JDemetra+ documentation

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

12/22/22

Table of contents

Preface

Welcome to the JDemetra+ on-line 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 you can visit Eurostat CROS Portal

To keep up with all JDemetra+ related news head to the **JDemetra+ Universe Blog**

This website is under construction, in the meantime you can fill a large number of the gaps by referring to the previous version of the on-line documentation.

JDemetra+ Software

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. 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.

Version 2.x and version 3

Version 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.

Structure of this book

This book is divided in 3 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 JDemetra+:

- Seasonal Adjustment
- Seasonal Adjustment of high-frequency data
- Reg-Arima modelling
- Outlier detection
- Generating Calendar regressors and input variables
- Benchmarking and temporal disaggregation
- Trend-Cycle estimation
- Nowcasting

Output series, diagnostics, as well as parameters (automatically estimated or user-defined) are detailed in the relevant chapters.

Tools

This part describes the tools allowing to access JDemetra+ algorithms:

- Graphical User Interface GUI
- …enhanced with additional plug-ins
- ..and a Cruncher for mass production
- R packages

Methods

This part describes in greater detail the core algorithms and their underlying statistical methods:

- Reg-Arima modelling
- X-11: moving average based decomposition
- SEATS: Arima model based decomposition
- STL: Loess based decomposition
- Benchmarking and temporal disaggregation
- Spectral analysis tools
- Trend Estimation
- Tests for seasonality and residuals
- Structural time series and state space framework

Part I

Algorithms

This part describes the algorithms featured in JDemetra+:

- Seasonal Adjustment
- Seasonal Adjustment of high-frequency data
- Reg-Arima modelling
- Outlier detection
- Generating Calendar regressors and input variables
- Benchmarking and temporal disaggregation
- Trend-Cycle estimation
- Nowcasting

Seasonal Adjustment

Overview

The goal of seasonal adjustment is to remove seasonal fluctuations a time series. Seasonal fluctuations are quasi-periodic infra-annual movements. They can mask evolution of greater interest for the user such as short term evolutions or long time trends.

This chapter focuses on the **practical step by step** use of JDemetra+ algorithms, restricted to monthly and quarterly series. For intra-monthly data see the following chapter. The use of graphical user interface and R packages are described simultaneously whenever relevant.

In-depth methodological explanations of the algorithms are covered in separated chapters, in the Methods part.

This chapter is under construction, missing parts will be updated in the coming months.

More information on the steps and best practices of a seasonal adjustment process can be found in the Eurostat guidelines on seasonal adjustment

For an overview on the algorithms and methodological issues, please refer to the Handbook on Seasonal Adjustment

Seasonal Adjustment Algorithms

Table 1: X13-Arima and Tramo-Seats are two-step algorithm with a pretreatment phase (Reg-Arima or Tramo) and a decomposition phase (X11 and Seats). STL is a local regression (Loess) based decomposition, without pre-treatment. In a Structural Time Series approach pre-treatment and decomposition are done simultaneously in a State Space Framework (SSF).

Algorithm	Access in GUI	Access in R (v2)	Access in R v3
X-13 Arima	yes	RJDemetra	rjd3x13
Reg-Arima only	yes	RJDemetra	rjd3x13
X11 decomposition only	yes	RJDemetra	rjd3x13
Tramo-Seats	yes	RJDemetra	rjd3tramoseats
Tramo only	yes	RJDemetra	rjd3tramoseats

Algorithm	Access in GUI	Access in R (v2)	Access in R v3
Seats only	-	-	-
STL	no	no	rjd3stl
STS	no	no	rjd3sts

Decomposition in unobserved components

To seasonally adjust a series, seasonal factors S_t will be estimated and removed from the original raw series: $Y_{sa} = Y_t/S_t$ or $Y_{sa} = Y_t - S_t$. To do so the series is first decomposed into unobservable components. Two decomposition models:

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

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 algorithms.

Pre-treatment principles

The goal of this step is to remove deterministic effects (calendar and outliers) in order to improve the decomposition.

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

- O_{it} are the i final outliers (AO, LS, TC)
- 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)
- $Y_{lin,t} \sim ARIMA(p,d,q)(P,D,Q)$

Detecting seasonal patterns

A large number of seasonality tests are available in JDemetra+. They can be accessed in the graphical user interface or via R.

In rjd3modelling package:

- Canova-Hansen (`seasonality.canovahansen()`)

In rjd3sa package:

- 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()`)

Performing pre-treatment

The following sections cover how to perform pre-treatment with Reg-ARIMA (or TRAMO) algorithms. Tramo and the Reg-Arima part of X13-Arima rely on very similar principles: Reg-Arima modelling. Thus Tramo will be mentioned only to highlight differences with the Reg-Arima part of X13-Arima. Reg-Arima modelling part can be of a seasonal adjustment process or run on its own, we focus first on launching pre-treatment as part of a SA processing.

Default Specifications

Default specifications are set for the whole SA procedure, pre-treatment end decomposition. They are slightly different for X13-ARIMA and Tramo-Seats and can be modified with user defined parameters.

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, for example in RSA3 there is no calendar effect correction
- automatic: test is performed
- outliers detection : AO/LS/TC type of outliers automatically detected under a critical T-Stat value (default value=4)
- calendar
- 2 regressors: weekdays vs week-ends + LY
- 7 regressors: each week day vs Sundays + LY
- always tested
- easter tested (length = 6 days in tramo, 8 days in X13-Arima)

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

Spec identifier	Log/level detection	Outliers detection	Calendar effects	ARIMA
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 in GUI and R, see below.

Spans

Estimation span

Specifies the span (data interval) of the time series to be used in the seasonal adjustment process. The user can restrict the span

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] included (yyyy-mm-dd to yyyy-mm-dd)
First	number of obs from the beginning 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	check 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 and expand the nodes.

Setting span in R

x13 in version 2

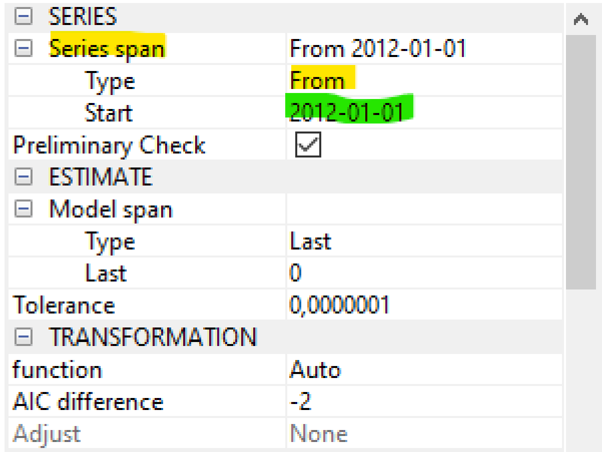


Figure 1: Setting series span

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

# estimation interval: option with dynamic numbers of observations

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

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

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

# Retrieve settings
```

For comprehensive details about x13_spec function see RJDemetra R help pages.

Tramo-Seats in version 2

```
#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)
```

For comprehensive details about tramoseats_spec function see RJDemetra R help pages

Setting the model span

The user can also specify the span (data interval) of the time series to be used for the estimation of the Reg-ARIMA model coefficients. It allows to impede a chosen part of the data from influencing the regression estimates. Setting works the same way as setting series (estimation) span described above.

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

Tolerance	Convergence tolerance for the non-linear 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

Setting in R

Tramo example in version 2

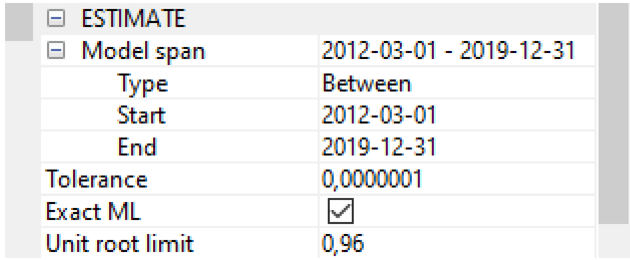


Figure 2: Model span setting

```
#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)
```

Decomposition Scheme

Parameters

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 default AIC difference value is -2.

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

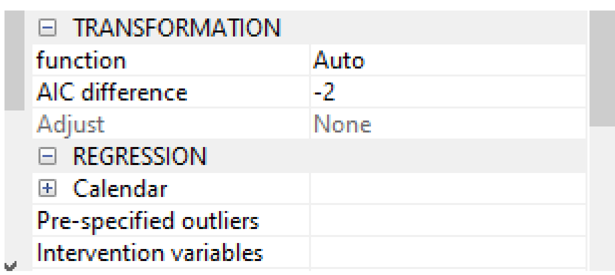


Figure 3: Model span setting

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

Default specifications for calendar correction

Options related to calendar regressors choice are embedded into default specifications described above.

National calendars not taken into account.

- working days
- trading days
- Leap year regressor
- Test: Remove/ Add / None
- Easter
- Test add / remove / non
- duration (enabled when testing removed)
- Julian
- pre-test

Retrieving parameters

In GUI

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

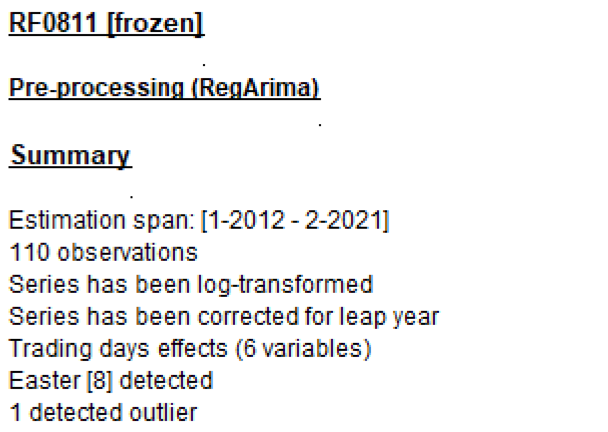
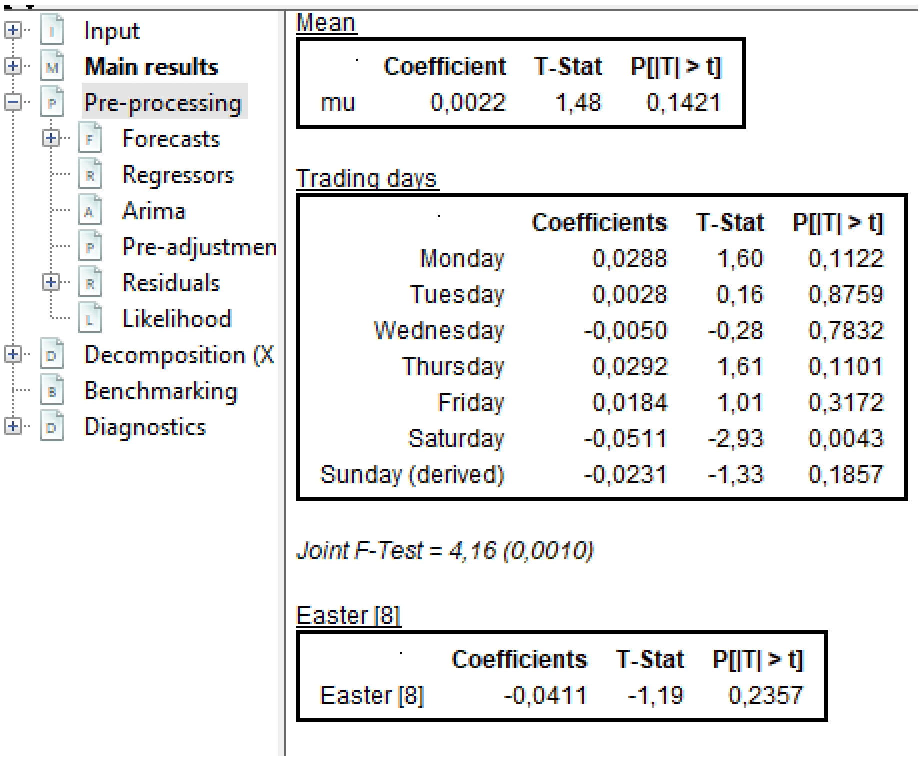


Figure 4: Text

Details of regression results are displayed in the pre processing panel



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

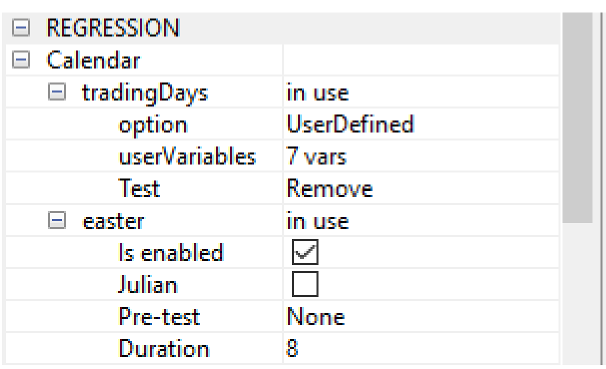
Joint F-Test = 4,16 (0,0010)

Easter [8]

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

Setting options

In GUI Use the specification window



REGRESSION	
Calendar	
tradingDays	in use
option	UserDefined
userVariables	7 vars
Test	Remove
easter	in use
Is enabled	☑
Julian	☐
Pre-test	None
Duration	8

Customizing calendars

Under construction.

Outliers and intervention variables

Under construction.

Arima Model

Default specifications

Key specifications on Arima modelling are embedded in default specifications: airline (default model) or full automatic research.

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

Modifying 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 cancellation 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 cancellation 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

Under construction

Setting in R

X13-Arima template in version 2

```
spec_2 <- 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 polynomials. 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 Reg-Arima 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

Under construction

Setting in R

X13-Arima template in version 2

```
spec_2 <- 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_)
```

Reg-Arima model Results and Diagnostics

Under construction.

X-11 Decomposition

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 decompose the **linearized series** using iteratively different moving averages. The effects of pre-treatment will be reallocated at the end.

The sections below (will) describe

- specifications needed to run X-11
- generated output

- series
- diagnostics
- final parameters
- user-defined parameters

Default specifications

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

Quick Launch

From GUI

With a workspace open, an SAProcessing created and open data provider:

- choose a default specification

- drop your data and press green arrow

In R

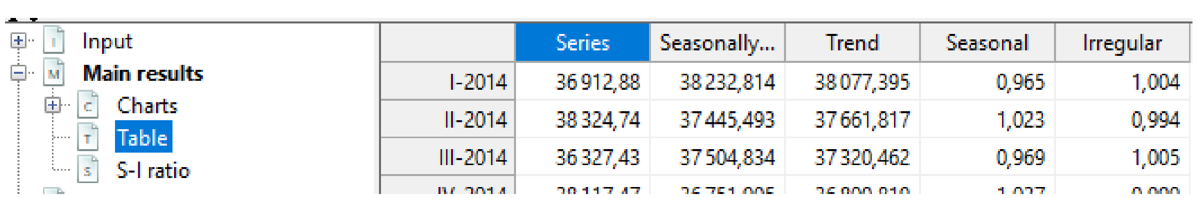
In version 2

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

The model_sa R object (list of lists) contains all parameters and results. It will be progressively detailed below.

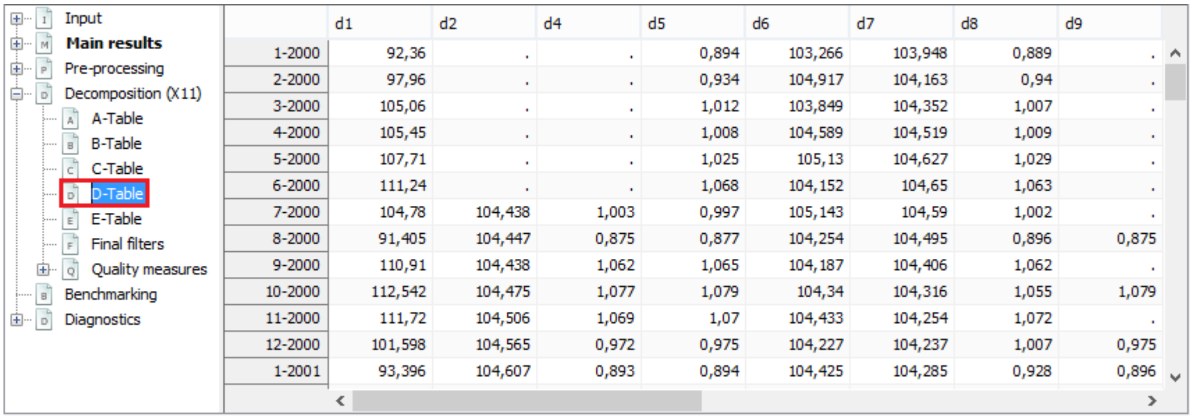
Output 1: series

Display in GUI

Main results 

(forecasts glued, values in *italic*)

X-11 Tables



	d1	d2	d4	d5	d6	d7	d8	d9
1-2000	92,36	.	.	0,894	103,266	103,948	0,889	.
2-2000	97,96	.	.	0,934	104,917	104,163	0,94	.
3-2000	105,06	.	.	1,012	103,849	104,352	1,007	.
4-2000	105,45	.	.	1,008	104,589	104,519	1,009	.
5-2000	107,71	.	.	1,025	105,13	104,627	1,029	.
6-2000	111,24	.	.	1,068	104,152	104,65	1,063	.
7-2000	104,78	104,438	1,003	0,997	105,143	104,59	1,002	.
8-2000	91,405	104,447	0,875	0,877	104,254	104,495	0,896	0,875
9-2000	110,91	104,438	1,062	1,065	104,187	104,406	1,062	.
10-2000	112,542	104,475	1,077	1,079	104,34	104,316	1,055	1,079
11-2000	111,72	104,506	1,069	1,07	104,433	104,254	1,072	.
12-2000	101,598	104,565	0,972	0,975	104,227	104,237	1,007	0,975
1-2001	93,396	104,607	0,893	0,894	104,425	104,285	0,928	0,896

Figure 5: Text

Output series can be exported out of GUI by two means:

- generating output files
- running the cruncher to generate 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
```

Output 2: diagnostics

X11 produces the following type diagnostics or quality measures

SI-ratios

0.0.0.0.1 * Display in GUI

NODE Main Results > SI-Ratios SA_MainResults_SI_ratios.png

In GUI all values cannot be retrieved

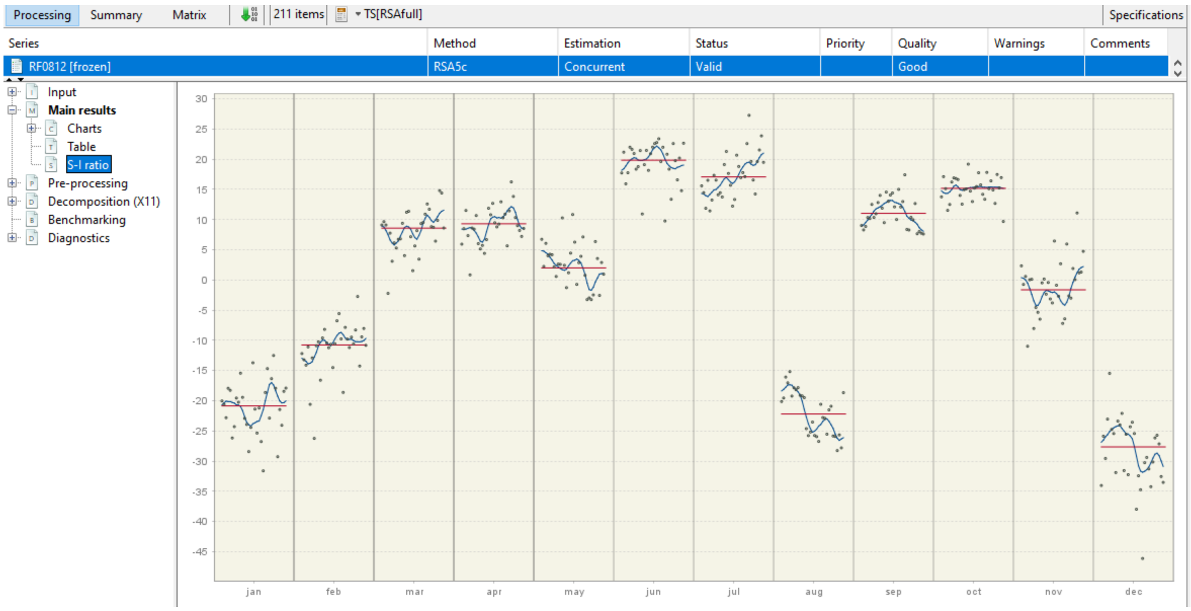


Figure 6: Text

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))
```

M-statistics

At the end of the decomposition, X-11 algorithm provides quality measure of the decomposition called "M statistics": 11 statistics (M1 to M11) and 2 summary indicators (Q et Q-M2). By design $0 < M_x < 3$ and acceptance region is $M_x \leq 1$

- **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

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.

0.0.0.0.1 * Display in GUI

To display results in GUI, expand NODE

Decomposition(X-11) > Quality Measures > Summary

Results displayed in red indicate that the test failed.

0.0.0.0.2 * Retrieve in R

In version 2

```
# this code snippet is not self-sufficient
# shpould it be
```

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 7: Text

```
model_sa$decomposition$mstats
```

Detailed Quality measures

In GUI all the diagnostics below can be displayed expanding the NODE

Decomposition(X-11) > Quality Measures > Details

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

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 Reg-Arima 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

For the multiplicative decomposition the following formula is used:

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

.

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

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 $(O_d^*)^2 \approx O_d^2$ 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

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

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

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

0.0.0.0.6 * Relative contribution to the stationary 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 stationary 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 8: 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

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).

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 9: Text

0.0.0.0.9 * Moving seasonality ratios (MSR)

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

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

and

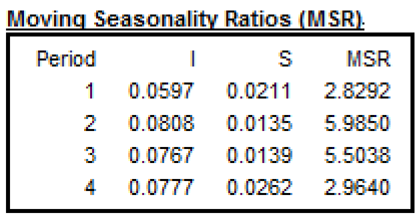
$$\overline{I}_i = \frac{1}{N_i - 1}\sum_{t=2}^{N_i}|I_{i,t} - I_{i,t-1}|$$

,

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 10: 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).

Display in GUI

Node Decomposition(X11) > Final Filters

Retrieve in R

In version 2

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

User-defined parameters

The following sections describe how to change default values or automatic choices.

0.0.0.0.1 * General settings

- **Mode**
- **Seasonal component**
- **Forecasts horizon**

Length of the forecasts generated by the Reg-Arima 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 Reg-Arima 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
  - 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 Reg-Arima 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.

- **Seasonal filter**
- default value: *MSR* (Moving seasonality ratio), automatic choice of final seasonal filter, initial filters are 3×3
- choices : 3×1, 3×3, 3×5, 3×9, 3×15 or Stable
- "Stable" : constant factor for each calendar period (simple moving average of a all $S+I$ values for each period)

User choices will be applied 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

All the parameters above can be set with in the specification box.

Parameter setting in R packages

In version 2

```
#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,
      x11.sigmaVector = NA,
    x11.excludeFcasts = NA
    # filters
    x11.trendAuto = NA,
    x11.trendma = 23,
    x11.seasonalma = "S3X9)

#New SA estimation : apply modified_spec

modified_sa_model<-x13(raw_series,modified_spec)
```

SEATS Decomposition

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.

The sections below will describe

- specifications needed to run SEATS
- generated output
- series
- diagnostics
- final parameters
- user-defined parameters

Not currently available. Under construction

STL

Loess based decomposition algorithm used on linearized data data, no pre-adjustment.

Not currently available. Under construction.

Basic Structural Models

Not currently available. Under construction.

Seasonal adjustment of high-frequency data

Overview

This chapter provides basic hints on seasonal adjustment of high-frequency data with JDemetra+ tailored algorithms.

Currently available topics:

- description of high frequency data specificities
- R functions for pre-treatment, extended X11 and extended Seats

Topics under construction

- graphical user interface 3.0 functionalities for high-frequency data
- STL functions
- State space framework

Data specificities

Intra-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.

Table 6: Periodicities (number of observations per cycle)

Data	Day	Week	Month	Quarter	Year
quarterly					4
monthly				3	12
weekly			4.3481	13.0443	52.1775
daily		7	30.4368	91.3106	365.2425
hourly	24	168	730.485	2191.4550	8765.82

Tailored algorithms in JDemetra+

Col1	Algorithm	GUI v 3.0	R package
Pre-treatment	Fractional Airline Model	yes	rjd3highfreq
Decomposition	Extended SEATS Fractional Airline Model	yes	rjd3highfreq
	Extended X-11	yes	rjd3highfreq
	Extended STL	no	rjd3stl
One-Step	SSF Framework	no	rjd3sts

Unobserved Components

Raw series decomposition

$$
\begin{array}{ccccccccc}
\text{series raw} & = & \text{trend} & \circ & \text{seasonal} & \circ & \text{calendar} & \circ & \text{irregular} \\
\downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\
y_t & = & t_t & \circ & s_t & \circ & c_t & \circ & i_t \\
 & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\
 & & \text{Variations} & & \text{Variations} & & \text{effects} & & \text{residual} \\
 & & \text{long-term} & & \text{infra-yearly} & & & & \\
 & & & & \text{periodic} & & \text{calendar} & & \\
 & & \searrow & & & & & & \swarrow \\
 & & & & \multicolumn{3}{c}{\text{seasonally adjusted (sa) data}} & &
\end{array}
$$

- decomposition Additive ($\circ = +$), multiplicative ($\circ = \times$)

Multiple seasonal factors

For high-frequency data multiple seasonal factors might be taken into account.

$$S_t = S_{t,7} \circ S_{t,30.44} \circ S_{t,365.25}$$

The decomposition is done iteratively periodicity by periodicity starting with the smallest one (highest frequency) as:

- highest frequencies usually display the biggest and most stable variations
- cycles of highest frequencies can mix up with lower ones

Identifying seasonal patterns

JDemetra+ provides the Canova-Hansen test in rjd3modelling package.

Pre-adjustment

In X13-Arima and Tramo-Seats, a pre-adjustment step is performed to remove deterministic effects, outliers and calendar with a Reg-Arima model. In the extended version for HF data, it is also the case with a **fractional airline model**.

A general Reg-ARIMA model is written as follows:

$$\left(Y_t - \sum \alpha_i X_{it}\right) \sim ARIMA(p,d,q)(P,D,Q)$$

These models contain seasonal backshift operators $B^s(y_t) = y_{t-s}$. Here s can be non integer. JDemetra+ will rely on a modified version of a frequently used Arima model: the "Airline" model:

$$(1-B)(1-B^s)y_t = (1-\theta_1 B)(1-\theta_2 B^s)\epsilon_t \qquad \epsilon_t \sim \text{NID}(0,\sigma^2_\epsilon)$$

For high frequency-data potentially non integer s will be written: $s = s' + \alpha$, with α real number i $]0,1[$ (for example $52.18 = 52 + 0.18$ is the yearly periodicity for weekly data)

Taylor development around 1 of $f(x) = x^\alpha$

$$\begin{array}{rcl} x^\alpha & = & 1 + \alpha(x-1) + \frac{\alpha(\alpha+1)}{2!}(x-1)^2 + \frac{\alpha(\alpha+1)(\alpha+2)}{3!}(x-1)^3 + \cdots \\ B^\alpha & \cong & (1-\alpha) + \alpha B \end{array}$$

Approximation of $B^{s+\alpha}$ in a fractional Airline model

$$B^{s+\alpha} \quad \cong \quad (1-\alpha)B^s + \alpha B^{s+1}$$

Example for a daily series displaying two periodicities $p_1 = 7$ and $p_2 = 365.25$:

$$(1-B)(1-B^7)(1-B^{365.25})(Y_t - \sum \alpha_i X_{it}) = (1-\theta_1 B)(1-\theta_2 B^7)(1-\theta_3 B^{365.25})\epsilon_t$$
$$\epsilon_t \sim \text{NID}(0,\sigma^2_\epsilon)$$

with

$$1 - B^{365.25} = (1 - 0.75B^{365} - 0.25B^{366})$$

Calendar correction

Calendar regressors can be defined with rjd3modelling package and added to pre-treatment function as a matrix.

```
#setting calendar variables
## define a calendar
frenchCalendar <- rjd3modelling::calendar.new()
rjd3modelling::calendar.holiday(frenchCalendar, "NEWYEAR")
#rjd3modelling::calendar.holiday(frenchCalendar, "GOODFRIDAY")
rjd3modelling::calendar.holiday(frenchCalendar, "EASTERMONDAY") # Lundi de Pâques
rjd3modelling::calendar.holiday(frenchCalendar, "MAYDAY") # 1er mai
rjd3modelling::calendar.fixedday(frenchCalendar, 5, 8,start="1982-05-08") # since 1982, 2n
rjd3modelling::calendar.easter(frenchCalendar,offset = 39) ## ascension
rjd3modelling::calendar.holiday(frenchCalendar, "WHITMONDAY") # Lundi de Pentecôte
rjd3modelling::calendar.fixedday(frenchCalendar, 7, 14)  # national holiday
rjd3modelling::calendar.holiday(frenchCalendar, "ASSUMPTION") # Assomption
rjd3modelling::calendar.holiday(frenchCalendar, "ALLSAINTSDAY") # Toussaint
rjd3modelling::calendar.holiday(frenchCalendar, "ARMISTICE") # first world war
rjd3modelling::calendar.holiday(frenchCalendar, "CHRISTMAS") #  25th December

## extract regressors: here daily series
q<-rjd3modelling::holidays(frenchCalendar, "1968-01-01",
                           length = length(y_raw_daily), type="All",
                           nonworking = as.integer(7))
```

Outliers and intervention variables

Outliers detection is available in the pre-treatment function. Detected outliers are AO, LS and WO. Critical value can be computed by the algorithm or user-defined.

Linearization

Example using rjd3highfreq::fractionalAirlineEstimation function:

```
pre_adjustment<- rjd3highfreq::fractionalAirlineEstimation(y_raw,
                x = q, # q= calendar regressors
                periods = c(7,365.25)
                ndiff = 2, ar = FALSE, mean = FALSE,
                outliers = c("ao","ls", "wo"),
```

```
                    criticalValue = 0, # computed in the algorithm
                    precision = 1e-9, approximateHessian = TRUE)
```

"pre_adjustment" R object is a list of lists in which the user can retrieve input series, parameters and output series. For more details see chapter on R packages and rjd3highfreq help pages R, where all parameters are listed.

Decomposition

Extended X-11

X-11 is the decomposition module of X-13-Arima, the linearized series from the pre-adjustment step is split into seasonal (S), trend (T) and irregular (I) components.In case of multiple periodicities the decomposition is done periodicity by periodicity starting with the smallest one. Global structure of the iterations is the same as in "classical" X-11 but modifications were introduced for tackling non integer periodicities. They rely on the Taylor approximation for the seasonal backshift operator:

$$B^{s+\alpha} \quad \cong \quad (1-\alpha)B^s + \alpha B^{s+1}$$

Modification of the first trend filter for removing seasonality

The first trend estimation is thanks to a generalization of the centred and symmetrical moving averages with an order equal to the periodicity p.

- filter length longueur l: smallest odd integer greater than p
- ex : p=7, l=7, p=12 l=13, p=365,25, l=367, p=52.18 l=53
- central coefficients $1/p$ (1/12,1/7, 1/365.25)
- extreme coefficients $\mathbb{I}\{E(p) \text{ even}\} + (p - E(p))/2p$
- example for p=12: (1/12 and 1/24) (we fall back on $M_{2\times 12}$ of the monthly case
- example for p=365.25: (1/365.25 and 0.25/(2*365.25)

Modification of seasonality extraction filters

Computation is done on a given period

Example $M_{3\times3}$

$$M_{3\times3}X = \frac{1}{9}(X_{t-2p}) + \frac{2}{9}(X_{t-p}) + \frac{3}{9}(X_t) + \frac{2}{9}(X_{t+p}) + \frac{1}{9}(X_{t+2p})$$

if p integer: no changes needed

if p non integer: Taylor approximation of the backshift operator

Modification of final trend estimation filter

As seasonality has been removed in the first step, there is no non integer periodicity issue in the final trend estimation, but extended X-11 offers additional features vs genuine X-11, in which final trend is estimated with Henderson filters and Musgrave asymmetrical surrogates. In extended in X-11, a generalization of this method with local polynomial approximation is available.

Example of decomposition

Here the raw series is daily and displays two periodicities $p = 7$ and $p = 365.25$

```
# extraction of day-of-the-week pattern (dow)
x11.dow <- rjd3highfreq::x11(y_linearized,
        period = 7,                 # DOW pattern
        mul = TRUE,
        trend.horizon = 9,  # 1/2 Filter length : not too long vs p
        trend.degree = 3,                         # Polynomial degree
        trend.kernel = "Henderson",               # Kernel function
        trend.asymmetric = "CutAndNormalize",     # Truncation method
        seas.s0 = "S3X9", seas.s1 = "S3X9",       # Seasonal filters
        extreme.lsig = 1.5, extreme.usig = 2.5)   # Sigma-limits


# extraction of day-of-the-week pattern (doy)

x11.doy <- rjd3highfreq::x11(x11.dow$decomposition$sa,  # previous sa
                    period = 365.2425,         # DOY pattern
                    mul = TRUE,
```

```
                          trend.horizon = 371, # 1/2 final filter length
                          trend.degree = 3,
                          trend.kernel = "Henderson",
                          trend.asymmetric = "CutAndNormalize",
                          seas.s0 = "S3X15", seas.s1 = "S3X5",
                          extreme.lsig = 1.5, extreme.usig = 2.5)
```

Arima Model Based (AMB) Decomposition (Extended Seats)

Example

```
# extracting doy pattern

amb.doy <- rjd3highfreq::fractionalAirlineDecomposition(
  amb.dow$decomposition$sa,  # DOW-adjusted linearised data
  period = 365.2425,         # DOY pattern
  sn = FALSE,                # Signal (SA)-noise decomposition
  stde = FALSE,              # Calculate standard deviations
  nbcasts = 0, nfcasts = 0)  # Numbers of back- and forecasts
```

Summary of the process

Seasonal adjustment processing in rjd3highfreq cannot be encompassed by one one function like for lower frequency, e.g rjd3x13::x13(y_raw)

The user has to run the steps one by one, here is an example with $p = 7$ and $p = 365.25$

- computation of the linearized series $Y_{lin} = FracAirline(Y)$
- computation of the calendar corrected series Y_{cal}
- computation of S_7 by decomposition of the linearized series
- computation of $S_{365.25}$ by decomposition of the seasonally adjusted series with $p = 7$
- finally adjusted series $sa_{final} = Y_{cal}/S_7/S_{365.25}$ (if multiplicative model)

STL decomposition

Not currently available. Under construction.

State Space framework

Not currently available. Under construction.

Quality assessment

Residual seasonality

JDemetra+ provides the Canova-Hansen test in rjd3modelling package which allows to check for any remaining seasonal periodicity in the final SA data.

Residual Calendar effects

Not currently available. Under construction.

Outlier detection

Under construction.

Overview

JDemetra+ provides outlier detection routines which can be used stand alone or as part of a seasonal adjustment process. They can be accessed via GUI or R packages.

With Reg Arima models

Specific TERROR tool

With structural models (BSM)

Calendar and user-defined corrections

This chapter will describe

- underlying principles and concepts of calendar correction
- generating process of calendars, calendar regressors, outliers and other input variables.

How to use these variables inside a seasonal adjustment process is detailed in chapters on SA or SA of HF data.

Overview of Calendar effects in JDemetra

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
Feb-2013	20	21.7411	-1.5625	-0.1786
Mar-2013	21	21.7411	0.4018	-1.1429
Apr-2013	22	21.7411	-0.3125	0.5714
May-2013	23	21.7411	0.4018	0.8571
Jun-2013	20	21.7411	-0.3125	-1.4286
Jul-2013	23	21.7411	0.4018	0.8571

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

Impact of the mean effects on the decomposition

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

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

For such a model the following relations hold:

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

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

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

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

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:

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

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

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

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

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

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

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

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

[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).

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

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

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

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

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 disagreggation

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/385
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.

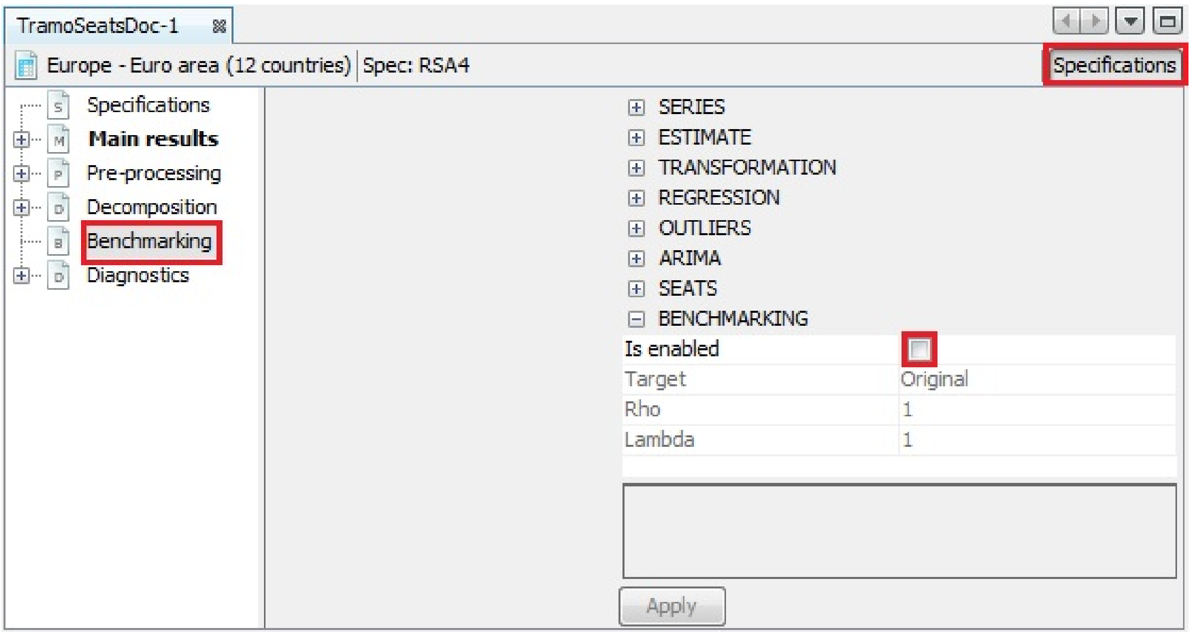


Figure 11: Text

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.

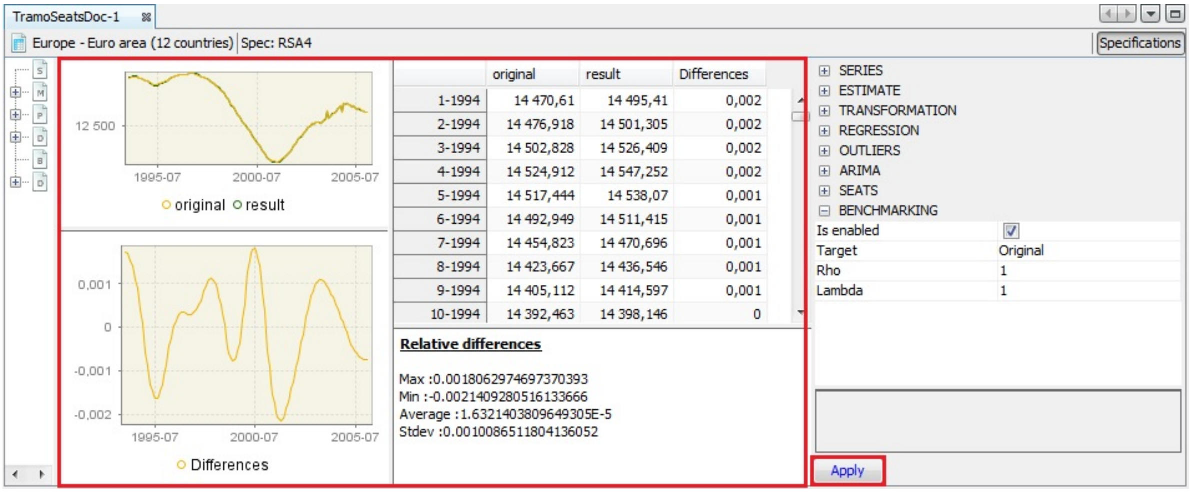


Figure 12: Text

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

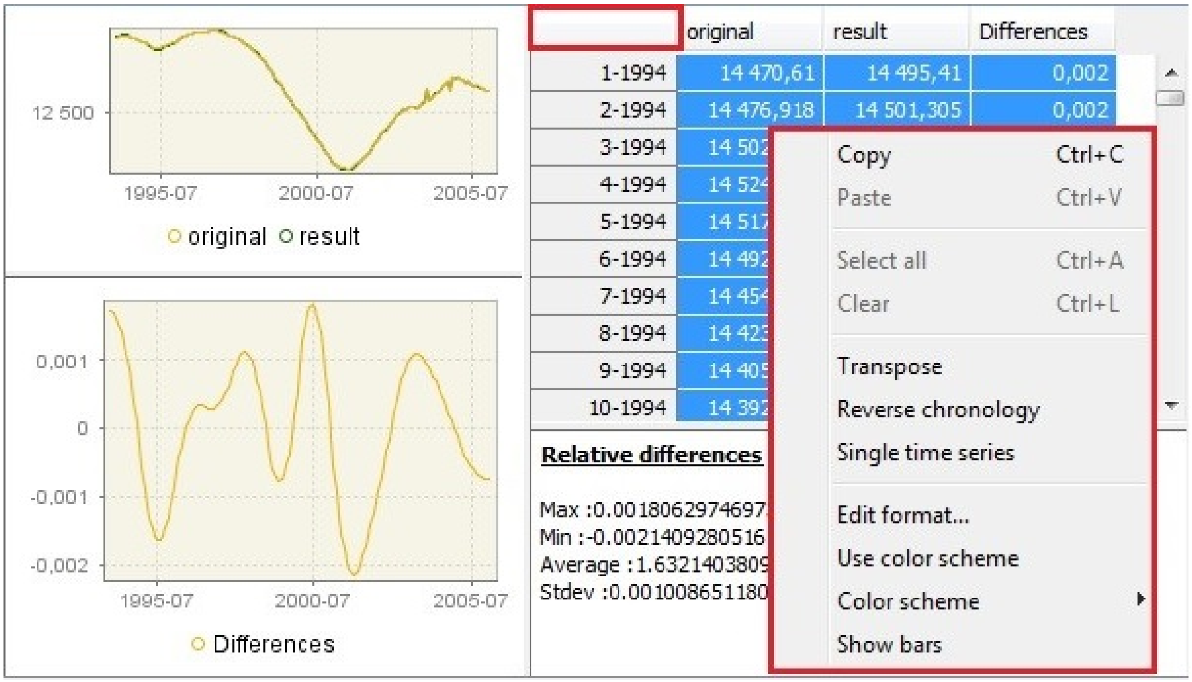


Figure 13: Text

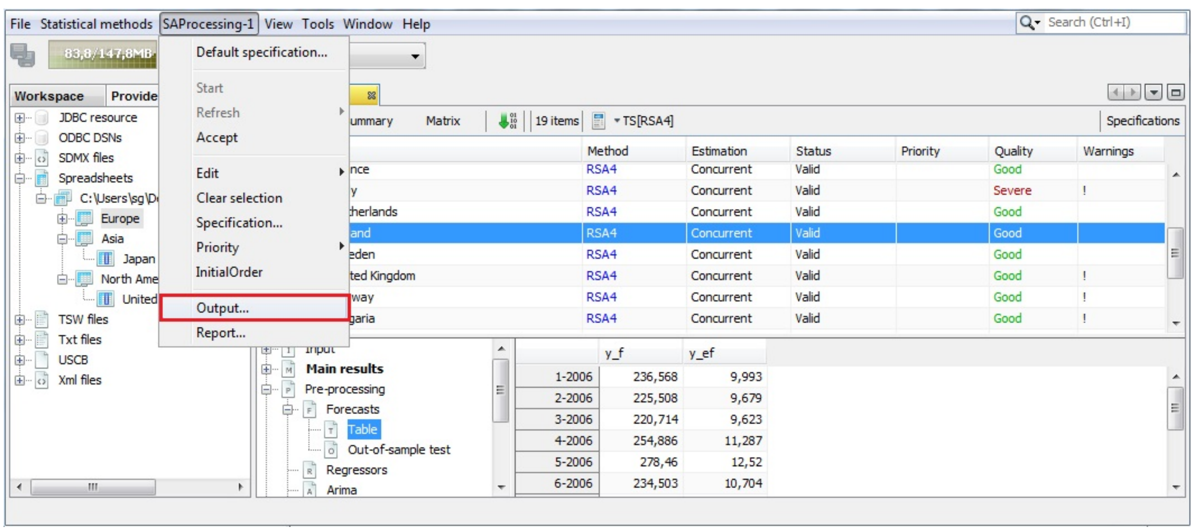


Figure 14: Text

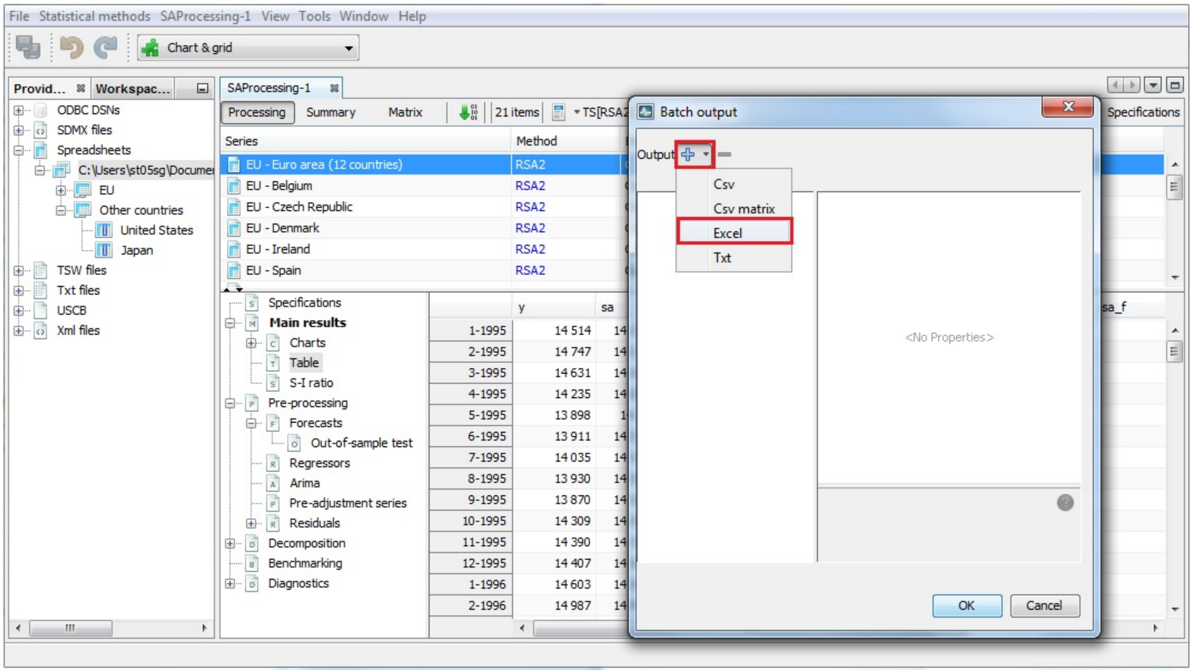


Figure 15: Text

Benchmarking in R

See package **rjd3bench** and its documentation pages in R.

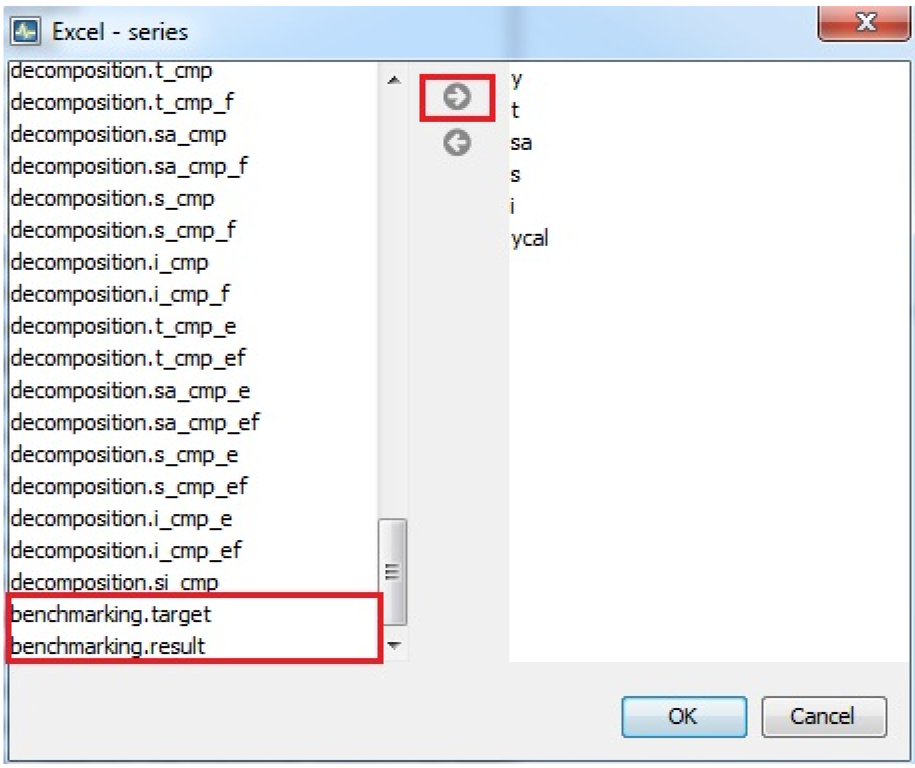


Figure 16: Text

Trend-cycle estimation

Under construction

Overview

Estimation Methods

Tools

rjd3highfeq package

rjdfilters package

Nowcasting

Under construction.

Part II

Tools

The different tools described in this part:

- Graphical User Interface GUI
- …enhanced with additional plug-ins
- ..and a Cruncher for mass production
- R packages

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).

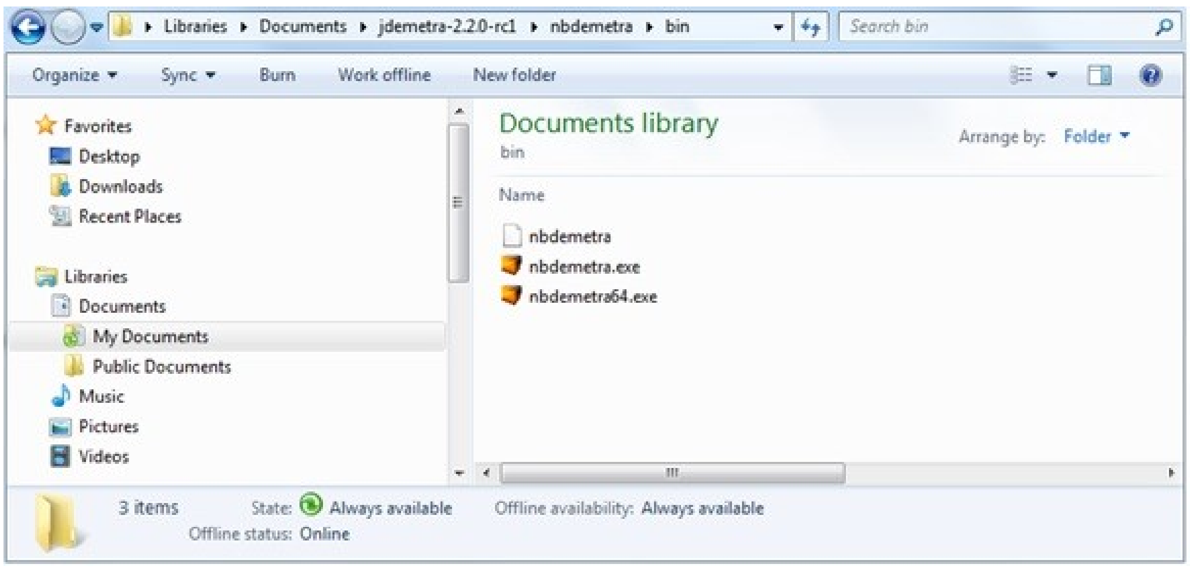


Figure 17: **Launching JDemetra+**

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).

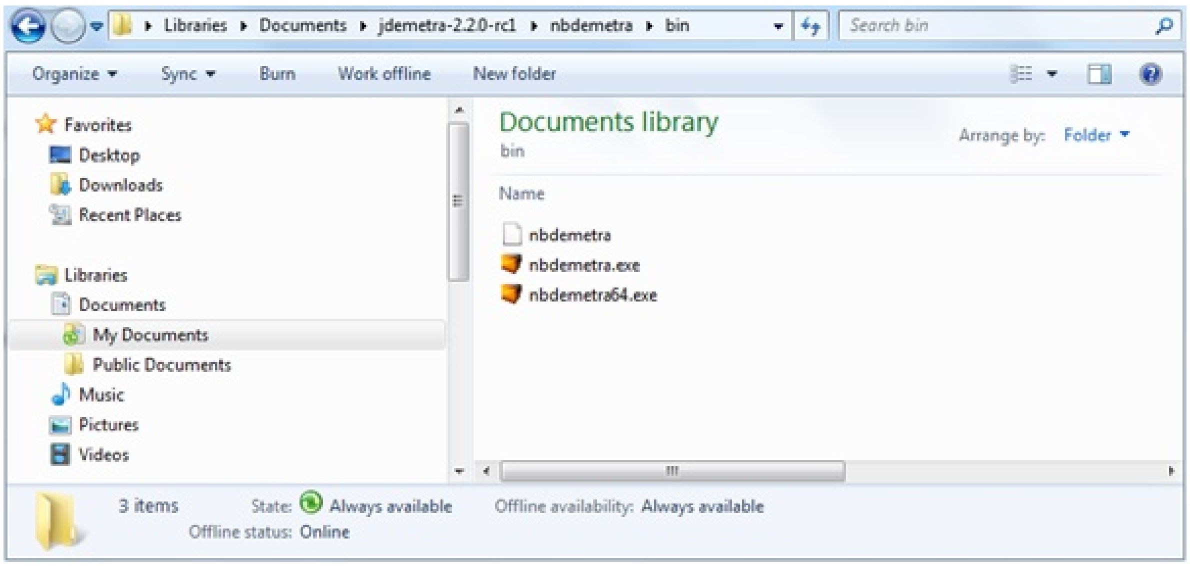


Figure 18: **Running JDemetra+**

Closing JDemetra+

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

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.

BELOW: menus and functions common to all the algos

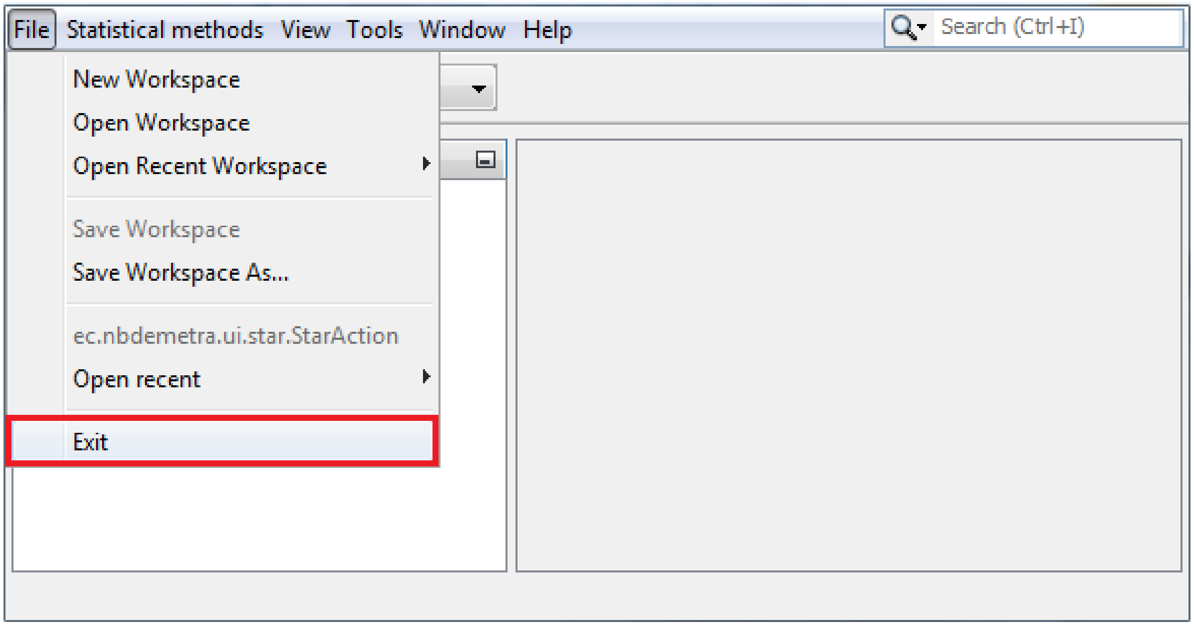


Figure 19: **Closing JDemetra+**

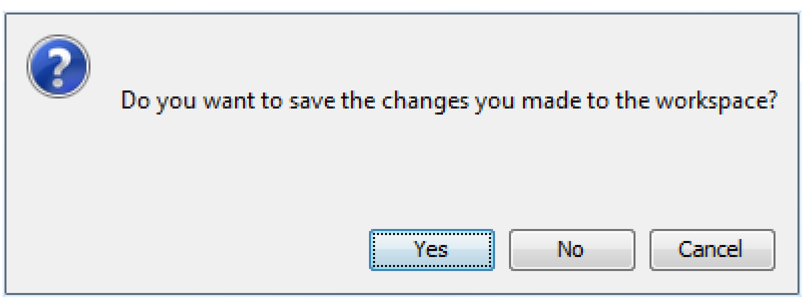


Figure 20: **The warning from leaving JDemetra+ without saving the workspace**

Interface Starting Winwow

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

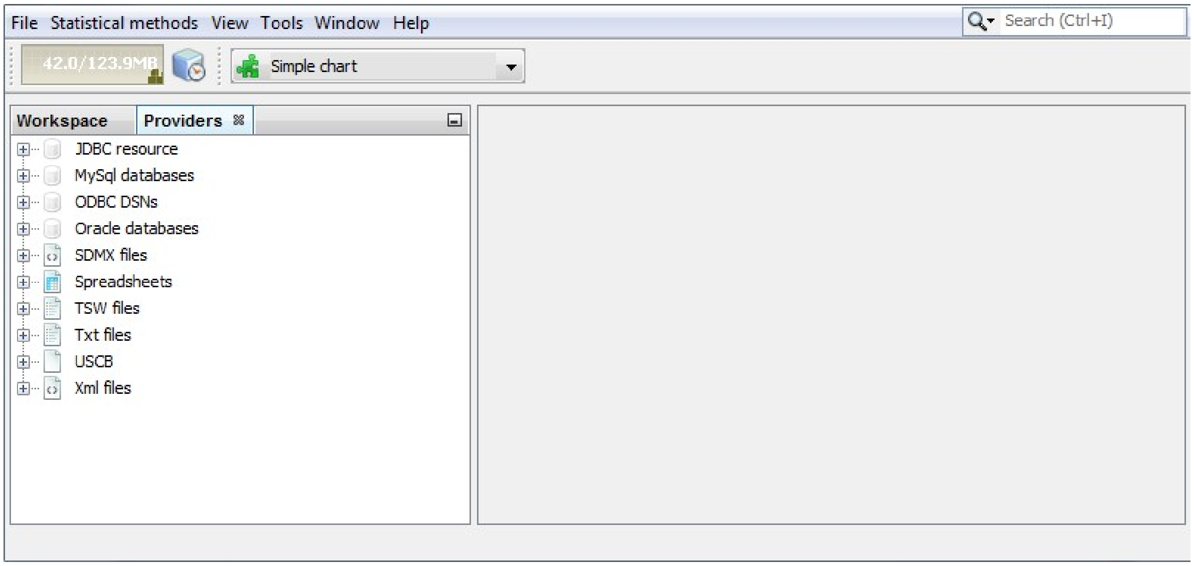


Figure 21: **JDemetra+ default window**

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.

