STATISTICS AUSTRIA
The Information Manager

Angelika Meraner, Gregor de Cillia, Markus Fröhlich, Alexander Kowarik
Statistics Austria

Vienna
December 2020

persephone

Seasonal Adjustment with an Object-oriented Wrapper for RJDemetra

Objective

Improve or replace R-package x12 currently used at Statistics Austria for seasonal adjustment.

Requirements:

- Easy processing of multiple time series during the production of seasonally adjusted estimates in an offical statistics setting
- Support of hierarchical time series
- Weighted aggregate series
- R environment

Release of **RJDemetra**, an R interface to JDemetra+ → Availability of Eurostat-recommended JDemetra+ seasonal adjustment (SA) software for R users.

→ Build wrapper around **RJDemetra** to fit our needs

Overview

R-package **persephone** builds on top of **RJDemetra**.

- **persephone** provides SA-infrastructure for official statistics, i.e. dealing with multiple (hierarchical) monthly/quarterly time series
- Functions of **RJDemetra** performing SA are called in the background.
- Available on `https://github.com/statistikat/persephone` (still under development)

Constructing Persephone Objects

- Objects of class `persephone` can be constructed from only one or multiple time series.
- Multiple time series can be connected through a certain hierarchy and weights can be assigned.
- Batch objects without a hierarchy can be represented as objects with a flat hierarchy at the moment (this will be improved in the future).

1. ‘Single’ persephone objects are constructed with the functions `per_x13()` or `per_tramo()` depending on the choice of SA method (X-13-ARIMA-SEATS or TRAMO-SEATS).
2. Multiple ‘single’ objects can then be combined with `per_hts()` to build a hierarchical persephone object.

Examples

Examples in the following slides:

- Example 1: Persephone Single Object
- Example 2: Persephone Hierarchical Object

Example 1: Persephone Single Object

Create a persephone ‘single’ object from a univariate time series object.

- A predefined JDemetra+ model specification has to be provided by the user, e.g. `"RSA3"` (Log/level detection = automatic, Outliers detection = automatic, Calendar effects = NA, ARIMA = automatic).

```r
data(AirPassengers, package = "datasets")
obj_x13 <- per_x13(AirPassengers, "RSA3")
```

- Now different methods can be called for the persephone object `obj_x13`, most importantly the `run()` method which performs the SA.

```r
obj_x13$run()
```

Example 1: Persephone Single Object

- The `output` field gives access to the original output object from **RJDemetra**.

```
obj_x13$output$regarima
```

```
## y = regression model + arima (0, 1, 1, 0, 1, 1)
## Log-transformation: yes
## Coefficients:
## Estimate Std. Error
## Theta(1)  -0.4018 0.081
## BTheta(1)  -0.5569 0.078
##
##
## Residual standard error: 0.03672 on 128 degrees of freedom
## Log likelihood = 244.7, aic = 987.2 aicc = 987.4,
bic(corrected for length) = -6.535
```

Example 1: Persephone Single Object

- We can change the predefined JD+ parameter settings with the `updateParams()` method. In this example we include some random outliers.

```
obj_x13$updateParams(usrdef.outliersEnabled = TRUE,
                     usrdef.outliersType = c("AO","LS","LS"),
                     usrdef.outliersDate=c("1950-01-01",
                                           "1955-04-01",
                                           "1959-10-01"))
obj_x13$run()
```

Example 1: Persephone Single Object

The output shows that the outliers are now included in the model.

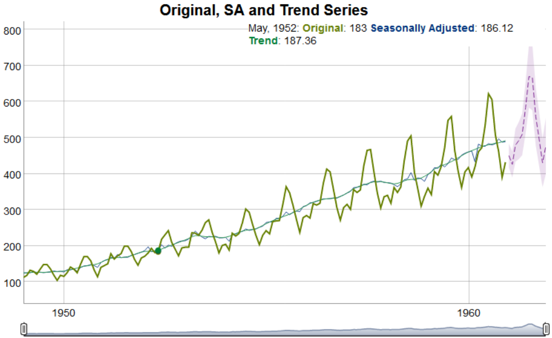
```
obj_x13$output$regarima
```

```
## y = regression model + arima (0, 1, 1, 0, 1, 1)
## Log-transformation: yes
## Coefficients:
## Estimate Std. Error
## Theta(1) -0.3914 0.082
## BTheta(1) -0.5619 0.079
##
## Estimate Std. Error
## AO (1-1950) -0.056775 0.029
## LS (4-1955) 0.023172 0.030
## LS (10-1959) 0.001298 0.031
##
##
## Residual standard error: 0.03607 on 125 degrees of freedom
## Log likelihood = 247, aic = 988.6 aicc = 989.3, bic(corrected
for length) = -6.458
```

