RJDemetra: an R interface to JDemetra+

Introduction

RJDemetra is a R interface to JDemetra+, the seasonal adjustment software officially recommended to the members of the ESS and the European System of Central Banks.

JDemetra+ is a developed by the National Bank of Belgium (NBB) in cooperation with the Deutsche Bundesbank and Eurostat in accordance with the Guidelines of the European Statistical System (ESS). It implements the two leading seasonal adjustment methods TRAMO/SEATS+ and X-12ARIMA/X-13ARIMA-SEATS.

R is a programming language and free software environment for statistical computing and free software widely use by statisticians.

RJDemetra is a R package that offers full access to all options and outputs of JDemetra+. It also offers many possibilities to the users of JDemetra+ to implements new tools for the production of seasonally adjusted series thanks to all the libraries already available in R. It's available in github: https://github.com/nbbrd/RJDemetra.

Methods

RJDemetra relies on the Java libraries use in JDemetra+: the algorithms are not implemented inside the package. The link between R and Java libraries is done with the rJava package, which is a low-level R to Java interface. The consequence is that the results of the seasonal adjustment done in R are certified by the use of JDemetra+ and the system requirements needed to install the package are the same needed to use JDemetra+ (Java SE 8 or later).

The goal of the RJDemetra package is to offer a "pure R" package to the users, more familiar to this language rather that Java. It allows them to integrate easily the seasonal adjustment process it their production and offers the possibility to implements new tools that are difficult to integrate into the graphical interface of JDemetra+, for example:

- comparison of the direct and indirect aggregates adjustment;
- automatic generation of dashboards to summarise information for people responsible of the ongoing production but not of the maintenance of the seasonally adjusted models (and so non-JDemetra+ users);
- more easily implement quality report procedures.

Results

In the current version of the RJDemetra package, users can:

- seasonally adjust their time series with the TRAMO-SEATS and X-13-ARIMA methods;
- use the regARIMA preadjustment method implemented in TRAMO-SEATS and X-13-ARIMA methods;
- manipulate JDemetra+ workspace to easily switch from R models to the JDemetra+ graphical interface. The current functionalities are:
  - importing JDemetra+ workspace in R to get input raw series or the seasonally adjust model as defined in RJDemetra;
  - exporting R models created via RJDemetra to a readable JDemetra+ workspace.

All the seasonally adjust object created by RJDemetra are S3 classes with basic methods implements: `print()`, `plot()` and `summary()` (for regARIMA models).

