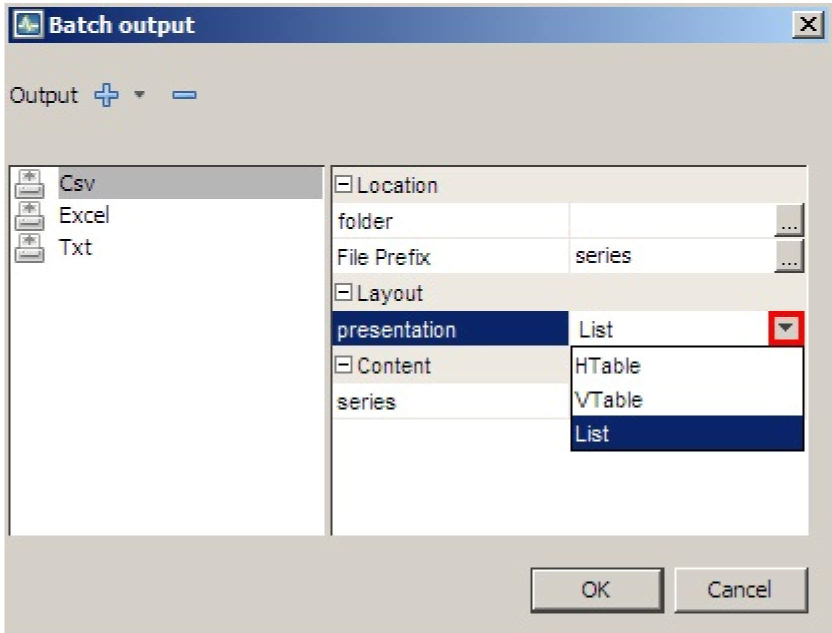



Figure 83: Text

given here.

- *BySeries* – all results for a given time series are placed in one sheet;
- *ByComponent* – results are grouped by components. Each component type is saved in a separate sheet.
- *OneSheet* – all results are saved in one sheet.

**Layout options for an *Excel* format**

11. If the user sets the option layout to *ByComponent*, the output will be generated as follows:

    **An Excel file view for the *ByComponent* option**

12. The option *OneSheet* will produce the following *XLS* file:

    **An Excel file view for the *OneSheet* option**

13. By default, the series in the Excel output files are organised vertically. When the user unmarks the check box the horizontal orientation is used.