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.

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

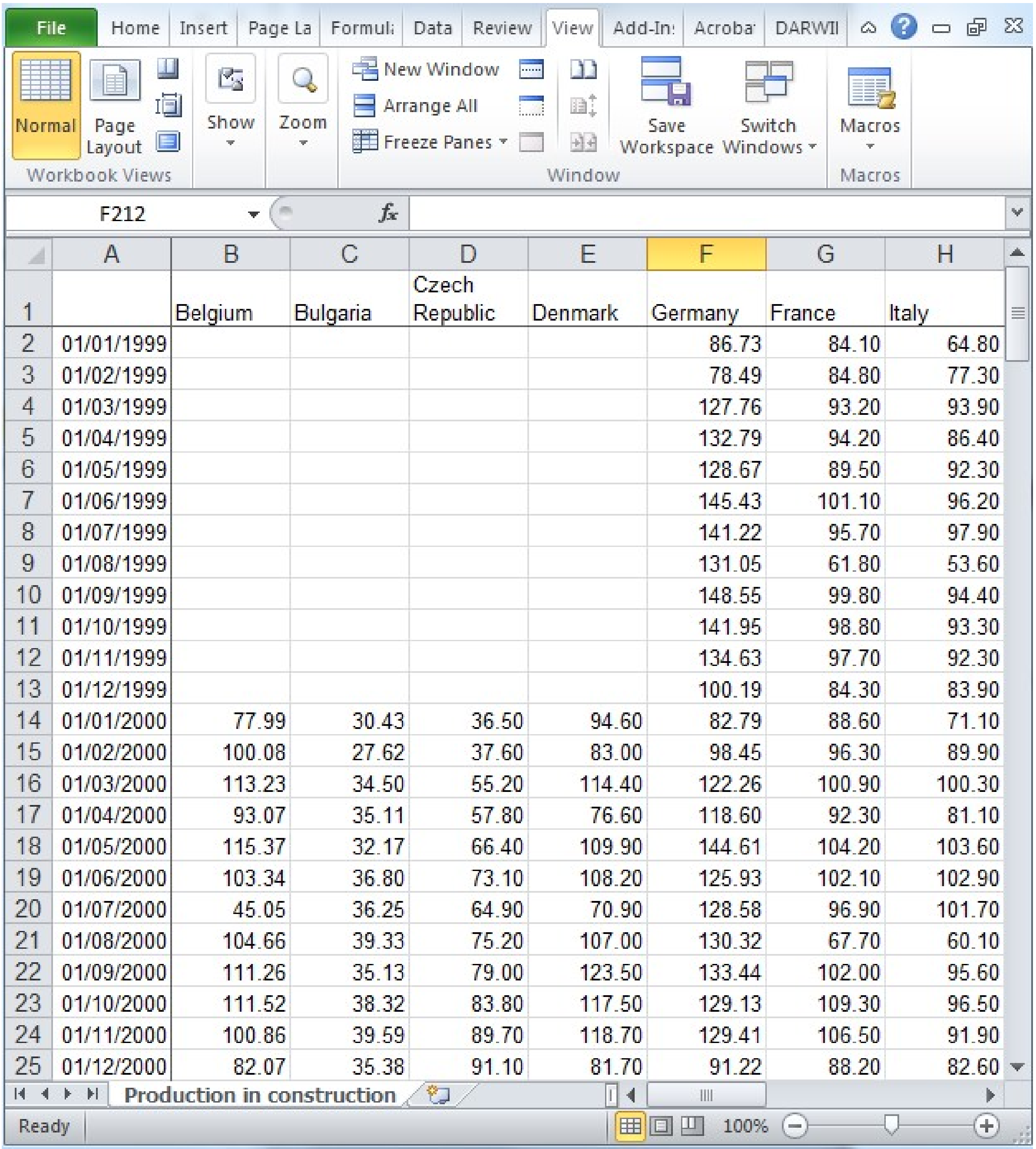
Search option

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 functions available in the main application menu are:



	A	B	C	D	E	F	G	H
1		Belgium	Bulgaria	Czech Republic	Denmark	Germany	France	Italy
2	01/01/1999					86.73	84.10	64.80
3	01/02/1999					78.49	84.80	77.30
4	01/03/1999					127.76	93.20	93.90
5	01/04/1999					132.79	94.20	86.40
6	01/05/1999					128.67	89.50	92.30
7	01/06/1999					145.43	101.10	96.20
8	01/07/1999					141.22	95.70	97.90
9	01/08/1999					131.05	61.80	53.60
10	01/09/1999					148.55	99.80	94.40
11	01/10/1999					141.95	98.80	93.30
12	01/11/1999					134.63	97.70	92.30
13	01/12/1999					100.19	84.30	83.90
14	01/01/2000	77.99	30.43	36.50	94.60	82.79	88.60	71.10
15	01/02/2000	100.08	27.62	37.60	83.00	98.45	96.30	89.90
16	01/03/2000	113.23	34.50	55.20	114.40	122.26	100.90	100.30
17	01/04/2000	93.07	35.11	57.80	76.60	118.60	92.30	81.10
18	01/05/2000	115.37	32.17	66.40	109.90	144.61	104.20	103.60
19	01/06/2000	103.34	36.80	73.10	108.20	125.93	102.10	102.90
20	01/07/2000	45.05	36.25	64.90	70.90	128.58	96.90	101.70
21	01/08/2000	104.66	39.33	75.20	107.00	130.32	67.70	60.10
22	01/09/2000	111.26	35.13	79.00	123.50	133.44	102.00	95.60
23	01/10/2000	111.52	38.32	83.80	117.50	129.13	109.30	96.50
24	01/11/2000	100.86	39.59	89.70	118.70	129.41	106.50	91.90
25	01/12/2000	82.07	35.38	91.10	81.70	91.22	88.20	82.60

Figure 22: **Example of an Excel spreadsheet that can be imported to JDemetra+**

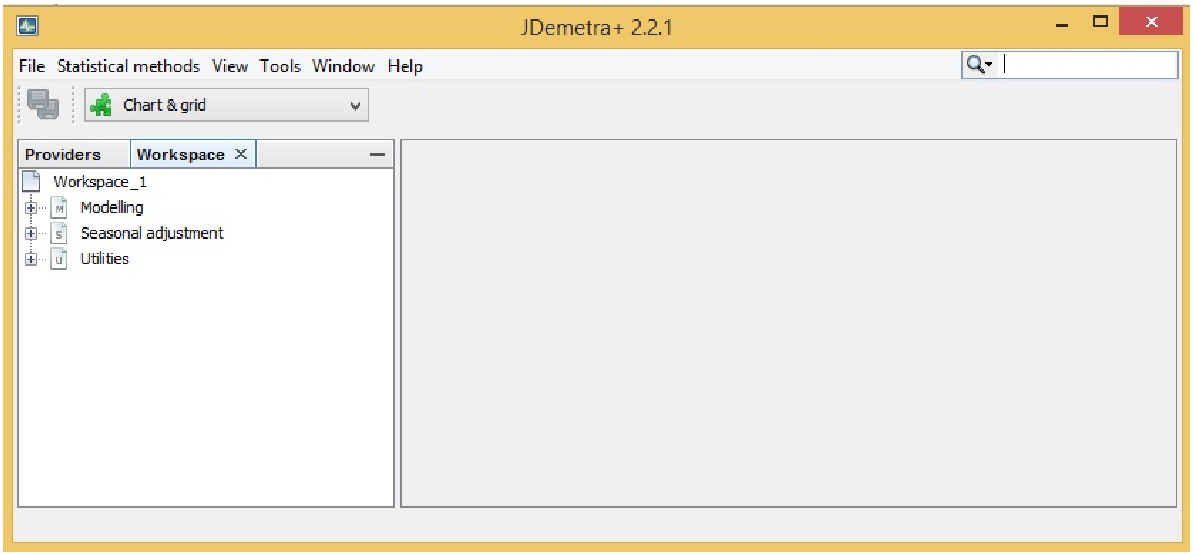


Figure 23: **JDemetra+ default view**

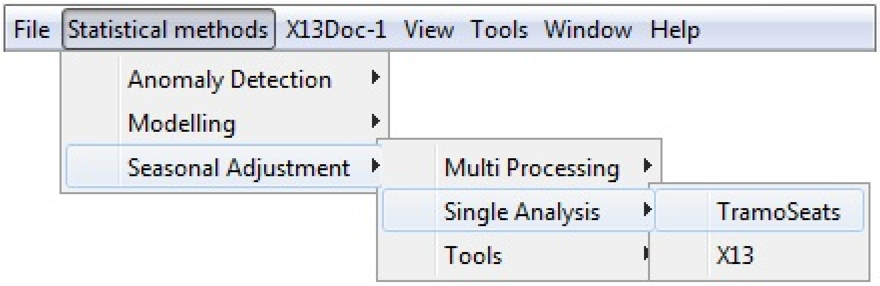


Figure 24: **The main menu with selected drop-down menu**

- File
- Statistical methods
- X-13Doc
- RegArimaDoc
- TramoDoc
- TramoSeatsDoc
- View
- Tools
- Window
- Help

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.

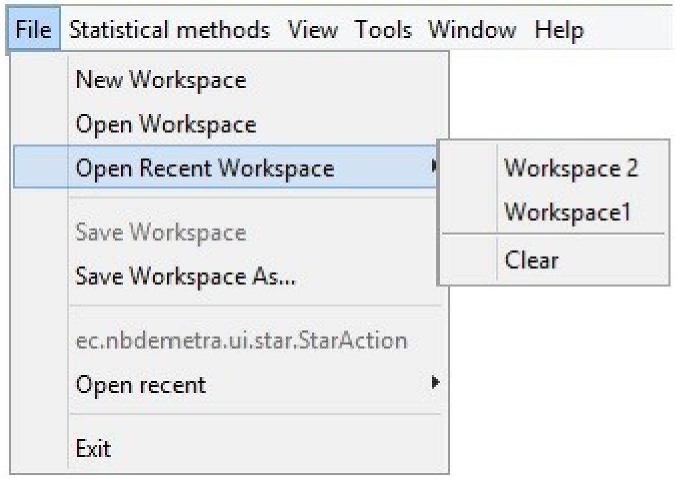


Figure 25: **The content of the *File* menu**

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.

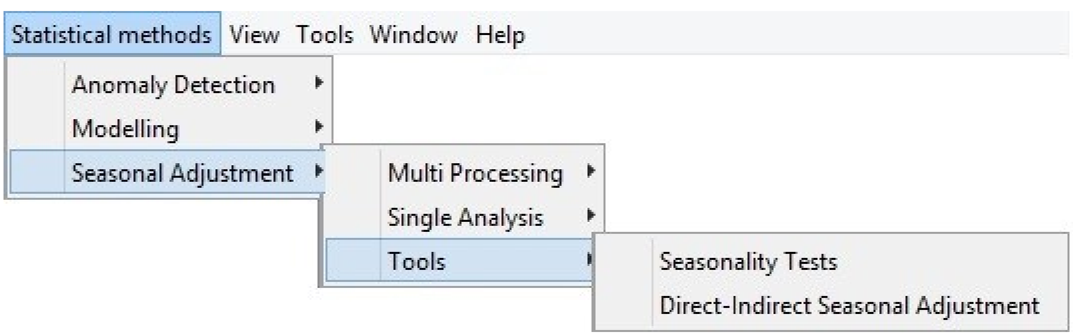


Figure 26: **The *Statistical methods* menu.**

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.

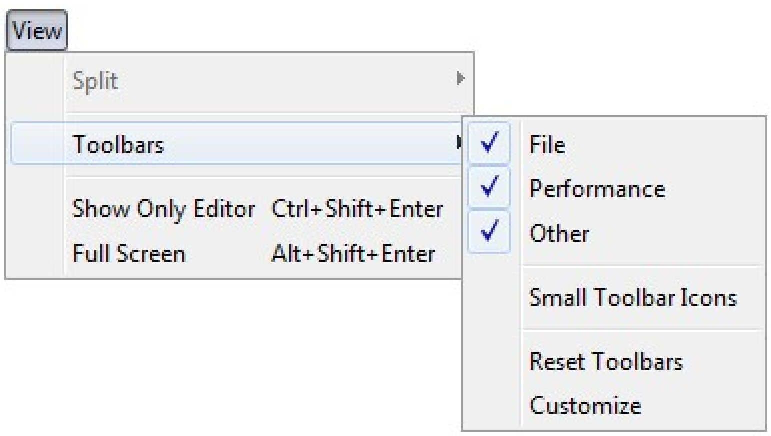


Figure 27: **The *View* menu**

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.

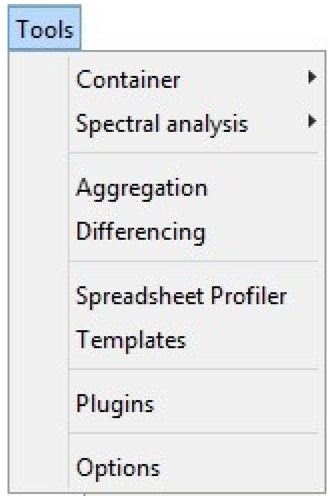


Figure 28: **The *Tools* menu**

Container

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

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

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.

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.

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.

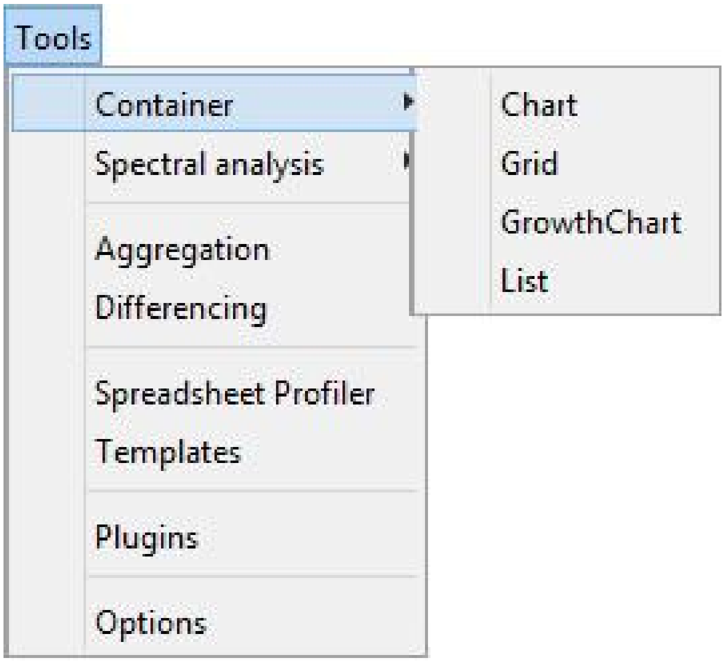


Figure 29: **The *Container* menu**

Figure 30: **Launching the *Chart* functionality**

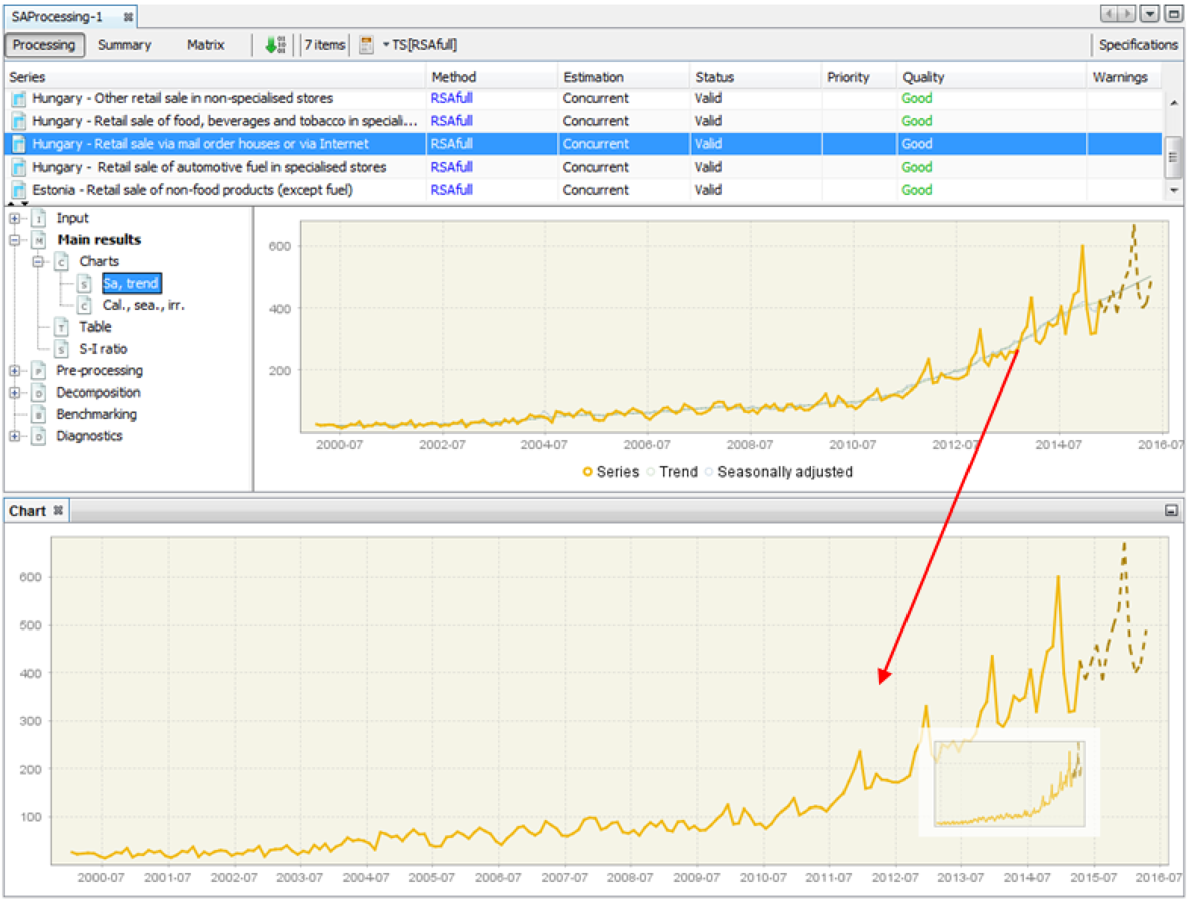


Figure 31: **Displaying the seasonally adjusted series on a separate chart**

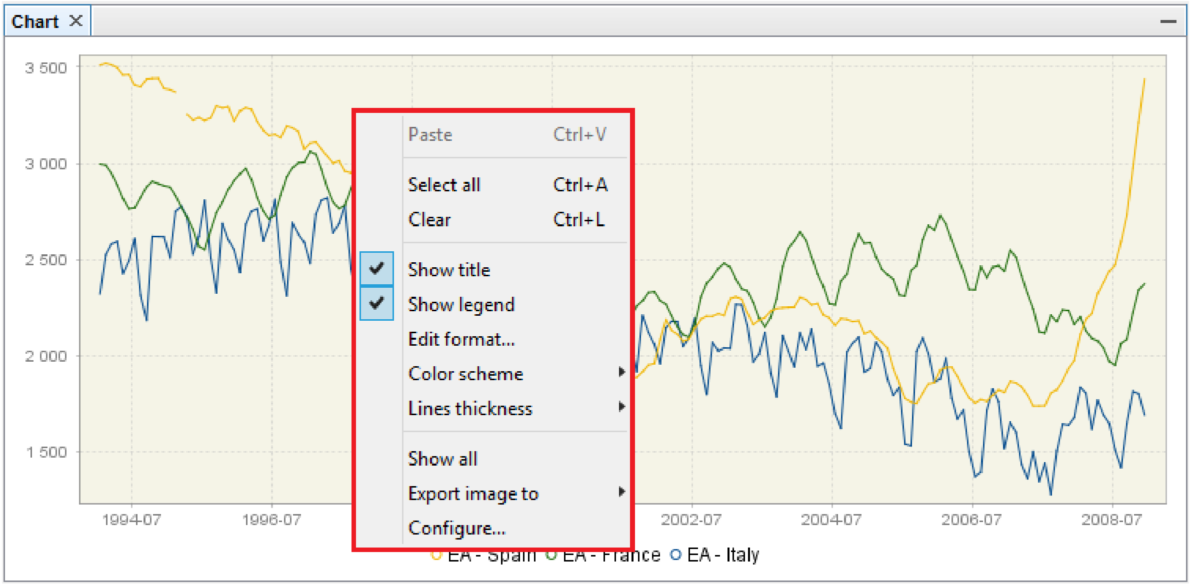


Figure 32: **Local menu basic options for the time series graph**

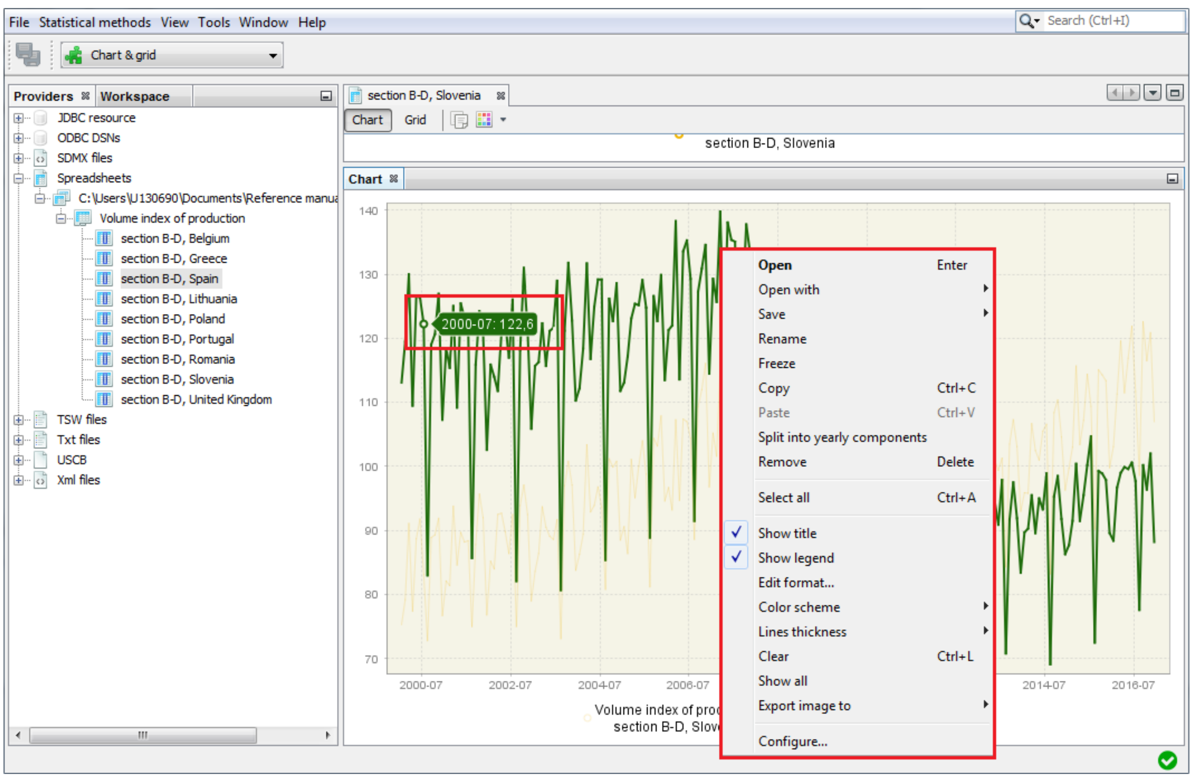


Figure 33: **Local menu options for chart**

A list of possible actions includes:

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

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

The options available in *Grid* are:

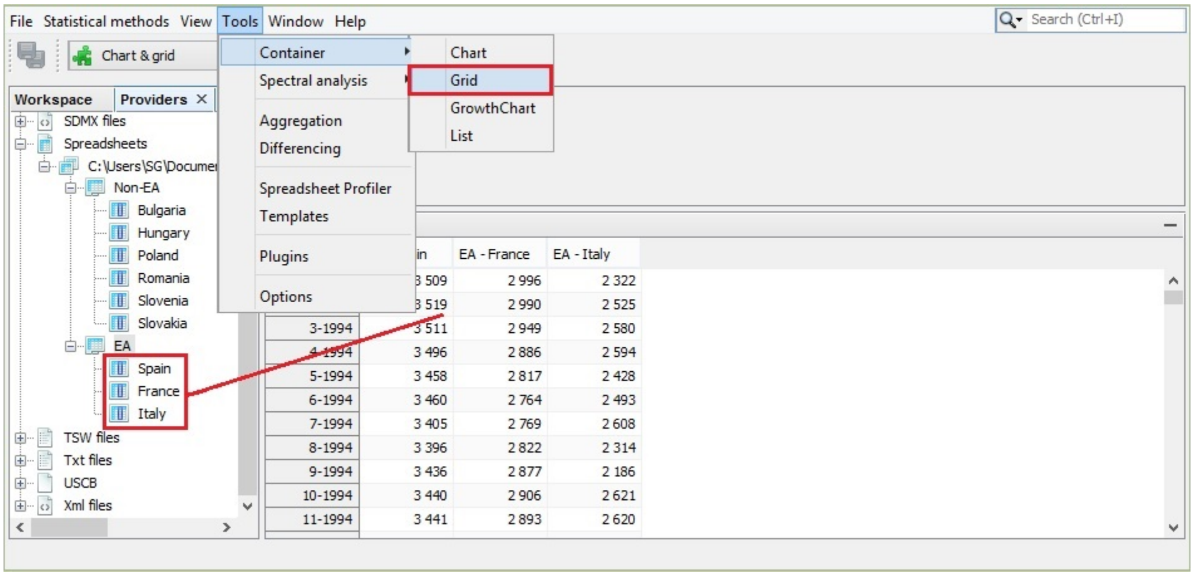


Figure 34: **Launching the *Grid* functionality**

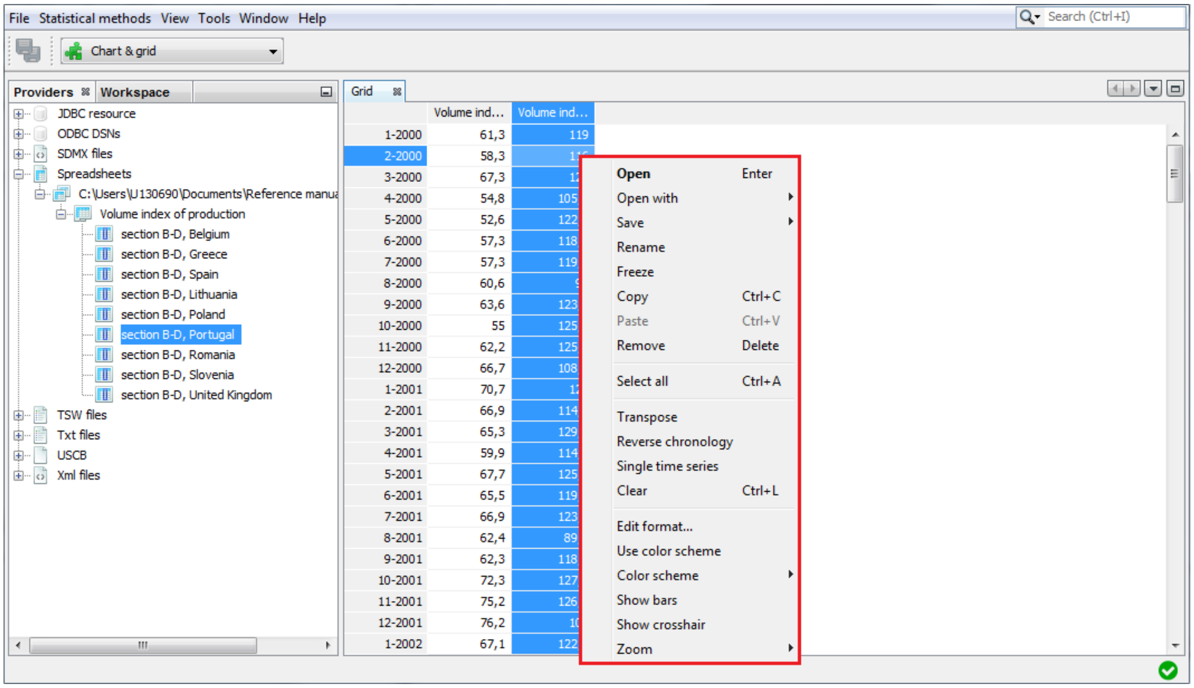


Figure 35: **Local menu options for the *Grid* view**

- **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.
- **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.

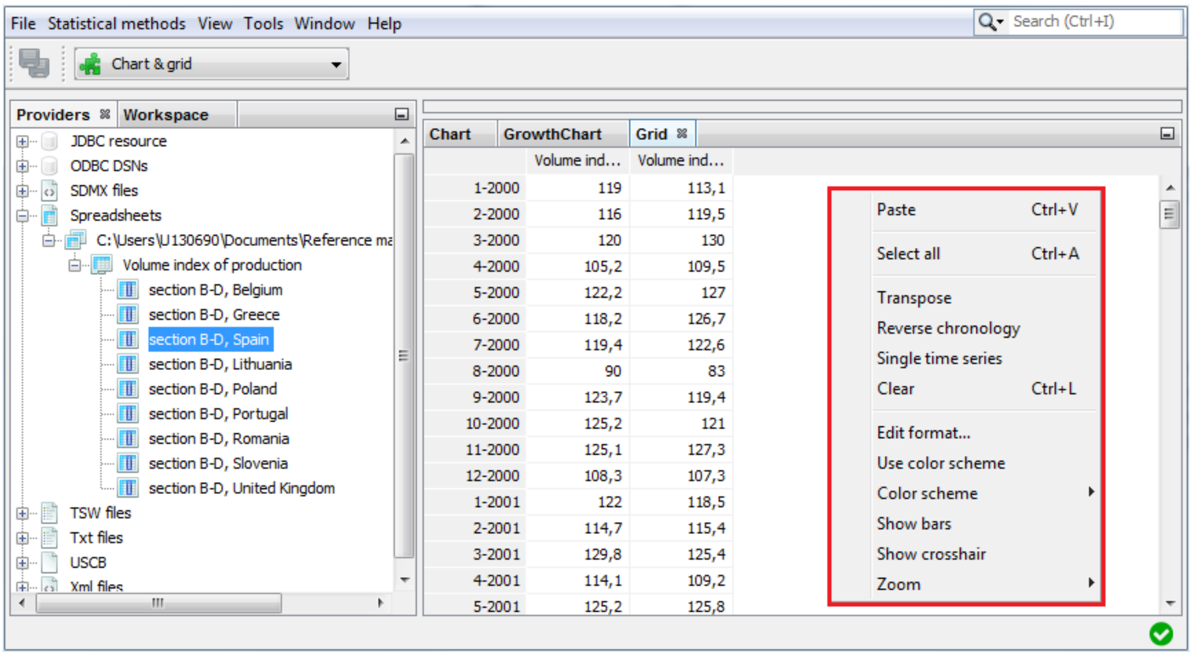


Figure 36: **A reduced list of options for *Grid***

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.

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

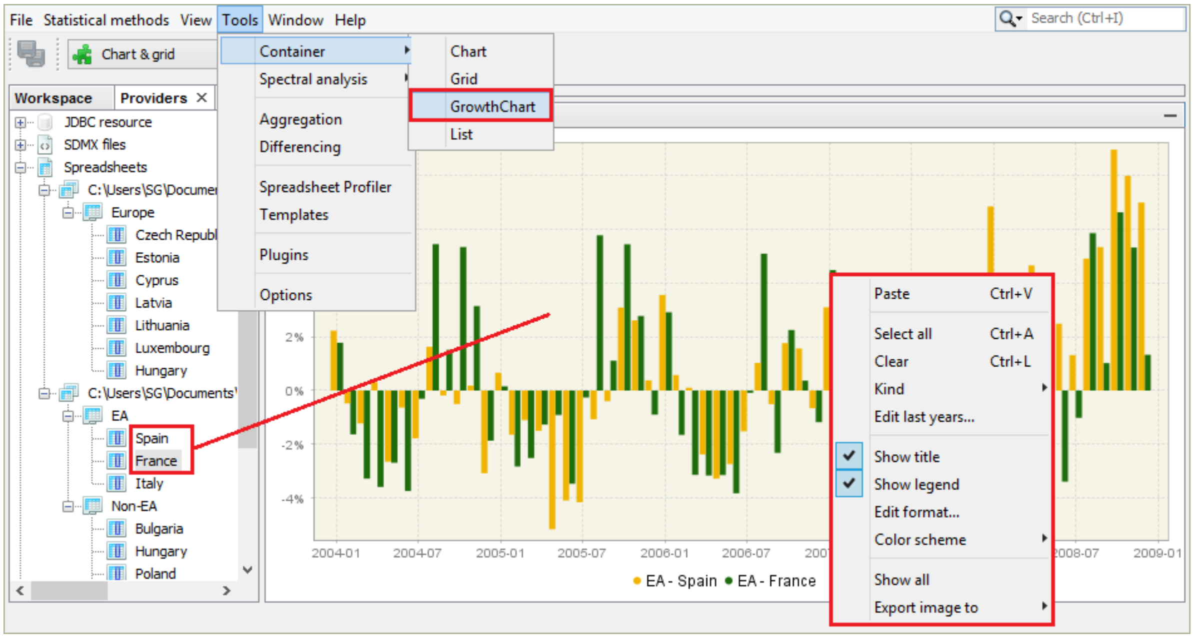


Figure 37: **The *Growth chart* view with a local menu**

- **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.

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.

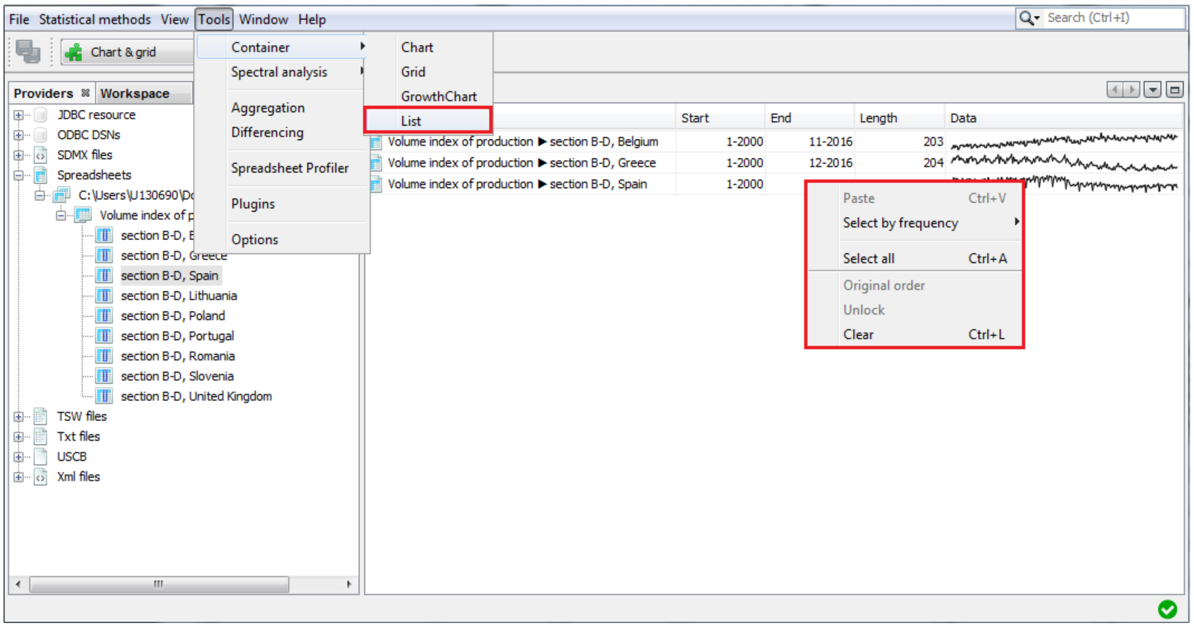


Figure 38: **A view of a list of series**

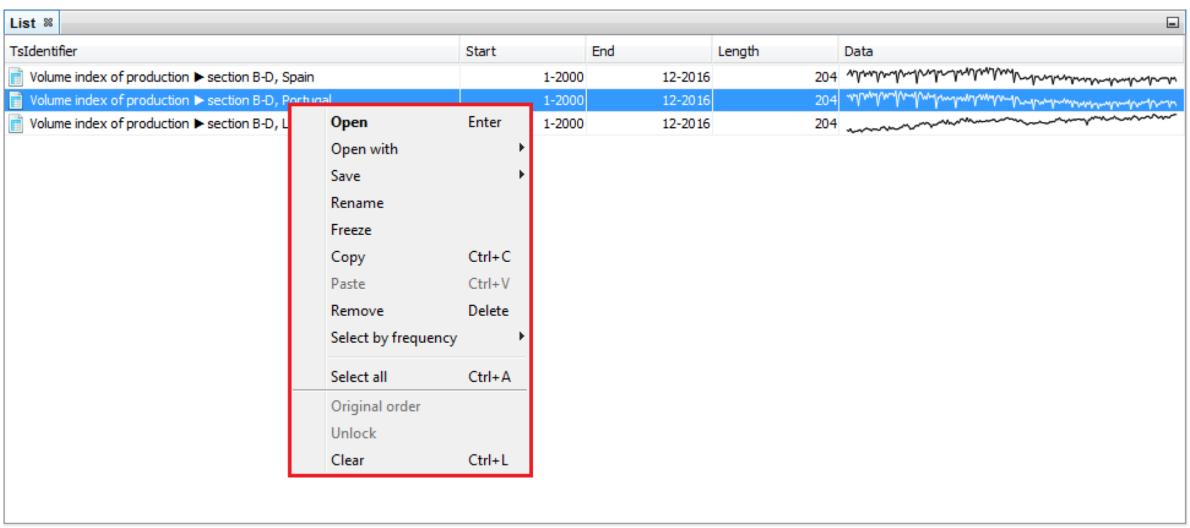


Figure 39: **Options available for a selected series from the list**

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.

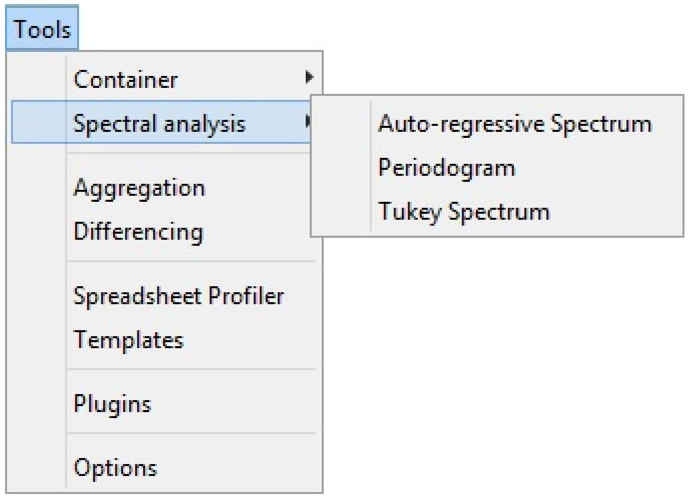


Figure 40: **Tools for spectral analysis**

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*).

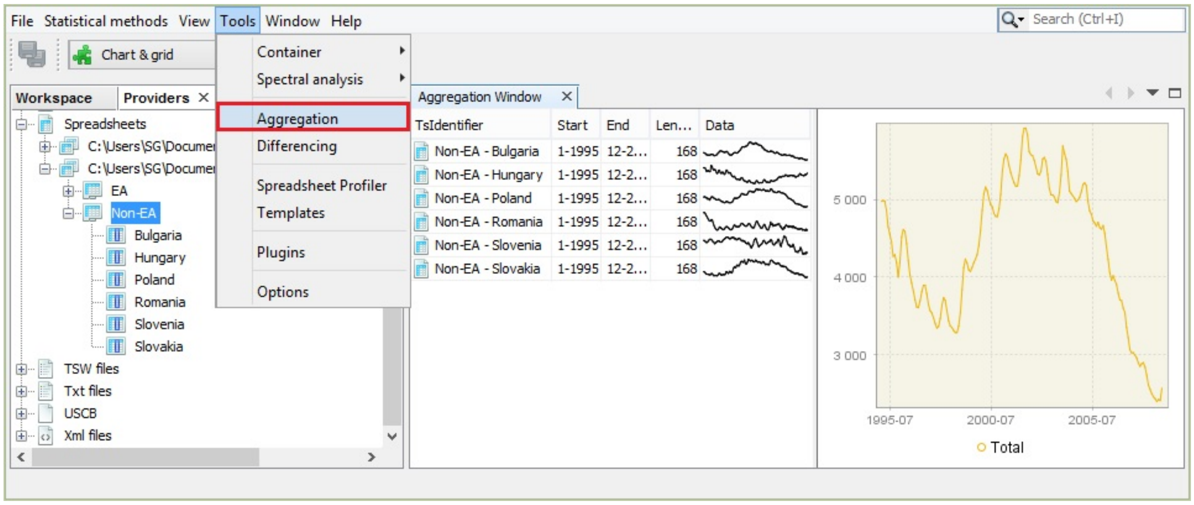


Figure 41: **The *Aggregation* tool**

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* → *Properties*). A description of a periodogram and the PACF function can be found here.

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.

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.

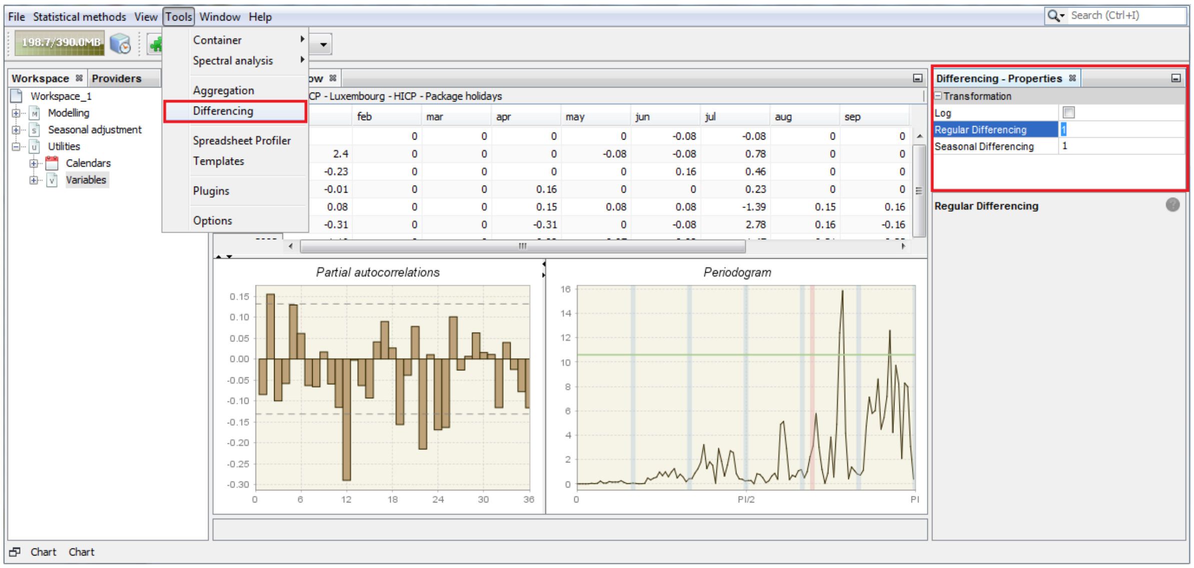


Figure 42: **The properties of the *Differencing* tool**

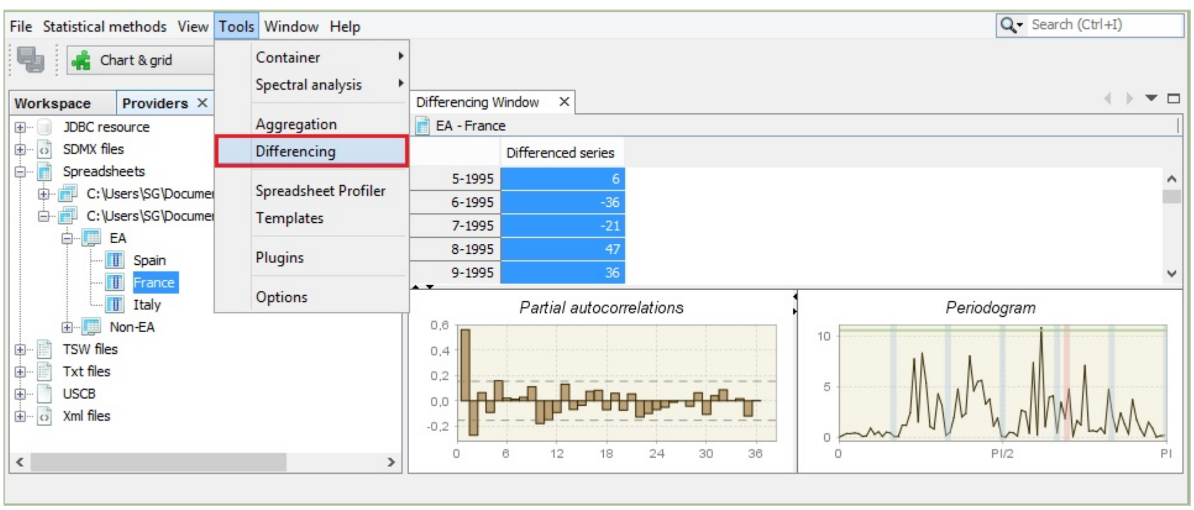
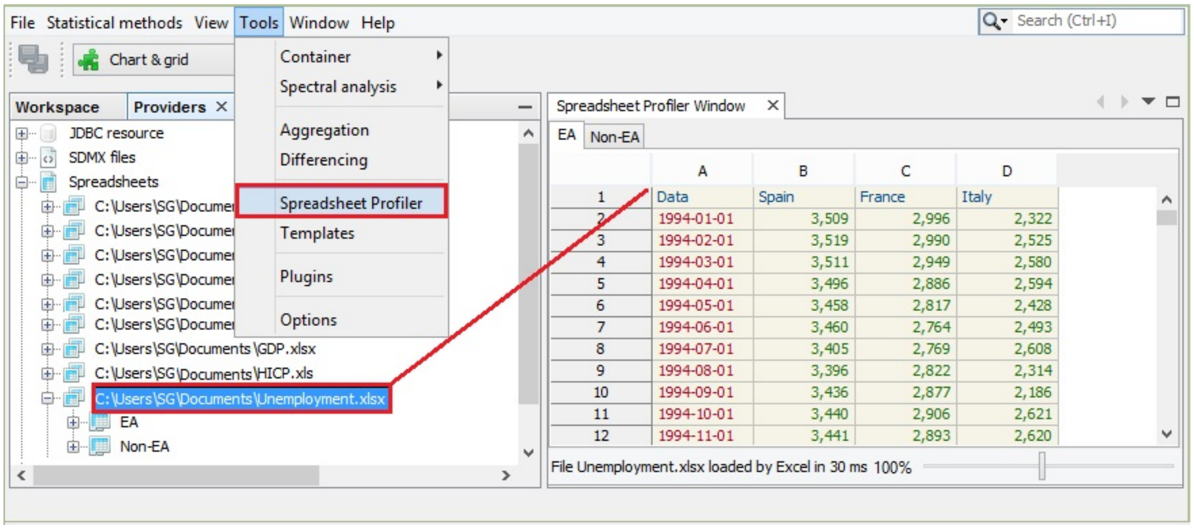


Figure 43: **The *Differencing* tool**



	A	B	C	D
1	Data	Spain	France	Italy
2	1994-01-01	3,509	2,996	2,322
3	1994-02-01	3,519	2,990	2,525
4	1994-03-01	3,511	2,949	2,580
5	1994-04-01	3,496	2,886	2,594
6	1994-05-01	3,458	2,817	2,428
7	1994-06-01	3,460	2,764	2,493
8	1994-07-01	3,405	2,769	2,608
9	1994-08-01	3,396	2,822	2,314
10	1994-09-01	3,436	2,877	2,186
11	1994-10-01	3,440	2,906	2,621
12	1994-11-01	3,441	2,893	2,620

Figure 44: **The *Spreadsheet Profiler* window**

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.

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.

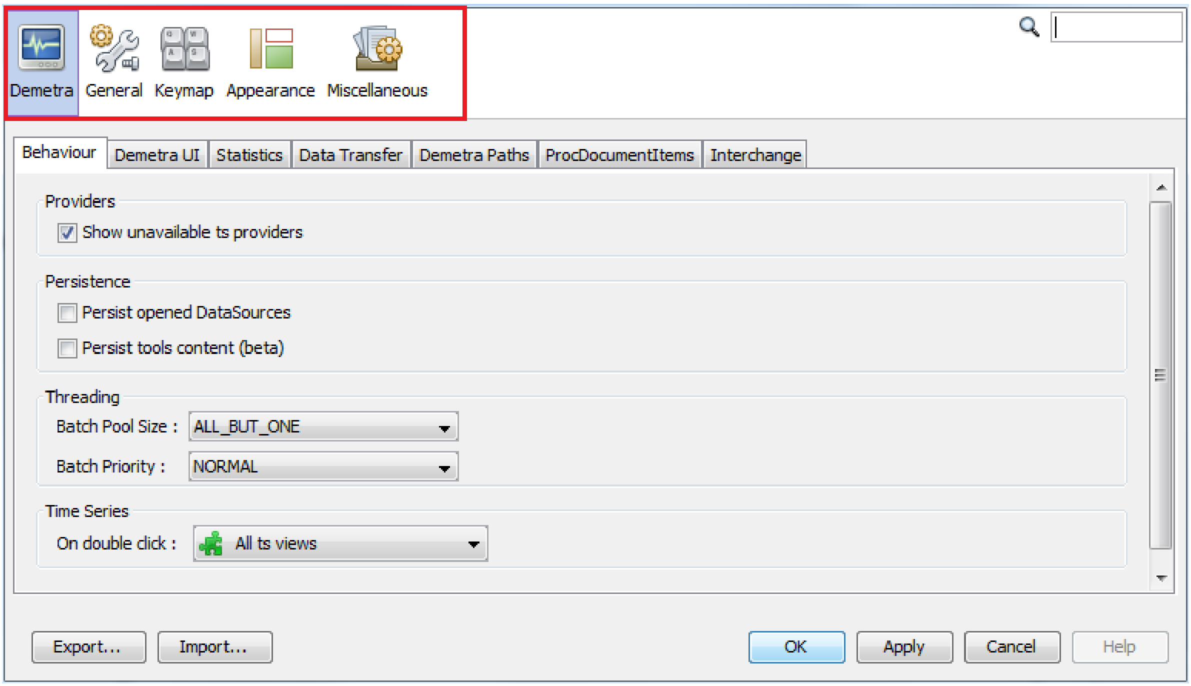


Figure 45: **The main sections of the *Options* window**

- **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.

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**).

The *Statistics* tab includes options to control:

- The number of years used for spectral analysis and for model stability (**Default Number of Last Years**);

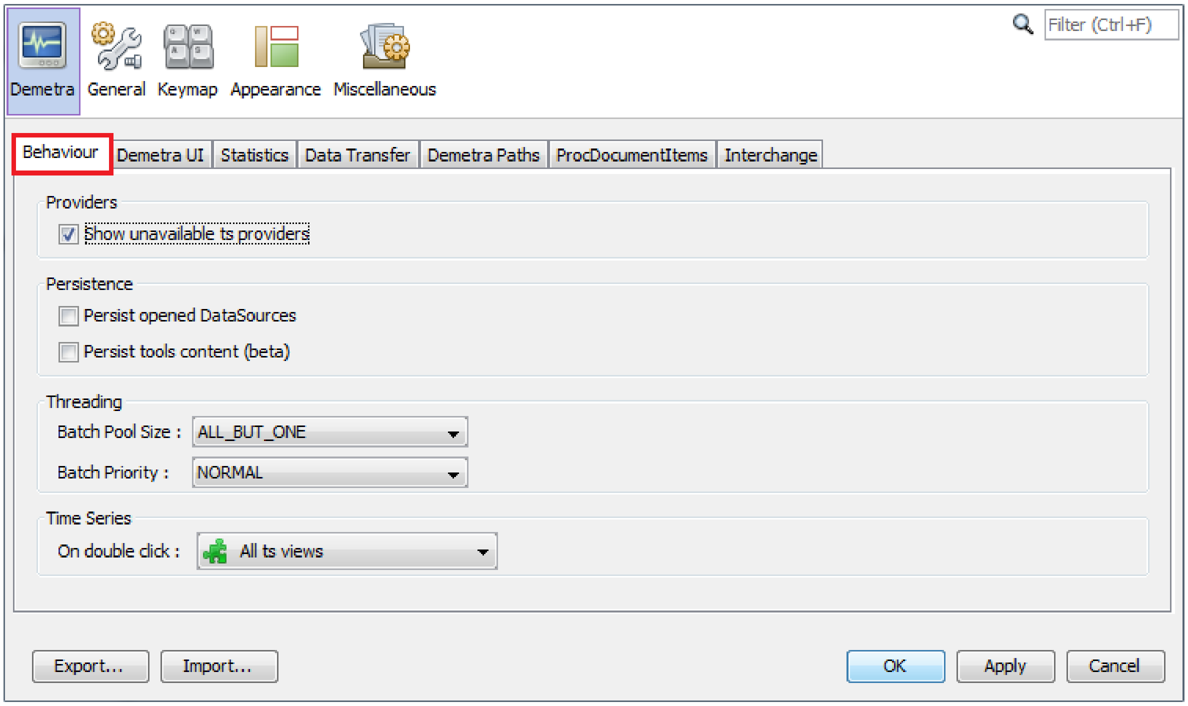


Figure 46: **The content of the *Behavior* tab**

- 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)

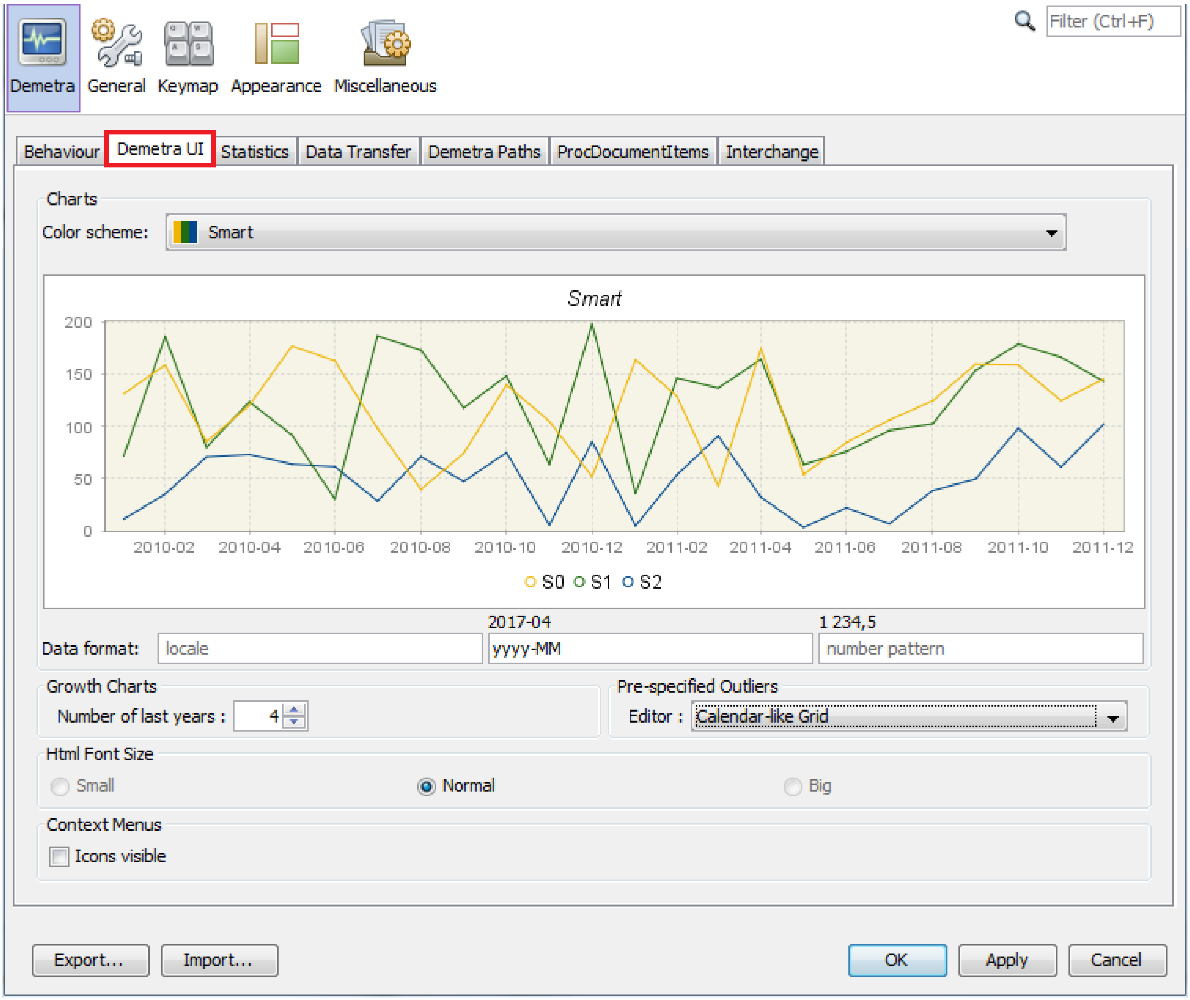


Figure 47: **The content of the *Demetra UI* tab**

– **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).