Let's see an example with the French industrial production index in manufacturing (https://www.insee.fr/en/statistiques/serie/010537903). The time series can be seasonally adjusted with the X-13-ARIMA method by the function `x13_def()`: main results are presented in figure 1 created by the `plot()` function.

```r
library(RJDemetra)
x13_ipi <- x13_def(ipi_french, spec = "RSA3")
print(x13_ipi,enable_print_style = FALSE)
```

```
##
##
## RegARIMA
## y = regression model + arima (3, 0, 0, 0, 1, 1)
## Log-transformation: yes
## Coefficients:
##           Estimate Std. Error
## Phi(1)     0.06496      0.060
## Phi(2)    -0.22318      0.058
## Phi(3)    -0.53899      0.059
## BTheta(1) -0.87452      0.044
##
##               Estimate Std. Error
## Mean          0.003291      0.002
## LS (11-2008) -0.127329      0.016
##
##
## Residual standard error: 0.02923 on 212 degrees of freedom
## Log likelihood = 439.1, aic =  1107 aicc =  1107, bic(corrected for length) = -6.914
##
##
##
## Decomposition
## Monitoring and Quality Assessment Statistics:
##       M stats
## M(1)    0.407
## M(2)    0.377
## M(3)    3.000
## M(4)    0.062
## M(5)    3.000
## M(6)    0.526
## M(7)    0.142
## M(8)    0.330
## M(9)    0.065
## M(10)   0.311
## Q       0.845
## Q-M2    0.903
##
## Final filters:
## Seasonal filter:  3x5
## Trend filter:  23-Henderson
##
##
## Final
```

```
## Last observed values
##                y       sa        t         s         i
## Sep 2017 108.54 102.2883 103.7189 1.0611187 0.9862062
## Oct 2017 113.60 104.8975 103.7928 1.0829621 1.0106431
## Nov 2017 110.70 107.2133 103.8364 1.0325213 1.0325214
## Dec 2017  97.97 101.4844 103.8632 0.9653705 0.9770966
## Jan 2018 102.92 106.2211 103.8905 0.9689223 1.0224332
## Feb 2018  99.45 103.1341 103.9304 0.9642786 0.9923378
## Mar 2018 112.96 103.6153 103.9858 1.0901868 0.9964372
## Apr 2018 103.38 102.3360 104.0587 1.0102017 0.9834449
## May 2018  99.27 102.7787 104.1677 0.9658616 0.9866661
## Jun 2018 112.60 103.7506 104.3134 1.0852952 0.9946043
## Jul 2018 106.76 106.7344 104.4946 1.0002399 1.0214348
## Aug 2018  82.14 106.0951 104.6781 0.7742111 1.0135370
##
## Forecasts:
##                 y_f     sa_f      t_f       s_f       i_f
## Sep 2018 110.17751 104.1644 104.8277 1.0577272 0.9936725
## Oct 2018 113.97238 105.2478 104.9164 1.0828954 1.0031594
## Nov 2018 108.36831 104.5309 104.9059 1.0367111 0.9964247
## Dec 2018 100.66007 104.6131 104.8079 0.9622127 0.9981419
## Jan 2019 101.78559 104.7078 104.6226 0.9720916 1.0008141
## Feb 2019 101.18138 105.0994 104.3965 0.9627208 1.0067329
## Mar 2019 112.46321 103.0487 104.1593 1.0913602 0.9893369
## Apr 2019 105.06030 104.2347 103.9424 1.0079205 1.0028124
## May 2019 100.34816 103.6569 103.7215 0.9680800 0.9993770
## Jun 2019 112.79333 104.1444 103.4615 1.0830472 1.0066012
## Jul 2019 104.40231 104.4259 103.2479 0.9997741 1.0114093
## Aug 2019  78.32815 100.9817 103.0722 0.7756672 0.9797177
##
##
## Diagnostics
##  Relative contribution of the components to the stationary portion of the variance in the original s
##  Trend computed by Hodrick-Prescott filter (cycle length = 8.0 years)
##            Component
##  Cycle         2.148
##  Seasonal     66.591
##  Irregular     4.205
##  TD & Hol.     0.000
##  Others       27.124
##  Total       100.068
##
##  Residual seasonality tests
##                                      P.value
##  qs test on sa                         0.017
##  qs test on i                             NA
##  f-test on sa (seasonal dummies)       0.930
##  f-test on i (seasonal dummies)           NA
##  Residual seasonality (entire series)   0.941
##  Residual seasonality (last 3 years)    0.912
##  f-test on sa (td)                     0.000
##  f-test on i (td)                         NA
##
##  Combined test in the entire series
```