    **The *VerticalOrientation* option**

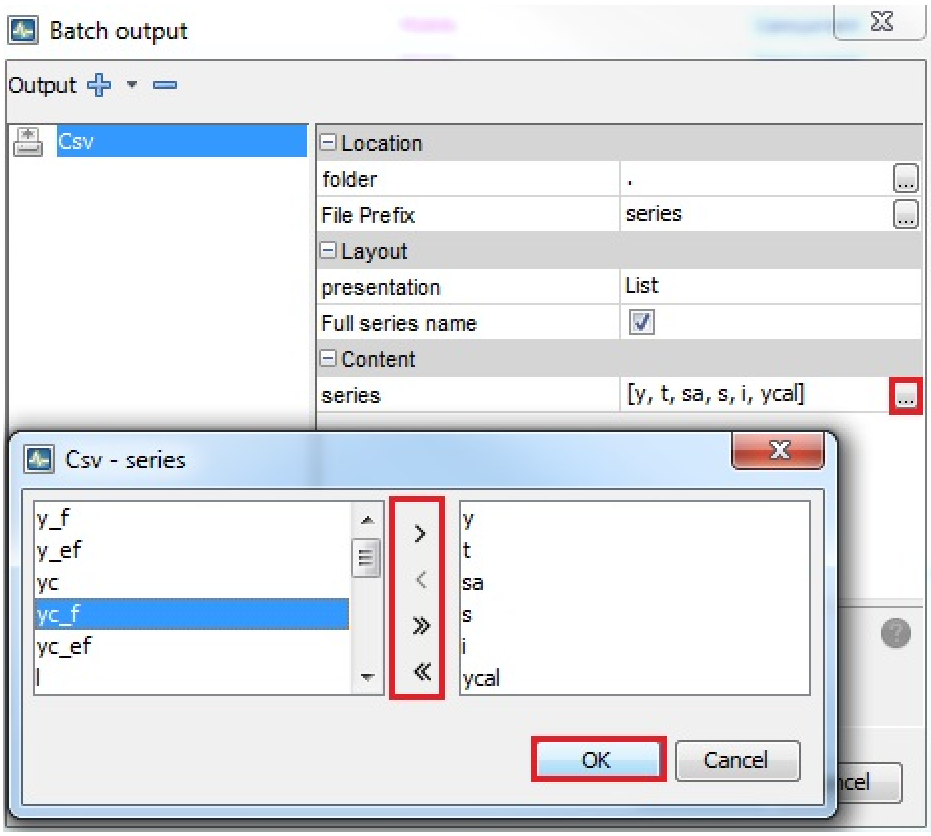



Figure 84: Text

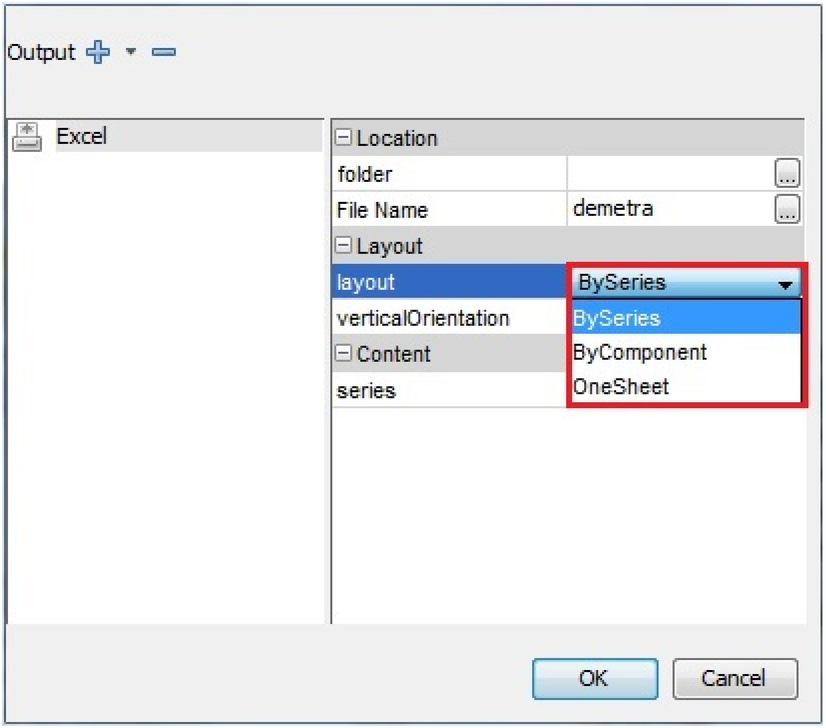



Figure 85: Text

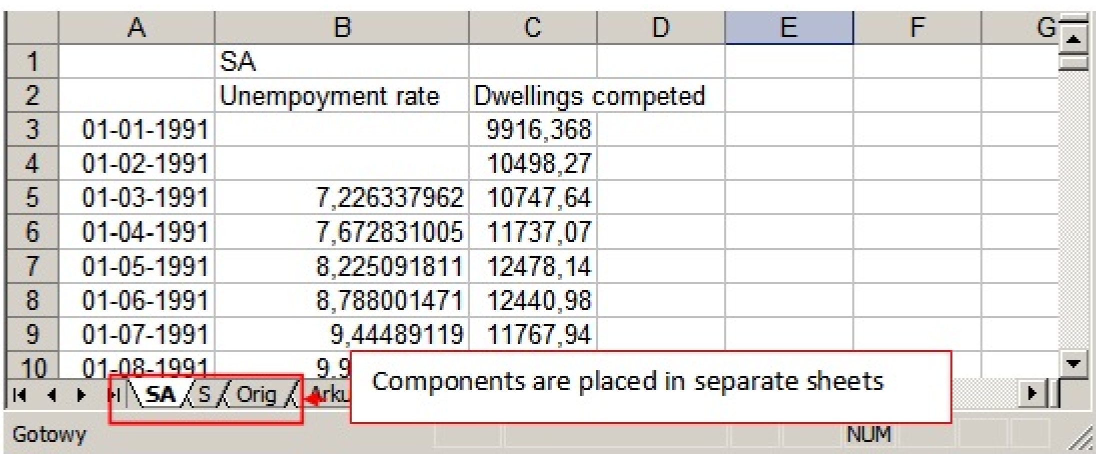
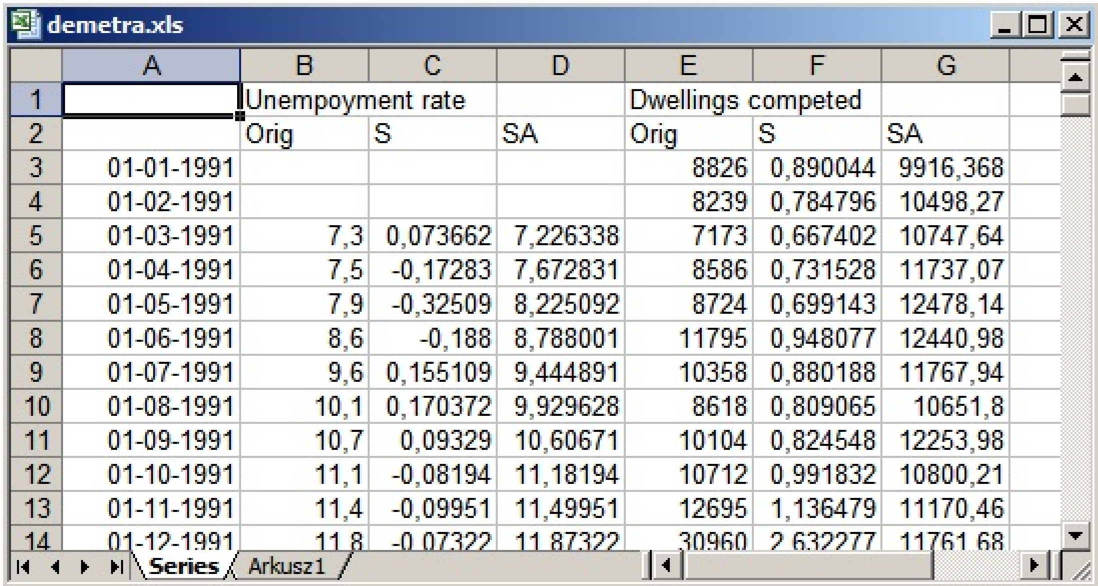