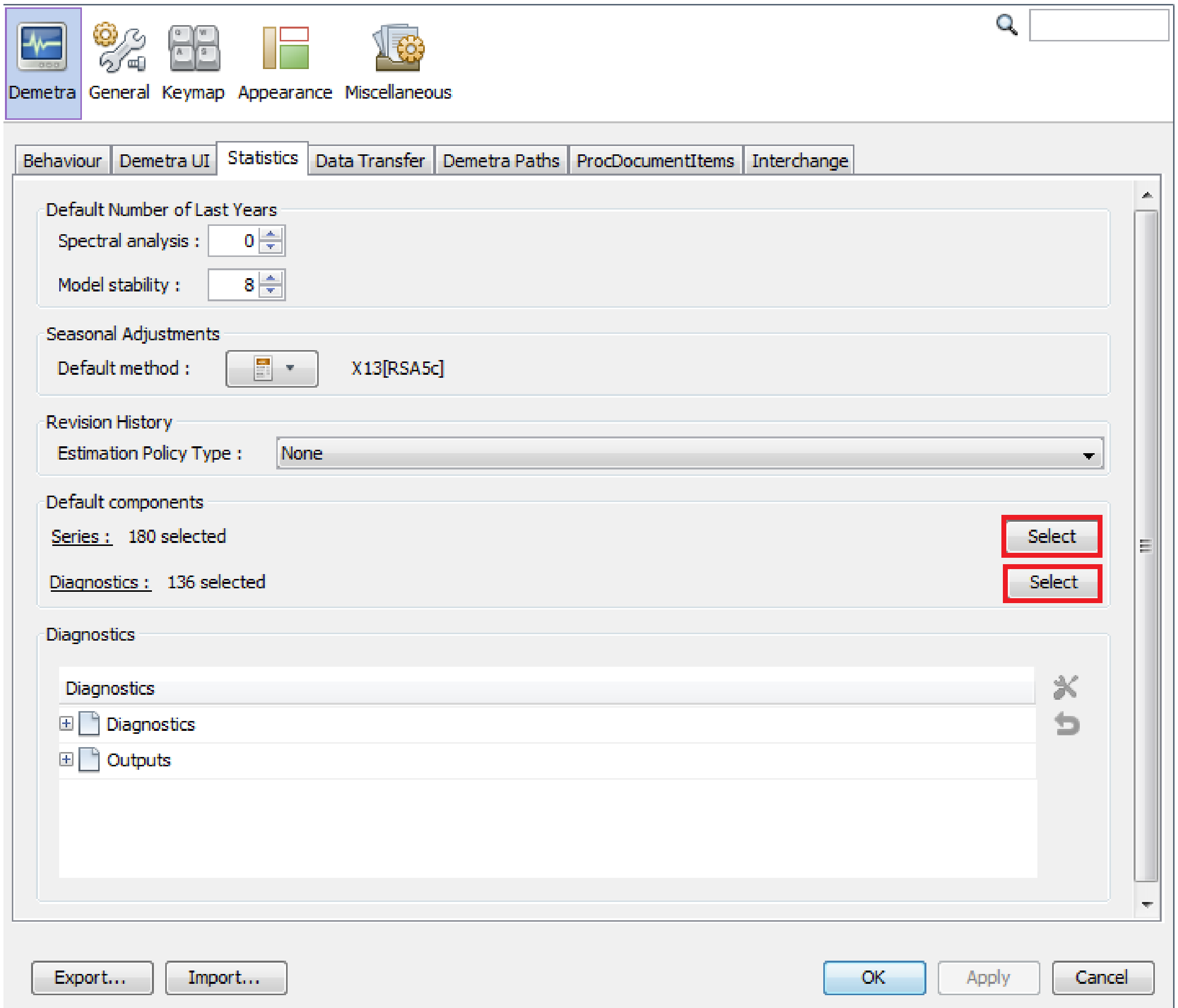


Figure 48: **The *Default components* section on the *Statistics* tab**

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.

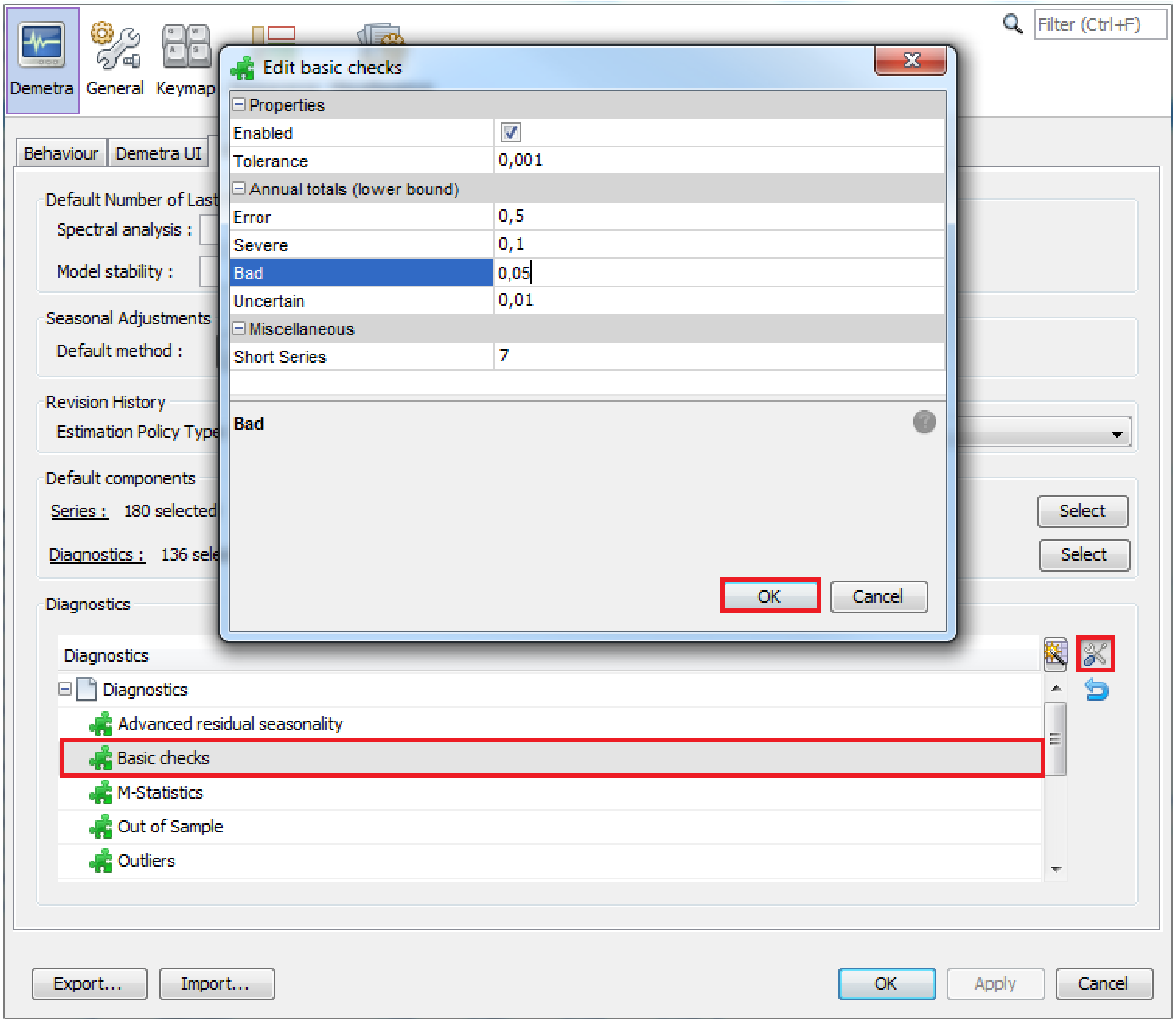


Figure 49: **The panel for modification of the settings for the tests in the *Basic checks* section**

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 *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.

Demetra Paths allows the user to specify the relative location of the folders where the data can

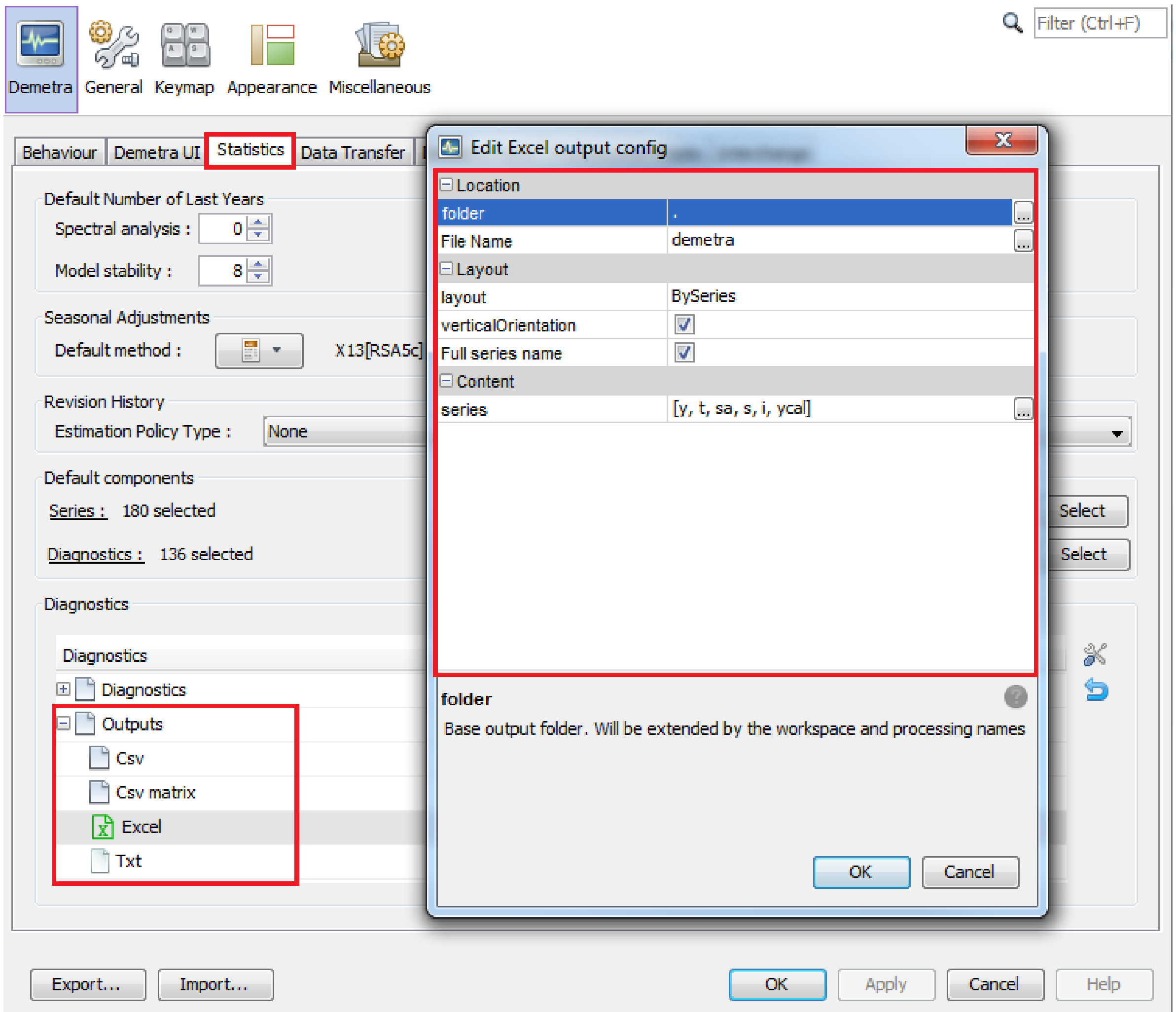


Figure 50: **The settings of the output files**

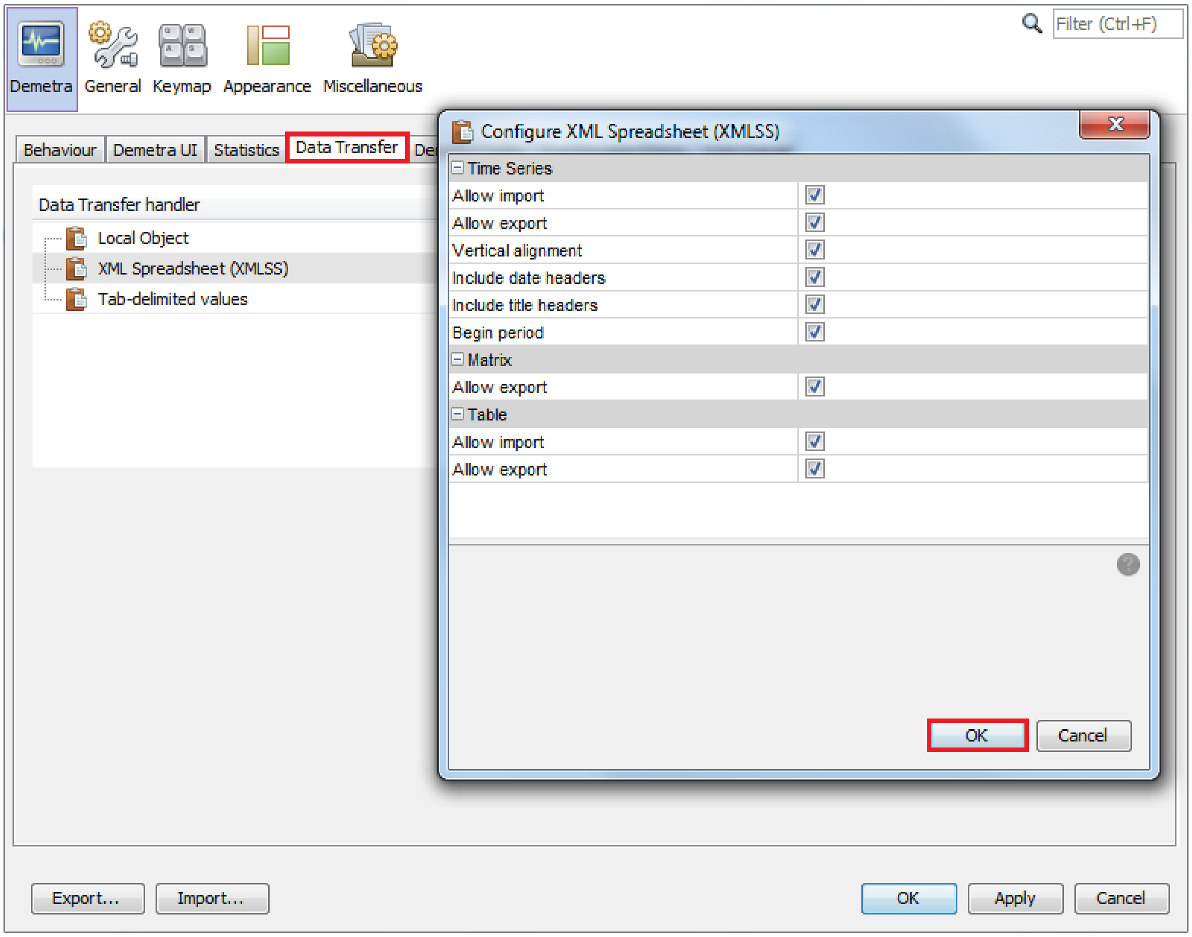


Figure 51: **The content of the *Data Transfer* tab**

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.

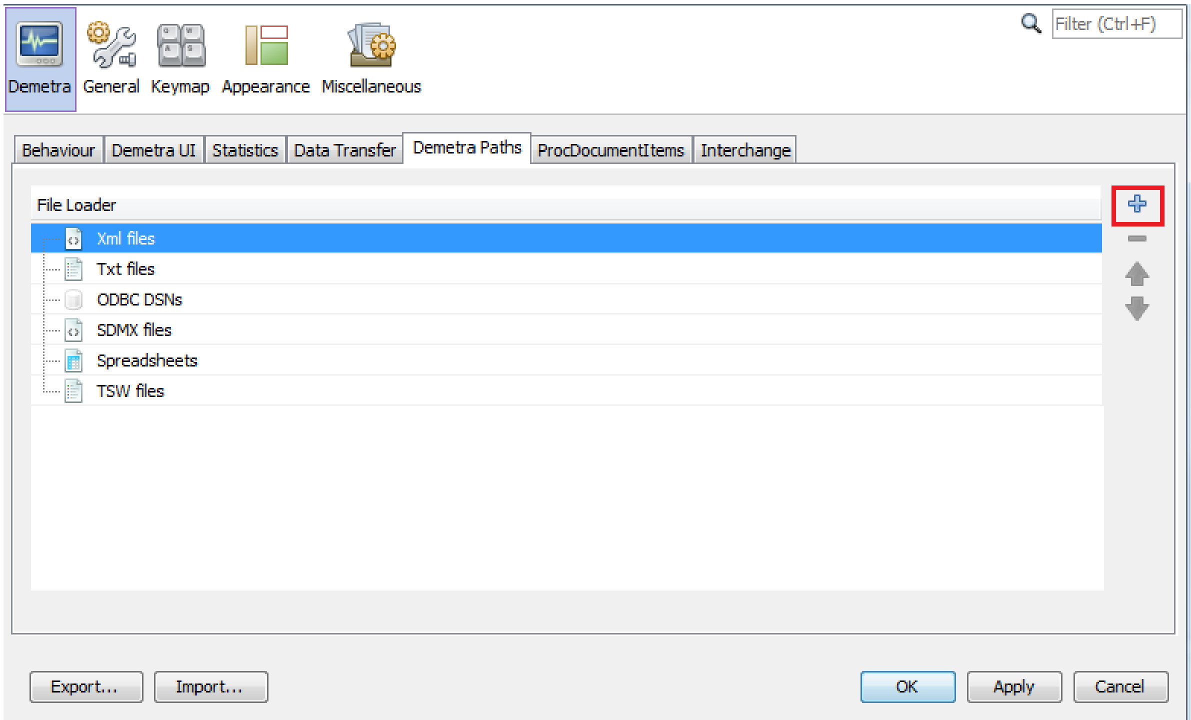


Figure 52: **The content of the *Demetra Paths* tab**

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.

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.

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

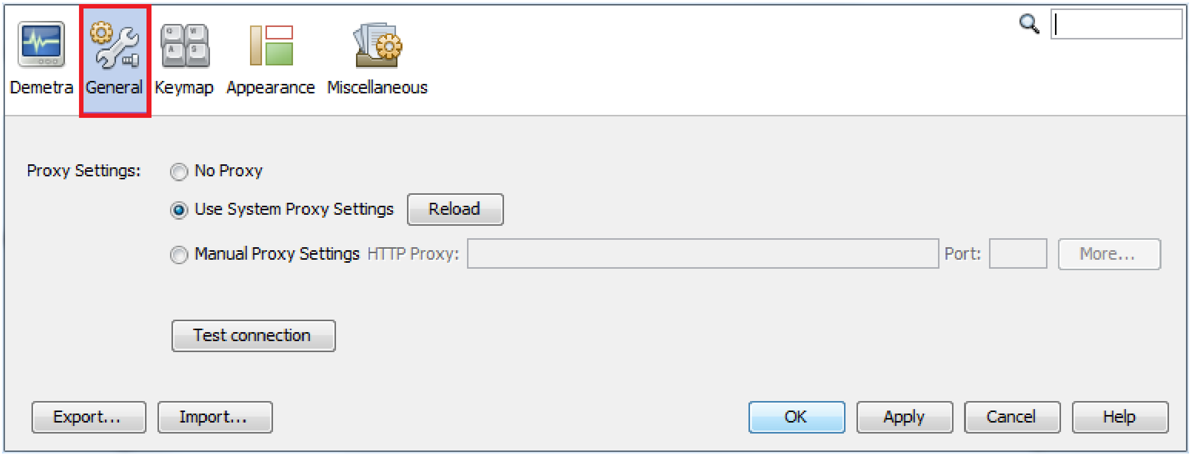


Figure 53: **The *General* tab**

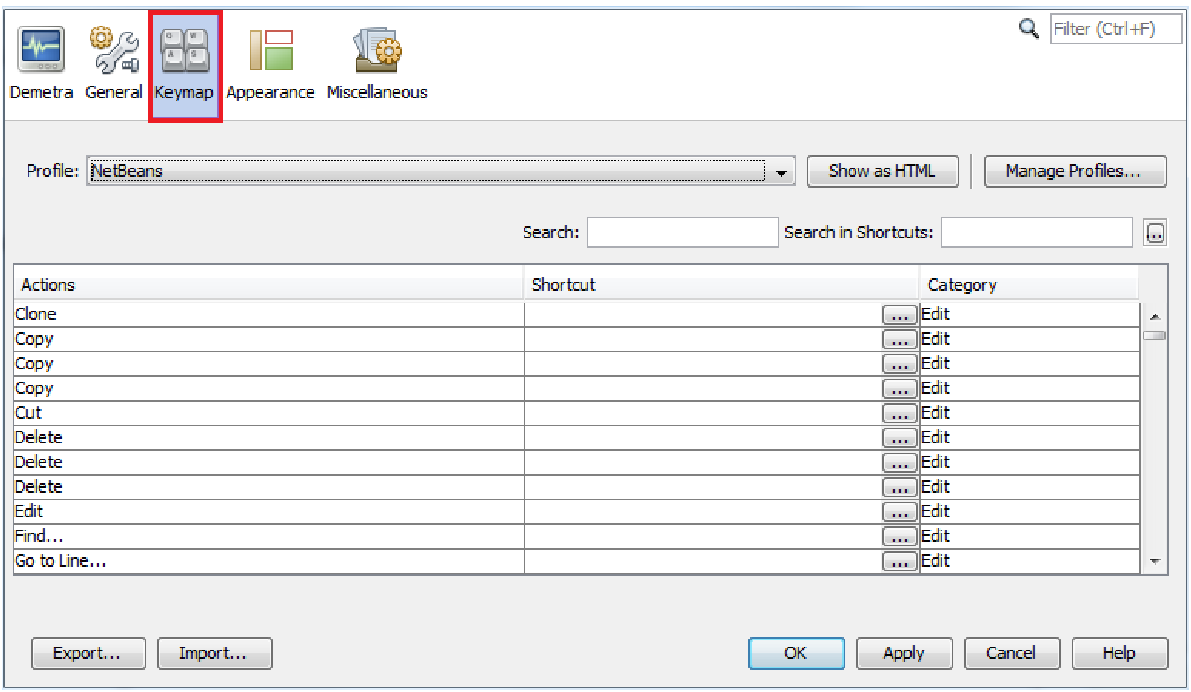


Figure 54: **The *Keymap* tab**

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.

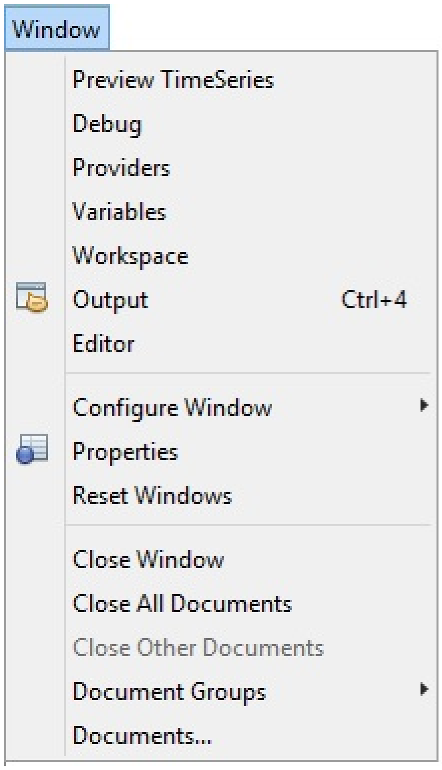


Figure 55: **The *Window* menu**

- **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.

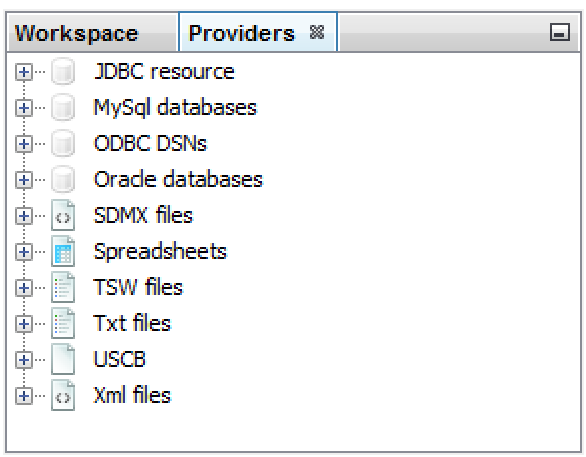


Figure 56: **The *Providers* window**

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 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.

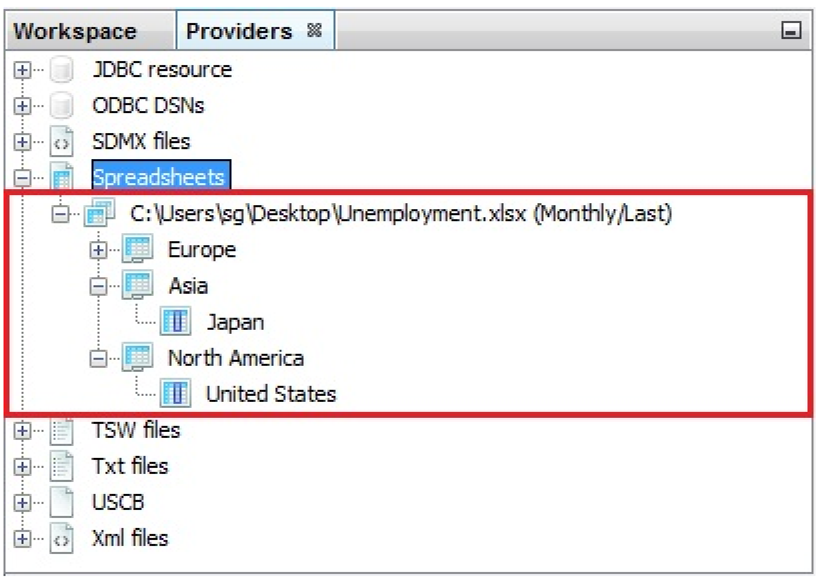


Figure 57: **A structure of a dataset**

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 58: **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.

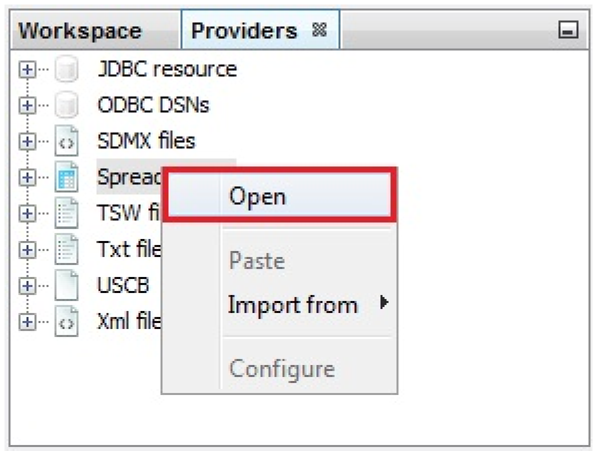


Figure 59: **Data provider available by default**

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.

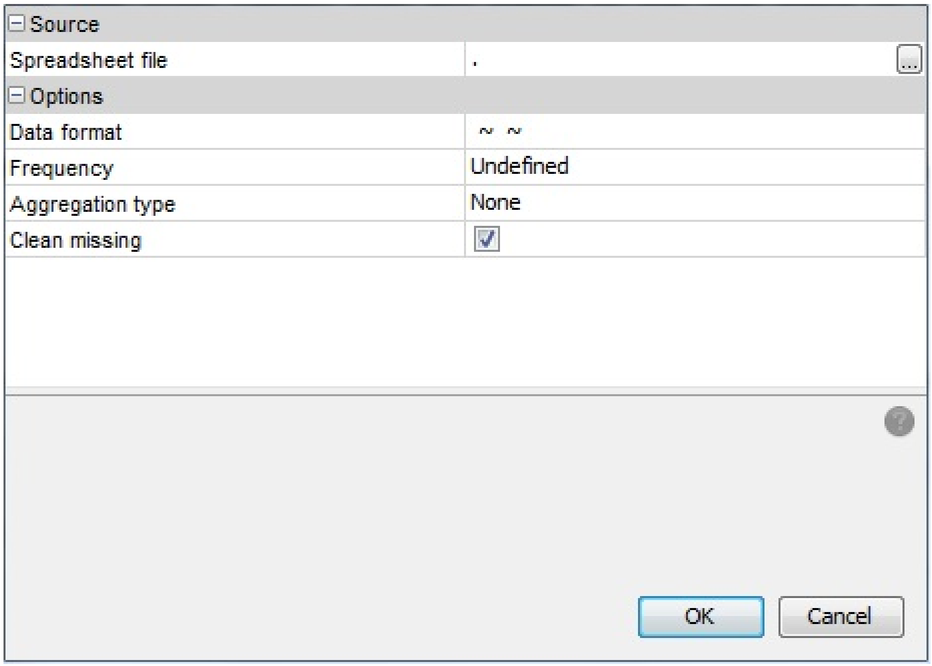


Figure 60: **Data source window**

3. Choose a file and click *OK*.
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.
5. The data are organized in a tree structure.

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

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.

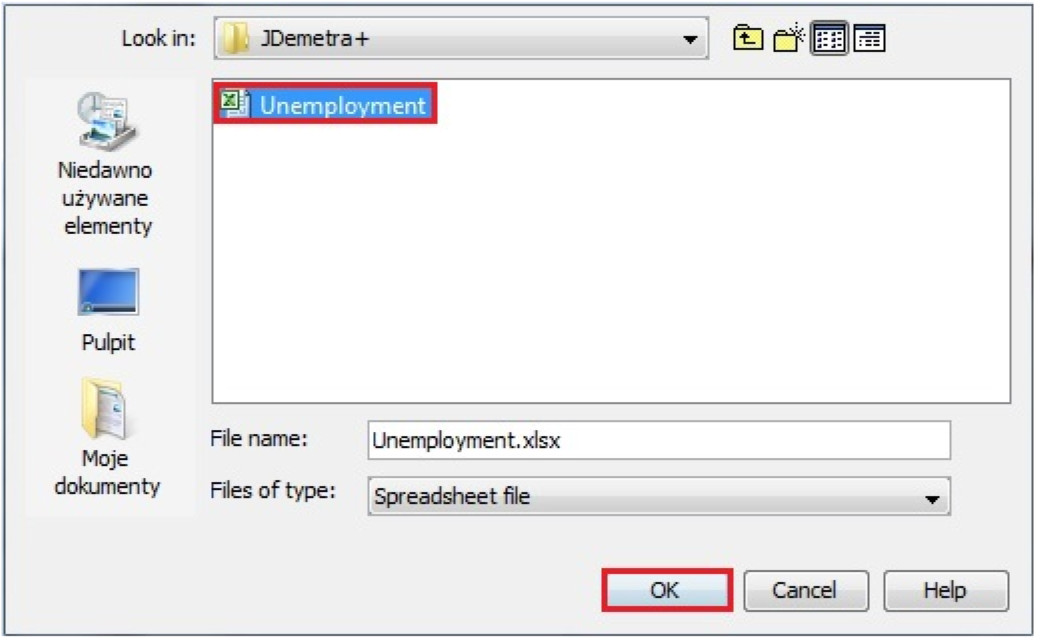


Figure 61: **Choice of an Excel spreadsheet**

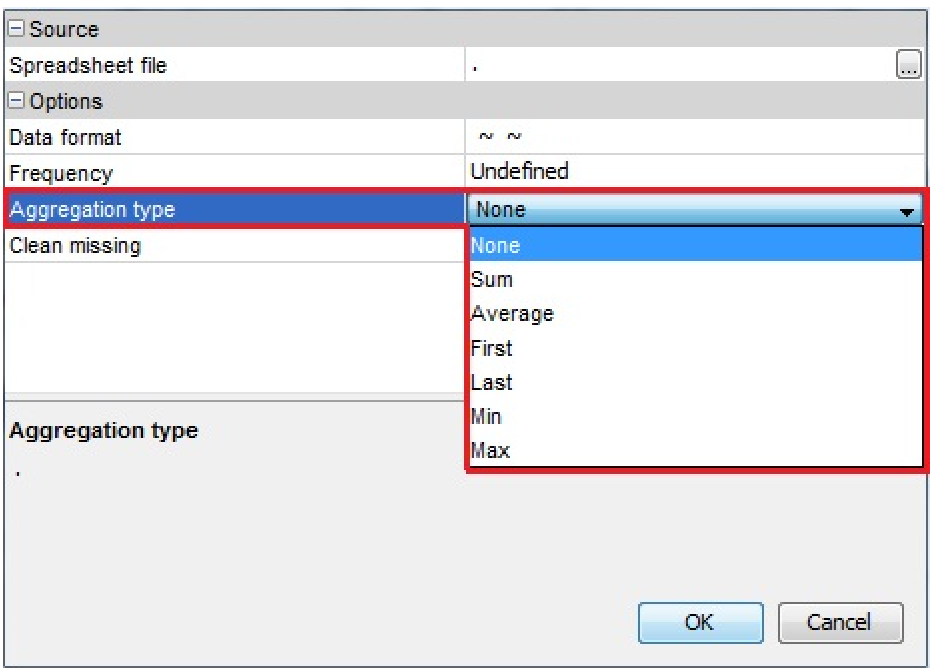


Figure 62: **Options for importing data**

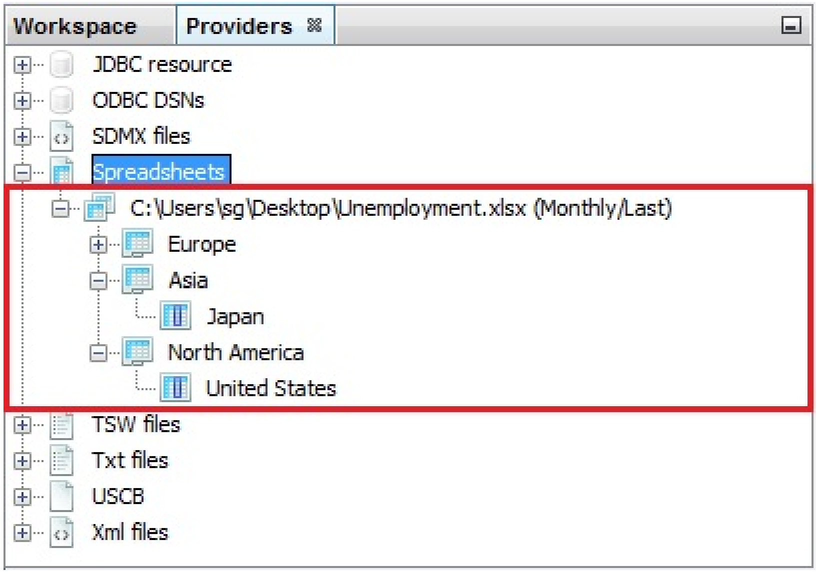


Figure 63: **Dataset structure**

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.

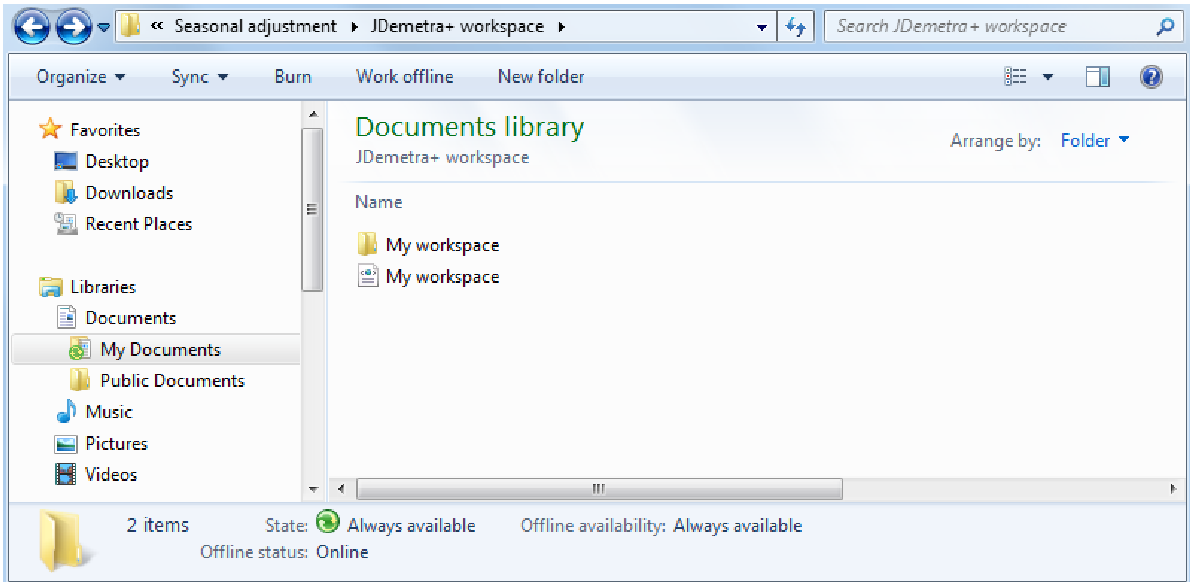


Figure 64: **A workspace saved on PC**

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).

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

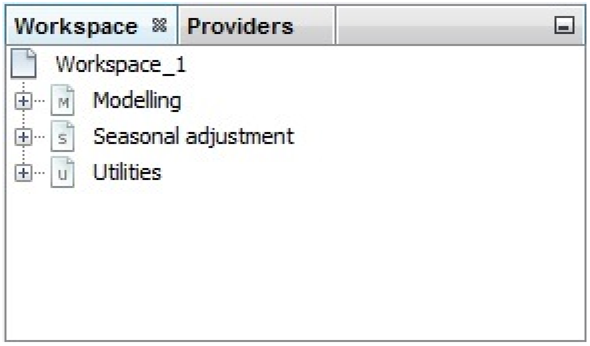


Figure 65: **The *Workspace* window**

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).

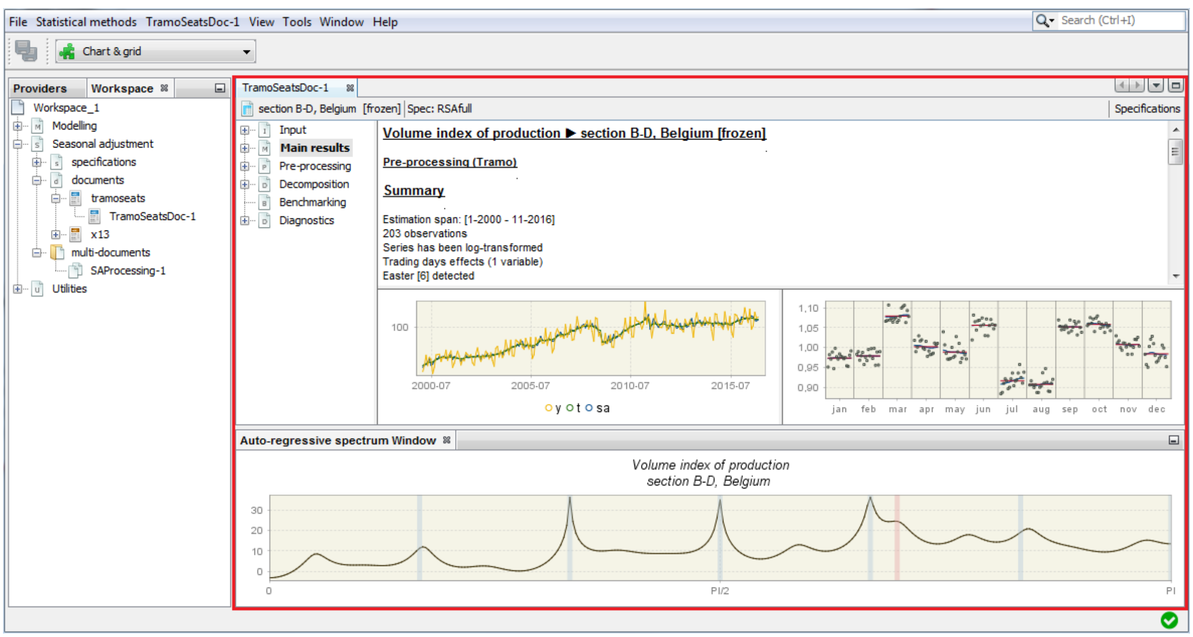


Figure 66: **The *Results* panel filled with two windows**

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.

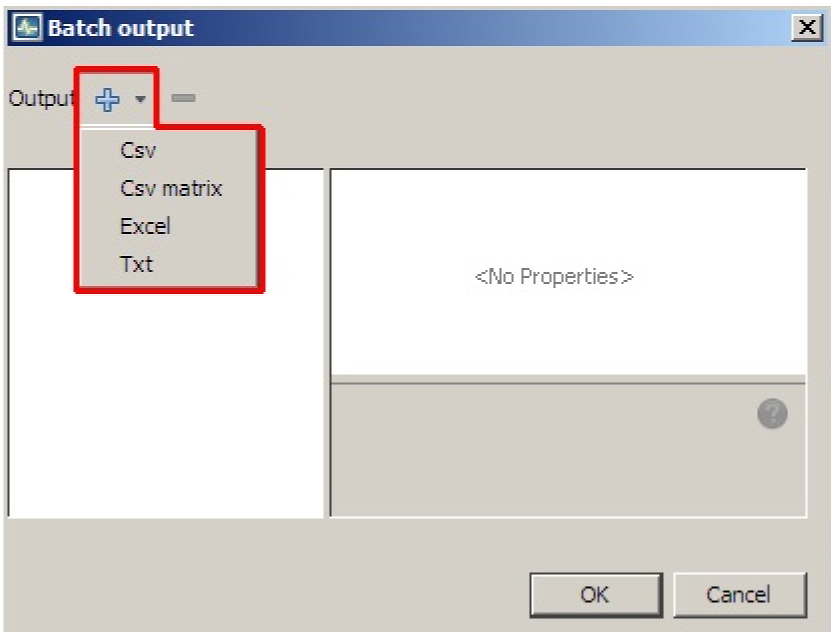


Figure 67: **Default output formats**

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

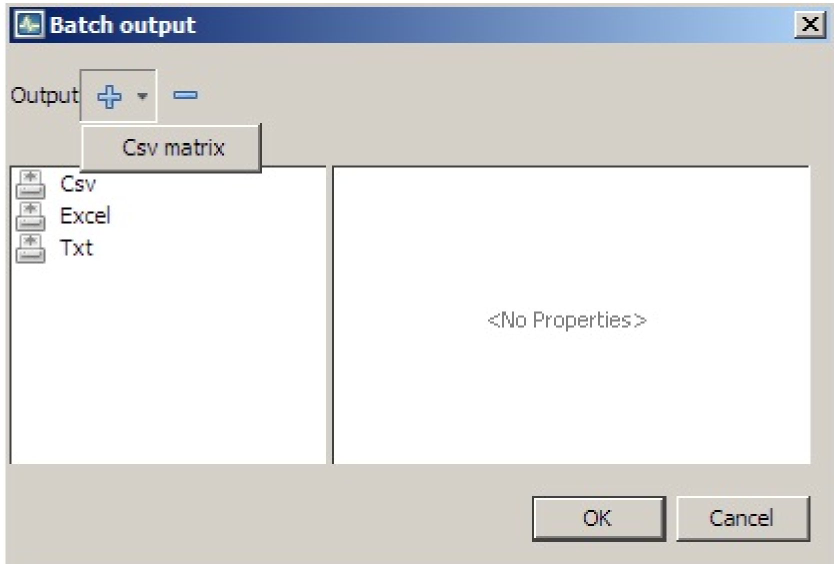


Figure 68: **Adding an output format to the list**

and *series* (series included in the file). These options are presented in the next points of this case study.

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).
7. With the option *File Prefix* the user can modify the default name of the output saved in the CSV file.
8. *Layout* controls how the output is divided into separate files. Expand the list to display available options:
   - *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.
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

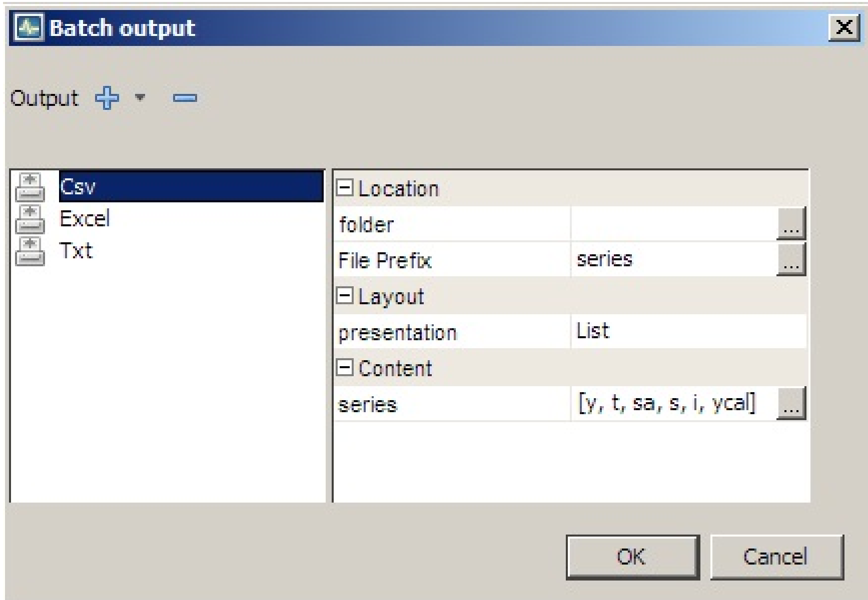


Figure 69: **Options for a *Csv* format**

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.

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 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.

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

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

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.

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.

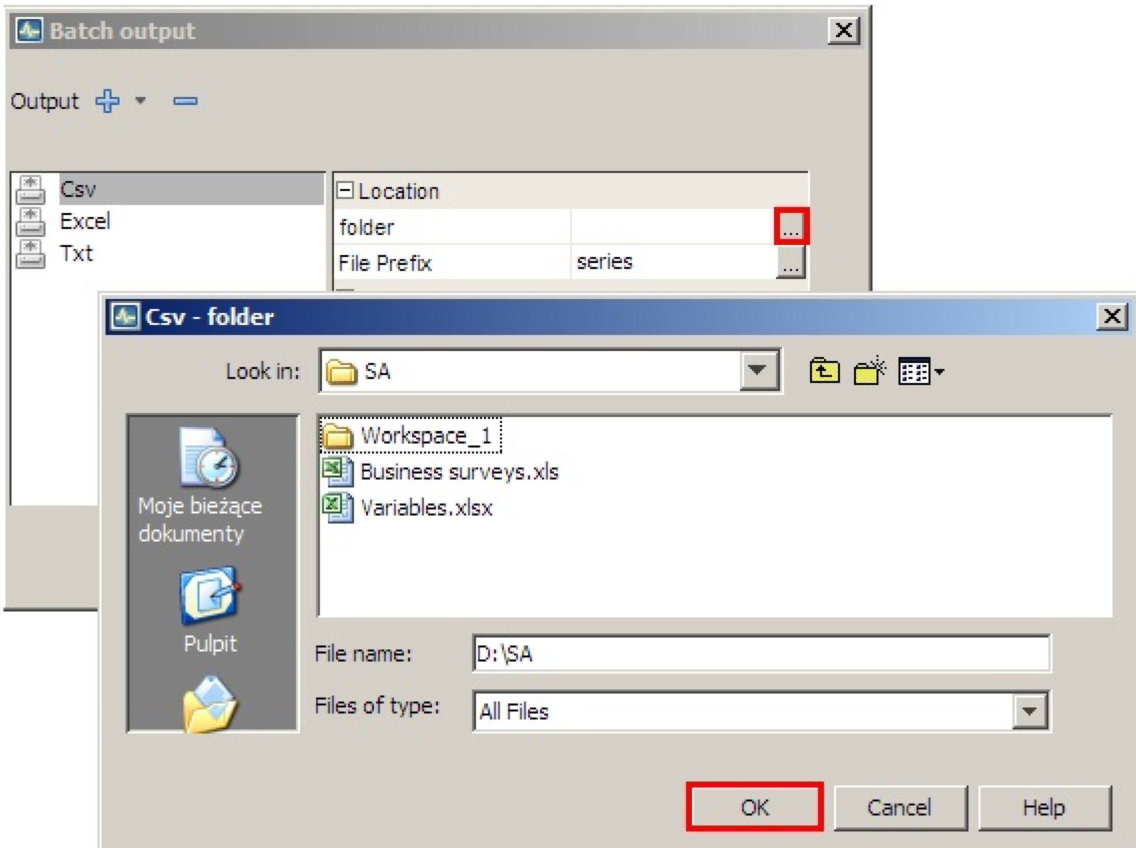


Figure 70: **Specifying a destination folder**

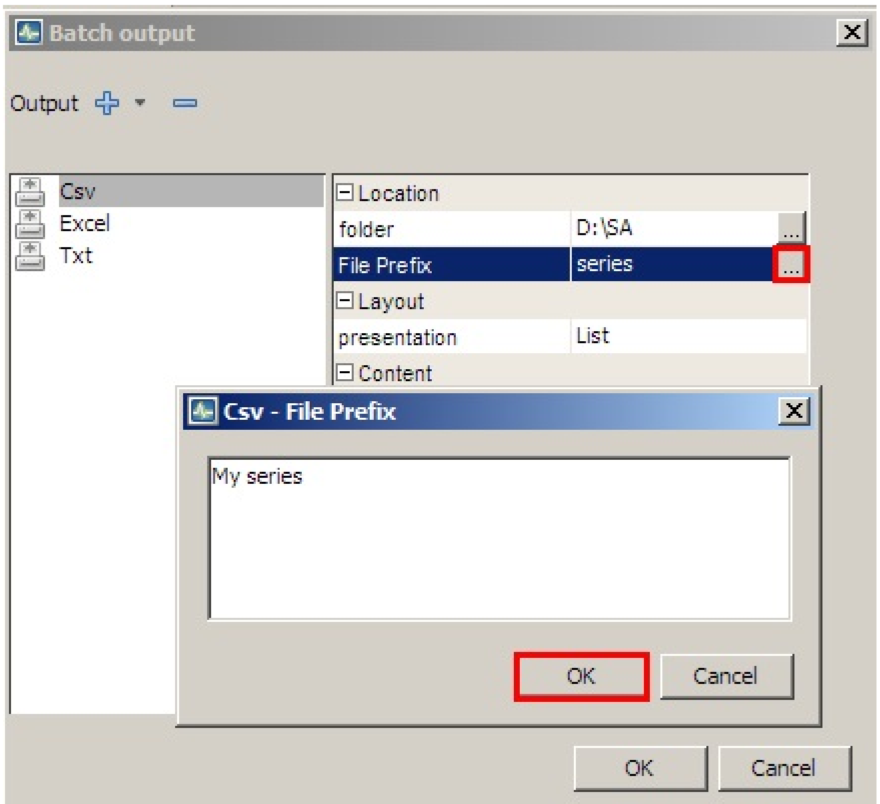


Figure 71: **Setting a *File Prefix* option**

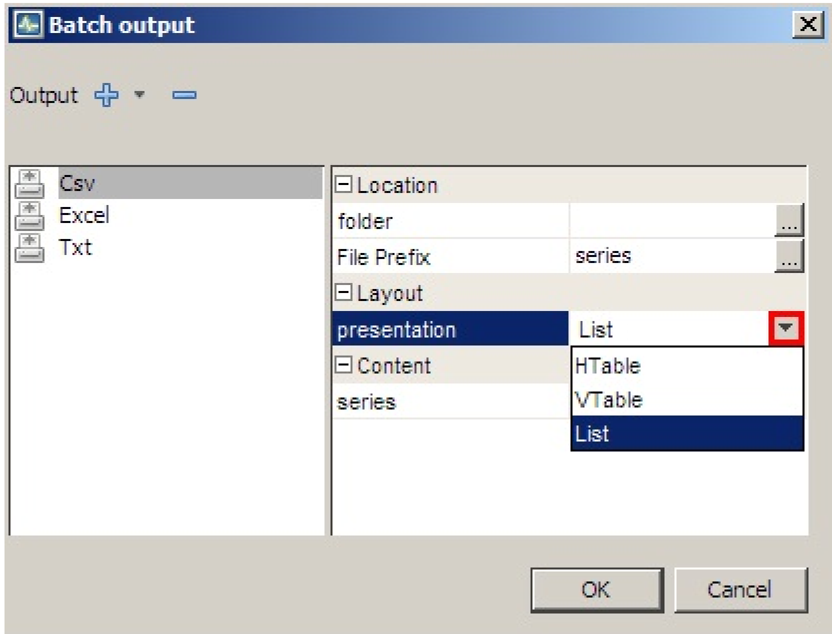


Figure 72: **Layout options for a *Csv* format**

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.
16. Once the output settings are selected, click the *OK* button.
17. For each output JDemetra+ provides information on the status of the operation. An example is presented below.

