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

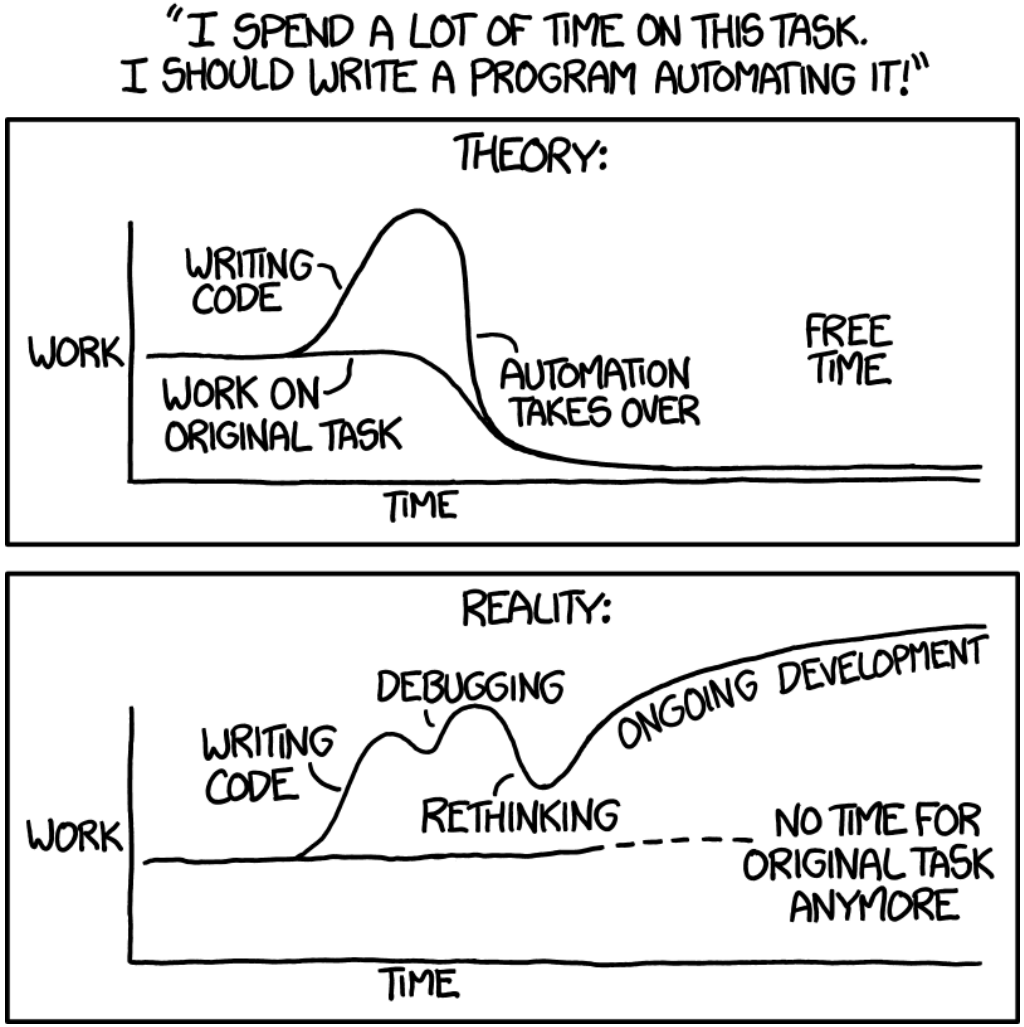
Wien
Februrary 2021

Persephone

Persephone, Production-Ready Seasonal Adjustment in R with RJDemetra

Disclaimer

still under development. . .



Objective

Replace R-package x12 in production @ Statistics Austria for seasonal adjustment.

Requirements:

- Easy processing of multiple time series
- Support of hierarchical time series
- Weighted aggregate series
- R environment

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

Overview

- **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`

Constructing Persephone Objects

1. 'Single' persephone objects are constructed with the functions `perX13()` or `perTramo()` depending on the choice of SA method (X-13-ARIMA-SEATS or TRAMO-SEATS).
2. 'Batch' persephone objects are constructed with the functions `perBatch()` as combination of single objects.
3. Multiple 'single' objects can be combined hierachically with `perHts()` to build a hierarchical persephone object.