Figure 86: Text



demetra.xls

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | | Unempoyment rate | | | Dwellings competed | | |
| 2 | | Orig | S | SA | Orig | S | SA |
| 3 | 01-01-1991 | | | | 8826 | 0,890044 | 9916,368 |
| 4 | 01-02-1991 | | | | 8239 | 0,784796 | 10498,27 |
| 5 | 01-03-1991 | 7,3 | 0,073662 | 7,226338 | 7173 | 0,667402 | 10747,64 |
| 6 | 01-04-1991 | 7,5 | -0,17283 | 7,672831 | 8586 | 0,731528 | 11737,07 |
| 7 | 01-05-1991 | 7,9 | -0,32509 | 8,225092 | 8724 | 0,699143 | 12478,14 |
| 8 | 01-06-1991 | 8,6 | -0,188 | 8,788001 | 11795 | 0,948077 | 12440,98 |
| 9 | 01-07-1991 | 9,6 | 0,155109 | 9,444891 | 10358 | 0,880188 | 11767,94 |
| 10 | 01-08-1991 | 10,1 | 0,170372 | 9,929628 | 8618 | 0,809065 | 10651,8 |
| 11 | 01-09-1991 | 10,7 | 0,09329 | 10,60671 | 10104 | 0,824548 | 12253,98 |
| 12 | 01-10-1991 | 11,1 | -0,08194 | 11,18194 | 10712 | 0,991832 | 10800,21 |
| 13 | 01-11-1991 | 11,4 | -0,09951 | 11,49951 | 12695 | 1,136479 | 11170,46 |
| 14 | 01-12-1991 | 11,8 | -0,07322 | 11,87322 | 30960 | 2,632277 | 11761,68 |

Series / Arkusz1

Figure 87: Text

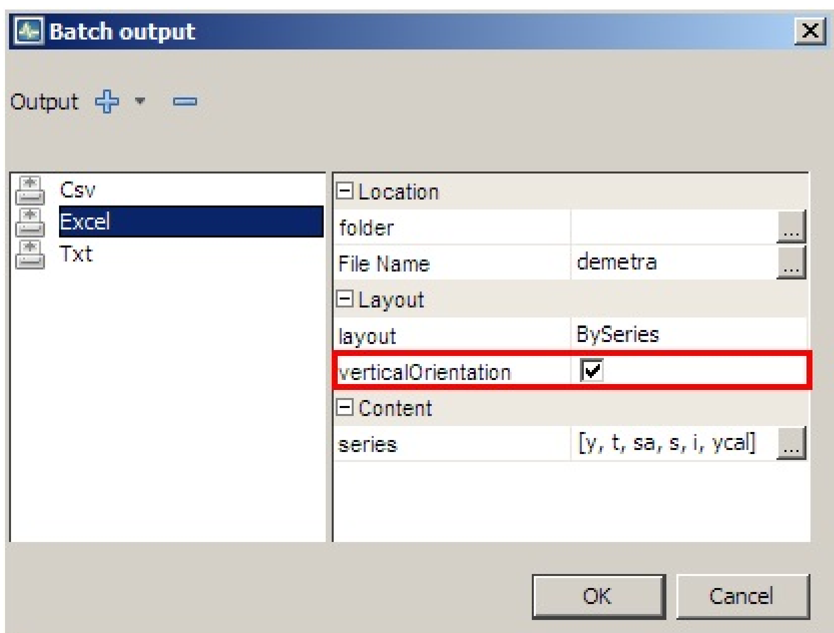



Figure 88: Text

14. In the case of the *TXT* format the only available options are *folder* (location of the file) and *series* (results included in the output file). The list of available codes in the *Content* section is given here.