Example 1: Persephone Single Object

- Several plot methods have been implemented with the focus on using interacive tools, e.g. the default S3 generic `plot()` shows a zoomable line representation of the series.

```
plot(obj_x13)
```



Example 1: Persephone Single Object

Other plot functions:

- `plotResiduals()`: Several interactive plots in connection with residuals for a persephone object
- `plotSeasIrrCal()`: Interactive plot of the seasonal component, irregular component and calendar effects for a persephone object
- `plotSiRatios()`: Interactive plot of SI-ratios (and, in case of x11, of seasonal factors) by period
- `plotSpectrum()`: Spectral Plots

Example 2: Persephone Hierarchical Object

Create a hierarchical persephone object e.g. from a list of persephone 'single' objects.

- Data included in the package to be used for examples and tests:
  - `pi_caladj, pi_sa, pi_unadj`. They are multiple time series objects of the volume index of production for total industry (except construction) of the EU-28 countries from Jan 2000 to June 2019 (monthly data published by Eurostat).
  - The corresponding weights `weights_pi_ea19, weights_pi_eu28`.
- For this example we use the calendar adjusted data `pi_caladj` after performing some preadjustments.

Example 2: Persephone Hierarchical Object

Preadjustments because of missing values and for the sake of readability.

```r
pi_caladj_orig <- pi_caladj
pi_caladj <- pi_caladj[, -c(1:2)]
pi_caladj <- window(pi_caladj, end = c(2019, 5))
# Euro-Area Countries
ea19 <- weights_pi_ea19$country
# EU-28 Countries
eu28 <- weights_pi_eu28$country
```

Example 2: Persephone Hierarchical Object

- We generate a list of persephone single objects with x13 as method for all countries' time series.

```
ts_28 <- lapply(pi_caladj, per_x13, template = "RSA3")
```

- We aggregate the Euro-area (EA-19) countries and set the method to be used for the direct adjustment of the aggregate series to x13 as well.

```
hts_EA19 <- per_hts(list = ts_28[ea19], method = "x13")
```

Example 2: Persephone Hierarchical Object

- We then generate our final hierarchical persephone object `hts_EU28` which consists of the Euro-area countries as a hierarchical object and the remaining 9 countries as single objects.

```
non_ea19 <- eu28[which(!eu28 %in% ea19)]
non_ea19
```

```
## [1] "BG" "CZ" "DK" "HR" "HU" "PL" "RO" "SE" "UK"
```

```
hts_EU28 <- per_hts(list = c(EA19 = hts_EA19, ts_28[non_ea19]))
```

Example 2: Persephone Hierarchical Object

- The structure of this object is represented in the `print` output. The "blank" component is the overall total.

```
hts_EU28
```

```
##  component run   class
##            FALSE hierarchicalTimeSeries
##  EA19      FALSE hierarchicalTimeSeries
##  EA19/BE   FALSE x13Single
##  EA19/DE   FALSE x13Single
##  EA19/EE   FALSE x13Single
##  EA19/IE   FALSE x13Single
##  EA19/EL   FALSE x13Single
##  EA19/ES   FALSE x13Single
##  EA19/FR   FALSE x13Single
##  EA19/IT   FALSE x13Single
##  EA19/CY   FALSE x13Single
##  EA19/LT   FALSE x13Single
##  EA19/LV   FALSE x13Single
##  EA19/LU   FALSE x13Single
##  EA19/MT   FALSE x13Single
##  EA19/NL   FALSE x13Single
##  EA19/AT   FALSE x13Single
##  EA19/PT   FALSE x13Single
##  EA19/SI   FALSE x13Single
##  EA19/SK   FALSE x13Single
##  EA19/FI   FALSE x13Single
##  BG        FALSE x13Single
##  CZ        FALSE x13Single
##  DK        FALSE x13Single
##  HR        FALSE x13Single
##  HU        FALSE x13Single
```

Example 2: Persephone Hierarchical Object

The following methods can be called for the hierarchical persephone object:

- perform a run, i.e. perform SA.

```
hts_EU28$run()
```

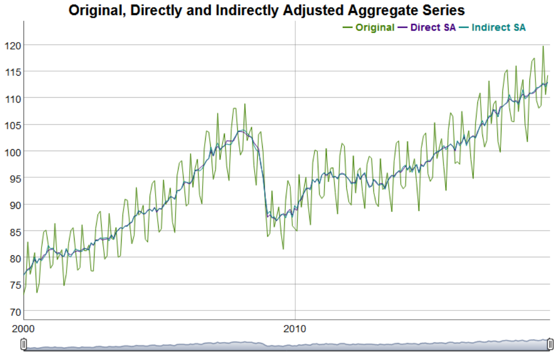
- generate a plot comparing original, directly and indirectly adjusted series.
- generate a Eurostat Quality Report.