Example 1: Persephone Single Object

- Starting from a predefined JDemetra+ model specification has to be provided by the user, e.g. `"RSA3"`
- Updating parameters as needed

```
data(AirPassengers, package = "datasets")
objX13 <- perX13(AirPassengers, "RSA3")
objX13$updateParams(transform.function = "Log")
```

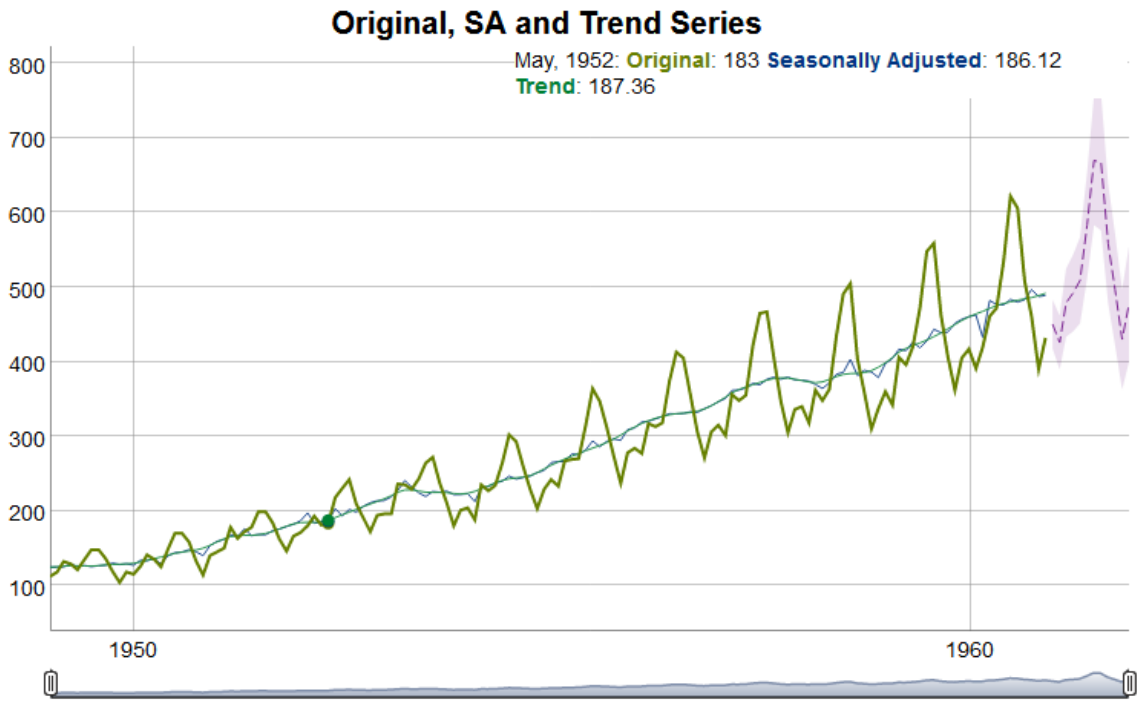
- Different methods can be called on the persephone objects

```
objX13$run()
```