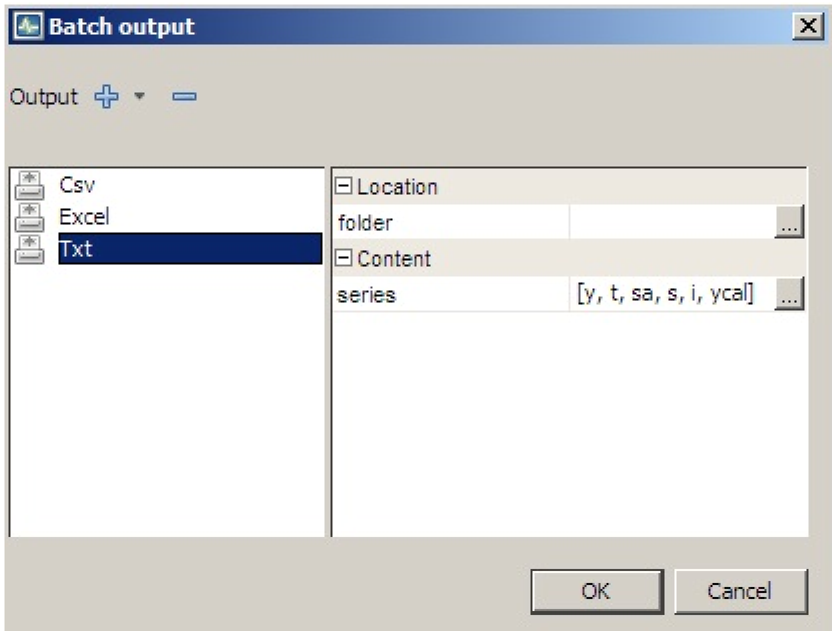



Figure 89: Text

**Options for the *Txt* output**

15. The *CSV matrix* produces the CSV file containing information about the model and quality diagnostics of the seasonal adjustment. The user may generate the list of default items or create their own quality report. By default, all the available items are included in the output. The list of the items is given here.