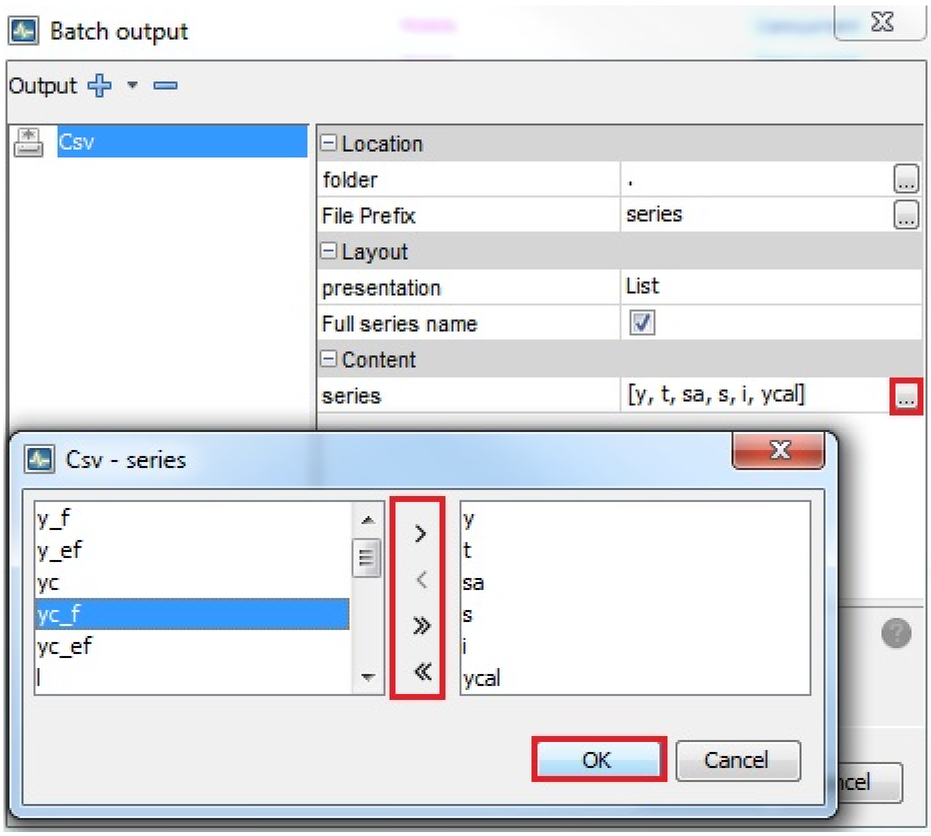


Figure 73: **Specifying a content of the output file**

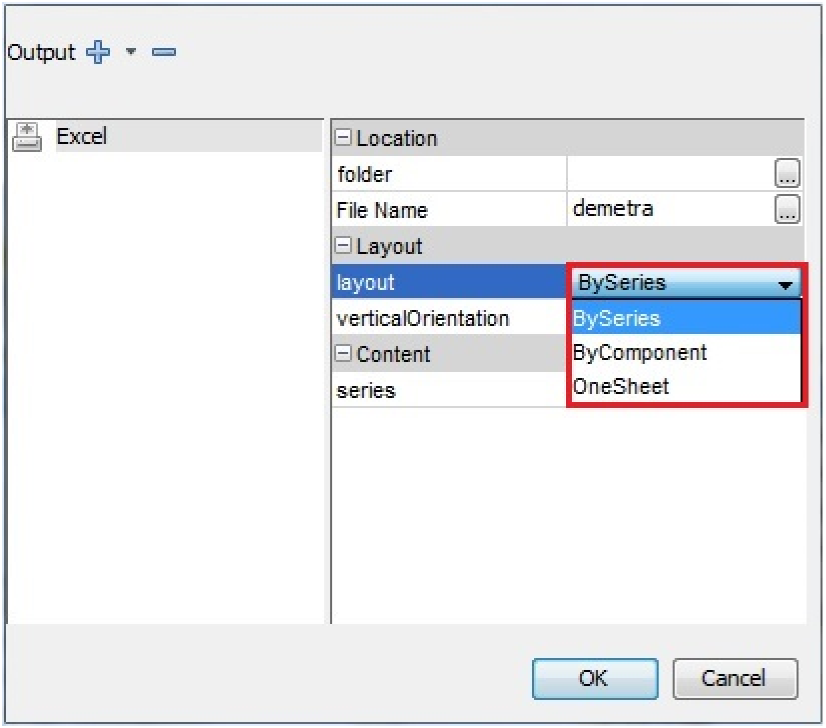


Figure 74: **Layout options for an *Excel* format**

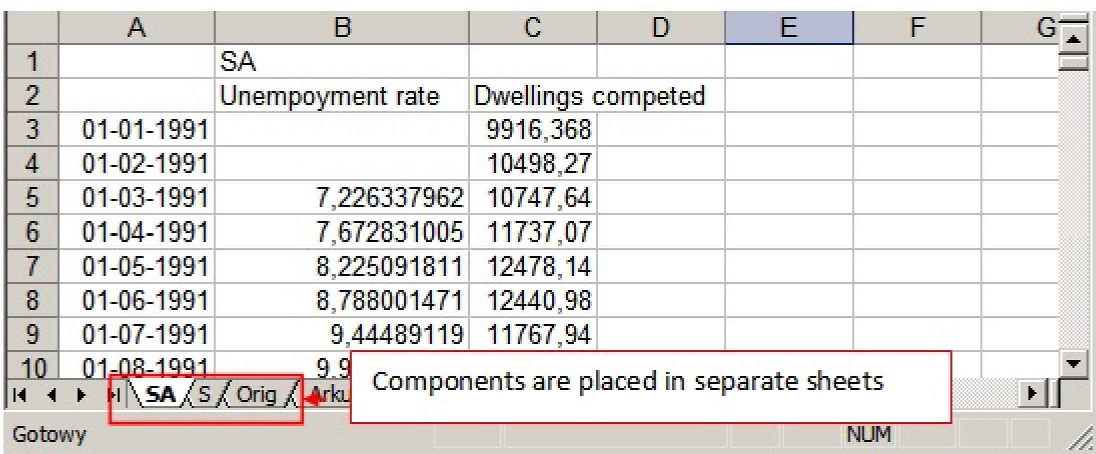
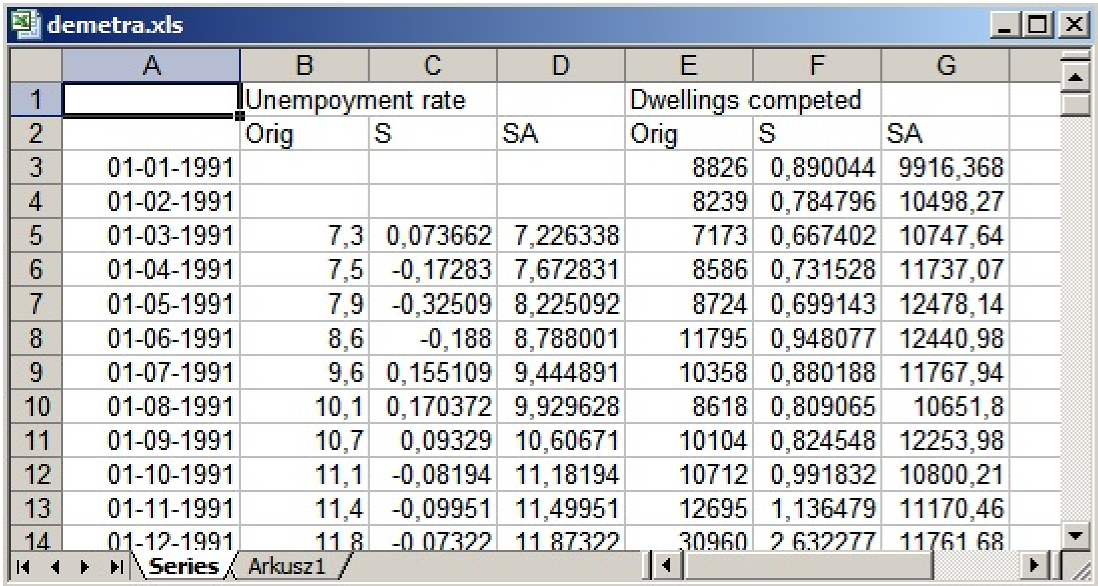


Figure 75: **An Excel file view for the *ByComponent* option**



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

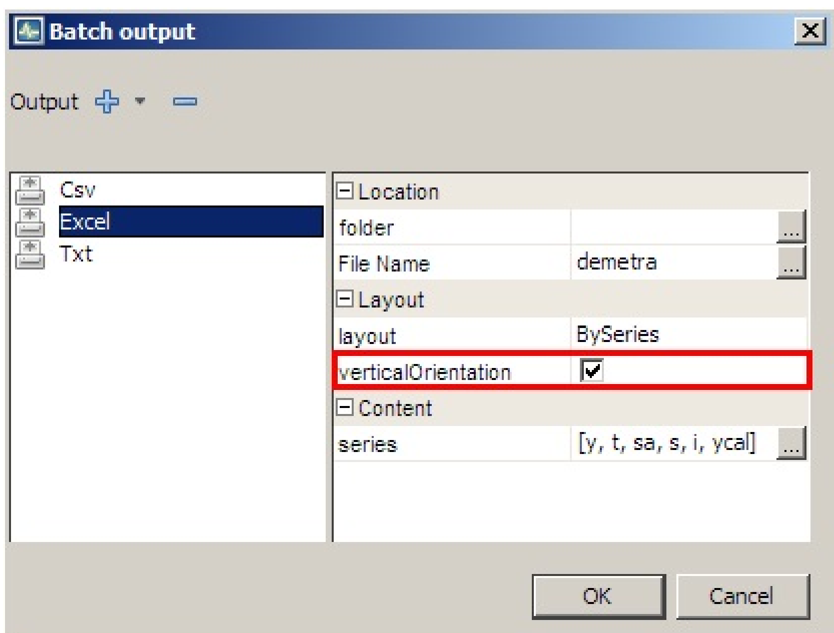


Figure 77: **The *VerticalOrientatio*n option**

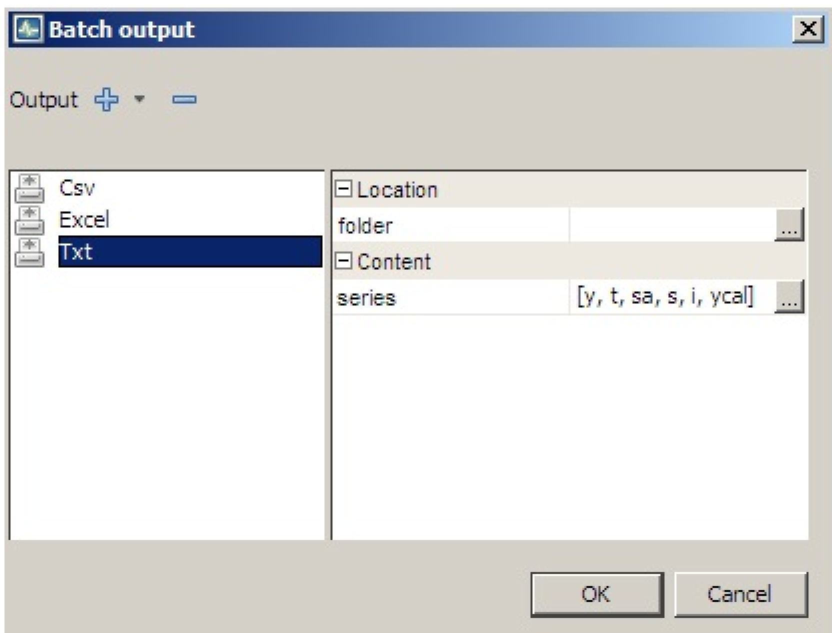


Figure 78: **Options for the *Txt* output**

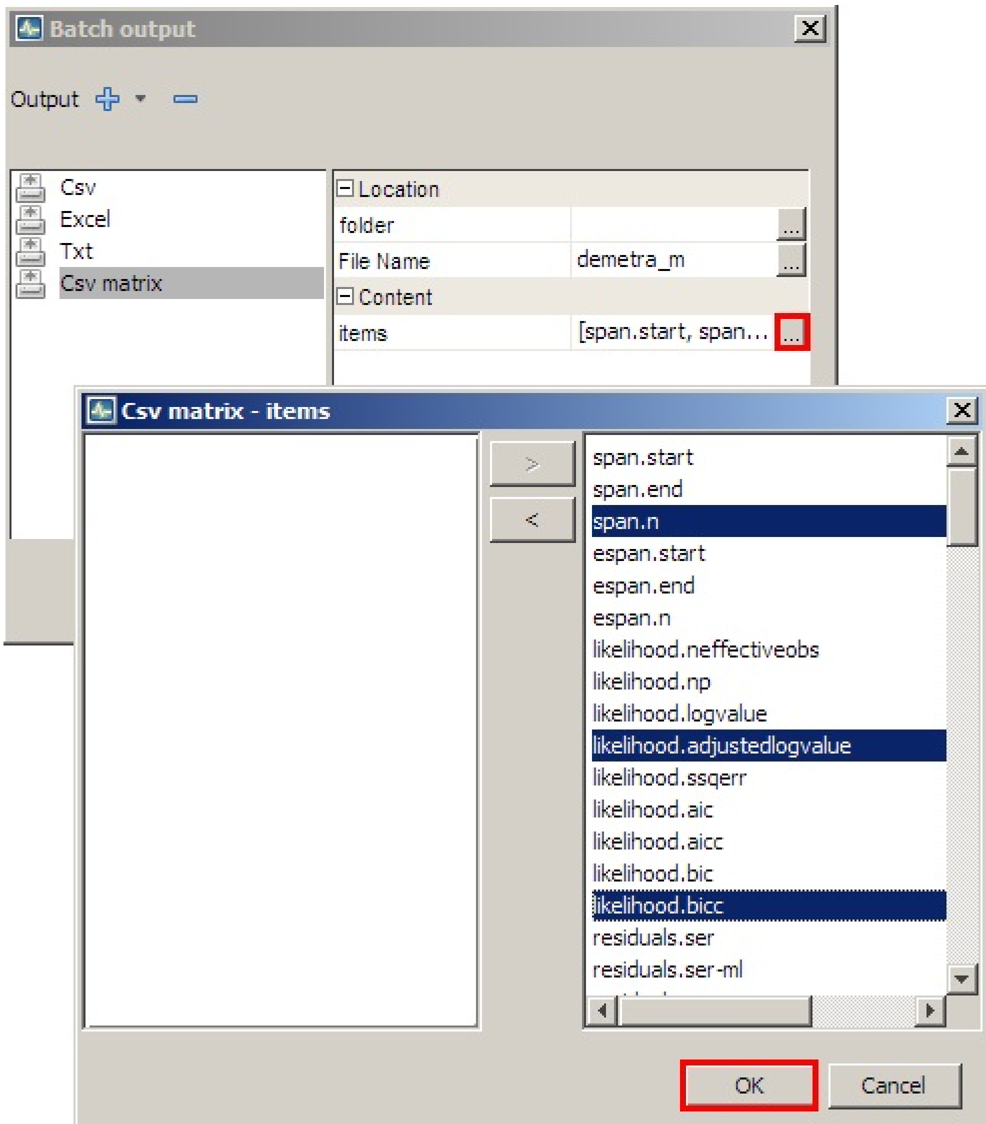


Figure 79: **List of items available for the *Csv matrix* output type**

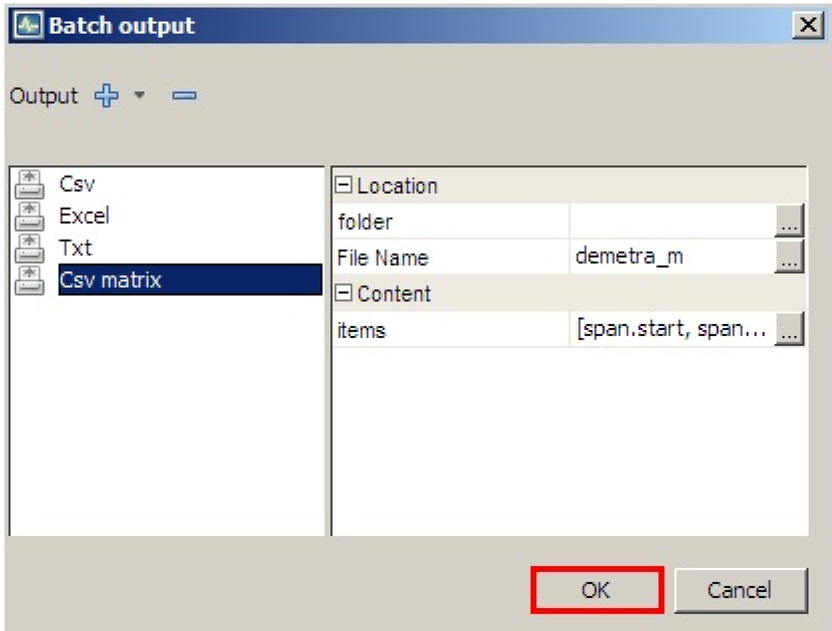


Figure 80: **Options for the *Csv matrix* output**

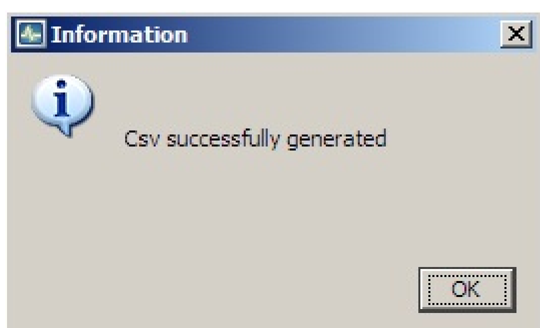


Figure 81: **Generating output - status information**

Using JDemetra+ in R

Overview

Core JDemetra Java algorithms can be accessed via several tools:

- Graphical User Interface GUI
- ...enhanced with additional plug-ins
- R packages

This chapter provides an overview of the packages linked to JDemetra core routines version 2.x and 3.x (under construction). More details on specific functions are available in the relevant chapters in the Algorithms part of this documentation. Help pages relative to each package in R are also very helpful.

Useful resources, including examples, are also available in this GitHub repository.

Scope of version 2

Packages corresponding to version 2.x core routines:

- **RJDemetra** on CRAN or https://github.com/jdemetra/rjdemetra
- **rjdworkspace** on https://github.com/InseeFrLab/rjdworkspace
- **JDCruncheR** on https://github.com/InseeFr/JDCruncheR
- **ggdemetra**
- **rjdqa**

Scope of version 3

Packages corresponding to version 3.x core routines (still under construction, moving perimeters) are listed

- **rjdemetra3** on https://github.com/palatej/rjdemetra3
- **rjd3toolkit** on https://github.com/palatej/rjd3toolkit
- **rjd3modelling** on https://github.com/palatej/rjd3modelling

- **rjd3sa** on https://github.com/palatej/rjd3sa
- **rjd3arima** on https://github.com/palatej/rjd3arima
- **rjd3x13** on https://github.com/palatej/rjd3x13
- **rjd3tramoseats** on https://github.com/palatej/rjd3tramoseats
- **rjd3sts** on https://github.com/palatej/rjd3sts
- **rjd3stl** on https://github.com/palatej/rjd3stl
- **rjd3highfreq** on https://github.com/palatej/rjd3highfreq
- **rjd3bench** on https://github.com/palatej/rjd3bench
- **ggdemetra3** on https://github.com/palatej/ggdemetra3

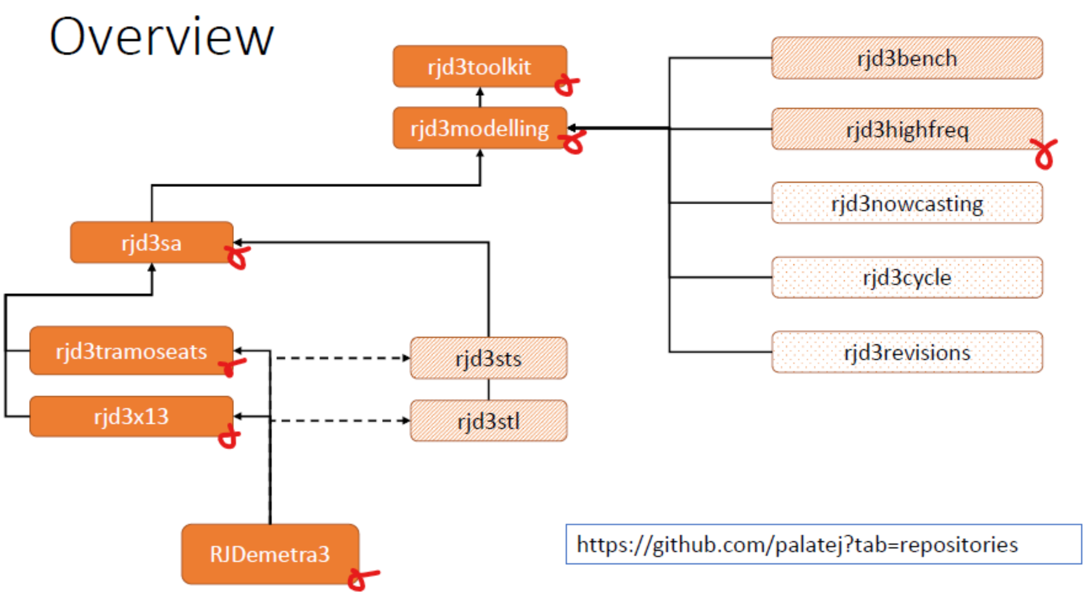


Figure 82: rjd3 overview

Algorithms available in R

Seasonal adjustment in version 2

Algorithm	Package	Comments
X13-Arima	RJDemetra	Reg-Arima and X-11 decomposition available independently
Tramo-Seats	RJDemetra	Tramo available independently

More details on functions parameters and retrieving output in the chapter dedicated to Seasonal Adjustment

Seasonal adjustment in version 3

Algorithm	Package	Comments
X13-Arima	rjd3x13	Reg-Arima and X-11 decomposition available independently
Extended X-11	rjd3highfreq	For high-frequency (intra-monthly) data
Tramo-Seats	rjd3tramoseats	Tramo available independently
Extended Tramo	rjd3highfreq	For high-frequency data
Extended Seats	rjd3highfreq	For high-frequency data
STL	rjd3stl	Including high-frequency data
Basic Structural Models	rjd3sts	States space framework

Running the cruncher and generating a quality report

JDemetra+ cruncher is an executable module designed for mass production of seasonally adjusted series .

Package	JD+ version	Comments
rjwsacruncher	2.x	estimation update and output
JDCruncheR	2.x	all the above + Quality Report

Wrangling JD+ workspaces

A workspace is a specific JDemetra+ data format (xml files) allowing to use the graphical user interface and the cruncher.

Package	JD+ version	Comments
rjdworkspace	2.x	update meta data, merge workspaces
rjdemetra3	3.x	under construction

Filtering and Trend estimation

Algorithm	Package	Comments
Moving average functions	rjdfilters	
Local Polynomial Trend Estimation	rjdfilters, rjd3highfreq	

Benchmarking and Temporal disaggregation

Algorithm	Package	Comment
Denton	rjd3bench	
Cholette	rjd3bench	
Cubic splines	rjd3bench	
Temporal Disaggregation	rjd3bench	

Generating additional output in SA estimation

This additional packages produce enhanced plots and diagnostic outputs.

Package	JD+ version	Comments
rjdmarkdown	2.x	enhanced print of diagnostics
ggdemetra	3.x	plots based on ggplot
ggdemetra3	3.x	plots based on ggplot
rjdqa	2.x	visual dashboard on one series

General structure of the packages

The R object resulting from an estimation is a list of lists containing raw data, parameters, output series and diagnostics.

Output structure for RJDemetra package

Organised by domain:

To retrieve any element just navigate this list of lists.

Output structure for rjd3x13 package

Results and specification are separated first and then organised by domain.

```
sa_x13_v3 <- RJDemetra::x13(y_raw, spec = "RSA5")
sa_x13_v3$result
sa_x13_v3$estimation_spec
sa_x13_v3$result_spec
```

```
SA
├─ regarima (≠ X-13 and TRAMO-SEAT)
│   ├─ specification
│   └─ ...
├─ decomposition (≠ X-13 and TRAMO-SEAT)
│   ├─ specification
│   └─ ...
├─ final
│   ├─ series
│   └─ forecasts
├─ diagnostics
│   ├─ variance_decomposition
│   ├─ combined_test
│   └─ ...
└─ user_defined
```

Figure 83: V2 SA structure

```r
sa_x13_v3$user_defined
```

To retrieve any element just navigate this list of lists.

Installation procedure

version 2

```r
install.packages("RJDemetra")
remotes::install_github("InseeFrLab/rjdworkspace")
remotes::install_github("InseeFr/JDCruncheR")
```

version 3

```r
#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")
```

```r
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")
```

The rjd3 suite of packages

The sections below provide an overview of the main functions by categories. For detailed description refer to each package's own R documentation pages.

Utility packages

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

```r
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

Input variables

This package allows creating input variables (regressors) for to be used in Reg-Arima modelling step:

- 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()`)
- outliers regressors (AO, LS, TC, SO, Ramp, intervention variables), calendar related regressors (stock, leap year, periodic dummies and contrasts, trigonometric variables)

Tests

- Range-mean regression test (to choose log transformation), Canova-Hansen (`td.ch()`) and trading-days f-test (`td.f()`)

Setting specification for Reg-Arima (Tramo) steps

Specification functions for `rjd3x13` and `rjd3tramoseats`: `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()`, `add_usrdefvar()`

rjd3sa package

Seasonality tests

- 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()`)

```
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

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

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` (upgraded from version 2)

rjd3highfreq

Seasonal adjustment of high frequency data:

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

rjd3stl

`rjd3stl` : STL, MSTL, ISTL, loess

Add-in packages

ggdemetra

Enhanced plots.

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 +
```

```r
    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

- 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 averages

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

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

```r
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)
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

```
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

Interface to structural time series and state space models

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

rjd3bench

Benchmarking and temporal disaggregation

Several examples here

Wrangling workspaces in R

Under construction

More here

Production, Cruncher and quality report

Under construction

Overview

The cruncher is an additional "executable" module. It can be launched via R or SAS for example.

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")
```

- checking the current value

```
getOption("cruncher_bin_directory")
```

Updating a workspace

The functions described in this section are identical for both packages.

Running estimations

The general context - First estimation

- Applying a revision policy to updated raw series

The function `cruncher_and_param()` allows to do that

```
cruncher_and_param(workspace = "D:/my_folder/my_ws.xml",
                   rename_multi_documents = FALSE,
                   policy = "lastoutliers" #name of the revision policy
                   log= my_log_file.txt)
```

To use the documentation, compute `help()` or `?function`:

```
?cruncher_and_param
help(cruncher_and_param)
```

Before running SA estimations, set the export options.

Configuring output options

After updating the workspace with the selected revision policies, the cruncher generates output - series (csv files) - diagnostics and parameters (demetra_m.csv file)

These files will be created in the workspace's repository, sub-repository 'Output"

```
path="My_Workspace/Output/SAProcessing"
```

Selecting time series to export

```r
# returns names of the currently exported series
getOption("default_tsmatrix_series")
# example of setting this option
options(default_tsmatrix_series = c("sa", "sa_f"))
# only seasonally adjsuted series ("sa") and its forecasts ("sa_f") will be exported
```

Selecting diagnostics and parameters to exoprt

```r
# returns names of the currently exported diagnostics and parameters
getOption("default_matrix_item")
# example of setting this option
options(default_matrix_item = c("likelihood.aic",
                                "likelihood.aicc",
                                "likelihood.bic",
                                "likelihood.bicc"))
```

Quality report with JDCruncheR

The JDCruncheR package also:

- computes a quality score
- creates a quality report from the diagnostics produced by JDemetra+

Main steps

The three main functions of the package are:

- `extract_QR` to extract the quality report from the csv file (demetra_m.csv) that contains all JD+ diagnostics;
- `compute_score` to compute a weighted score based on the diagnostics
- `export_xlsx` to export the quality report.

```r
# choose the demetra_m.csv file generated by the cruncher
QR <- extract_QR()
QR
```

```
?compute_score # to see how the score is calculated (formula)
QR <- compute_score(QR,
                    n_contrib_score = 3)

QR

QR <- sort(QR, decreasing = TRUE, sort_variables = "score")
export_xlsx(QR,
            file_name = "U:/quality_report.xls")
```

Piling up results for several SAP's

When working with several workspaces (or SAPs), quality reports can be piled up with the function `rbind()` or by creating a mQR_matrix object with the function `mQR_matrix()`

```
QR1 <- extract_QR()
QR2 <- extract_QR()
mQR <- mQR_matrix(QR1, QR2)
mQR
# naming each object
names(mQR) <- c("report_1", "report_2")
# Equivalent to:
mQR <- mQR_matrix(report_1 = QR1, report_2 = QR2)
mQR

# score calculation for all reports
mQR <- compute_score(mQR,
                    n_contrib_score = 3)
export_xlsx(mQR,
            export_dir = "U:/")
```

Conditionnal score

Missing values can be ignored and conditions can be set for indicators:

```
# oos_mse weight reduced to 1 when the other
# indicators are "Bad" ou "Severe"
condition1 <- list(indicator = "oos_mse",
                   conditions = c("residuals_independency",
```

```
                                   "residuals_homoskedasticity",
                                   "residuals_normality"),
                        conditions_modalities = c("Bad","Severe"))
BQ <- compute_score(BQ, n_contrib_score = 5,
                         conditional_indicator = list(condition1),
                         na.rm = TRUE)
```

Customize the score computation

Practical steps if you want to customize the score computation (see package documentation in R)

- select your indicators of interest
- adjust "good", "bad"…threshold in JD+ GUI if necessary
- by default good=0, uncertain=1, bad or severe=3
- change this grading system and/or the weights directly in the package functions
- rebuild your package

List of exportable diagnostics and parameters

```
options(default_matrix_item = c("period", "span.start", "span.end", "span.n", "span.missin
                                "espan.start", "espan.end", "espan.n", "log", "adjust", "r
                                "regression.ntd", "regression.nmh", "regression.td-derived
                                "regression.td-ftest", "regression.easter", "regression.no
                                "regression.noutao", "regression.noutls", "regression.nout
                                "regression.noutso", "regression.td(*):4", "regression.out
                                "regression.user(*)", "likelihood.neffectiveobs", "likelih
                                "likelihood.logvalue", "likelihood.adjustedlogvalue", "lik
                                "likelihood.aic", "likelihood.aicc", "likelihood.bic", "li
                                "residuals.ser", "residuals.ser-ml", "residuals.mean", "re
                                "residuals.kurtosis:3", "residuals.dh", "residuals.lb", "r
                                "residuals.seaslb", "residuals.bp", "residuals.bp2", "resi
                                "residuals.nudruns", "residuals.ludruns", "residuals.nruns
                                "residuals.lruns", "arima", "arima.mean", "arima.p", "arim
                                "arima.q", "arima.bp", "arima.bd", "arima.bq", "arima.phi(
                                "arima.bphi(*)", "arima.th(*)", "arima.bth(*)", "decomposi
                                "decomposition.parameters_cutoff", "decomposition.model_ch
```

```
"decomposition.tvar-estimator", "decomposition.tvar-estima
"decomposition.tvar-pvalue", "decomposition.savar-estimato
"decomposition.savar-estimate", "decomposition.savar-pvalu
"decomposition.svar-estimator", "decomposition.svar-estima
"decomposition.svar-pvalue", "decomposition.ivar-estimator
"decomposition.ivar-estimate", "decomposition.ivar-pvalue"
"decomposition.tscorr-estimate", "decomposition.tscorr-pva
"decomposition.ticorr-estimator", "decomposition.ticorr-es
"decomposition.ticorr-pvalue", "decomposition.sicorr-estim
"decomposition.sicorr-estimate", "decomposition.sicorr-pva
"decomposition.ar_root(*)", "decomposition.ma_root(*)", "m
"variancedecomposition.cycle", "variancedecomposition.seas
"variancedecomposition.irregular", "variancedecomposition.
"variancedecomposition.others", "variancedecomposition.tot
"diagnostics.logstat", "diagnostics.levelstat", "diagnosti
"diagnostics.fcast-outsample-mean", "diagnostics.fcast-out
"diagnostics.seas-lin-f", "diagnostics.seas-lin-qs", "diag
"diagnostics.seas-lin-friedman", "diagnostics.seas-lin-per
"diagnostics.seas-lin-spectralpeaks", "diagnostics.seas-si
"diagnostics.seas-si-evolutive", "diagnostics.seas-si-stab
"diagnostics.seas-res-f", "diagnostics.seas-res-qs", "diag
"diagnostics.seas-res-friedman", "diagnostics.seas-res-per
"diagnostics.seas-res-spectralpeaks", "diagnostics.seas-re
"diagnostics.seas-res-combined3", "diagnostics.seas-res-ev
"diagnostics.seas-res-stable", "diagnostics.seas-i-f", "di
"diagnostics.seas-i-kw", "diagnostics.seas-i-periodogram",
"diagnostics.seas-i-combined", "diagnostics.seas-i-combine
"diagnostics.seas-i-evolutive", "diagnostics.seas-i-stable
"diagnostics.seas-sa-f", "diagnostics.seas-sa-qs", "diagno
"diagnostics.seas-sa-friedman", "diagnostics.seas-sa-perio
"diagnostics.seas-sa-spectralpeaks", "diagnostics.seas-sa-
"diagnostics.seas-sa-combined3", "diagnostics.seas-sa-evol
"diagnostics.seas-sa-stable", "diagnostics.seas-sa-ac1", "
"diagnostics.td-sa-last", "diagnostics.td-i-all", "diagnos
"diagnostics.td-res-all", "diagnostics.td-res-last", "diag
"diagnostics.ic-ratio", "diagnostics.msr-global", "diagnos
"decomposition.trendfilter", "decomposition.seasfilter", "
"m-statistics.m2", "m-statistics.m3", "m-statistics.m4", "
"m-statistics.m6", "m-statistics.m7", "m-statistics.m8", "
"m-statistics.m10", "m-statistics.m11", "m-statistics.q",
"diagnostics.basic checks.definition:2", "diagnostics.basi
```