More methods directed specifically at multiple (hierarchical) time series still need to be implemented/developed.

Example 2: Persephone Hierarchical Object

- General comparison line chart called through the S3 generic `plot()` (only plot function for hierarchical persephone objects at the moment)

```
plot(hts_EU28)
```



Example 2: Persephone Hierarchical Object

- Generate Eurostat Quality Report with the function `generate_qr_table()`.

```r
head(generate_qr_table(hts_EU28), n = 4)
```

```
##   component Method Period Nobs  Start    End Log.Transformation    ARIMA.Model
## 1               TS     12  233 1-2000 5-2019               TRUE (0 1 1)(0 1 1)
## 2      EA19    x13     12  233 1-2000 5-2019              FALSE (1 1 0)(0 1 1)
## 3   EA19/BE    x13     12  233 1-2000 5-2019              FALSE (0 1 1)(0 1 1)
## 4   EA19/DE    x13     12  233 1-2000 5-2019               TRUE (1 1 2)(0 1 1)
##   LeapYear MovingHoliday NbTD Noutliers     Outlier1     Outlier2    Outlier3
## 1    FALSE          TRUE    6         4 LS (11-2008) LS (12-2008) LS (1-2009)
## 2     TRUE          TRUE    7         2 LS (11-2008) TC (11-2008)        <NA>
## 3    FALSE         FALSE    0         2 LS (11-2008)  AO (5-2009)        <NA>
## 4    FALSE         FALSE    0         2  LS (1-2009) LS (11-2008)        <NA>
##   CombinedTest_SI Residual.Seasonality Residual.TD.Effect Q.Stat
## 1         Present                   No                 No     NA
## 2         Present                   No                 No   0.16
## 3         Present                   No                Yes   0.35
## 4         Present                   No                Yes   0.22
##   Final.Henderson.Filter Stage.2.Henderson.Filter Seasonal.Filter Max.Adj
## 1                   <NA>                     <NA>            <NA>     14%
## 2                    H13                      H13             3x5     19%
## 3                    H13                      H13             3x5     14%
## 4                    H13                      H13             3x5      9%
```

Example 2: Persephone Hierarchical Object

Methods developed for univariate analysis can be used by accessing the components of a hierarchical persephone object.

Example: Generate Eurostat Quality Report with the function `generate_qr_table()` for only one of the components, e.g. the persephone 'single' object for Belgium.

```r
generate_qr_table(hts_EU28$get_component("EA19/BE"))
```

```
##    component Method Period Nobs   Start    End
## 1              x13      12  233 1-2000 5-2019
##   Log.Transformation    ARIMA.Model LeapYear MovingHoliday
## 1              FALSE (0 1 1)(0 1 1)    FALSE         FALSE
##   NbTD Noutliers     Outlier1    Outlier2 Outlier3
## 1    0         2 LS (11-2008) AO (5-2009)     <NA>
##   CombinedTest_SI Residual.Seasonality Residual.TD.Effect
## 1         Present                   No                Yes
##   Q.Stat Final.Henderson.Filter Stage.2.Henderson.Filter
## 1   0.35                    H13                      H13
##   Seasonal.Filter Max.Adj
## 1             3x5     14%
```

Example 2: Persephone Hierarchical Object

With **persephone** we can also generate weighted aggregates, e.g. for indices.

The `weights` argument of `per_hts()` is used for this purpose.

- Time-invariant weights -> vector (mts object with same weight for each time point created internally)
- Time-variant weights -> mts object (or named list of ts objects)

We can have a look at the weights of an object by accessing the `weights` field.

```
hts_EU28$weights
```

A couple of time-invariant weights are included in the package as example data sets: `weights_pi_ea19, weights_pi_eu28`.

Future Enhancements

- Diagnostics for hierarchical time series
- Benchmark method for direct adjustments
- Indirect adjustment of chain-linked indices
- Pure batch objects
- Summary method
- Dashboard for large numbers of time series

Please address queries to:
Angelika Meraner, Gregor de Cillia,
Markus Fröhlich, Alexander Kowarik

Contact information:
Guglgasse 13, 1110 Wien
phone: +43 (1) 71128−7186
angelika.meraner@statistik.gv.at

persephone

Seasonal Adjustment with an Object-oriented Wrapper for RJDemetra