**List of items available for the *Csv matrix* output type**

16. Once the output settings are selected, click the *OK* button.

**Options for the *Csv matrix* output**

17. For each output JDemetra+ provides information on the status of the operation. An example is presented below.

**Generating output - status information**

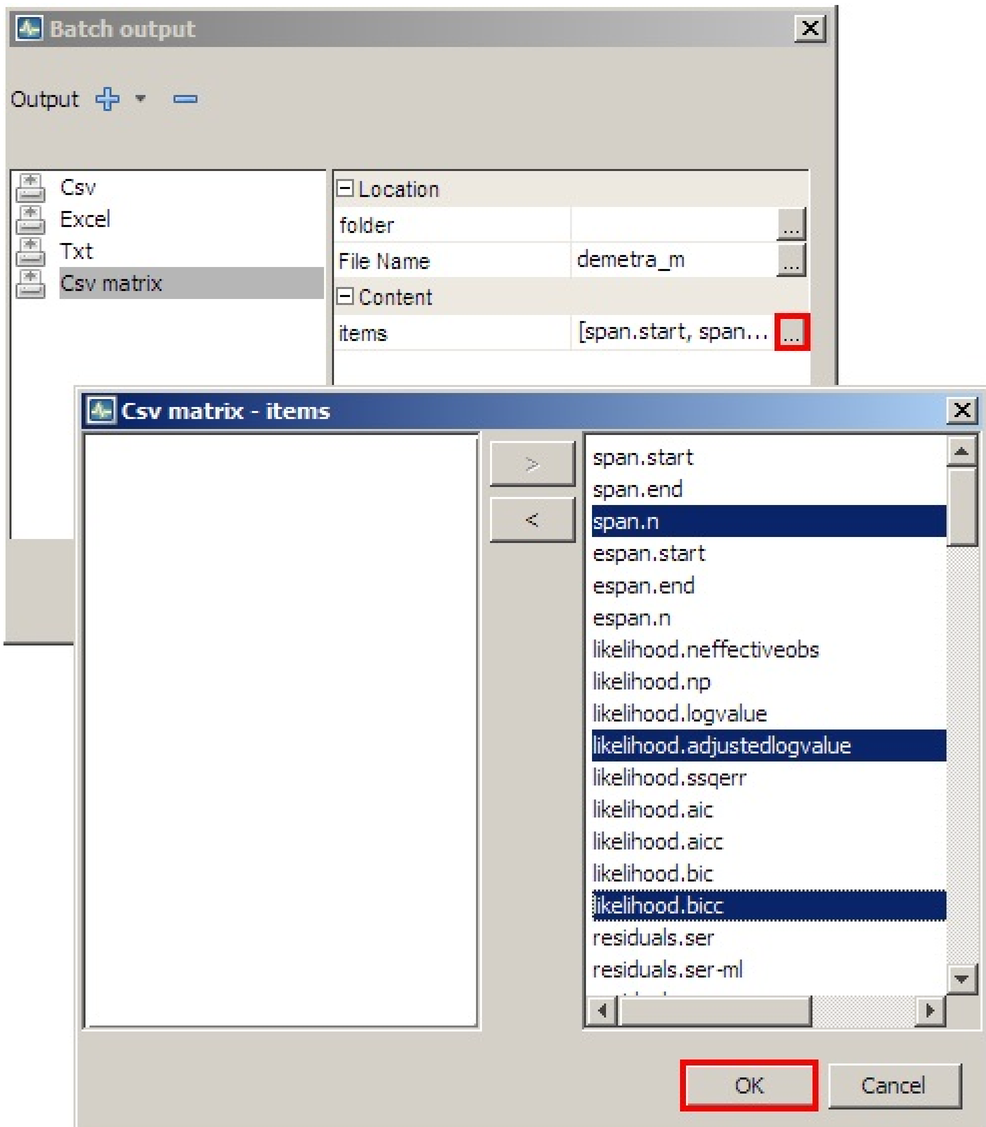



Figure 90: Text

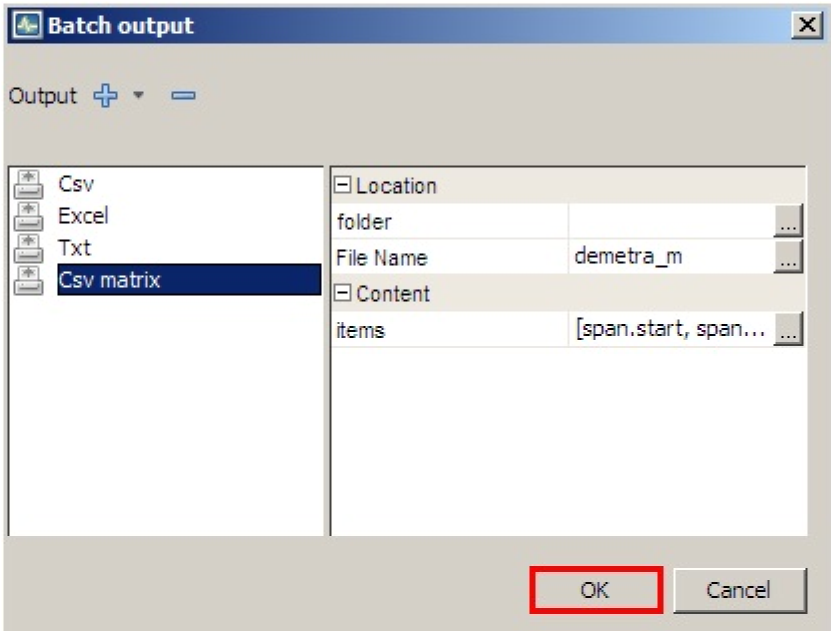



Figure 91: Text

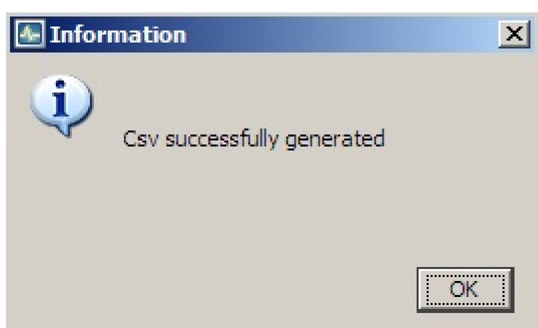



Figure 92: Text

# R packages

## Introduction

Core JDemetra (java) algorithms can be accessied via several tools: - Graphical User Interface GUI - …enhanced with additionnal plug-ins - `R` packages

- core packages
- addtionnal ones

The first package allowing to use JDemetra+ core algorithms {\tt \{RJDemetra\}} {\sf R} was created in 2018 \cite{RJD} and since then the software has been rapidly expanding: algorithms have been extended and tools upgraded. In version 3.0 {\tt \{RJDemetra\}} {\sf R} will be replaced with a family of packages covering each a very specific perimeter. This modular structure makes the functions more readable, fosters evolution and encourages the user to enhance the functions according to its own needs

version 2 vs version 3

here over view + general structure for more details on functions - setp 1 relevant algorithms chapters (SA,…), outlines context - Set 2 R help pages (how to link this ?)

## Available functions and algorithms

What JD+ core algorithms can be accessed ? V2 and V3 ?

### Seasonal adjustment

Table 24: Main Algorithms accessible via R packages

| Domain | Algorithm | Package | Comments |
|---|---|---|---|
| Seasonal Adjustment | X13-Arima | | |
| | Tramo-Seats | | |
| | | rjd3highfreq | |

| Domain | Algorithm | Package | Comments |
|---|---|---|---|
| | STL() | rjd3stl | |
| | Basic Structural Models | rjd3sts | |
| Trend estimation | Moving Averages, Local Polynomial | rjdfilters | |
| | idem | rjd3highfreq | |
| Benchmarking and TD | rjd3bench | | |

when package mentionned in table, add link to the relavant detailed part

## Filtering and Trend estimation

## Benchmarking and Temporal disaggregation

## State space framework

## Running the cruncher and generating quality report

the cruncher nb

| Package | JD+ version | Comments |
|---|---|---|
| rjwsacruncher | | |
| JDcruncher | | |

## Wrangling JD+ workspaces

A workspace (link) is..here…

| Package | JD+ version | Comments |
|---|---|---|
| rjdworkspace | | |
| rjd | | |

### Generating additionnal output

enhancing plots, prints

| Package | JD+ version | Comments |
|---|---|---|
| rjdmarkdown | | |
| ggdemetra | | |
| ggdemetra3 | | |
| rjdqa | | |

## Structure of the packages

see if relevant part

form of output type of functions how v3 differs from v2

### RJDemetra and JD+ version 2.2.3

Note on RJDemetra

examples in A_sa : v2 and v3 rest of the chapter on version 3

(ecriture standard des noms des packages..cf K) a suite (order)

dependencies

utility packages

general output organisation

### version 3

### Mixing v2 and V3

what is doable ?