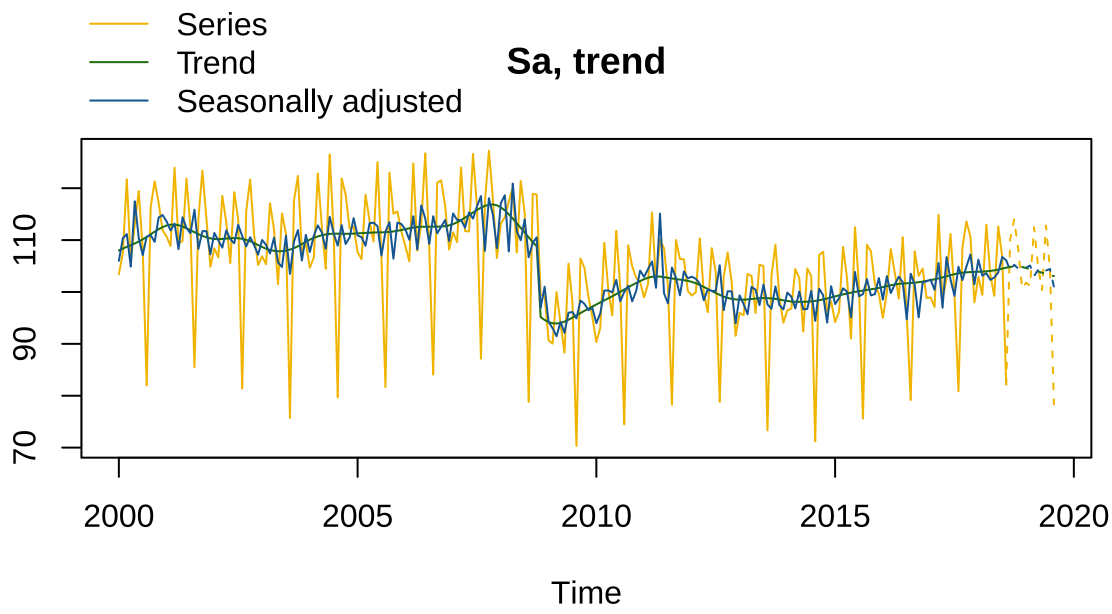

Figure 1: The result of the seasonal adjustment process for the french IPI in manufacturing

```
##  Non parametric tests for stable seasonality
##                                                              P.value
##    Kruskall-Wallis test                                        0.000
##    Test for the presence of seasonality assuming stability     0.000
##    Evolutive seasonality test                                  0.955
##
##  Identifiable seasonality present
##
##  Combined test in the last 3 years
##  Non parametric tests for stable seasonality
##                                                              P.value
##    Kruskall-Wallis test                                        0.002
##    Test for the presence of seasonality assuming stability     0.000
##    Evolutive seasonality test                                  0.212
##
##  Identifiable seasonality probably present
##
##
## Additional output variables
```

```
plot(x13_ipi, type_chart = "sa-trend")
```

More complex figures are also implemented to have more details on the decomposition of the model, like S-I-ratio (figures 2).

```
plot(x13_ipi$decomposition, ask = FALSE)
```

Other R libraries can also be used on the result of the seasonal adjustment model, not available in JDemetra+. So, for example, the Diebold-Mariano test (implemented in the forecast package) can be used to compare the forecast accuracy of two regARIMA models,as the Shapiro-Wilk test of normality on the residuals of a

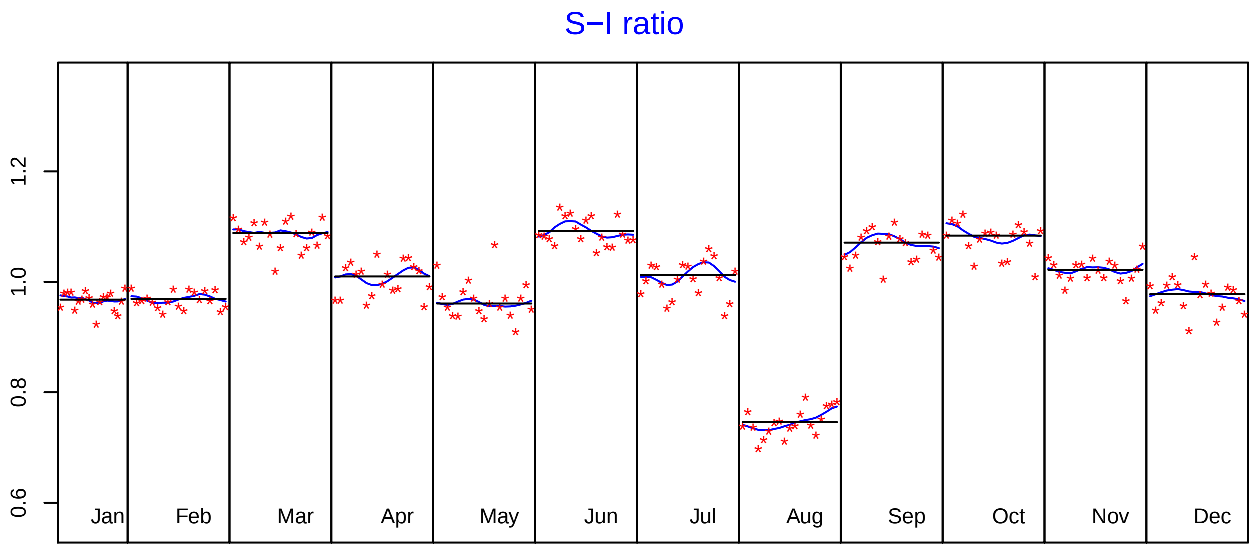

Figure 2: S-I ratio

regARIMA model:

```
shapiro.test(x13_ipi$regarima$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  x13_ipi$regarima$residuals
## W = 0.99401, p-value = 0.5564
```

Conclusions

Implemen