JDemetra+ and R: Analysing and Visualising Time Series

Atanaska Nikolova

14/09/2018
Outline

- What is JDemetra+
- RJDemetra package
- Seasonal adjustment
- Go interactive!
- Future developments
What is JDemetra+?

- Java based software
- National Bank of Belgium
- Deutsche Bundesbank and Eurostat
- Recommended for seasonal adjustment for official statistics

https://ec.europa.eu/eurostat/cros/content/software-jdemetra_en
What is JDemetra+?

- Open source: https://github.com/jdemetra
- Java libraries used to build a stand-alone software and R functions
- Specialised R package RJDemetra
JDemetra+ Software
Use through R

- Requirements: *rJava* and Java SE 8+
- *RJDemetra* interface for JDemetra+
- *jdemetra-R* for more JD+ routines in R
- Available on Github
  
  https://github.com/nbbrd/rjdemetra
  
  https://github.com/nbbrd/jdemetra-R
jdemetra-R

No description, website, or topics provided.

- 36 commits
- 1 branch
- 0 releases
- 3 contributors

Branch: master

New pull request

Updated files:

Data
- Reading JD+ workspace

Doc
- Documentation and small improvements/normalizations

Java
- Reading JD+ workspace

R files
- Corrections in jd_td + paths

.gitignore
- JD Classes. Test

README.md
- Update README.md

---

**jdemetra-R**

This repository provides R files that simplify the use of JDemetra+ from R. The current solution uses the R-package "rJava" (not provided).

For running the examples ("jd_test.R"), the user must download the complete repository.
Functions need to be sourced as needed:

source("./R files/jd_init.R")
source("./R files/jd_ts.R")
source("./R files/jd_tempdisagg.R")
source("./R files/jd_cholette.R")
RJDemetra

- Installation from Github:

  ```
  install.packages("devtools")
  library(devtools)
  devtools::install_github("nbbrd/rjdemetra")
  ```

- Installation from zip file:

  ```
  install("D:/R/Lib/rjdemetra-master")
  ```
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>add_sa_item</td>
<td>Add a seasonally adjust model to a multi-processing</td>
</tr>
<tr>
<td>compute</td>
<td>Compute the multi-processing from a workspace</td>
</tr>
<tr>
<td>count</td>
<td>Count the number of objects inside a workspace or...</td>
</tr>
<tr>
<td>get_model</td>
<td>Get the seasonally adjusted model from a workspace</td>
</tr>
<tr>
<td>get_name</td>
<td>Get the Java name of a multiprocessing or a sa_item</td>
</tr>
<tr>
<td>get_object</td>
<td>Get objects inside a workspace or multiprocessing</td>
</tr>
<tr>
<td>get_ts</td>
<td>Get the input time series</td>
</tr>
<tr>
<td>load_workspace</td>
<td>Load a JDemetra+ workspace</td>
</tr>
<tr>
<td>myseries</td>
<td>Time series dataset</td>
</tr>
<tr>
<td>new_workspace</td>
<td>Create a workspace or a multi-processing</td>
</tr>
<tr>
<td>plot</td>
<td>Plotting regarima, decomposition or final results of SA</td>
</tr>
<tr>
<td>regarima</td>
<td>RegARIMA model, pre-adjustment in X13 and TRAMO-SEATS</td>
</tr>
<tr>
<td>regarima_spec_def_tramo seats</td>
<td>RegARIMA model specification, pre-adjustment in TRAMO-SEATS</td>
</tr>
<tr>
<td>regarima_spec_def_x13</td>
<td>RegARIMA model specification, pre-adjustment in X13</td>
</tr>
<tr>
<td>save_spec</td>
<td>Saving and loading a model specification, SA and...</td>
</tr>
<tr>
<td>save_workspace</td>
<td>Save a workspace</td>
</tr>
<tr>
<td>specification</td>
<td>Access model specification, SA and pre-adjustment in X13 and...</td>
</tr>
</tbody>
</table>
Creating specifications

- Two methods available:
  - TRAMO/SEATS
    - `tramoseats.spec()`
  - X-13
    - `x13.spec()`
Creating specifications

- Two methods available:
  - TRAMO/SEATS
    \texttt{tramoseats\_spec\_def()}
  - X-13
    \texttt{x13\_spec\_def()}
- Pre-defined default specifications available
Pre-defined specifications