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(objX13)
```



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, perX13, 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 <- perHts(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 <- perHts(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.
- With a simple call to the `run()` method all subseries will be adjusted

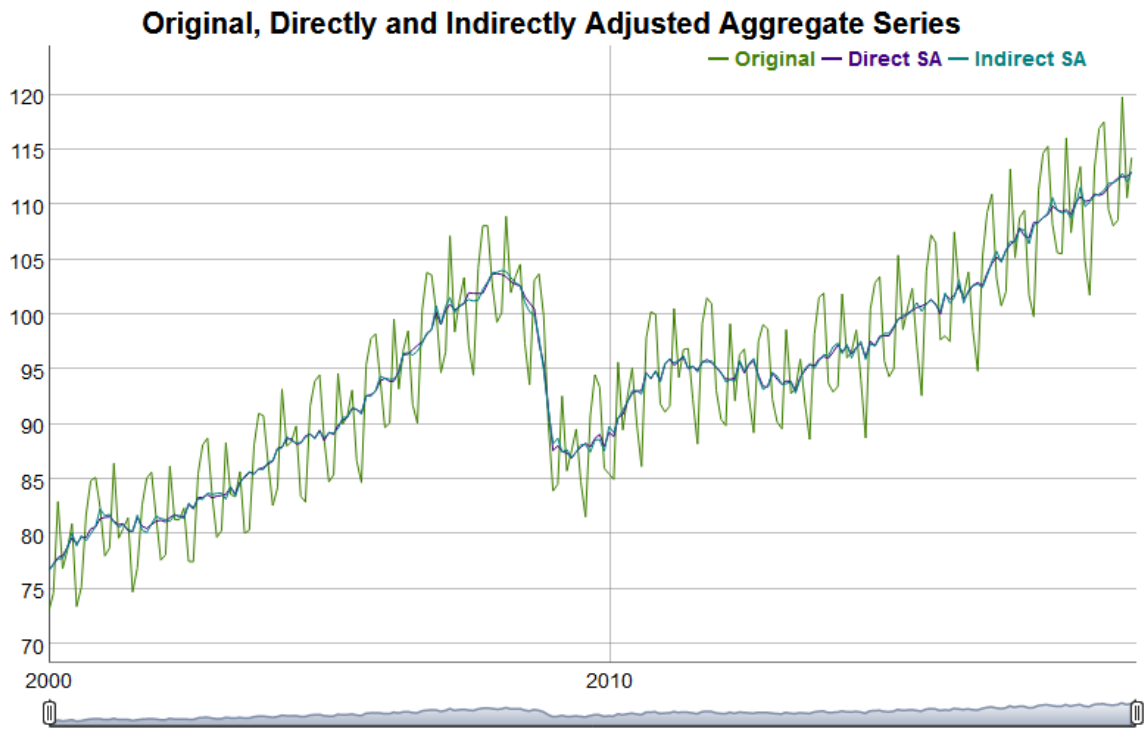
```
hts_EU28
```

```
##   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
```

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 `generateQrTable()`

.

```
head(generateQrTable(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          TRUE    0         2 LS (11-2008)  AO (5-2009)        <NA>
## 4    FALSE          TRUE    0         3  LS (1-2009) LS (11-2008) AO (4-2009)
##   CombinedTest_SI Residual.Seasonality Residual.TD.Effect Q.Stat
## 1         Present                   No                 No     NA
## 2         Present                   No                 No   0.16
## 3         Present                   No                Yes   0.34
## 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%
```

New Features I - Processing

- Persephone Batch Object `perBatch()`
- `fixModel()` or `fixOutlier()`s (except for a timespan at the end of the series)

```
objX13$fixModel(verbose=TRUE)
```

```
## The model( 0 1 1 ) ( 0 1 1 ) is now fixed.
```

```
objX13$fixOutlier(verbose=TRUE)
```

```
## No automatic outliers found.
```

```
## Updating parameter outlier.from to '1960-01-01'
```

New Features II - Generate calendar regressor

```
td7 <- genTd(freq = 4, hd = list("01-01", "01-06",
                "easter+1", "easter+39"),
                 weight = c(rep(1,11), 0.5, rep(1,2), 0.5))
```

More to come

- Benchmark method for direct adjustments
- Indirect adjustment of chain-linked indices
- Summary method
- Dashboard for large numbers of time series

Rückfragen bitte an:
Alexander Kowarik, Angelika Meraner,
Markus Fröhlich, Gregor de Cillia

Kontakt:
Guglgasse 13, 1110 Wien
Tel: +43 (1) 71128 -7513
alexander.kowarik@statistik.gv.at

Persephone

Persephone, Production-Ready Seasonal Adjustment in R with RJDemetra