```
                                      "diagnostics.visual spectral analysis.spectral seas peaks"
                                      "diagnostics.regarima residuals.normality:2", "diagnostics
                                      "diagnostics.regarima residuals.spectral td peaks:2", "dia
                                      "diagnostics.outliers.number of outliers:2", "diagnostics.
                                      "diagnostics.out-of-sample.mse:2", "diagnostics.m-statisti
                                      "diagnostics.m-statistics.q-m2:2", "diagnostics.seats.seas
                                      "diagnostics.seats.irregular variance:2", "diagnostics.sea
                                      "diagnostics.residual seasonality tests.qs test on sa:2",
                                      "diagnostics.residual seasonality tests.f-test on sa (seas
                                      "diagnostics.residual seasonality tests.f-test on i (seaso
                                      "diagnostics.combined seasonality test.combined seasonalit
                                      "diagnostics.combined seasonality test.combined seasonalit
                                      "diagnostics.combined seasonality test.combined seasonalit
                                      "diagnostics.residual trading days tests.f-test on sa (td)
                                      "diagnostics.residual trading days tests.f-test on i (td):
                                      "diagnostics.quality"
))
```

Production and revision policies

Under construction.

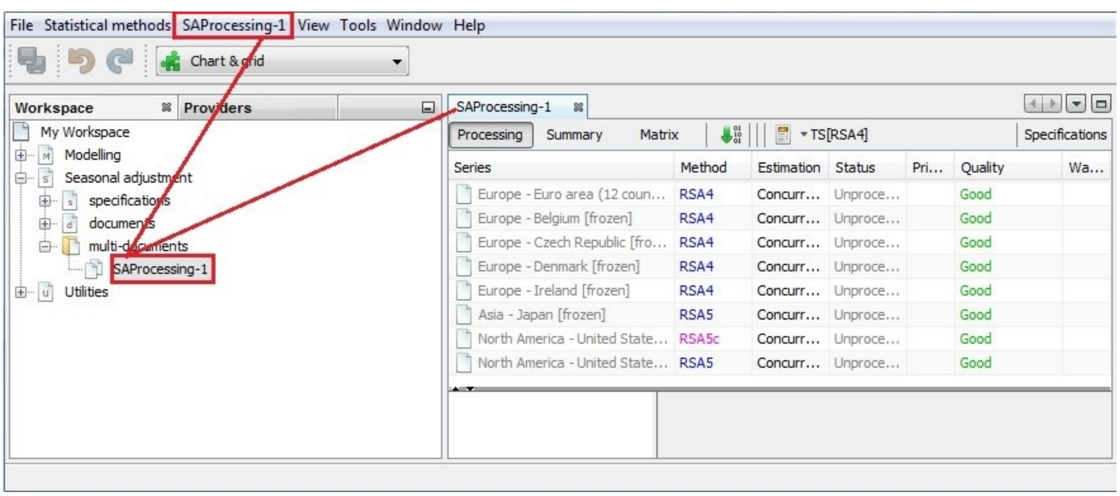
Context of use

Raw data has been modified (extended and/or revised) and the previous SA estimation needs updating, but with keeping certain parameters fixed. The set of constraints on the parameters is called "revision policy" or "refresh policy".

Revision policies in GUI

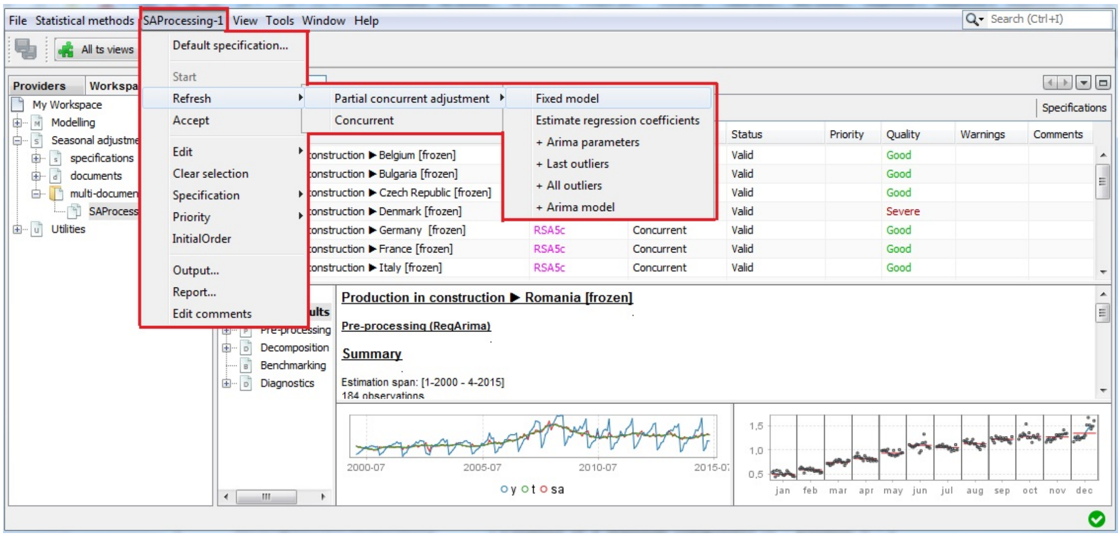
The saved results from a seasonal adjustment processing can be refreshed when new or modified observations are available.

1. To refresh the results open a previously saved workspace using the path *File* → *Open Workspace*. Choose the multi-document option from the *Workspace* window and double click on it to display the multi-document menu (*SAProcessing*).



Opening a multi-document

2. Several refreshment options are available.



The *Refresh* menu

Concurrent

According to the ESS Guidelines on Seasonal Adjustment (2015) , concurrent adjustment means that the model, filters, outliers, regression parameters and transformation type are all re-identified and the respective parameters and factors re-estimated every time new observations are available. This option in JDemetra+ means that a completely new model is identified, and the previous results are not taken into account.

The picture below presents the initial model (on the left) and the results of the refreshment procedure with the *Concurrent adjustment* option (on the right). The transformation type has changed from none to log. The ARIMA model has been re-identified (it has changed from (0,1,1)(1,1,0) to (1,1,0)(0,1,1)). In contrast to the initial model, in the updated model trading day effects and a leap year effect are no longer included. Also the automatically identified outliers are not the same in both models.

Summary

Estimation span: [7-1996 - 12-2016]
246 observations
Trading days effects (7 variables)
Easter [8] detected
5 detected outliers

Final model

Likelihood statistics
Number of effective observations = 233
Number of estimated parameters = 16

Loglikelihood = -559.6616717869163
Standard error of the regression (ML estimate) = 2.669031738674505
AIC = 1151.3233435738325
AICC = 1153.841862092351
BIC (corrected for length) = 2.314356746231504

Scores at the solution
-0,000004 -0,000414

Arima model
[(0,1,1)(1,1,0)]

	Coefficients	T-Stat	P[|T| > t]
Theta(1)	-0,4902	-8,33	0,0000
BPhi(1)	0,1680	2,45	0,0152

Correlation of the estimates

	Theta(1)	BPhi(1)
Theta(1)	1,0000	0,0784
BPhi(1)	0,0784	1,0000

Regression model
Trading days

	Coefficients	T-Stat	P[|T| > t]
Monday	-0,5102	-1,92	0,0562
Tuesday	0,2288	0,88	0,3774
Wednesday	0,1073	0,40	0,6905
Thursday	0,2028	0,75	0,4536
Friday	0,9280	3,48	0,0006
Saturday	-0,3434	-1,27	0,2071
Sunday (derived)	-0,6134	-2,28	0,0235

Joint F-Test = 8,13 (0,0000)

Leap year

	Coefficients	T-Stat	P[|T| > t]
	2,6712	3,07	0,0024

Easter [8]

	Coefficients	T-Stat	P[|T| > t]
	1,6759	3,08	0,0023

Outliers

	Coefficients	T-Stat	P[|T| > t]
AO (4-2004)	19,8713	10,12	0,0000
LS (1-2001)	-8,5643	-4,62	0,0000
AO (4-2010)	-8,7157	-4,62	0,0000
AO (3-2004)	8,6114	4,40	0,0000
AO (12-2003)	7,2918	3,90	0,0001

Summary

Estimation span: [1-2005 - 12-2016]
144 observations
Series has been log-transformed
No trading days effects
Easter [1] detected
1 detected outlier

Final model

Likelihood statistics
Number of effective observations = 131
Number of estimated parameters = 5

Loglikelihood = 158.22155489483504
Transformation adjustment = -648.184301988354
Adjusted loglikelihood = -489.962747093519

Standard error of the regression (ML estimate) = 0.06861665286654098
AIC = 989.925494187038
AICC = 990.405494187038
BIC (corrected for length) = -5.2095790541390326

Scores at the solution
0,002923 -0,004642

Arima model
[(1,1,0)(0,1,1)]

	Coefficients	T-Stat	P[|T| > t]
Phi(1)	0,4240	5,25	0,0000
BTheta(1)	-0,8247	-13,50	0,0000

Correlation of the estimates

	Phi(1)	BTheta(1)
Phi(1)	1,0000	0,0857
BTheta(1)	0,0857	1,0000

Regression model
Easter [1]

	Coefficients	T-Stat	P[|T| > t]
	-0,0398	-1,94	0,0543

Outliers

	Coefficients	T-Stat	P[|T| > t]
TC (1-2011)	-0,4462	-7,36	0,0000

The *Concurrent adjustment* revision policy results

Partial concurrent adjustment → Fixed model

The *Partial concurrent adjustment* → *Fixed model* strategy means that the ARIMA model, outliers and other regression parameters are not re-identified and the values of the parameters are fixed. In particular, no new outliers or calendar variables are added to the model as well as no changes neither in the calendar variables nor in the outliers' types are allowed. The transformation type remains unchanged.

The picture below presents the initial model (on the left) and the results of the refreshment procedure with the *Partial concurrent adjustment* → *Fixed model* option (on the right). The parameters of the ARIMA part are not estimated and their values are the same as before. The trading days and outliers are fixed too and no new regression effects are identified.

Summary

Estimation span: [7-1996 - 12-2016]
246 observations
Trading days effects (7 variables)
Easter [8] detected
5 detected outliers

Final model

Likelihood statistics
Number of effective observations = 233
Number of estimated parameters = 16

Loglikelihood = -559.6616717869163
Standard error of the regression (ML estimate) = 2.669031738674505
AIC = 1151.3233435738325
AICC = 1153.841862092351
BIC (corrected for length) = 2.314356746231504

Scores at the solution
-0,000004 -0,000414

Arima model
[(0,1,1)(1,1,0)]

	Coefficients	T-Stat	P[|T| > t]
Theta(1)	-0,4902	-8,33	0,0000
BPhi(1)	0,1680	2,45	0,0152

Correlation of the estimates

	Theta(1)	BPhi(1)
Theta(1)	1,0000	0,0784
BPhi(1)	0,0784	1,0000

Regression model
Trading days

	Coefficients	T-Stat	P[|T| > t]
Monday	-0,5102	-1,92	0,0562
Tuesday	0,2288	0,88	0,3774
Wednesday	0,1073	0,40	0,6905
Thursday	0,2028	0,75	0,4536
Friday	0,9280	3,48	0,0006
Saturday	-0,3434	-1,27	0,2071
Sunday (derived)	-0,6134	-2,28	0,0235

Joint F-Test = 8,13 (0,0000)

Summary

Estimation span: [7-1996 - 7-2017]
253 observations
Fixed Trading days effects (7 variables)
Fixed Easter [8] effect
5 fixed outliers

Final model

Likelihood statistics
Number of effective observations = 240
Number of estimated parameters = 1

Loglikelihood = -692.4385696741856
Standard error of the regression (ML estimate) = 4.327256562481812
AIC = 1386.8771393483712
AICC = 1386.8939460710603
BIC (corrected for length) = 2.9298675057422012

Arima model
[(0,1,1)(1,1,0)]

	Coefficients	T-Stat	P[|T| > t]
Theta(1)	-0,4902		
BPhi(1)	0,1680		

Regression model
Fixed calendar effects

	Coefficients
Monday	-0,5102
Tuesday	0,2288
Wednesday	0,1073
Thursday	0,2028
Friday	0,9280
Saturday	-0,3434
Leap year	2,6712

Fixed moving holidays effects

	Coefficients
Easter [8]	1,6759

Fixed outliers

	Coefficients
AO (4-2004)	19,8713
LS (1-2001)	-8,5643
AO (4-2010)	-8,7157
AO (3-2004)	8,6114
AO (12-2003)	7,2918

A *Partial concurrent adjustment → fixed model* revision policy results

Partial concurrent adjustment → Estimate regression coefficients

The *Partial current adjustment → Estimate regression coefficients* option means that the ARIMA model, outliers and other regression parameters are not re-identified. The coefficients of the ARIMA model are fixed, other coefficients are re-estimated. In particular, no new outliers or calendar variables are added to the model as well as no changes neither in the calendar variables nor in the outliers' types are allowed. The transformation type remains unchanged.

The picture below presents the initial model (on the left) and the results of the refreshment procedure with the *Partial concurrent adjustment → Estimate regression coefficients* option (on the right). The number of estimated parameters is 16 in the initial model and 14 in the revised model (the parameters of the ARIMA model are not estimated.

Summary

Estimation span: [7-1996 - 12-2016]
246 observations
Trading days effects (7 variables)
Easter [8] detected
5 detected outliers

Final model

Likelihood statistics
Number of effective observations = 233
Number of estimated parameters = 16

Loglikelihood = -559.6616717869163
Standard error of the regression (ML estimate) = 2.669031738674505
AIC = 1151.3233435738325
AICC = 1153.841862092351
BIC (corrected for length) = 2.314356746231504

Scores at the solution
-0,000004 -0,000414 .

Arima model
[(0,1,1)(1,1,0)].

	Coefficients	T-Stat	P[|T| > t]
Theta(1)	-0,4902	-8,33	0,0000
BPhi(1)	0,1680	2,45	0,0152

Correlation of the estimates

	Theta(1)	BPhi(1)
Theta(1)	1,0000	0,0784
BPhi(1)	0,0784	1,0000

Regression model
Trading days

	Coefficients	T-Stat	P[|T| > t]
Monday	-0,5102	-1,92	0,0562
Tuesday	0,2288	0,88	0,3774
Wednesday	0,1073	0,40	0,6905
Thursday	0,2028	0,75	0,4536
Friday	0,9280	3,48	0,0006
Saturday	-0,3434	-1,27	0,2071
Sunday (derived)	-0,6134	-2,28	0,0235

Joint F-Test = 8,13 (0,0000)

Leap year

Coefficients	T-Stat	P[|T| > t]
2,6712	3,07	0,0024

Easter [8]

Coefficients	T-Stat	P[|T| > t]
1,6759	3,08	0,0023

Outliers

	Coefficients	T-Stat	P[|T| > t]
AO (4-2004)	19,8713	10,12	0,0000
LS (1-2001)	-8,5643	-4,62	0,0000
AO (4-2010)	-8,7157	-4,62	0,0000
AO (3-2004)	8,6114	4,40	0,0000
AO (12-2003)	7,2918	3,90	0,0001

Summary

Estimation span: [7-1996 - 7-2017]
253 observations
Trading days effects (7 variables)
Easter [8] detected
5 pre-specified outliers

Final model

Likelihood statistics
Number of effective observations = 240
Number of estimated parameters = 14

Loglikelihood = -665.9671390063976
Standard error of the regression (ML estimate) = 3.875350557203808
AIC = 1359.9342780127952
AICC = 1361.8009446794617
BIC (corrected for length) = 3.0061401918583264

Arima model
[(0,1,1)(1,1,0)].

	Coefficients	T-Stat	P[|T| > t]
Theta(1)	-0,4902		
BPhi(1)	0,1680		

Regression model
Trading days

	Coefficients	T-Stat	P[|T| > t]
Monday	-0,5065	-1,34	0,1819
Tuesday	0,3467	0,94	0,3463
Wednesday	0,1225	0,32	0,7479
Thursday	0,2916	0,75	0,4525
Friday	0,7808	2,07	0,0393
Saturday	-0,6722	-1,74	0,0840
Sunday (derived)	-0,3629	-0,95	0,3437

Joint F-Test = 3,52 (0,0024)

Leap year

Coefficients	T-Stat	P[|T| > t]
1,4721	1,23	0,2218

Easter [8]

Coefficients	T-Stat	P[|T| > t]
1,5126	2,01	0,0451

Prespecified outliers

	Coefficients	T-Stat	P[|T| > t]
AO (4-2004)	38,7128	13,68	0,0000
LS (1-2001)	-8,7841	-3,28	0,0012
AO (4-2010)	-8,8868	-3,27	0,0012
AO (3-2004)	8,0340	2,85	0,0048
AO (12-2003)	6,9950	2,59	0,0101

The *Partial concurrent adjustment→ Estimate regression coefficients* revision policy results

Partial concurrent adjustment → Estimate regression coefficients + Arima parameters

The *Partial concurrent adjustment → Estimate regression coefficients + Arima parameters* strategy means that the ARIMA model, outliers and other regression parameters are not re-identified. All parameters of the Reg-ARIMA model are re-estimated but the explanatory variables remain the same. The transformation type remains unchanged.

The picture below presents the initial model (on the left) and the results of the refreshment procedure with the *Partial concurrent adjustment → Estimate regression coefficient + Arima parameters* option (on the right). The parameters of the ARIMA part have been re-estimated and their values have been updated. Also regression coefficients have been re-estimated and the number of estimated coefficients in the revised model is the same as in the initial model (i.e. 16 estimated coefficients). The structure of the model remains unchanged while all coefficients have been updated.

Summary

Estimation span: [7-1996 - 12-2016]
246 observations
Trading days effects (7 variables)
Easter [8] detected
5 detected outliers

Final model

Likelihood statistics
Number of effective observations = 233
Number of estimated parameters = 16

Loglikelihood = -559.6616717869163
Standard error of the regression (ML estimate) = 2.669031738674505
AIC = 1151.3233435738325
AICC = 1153.841862092351
BIC (corrected for length) = 2.314356746231504

Scores at the solution
-0,000004 -0,000414

Arima model
[(0,1,1)(1,1,0)]

	Coefficients	T-Stat	P[|T| > t]
Theta(1)	-0,4902	-8,33	0,0000
BPhi(1)	0,1680	2,45	0,0152

Correlation of the estimates

	Theta(1)	BPhi(1)
Theta(1)	1,0000	0,0784
BPhi(1)	0,0784	1,0000

Regression model
Trading days

	Coefficients	T-Stat	P[|T| > t]
Monday	-0,5102	-1,92	0,0562
Tuesday	0,2288	0,88	0,3774
Wednesday	0,1073	0,40	0,6905
Thursday	0,2028	0,75	0,4536
Friday	0,9280	3,48	0,0006
Saturday	-0,3434	-1,27	0,2071
Sunday (derived)	-0,6134	-2,28	0,0235

Joint F-Test = 8,13 (0,0000)

Summary

Estimation span: [7-1996 - 7-2017]
253 observations
Trading days effects (7 variables)
Easter [8] detected
5 pre-specified outliers

Final model

Likelihood statistics
Number of effective observations = 240
Number of estimated parameters = 16

Loglikelihood = -653.0614639550819
Standard error of the regression (ML estimate) = 3.668082944975884
AIC = 1338.1229279101638
AICC = 1340.5623897935718
BIC (corrected for length) = 2.9418782672541677

Scores at the solution
-0,000018 -0,000041

Arima model
[(0,1,1)(1,1,0)]

	Coefficients	T-Stat	P[|T| > t]
Theta(1)	-0,2036	-3,12	0,0020
BPhi(1)	0,3022	3,48	0,0006

Correlation of the estimates

	Theta(1)	BPhi(1)
Theta(1)	1,0000	0,0334
BPhi(1)	0,0334	1,0000

Regression model
Trading days

	Coefficients	T-Stat	P[|T| > t]
Monday	-0,5098	-1,54	0,1252
Tuesday	0,4101	1,28	0,2024
Wednesday	0,1146	0,34	0,7327
Thursday	0,2243	0,65	0,5135
Friday	0,7773	2,32	0,0210
Saturday	-0,5492	-1,61	0,1078
Sunday (derived)	-0,4672	-1,38	0,1676

Joint F-Test = 5,80 (0,0000)

The *Partial concurrent adjustment → Estimate regression coefficients + Arima parameters* revision policy results

Partial concurrent adjustment → Estimate regression coefficients + outliers

The *Partial concurrent adjustment → Estimate regression coefficients + outliers* option means that the ARIMA model and regression parameters, except outliers, are not re-identified. The parameters of these variables, however, are re-estimated. All outliers are re-identified, i.e. the previous outcome of the outlier detection procedure is not taken into account and all outliers are identified and estimated once again. The transformation type remains unchanged.

The picture below presents the initial model (on the left) and the results of the refreshment procedure with the *Partial concurrent adjustment → Estimate regression coefficients + outliers* option (on the right). The parameters of the ARIMA part have been re-estimated and their values have been updated. Also regression coefficients for the calendar variables have been re-estimated. In the revised model there is no *Prespecified outliers* section. Instead, the outliers were re-identified.

Summary

Estimation span: [7-1996 - 12-2016]
246 observations
Trading days effects (7 variables)
Easter [8] detected
5 detected outliers

Final model

Likelihood statistics
Number of effective observations = 233
Number of estimated parameters = 16

Loglikelihood = -559.6616717869163
Standard error of the regression (ML estimate) = 2.669031738674505
AIC = 1151.3233435738325
AICC = 1153.841862092351
BIC (corrected for length) = 2.314356746231504

Scores at the solution
-0,000004 -0,000414

Arima model
[(0,1,1)(1,1,0)].

	Coefficients	T-Stat	P[|T| > t]
Theta(1)	-0,4902	-8,33	0,0000
BPhi(1)	0,1680	2,45	0,0152

Correlation of the estimates

	Theta(1)	BPhi(1)
Theta(1)	1,0000	0,0784
BPhi(1)	0,0784	1,0000

Regression model
Trading days

	Coefficients	T-Stat	P[|T| > t]
Monday	-0,5102	-1,92	0,0562
Tuesday	0,2288	0,88	0,3774
Wednesday	0,1073	0,40	0,6905
Thursday	0,2028	0,75	0,4536
Friday	0,9280	3,48	0,0006
Saturday	-0,3434	-1,27	0,2071
Sunday (derived)	-0,6134	-2,28	0,0235

Joint F-Test = 8,13 (0,0000)

Leap year

Coefficients	T-Stat	P[|T| > t]
2,6712	3,07	0,0024

Easter [8]

Coefficients	T-Stat	P[|T| > t]
1,6759	3,08	0,0023

Outliers

	Coefficients	T-Stat	P[|T| > t]
AO (4-2004)	19,8713	10,12	0,0000
LS (1-2001)	-8,5643	-4,62	0,0000
AO (4-2010)	-8,7157	-4,62	0,0000
AO (3-2004)	8,6114	4,40	0,0000
AO (12-2003)	7,2918	3,90	0,0001

Summary

Estimation span: [7-1996 - 7-2017]
253 observations
Trading days effects (7 variables)
Easter [8] detected
6 detected outliers

Final model

Likelihood statistics
Number of effective observations = 240
Number of estimated parameters = 17

Loglikelihood = -573.4952681957561
Standard error of the regression (ML estimate) = 2.6361922842978505
AIC = 1180.9905363915123
AICC = 1183.747293148269
BIC (corrected for length) = 2.3040470471520735

Scores at the solution
-0,000185 -0,000728

Arima model
[(0,1,1)(1,1,0)].

	Coefficients	T-Stat	P[|T| > t]
Theta(1)	-0,4908	-8,49	0,0000
BPhi(1)	0,1679	2,53	0,0120

Correlation of the estimates

	Theta(1)	BPhi(1)
Theta(1)	1,0000	0,0615
BPhi(1)	0,0615	1,0000

Regression model
Trading days

	Coefficients	T-Stat	P[|T| > t]
Monday	-0,5212	-2,01	0,0454
Tuesday	0,2349	0,93	0,3516
Wednesday	0,1042	0,40	0,6899
Thursday	0,2102	0,79	0,4294
Friday	0,9059	3,51	0,0005
Saturday	-0,3332	-1,25	0,2117
Sunday (derived)	-0,6007	-2,29	0,0230

Joint F-Test = 8,39 (0,0000)

Leap year

Coefficients	T-Stat	P[|T| > t]
2,5121	3,04	0,0026

Easter [8]

Coefficients	T-Stat	P[|T| > t]
1,6820	3,27	0,0012

Outliers

	Coefficients	T-Stat	P[|T| > t]
AO (4-2004)	38,9692	20,10	0,0000
LS (1-2017)	38,7633	16,13	0,0000
LS (1-2001)	-8,5669	-4,68	0,0000
AO (4-2010)	-8,6967	-4,67	0,0000
AO (3-2004)	8,5865	4,45	0,0000
AO (12-2003)	7,2801	3,94	0,0001

The *Partial concurrent adjustment* → *Estimate regression coefficient + outliers*

revision policy results

The *Partial concurrent adjustment → Estimate regression coefficients + Arima model*

The *Partial concurrent adjustment → Estimate regression coefficients + Arima model* option means that the ARIMA model, outliers and regression variables (except the calendar variables) are re-identified. All parameters are re-estimated. The transformation type remains unchanged.

The picture below presents the initial model (on the left) and the results of the refreshment procedure with the *Partial concurrent adjustment → Estimate regression coefficients + Arima model* option (on the right). The ARIMA part has been re-identified (a change from (2,1,0)(0,1,1) to (0,1,1)(1,1,1)). Also the regression coefficients for the calendar variables have been re-estimated. In the revised model there is no *Prespecified outliers* section. Therefore, the outliers were re-identified.

Summary

Estimation span: [1-2005 - 12-2016]
144 observations
Series has been log-transformed
Series has been corrected for leap year
Trading days effects (6 variables)
Easter [15] detected
4 detected outliers

Final model

Likelihood statistics
Number of effective observations = 131
Number of estimated parameters = 15

Loglikelihood = 330.49158009584664
Transformation adjustment = -608.1459835096218
Adjusted loglikelihood = -277.6544034137752

Standard error of the regression (ML estimate) = 0.01896469203769424
AIC = 585.3088068275504
AICC = 589.4827198710286
BIC (corrected for length) = -7.409339230469036

Scores at the solution
-0,004391 0,000967 -0,012902

Arima model
[(2,1,0)(0,1,1)].

	Coefficients	T-Stat	P[|T| > t]
Phi(1)	0,5040	5,65	0,0000
Phi(2)	0,2895	3,27	0,0014
BTheta(1)	-0,6188	-8,29	0,0000

Correlation of the estimates

	Phi(1)	Phi(2)	BTheta(1)
Phi(1)	1,0000	0,3982	0,1323
Phi(2)	0,3982	1,0000	0,0791
BTheta(1)	0,1323	0,0791	1,0000

Regression model
Trading days

	Coefficients	T-Stat	P[|T| > t]
Monday	0,0000	0,00	0,9975
Tuesday	-0,0046	-1,49	0,1386
Wednesday	0,0044	1,45	0,1505
Thursday	-0,0032	-1,03	0,3070
Friday	0,0103	3,37	0,0010
Saturday	-0,0031	-0,98	0,3313
Sunday (derived)	-0,0038	-1,20	0,2308

Joint F-Test = 3,86 (0,0015)

Easter [15]

Coefficients	T-Stat	P[|T| > t]
0,0831	12,38	0,0000

Outliers

	Coefficients	T-Stat	P[|T| > t]
AO (12-2006)	0,1164	7,00	0,0000
AO (12-2015)	-0,0990	-5,79	0,0000
AO (11-2015)	-0,0727	-4,23	0,0000
TC (1-2009)	0,0854	5,20	0,0000

Summary

Estimation span: [1-2005 - 12-2017]
156 observations
Series has been log-transformed
Series has been corrected for leap year
Trading days effects (6 variables)
Easter [15] detected
1 detected outlier

Final model

Likelihood statistics
Number of effective observations = 143
Number of estimated parameters = 13

Loglikelihood = 383.2719601133891
Transformation adjustment = -713.7404772425816
Adjusted loglikelihood = -330.4685171291925

Standard error of the regression (ML estimate) = 0.015644939706339882
AIC = 686.937034258385
AICC = 689.7587396847416
BIC (corrected for length) = -7.89875302500467

Scores at the solution
0,001814 -0,001954 -0,000274

Arima model
[(0,1,1)(1,1,1)].

	Coefficients	T-Stat	P[|T| > t]
Theta(1)	-0,4311	-5,41	0,0000
BPhi(1)	-0,3549	-3,54	0,0006
BTheta(1)	-0,9507	-12,77	0,0000

Correlation of the estimates

	Theta(1)	BPhi(1)	BTheta(1)
Theta(1)	1,0000	0,0569	0,4292
BPhi(1)	0,0569	1,0000	-0,0063
BTheta(1)	0,4292	-0,0063	1,0000

Regression model
Mean

	Coefficient	T-Stat	P[|T| > t]
mu	-0,0006	-2,10	0,0380

Trading days

	Coefficients	T-Stat	P[|T| > t]
Monday	-0,0027	-1,14	0,2578
Tuesday	0,0015	0,64	0,5209
Wednesday	0,0023	0,96	0,3401
Thursday	0,0006	0,26	0,7982
Friday	0,0087	3,77	0,0002
Saturday	-0,0033	-1,37	0,1743
Sunday (derived)	-0,0072	-2,96	0,0036

Joint F-Test = 8,70 (0,0000)

Easter [15]

Coefficients	T-Stat	P[|T| > t]
0,0197	3,92	0,0001

Outliers

	Coefficients	T-Stat	P[|T| > t]
AO (4-2010)	-0,0554	-4,02	0,0001

The *Partial concurrent adjustment* → *Estimate regression coefficient + Arima model* revision policy results

Partial concurrent adjustment → Estimate regression coefficients + Last outliers

The *Partial concurrent adjustment → Estimate regression coefficients + Last outliers* strategy means that the ARIMA model, outliers (except for the last year of the sample) and other regression parameters are not re-identified. All parameters of the Reg-ARIMA model are re-estimated. The software tests for outliers in the last year of the data span and will include in the model those which are statistically significant. The transformation type remains unchanged.

The picture below presents the initial model (on the left) and the results of the refreshment procedure with the *Partial concurrent adjustment → Estimate regression coefficients + Last outliers* option (on the right). The parameters of the ARIMA part have been re-estimated and their values have been updated. Also the regression coefficients have been re-estimated. The number of estimated coefficients in the revised model is larger than the initial model because an additional outlier has been identified in the last year of the data span.

Summary

Estimation span: [7-1996 - 12-2016]
246 observations
Trading days effects (7 variables)
Easter [8] detected
5 detected outliers

Final model

Likelihood statistics

Number of effective observations = 233
Number of estimated parameters = 16

Loglikelihood = -559.6616717869163
Standard error of the regression (ML estimate) = 2.669031738674505
AIC = 1151.3233435738325
AICC = 1153.841862092351
BIC (corrected for length) = 2.314356746231504

Scores at the solution
-0,000004 -0,000414

Arima model

[(0,1,1)(1,1,0)]

	Coefficients	T-Stat	P[|T| > t]
Theta(1)	-0,4902	-8,33	0,0000
BPhi(1)	0,1680	2,45	0,0152

Correlation of the estimates

	Theta(1)	BPhi(1)
Theta(1)	1,0000	0,0784
BPhi(1)	0,0784	1,0000

Regression model

Trading days

	Coefficients	T-Stat	P[|T| > t]
Monday	-0,5102	-1,92	0,0562
Tuesday	0,2288	0,88	0,3774
Wednesday	0,1073	0,40	0,6905
Thursday	0,2028	0,75	0,4536
Friday	0,9280	3,48	0,0006
Saturday	-0,3434	-1,27	0,2071
Sunday (derived)	-0,6134	-2,28	0,0235

Joint F-Test = 8,13 (0,0000)

Leap year

	Coefficients	T-Stat	P[|T| > t]
	2,6712	3,07	0,0024

Easter [8]

	Coefficients	T-Stat	P[|T| > t]
	1,6759	3,08	0,0023

Outliers

	Coefficients	T-Stat	P[|T| > t]
AO (4-2004)	19,8713	10,12	0,0000
LS (1-2001)	-8,5643	-4,62	0,0000
AO (4-2010)	-8,7157	-4,62	0,0000
AO (3-2004)	8,6114	4,40	0,0000
AO (12-2003)	7,2918	3,90	0,0001

Summary

Estimation span: [7-1996 - 7-2017]
253 observations
Trading days effects (7 variables)
Easter [8] detected
5 pre-specified outliers
1 detected outlier

Final model

Likelihood statistics

Number of effective observations = 240
Number of estimated parameters = 17

Loglikelihood = -573.4952681950751
Standard error of the regression (ML estimate) = 2.6361922969352976
AIC = 1180.9905363901503
AICC = 1183.747293146907
BIC (corrected for length) = 2.3040470567397255

Scores at the solution
-0,000012 -0,000444

Arima model

[(0,1,1)(1,1,0)]

	Coefficients	T-Stat	P[|T| > t]
Theta(1)	-0,4908	-8,49	0,0000
BPhi(1)	0,1679	2,53	0,0120

Correlation of the estimates

	Theta(1)	BPhi(1)
Theta(1)	1,0000	0,0615
BPhi(1)	0,0615	1,0000

Regression model

Trading days

	Coefficients	T-Stat	P[|T| > t]
Monday	-0,5212	-2,01	0,0454
Tuesday	0,2349	0,93	0,3516
Wednesday	0,1042	0,40	0,6899
Thursday	0,2102	0,79	0,4294
Friday	0,9059	3,51	0,0005
Saturday	-0,3332	-1,25	0,2117
Sunday (derived)	-0,6007	-2,29	0,0230

Joint F-Test = 8,39 (0,0000)

Leap year

	Coefficients	T-Stat	P[|T| > t]
	2,5121	3,04	0,0026

Easter [8]

	Coefficients	T-Stat	P[|T| > t]
	1,6820	3,27	0,0012

Prespecified outliers

	Coefficients	T-Stat	P[|T| > t]
AO (4-2004)	38,9692	20,10	0,0000
LS (1-2001)	-8,5669	-4,68	0,0000
AO (4-2010)	-8,6967	-4,67	0,0000
AO (3-2004)	8,5865	4,45	0,0000
AO (12-2003)	7,2801	3,94	0,0001

Outliers

	Coefficients	T-Stat	P[|T| > t]
LS (1-2017)	38,7633	16,13	0,0000

The *Partial concurrent adjustment* → *Estimate regression coefficient + Last outliers* revision policy results

Plug-ins for JDemetra+

JDemetra+ is an application that supports plug-ins, which are components adding specific features to the existing software.

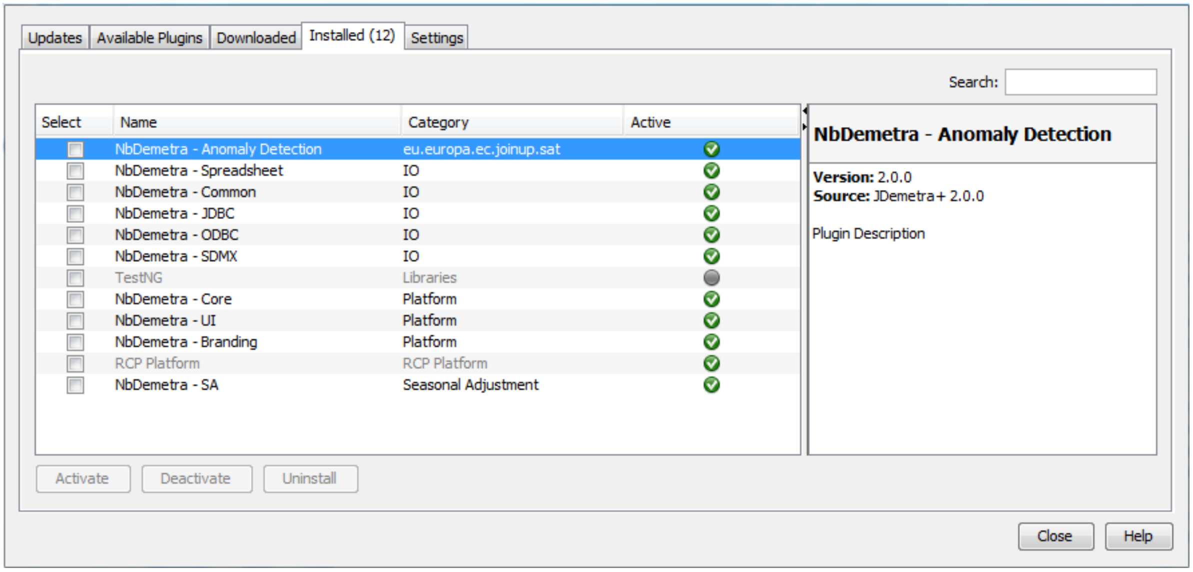
Main functions

Default Plugins

Name	Category	Description
NbDemetra – Anomaly detection	SA core algorithms	Identification of outliers
NbDemetra – Spreadsheet	IO (Input/output)	Time series providers for spreadsheet (Excel, OpenOffice)
NbDemetra – Common	IO (Input/output)	Common time series providers, like XML and TXT
NbDemetra – JDBC	IO (Input/output)	Time series provider for the JDBC sources
NbDemetra – ODBC	IO (Input/output)	Time series provider for the ODBC sources
NbDemetra – SDMX	IO (Input/output)	Time series provider for SDMX files
NbDemetra – Core	SA core algorithms	Encapsulation of the core algorithms
NbDemetra – UI	SA core algorithms	Basic graphical components
NbDemetra – Branding	SA core algorithms	

Name	Category	Description
NbDemetra – SA	SA core algorithms	Default SA framework, including TRAMO-SEATS and X-13ARIMA-SEATS. This implementation can lead to small differences in comparison with the original programs.

This list is displayed in the *Installed* panel. This panel is available from the *Plugin* functionality and it is activated from the *Tools* menu (Figure *Activation of the Plugin functionality from the Tools menu* in Plugins section).



Plugins-list

Bundesbank plug-ins

- ConCur: The plug-in ConCur supports the controlled current adjustment approach. It supports the storage of the current components and offers graphical tools to compare forecasted and re-estimated figures. Furthermore, a pre-defined summary of the output containing the most important quality measures can be exported to HTML files.
- KIX: The plug-in KIX (German for chain-linked index) has been designed to facilitate the handling of this index type. It offers addition and subtraction of two or more chain-linked time series as well as the computation of contributions of growth.
  - KIX2.0: KIX 2.0 offers addition and subtraction of two or more chain-linked time series as well as the computation of contributions of growth following the concept of

annual overlap. Contributions to growth are calculated with the partial contribution to growth approach.

  - KIXE: KIX_E offers addition and subtraction of two or more chain-linked time series as well as the computation of contributions of growth following the concept of one-period overlap. Contributions to growth are calculated with the aid of the Ribe (1999) contribution to growth approach.
  - KIX: The program KIX-CC offers for continuously chain-linked indices the aggregation or disaggregation of two or more indices, or the calculation of contributions to growth.

- TransReg: The plug-in TransReg allows the user to carry out grouping and centring of user-defined regression variables in JD+.
- Xlsx2Ws: The plug-in Xlsx2Ws allows the converting of specific workspace information to a xlsx file and vice versa.

National Bank of Belgium plug-ins

- AccessThis JDemetra+ extension is a pure java library for reading time series from MS Access databases. It currently supports versions 2000-2016 read/write and 97 read-only.Being a pure Java library, you don't need MS Access installed in order to read Access files. (edit versions info here)
- SDMX: This plugin provides time series from SDMX to JDemetra+ by querying web services or parsing files.
- SA Advanced: This module provides some experimental seasonal adjustment methods (with RegArima preprocessing), basic structural models, generalized airline models and airline + seasonal noise models (called mixed airline).
  - gairline: generalized airline model
  - mairline: mixed airline model
  - mixedfreq: mixed frequencies seasonal adjustment
  - sssts: Seasonal specific structural time series
  - sts: Structural time series
- Benchmarking: This module provides some experimental methods for temporal disaggregation and multi-variate benchmarking:Chow-Lin, Fernandez, Litterman, Cholette, Calendarization.
- Nowcasting: Nowcasting is often defined as the prediction of the present, the very near future and the very recent past. The plug-in developed at the National Bank of Belgium helps to operationalize the process of nowcasting. It can be used to specify and estimate dynamic factor models and visualize how the real-time dataflow updates expectations,

as for instance in Banbura and Modugno (2010). The software can also be used to perform pseudo out-of-sample forecasting evaluations that consider the calendar of data releases, contributing to the formalization of the nowcasting problem originally proposed by Giannone, et al. (2008) or Evans (2005).

Installation procedure

Installation from GUI

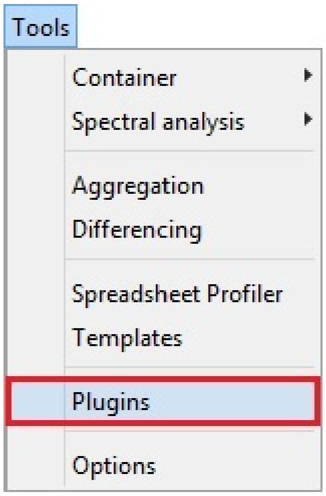
menu>tools> plug-ins

The *Plugins* window includes five panels: *Updates*, *Available plugins*, *Downloaded*, *Installed* and *Settings*, some of them however are not operational in the current version of the software.

- The *Updates* panel offers the user the option to manually check if some updates of the already installed plugins are available. This functionality, however, is currently not operational for the JDemetra+ plugins.
- The *Available plugins* panel allows the downloading of all plugins that are related to JDemetra+. This functionality, however, is currently not operational for the JDemetra+ plugins.
- The *Downloaded* panel is designed for the installation of new plugins from a local machine. This process in explained in more detail below.
- The *Settings* panel is designated for adding update centres, which are the locations that hold plugins. For each centre the user can specify proxy settings and a time interval to automatically check for any updates. At the moment this functionality is not operational for the JDemetra+ plugins.

Installation of the new plugins from the local machine can be done from the *Plugin* functionality activated from the *Tools* menu.

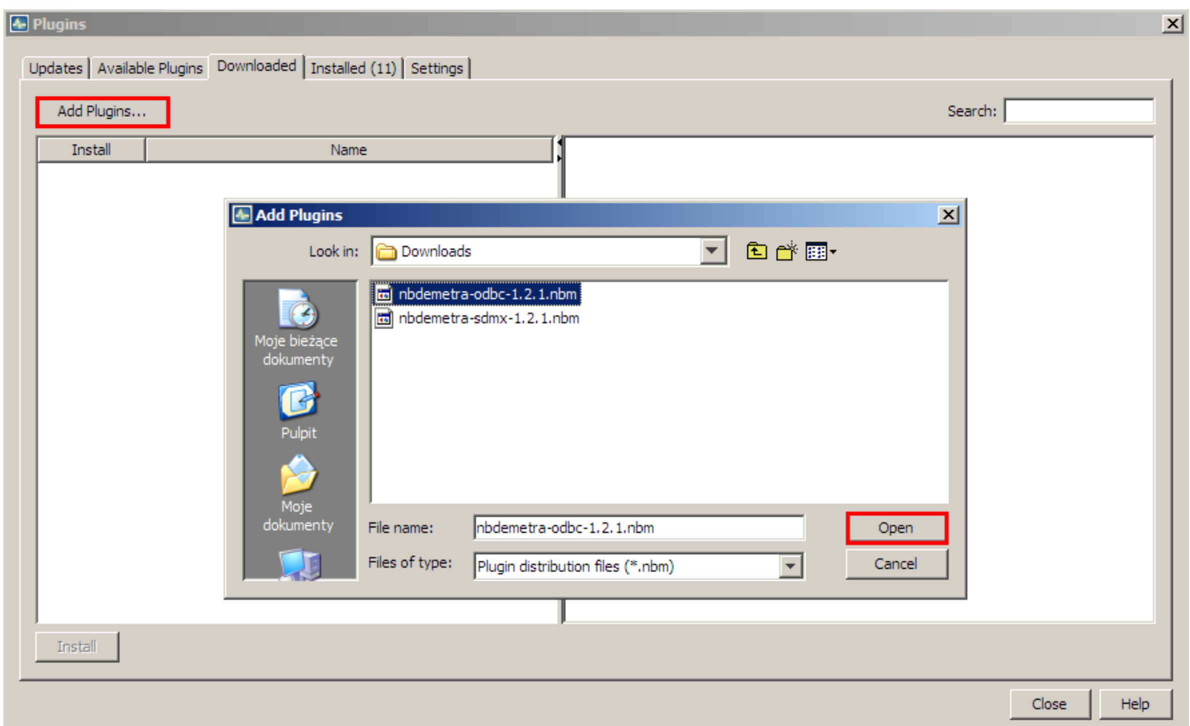
Installation of the new plugins from the local machine can be done from the *Plugin* functionality activated from the *Tools* menu.



Activation of the *Plugin* functionality from the *Tools* menu

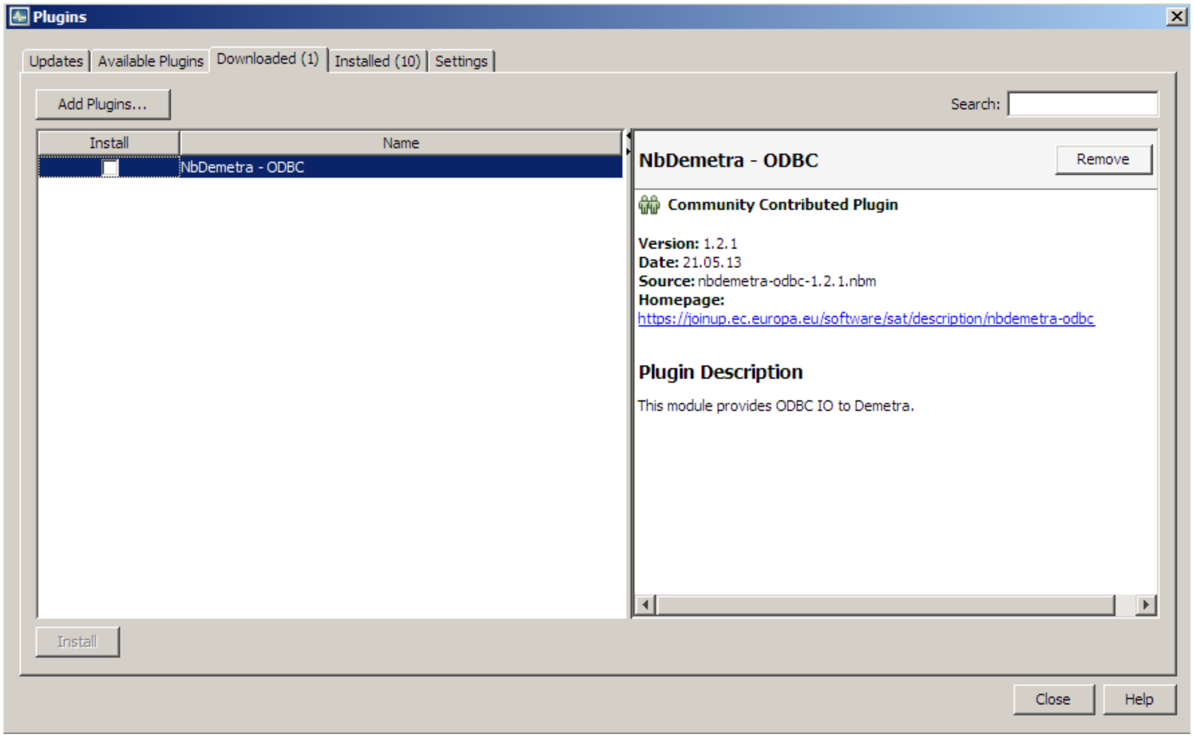
To start the process, go to the *Downloaded* panel and click on the **Add Plugins...** option. Next the user should select the plugins from the folder in which the plugins have been saved and click the **OK** button.

"



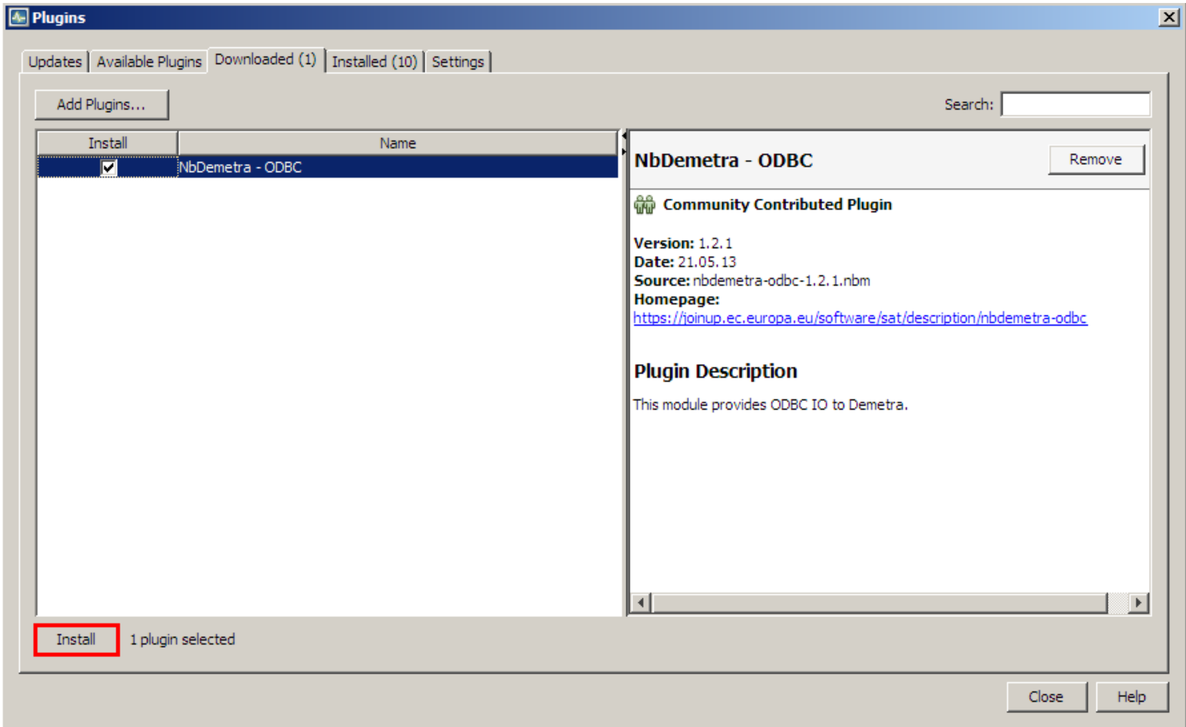
The *Downloaded* panel – the choice of available plugins

The new plugin is now visible in the panel.



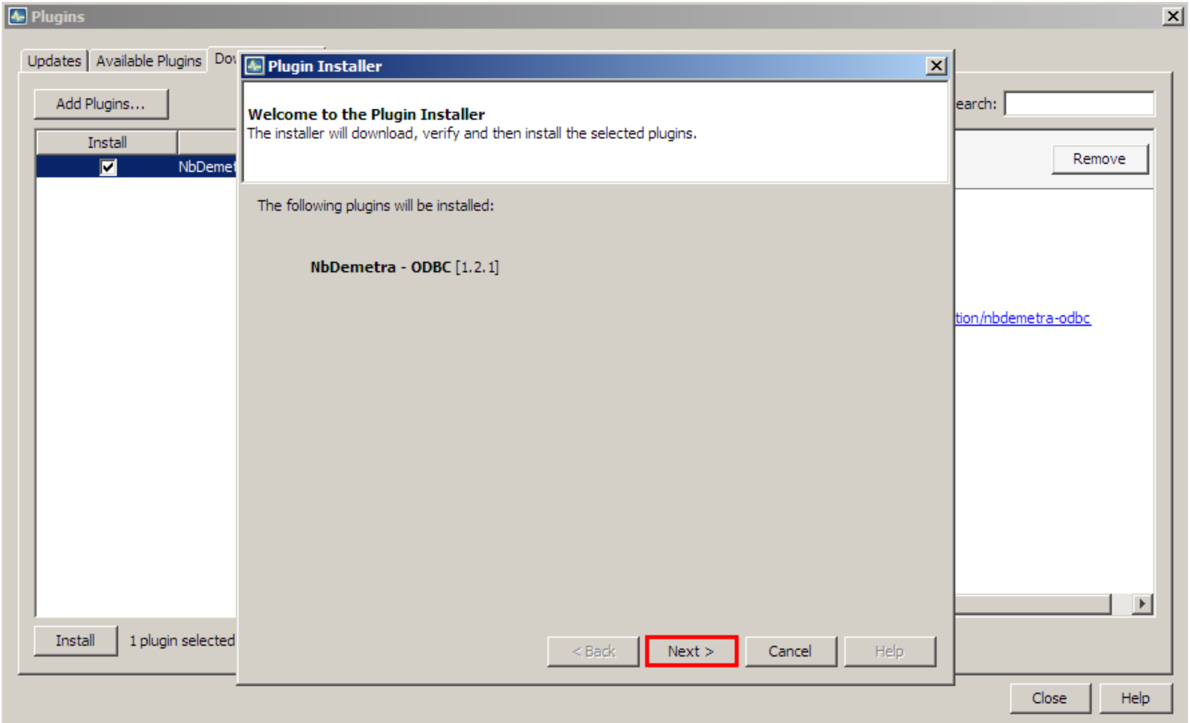
A downloaded plugin

Click on it and choose the **Install** button.



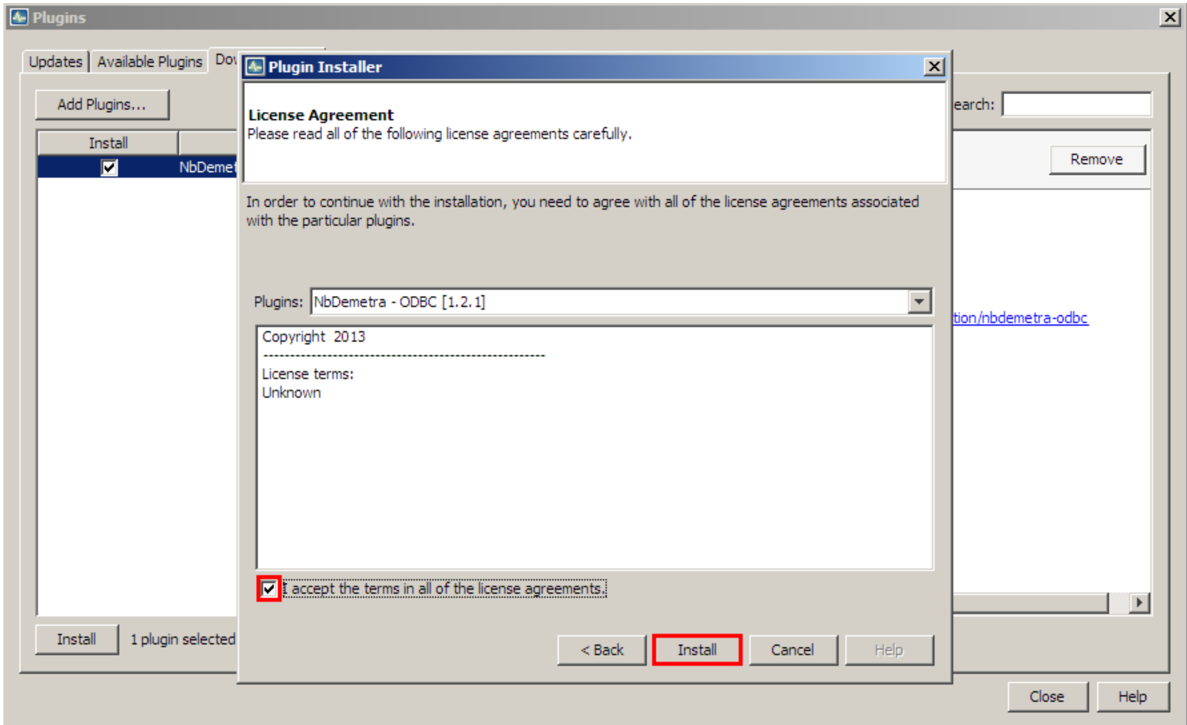
Starting an installation procedure

There is a wizard that allows the user to install the marked plugin(s). In the first step choose **Next** to continue or **Cancel** to terminate the process.



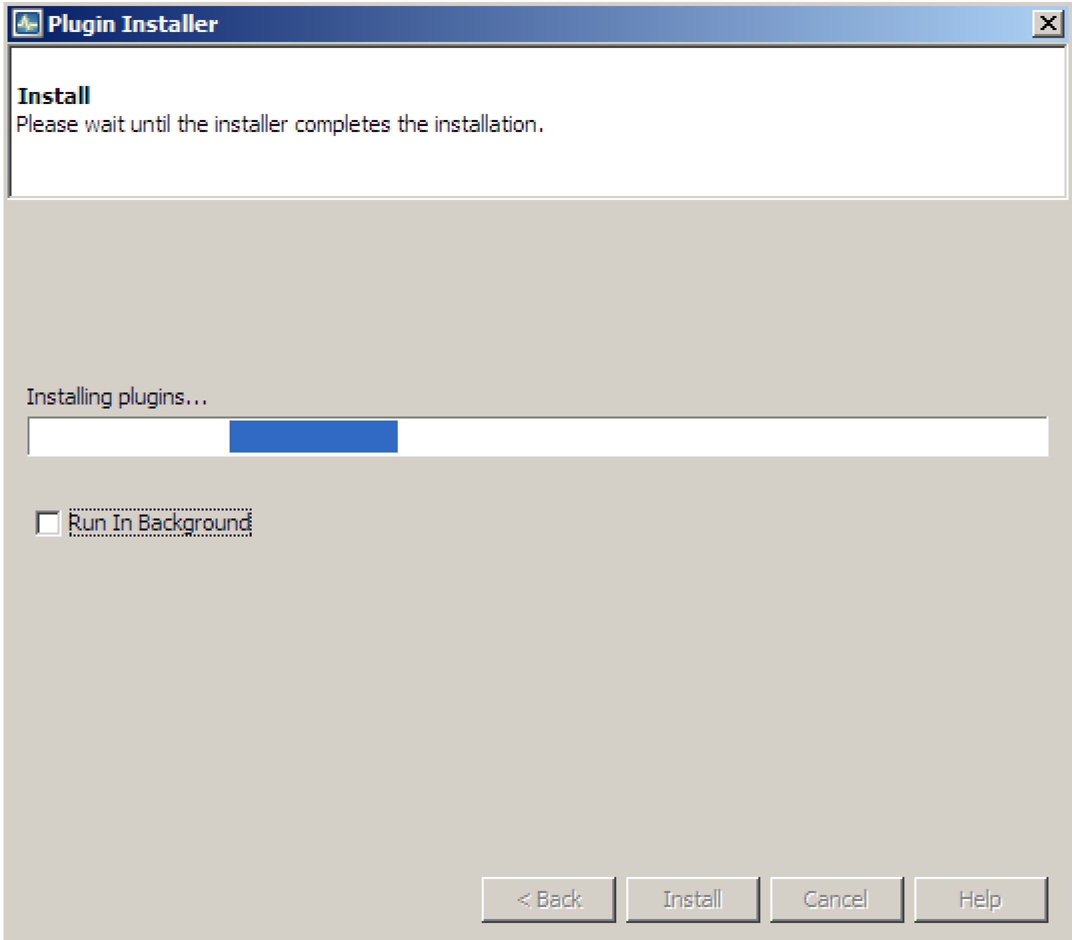
Installation wizard window

Next, mark the terms of agreements and choose **Install**.



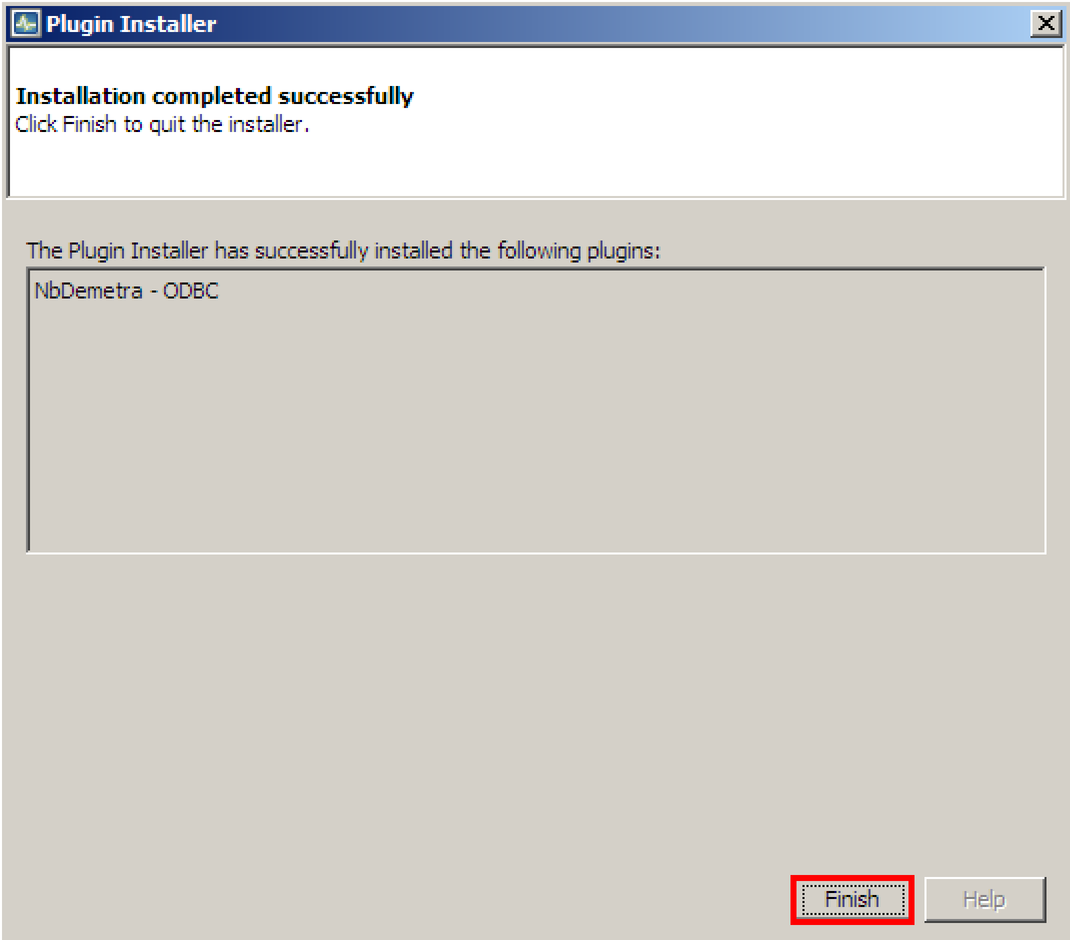
Initiating installation process

Then the process is started.



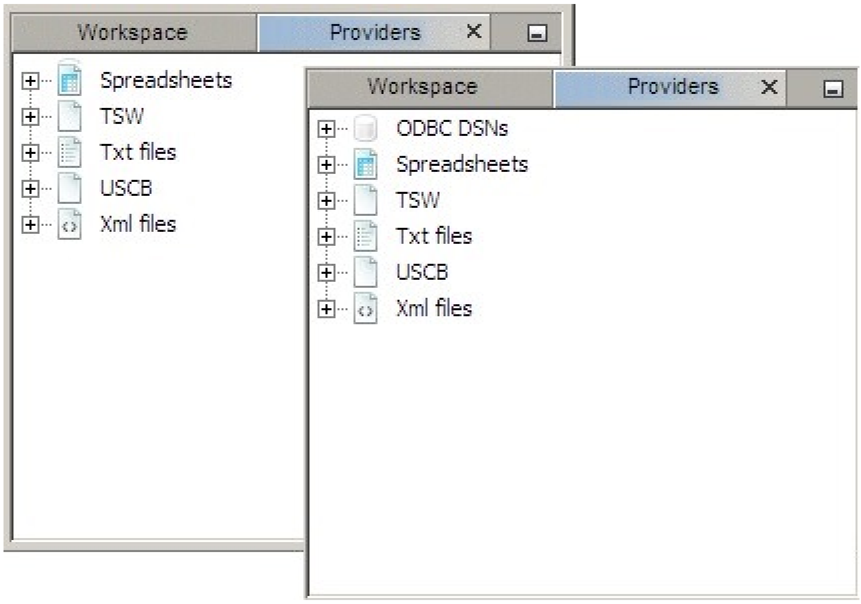
Installation in progress

After a while JDemetra+ will provide an update in the installation process. Click **Finish** to close the window.



Installation completed

Once the process is finished, the newly installed plugin is automatically integrated within the software. The picture below compares the view of the *Workspace* window before (on the left) and after (on the right) the installation of the NbDemetra-ODBC plugin.



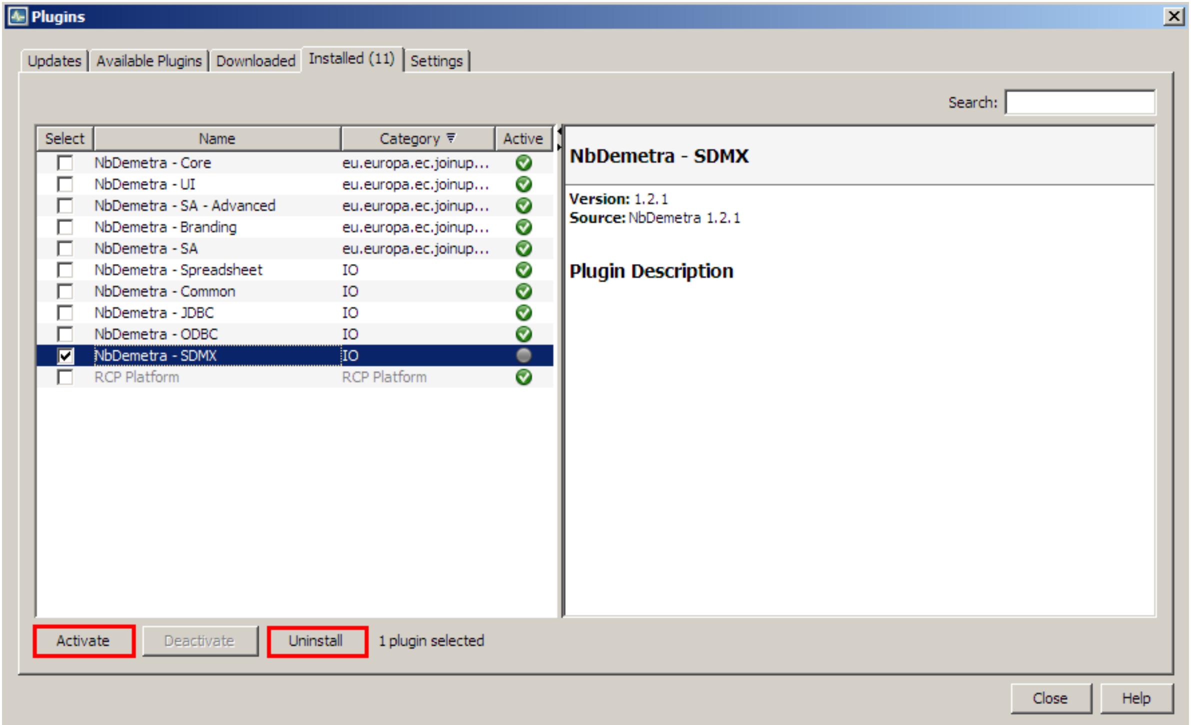
The impact of the plugin on the interface

The list of all installed plugins is displayed in the fourth panel. To modify the current settings mark the plugin (by clicking the checkbox in the *Select* column) and chose an action.

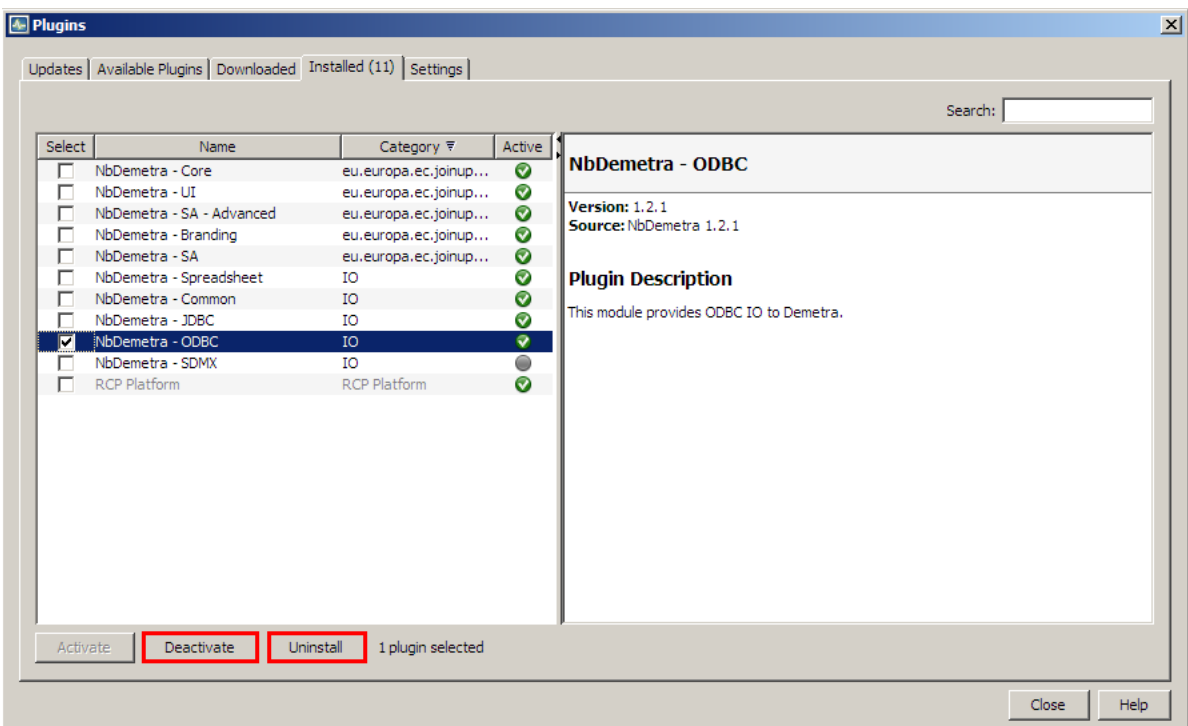
The following options are available:

- **Activate** – activates the marked plugin if it is currently inactive. The option is available for inactive plugins (see the picture below);
- **Deactivate** – deactivates the marked plugin if it is currently active. The option is available for active plugins (see the picture below);
- **Uninstall** – uninstalls the marked plugin.

Inactive plugins can be activated or uninstalled.

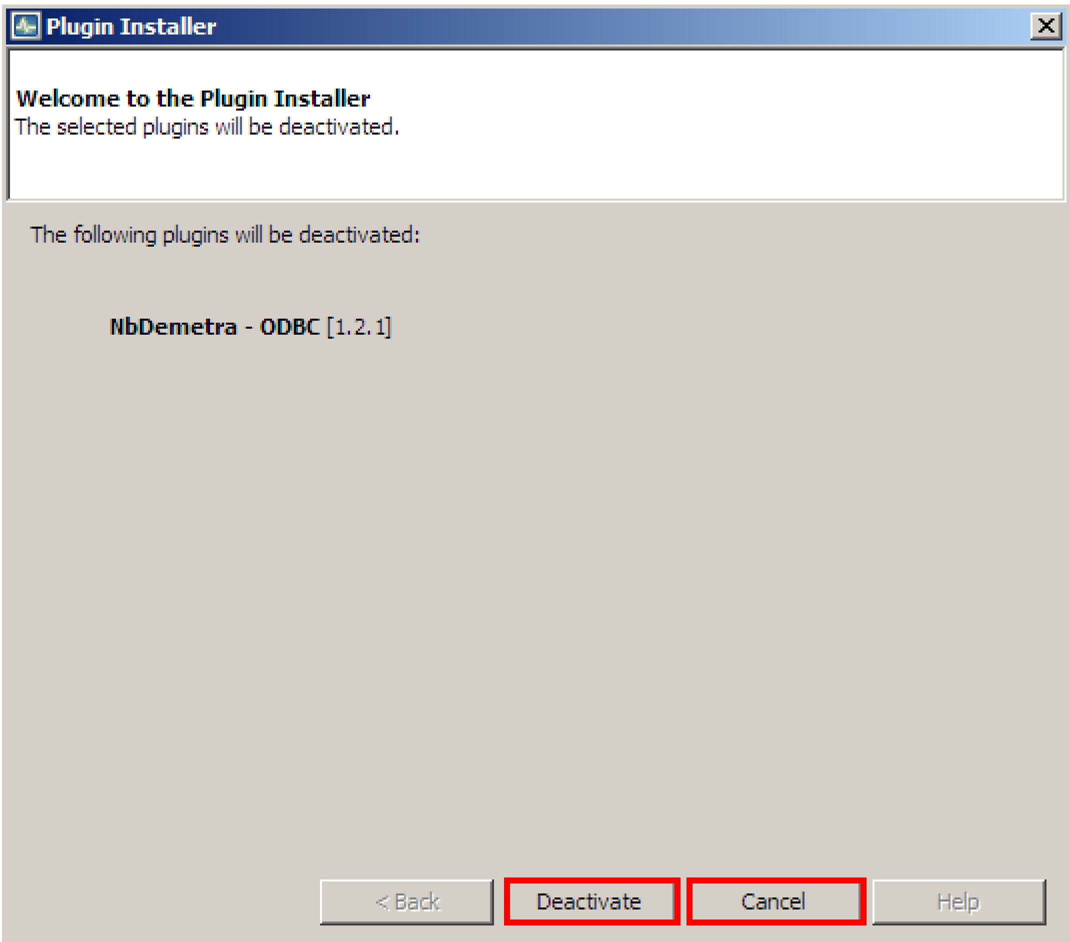


Active plugins can be deactivated or uninstalled



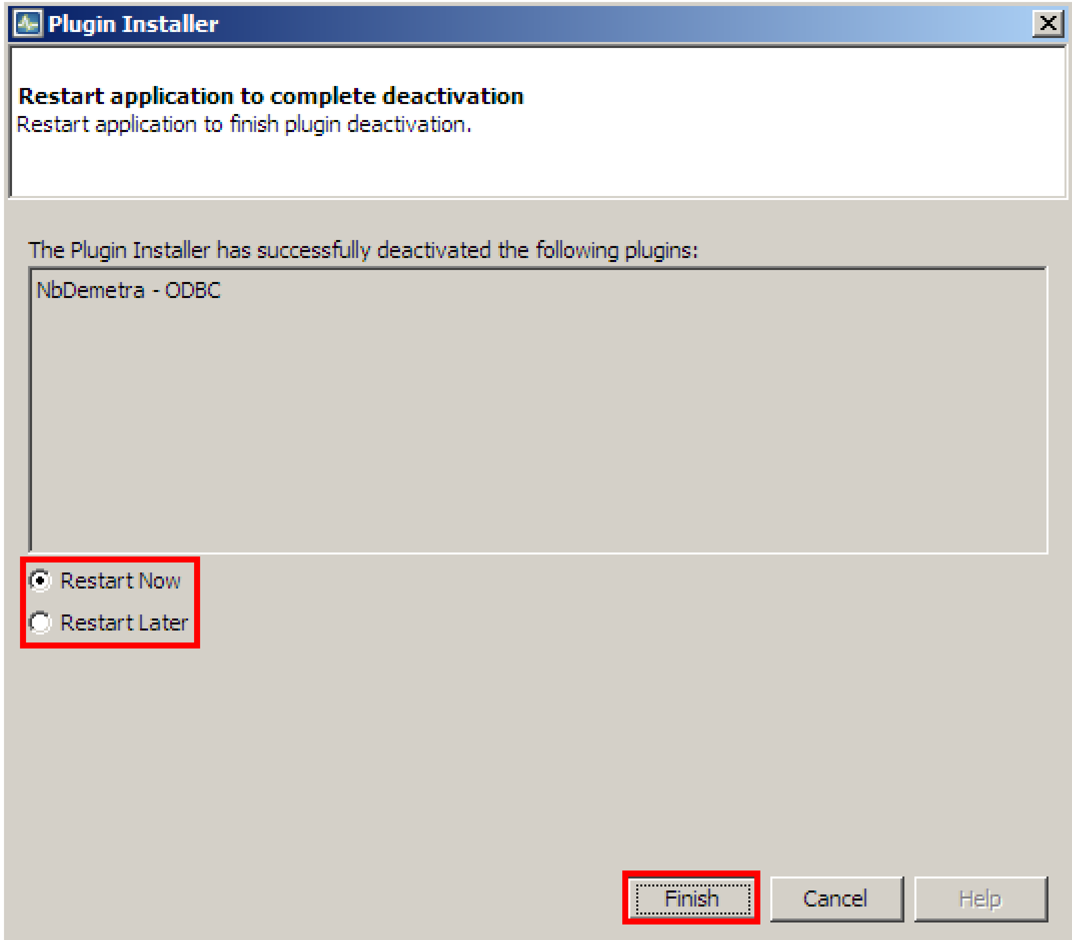
List of plugins – deactivation

There is a wizard that allows the user to activate/deactivate/uninstall the marked plugin(s). The example below illustrates the deactivation process. In the first step the user is expected to confirm or cancel the deactivation.



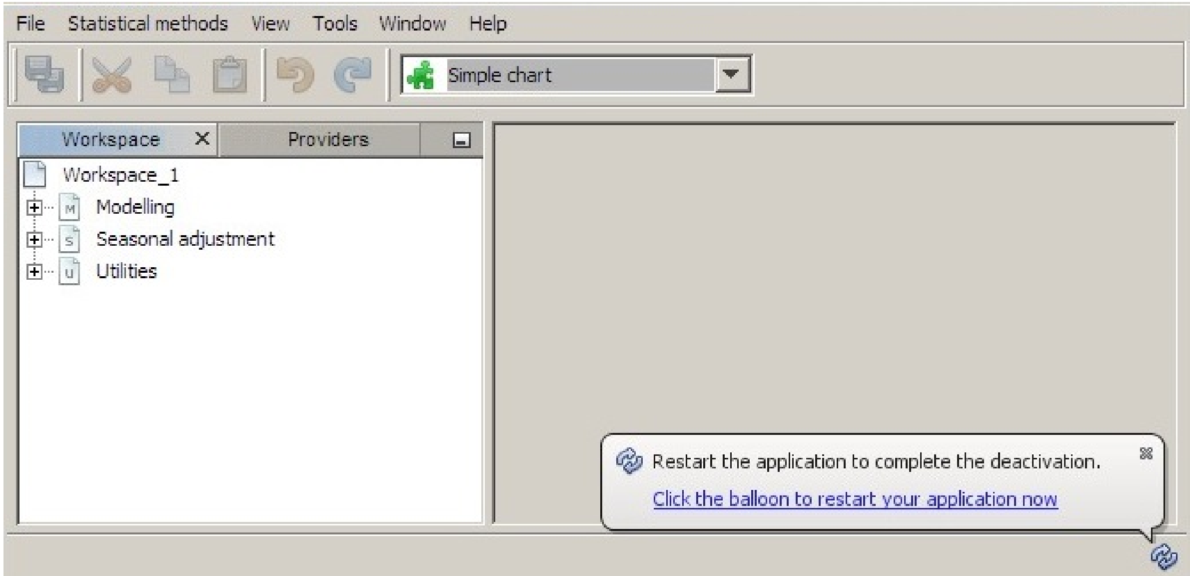
Plugin's deactivation process

In the second step the user should decide if the software will be restarted immediately after the uninstallation is completed or not.



The final step of the installation procedure

It is possible to delay the restart of the application, although the restart is necessary to complete the process.


File Statistical methods View Tools Window Help
Simple chart
Workspace
Providers
Workspace_1
Modelling
Seasonal adjustment
Utilities
Restart the application to complete the deactivation.
Click the balloon to restart your application now

Part III

Methods

This part describes in greater detail:

- Reg-Arima modelling
- X-11: moving average based decomposition
- SEATS: Arima model based decomposition
- STL: Loess based decomposition
- Benchmarking and temporal disaggregation
- Spectral analysis tools
- Trend Estimation
- Tests for seasonality and residuals
- Structural time series and state space framework

Spectral Analysis Principles and Tools

Under construction.

Spectral analysis concepts

A time series x_t with stationary covariance, mean μ and k^{th} autocovariance $E(x_t - \mu)(x_{t-k}\mu)) = \gamma(k)$ can be described as a weighted sum of periodic trigonometric functions: $sin(\omega t)$ and $cos(\omega t)$, where $\omega = \frac{2*pi}{T}$ denotes frequency. Spectral analysis investigates this frequency domain representation of x_t to determine how important cycles of different frequencies are in accounting for the behaviour of x_t.

Assuming that the autocovariances $\gamma(k)$ are absolutely summable($\sum_{k=-\infty}^{\infty} |\gamma(k)| < \infty$), the autocovariance generating function, which summarizes these autocovariances through a scalar valued function, is given by equation Equation **??** [4].

$$acgf(z) = \sum_{k=-\infty}^{\infty} z^k \gamma(k)$$

where z denotes complex scalar.

Once the equation Equation **??** is divided by π and evaluated at some $z = e^{-i\omega} = cos\omega - isin\omega$, where $i = \sqrt{-1}$ and ω is a real scalar, $-\infty < \omega < \infty$, the result of this transformation is called a population spectrum $f(\omega)$for x_t, given in equation Equation **??** [5].

$$f(\omega) = \frac{1}{\pi} \sum_{k=-\infty}^{\infty} e^{-ik\omega} \gamma(k)$$

Therefore, the analysis of the population spectrum in the frequency domain is equivalent to the examination of the autocovariance function in the time domain analysis; however it provides an alternative way of inspecting the process. Because $f(\omega)$d is interpreted as a contribution to the variance of components with frequencies in the range $(\omega, \ \omega + d\omega)$, a peak

[4]HAMILTON, J.D. (1994).
[5]HAMILTON, J.D. (1994).

in the spectrum indicates an important contribution to the variance at frequencies near the value that corresponds to this peak.

As $e^{-i\omega} = cos\omega - isin\omega$, the spectrum can be also expressed as in equation Equation **??**.

$$f(\omega) = \frac{1}{\pi} \sum_{k=-\infty}^{\infty} (cos\omega k - isin\omega k)\gamma(k)$$

Since $\gamma(k) = \gamma(-k)$ (i.e. $\gamma(k)$ is an even function of k) and $\sin{(-x)}$ = -$\sin x$, Equation **??** can be presented as equation

$$f(\omega) = \frac{1}{\pi}[\ \gamma(0) + 2\sum_{k=1}^{\infty}\ \gamma(k)cos\ \mathrm{k}]$$

This implies that if autocovariances are absolutely summable the population spectrum exists and is a continuous, real-valued function of ω. Due to the properties of trigonometric functions ($\cos(-\omega k) = \cos($ k) and $\cos(\omega + 2\pi j)k = cos(\omega k)$) the spectrum is a periodic, even function of ω, symmetric around $\omega = 0$. Therefore, the analysis of the spectrum can be reduced to the interval $(-\pi, \pi)$. The spectrum is non-negative for all $\omega \in (-\pi, \pi)$.

The shortest cycle that can be distinguished in a time series lasts two periods. The frequency which corresponds to this cycle is $\omega = \pi$ and is called the Nyquist frequency. The frequency of the longest cycles that can be observed in the time series with n observations is $\omega = \frac{2\pi}{n}$ and is called the fundamental (Fourier) frequency.

Note that if x_t is a white noise process with zero mean and variance σ^2, then for all $|k| > 0$ $\gamma(k) = 0$ and the spectrum of x_t is constant ($f(\omega) = \frac{\sigma^2}{\pi}$) since each frequency in the spectrum contributes equally to the variance of the process[6].

The aim of spectral analysis is to determine how important cycles of different frequencies are in accounting for the behaviour of a time series[7]. Since spectral analysis can be used to detect the presence of periodic components, it is a natural diagnostic tool for detecting trading day effects as well as seasonal effects[8]. Among the tools used for spectral analysis are the autoregressive spectrum and the periodogram.

The explanations given in the subsections of this node derive mainly from DE ANTONIO, D., and PALATE, J. (2015) and BROCKWELL, P.J., and DAVIS, R.A. (2006).

[6] BROCKWELL, P.J., and DAVIS, R.A. (2002).
[7] HAMILTON, J.D. (1994).
[8] SOKUP, R.J., and FINDLEY, D. F. (1999).

Theoretical spectral density of an ARIMA model

Spectral density estimation

Method 1: The periodogram

For any given frequency ω the sample periodogram is the sample analog of the sample spectrum. In general, the periodogram is used to identify the periodic components of unknown frequency in the time series. X-13ARIMA-SEATS and TRAMO-SEATS use this tool for detecting seasonality in raw time series and seasonally adjusted series. Apart from this it is applied for checking randomness of the residuals from the ARIMA model.

To define the periodogram, first consider the vector of complex numbers[9]:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_n \end{bmatrix} \in \mathbb{C}^n$$

where $\mathbb{C}^n$ is the set of all column vectors with complex-valued components.

The Fourier frequencies associated with the sample size n are defined as a set of values $\omega_j = \frac{2\pi j}{n}$, $j = -[\frac{n-1}{2}], \ldots, [\frac{n}{2}]$, $-\pi < \omega_j \leq \pi$, $j \in F_n$, where $[n]$ denotes the largest integer less than or equal to n. The Fourier frequencies, which are called harmonics, are given by integer multiples of the fundamental frequency $\frac{2\pi}{n}$.