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Log/level detection</th>
<th>Outliers detection</th>
<th>Calendar effects</th>
<th>ARIMA</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSA0</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>Airline(+mean)</td>
</tr>
<tr>
<td>RSA1</td>
<td>automatic</td>
<td>AO/LS/TC</td>
<td>NA</td>
<td>Airline(+mean)</td>
</tr>
<tr>
<td>RSA2c</td>
<td>automatic</td>
<td>AO/LS/TC</td>
<td>2 td vars + Easter</td>
<td>Airline(+mean)</td>
</tr>
<tr>
<td>RSA3</td>
<td>automatic</td>
<td>AO/LS/TC</td>
<td>NA</td>
<td>automatic</td>
</tr>
<tr>
<td>RSA4c</td>
<td>automatic</td>
<td>AO/LS/TC</td>
<td>2 td vars + Easter</td>
<td>automatic</td>
</tr>
<tr>
<td>RSA5c</td>
<td>automatic</td>
<td>AO/LS/TC</td>
<td>7 td vars + Easter</td>
<td>automatic</td>
</tr>
</tbody>
</table>
Pre-defined specifications

- Editing default specifications

```r
spec_user <- x13_spec_def(spec = "RSA5c"),
usrdef.outliersEnabled = TRUE,
usrdef.outliersType = c("LS", "AO"),
usrdef.outliersDate = c("2009-01-01", "2014-05-01"),
outlier.enabled = FALSE)
```
Seasonal Adjustment

- Using the defined specs
  \[SA_{x13} \leftarrow x13(MBS, \text{spec\_user})\]
- Plotting the result (overlay and components)
  \[\text{plot}(SA_{x13})\]
Overlay graph only

```r
plot(SA_x13, type_chart="sa-trend")
```
Components graph only

```r
plot(SA_x13, type_chart="cal-seas-irr")
```
Various Options
Let it Shine

- Using output for Shiny application
- Useful for training purposes
- Interactive dashboards for client business areas
- Facilitate understanding
Let it Shine  (credit to Jennie Davies)

Seasonal adjustment training

Choose an exercise:

Exercise 1

Seasonal adjustment 1
Transformation:

Log

Easter:

Easter[15]

Default trend moving average
Seasonal moving average:

S3x0

Seasonal break:

year.period

Seasonal adjustment 2
Transformation:

None

Easter:

None

Default trend moving average
Trend moving average:

9

Seasonal moving average:

S3X1
Let it Shine  (credit to Jennie Davies)

Seasonal adjustment training

Choose an exercise:
- Exercise 1

Seasonal adjustment 1
- Transformation: Log

Easter:
- Easter[15]

Default trend moving average
- Default trend moving average

Seasonal moving average:
- S3X9

Seasonal break:
- year.period

Seasonal adjustment 2
- Transformation: None

Easter:
- None

Default trend moving average
- Default trend moving average

Trend moving average:
- 9

Seasonal moving average:
- S3X1
Benefits Over Other Packages

- Speed (e.g. compared to seasonal)
Benefits Over Other Packages

- Speed (e.g. compared to *seasonal*)

```r
system.time(tramoseats_def(MBS))
```

```
  user  system elapsed
0.0017 0.2228  0.2228
```

```r
system.time(seas(MBS))
```

```
  user  system elapsed
0.0410 0.6010  0.6010
```

- Direct function call to Java library (no use of intermediary files)
Benefits Over Other Packages

- Fitting ARIMA model benefits from automatic check procedures
- Deterministic effects
- Transformation
Benefits Over Other Packages

- **stats** (2-year MAPE = 2.36):  
  \[
  \text{arima}(\text{MBS}, \text{order}=c(1,0,0), \\
  \text{seasonal}=\text{list}(\text{order}=c(0,1,1),\text{period}=12))
  \]

- **RJDemetra** (2-year MAPE = 1.29)  
  \[
  \text{regarima}(\text{MBS}, \text{regarima\_spec\_def\_tramoseats}( \\
  \text{arima\_p}=1,\text{arima\_d}=0, \\
  \text{arima\_q}=0,\text{arima\_bp} = 0, \\
  \text{arima\_bd} = 1,\text{arima\_bq} = 1))
  \]
Benefits Over Other Packages

• Speed
• Officially recommended by ECB
• Open source - ongoing developments
• Opportunity for development participation and feedback
• Good maintenance
• Support by ESS Seasonal Adjustment Helpdesk
Future Developments

- JDemetra+ 3.0
  https://github.com/nbbrd/jd3-rtests
- Available for testing
- High frequency data: daily and weekly
Future Developments

• Example functions (*jd3-rtests-master*)
• *jd3_fractionalairlinedecomposition*
• *jd3_x11*
• *jd3_stl*
• More information: https://github.com/nbbrd/jd3-rtests/wiki/High-frequency-series
Future Developments

Daily Decomposition

Daily deaths (England & Wales)

Index

1200 1400 1600 1800 2000 2200 2400 2600

0 1000 2000 3000 4000 5000
Future Developments

- Package RJDemetra still under development
- Opportunities for contribution
- Continuously evolving and improving
Special thanks to:

- Seasonal Adjustment Centre of Excellence (SACE)
- RJDemetra main developers:
  Anna Michlek (ECB-NBB)
  Alain Quartier-la-Tente (INSEE)

Thank you!
atanaska.nikolova@ons.gov.uk