## Installation procedure

### version 2

```
install.packages("RJDemetra")
```

### version 3

```
#install.packages("remotes")
remotes::install_github("palatej/rjd3toolkit")
remotes::install_github("palatej/rjd3modelling")
remotes::install_github("palatej/rjd3sa")
remotes::install_github("palatej/rjd3arima")
remotes::install_github("palatej/rjd3x13")
remotes::install_github("palatej/rjd3tramoseats")
remotes::install_github("palatej/rjdemetra3")
remotes::install_github("palatej/rjdfilters")
remotes::install_github("palatej/rjd3sts")
remotes::install_github("palatej/rjd3highfreq")
remotes::install_github("palatej/rjd3stl")
remotes::install_github("palatej/rjd3bench")
remotes::install_github("AQLT/ggdemetra3")
```

Below you will find a comprehensive list and main functions by categories

for detailed function, you can refer to each package's own R documentation

## Utility packages

might not be the right title relevant for V3 only ? cf estp adv, arima aux functions

### rjd3toolkit

Contains utility functions used in other `rjd` packages and several functions to perform tests.

Tests - Normality tests: Bowman-Shenton (`bowmanshenton()`), Doornik-Hansen (`doornikhansen()`), Jarque-Bera (`jarquebera()`)

- Runs tests (randomness of data): mean or the median (`testofruns()`) or up and down runs test (`testofupdownruns()`)

- autocorrelation functions (usual, inverse, partial)
- `aggregate()` to aggregate a time series to a higher frequency

Example

```
library(rjd3toolkit)
set.seed(100)
x = rnorm(1000);y = rlnorm(1000)
bowmanshenton(x) # normal distribution
bowmanshenton(y) # log-normal distribution
testofruns(x) # random data
testofruns(y) # random data
testofruns(1:1000) # non-random data
autocorrelations(x)
autocorrelations.inverse(x)
autocorrelations.partial(x)
```

### rjd3modelling

Purpose : creating input variables (regressors) for to be used in Reg-Arima modelling

- create user-defined calendar and trading-days regressors: `calendar.new()` (create a new calendar), `calendar.holiday()` (add a specific holiday, e.g. christmas), `calendar.easter()` (easter related day) and `calendar.fixedday()`) (for tecnical details on building those regressors link to calendar chapter)
- outliers regressors (AO, LS, TC, SO, Ramp, intervention variables), calendar related regressors (stock, leap year, periodic dummies and contrasts, trigonometric variables) -> to be added quadratic ramps
- Range-mean regression test (to choose log transformation), Canova-Hansen (`td.ch()`) and trading-days f-test (`td.f()`)
- specification functions for `rjd3x13` and `rjd3tramoseats`

more explanations and examples needed here

#### Example of calendar specification

move to calendar chapter

```r
library(rjd3modelling)
fr_cal <- calendar.new()
calendar.holiday(fr_cal, "NEWYEAR")
calendar.holiday(fr_cal, "EASTERMONDAY")
calendar.holiday(fr_cal, "MAYDAY")
calendar.fixedday(fr_cal, month = 5, day = 8,
                  start = "1953-03-20")
# calendar.holiday(fr_cal, "WHITMONDAY") # Equivalent to:
calendar.easter(fr_cal, offset = 61)

calendar.fixedday(fr_cal, month = 7, day = 14)
# calendar.holiday(fr_cal, "ASSUMPTION")
calendar.easter(fr_cal, offset = 61)
calendar.holiday(fr_cal, "ALLSAINTSDAY")
calendar.holiday(fr_cal, "ARMISTICE")
calendar.holiday(fr_cal, "CHRISTMAS")
```

Use `holidays()` to get the days of the holidays and `htd()` to get the trading days regressors

```r
holidays(fr_cal, "2020-12-24", 10,single = T)
s = ts(0, start = 2020, end = c(2020, 11), frequency = 12)
# Trading-days regressors (each day has a different effect, sunday as contrasts)
td_reg <- htd(fr_cal, s = s, groups = c(1, 2, 3, 4, 5, 6, 0))
# Working-days regressors (Monday = ... = Friday; Saturday = Sunday = contrasts)
wd_reg <- htd(fr_cal, s = s, groups = c(1, 1, 1, 1, 1, 0, 0))
# Monday = ... = Friday; Saturday; Sunday = contrasts
wd_reg <- htd(fr_cal, s = s, groups = c(1, 1, 1, 1, 1, 2, 0))
wd_reg
```

more explanations on contrasts and references links to calendar correction chapters

#### Example of outliers

```r
s = ts(0, start = 2000, end = 2005, frequency = 12)
ao = ao.variable(s = s, date = "2001-03-01")
ls = ls.variable(s = s, date = "2001-01-01")
tc = tc.variable(s = s, date = "2001-01-01", rate = 0.7)
so = so.variable(s = s, date = "2003-05-01")
ramp = ramp.variable(s = s, range = c("2001-01-01","2001-12-01"))
plot(ts.union(ao, ls, tc, so, ramp), plot.type = "single",
```

```
      col = c("red","lightgreen","orange","blue","black"))
```