Now the n vectors $e_j = n^{-\frac{1}{2}}(e^{-i\omega_j}, e^{-i2\omega_j}, \ldots, e^{-in\omega_j})'$ can be defined. Vectors $e_1, \ldots, e_n$ are orthonormal in the sense that:

$$\mathbf{e}_j^* \mathbf{e}_k = n^{-1} \sum_{r=1}^{n} e^{ir(\omega_j - \omega_k)} = \begin{matrix} 1, \ if \ j = k \\ 0, \ if \ j \neq k \end{matrix}$$

where $\mathbf{e}_j^*$ denotes the row vector, which k^{th} component is the complex conjugate of the k^{th} component of $\mathbf{e}_j$.[10] These vectors are a basis of F_n, so that any $\mathbf{x} \in \mathbb{C}^n$ can be expressed as a sum of n components:

[9] BROCKWELL, P.J., and DAVIS, R.A. (2002).

[10] For details see BROCKWELL, P.J., and DAVIS, R.A. (2006).

$$\mathbf{x} = \sum_{j=-[\frac{n-1}{2}]}^{[\frac{n}{2}]} a_j \mathbf{e}_j$$

where the coefficients $a_j = \mathbf{e}_j^{*}\mathbf{x} = n^{-\frac{1}{2}} \sum_{t=1}^{n} x_t e^{-it\omega_j}$ are derived from Equation **??** by multiplying the equation on the left by $\mathbf{e}_j^{*}$ and using Equation **??**.

The sequence of $\{a_j, j \in F_n\}$ is referred as a discrete Fourier transform of $\mathbf{x} \in \mathbb{C}^n$ and the periodogram $I(\omega_j)$ of $\mathbf{x}$ at Fourier frequency $\omega_j = \frac{2\pi j}{n}$ is defined as the square of the Fourier transform $\{a_j\}$ of $\mathbf{x}$:

$$I(\omega_j) = |a_j|^2 = n^{-1} | \sum_{t=1}^{n} x_t e^{-it\omega_j} |^2$$

From Equation **??** and Equation **??** it can be shown that in fact the periodogram decomposes the total sum of squares $\sum_{t=1}^{n} |x_t|^2$ into a sums of components associated with the Fourier frequencies ω_j:

$$\sum_{t=1}^{n} |x_t|^2 = \sum_{j=-[\frac{n-1}{2}]}^{[\frac{n}{2}]} |a_j|^2 = \sum_{j=-[\frac{n-1}{2}]}^{[\frac{n}{2}]} I(\omega_j)$$

If $\mathbf{x} \in R^n$, ω_j and $-\omega_j$ are both in $[-\pi, -\pi]$ and a_j is presented in its polar form (i.e. $a_j = r_j \exp(i\theta_j)$), where r_j is the modulus of a_j, then Equation **??** can be rewritten in the form:

$$\mathbf{x} = a_0 \mathbf{e}_0 + \sum_{j=1}^{[\frac{n-1}{2}]} 2^{1/2} r_j (\mathbf{c}_j \cos\theta_j - \mathbf{s}_j \sin\theta_j) + a_{n/2} \mathbf{e}_{n/2}$$

The orthonormal basis for R^n is $\{\mathbf{e}_0, \mathbf{c}_1, \mathbf{s}_1, \dots, \mathbf{c}_{[\frac{n-1}{2}]}, \mathbf{s}_{[\frac{n-1}{2}]}, \mathbf{e}_{\frac{n}{2}(excluded\ if\ n\ is\ odd)}\}$, where:

$\mathbf{e}_0$ is a vector composed of n elements equal to $n^{-1/2}$, which implies that $\mathbf{a}_0\mathbf{e}_0 = (n^{-1}\sum_{t=1}^{n} x_t, \dots, n^{-1}\sum_{t=1}^{n} x_t)^{'}$;

$$\mathbf{c}_j = (\frac{n}{2})^{-1/2} (\cos\omega_j, \cos 2\omega_j, \dots, \cos n\omega_j)^{'}, for 1 \le j \le [\frac{(n-1)}{2}]$$

;

$$\mathbf{s}_j = (\frac{n}{2})^{-1/2} (\sin\omega_j, \sin 2\omega_j, \dots, \sin n\omega_j)^{'},\ for\ 1 \le j \le [\frac{(n-1)}{2}]$$

;

$$\mathbf{e}_{n/2} = (-(n^{-\frac{1}{2}}), n^{-\frac{1}{2}}, \ldots, -(n)^{-\frac{1}{2}}), n^{-\frac{1}{2}})'$$

.

Equation Equation **??** can be seen as an OLS regression of x_t on a constant and the trigonometric terms. As the vector of explanatory variables includes n elements, the number of explanatory variables in Equation **??** is equal to the number of observations. HAMILTON, J.D. (1994) shows that the explanatory variables are linearly independent, which implies that an OLS regression yields a perfect fit (i.e. without an error term). The coefficients have the form of a simple OLS projection of the data on the orthonormal basis:

$$\hat{a}_0 = \frac{1}{\sqrt{n}} \sum_{t=1}^{n} x_t \tag{0.1}$$

$$\hat{a}_{n/2} = \frac{1}{\sqrt{n}} \sum_{t=1}^{n} (-1)^t x_t (\text{only when n is even}) \tag{0.2}$$

$$\hat{a}_0 = \frac{1}{\sqrt{n}} \sum_{t=1}^{n} x_t \tag{0.3}$$

$$\hat{\alpha}_j = 2^{1/2} r_j \cos\theta_j = (\frac{n}{2})^{-1/2} \sum_{t=1}^{n} x_t \cos(t\frac{2\pi j}{n}), j = 1, \ldots, [\frac{n-1}{2}] \tag{0.4}$$

$$\hat{\beta}_j = 2^{1/2} r_j \sin\theta_j = (\frac{n}{2})^{-1/2} \sum_{t=1}^{n} x_t \sin(t\frac{2\pi j}{n}), j = 1, \ldots, [\frac{n-1}{2}] \tag{0.5}$$

With Equation **??** the total sum of squares $\sum_{t=1}^{n} |x_t|^2$ can be decomposed into $2 \times [\frac{n-1}{2}]$ components corresponding to $\mathbf{c}_j$ and $\mathbf{s}_j$, which are grouped to produce the "frequency ω_j" component for $1 \geq j \geq [\frac{n-1}{2}]$. As it is shown in the table below, the value of the periodogram at the frequency ω_j is the contribution of the j^{th} harmonic to the total sum of squares $\sum_{t=1}^{n} |x_t|^2$.

Decomposition of sum of squares into components corresponding to the harmonics

Frequency	**Degrees of freedom**	**Sum of squares decomposition**
ω_0(mean)	1	$a_0^2 \ = n^{-1}(\sum_{t=1}^{n} x_t)^2 = I(0)$
ω_1	2	$2r_1^2 \ = 2\|a_1\|^2 = 2I(\omega_1)$
$\vdots$	$\vdots$	$\vdots$
ω_k	2	$2r_k^2 \ = 2\|a_k\|^2 = 2I(\omega_k)$

Frequency	Degrees of freedom	Sum of squares decomposition
$\vdots$	$\vdots$	$\vdots$
$\omega_{n/2} = \pi$ (excluded if n is odd)	1	$a_{n/2}^2 = I(\pi)$
Total	**n**	$\sum_{\mathbf{t=1}}^{\mathbf{n}} \mathbf{x_t^2}$

Source: DE ANTONIO, D., and PALATE, J. (2015).

Obviously, if series were random then each component $I(\omega_j)$ would have the same expectation. On the contrary, when the series contains a systematic sine component having a frequency j and amplitude A then the sum of squares $I(\omega_j)$ increases with A. In practice, it is unlikely that the frequency j of an unknown systematic sine component would exactly match any of the frequencies, for which periodogram have been calculated. Therefore, the periodogram would show an increase in intensities in the immediate vicinity of j.[11]

Note that in JDemetra+ the periodogram object corresponds exactly to the contribution to the sum of squares of the standardised data, since the series are divided by their standard deviation for computational reasons.

Using the decomposition presented in table above the periodogram can be expressed as:

$$I(\omega_j)=r_j^2 = \frac{1}{2}(\alpha_j^2 + \beta_j^2) = \ \frac{1}{n}(\sum_{t=1}^{n} x_t \cos{(t\frac{2\pi j}{n}\ )}\ )^2 + \frac{1}{n}(\sum_{t=1}^{n} x_t \sin(t\frac{2\pi j}{n})\ )^2 \tag{0.6}$$

where $j = 0, \ldots, lbrack\frac{n}{2}rbrack$.

Since $\mathbf{x}-\overline{\mathbf{x}}$ are generated by an orthonormal basis, and $\overline{\mathbf{x}}=a_0\mathbf{e}_0$ Equation **??** can be rearranged to show that the sum of squares is equal to the sum of the squared coefficients:

$$\mathbf{x} - a_0\mathbf{e}_0 = \sum_{j=1}^{[(n-1)/2]} (\alpha_j\mathbf{c}_j + \beta_j\mathbf{s}_j) + a_{n/2}\mathbf{e}_{n/2} \tag{0.7}$$

Thus the sample variance of x_t can be expressed as:

$$n^{-1}\sum_{t=1}^{n}(x_t - \overline{x})^2 = n^{-1}(\sum_{k=1}^{[(n-1)/2]} 2{r_j}^2 + {a_{n/2}}^2) \tag{0.8}$$

where $a_{n/2}^2$ is excluded if n is odd.

[11] BOX, G.E.P., JENKINS, G.M., and REINSEL, G.C. (2007).

The term $2{r_j}^2$ in Equation **??** is then the contribution of the j^{th} harmonic to the variance and Equation **??** shows then how the total variance is partitioned.

The periodogram ordinate $I(\omega_j)$ and the autocovariance coefficient $\gamma(k)$ are both quadratic forms of x_t. It can be shown that the periodogram and autocovarinace function are related and the periodogram can be written in terms of the sample autocovariance function for any non-zero Fourier frequency ω_j :[12]

$$I(\omega_j) = \sum_{|k|<n} \hat{\gamma}(k)\ e^{-ik\omega_j} = \hat{\gamma}(0)\ \ + 2\sum_{k=1}^{n-1} \hat{\gamma}(k) \cos{(k\omega_j)}$$

and for the zero frequency $I(0) = n|\overline{x}|^2$.

Once comparing Equation **??** with an expression for the spectral density of a stationary process:

$$f(\omega\ ) = \frac{1}{2\pi} \sum_{k<-\infty}^{\infty} \gamma(k)\ e^{-ik\omega}\ \ = \frac{1}{2\pi} lbrack \gamma(0)\ \ + 2(\sum_{k=1}^{\infty} \gamma(k) \cos{(k\omega\ )}) rbrack$$

It can be noticed that the periodogram is a sample analogue of the population spectrum. In fact, it can be shown that the periodogram is asymptotically unbiased but inconsistent estimator of the population spectrum $f(\omega)$.[^75] Therefore, the periodogram is a wildly fluctuating, with high variance, estimate of the spectrum. However, the consistent estimator can be achieved by applying the different linear smoothing filters to the periodogram, called lag-window estimators. The lag-window estimators implemented in JDemetra+ includes square, Welch, Tukey, Barlett, Hanning and Parzen. They are described in DE ANTONIO, D., and PALATE, J. (2015). Alternatively, the model-based consistent estimation procedure, resulting in autoregressive spectrum estimator, can be applied.

Method 2: Autoregressive spectrum estimation

BROCKWELL, P.J., and DAVIS, R.A. (2006) point out that for any real-valued stationary process (x_t) with continuous spectral density $f(\omega)$ it is possible to find both $AR(p)$ and $MA(q)$ processes which spectral densities are arbitrarily close to $f(\omega)$. For this reason, in some sense, (x_t) can be approximated by either $AR(p)$ or $MA(q)$ process. This fact is a basis of one of the methods of achieving a consistent estimator of the spectrum, which is called an autoregressive spectrum estimation. It is based on the approximation of the stochastic process (x_t) by an autoregressive process of sufficiently high order p:

$$x_t = \mu + (\phi_1 B + ... + \phi_p B^p) x_t + \varepsilon_t$$

[12]The proof is given in BROCKWELL, P.J., and DAVIS, R.A. (2006).

where ε_t is a white-noise variable with mean zero and a constant variance.

The autoregressive spectrum estimator for the series x_t is defined as: [13]

$$\hat{s}(\omega) = 10 \times \log_{10} \frac{\sigma_x^2}{2\pi \left|1 - \sum_{k=1}^{p} \hat{\phi}_k e^{-ik\omega}\right|^2}$$

where:

- ω– frequency, $0 \leq \omega \leq \pi$;
- σ_x^2 – the innovation variance of the sample residuals;
- $\hat{\phi}_k$ – AR(k) coefficient estimates of the linear regression of $x_t - \overline{x}$ on $x_{t-k} - \overline{x}$, $1 \leq k \leq p$.

The autoregressive spectrum estimator is used in the visual spectral analysis tool for detecting significant peaks in the spectrum. The criterion of *visual significance*, implemented in JDemetra+, is based on the range $\hat{s}^{\max} - \hat{s}^{\min}$ of the $\hat{s}(\omega)$ values, where $\hat{s}^{\max} = \max_k \hat{s}(\omega_k)$; $\hat{s}^{\min} = \min_k \hat{s}(\omega_k)$; and $\hat{s}(\omega_k)$ is k^{th} value of autoregressive spectrum estimator.

The particular value is considered to be visually significant if, at a trading day or at a seasonal frequency ω_k (other than the seasonal frequency $\omega_{60} = \pi$), $\hat{s}(\omega_k)$ is above the median of the plotted values of $\hat{s}(\omega_k)$ and is larger than both neighbouring values $\hat{s}(\omega_{k-1})$ and $\hat{s}(\omega_{k+1})$ by at least $\frac{6}{52}$ times the range $\hat{s}^{\max} - \hat{s}^{\min}$.

Following the suggestion of SOUKUP, R.J., and FINDLEY, D.F. (1999), JDemetra+ uses an autoregressive model spectral estimator of model order 30. This order yields high resolution of strong components, meaning peaks that are sharply defined in the plot of $\hat{s}(\omega)$ with 61 frequencies. The minimum number of observations needed to compute the spectrum is set to $n = 80$ for monthly data and to $n = 60$ for quarterly series while the maximum number of observations considered for the estimation is 121. Consequently, with these settings it is possible to identify up to 30 peaks in the plot of 61 frequencies. By choosing $\omega_k = \frac{\mathrm{k}}{60}$ for $k = 0, 1, ..., 60$ the density estimates are calculated at exact seasonal frequencies (1, 2, 3, 4, 5 and 6 cycles per year).

The model order can also be selected based on the AIC criterion (in practice it is much lower than 30). A lower order produces the smoother spectrum, but the contrast between the spectral amplitudes at the trading day frequencies and neighbouring frequencies is weaker, and therefore not as suitable for automatic detection.

SOUKUP, R.J., and FINDLEY, D.F. (1999) also explain that the periodogram can be used in the *visual significance* test as it has as good as those of the AR(30) spectrum abilities to detect trading day effect, but also has a greater false alarm rate[14].

[13] Definition from '*X-12-ARIMA Reference Manual*' (2011).

[14] The false alarm rate is defined as the fraction of the 50 replicates for which a visually significant spectral peak occurred at one of the trading day frequencies being considered in the designated output spectra (SOUKUP,

Method 3: Tukey spectrum

Identification of spectral peaks

Identification of seasonal peaks in a Tukey periodogram and in an autoregressive spectrum

In order to decide whether a series has a seasonal component that is predictable (stable) enough, these tests use visual criteria and formal tests for the periodogram. The periodogram is calculated using complete years, so that the set of Fourier frequencies contains exactly all seasonal frequencies[15].

The tests rely on two basic principles:

- The peaks associated with seasonal frequencies should be larger than the median spectrum for all frequencies and;
- The peaks should exceed the spectrum of the two adjacent values by more than a critical value.

  JDemetra+ performs this test on the original series. If these two requirements are met, the test results are displayed in green. The statistical significance of each of the seasonal peaks (i.e. frequencies $\frac{\pi}{6}$, $\frac{\pi}{3}$, $\frac{\pi}{2}$, $\frac{2\pi}{3}$ and $\frac{5\pi}{6}$ corresponding to 1, 2, 3, 4 and 5 cycles per year) is also displayed. The seasonal and trading days frequencies depends on the frequency of time series. They are shown in the table below. The symbol d denotes a default frequency and is described below the table.

The seasonal and trading day frequencies by time series frequency

Number of months per fullperiod	Seasonal frequency	Trading day frequency (radians)
12	$\frac{\pi}{6}$, $\frac{\pi}{3}$, $\frac{\pi}{2}$, $\frac{2\pi}{3}$, $\frac{5\pi}{6}$, π	d,2.714
4	$\frac{\pi}{2}$, π	d, 1.292, 1.850, 2.128
3	π	d
2	π	d

The calendar (trading day or working day) effects, related to the variation in the number of different days of the week per period, can induce periodic patterns in the data that can be similar to those resulting from pure seasonal effects. From the theoretical point of view, trading day variability is mainly due to the fact that the average number of days in the months or quarters is not equal to a multiple of 7 (the average number of days of a month in the year of 365.25 days is equal to $\frac{365.25}{12} = 30.4375 days$). This effect occurs $\frac{365.25}{12} \times \frac{1}{7} =$

R.J., and FINDLEY, D.F. (1999)).

[15]For definition of the periodogram and Fourier frequencies see section Spectral Analysis

4.3482$times per month$: one time for each one of the four complete weeks of each month, and a residual of 0.3482 cycles per month, i.e. $0.3482 \times 2\pi = 2.1878 radians$. This turns out to be a fundamental frequency for the effects associated with monthly data. In JDemetra+ the fundamental frequency corresponding to 0.3482 cycles per month is used in place of the closest frequency $\frac{k}{60}$. Thus, the quantity $\frac{\pi \times 42}{60}$ is replaced by $\omega_{42} = 0.3482 \times 2\pi = 2.1878$. The frequencies neighbouring ω_{42}, i.e. ω_{41} and ω_{43} are set to, respectively, $2.1865 - \frac{1}{60}$ and $2.1865 + \frac{1}{60}$.

The default frequencies (d) for calendar effect are: 2.188 (monthly series) and 0.280 (quarterly series). They are computed as:

$$\omega_{\text{ce}} = \frac{2\pi}{7}\left(n - 7 \times \left[\frac{n}{7}\right]\right) \tag{0.9}$$

where $n = \frac{365.25}{s}$, $s = 4$ for quarterly series and $s = 12$ for monthly series.

Other frequencies that correspond to trading day frequencies are: 2.714 (monthly series) and 1.292, 1.850, 2.128 (quarterly series).

In particular, the calendar frequency in monthly data (marked in red on the figure below) is very close to the seasonal frequency corresponding to 4 cycles per year $_{40} = \frac{2}{3}\pi = 2.0944$.

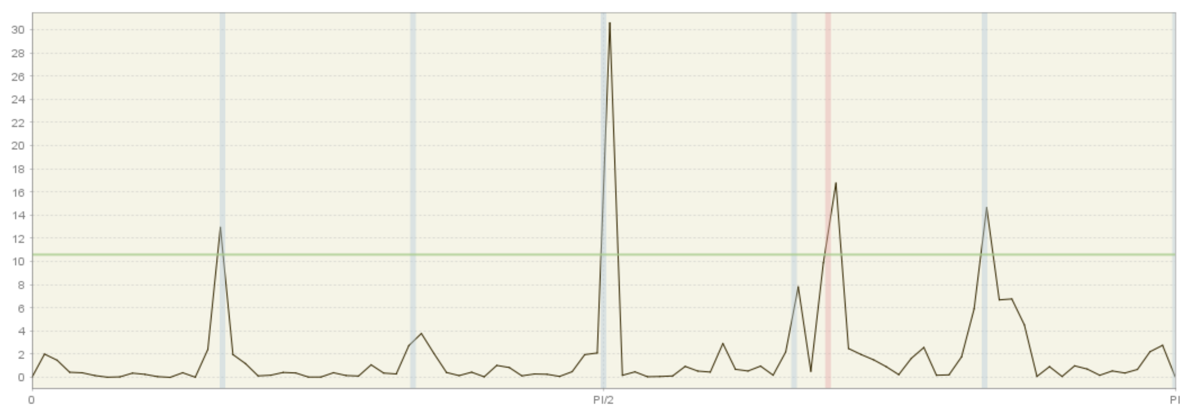


Figure 84: **Periodogram with seasonal (grey) and calendar (red) frequencies highlighted**

This implies that it may be hard to disentangle both effects using the frequency domain techniques.

comment3: end part theory>spectral analysis>identification of spectral peaks

in Tukey spectrum

comes from Identification of seasonal peaks in a Tukey spectrum

Tukey Spectrum definition

The Tukey spectrum belongs to the class of lag-window estimators. A lag window estimator of the spectral density $f(\omega) = \frac{1}{2\pi} \sum_{k<-\infty}^{\infty} \gamma(k) e^{ik\omega}$ is defined as follows:

$$\hat{f}_L(\omega) = \frac{1}{2\pi} \sum_{|h| \leq r} w(h/r) \hat{\gamma}(h) e^{ih\omega}$$

where $\hat{\gamma}(.)$ is the sample autocovariance function, $w(.)$ is the lag window, and r is the truncation lag. $|w(x)|$ is always less than or equal to one, $w(0) = 1$ and $w(x) = 0$ for $|x| > 1$. The simple idea behind this formula is to down-weight the autocovariance function for high lags where $\hat{\gamma}(h)$ is more unreliable. This estimator requires choosing r as a function of the sample size such that $r/n \to 0$ and $r \to \infty$ when $n \to \infty$. These conditions guarantee that the estimator converges to the true density.

JDemetra+ implements the so-called Blackman-Tukey (or Tukey-Hanning) estimator, which is given by $w(h/r) = 0.5(1 + cos(\pi h/r))$ if $|h/r| \leq 1$ and 0 otherwise.

The choice of large truncation lags r decreases the bias, of course, but it also increases the variance of the spectral estimate and decreases the bandwidth.

JDemetra+ allows the user to modify all the parameters of this estimator, including the window function.

Graphical Test

The current JDemetra+ implementation of the seasonality test is based on a $F(d_1, d_2)$ approximation that has been originally proposed by Maravall (2012) for TRAMO-SEATS. This test is has been designed for a Blackman-Tukey window based on a particular choices of the truncation lag r and sample size. Following this approach, we determine visually significant peaks for a frequency ω_j when

$$\frac{2 f_x(\omega_j)}{\left[f_x(\omega_{j+1}) + f_x(\omega_{j-1})\right]} \geq CV(\omega_j)$$

where $CV(\omega_j)$ is the critical value of a $F(d_1, d_2)$ distribution, where the degrees of freedom are determined using simulations. For $\omega_j = \pi$, we have a significant peak when $\frac{f_x(\omega_{[n/2]})}{\left[f_x(\omega_{[(n-1)/2]})\right]} \geq CV(\omega_j)$

Two significant levels for this test are considered: $\alpha = 0.05$ (code "t") and $\alpha = 0.01$ (code "T").

As opposed to the AR spectrum, which is computed on the basis of the last 120 data points, we will use here all available observations. Those critical values have been calculated given the recommended truncation lag $r = 79$ for a sample size within the interval $\in [80, 119]$ and $r = 112$ for $n \in [120, 300]$. The F approximation is less accurate for sample sizes larger than 300. For quarterly data, $r = 44$, but there are no recommendations regarding the required sample size.

Use

The test can be applied directly to any series by selecting the option *Statistical Methods » Seasonal Adjustment » Tools » Seasonality Tests*. This is an example of how results are displayed for the case of a monthly series:

4. Identification of seasonal peaks in a Tukey periodogram and in an auto-regressive spectrum

Seasonality present

T or t for Tukey periodogram, A or a for auto-regressive spectrum; 'T' or 'A' for very signficant peaks, 't' or 'a' for signficant peaks, '_' otherwise

AT.AT.AT.AT.AT.A-

Figure 85: tktest

JDemetra+ considers critical values for $\alpha = 1\%$ (code "T") and $\alpha = 5\%$ (code "t") at each one of the seasonal frequencies represented in the table below, e.g. frequencies $\frac{\pi}{6}$, $\frac{\pi}{3}$, $\frac{\pi}{2}$, $\frac{2\pi}{3}$ and $\frac{5\pi}{6}$ corresponding to 1, 2, 3, 4, 5 and 6 cycles per year in this example, since we are dealing with monthly data. The codes "a" and "A" correpond to the so-called AR spectrum, so ignore them for the moment.

The seasonal and trading day frequencies by time series frequency

Number of months per full period	Seasonal frequency	Trading day frequency (radians)
12	$\frac{\pi}{6}, \frac{\pi}{3}, \frac{\pi}{2}, \frac{2\pi}{3}, \frac{5\pi}{6}, \pi$	d, 2.714
6	$\frac{\pi}{3}, \frac{2\pi}{3}, \pi$	d
4	$\frac{\pi}{2}, \pi$	d, 1.292, 1.850, 2.128
3	π	d
2	π	d

Currently, only seasonal frequencies are tested, but the program allows you to manually plot the Tukey spectrum and focus your attention on both seasonal and trading day frequencies.

References

- Tukey, J. (1949). The sampling theory of power spectrum estimates., Proceedings Symposium on Applications of Autocorrelation Analysis to Physical Problems, NAVEXOS-P-735, Office of Naval Research, Washington, 47-69

in AR Spectrum definition

comes from: "Identification of seasonal peaks in autoregressive spectrum"

The estimator of the spectral density at frequency $\lambda \in [0, \pi]$ will be given by the assumption that the series will follow an AR(p) process with large p. The spectral density of such model, with an innovation variance $var(x_t) = \sigma_x^2$, is expressed as follows:

$$10 \times log_{10} f_x(\lambda) = 10 \times log_{10} \frac{\sigma_x^2}{2\pi \left|\phi(e^{i\lambda})\right|^2} = 10 \times log_{10} \frac{\sigma_x^2}{2\pi \left|1 - \sum_{k=1}^{p} \phi_k e^{ik\lambda})\right|^2}$$

where:

- ϕ_k denotes the AR(k) coefficient ;
- $e^{-ik\lambda} = cos\ (-ik\lambda) + isin\ (-ik\lambda)$.

Soukup and Findely (1999) suggest the use of p=30, which in practice much larger than the order that would result from the AIC criterion. The minimum number of observations needed to compute the spectrum is set to *n=80* for monthly data (or *n=60*) for quarterly series. In turn, the maximum number of observations considered for the estimation is *n=121*. This choice offers enough resolution, being able to identify a maximum of 30 peaks in a plot of 61 frequencies: by choosing $\lambda_j = \pi j/60$,for $j = 0, 1, ..., 60$, we are able to calculate our density estimates at exact seasonal frequencies (1, 2, 3, 4, 5 and 6 cycles per year). Note that x cycles per year can be converted into cycles per month by simply dividing by twelve, $x/12$, and to radians by applying the transformation $2\pi(x/12)$.

The traditional trading day frequency corresponding to 0.348 cycles per month is used in place of the closest frequency $\pi j/60$. Thus, we replace $\pi 42/60$ by $\lambda_{42} = 0.348 \times 2\pi = 2.1865$. The frequencies neighbouring λ_{42} are set to $\lambda_{41} = 2.1865 - 1/60$ and $\lambda_{43} = 2.1865 + 1/60$. The periodogram below illustrates the proximity of this trading day frequency λ_{42} (red shade) and the frequency corresponding to 4 cycles per year $\lambda_{40} = 2.0944$. This proximity is precisely what poses the identification problems: the AR spectrum boils down to a smoothed version of the periodogram and the contribution of the of the trading day frequency may be obscured by the leakage resulting from the potential seasonal peak at λ_{40}, and vice-versa.

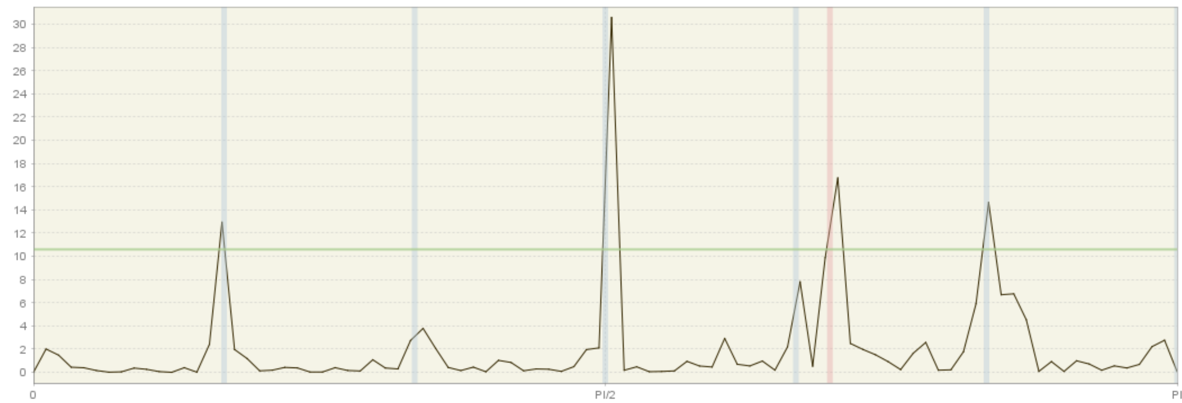


Figure 86: **Periodogram with seasonal (grey) and calendar (red) frequencies highlighted**

JDemetra+ allows the user to modify the number of lags of this estimator and to change the number of observations used to determine the AR parameters. These two options can improve the resolution of this estimator.

Graphical Test

The statistical significance of the peaks associated to a given frequency can be informally tested using a visual criterion, which has proved to perform well in simulation experiments. Visually significant peaks for a frequency λ_j satisfy both conditions:

- $\frac{f_x(\lambda_j)-\max\{f_x(\lambda_{j+1}),f_x(\lambda_{j-1})\}}{[\max_k f_x(\lambda_k)-\min_i f_x(\lambda_i)]} \geq CV(\lambda_j)$, where $CV(\lambda_j)$ can be set equal to 6/52 for all j
- $f_x(\lambda_j) > median_j\left\{f_x(\lambda_j)\right\}$, which guarantees $f_x(\lambda_j)$ it is not a local peak.

The first condition implies that if we divide the range $\max_k f_x(\lambda_k) - \min_i f_x(\lambda_i)$ in 52 parts (traditionally represented by stars) the height of each pick should be at least 6 stars.

Use

The test can be applied directly to any series by selecting the option *Statistical Methods » Seasonal Adjustment » Tools » Seasonality Tests*. This is an example of how results are displayed for the case of a monthly series:

JDemetra+ considers critical values for $\alpha = 1\%$ (code "A") and $\alpha = 5\%$ (code "a") at each one of the seasonal frequencies represented in the table below, e.g. frequencies $\frac{\pi}{6}$, $\frac{\pi}{3}$, $\frac{\pi}{2}$, $\frac{2\pi}{3}$ and $\frac{5\pi}{6}$ corresponding to 1, 2, 3, 4, 5 and 6 cycles per year in this example, since we are dealing with

4. Identification of seasonal peaks in a Tukey periodogram and in an auto-regressive spectrum

Seasonality present

T or t for Tukey periodogram, A or a for auto-regressive spectrum; 'T' or 'A' for very signficant peaks, 't' or 'a' for signficant peaks, '_' otherwise

AT.AT.AT.AT.AT.A-

Figure 87: artest

monthly data. The codes "t" and "T" correpond to the so-called Tukey spectrum, so ignore them for the moment.

The seasonal and trading day frequencies by time series frequency

Number of months per full period	Seasonal frequency	Trading day frequency (radians)
12	$\frac{\pi}{6}$, $\frac{\pi}{3}$, $\frac{\pi}{2}$, $\frac{2\pi}{3}$, $\frac{5\pi}{6}$, π	d, 2.714
6	$\frac{\pi}{3}$, $\frac{2\pi}{3}$, π	d
4	$\frac{\pi}{2}$, π	d, 1.292, 1.850, 2.128
3	π	d
2	π	d

Currently, only seasonal frequencies are tested, but the program allows you to manually plot the AR spectrum and focus your attention on both seasonal and trading day frequencies. Agustin Maravall has conducted a simulation experiment to calculate $CV(\lambda_{42})$ (trading day frequency) and proposes to set for all j equal to the critical value associated to the trading frequency, but this is currently not part of the current automatic testing procedure of JDemetra+.

References

- Soukup, R.J., and D.F. Findley (1999) On the Spectrum Diagnosis used by X12-ARIMA to Indicate the Presence of Trading Day Effects After Modeling or Adjustment. In Proceedengs of the American Statistical Association. Business and Economic Statistics Section, 144-149, Alexandria, VA.

in a Periodogram

comes from: Identification of seasonal peaks in periodogram

The periodogram $I(\omega_j)$ of $\mathbf{X} \in \mathbb{C}^n$ is defined as the squared of the Fourier transform

$$I(\omega_j) = a_j^2 = n^{-1} \left| \sum_{t=1}^{n} \mathbf{X_t} e^{-it\omega_j} \right|^2,$$

where the Fourier frequencies ω_j are given by multiples of the fundamental frequency $\frac{2\pi}{n}$:

$$\omega_j = \frac{2\pi j}{n}, -\pi < \omega_j \leq \pi$$

An orthonormal basis in $\mathbb{R}^n$:

$$\left\{ e_0, \quad c_1, s_1, \quad \dots \quad , \quad c_{[(n-1)/2]}, s_{[(n-1)/2]} \quad , \quad e_{n/2} \right\},$$

where $e_{n/2}$ is excluded if n is odd,

can be used to project the data and obtain the spectral decomposition

Thus, the periodogram is given by the projection coefficients and represents the contribution of the jth harmonic to the total sum of squares, as illustrated by Brockwell and Davis (1991):

Source	Degrees of freedom
Frequency ω_0	1
Frequency ω_1	2
$\vdots$	$\vdots$
Frequency ω_k	2
$\vdots$	$\vdots$
Frequency $\omega_{n/2} = \pi$ (excluded if n is odd)	1
=========	======
Total	n

In JDemetra+, the periodogram of $\mathbf{X} \in \mathbb{R}^n$ is computed for the standardized time series.

Defining a F-test

Brockwell and Davis (1991, section 10.2) exploit the fact that the periodogram can be expressed as the projection on the orthonormal basis defined above to derive a test. Thus, under the null hypothesis:

- $2I(\omega_k) = \|P_{\bar{sp}_{\{c_k,s_k\}}}\mathbf{X}\|^2 \sim \sigma^2\chi^2(2)$, for Fourier frequencies $0 < \omega_k = 2\pi k/n < \pi$
- $I(\pi) = \|P_{\bar{sp}_{\left\{e_{n/2}\right\}}}\mathbf{X}\|^2 \sim \sigma^2\chi^2(1)$, for π

Because $I(\omega_k)$ is independent from the projection error sum of squares, we can define our F-test statistic as follows:

- $\frac{2I(\omega_k)}{\|\mathbf{X}-P_{\bar{sp}_{\{e_0,c_k,s_k\}}}\mathbf{X}\|^2}\frac{n-3}{2} \sim F(2, n-3)$, for Fourier frequencies $0 < \omega_k = 2\pi k/n < \pi$
- $\frac{I(\pi)}{\|\mathbf{X}-P_{\bar{sp}_{\left\{e_0,e_{n/2}\right\}}}\mathbf{X}\|^2}\frac{n-2}{1} \sim F(1, n-2)$, for π

where $\|\mathbf{X} - P_{\bar{sp}_{\{e_0,c_k,s_k\}}}\mathbf{X}\|^2 = \sum_{i=1}^{n}\mathbf{X_i^2} - I(0) - 2I(\omega_k) \sim \sigma^2\chi^2(n-3)$ for Fourier frequencies $0 < \omega_k = 2\pi k/n < \pi$ - $\|\mathbf{X} - P_{\bar{sp}_{\left\{e_0,e_{n/2}\right\}}}\mathbf{X}\|^2 = \sum_{i=1}^{n}\mathbf{X_i^2} - I(0) - I(\pi) \sim \sigma^2\chi^2(n-2)$ for π

Thus, we reject the null if our F-test statistic computed at a given seasonal frequency (different from π) is larger than $F_{1-\alpha}(2, n-3)$. If we consider π, our test statistic follows a $F_{1-\alpha}(1, n-2)$ distribution.

Seasonality test

The implementation of JDemetra+ considers simultaneously the whole set of seasonal frequencies (1, 2, 3, 4, 5 and 6 cycles per year). Thus, the resulting test-statistic is:

$$\frac{2I(\pi/6) + 2I(\pi/3) + 2I(2\pi/3) + 2I(5\pi/6) + \delta I(\pi)}{\left\|\mathbf{X} - P_{\bar{sp}_{\left\{e_0,c_1,s_1,c_2,s_2,c_3,s_3,c_4,s_4,c_5,s_5,\delta e_{n/2}\right\}}}\mathbf{X}\right\|^2}\frac{n-12}{11} \sim F(11-\delta, n-12+\delta)$$

where $\delta = 1$ if n is even and 0 otherwise.

In small samples, the test performs better when the periodogram is evaluated as the exact seasonal frequencies. JDemetra+ modifies the sample size to ensure the seasonal frequencies belong to the set of Fourier frequencies. This strategy provides a very simple and effective way to eliminate the leakage problem.

Example of how results are displayed:

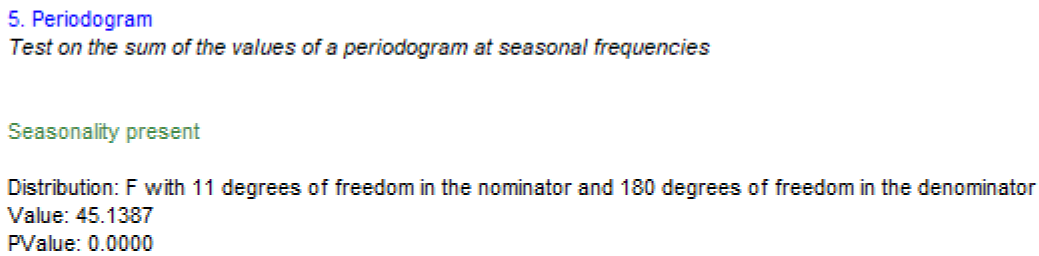


Figure 88: periodtest

References

Brockwell, P.J., and R.A. Davis (1991). Times Series: Theory and Methods. Springer Series in Statistics.

Spectral graphs

probably moove this part to GUI (Tools), just leave a link

comment3: start part case studies > spectral graphs

This scenario is designed for advanced users interested in an in-depth analysis of time series in the frequency domain using three spectral graphs. Those graphs can also be used as a complementary analysis for a better understanding of the results obtained with some of the tests described above.

Economic time series are usually presented in a time domain (X-axis). However, for analytical purposes it is convenient to convert the series to a frequency domain due to the fact that any stationary time series can be expressed as a combination of cosine (or sine) functions. These functions are characterized with different periods (amount of time to complete a full cycle) and amplitudes (maximum/minimum value during the cycle).

The tool used for the analysis of a time series in a frequency domain is called a spectrum. The peaks in the spectrum indicate the presence of cyclical movements with periodicity between two months and one year. A seasonal series should have peaks at the seasonal frequencies. Calendar adjusted data are not expected to have peak at with a calendar frequency.

The periodicity of the phenomenon at frequency f is $\frac{2\pi}{f}$. It means that for a monthly time series the seasonal frequencies $\frac{\pi}{6}$, $\frac{\pi}{3}$, $\frac{\pi}{2}$, $\frac{2\pi}{3}$, $\frac{5\pi}{6}$ and π correspond to 1, 2, 3, 4, 5 and 6 cycles per year. For example, the frequency $\frac{\pi}{3}$ corresponds to a periodicity of 6 months (2 cycles per year are completed). For the quarterly series there are two seasonal frequencies: $\frac{\pi}{2}$ (one cycle per year) and π (two cycles per year). A peak at the zero frequency always corresponds to the trend component of the series. Seasonal frequencies are marked as grey vertical lines, while violet vertical lines represent the trading-days frequencies. The trading day frequency is 0.348 and derives from the fact that a daily component which repeats every seven days goes through 4.348 cycles in a month of average length 30.4375 days. It is therefore seen to advance 0.348 cycles per month when the data are obtained at twelve equally spaced times in 365.25 days (the average length of a year).

The interpretation of the spectral graph is rather straightforward. When the values of a spectral graph for low frequencies (i.e. one year and more) are large in relation to its other values it means that the long-term movements dominate in the series. When the values of a spectral graph for high frequencies (i.e. below one year) are large in relation to its other values it means that the series are rather trendless and contains a lot of noise. When the values of a spectral graph are distributed randomly around a constant without any visible peaks, then it is highly probable that the series is a random process. The presence of seasonality in a time series is manifested in a spectral graph by the peaks on the seasonal frequencies.

Spectral graphs in GUI

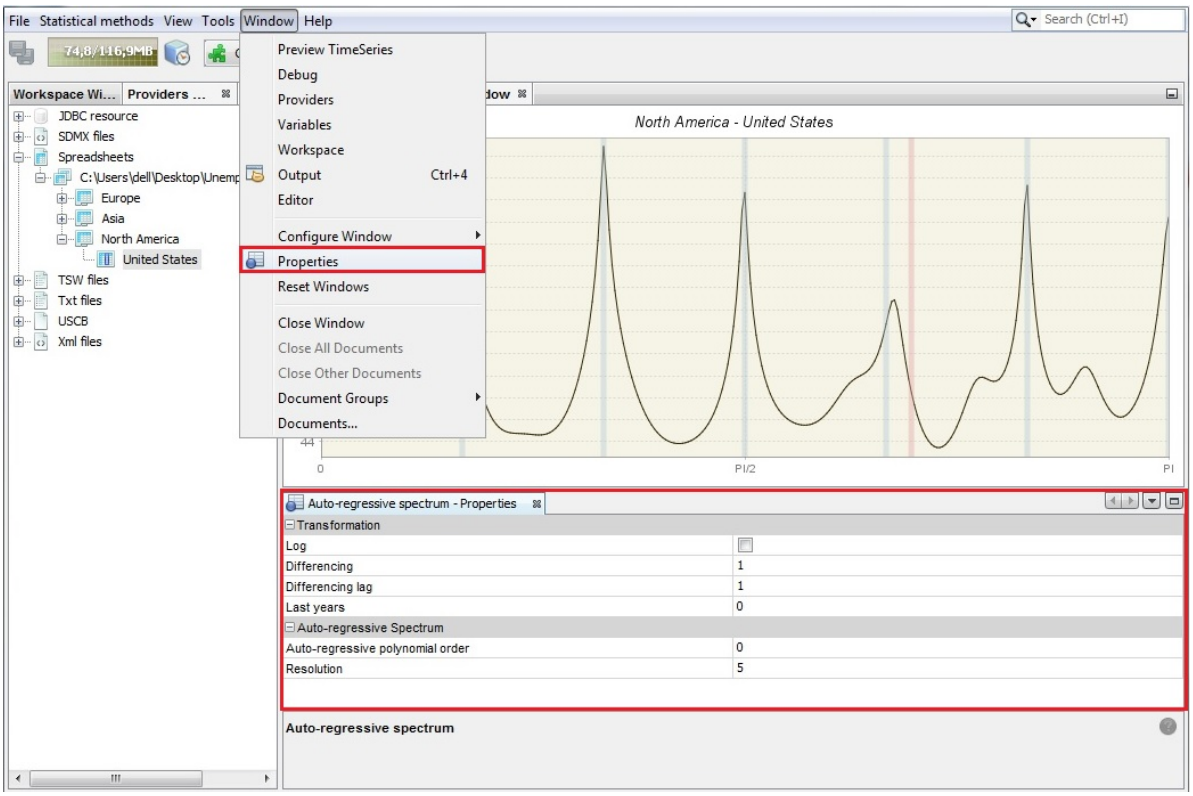


Figure 89: **Auto-regressive spectrum's properties**

1. The spectral graphs are available from: *Tools* → *Spectral analysis.*

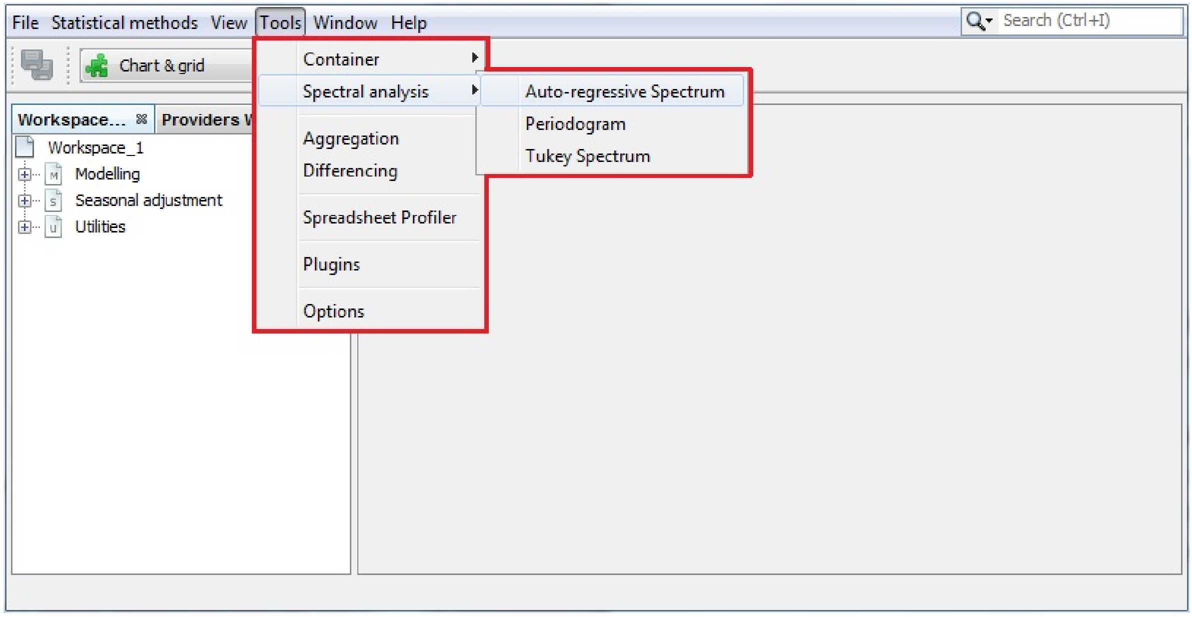


Figure 90: Tools for spectral analysis

2. When the first option is chosen JDemetra+ displays an empty *Auto-regressive spectrum* window. To start an analysis drag a single time series from the *Providers* window and drop it into the *Drop data here* area.

3. An auto-regressive spectrum graph available in JDemetra+ is based on the relevant tool from the X-13ARIMA-SEATS program. It shows the spectral density (spectrum) function, which reformulates the content of the stationary time series' autocovariances in terms of amplitudes at frequencies of half a cycle per month or less. The number of observations, data transformations and other options such as the specification of the frequency grid and the order of the autoregressive polynomial (30 by default) can be specified by opening the *Window* → *Properties* from the main menu.

   The *Auto-regressive - Properties* window contains the following options:

   - **Log** - a log transformation of a time series;
   - **Differencing** - transforms a data by calculating a regular (order 1,2..) or seasonal (order 4, 12, depending on the time series frequency) differences;
   - **Differencing lag** - the number of lags that the program will use to take differences. For example, if *Differencing lag = 3* then the differencing filter does not apply to the first lag (default) but to the third lag.

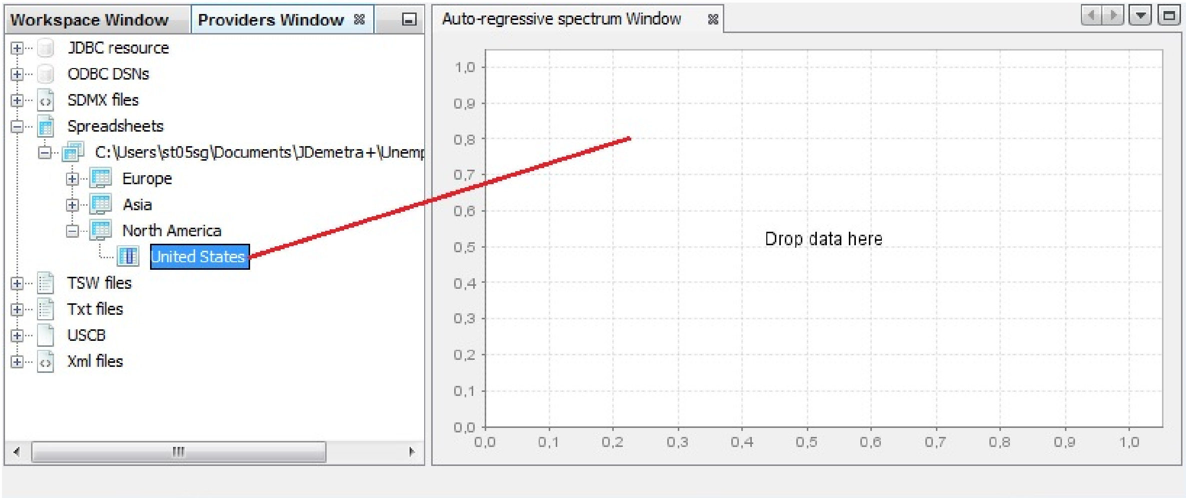


Figure 91: Launching an auto-regressive spectrum

- **Last years** - a number of years at the end of the time series taken to produce autoregresive spectrum. By default, it is 0, which means that the whole time series is considered.
- **Auto-regressive polynomial order** - the number of lags in the AR model that is used to estimate the spectral density. By default, the order of the autoregressive polynomial is set to 30 lags.
- **Resolution** - the value 1 plots the spectral density estimate for the frequencies $\omega_j = \frac{2\pi j}{n}$, where $n \in (-\pi; \pi)$ is the size of the sample used to estimate the AR model. Increasing this value, which is set to 5 by default, will increase the precision of this grid.

4. The seasonality test described above uses an empirical criterion to check whether the series has a seasonal component that is predictable (stable) enough that it can be estimated with reasonable success. The peak in the auto-regressive spectrum has to be greater than the median of the 61 spectrum ordinates and has to exceed the two adjacent spectral values by more than a critical value. When such a case is detected, the test results are displayed in green.
5. The second spectral graph is a periodogram. To perform the analysis of a single time series using this tool, choose *Tools* →*Spectral analysis* → *Periodogram* and drag and drop a series from the *Providers* window to the empty *Periodogram* window.
6. The sample size and data transformations can be specified by opening the *Window* → *Properties*, in the main menu. The *Periodogram* - *Properties* window contains the following options:

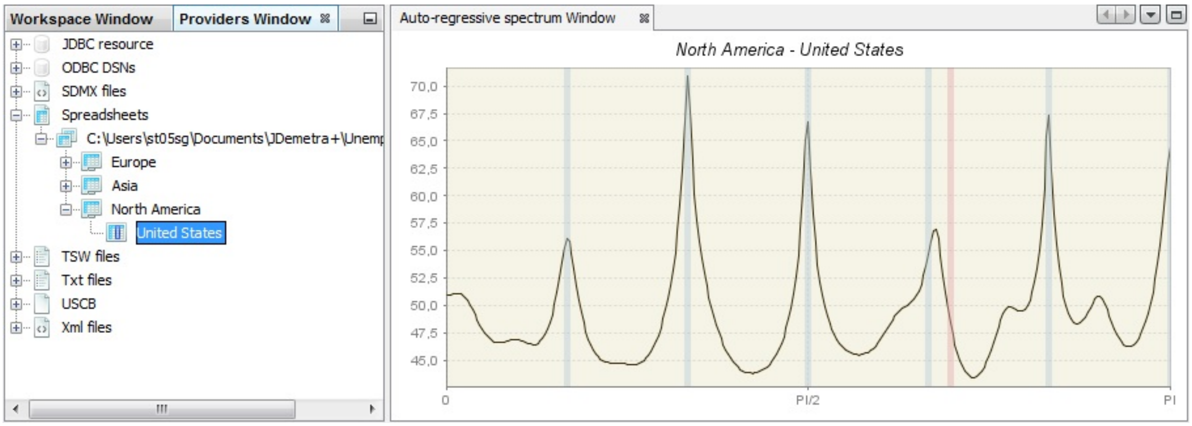


Figure 92: An example of an-auto-regressive spectrum

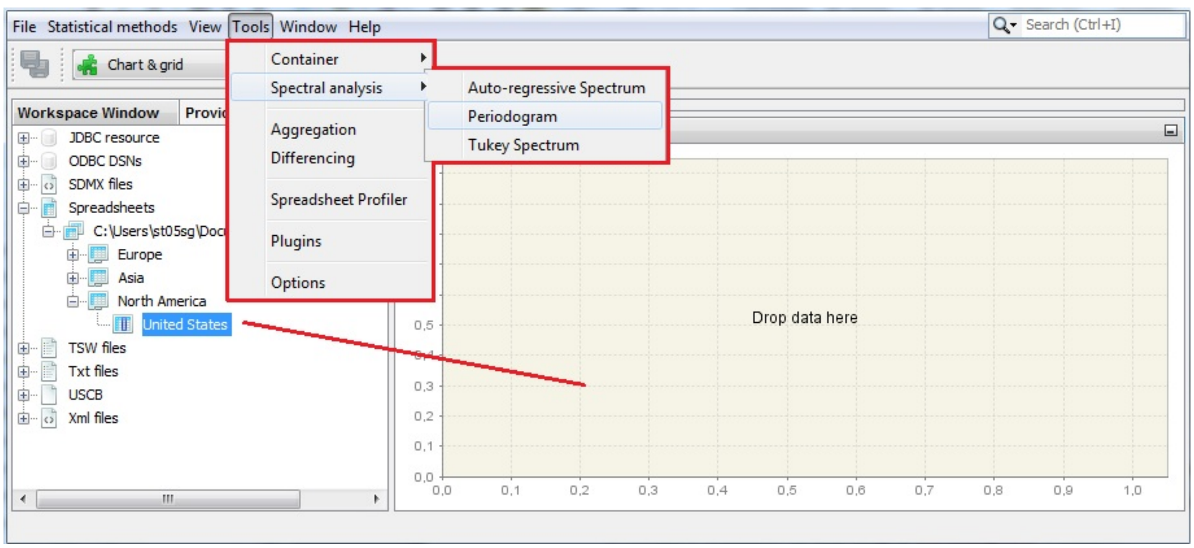


Figure 93: Launching a periodogram

- **Log** - a log transformation of a time series;
- **Differencing** - transforms the data by calculating regular (order 1,2..) or seasonal (order 4, 12, depending on the time series frequency) differences;
- **Differencing lag** - the number of lags that you will use to take differences. For example, if *Differencing lag = 3* then the differencing filter does not apply to the first lag (default) but to the third lag.
- **Last years** - the number of years at the end of the time series taken to produce periodogram. By default it is 0, which means that the whole time series is considered.

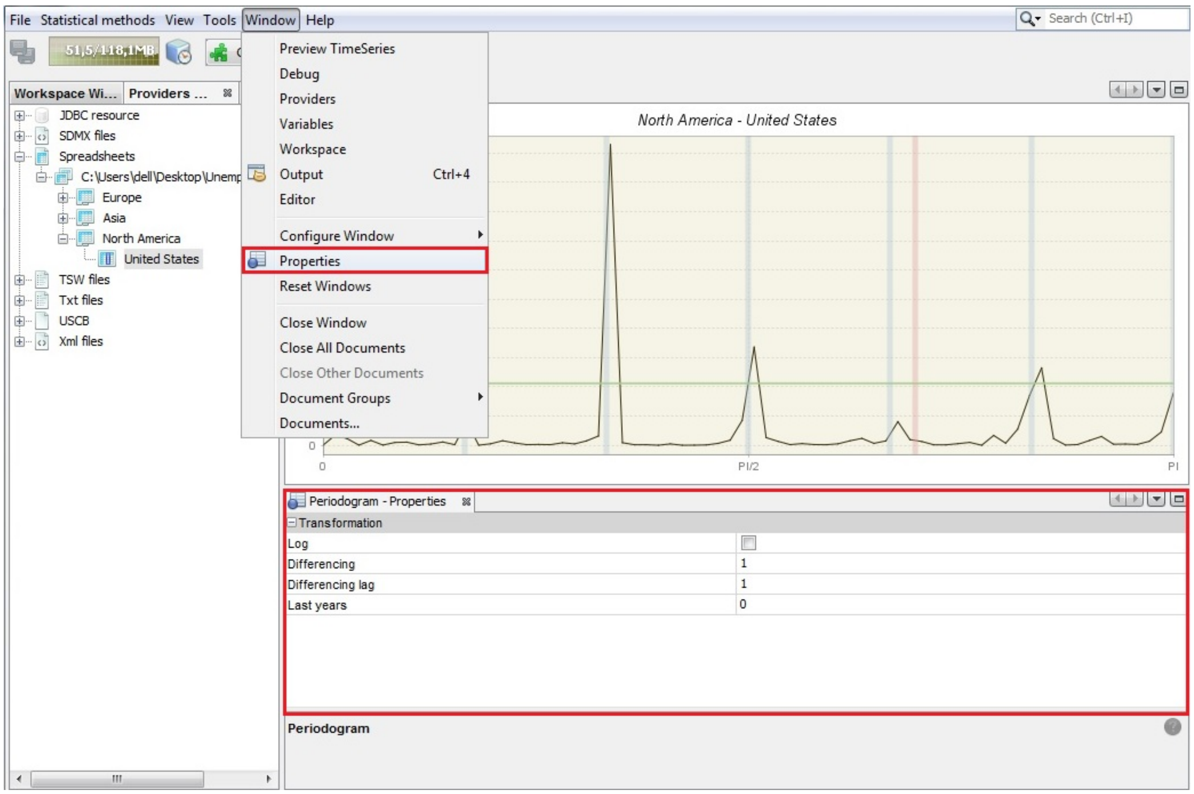


Figure 94: Periodogram's properties

7. The periodogram was one of the earliest tools used for the analysis of time series in the frequency domain. It enables the user to identify the dominant periods (or frequencies) of a time series. In general, the periodogram is a wildly fluctuating estimate of the spectrum with a high variance and is less stable than an auto-regressive spectrum.
8. The third spectral graph is the Tukey spectrum. To perform the analysis of time series using this tool, choose *Tools* → *Spectral analysis* → *Tukey spectrum* and drag and drop

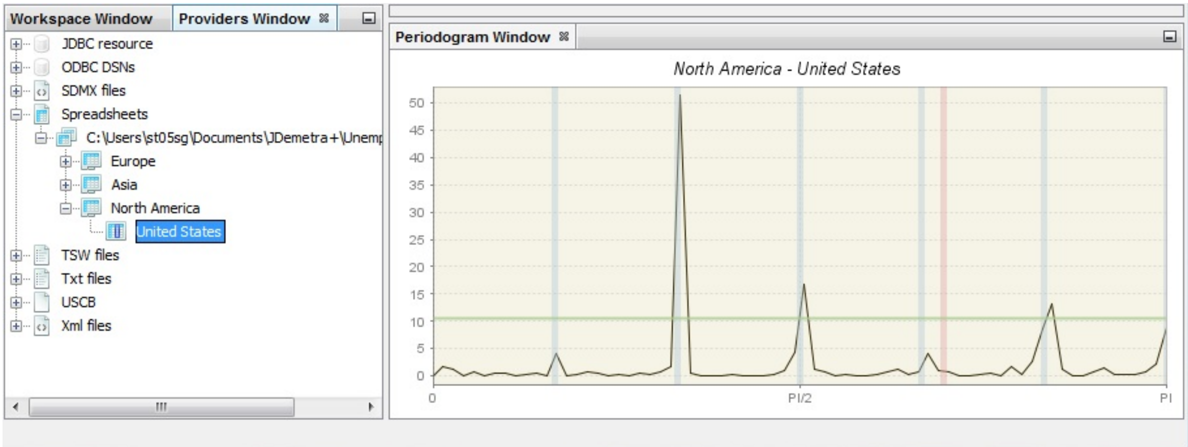


Figure 95: An example of a periodogram

a single series from the *Providers* window to the empty *Periodogram* window.

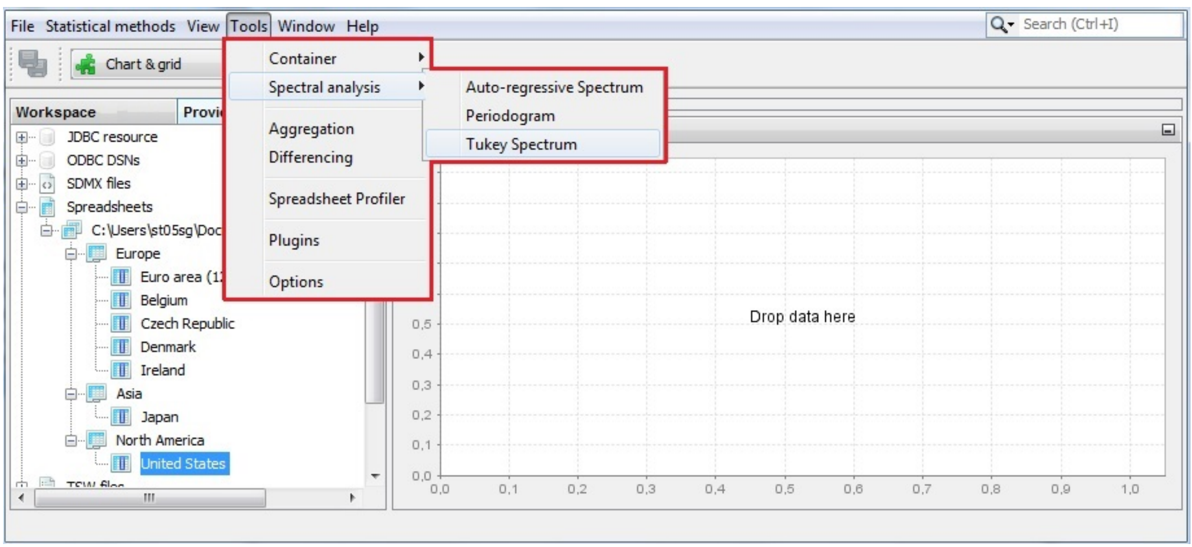


Figure 96: Launching a Tukey spectrum

9. The Tukey spectrum estimates the spectral density by smoothing the periodogram.
10. The options for the Tuckey window can be specified by opening the *Window* $\rightarrow$ *Properties* from the main menu. The *Periodogram - Properties* window contains the following options:
    - **Log** - a log transformation of a time series.

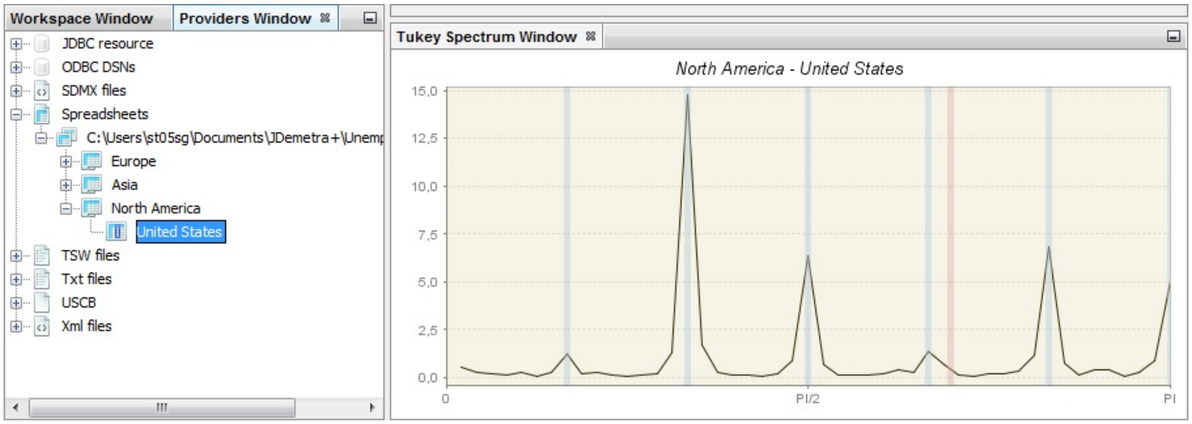


Figure 97: An example of a Tukey spectrum

- **Differencing** - transforms the data by calculating regular (order 1, 2..) or seasonal (order 4, 12, depending on the time series frequency) differences.
- **Differencing lag** - the number of lags that you will use to take differences. For example, if *Differencing lag = 3* then the differencing filter does not apply to the first lag (default) but to the third lag.
- **Taper part** – parameter larger than 0 and smaller or equal to one that shapes the curvature of the smoothing function that is applied to the auto-covariance function.
- **Window length** – the size of the window that is used to smooth the auto-covariance function. A value of zero includes the whole series.
- **Window type** – it refers to the weighting scheme that it is used to smooth the auto-covariance function. The available windows types (*Square*, *Welch*, *Tukey*, *Barlett*, *Hamming*, *Parzen*) are suitable to estimate the spectral density.

comment3: end part case studies > spectral graphs

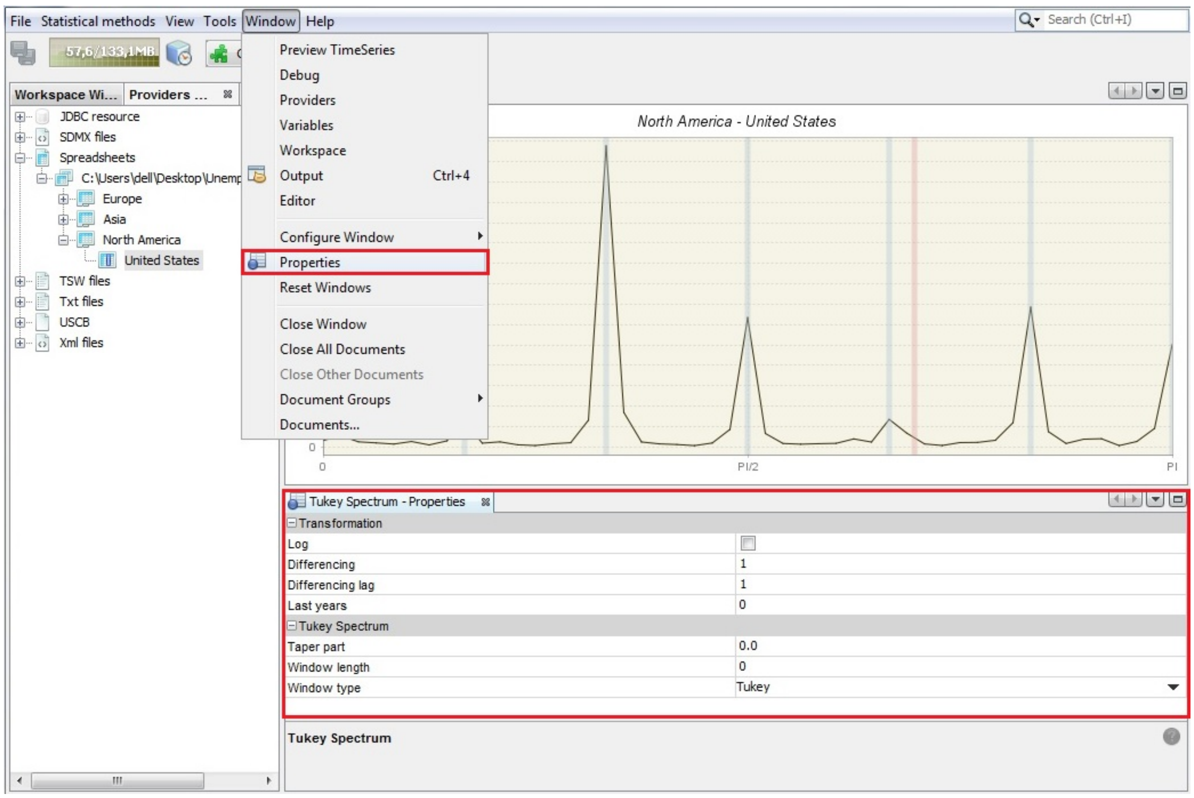


Figure 98: Tukey spectrum's properties

Reg-Arima models

Under construction.

Overview

The primary aim of seasonal adjustment is to remove the unobservable seasonal component from the observed series. The decomposition routines implemented in the seasonal adjustment methods make specific assumptions concerning the input series. One of the crucial assumptions is that the input series is stochastic, i.e. it is clean of deterministic effects. Another important limitation derives from the symmetric linear filter used in TRAMO-SEATS and X-13ARIMA-SEATS. A symmetric linear filter cannot be applied to the first and last observations with the same set of weights as for the central observations[^1]. Therefore, for the most recent observations these filters provide estimates that are subject to revisions.

To overcome these constrains both seasonal adjustment methods discussed here include a modelling step that aims to analyse the time series development and provide a better input for decomposition purposes. The tool that is frequently used for this purpose is the ARIMA model, as discussed by BOX, G.E.P., and JENKINS, G.M. (1970). However, time series are often affected by the outliers, other deterministic effects and missing observations. The presence of these effects is not in line with the ARIMA model assumptions. The presence of outliers and other deterministic effects impede the identification of an optimal ARIMA model due to the important bias in the estimation of parameters of sample autocorrelation functions (both global and partial)[^3]. Therefore, the original series need to be corrected for any deterministic effects and missing observations. This process is called linearisation and results in the stochastic series that can be modelled by ARIMA.

For this purpose both TRAMO and Reg-ARIMA use regression models with ARIMA errors. With these models TRAMO and Reg-ARIMA also produce forecasts.

X-11 decomposition

Introduction

A complete documentation of the X-11 method is available in LADIRAY, D., and QUENNEVILLE, B. (2001). The X-11 program is the result of a long tradition of non-parametric smoothing based on moving averages, which are weighted averages of a moving span of a time series (see hereafter). Moving averages have two important drawbacks:

- They are not resistant and might be deeply impacted by outliers;
- The smoothing of the ends of the series cannot be done except with asymmetric moving averages which introduce phase-shifts and delays in the detection of turning points.

These drawbacks adversely affect the X-11 output and stimulate the development of this method. To overcome these flaws first the series are modelled with a RegARIMA model that calculates forecasts and estimates the regression effects. Therefore, the seasonal adjustment process is divided into two parts.

- In a first step, the RegARIMA model is used to clean the series from non-linearities, mainly outliers and calendar effects. A global ARIMA model is adjusted to the series in order to compute the forecasts.
- In a second step, an enhanced version of the X-11 algorithm is used to compute the trend, the seasonal component and the irregular component.

Moving averages

The moving average of coefficient θ_i is defined as:

$$M\left(X_t\right) = \sum_{k=-p}^{+f} \theta_k X_{t+k}$$

The value at time t of the series is therefore replaced by a weighted average of p "past" values of the series, the current value, and f "future" values of the series. The quantity $p+f+1$ is called the moving average order. When p is equal to f, that is, when the number of points in

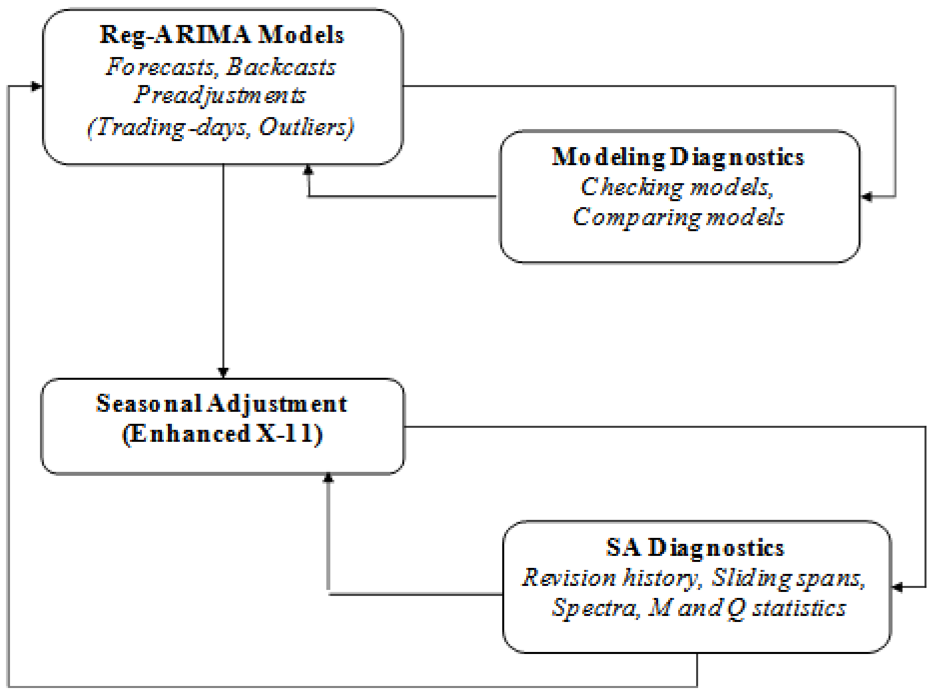


Figure 99: **The flow diagram for seasonal adjustment with X-13ARIMA-SEATS using the X-11 algorithm.**

the past is the same as the number of points in the future, the moving average is said to be centred. If, in addition, $\theta_{-k} = \theta_k$ for any k, the moving average M is said to be symmetric. One of the simplest moving averages is the symmetric moving average of order $P = 2p + 1$ where all the weights are equal to $\frac{1}{P}$.

This moving average formula works well for all time series observations, except for the first p values and last f values. Generally, with a moving average of order $p + f + 1$ calculated for instant t nwith points p in the past and points f in the future, it will be impossible to smooth out the first p values and the last f values of the series because of lack of input to the moving average formula.

In the X-11 method, symmetric moving averages play an important role as they do not introduce any phase-shift in the smoothed series. But, to avoid losing information at the series ends, they are either supplemented by *ad hoc* asymmetric moving averages or applied on the series extended by forecasts.

For the estimation of the seasonal component, X-13ARIMA-SEATS uses $P \times Q$ composite moving averages, obtained by composing a simple moving average of order P, which coefficients are all equal to $\frac{1}{P}$, and a simple moving average of order Q, which coefficients are all equal to $\frac{1}{Q}$.

The composite moving averages are widely used by the X-11 method. For an initial estimation of trend X-11 method uses a 2×4 moving average in case of a quarterly time series while for a monthly time series a 2×12 moving average is applied. The 2×4 moving average is an average of order 5 with coefficients $\frac{1}{8}\{1, 2, 2, 2, 1\}$. It eliminates frequency $\frac{\pi}{2}$ corresponding to period 4 and therefore it is suitable for seasonal adjustment of the quarterly series with a constant seasonality. The 2×12 moving average, with coefficients $\frac{1}{24}\{1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1\}$ that retains linear trends, eliminates order-12 constant seasonality and minimises the variance of the irregular component. The 2×4 and 2×12 moving averages are also used in the X-11 method to normalise the seasonal factors. The composite moving averages are also used to extract the seasonal component. These, which are used in the purely automatic run of the X-11 method (without any intervention from the user) are 3×3, 3×5 and 3×9.

In the estimation of the trend also Henderson moving averages are used. These filters have been chosen for their smoothing properties. The coefficients of a Henderson moving average of order $2p + 1$ may be calculated using the formula:

$$\theta_i = \frac{315\left[(n-1)^2 - i^2\right]\left[n^2 - i^2\right]\left[(n+1)^2 - i^2\right]\left[3n^2 - 16 - 11i^2\right]}{8n\left(n^2-1\right)\left(4n^2-1\right)\left(4n^2-9\right)\left(4n^2-25\right)}$$

where $n = p + 2$.

The basic algorithm of the X-11 method

The X-11 method is based on an iterative principle of estimation of the different components using appropriate moving averages at each step of the algorithm. The successive results are saved in tables. The list of the X-11 tables displayed in JDemetra+ is included at the end of this section.

The basic algorithm of the X-11 method will be presented for a monthly time series X_t that is assumed to be decomposable into trend, seasonality and irregular component according to an additive model $X_t = TC_t + S_t + I_t$.

A simple seasonal adjustment algorithm can be thought of in eight steps. The steps presented below are designed for the monthly time series. In the algorithm that is run for the quarterly time series the 2×4 moving average instead of the 2×12 moving average is used.

Step 1: Estimation of Trend by $\mathbf{2 \times 12}$ moving average:

$$TC_t^{(1)} = M_{2\times12}(X_t)$$

Step 2: Estimation of the Seasonal-Irregular component:

$$(S_t + I_t)^{(1)} = X_t - \mathrm{TC}_t^{(1)}$$

Step 3: Estimation of the Seasonal component by

$$\mathbf{3 \times 3}$$

moving average over each month:

$$S_t^{(1)} - M_{3\times3}\left[(S_t + I_t)^{(1)}\right]$$

The moving average used here is a 3×3 moving average over 5 terms, with coefficients $\frac{1}{9}\{1, 2, 3, 2, 1\}$. The seasonal component is then centred using a 2×12 moving average.

$$\widetilde{S}_t^{(1)} = S_t^{(1)} - M_{2\times12}\left(S_t^{(1)}\right)$$

Step 4: Estimation of the seasonally adjusted series:

$$SA_t^{(1)} = (\mathrm{TC}_t + I_t)^{(1)} = X_t - \widetilde{S}_t^{(1)}$$

This first estimation of the seasonally adjusted series must, by construction, contain less seasonality. The X-11 method again executes the algorithm presented above, changing the moving averages to take this property into account.

Step 5: Estimation of Trend by 13-term Henderson moving average:

$$TC_t^{(2)} = H_{13}\left(\mathrm{SA}_t^{(1)}\right)$$

Henderson moving averages, while they do not have special properties in terms of eliminating seasonality (limited or none at this stage), have a very good smoothing power and retain a local polynomial trend of degree 2 and preserve a local polynomial trend of degree 3.

Step 6: Estimation of the Seasonal-Irregular component:

$$(S_t + I_t)^{(2)} = X_t - \mathrm{TC}_t^{(2)}$$

Step 7: Estimation of the Seasonal component by $\mathbf{3} \times \mathbf{5}$ moving average over each month:

$$S_t^{(2)} - M_{3\times 3}\left[(S_t + I_t)^{(2)}\right]$$

The moving average used here is a 3×5 moving average over 7 terms, of coefficients $\frac{1}{15}\{1,\ 2,\ 3,\ 3,\ 3,\ 2,\ 1\}$ and retains linear trends. The coefficients are then normalised such that their sum over the whole 12-month period is approximately cancelled out:

$$\widetilde{S}_t^{(2)} = S_t^{(2)} - M_{2\times 12}\left(S_t^{(2)}\right)$$

Step 8: Estimation of the seasonally adjusted series:

$$SA_t^{(2)} = (TC_t + I_t)^{(2)} = X_t - \widetilde{S}_t^{(2)}$$

The whole difficulty lies, then, in the choice of the moving averages used for the estimation of the trend in steps 1 and 5 on the one hand, and for the estimation of the seasonal component in steps 3 and 5. The course of the algorithm in the form that is implemented in JDemetra+ is presented in the figure below. The adjustment for trading day effects, which is present in the original X-11 program, is omitted here, as since calendar correction is performed by the RegARIMA model, JDemetra+ does not perform further adjustment for these effects in the decomposition step.

A workflow diagram for the X-11 algorithm based upon training material from the Deutsche Bundesbank

0.0.0.0.1 * The iterative principle of X-11

To evaluate the different components of a series, while taking into account the possible presence of extreme observations, X-11 will proceed iteratively: estimation of components, search for disruptive effects in the irregular component, estimation of components over a corrected series, search for disruptive effects in the irregular component, and so on.

The Census X-11 program presents four processing stages (A, B, C, and D), plus 3 stages, E, F, and G, that propose statistics and charts and are not part of the decomposition per se. In stages B, C and D the basic algorithm is used as is indicated in the figure below.

A workflow diagram for the X-11 algorithm implemented in JDemetra+. Source: Based upon training material from the Deutsche Bundesbank

- **Part A: Pre-adjustments**

This part, which is not obligatory, corresponds in X-13ARIMA-SEATS to the first cleaning of the series done using the RegARIMA facilities: detection and estimation of outliers and calendar effects (trading day and Easter), forecasts and backcasts[^61] of the series. Based on these results, the program calculates prior adjustment factors that are applied to the raw series. The series thus corrected, Table B1 of the printouts, then proceeds to part B.

- **Part B: First automatic correction of the series**

This stage consists of a first estimation and down-weighting of the extreme observations and, if requested, a first estimation of the calendar effects. This stage is performed by applying the basic algorithm detailed earlier. These operations lead to Table B20, adjustment values for extreme observations, used to correct the unadjusted series and result in the series from Table C1.

- **Part C: Second automatic correction of the series**

Applying the basic algorithm once again, this part leads to a more precise estimation of replacement values of the extreme observations (Table C20). The series, finally “cleaned up”, is shown in Table D1 of the printouts.

- **Part D: Seasonal adjustment**

This part, at which our basic algorithm is applied for the last time, is that of the seasonal adjustment, as it leads to final estimates:

- of the seasonal component (Table D10);
- of the seasonally adjusted series (Table D11);
- of the trend component (Table D12);
- of the irregular component (Table D13).

- **Part E: Components modified for large extreme values**

Parts E includes:

- Components modified for large extreme values;
- Comparison the annual totals of the raw time series and seasonally adjusted time series;
- Changes in the final seasonally adjusted series;
- Changes in the final trend;
- Robust estimation of the final seasonally adjusted series.

The results from part E are used in part F to calculate the quality measures.

- **Part F: Seasonal adjustment quality measures**

Part F contains statistics for judging the quality of the seasonal adjustment. JDemetra+ presents selected output for part F, i.e.:

- M and Q statistics;
- Tables.

- **Part G: Graphics**

Part G presents spectra estimated for:

- Raw time series adjusted a priori (Table B1);
- Seasonally adjusted time series modified for large extreme values (Table E2);
- Final irregular component adjusted for large extreme values (Table E3).

Originally, graphics were displayed in character mode. In JDemetra+, these graphics are replaced favourably by the usual graphics software.

The Henderson moving average and the trend estimation

In iteration B (Table B7), iteration C (Table C7) and iteration D (Table D7 and Table D12) the trend component is extracted from an estimate of the seasonally adjusted series using Henderson moving averages. The length of the Henderson filter is chosen automatically by X-13ARIMA-SEATS in a two-step procedure.

It is possible to specify the length of the Henderson moving average to be used. X-13ARIMA-SEATS provides an automatic choice between a 9-term, a 13-term or a 23-term moving average. The automatic choice of the order of the moving average is based on the value of an indicator called $\frac{\bar{I}}{\bar{C}}$ ratio which compares the magnitude of period-on-period movements in the irregular component with those in the trend. The larger the ratio, the higher the order of the moving average selected. Moreover, X-13ARIMA-SEATS allows the user to choose manually any odd-numbered Henderson moving average. The procedure used in each part is very similar; the only differences are the number of options available and the treatment of the observations in the both ends of the series. The procedure below is applied for a monthly time series.

In order to calculate $\frac{\overline{I}}{\overline{C}}$ ratio a first decomposition of the SA series (seasonally adjusted) is computed using a 13-term Henderson moving average.

For both the trend (C) and irregular (I) components, the average of the absolute values for monthly growth rates (multiplicative model) or for monthly growth (additive model) are computed. They are denoted as $\overline{C}$ and $\overline{I}$, receptively, where:

- $\overline{C} = \frac{1}{n-1}\sum_{t=2}^{n}|C_t - C_{t-1}|$;
- $\overline{I} = \frac{1}{n-1}\sum_{t=2}^{n}|I_t - I_{t-1}|$.

Then the value of $\frac{\overline{I}}{\overline{C}}$ ratio is checked and in iteration B:

- If the ratio is smaller than 1, a 9-term Henderson moving average is selected;
- Otherwise, a 13-term Henderson moving average is selected.

Then the trend is computed by applying the selected Henderson filter to the seasonally adjusted series from Table B6. The observations at the beginning and at the end of the time series that cannot be computed by means of symmetric Henderson filters are estimated by ad hoc asymmetric moving averages.

In iterations C and D:

- If the ratio is smaller than 1, a 9-term Henderson moving average is selected;
- If the ratio is greater than 3.5, a 23-term Henderson moving average is selected.
- Otherwise, a 13-term Henderson moving average is selected.

The trend is computed by applying selected Henderson filter to the seasonally adjusted series from Table C6, Table D7 or Table D12, accordingly. At the both ends of the series, where a central Henderson filter cannot be applied, the asymmetric ends weights for the 7 term Henderson filter are used.

0.0.0.0.2 * Choosing the composite moving averages when estimating the seasonal component

In iteration D, Table D10 shows an estimate of the seasonal factors implemented on the basis of the modified SI (Seasonal – Irregular) factors estimated in Tables D4 and D9bis. This component will have to be smoothed to estimate the seasonal component; depending on the importance of the irregular in the SI component, we will have to use moving averages of varying length as in the estimate of the Trend/Cycle where the $\frac{\overline{I}}{\overline{C}}$ ratio was used to select the length of the Henderson moving average. The estimation includes several steps.

Step 1: Estimating the irregular and seasonal components

An estimate of the seasonal component is obtained by smoothing, month by month and therefore column by column, Table D9bis using a simple 7-term moving average, i.e. of coefficients $\frac{1}{7}\{1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1\}$. In order not to lose three points at the beginning and end of each column, all columns are completed as follows. Let us assume that the column that corresponds to the month is composed of N values $\{x_1,\ x_2,\ x_3,\ \dots x_{N-1},\ x_N\}$. It will be transformed into a series $\left\{x_{-2}, x_{-1}, x_0, x_1,\ x_2,\ x_3,\ \dots x_{N-1},\ x_N, x_{N+1},\ x_{N+1},\ x_{N+2},\ x_{N+3}\right\}$ with $x_{-2} = x_{-1} = x_0 = \frac{x_1+x_2+x_3}{3}$ and $x_{N+1} = x_{N+2} = x_{N+3} = \frac{x_N+x_{N-1}+x_{N-2}}{3}$. We then have the required estimates: $S = M_7(D9bis)$ and $I = D9bis - S$.

Step 2: Calculating the Moving Seasonality Ratios

For each i^{th} month the mean annual changes for each component is obtained by calculating

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

and

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

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

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

These ratios are presented in *Details* of the *Quality Measures* node under the *Decomposition (X11)* section. These ratios are used to compare the year-on-year changes in the irregular component with those in the seasonal component. The idea is to obtain, for each month, an indicator capable of selecting the appropriate moving average for the removal of any noise and providing a good estimate of the seasonal factor. The higher the ratio, the more erratic the series, and the greater the order of the moving average should be used. As for the rest, by default the program selects the same moving average for each month, but the user can select different moving averages for each month.

Step 3: Calculating the overall Moving Seasonality Ratio

The overall Moving Seasonality Ratio is calculated as follows:

$$\mathrm{MSR}_i = \frac{\sum_i N_i \ \overline{I}_i}{\sum_i N_i \overline{S}_i}$$

Step 4: Selecting a moving average and estimating the seasonal component

Depending on the value of the ratio, the program automatically selects a moving average that is applied, column by column (i.e. month by month) to the Seasonal/Irregular component in Table D8 modified, for extreme values, using values in Table D9.

The default selection procedure of a moving average is based on the Moving Seasonality Ratio in the following way:

- If this ratio occurs within zone A (MSR $<$ 2.5), a 3×3 moving average is used; if it occurs within zone C ($3.5 < \mathrm{MSR} < 5.5$), a 3×5 moving average is selected; if it occurs within zone E (MSR 6.5), a 3×9 moving average is used;
- If the MSR occurs within zone B or D, one year of observations is removed from the end of the series, and the MSR is recalculated. If the ratio again occurs within zones B or D, we start over again, removing a maximum of five years of observations. If this does not work, i.e. if we are again within zones B or D, a 3×5 moving average is selected.

The chosen symmetric moving average corresponds, as the case may be 5 (3×3), 7 (3×5) or 11 (3×9 3×9) terms, and therefore does not provide an estimate for the values of seasonal factors in the first 2 (or 3 or 5) and the last 2 (or 3 or 5) years. These are then calculated using associated asymmetric moving averages.

Moving average selection procedure, source: DAGUM, E. B.(1999)

0.0.0.0.3 * Identification and replacement of extreme values

X-13ARIMA-SEATS detects and removes outliers in the RegARIMA part. However, if there is a seasonal heteroscedasticity in a time series i.e. the variance of the irregular component is different in different calendar months. Examples for this effect could be the weather and snow-dependent output of the construction sector in Germany during winter, or changes in Christmas allowances in Germany and resulting from this a transformation in retail trade turnover before Christmas. The ARIMA model is not on its own able to cope with this characteristic. The practical consequence is given by the detection of additional extreme values by X-11. This may not be appropriate if the seasonal heteroscedasticity is produced by political interventions or other influences. The ARIMA models assume a constant variance and are therefore not by themselves able to cope with this problem. Choosing longer (in the case of diverging weather conditions in the winter time for the construction sector) or shorter filters (in the case of a changing pattern of retail trade turnover in the Christmas time) may be reasonable in such cases. It may even be sensible to take into account the possibility of period-specific (e.g. month-specific) standard deviations, which can be done by changing the default settings of the **calendarsigma** parameter (see Specifications-X13 section). The value of the **calendarsigma** parameter will have an impact on the method of calculation of the moving standard deviation in the procedure for extreme values detection presented below.

Step 1: Estimating the seasonal component

The seasonal component is estimated by smoothing the SI component separately for each period using a 3×3 moving average, i.e.:

$$\frac{1}{9} \times \begin{Bmatrix} 1,0,0,0,0,0,0,0,0,0,0,0, \\ 2,0,0,0,0,0,0,0,0,0,0,0, \\ 3,0,0,0,0,0,0,0,0,0,0,0, \\ 2,0,0,0,0,0,0,0,0,0,0,0, \\ 1,0,0,0,0,0,0,0,0,0,0,0, \end{Bmatrix}$$

Step 2: Normalizing the seasonal factors

The preliminary seasonal factors are normalized in such a way that for one year their average is equal to zero (additive model) or to unity (multiplicative model).

Step 3: Estimating the irregular component

The initial normalized seasonal factors are removed from the Seasonal-Irregular component to provide an estimate of the irregular component.

Step 4: Calculating a moving standard deviation

By default, a moving standard deviation of the irregular component is calculated at five-year intervals. Each standard deviation is associated with the central year used to calculate it. The values in the central year, which in the absolute terms deviate from average by more than the **Usigma** parameter are marked as extreme values and assigned a zero weight. After excluding the extreme values the moving standard deviation is calculated once again.

Step 5: Detecting extreme values and weighting the irregular

The default settings for assigning a weight to each value of irregular component are:

- Values which are more than **Usigma** (2.5, by default) standard deviations away (in the absolute terms) from the 0 (additive) or 1 (multiplicative) are assigned a zero weight;
- Values which are less than 1.5 standard deviations away (in the absolute terms) from the 0 (additive) or 1 (multiplicative) are assigned a full weight (equal to one);

- Values which lie between 1.5 and 2.5 standard deviations away (in the absolute terms) from the 0 (additive) or 1 (multiplicative) are assigned a weight that varies linearly between 0 and 1 depending on their position.

The default boundaries for the detection of the extreme values can be changed with **LSigma** and **USigma** parameters

Step 6: Adjusting extreme values of the seasonal-irregular component

Values of the SI component are considered extreme when a weight less than 1 is assigned to their irregular. Those values are replaced by a weighted average of five values:

- The value itself with its weight;
- The two preceding values, for the same period, having a full weight(if available);
- The next two values, for the same period, having full a weight (if available).

When the four full-weight values are not available, then a simple average of all the values available for the given period is taken.

This general algorithm is used with some modification in parts B and C for detection and replacement of extreme values.

0.0.0.0.4 * X-11 tables

The list of tables produced by JDemetra+ is presented below. It is not identical to the output produced by the original X-11 program.

Part A – Preliminary Estimation of Outliers and Calendar Effects.

This part includes prior modifications to the original data made in the RegARIMA part:

- Table A1 – Original series;
- Table A1a – Forecast of Original Series;
- Table A2 – Leap year effect;
- Table A6 – Trading Day effect (1 or 6 variables);
- Table A7 – The Easter effect;
- Table A8 – Total Outlier Effect;
- Table A8i – Additive outlier effect;
- Table A8t – Level shift effect;
- Table A8s – Transitory effect;

- Table A9 – Effect of user-defined regression variables assigned to the seasonally adjusted series or for which the component has not been defined;
- Table 9sa – Effect of user-defined regression variables assigned to the seasonally adjusted series;
- Table9u – Effect of user-defined regression variables for which the component has not been defined.

Part B – Preliminary Estimation of the Time Series Components:

- Table B1 – Original series after adjustment by the RegARIMA model;
- Table B2 – Unmodified Trend (preliminary estimation using composite moving average);
- Table B3 – Unmodified Seasonal – Irregular Component (preliminary estimation);
- Table B4 – Replacement Values for Extreme SI Values;
- Table B5 – Seasonal Component;
- Table B6 – Seasonally Adjusted Series;
- Table B7 – Trend (estimation using Henderson moving average);
- Table B8 – Unmodified Seasonal – Irregular Component;
- Table B9 – Replacement Values for Extreme SI Values;
- Table B10 – Seasonal Component;
- Table B11 – Seasonally Adjusted Series;
- Table B13 – Irregular Component;
- Table B17 – Preliminary Weights for the Irregular;
- Table B20 – Adjustment Values for Extreme Irregulars.

Part C – Final Estimation of Extreme Values and Calendar Effects:

- Table C1 – Modified Raw Series;
- Table C2 – Trend (preliminary estimation using composite moving average);
- Table C4 – Modified Seasonal – Irregular Component;
- Table C5 – Seasonal Component;
- Table C6 – Seasonally Adjusted Series;
- Table C7 – Trend (estimation using Henderson moving average);
- Table C9 – Seasonal – Irregular Component;

- Table C10 – Seasonal Component;
- Table C11 – Seasonally Adjusted Series;
- Table C13 – Irregular Component;
- Table C20 – Adjustment Values for Extreme Irregulars.

Part D – Final Estimation of the Different Components:

- Table D1 – Modified Raw Series;
- Table D2 – Trend (preliminary estimation using composite moving average);
- Table D4 – Modified Seasonal – Irregular Component;
- Table D5 – Seasonal Component;
- Table D6 – Seasonally Adjusted Series;
- Table D7 – Trend (estimation using Henderson moving average);
- Table D8 – Unmodified Seasonal – Irregular Component;
- Table D9 – Replacement Values for Extreme SI Values;
- Table D10 – Final Seasonal Factors;
- Table D10A – Forecast of Final Seasonal Factors;
- Table D11 – Final Seasonally Adjusted Series;
- Table D11A – Forecast of Final Seasonally Adjusted Series;
- Table D12 – Final Trend (estimation using Henderson moving average);
- Table D12A – Forecast of Final Trend Component;
- Table D13 – Final Irregular Component;
- Table D16 – Seasonal and Calendar Effects;
- Table D16A – Forecast of Seasonal and Calendar Component;
- Table D18 – Combined Calendar Effects Factors.

Part E – Components Modified for Large Extreme Values:

- Table E1 – Raw Series Modified for Large Extreme Values;
- Table E2 – SA Series Modified for Large Extreme Values;
- Table E3 – Final Irregular Component Adjusted for Large Extreme Values;
- Table E11 – Robust Estimation of the Final SA Series.

Part F – Quality indicators:

- Table F2A – Changes, in the absolute values, of the principal components;
- Table F2B – Relative contribution of components to changes in the raw series;
- Table F2C – Averages and standard deviations of changes as a function of the time lag;
- Table F2D – Average duration of run;
- Table F2E – I/C ratio for periods span;
- Table F2F – Relative contribution of components to the variance of the stationary part of the original series;
- Table F2G – Autocorrelogram of the irregular component.

Filter length choice

A seasonal filter is a weighted average of a moving span of fixed length within a time series that can be used to remove a fixed seasonal pattern. X-13ARIMA-SEATS uses several of these filters, according to the needs of the different stages of the program. As only X-13ARIMA-SEATS allows the user to manually select seasonal filters, this case study can be applied only to the X-13ARIMA-SEATS specifications.

The automatic seasonal adjustment procedure uses the default options to select the most appropriate moving average. However there are occasions when the user will need to specify a different seasonal moving
average to that identified by the program. For example, if the SI values do not closely follow the seasonal component, it may be appropriate to use a shorter moving average. Also the presence of sudden breaks in the seasonal pattern – e.g. due to changes in the methodology – can negatively impact on the automatic selection of the most appropriate seasonal filter. In such cases the usage of short seasonal filters in the selected months or quarters can be considered. Usually, a shorter seasonal filter (3×1) allows seasonality to change very rapidly over time. However, a very short seasonal filter should not normally be used, as it might often lead to large revisions as new data becomes available. If a short filter is to be used it will usually be limited to one month/quarter with a known reason for wanting to capture a rapidly changing seasonality.

In the standard situation one seasonal filter is applied to all individual months/quarters. The estimation of seasonal movements is therefore based on the sample windows of equal lengths for each individual month/quarter (i.e. for each month/quarter the seasonal filter length or the number of years representing the major part of the seasonal filter weights is identical). This approach relies on the assumption that the number of past periods in which the conditions causing seasonal behaviour are sufficiently homogenous is the same in all months/quarters. However, this assumption does not always hold. Seasonal causes may change in one month, while staying the same in others[16]. For instance, seasonal heteroskedasticity might require different filter lengths in different months or quarters.

Another interesting example is industrial production in Germany. It can be influenced by school holidays, since many employees have school-age children, which interrupt their working pattern during these school holidays. Consequently, businesses may temporarily suspend or lower production during these periods. Since school holidays do not occur at the same time throughout Germany and their timing varies from year to year in the individual federal states, the effect is not completely captured by seasonal adjustment. And since school holidays are treated as usual working days, these effects are not captured by calendar adjustment either. The majority of school holidays in Germany can take place either in July or in August. This yields higher variances in the irregular component for these months compared to the rest of the year. Therefore, in this case a longer seasonal filter is used for these months to account for this.

Another example might be given by German retail trade. Due to changes in the consumers' behaviour around Christmas – possibly more gifts of money – the seasonal peak in December has become steadily less pronounced. To account for this moving seasonality, shorter seasonal filters in December than during the rest of the year need to be applied.

JDemetra+ offers the options to assign a different seasonal filter length to each period (month or quarter). The program offers these options in the *single spec* mode as well as in the *multispec* mode, albeit they are available only in the *Specifications* window, after a document is created.

M-stats

The details about the measures are given below.

- $M1$ measures the contribution of the irregular component to the total variance. When it is above 1 some changes in outlier correction should be considered.
- $M2$, which is a very similar to $M1$, is calculated on the basis of the contribution of the irregular component to the stationary portion of the variance. When it is above 1, some changes in an outlier correction should be considered.
- $M3$ compares the irregular to the trend taken from a preliminary estimate of the seasonally adjusted series. If this ratio is too large, it is difficult to separate the two components from each other. When it is above 1 some changes in outlier correction should be considered.
- $M4$ tests the randomness of the irregular component. A value above 1 denotes a correlation in the irregular component. In such case a shorter seasonal moving average filter should be considered.

[16] When the series are non-stationary differentiation is performed before the seasonality tests.

- $M5$ is used to compare the significance of changes in the trend with that in the irregular. When it is above 1 some changes in outlier correction should be considered.
- $M6$ checks the SI (seasonal – irregular components ratio). If annual changes in the irregular component are too small in relation to the annual changes in the seasonal component, the 3×5 seasonal filter used for the estimation of the seasonal component is not flexible enough to follow the seasonal movement. In such case a longer seasonal moving average filter should be considered. It should be stressed that $M6$ is calculated only if the 3×5 filter has been applied in the model.
- $M7$ is the combined test for the presence of an identifiable seasonality. The test compares the relative contribution of stable and moving seasonality[17].
- $M8$ to $M11$ measure if the movements due to the short-term quasi-random variations and movements due to the long-term changes are not changing too much over the years. If the changes are too strong then the seasonal factors could be erroneous. In such case a correction for a seasonal break or the change of the seasonal filter should be considered.

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

Edit : problem with tables display

$$Q = \frac{10M1 + 11M2 + 10M3 + 8M4 + 11M5 + 10M6 + 18M7 + 7M8 + 7M9 + 4M10 + 4M11}{100} \tag{0.10}$$

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

$$Q - M2 = \frac{10M1 + 10M3 + 8M4 + 11M5 + 10M6 + 18M7 + 7M8 + 7M9 + 4M10 + 4M11}{89} \tag{0.11}$$

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.

The tables displayed in the *Quality measures* $\rightarrow$ *Details* node correspond to the F-set of tables produced by the original X-11 algorithm. To facilitate the analysis of the results, the numbers and the names of the tables are given under each table following the convention used in LADIRAY, D., and QUENNEVILLE, B. (1999).

[17]See section Combined seasonality tests.

Detailed tables

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[18]:

$$\text{Component}_d = \frac{1}{n-d}\sum_{t=d+1}^{n} |Table_t - Table_{t-d}| \tag{0.12}$$

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 100: **Table F2A – changes, in the absolute values, of the principal components**

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:

$$O_d^2 \approx C_d^2 + S_d^2 + I_d^2 + P_d^2 + TD\&H_d^2 \tag{0.13}$$

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

$$O_d^{*2} = C_d^2 + S_d^2 + I_d^2 + P_d^2 + TD\&H_d^2 \tag{0.14}$$

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:

$$\frac{I_d^2}{O_d^{*2}}, \frac{C_d^2}{O_d^{*2}}, \frac{S_d^2}{O_d^{*2}}, \frac{P_d^2}{O_d^{*2}}, \frac{TD\&H_d^2}{O_d^{*2}}$$

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:

$$100 \times \frac{O_d^{*2}}{O_d^2}$$

[18]For the multiplicative decomposition the bfollowing formula is used:

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

which is an indicator of how well the approximation $(O_d^*)^2 \approx O_d^2$ 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 101: **Table F2B – relative contribution of components to changes in the raw series**

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 102: **Table F2C – Averages and standard deviations of changes as a function of the time lag**

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 103: **Table F2D – Average duration of run**

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.

The relative contribution of components to the variance of the stationary part of the original

I/C Ratio for indicated span

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

I/C Ratio: 0.314

Figure 104: **Table F2E –** $\frac{\mathbf{I}}{\mathbf{C}}$ ratio for periods span

series is calculated for the irregular component (I), trend made stationary[19] (C), seasonal component (S) and calendar effects (TD&H). The short description of the calculation method is given in LADIRAY, D., and QUENNEVILLE, B. (1999).

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 105: **Table F2F – Relative contribution of components to the variance of the stationary part of the original series**

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[20]:

$$\mathrm{Corr}_k I = \frac{\sum_{t=k+1}^{N}(I_t - 1)(I_{t-k} - 1)}{\sum_{t=1}^{N}(I_t - 1)^2}] \tag{0.15}$$

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

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

[19] The component is estimated by extracting a linear trend from the trend component presented in Table D12.

[20] 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 106: **Table F2G – Autocorrelation 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).

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 107: **Cochran test**

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

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

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:

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

These ratios are published in Table D9A in X13ARIMA-SEATS software. In JDemetra+ they are presented in the details of the quality measures.

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 108: **Table D9a – Moving seasonality ratios**

STL: Local regression decomposition

Under construction

SEATS decomposition

Under construction.

Introduction

SEATS is a program for estimating unobserved components in a time series. It follows the ARIMA-model-based (AMB) method, developed from the work of CLEVELAND, W.P., and TIAO, G.C. (1976), BURMAN, J.P. (1980), HILLMER, S.C., and TIAO, G.C. (1982), BELL, W.R., and HILLMER, S.C. (1984) and MARAVALL, A., and PIERCE, D.A. (1987).

In JDemetra+ the input for the model based signal extraction procedure is always provided by TRAMO and includes the original series y_t, the linearized series x_t (i.e. the original series y_t with the deterministic effects removed), the ARIMA model for the stochastic (linearized) time series x_t and the deterministic effects (calendar effects, outliers and other regression variable effects)[21]. SEATS decomposes the linearized series (and the ARIMA model) into trend, seasonal, transitory and irregular components, provides forecasts for these components, together with the associated standard errors, and finally assign the deterministic effects to each component yielding the *final* components[22]. The Minimum Mean Square Error (MMSE) estimators of the components are computed with a Wiener-Kolmogorov filter applied to the finite series extended with forecasts and backcasts[23].

ARIMA modellling of the input series

One of the fundamental assumptions made by SEATS is that the linearized time series x_t follows the ARIMA model

$$\phi(B)\delta\left(B\right)x_t = \theta(B)a_t \tag{0.16}$$

where:

[21] In the original software SEATS can be used either with TRAMO, operating on the input received from the latter, or alone, fitting an ARIMA model to the series.

[22] GÓMEZ, V., and MARAVALL, A. (1998).

[23] GÓMEZ, V., and MARAVALL, A. (1997).

- B – the backshift operator ($Bx_t = x_{t-1}$);
- $\delta(B)$ – a non-stationary autoregressive (AR) polynomial in B (unit roots);
- $\theta(B)$ – an invertible moving average (MA) polynomial in B and in B^S, which can be expressed in the multiplicative form $(1 + \vartheta_1 B + \ldots + \vartheta q B^q)\left(\ 1 + \Theta 1 B^s + \ldots + \Theta_Q B^{\mathrm{sQ}}\right)$;
- $\phi(B)$ – a stationary autoregressive (AR) polynomial in B and in B^S containing regular and seasonal unit roots, with s representing the number of observations per year;
- a_t – a white-noise variable with the variance $V(a)$.

It should be noted that the stochastic time series can be predicted using its past observations and making an error. The variable a_t, which is assumed to be white noise, is the fundamental *innovation* to the series at time t, that is the part that cannot be predicted based on the past history of the series.

Denoting $\varphi(B) = \phi(B)\,\delta(B)$, Equation **??** can be written in a more concise form as

$$\varphi(B)\,x_t = \theta(B)a_t \tag{0.17}$$

where $\varphi(B)$ contains both the stationary and the nonstationary roots.

Derivation of the models for the components

Let us consider the additive decomposition model

$$x_t = \sum_{i=1}^{k} x_{\mathrm{it}} \tag{0.18}$$

where i refers to the orthogonal components: trend, seasonal, transitory or irregular. Apart from the irregular component, supposed to be a white noise, it is assumed that each component follows the ARIMA model which can be represented, using the notation of Equation **??** , as:

$$\varphi_i(B)\ \ x_{\mathrm{it}} = \theta_i(B)a_{\mathrm{it}} \tag{0.19}$$

where

- $\varphi_i(B) = \phi_i(B)\,\delta_i(B)\,,\ \ x_{\mathrm{it}}$ is the *i-th* unobserved component,
- $\varphi_i(B)$ and $\theta_i(B)$ are finite polynomials of order p_i and q_i, respectively,

- a_{it}, the disturbance associated with such component, is a white noise process with zero mean and constant variance $V(a_i)$ and a_{it} and a_{jt} are not correlated for $i \neq j$ and for any t..

These disturbances are functions of the innovations in the series and are called "pseudo-innovations" in the literature concerning the AMB decomposition as they refer to the components that are never observed [24]. In the JDemetra+ documentation the term "innovations" is used to refer to the "pseudo-innovations".