### rjd3sa package

scope of up-garde vs V2

#### Seasonality tests

(for each code snippet example and link to M_Test chapter)

- Canova-Hansen (`seasonality.canovahansen()`)
- X-12 combined test (`seasonality.combined()`)
- F-test on seasonal dummies (`seasonality.f()`)
- Friedman Seasonality Test (`seasonality.friedman()`)
- Kruskall-Wallis Seasonality Test (`seasonality.kruskalwallis()`)
- Periodogram Seasonality Test (`seasonality.periodogram()`)
- QS Seasonality Test (`seasonality.qs()`)

EXP : Always correct the trend and remove the mean before seasonality tests:

```
library(rjd3sa)
y = diff(rjd3toolkit::ABS$X0.2.09.10.M, 1); y = y - mean(y)
seasonality.f(y, 12)
seasonality.friedman(y, 12)
seasonality.kruskalwallis(y, 12)
seasonality.combined(y, 12)
```

# Seasonal adjustment packages

## rjd3arima

`rjd3arima` is devoted to formatting the output of Arima related results

## rjd3x13 and rjd3tramoseats

attention : splitting between package descrption here and practical usage in A_sa chapter, here general philosphy of the package

### Common functions

In `RJDemetra` you have one function to set the specification (`regarima_spec_x13()`, `regarima_spec_tramo()`, `x13_spec()` and `tramoseats_spec()`) now one function for each part of the specification

Common functions (defined in `rjd3modelling`) to set the specification of the preprocessing:

`set_arima()`, `set_automodel()`, `set_basic()`, `set_easter()`, `set_estimate()`, `set_outlier()`, `set_tradingdays()`, `set_transform()`, `add_outlier()` and `remove_outlier()`, `add_ramp()` and `remove`_ramp()‘

- warning `add_usrdefvar()` not yet available

### rjd3x13

Main functions:

- Specification: created with `spec_x11_default()`, `spec_x13_default()`, `spec_regarima_default()` and customized with `rjd3arima` functions + `set_x11()`
- Apply model with `x11()`, `x13()`, `fast.x13()`, `regarima()`, `fast.regarima()`
- Refresh policies: `regarima.refresh()` and `x13.refresh()`

### rjd3tramoseats

Main functions:

- Specification: created with `spec_tramoseats_default()`, `spec_tramo_default()` and customized with `rjd3arima` functions + `set_seats()`
- Apply model with `tramoseats()`, `fast.tramoseats()`, `tramo()`, `fast.tramo()`
- Refresh policies: `tramo.refresh()` and `tramoseats.refresh()`

### Example

```
# {r tramoseats, out.height="70%"}
# spec = spec_tramoseats_default("rsafull") |>
#   set_easter(type = "IncludeEasterMonday") |>
#   set_tradingdays(test = "Separate_T") |>
#   set_seats(algorithm = "KalmanSmoother")
# m = rjd3tramoseats::tramoseats(y, spec)
# # More informations:
# names(m)
# m$result
```

### rjdemetra3

Functions to manipulate JDemetra+ workspaces:

- Still in construction: you can load an existing workspace but not create a new one (use `jws.load()` for example)
- Will contain all the functionalities of `rjdworkspace`

### rjd3highfreq

Seasonal adjustment of high frequency data:

- fractional and multi airline decomposition
- Extension of X-11 decomposition with non integer periodicity

`rjd3stl` : STL, MSTL, ISTL, loess

### rjd3stl

## Add-in packages

table ggdemetra rjworkspace

attention version 2 and version: adaptation under construction version 2, retrieve R tools info here

### ggdemetra

### ggdemetra3

Like `ggdemetra` but compatible with `rjdemetra3`: ggplot2 to add seasonal adjustment statistics to your plot. Also compatible with high-frequency methods (WIP):

```
library(ggdemetra3)
spec <- spec_x13_default("rsa3") |> set_tradingdays(option = "WorkingDays")
p_ipi_fr <- ggplot(data = ipi_c_eu_df, mapping = aes(x = date, y = FR)) +
    geom_line() +
    labs(title = "SA - IPI-FR",
         x = NULL, y = NULL)
p_sa <- p_ipi_fr +
    geom_sa(component = "y_f(12)", linetype = 2,
            spec = spec) +
    geom_sa(component = "sa", color = "red") +
    geom_sa(component = "sa_f", color = "red", linetype = 2)
p_sa
p_sa +
    geom_outlier(geom = "label_repel",
                 coefficients = TRUE,
                 ylim = c(NA, 65), force = 10,
                 arrow = arrow(length = unit(0.03, "npc"),
                               type = "closed", ends = "last"),
                 digits = 2)
```

## rjdfilters

### rjdfilters (1)

- easily create/combine/apply moving averages `moving_average()` (much more general than `stats::filter()`) and study their properties: plot coefficients (`plot_coef()`), gain (`plot_gain()`), phase-shift (`plot_phase()`) and different statics (`diagnostic_matrix()`)
- trend-cycle extraction with different methods to treat endpoints:
- `lp_filter()` local polynomial filters of Proietti and Luati (2008) (including Musgrave): Henderson, Uniform, biweight, Trapezoidal, Triweight, Tricube, "Gaussian", Triangular, Parabolic (= Epanechnikov)
- `rkhs_filter()` Reproducing Kernel Hilbert Space (RKHS) of Dagum and Bianconcini (2008) with same kernels
- `fst_filter()` FST approach of Grun-Rehomme, Guggemos, and Ladiray (2018)
- `dfa_filter()` derivation of AST approach of Wildi and McElroy (2019)
- change the filter used in X-11 for TC extraction

### Create moving average `moving_average()`

```
library(rjdfilters)
m1 = moving_average(rep(1,3), lags = 1); m1 # Forward MA
m2 = moving_average(rep(1,3), lags = -1); m2 # centered MA
m1 + m2
m1 - m2
m1 * m2
```

Can be used to create all the MA of X-11:

```
e1 <- moving_average(rep(1,12), lags = -6)
e1 <- e1/sum(e1)
e2 <- moving_average(rep(1/12, 12), lags = -5)
# used to have the 1rst estimate of the trend
tc_1 <- M2X12 <- (e1 + e2)/2
coef(M2X12) |> round(3)
```

```r
si_1 <- 1 - tc_1
M3 <- moving_average(rep(1/3, 3), lags = -1)
M3X3 <- M3 * M3
# M3X3 moving average applied to each month
coef(M3X3) |> round(3)
M3X3_seasonal <- to_seasonal(M3X3, 12)
coef(M3X3_seasonal) |> round(3)
s_1 = M3X3_seasonal * si_1
s_1_norm = (1 - M2X12) * s_1
sa_1 <- 1 - s_1_norm
henderson_mm = moving_average(lp_filter(horizon = 6)$
                                  filters.coef[, "q=6"],
                              lags = -6)
tc_2 <- henderson_mm * sa_1
si_2 <- 1 - tc_2
M5 <- moving_average(rep(1/5, 5), lags = -2)
M5X5_seasonal <- to_seasonal(M5 * M5, 12)
s_2 = M5X5_seasonal * si_2
s_2_norm = (1 - M2X12) * s_2
sa_2 <- 1 - s_2_norm
tc_f <- henderson_mm * sa_2
```

```r
par(mai = c(0.3, 0.3, 0.2, 0))
layout(matrix(c(1,1,2,3), 2, 2, byrow = TRUE))

plot_coef(tc_f);plot_coef(sa_2, col = "orange", add = TRUE)
legend("topleft",
       legend = c("Final TC filter", "Final SA filter"),
       col= c("black", "orange"), lty = 1)
plot_gain(tc_f);plot_gain(sa_2, col = "orange", add = TRUE)
plot_phase(tc_f);plot_phase(sa_2, col = "orange", add = TRUE)
```

### Apply a moving average

```r
y <- retailsa$AllOtherGenMerchandiseStores
trend <- y * tc_1
sa <- y * sa_1
plot(window(ts.union(y, trend, sa), start = 2000),
     plot.type = "single",
```

```
        col = c("black","orange", "lightblue"))
```

## rjd3sts

### rjd3sts

Interface to structural time series and state space models

Several examples available here https://github.com/palatej/test_rjd3sts

## rjd3bench

### rjd3bench

Benchmarking and temporal disaggregation

Several examples here: https://github.com/palatej/test_rjd3bench

## Interaction with workspaces and GUI

# Cruncher and quality report

## Overview

The cruncher is an additional "executable" module, not an R package. It can be launched via R, SAS.

Objective of the cruncher:

- update a JDemetra+ workspace (with a given revision policy)
- export the results (series and diagnostics),

without having to open the graphical interface and operate manually. Suitable for a production process.

## Installation procedure

- Download the cruncher

Available here https://github.com/jdemetra/jwsacruncher/releases

Click on the zip code line of the latest release

- Unzip locally (or on server)

## Help pages

Documentation is available here or click on the wiki icon on the Github page https://github.com/jdemetra/jwsacruncher/wiki

## Running the cruncher in R

Two R packages are currently available

- rjwsacruncher on CRAN: workspace update and output production
- Cruncher (https://github.com/InseeFr/JDCruncheR): same functions as rjwsacruncher but adds a quality report

### Installation

- rjwsacruncher

```
install.packages(rjwsacruncher)
```

- JDCruncheR package: download the **.zip** or **.tar.gz** file from https://github.com/InseeFr/JDCruncheR/releases.

Additional packages needed

```
install.packages(c("XLConnect", "XML"))
```

### Loading

```
library(JDCruncheR)
```

or

```
library(rjwsacruncher)
```

### Connecting the cruncher module

To connect the cruncher to the R package, the path to the bin directory containing the **cruncher.bat** file must be specified. This directory is available once the zip file has been unzipped.

```
options(cruncher_bin_directory =
    "C:/Software/jwsacruncher-2.2.3/jdemetra-cli-2.2.3/bin")
```