The following assumptions hold for Equation **??** . For each i the polynomials $\phi_i\left(B\right)$, $\delta_i\left(B\right)$ and $\theta_i(B)$ are prime and of finite order. The roots of $\delta_i\left(B\right)$ lies on the unit circle; those of $\phi_i\left(B\right)$ lie outside, while all the roots of $\theta_i\left(B\right)$ are on or outside the unit circle. This means that nonstationary and noninvertible components are allowed. Since different roots of the AR polynomial induce peaks in the spectrum[25] of the series at different frequencies, and given that different components are associated with the spectral peaks for different frequencies, it is assumed that for $i \neq j$ the polynomials $\phi_i\left(B\right)$ and $\phi_j\left(B\right)$ do not share any common root (they are coprime). Finally, it is assumed that the polynomials $\theta_i\left(B\right),\ i = 1, \dots, k$ are prime share no unit root in common, guaranteeing the invertibility of the overall series. In fact, since the unit root of $\theta_i\left(B\right)$ induce a spectral zero, when the polynomials $\theta_i\left(B\right),\ i = 1, \dots, k$ share no unit root in common, there is no frequency for which all component spectra become zero[26].

Since aggregation of ARIMA models yields ARIMA models, the series x_t will also follow an ARIMA model, as in Equation **??** , and consequently the following identity can be derived:

$$\frac{\theta(B)}{\varphi(B)}a_t = \sum_{i=1}^{k}\frac{\theta_i(B)}{\varphi_i(B)}a_{\text{it}} \tag{0.20}$$

In the ARIMA model based approach implemented in SEATS, the ARIMA model identified and estimated for the observed series x_t is decomposed to derive the models for the components. In particular, the AR polynomials for the components, $\varphi_i\left(B\right)$, are easily derived through the factorization of the AR polynomial $\varphi\left(B\right)$:

$$\varphi\left(B\right) = \prod_{i=1}^{k}\varphi_i\left(B\right) \tag{0.21}$$

while the MA polynomials for the components, together with the innovation variances $V(a_i)$, cannot simply be obtained through the relationship:

[24] GÓMEZ, V., and MARAVALL, A. (2001a).
[25] For description of the spectrum see section Spectral Analysis.
[26] MARAVALL, A. (1995).

$$\theta(B)a_t = \sum_{i=1}^{k} \varphi_{\text{ni}}\,(B)\theta_i(B)a_{\text{it}} \tag{0.22}$$

where $\varphi_{\text{ni}}\,(B)$ is the product of all $\varphi_j\,(B)\,,\ j = 1, \dots, k$, except from $\varphi_i\,(B)$. Further assumptions are therefore needed to cope with the underidentification problem: i) $p_i \geq q_i$ and ii) the canonical decomposition, i.e. the decomposition that allocate all additive white noise to the irregular component (yielding noninvertible components except the irregular).

To understand how SEATS factorizes the AR polynomials, first a concept of a root will be explored[27].

The equation Equation **??** can be expressed as:

$$\psi^{-1}(B)x_t = a_t(1 + \varphi_1 B + \dots \varphi_p B^p)x_t = (1 + \theta_1 B + \dots \theta_q B^q)a_t \tag{0.23}$$

Let us now consider Equation **??** in the inverted form:

$$\theta\,(B)\,y_t = \varphi(B)a_t \tag{0.24}$$

If both sides of Equation **??** are multiplied by x_{t-k} with $k > q$, and expectations are taken, the right hand side of the equation vanishes and the left hand side becomes:

$$\varphi(B)\gamma_k = \gamma_k + \varphi_1\gamma_{k-1} + \dots \varphi_p\gamma_{k-p} = 0 \tag{0.25}$$

where B operates on the subindex k.

The autocorrelation function γ_k is a solution of Equation **??** with the characteristic equation:

$$z^p + \varphi_1 z^{p-1} + \dots \varphi_{p-1} z + \varphi_p = 0 \tag{0.26}$$

If z_1,..., z_p are the roots of Equation **??** , the solutions of Equation **??** can be expressed as:

$$\gamma_k = \sum_{i=1}^{p} z_i^k \tag{0.27}$$

and will converge to zero as $k \to \infty$ when $|r_i| < 1,\ i = 1, \dots, p$. From Equation **??** and Equation **??** it can be noticed that $z_1 = B_i^{-1}$, meaning that z_1,..., z_p are the inverses of the roots $B_1, \dots, B_p$ of the polynomial $\varphi(B)$. The convergence of γ_k implies that the roots of the $\varphi(B)$ are larger than 1 in modulus (lie outside the unit circle). Therefore, from the equation

[27] Description based on KAISER, R., and MARAVALL, A. (2000) and MARAVALL, A. (2008c).

$$\varphi(B)^{-1} = \frac{1}{(1-z_1)\ldots(1-z_1)} \tag{0.28}$$

it can be derived that $\varphi(B)^{-1}$ is convergent and all its inverse roots are less than 1 in modulus.

Equation Equation **??** has real and complex roots (solutions). Complex number $x = a + bi$, with a and b both real numbers, can be represented as $x = r\,(cos(\omega) + i\ sin(\omega))$, where i is the imaginary unit ($i^2 = -1$), r is the modulus of x, that is $r = |x| = \sqrt{a^2 + b^2}$ and ω is the argument (frequency). When roots are complex, they are always in pairs of complex conjugates. The representation of the complex number $x = a+bi$ has a geometric interpretation in the complex plane established by the real axis and the orthogonal imaginary axis.

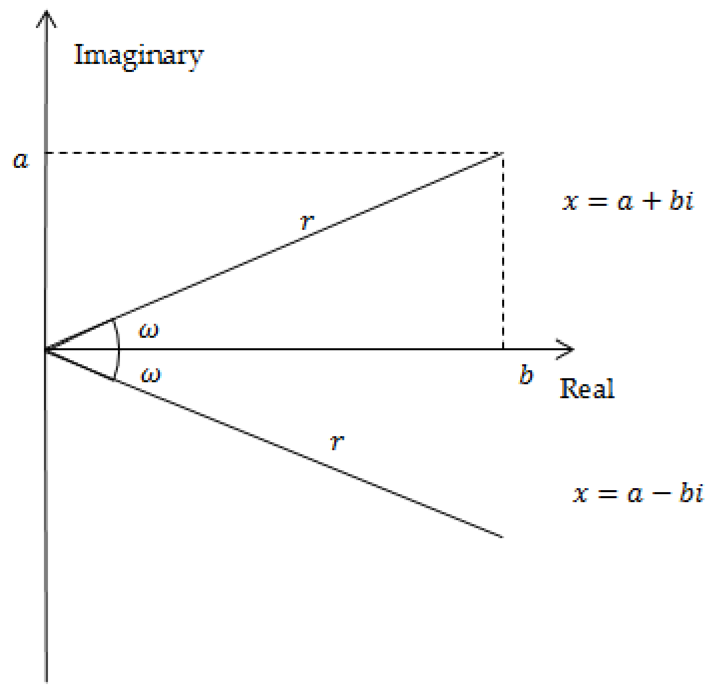


Figure 109: **Geometric representation of a complex number and of its conjugate**

Representing the roots of the characteristic equation Equation **??** in the complex plane enhances understanding how they are allocated to the components. When the modulus r of the roots in z are greater than 1 (i.e. modulus of the roots in $\varphi(B)\ < 1$), the solution of the characteristic equation has a systematic explosive process, which means that the impact of the given impulse on the time series is more and more pronounced in time. This behaviour is not in line with the developments that can be identified in actual economic series. Therefore, the models estimated by TRAMO-SEATS (and X-13ARIMA-SEATS) have never inverse roots in B with modulus greater than 1.

The characteristic equations associated with the regular and the seasonal differences have roots in $\varphi(B)$ with modulus $r = 1$. They are called non-stationary roots and can be represented on the unit circle. Let us consider the seasonal differencing operator applied to a quarterly time series $(1 - B^4)$. Its characteristic equation is $(z^4 - 1) = 0$ with solutions given by $z = \sqrt[4]{1}$, i.e. $z_{1,2} = \pm 1$ and $z_{3,4} = \pm i1$. The first two solutions are real and the last two are complex conjugates. They are represented by the black points on the unit circle on the figure below.

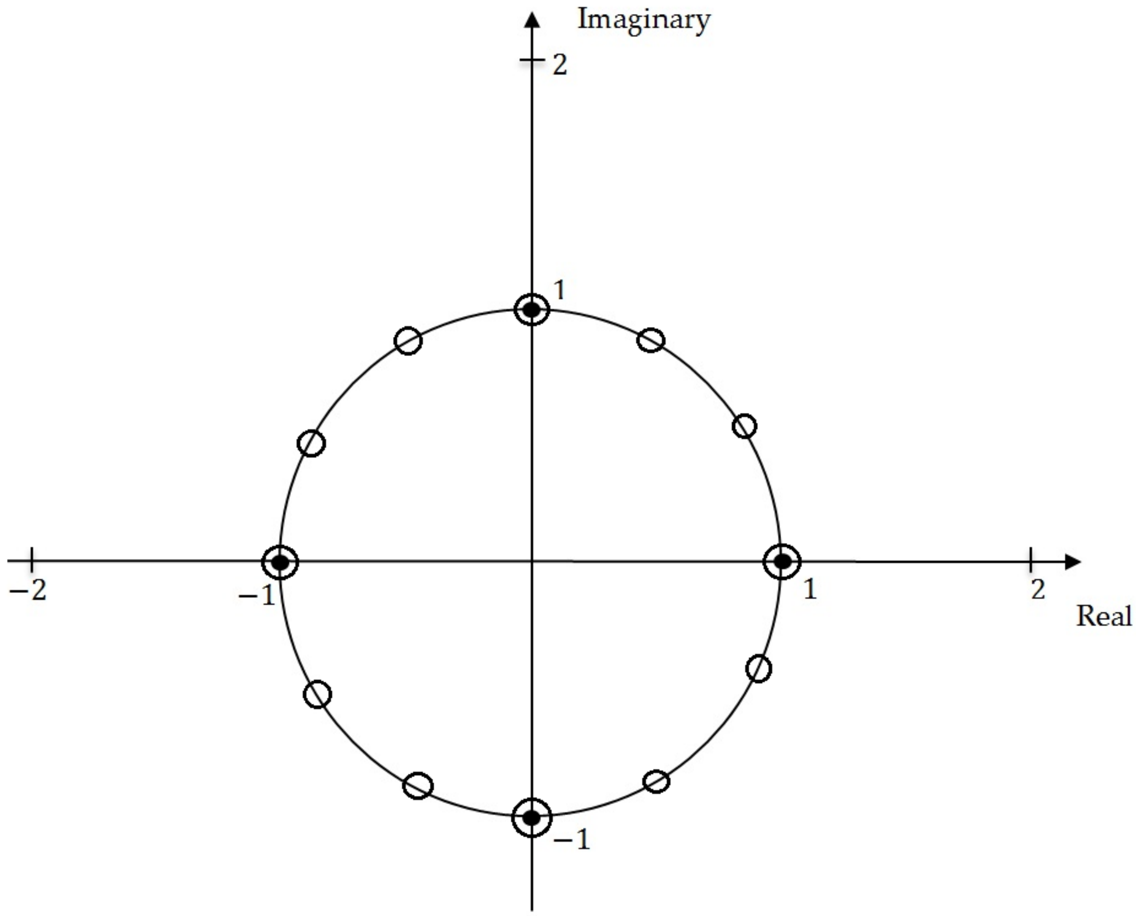


Figure 110: **Unit roots on the unit circle**

For the seasonal differencing operator $(1 - B^{12})$ applied to the monthly time series the characteristic equation $(z^{12} - 1) = 0$ has twelve non-stationary solutions given by $z = \sqrt[12]{1}$: two real and ten complex conjugates, represented by the white circles in unit roots figure above.

The complex conjugates roots generate the periodic movements of the type:

$$z_t = A^t \cos(\omega t + W). \quad (0.29)$$

where:

- A – amplitude;
- ω – angular frequency (in radians);
- W – phase (angle at $t = 0$).

The frequency f, i.e. the number of cycles per unit time, is $\frac{\omega}{2\pi}$. If it is multiplied by s, the number of observations per year, the number of cycles completed in one year is derived. The period of function Equation **??** , denoted by τ, is the number of units of time (months/quarters) it takes for a full circle to be completed.

For quarterly series the seasonal movements are produced by complex conjugates roots with angular frequencies at $\frac{\pi}{2}$ (one cycle per year) and π (two cycles per year). The corresponding number of cycles per year and the length of the movements are presented in the table below.

Seasonal frequencies for a quarterly time series

Angular frequency (ω)	Frequency (cycles per unit time) (f)	Cycles per year	Length of the movement measured in quarters (τ)
$\frac{\pi}{2}$	0.25	1	4
π	0.5	2	2

For monthly time series the seasonal movements are produced by complex conjugates roots at the angular frequencies: $\frac{\pi}{6}, \frac{\pi}{3}, \frac{\pi}{2}, \frac{2\pi}{3}, \frac{5\pi}{6}$ and π. The corresponding number of cycles per year and the length of the movements are presented in the table below:

Seasonal frequencies for a monthly time series

Angular frequency (ω)	Frequency (cycles per unit time) (f)	Cycles per year	Length of the movement measured in months (τ)
$\frac{\pi}{6}$	0.083	1	12
$\frac{\pi}{3}$	0.167	2	6
$\frac{\pi}{2}$	0.250	3	4
$\frac{2\pi}{3}$	0.333	4	3
$\frac{5\pi}{6}$	0.417	5	2.4
π	0.500	6	2

In JDemetra+ SEATS assigns the roots of the AR full polynomial to the components according to their associated modulus and frequency, i.e.:[28]

- Roots of $(1-B)^{d}$ are assigned to trend component.
- Roots of $(1-B^{s})^{d_s} = ((1-B)(1+B+...+B^{s-1}))^{d_s}$ are assigned to the trend component (root of $(1-B)^{d_s}$) and to the seasonal component (roots of $(1+B+...+B^{s-1})^{d_s}$).
- When the modulus of the inverse of a real positive root of $\varphi(B)$ is greater than k or equal to k, where k is the threshold value controlled by the *Trend boundary* parameter(in the original SEATS it is controlled by *rmod*)[29], then the root is assigned to the trend component. Otherwise it is assigned to the transitory component.
- Real negative inverse roots of $_p(B)$ associated with the seasonal two-period cycle are assigned to the seasonal component if their modulus is greater than k, where k is the threshold value controlled by the *Seasonal boundary* and the *Seas. boundary (unique)* parameters. Otherwise they are assigned to the transitory component.
- Complex roots, for which the argument (angular frequency) is close enough to the seasonal frequency are assigned to the seasonal component. Closeness is controlled by the *Seasonal tolerance* and *Seasonal tolerance (unique)* parameters (in the original SEATS it is controlled by *epsphi*). Otherwise they are assigned to the transitory component.
- If d_s (seasonal differencing order) is present and Bphi < 0 (Bphi is the estimate of the seasonal autoregressive parameter), the real positive inverse root is assigned to the trend component and the other $(s-1)$ inverse roots are assigned to the seasonal component. When $d_s = 0$, the root is assigned to the seasonal when Bphi < -0.2 and/or the overall test for seasonality indicates presence of seasonality. Otherwise it goes to the transitory component. Also, when Bphi > 0, roots are assigned to the transitory component.

For further details about JDemetra+ parameters see section TramoSeats.

It should be highlighted that when $Q > P$, where Q and P denote the orders of the polynomials $\varphi(B)$ and $\theta(B)$, the SEATS decomposition yields a pure MA $(Q-P)$ component (hence transitory). In this case the transitory component will appear even when there is no AR factor allocated to it.

Once these rules are applied, the factorization of the AR polynomial presented by Equation **??** yields to the identification of the AR polynomials for the components which contain, respectively, the AR roots associated with the trend component, the seasonal component and the

[28]For details see MARAVALL, A., CAPORELLO, G., PÉREZ, D., and LÓPEZ, R. (2014).
[29]In JDemetra+ this argument is called *Trend boundary*.

transitory component.[30]

Then with the partial fraction expansion the spectrum of the final components are obtained.

For example, the Airline model for a monthly time series:

$$(1-B)(1-B^{12})x_t = (1+\theta_1 B)(1+\Theta_1 B^{12})\ a_t \tag{0.30}$$

is decomposed by SEATS into the model for the trend component:

$$(1-B)(1-B)c_t = (1+\theta_{c,1}B+\theta_{c,2}B^2)a_{c,t} \tag{0.31}$$

and the model for the seasonal component:

$$(1+B+...+B^{11})\,s_t = \left(1+\theta_{s,1}B+...+\theta_{s,11}B^{11}\right)a_{s,t}, \tag{0.32}$$

As a result, the Airline model is decomposed as follows:

$$\frac{(1+\theta_1 B)(1+\Theta_1 B^{12})}{(1-B)(1-B)}a_t = \frac{\left(1+\theta_{s,1}B+...+\theta_{s,11}B^{11}\right)}{(1+B+...+B^{11})}a_{s,t} + \frac{(1+\theta_{c,1}B+\theta_{c,2}B^2)}{(1-B)(1-B)}a_{c,t} + u_t \tag{0.33}$$

The transitory component is not present in this case and the irregular component is the white noise.

The partial fractions decomposition is performed in a frequency domain. In essence, it consists in portioning of the pseudo-spectrum[31] of x_t into additive spectra of the components. When the AMB decomposition of the ARIMA model results in the non-negative spectra for all components, the decomposition is called admissible[32]. In such case an infinite number of admissible decompositions exists, i.e. decompositions that yield the non-negative spectra of all components. Therefore, the MA polynomials and the innovation variances cannot be yet identified from the model of x_t. As sketched above, to solve this underidentification problem and identify a unique decomposition, it is assumed that for each component the order of the

[30] The AR roots close to or at the trading day frequency generates a stochastic trading day component. A stochastic trading day component is always modelled as a stationary ARMA(2,2), where the AR part contains the roots close to the TD frequency, and the MA(2) is obtained from the model decomposition (MARAVALL, A., and PÉREZ, D. (2011)). This component, estimated by SEATS, is not implemented by the current version of JDemetra+.

[31] The term pseudo-spectrum is used for a non-stationary time series, while the term spectrum is used for a stationary time series.

[32] If the ARIMA model estimated in TRAMO does not accept an admissible decomposition, SEATS replaces it with a decomposable approximation. The modified model is therefore used to decompose the series. There are also other rare situations when the ARIMA model chosen by TRAMO is changed by SEATS. It happens when, for example, the ARIMA models generate unstable seasonality or produce a senseless decomposition. Such examples are discussed by MARAVALL, A. (2009).

MA polynomial is no greater than the order of the AR polynomial and the canonical solution of S.C. Hillmer and G.C. Tiao is applied[33], i.e. all additive white noise is added to the irregular component As a consequence all components derived from the canonical decomposition, except from the irregular, have a spectral minimum of zero and are thus noninvertible[34]. Given the stochastic features of the series, it can be shown by that the canonical decomposition produces as stable as possible trend and seasonal components since it maximizes the variance of the irregular and minimizes the variance of the other components[35]. However, there is a price to be paid as canonical components can produce larger revisions in the preliminary estimators of the component[36] than any other admissible decomposition.

The figure below represents the pseudo-spectrum for the canonical trend and an admissible trend.

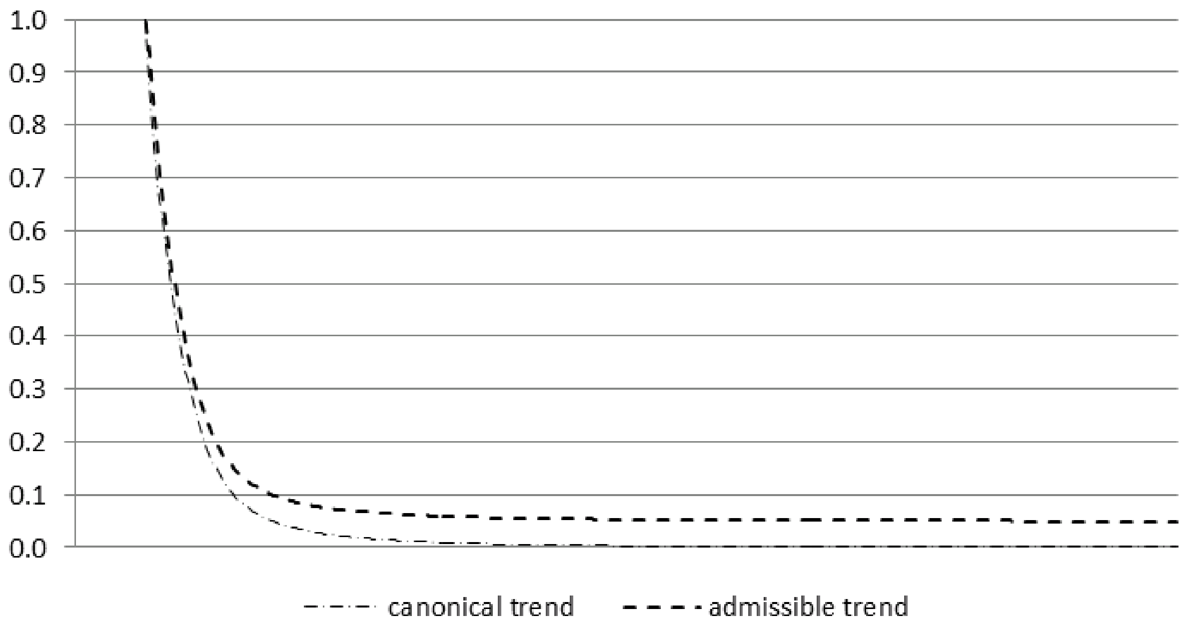


Figure 111: **A comparison of canonical trend and admissible trend**

A pseudo-spectrum is denoted by $g_i(\omega)$, where ω represents the angular frequency. The pseudo-spectrum of x_{it} is defined as the Fourier transform of ACGF of x_t which is expressed as:

$$\frac{\psi_i\left(B\right)\psi_i\left(F\right)}{\delta_i\left(B\right)\delta_i\left(F\right)}V(a_i) \tag{0.34}$$

where:

[33]HILLMER, S.C., and TIAO, G.C. (1982).
[34]GÓMEZ, V., and MARAVALL, A. (2001a).
[35]HILLMER, S.C., and TIAO, G.C. (1982).
[36]MARAVALL, A. (1986).

- $\psi_i(F) = \frac{\theta_i(F)}{\phi_i(F)}$
- $\psi_i(B) = \frac{\theta_i(B)}{\phi_i(B)}$
- B is the backward operator,
- F is the forward operator.

A pseudo-spectrum for a monthly time series x_t is presented in the figure below: The pseudo-spectrum for a monthly series. The frequency $\omega = 0$ is associated with the trend, frequencies in the range $[0+\epsilon_1,\ \frac{\pi}{6}-\epsilon_2]$ with $[0+\epsilon_1,\ \frac{\pi}{6}-\epsilon_2]$ $\epsilon_1,\ \epsilon_2 > 0$ and $\epsilon_1 < \frac{\pi}{6}-\epsilon_2$ are usually associated with the business-cycle and correspond to a period longer than a year and bounded[37]. The frequencies in the range $[\frac{\pi}{6}, \pi]$ are associated with the short term movements, whose cycle is completed in less than a year. If a series contains an important periodic component, its spectrum reveals a peak around the corresponding frequency and in the ARIMA model it is captured by an AR root. In the example below spectral peaks occur at the frequency $\omega = 0$ and at the seasonal frequencies ($\frac{\pi}{6}$, $\frac{2\pi}{6}$, $\frac{3\pi}{6}$, $\frac{4\pi}{6}$, $\frac{5\pi}{6}$, π). [38]

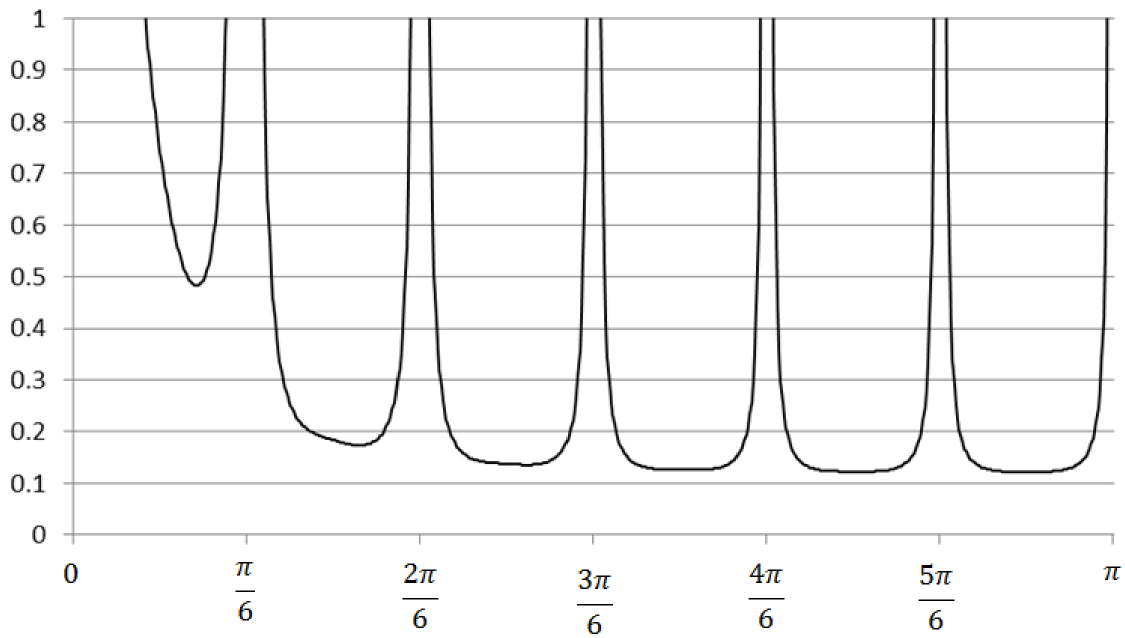


Figure 112: **The pseudo-spectrum for a monthly series**

In the decomposition procedure, the pseudo-spectrum of the time series x_t is divided into the spectra of its components (in the example figure below, four components were obtained).

[37] KAISER, R., and MARAVALL, A. (2000).
[38] KAISER, R., and MARAVALL, A. (2000).

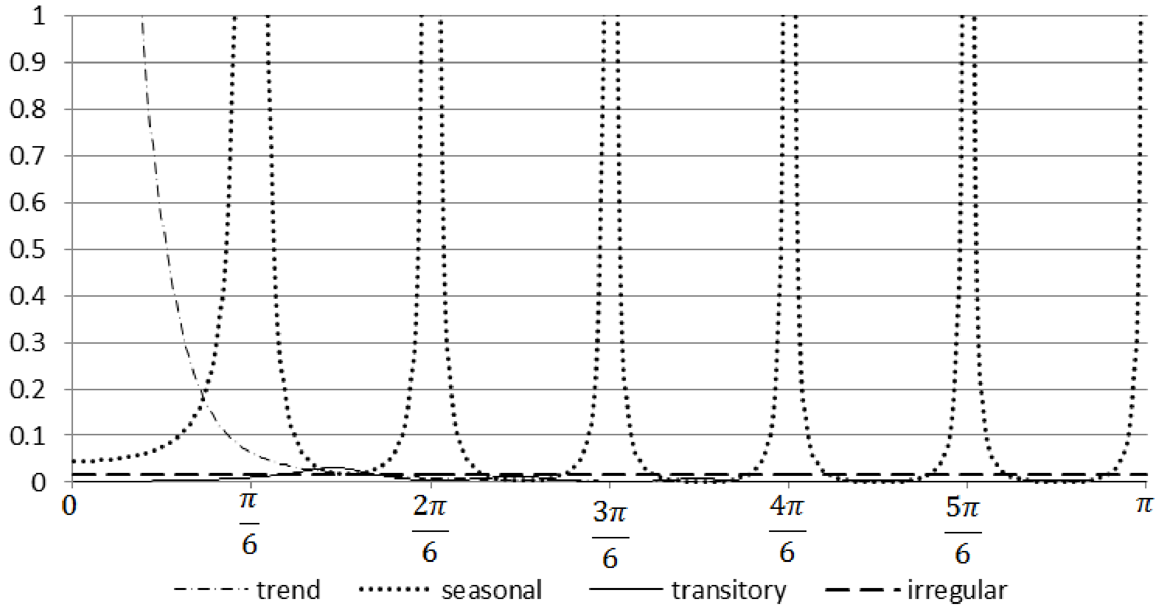


Figure 113: **The pseudo-spectra for the components**

Estimation of the components with the Wiener-Kolmogorow filter

The various components are estimated using Wiener-Kolmogorow (WK) filters. JDemetra+ includes three options to estimate the WK filter, namely *Burman*, *KalmanSmoother* and *MCElroyMatrix*[39]. Here the first of abovementioned options, proposed by BURMAN, J.P. (1980) will be explained.

The estimation procedure and the properties of the WK filter are easier to explain with a two-component model. Let the seasonally adjusted series (s_t) be the signal of interest and the seasonal component (n_t) be the remainder, "the noise". The series is given by the model Equation **??** and from Equation **??** the models for theoretical components are:

$$\varphi_s(B)s_t = \theta_s(B)a_{\text{st}} \tag{0.35}$$

and

$$\varphi_n(B)n_t = \theta_n(B)a_{\text{nt}} \tag{0.36}$$

[39]The choice of the estimation method is controlled by the *Method* parameter, explained in the SEATS specification section.

From Equation **??** and Equation **??** it is clear that $\varphi(B) = \varphi_s(B)\varphi_n(B)$ and $\theta(B)\, a_t = \theta_s(B)a_{\text{st}} + \theta_n(B)a_{\text{nt}}$.

As the time series components are never observed, their estimators have to be used. Let us note X_T an infinite realization of the time series x_t. SEATS computes the Minimum Mean Square Error (MMSE) estimator of s_t, e.g. the estimator $\hat{s}_t$ that minimizes $E[\left(s_t - \hat{s}_t\right)^2 | X_T\big)]$. Under the normality assumption $\hat{s}_{t|T}$ is also equal to the conditional expectation $E\left(s_t | X_T\right)$, so it can be presented as a linear function of the elements in X_T.[40] WHITTLE (1963) shows that the MMSE estimator of $\hat{s}_t$ is:

$$\hat{s}_t = k_s \frac{\psi_s(B)\psi_s(F)}{\psi(B)\psi(F)} x_t \tag{0.37}$$

where

- $\psi(B) = \frac{\theta(B)}{\phi(B)}$,
- $F = B^{-1}$,
- $k_s = \frac{V(a_s)}{V(a)}$,

$V(a_s)$ is the variance of a_{st} and $V(a)$ is the variance of a_t.

Expressing the $\psi(B)$ polynomials as functions of the AR and MA polynomials, after cancelation of roots, the estimator of s_t can be expressed as:

$$\hat{s}_t = k_s \frac{\theta_s(B)\,\theta_s(F)\,\varphi_n(B)\,\delta_n(B)\,\varphi_n(F)\,\delta_n(F)}{\theta(B)\,\theta(F)} x_t \tag{0.38}$$

where:

$$\nu_s(B, F) = k_s \frac{\theta_s(B)\,\theta_s(F)\,\varphi_n(B)\,\delta_n(B)\,\varphi_n(F)\,\delta_n(F)}{\theta(B)\,\theta(F)} \tag{0.39}$$

is a WK filter.

Equation Equation **??** shows that the WK filter is two-sided (uses observations both from the past and from the future), centered (the number of points in the past is the same as in the future) and symmetric (for any k the weight applied to x_{t-k} and x_{t+k} is the same), which allows the phase effect to be avoided. Due to invertibility of $\theta(B)$ (and $\theta(F)$) the filter is convergent in the past and in the future.

The estimator can be presented as

[40] MARAVALL, A. (2008c).

$$\hat{s}_t = \nu_i\left(B, F\right) x_t \tag{0.40}$$

where $\nu_i\left(B, F\right) = \nu_0 + \sum_{j=1}^{\infty} \nu_{ij}(B^j + F^j)$ is the WK filter.

The example of the WK filters obtained for the pseudo-spectra of the series illustrated above is shown on the figure below: WK filters for components.

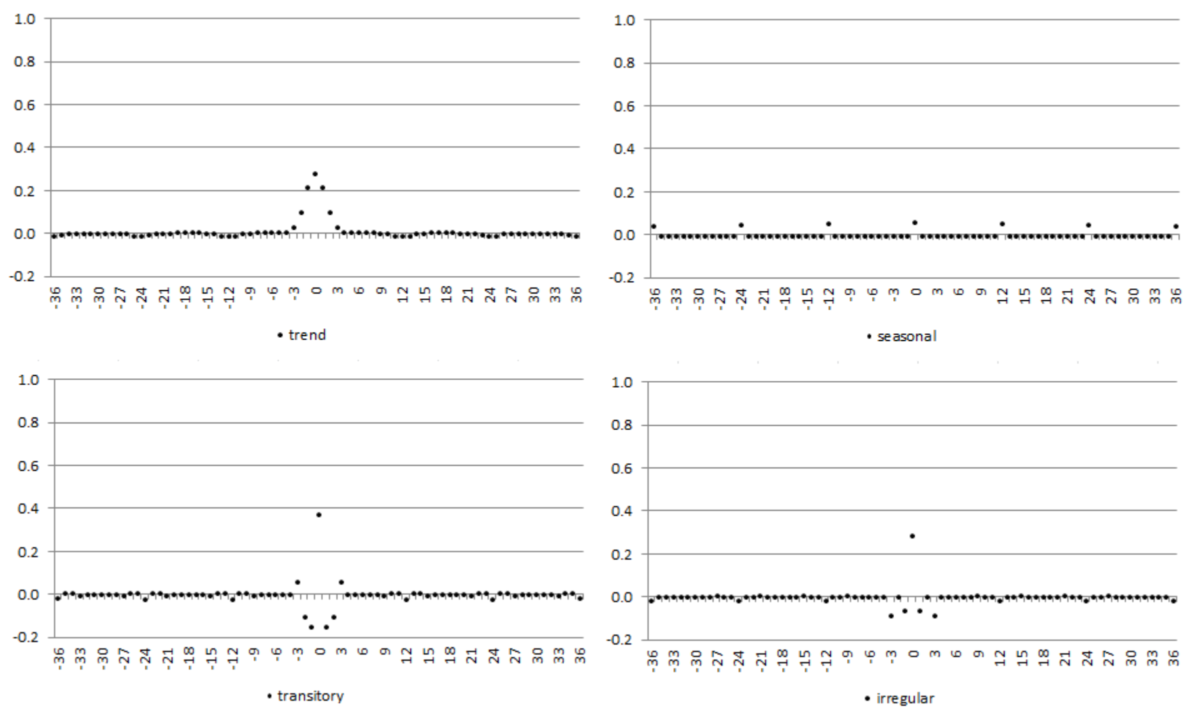


Figure 114: **WK filters for components**

The WK filter from Equation **??** can also be expressed as a ratio of two pseudo-autocovariance generating functions (p-ACGF). The p-ACGF function summarizes the sequence of absolutely summable autocovariances of a stationary process x_t (see section section Spectral Analysis).

The ACGF function of an ARIMA process is expressed as:

$$acgf(B) = \frac{\theta\left(B\right)\theta\left(F\right)}{\phi\left(B\right)\delta\left(B\right)\phi\left(F\right)\delta\left(F\right)} V(a) \tag{0.41}$$

And, the WK filter can be rewritten as:

$$\nu_s\left(B, F\right) = \frac{\gamma_s(B, F)}{\gamma(B, F)} \tag{0.42}$$

where:

- $\gamma_s(B,F) = \frac{\theta_s(B)\theta_s(F)}{\phi_s(B)\delta_s(B)\phi_s(F)\delta_s(F)} V(a_s)$ is the p-ACGF of s_t;
- $\gamma(B,F) = \frac{\theta(B)\theta(F)}{\phi(B)\delta(B)\phi(F)\delta(F)} V(a)$ is the p-ACGF of x_t.

From Equation **??** it can be seen that the WK filter depends on both the component and the series models. Consequently, the estimator of the component and the WK filter reflect the characteristic of data and by construction, the WK filter adapts itself to the series under consideration. Therefore, the ARIMA model is of particular importance for the SEATS method. Its misspecification results in an incorrect decomposition.

This adaptability, if the model has been correctly determined, avoids the dangers of under and overestimation with an ad-hoc filtering. For example, for the series with a highly stochastic seasonal component the filter adapts to the width of the seasonal peaks and the seasonally adjusted series does not display any spurious seasonality[41]. Examples of WK filters for stochastic and stable seasonal components are presented on the figure below.

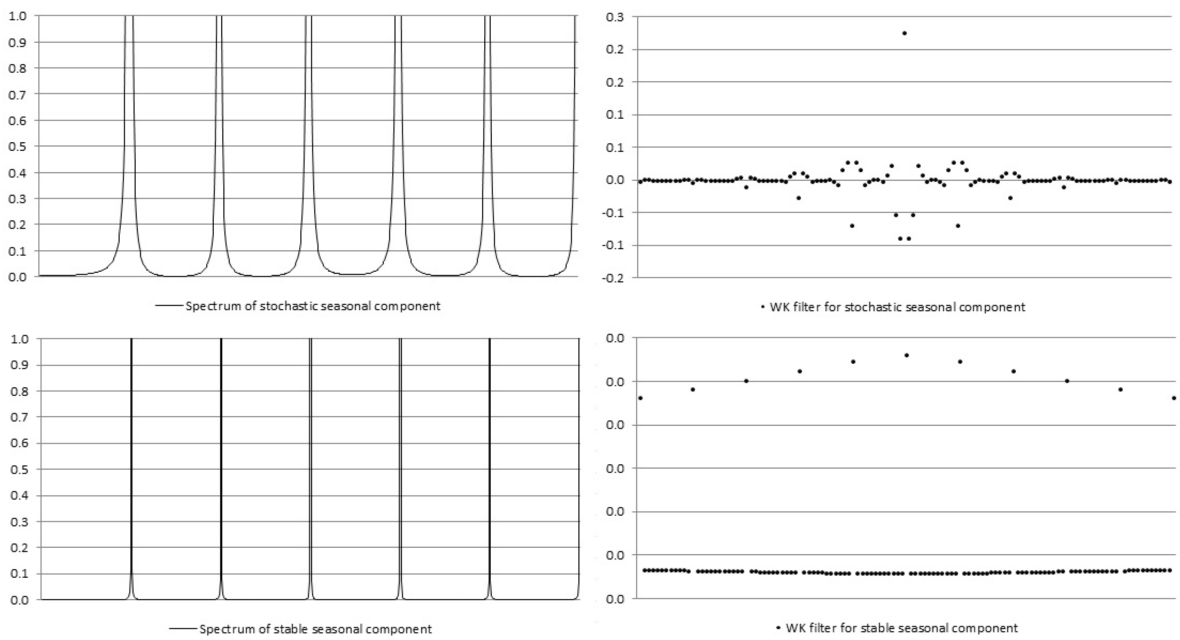


Figure 115: **WK filters for stable and stochastic seasonal components**

The derivation of the components requires an infinite realization of x_t in the direction of the past and of the future. However, the convergence of the WK filter guarantees that, in practice, it could be approximated by a truncated (finite) filter and, in most applications, for large k

[41] MARAVALL, A. (1995).

the estimator for the central periods of the series can be safely seen as generated by the WK filter[42]:

$$\hat{s}_t = \nu_k x_{t-k} + ... + \nu_0 x_t + ... + \nu_k x_{t+k} \tag{0.43}$$

When $T > 2L + 1$, where T is the last observed period, and L is an a priori number that typically expands between 3 and 5 years, the estimator expressed by Equation **??** can be assumed as the final (historical) estimator for the central observations of the series[43]. In practice, the Wiener-Kolmogorov filter is applied to x_t extended with forecasts and backcasts from the ARIMA model. The final or historical estimator of $\hat{s}_t$, is obtained with a doubly infinite filter, and therefore contains an error e_{st} called final estimation error, which is equal $e_{st} = s_t - \hat{s}_t$.

In the frequency domain, the Wiener-Kolmogorov filter $\nu(B, F)$ that provides the final estimator of s_t is expressed as the ratio of the s_t and x_t pseudo-spectra:

$$\tilde{\nu}\left(\omega\right) = \frac{g_s(\omega)}{g_x(\omega)} \tag{0.44}$$

The function $\tilde{\nu}\left(\omega\right)$ is also referred as the gain of the filter.[44] GÓMEZ, V., and MARAVALL, A. (2001a) show that when for some frequency the signal (the seasonally adjusted series) dominates the noise (seasonal fluctuations) the gain $\tilde{\nu}\left(\omega\right)$ approaches 1. On the contrary, when for some frequency the noise dominates the gain $\tilde{\nu}\left(\omega\right)$ approaches 0.

The spectrum of the estimator of the seasonal component is expressed as:

$$g_{\hat{s}}\left(\omega\right) = \left[\frac{g_s(\omega)}{g_x(\omega)}\right]^2 g_x(\omega) \tag{0.45}$$

where

- $\left[\tilde{\nu}\left(\omega\right)\right]^2 = \left[\frac{g_s(\omega)}{g_x(\omega)}\right]^2 = \left[\frac{g_s(\omega)}{g_s(\omega)+g_n(\omega)}\right]^2 = \left[\frac{1}{1+\frac{1}{r(\omega)}}\right]^2$ is the squared gain of the filter ;
- $r\left(\omega\right) = \frac{g_s(\omega)}{g_n(\omega)}$ represents the signal-to-noise ratio.

For each ω, the MMSE estimation gives the signal-to-noise ratio. If this ratio is high, then the contribution of that frequency to the estimation of the signal will be also high. Assume that the trend is a signal that needs to be extracted from a seasonal time series. Then $R\left(0\right) = 1$ and the frequency $\omega = 0$ will only be used for trend estimations. For seasonal frequencies $R\left(\omega\right) = 0$, so that these frequencies are ignored in computing the trend resulting in spectral zeros in $g_{\hat{s}}\left(\omega\right)$.

[42]MARAVALL, A., and PLANAS, C. (1999).
[43]MARAVALL, A. (1998).
[44]GÓMEZ, V., and MARAVALL, A. (2001a).

For this reason, unlike the spectrum of the component, the component spectrum contains dips as it can be seen on the figure below: Component spectrum and estimator spectrum for trend.

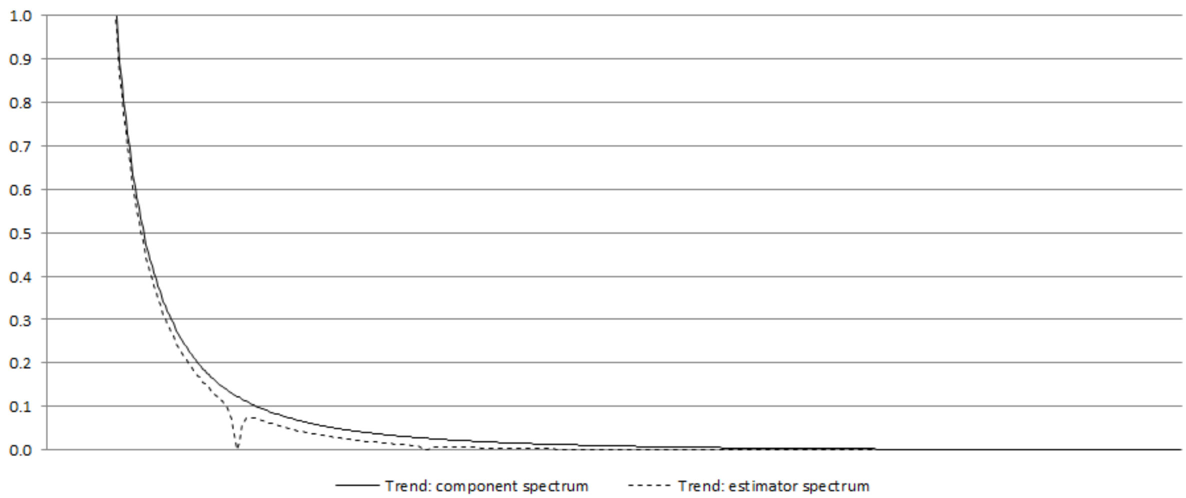


Figure 116: **Component spectrum and estimator spectrum for trend**

From the equation Equation **??** it is clear that the squared gain of the filter determines how the variance of the series contributes to the variance of the seasonal component for the different frequencies. When $\tilde{\nu}(\omega) = 1$, the full variation of x_t for that frequency is passed to $\hat{s}_t$, while if $\tilde{\nu}(\omega) = 0$ the variation of x_t for that frequency is fully ignored in the computation of $\hat{s}_t$. These two cases are well illustrated by the figure below that shows the square gain of the WK filter for two series already analysed in the figure above (Figure: WK filters for stable and stochastic seasonal components).

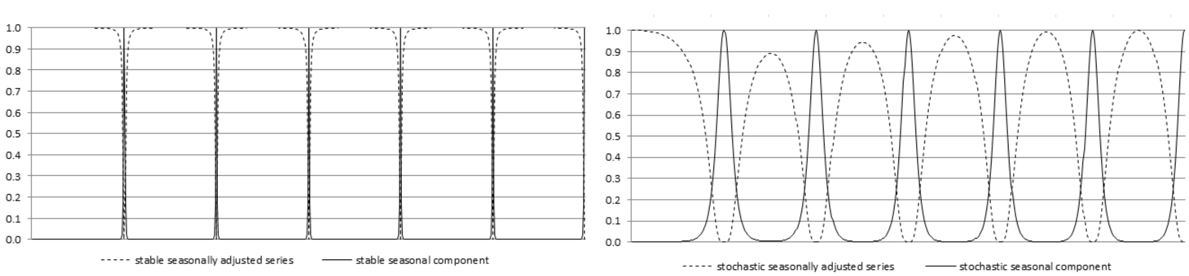


Figure 117: **The squared gain of the WK filter for stable and stochastic seasonal components.**

Since $r(\omega) \geq 0$, then $\tilde{\nu}(\omega) \leq 1$ and from Equation **??** it can be derived that $g_{\hat{s}}(\omega) = \tilde{\nu}(\omega)\, g_s(\omega)$. As a result, the estimator will always underestimate the component, i.e. it will be always more stable that the component.[45]

Since $g_{\widehat{n}}(\omega) < g_n(\omega)$ and $g_{\hat{s}}(\omega) < g_s(\omega)$ the expression: $g_x(\omega) - [g_{\widehat{n}}(\omega) + g_{\hat{s}}(\omega)] \geq 0$ is the cross-spectrum. As it is positive, the MMSE yields correlated estimators. This effect emerges since variance of estimator is smaller than the variance of component. Nevertheless, if at least one non-stationary component exists, cross-correlations estimated by TRAMO-SEATS will tend to zero as cross-covariances between estimators of the components are finite. In practice, the inconvenience caused by this property will likely be of little relevance.

Preliminary estimators for the components

GÓMEZ, V., and MARAVALL, A. (2001a) point out that *the properties of the estimators have been derived for the final (or historical) estimators. For a finite (long enough) realization, they can be assumed to characterize the estimators for the central observations of the series, but for periods close to the beginning of the end the filter cannot be completed and some preliminary estimator has to be used.* Indeed, the historical estimator shown in Equation **??** is obtained for the central periods of the series. However, when t approaches T (last observation), the WK filter requires observations, which are not available yet. For this reason a preliminary estimator needs to be used.

To introduce preliminary estimators let us consider a semi-finite realization $[x_{-\infty}, \ldots x_T]$, where T is the last observed period. The preliminary estimator of x_{it} obtained at T, $(T - t = k \geq 0)$ can be expressed as

$$\hat{x}_{it|t+k} = \nu_i(B, F)\, x^e_{t|T} \tag{0.46}$$

where

- $\nu_i(B, F)$ is the WK filter ;
- $x^e_{t|T}$ is the extended series, such that $x^e_{t|T} = x_t$ for $t \leq T$ and $x^e_{t|T} = \hat{x}_{t|T}$ for $t > T$, where $\hat{x}_{t|T}$ denotes the forecast of x_t obtained at period T.

The future k values necessary to apply the filter are not yet available and are replaced by their optimal forecasts from the ARIMA model on x_t. When $k = 0$ the preliminary estimator becomes the concurrent estimator. As the forecasts are linear functions of present and past observations of x_t, the preliminary estimator $\hat{x}_{it}$ will be a truncated asymmetric filter applied to x_t that generates a phase effect[46].

When a new observation x_{T+1} becomes available the forecast $\hat{x}_{T+1|T}$ is replaced by the observation and the forecast $\hat{x}_{iT+j|T}$, $j > 1$ are updated to $x_{T+j|T+1}$ resulting in the revision error[47]. The total error in the preliminary estimator $d_{it|t+k}$ is expressed as a sum of the final estimation error (e_{it}) and the revision error $(r_{it|t+k})$, i.e.:

[45] Ibid.
[46] KAISER, R., and MARAVALL, A. (2000).
[47] MARAVALL, A. (1995).

$$d_{it|t+k} = x_{it} - \hat{x}_{it|t+k} = (x_{it} - \hat{x}_{it}) + \left(\hat{x}_{it} - \hat{x}_{it|t+k}\right) = e_{it} + r_{it|t+k} \tag{0.47}$$

where:

- $x_{it} - i^{th}$ component;
- $\hat{x}_{it|t+k}$- the estimator of x_{it} when the last observation is x_{t+k}.

Therefore the preliminary estimator is subject not only to the final error but also to a revision error, which are orthogonal to each other[48]. The revision error decreases as k increases, until it can be assumed equal to 0 for large enough k.

It's worth remembering that SEATS estimates the unobservable components of the time series so the "true" components are never observed. Therefore, MARAVALL, A. (2009) stresses that *the error in the historical estimator is more of academic rather than practical interest. In practice, interest centres on revisions. (…) the revision standard deviation will be an indicator of how far we can expect to be from the optimal estimator that will be eventually attained, and the speed of convergence of* $\theta(B)^{-1}$ *will dictate the speed of convergence of the preliminary estimator to the historical one.* The analysis of an error is therefore useful for making decision concerning the revision policy, including the policy for revisions and horizon of revisions.

PsiE-weights

The estimator of the component is calculated as $\hat{x}_{it} = \nu_s(B,F)\,x_t$. By replacing $x_{it} = \frac{\theta(B)}{\gamma(B)\delta(B)}a_t$, the component estimator can be expressed as[49]:

$$\hat{x}_{it} = \xi_s(B,F)\,a_t \tag{0.48}$$

where $\xi_s(B,F) = ... + \xi_j B^j + ... + \xi_1 B + \xi_0 + \xi_{-1}F ... \xi_{-j}F^j + ...$

This representation shows the estimator as a filter applied to the innovation a_t, rather than on the series x_t[50]. Hence, the filter from Equation **??** can be divided into two components: the first one, i.e. $... + \xi_j B^j + ... + \xi_1 B + \xi_0$, applies to prior and concurrent innovations, the second one, i.e. $\xi_{-1}F + ... + \xi_{-j}F^j$ applies to future (i.e. posterior to t) innovations. Consequently, ξ_j determines the contribution of a_{t-j} to $\hat{s}_t$ while ξ_{-j} determines the contribution of a_{t+j} to $\hat{s}_t$. Finally, the estimator of the component can be expressed as:

$$\hat{x}_{it} = \xi_i(B)^{-}a_t + \xi_i(F)^{+}a_{t+1} \tag{0.49}$$

[48]MARAVALL, A. (2009).
[49]The section is based on KAISER, R., and MARAVALL, A. (2000).
[50]See section PsiE-weights. For further details see MARAVALL, A. (2008).

where:

- $\xi_i(B)^{-} a_t$ is an effect of starting conditions, present and past innovations in series;
- $\xi_i(F)^{+} a_{t+1}$ is an effect of future innovations.

For the two cases already presented in figure *WK filters for stable and stochastic seasonal components* and figure *The squared gain of the WK filter for stable and stochastic seasonal components* above, the psi-weights are shown in the figure below.

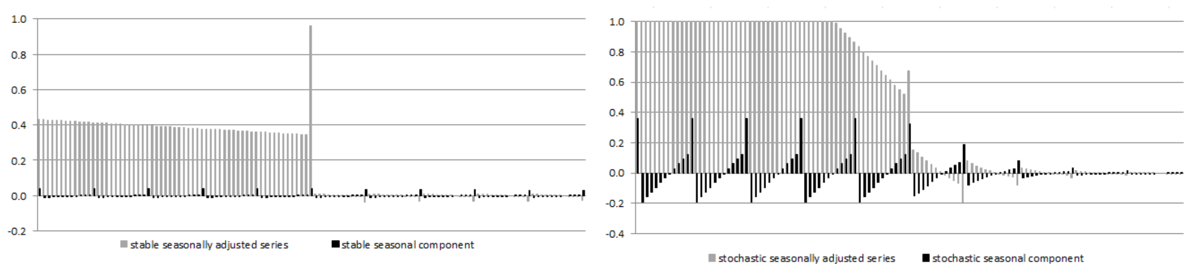


Figure 118: **WK filters and squared gain of the WK filter**

It can be shown that $\xi_{-1}, \dots, \xi_{-j}$ are convergent and $\xi_j, \dots, \xi_1, \xi_0$ are divergent. From Equation **??** , the concurrent estimator is equal to

$$\hat{x}_{it|t} = E_t x_{it} = E_t \hat{x}_{it} = \xi_i(B)^{-} a_t \tag{0.50}$$

so that the revision

$$r_{it} = \hat{x}_{it} - \hat{x}_{it|t} = \xi_i(F)^{+} a_{t+1} \tag{0.51}$$

is a zero-mean stationary MA process. As a result, historical and preliminary estimators are cointegrated. From expression Equation **??** the relative size of the full revision and the speed of convergence can be obtained.

Local Polynomials Methods for Trend Estimation

Under construction.

Tests

Introduction

This chapter describes all the tests available in JDemetra+, via Graphical User interface and/or R packages. An outline of the underlying theoretical principles of each test is provided.

The procedure to apply these tests in context is described when their use is relevant in the chapters dedicated to algorithms description, mainly on seasonal adjustment.

Tests on residuals

Test	Purpose	GUI	R package
Ljung-Box	autocorrelation	yes	
Box-Pierce	autocorrelation	yes	
Doornik-Hansen	normality	yes	rjd3tookit

Ljung-Box

The Ljung-Box Q-statistics are given by:

$$\text{LB}\left(k\right) = n \times \left(n+2\right) \times \sum_{k=1}^{K} \frac{\rho_{a,k}^{2}}{n-k} \tag{0.52}$$

where $\rho_{a,k}^{2}$ is the autocorrelation coefficient at lag k of the residuals $\hat{a}_t$, n is the number of terms in differenced (? differenciated ?) series, K is the maximum lag being considered, set in JDemetra+ to 24 (monthly series) or 8 (quarterly series).

If the residuals are random (which is the case for residuals from a well specified model), they will be distributed as $\chi_{(K-m)}^{2}$, where m is the number of parameters in the model which has been fitted to the data. (edit: not the residuals, but $\hat{\rho}$)

The Ljung-Box and Box-Pierce tests sometimes fail to reject a poorly fitting model. Therefore, care should be taken not to accept a model on a basis of their results. For the description of autocorrelation concept see section Autocorrelation function and partial autocorrelation function.

Box-Pierce

The Box-Pierce Q-statistics are given by:

$$\text{BP}\left(k\right) = n \sum_{k=1}^{K} \rho_{a,k}^{2}$$

where:

- $\rho_{a,k}^{2}$ is the autocorrelation coefficient at lag k of the residuals $\hat{a}_t$.
- n is the number of terms in differenced (differenciated?) series;
- K is the maximum lag being considered, set in JDemetra+ to 24 (monthly series) or 8 (quarterly series).

If the residuals are random (which is the case for residuals from a well specified model), they will be distributed as $\chi^2_{(K-m)}$ degrees of freedom, where m is the number of parameters in the model which has been fitted to the data.(edit: same as above)

Dornik-Hansen

The Doornik-Hansen test for multivariate normality (DOORNIK, J.A., and HANSEN, H. (2008)) is based on the skewness and kurtosis of multivariate data that is transformed to ensure independence. It is more powerful than the Shapiro-Wilk test for most tested multivariate distributions[51].

The skewness and kurtosis are defined, respectively, as: $s = \frac{m_3}{\sqrt{m_2}^3}$ and $k = \frac{m_4}{m_2^2}$,

where:

- $m_i = \frac{1}{n}\sum_{i=1}^{n}\left(x_i - \overline{x}\right)^i$;
- $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$;
- n is a number of (non-missing) residuals.

[51] The description of the test derives from DOORNIK, J.A., and HANSEN, H. (2008).

The Doornik-Hansen test statistic derives from SHENTON, L.R., and BOWMAN, K.O. (1977) and uses transformed versions of skewness and kurtosis.

The transformation for the skewness s into z_1 is as in D'AGOSTINO, R.B. (1970):

$$\beta = \frac{3(n^2+27n-70)(n+1)(n+3)}{(n-2)(n+5)(n+7)(n+9)}$$

$$\omega^2 = -1 + \sqrt{2(\beta - 1)}$$

$$\delta = \frac{1}{\sqrt{\log\left(\omega^2\right)}}$$

$$y = s\sqrt{\frac{(\omega^2-1)(n+1)(n+3)}{12(n-2)}}$$

$$z_1 = \delta log(y + \sqrt{y^2-1})$$

The kurtosis k is transformed from a gamma distribution to χ^2, which is then transformed into standard normal z_2 using the Wilson-Hilferty cubed root transformation:

$$\delta = (n-3)(n+1)(n^2+15n-4)$$

$$a = \frac{(n-2)(n+5)(n+7)(n^2+27n-70)}{6\delta}$$

$$c = \frac{(n-7)(n+5)(n+7)(n^2+2n-5)}{6\delta}$$

$$l = \frac{(n+5)(n+7)(n^3+37n^2+11n-313)}{12\delta}$$

$$\alpha = a + c \times s^2$$

$$\chi = 2l(k-1-s^2)$$

$$z_2 = \sqrt{9\alpha}\left(\frac{1}{9\alpha} - 1 + \sqrt[3]{\frac{\chi}{2\alpha}}\right)$$

Finally, the Doornik-Hansen test statistic is defined as the sum of squared transformations of the skewness and kurtosis. Approximately, the test statistic follows a χ^2 distribution, i.e.:

$$DH = z_1^2 + z_2^2 \sim \chi^2(2)$$

Seasonality tests

table with all tests by purpose and accessibility

Test	Purpose	GUI	R package
QS test	Autocorrelation at seasonal lags	yes	
F-test with seasonal dummies	Stable seasonality	yes	rjd3sa
Identification of spectral peaks	Seasonal frequencies	yes	rjd3sa
Friedman test	Stable seasonality	yes	rjd3sa
Two-way variance analysis	Moving seasonality	yes	

QS Test on autocorrelation at seasonal lags

The QS test is a variant of the Ljung-Box test computed on seasonal lags, where we only consider positive auto-correlations

More exactly,

$$QS = n\left(n+2\right)\sum_{i=1}^{k}\frac{\left[\max\left(0, \hat{\gamma}_{i\cdot l}\right)\right]^2}{n - i\cdot l}$$

where $k = 2$, so only the first and second seasonal lags are considered. Thus, the test would checks the correlation between the actual observation and the observations lagged by one and two years. Note that $l = 12$ when dealing with monthly observations, so we consider the autocovariances $\hat{\gamma}_{12}$ and $\hat{\gamma}_{24}$ alone. In turn, $k = 4$ in the case of quarterly data.

Under H0, which states that the data are independently distributed, the statistics follows a $\chi\,(k)$ distribution. However, the elimination of negative correlations makes it a bad approximation. The p-values would be given by $P(\chi^2\,(k) > Q)$ for $k = 2$. As $P(\chi^2(2)) > 0.05 = 5.99146$ and $P(\chi^2(2)) > 0.01 = 9.21034$, $QS > 5.99146$ and $QS > 9.21034$ would suggest rejecting the null hypothesis at 95% and 99% significance levels, respectively.

Modification

Maravall (2012) proposes approximate the correct distribution (p-values) of the QS statistic using simulation techniques. Using 1000K replications of sample size 240, the correct critical values would be 3.83 and 7.09 with confidence levels of 95% and 99%, respectively (lower than the 5.99146 and 9.21034 shown above). For each of the simulated series, he obtains the distribution by assuming $QS = 0$ when $\hat{\gamma}_{12}$, so in practice this test will detect seasonality only when any of these conditions hold: - Statistically significant positive autocorrelation at lag 12 - Non-negative sample autocorrelation at lag 12 and statistically significant positive autocorrelation at lag 24

Use

The test can be applied directly to any series by selecting the option *Statistical Methods » Seasonal Adjustment » Tools » Seasonality Tests*. This is an example of how results are displayed for the case of a monthly series:

1. Tests on autocorrelations at seasonal lags

Seasonality present

ac(12)=0.8238
ac(24)=0.7006

Distribution: Chi2 with 2 degrees of freedom
Value: 258.5028
PValue: 0.0000

Figure 119: qs

The test can be applied to the input series before any seasonal adjustment method has been applied. It can also be applied to the seasonally adjusted series or to the irregular component.

References

- LJUNG G. M. and G. E. P. BOX (1978). "On a Measure of a Lack of Fit in Time Series Models". Biometrika 65 (2): 297–303. doi:10.1093/biomet/65.2.297
- MARAVALL, A. (2011). "Seasonality Tests and Automatic Model Identification in Tramo-Seats". Manuscript

- MARAVALL, A. (2012). "Update of Seasonality Tests and Automatic Model Identification in TRAMO-SEATS". Bank of Spain (November 2012)

F-test on seasonal dummies

The F-test on seasonal dummies checks for the presence of deterministic seasonality. The model used here uses seasonal dummies (mean effect and 11 seasonal dummies for monthly data, mean effect and 3 for quarterly data) to describe the (possibly transformed) time series behaviour. The test statistic checks if the seasonal dummies are jointly statistically not significant. When this hypothesis is rejected, it is assumed that the deterministic seasonality is present and the test results are displayed in green.

This test refers to Model-Based χ^2 and F-tests for Fixed Seasonal Effects proposed by LYTRAS, D.P., FELDPAUSCH, R.M., and BELL, W.R. (2007) that is based on the estimates of the regression dummy variables and the corresponding t-statistics of the Reg-Arima model, in which the ARIMA part of the model has a form (0,1,1)(0,0,0). The consequences of a misspecification of a model are discussed in LYTRAS, D.P., FELDPAUSCH, R.M., and BELL, W.R. (2007).

For a monthly time series the Reg-Arima model structure is as follows:

$$(1-B)\left(y_t - \beta_1 M_{1,t} - ... - \beta_{11} M_{11,t} - \gamma X_t\right) = \mu + (1-B)a_t$$

where:

- $M_{j,t} = \begin{cases} 1 & \text{in month } j = 1, \ldots, 11 \\ -1 & \text{in December} \\ 0 & \text{otherwise} \end{cases}$ - dummy variables;
- y_t – the original time series;
- B – a backshift operator;
- X_t – other regression variables used in the model (e.g. outliers, calendar effects, user-defined regression variables, intervention variables);
- μ – a mean effect;
- a_t – a white-noise variable with mean zero and a constant variance.

In the case of a quarterly series the estimated model has a form:

$$(1-B)\left(y_t - \beta_1 M_{1,t} - ... - \beta_3 M_{3,t} - \gamma X_t\right) = \mu + (1-B)a_t \tag{0.53}$$

where: