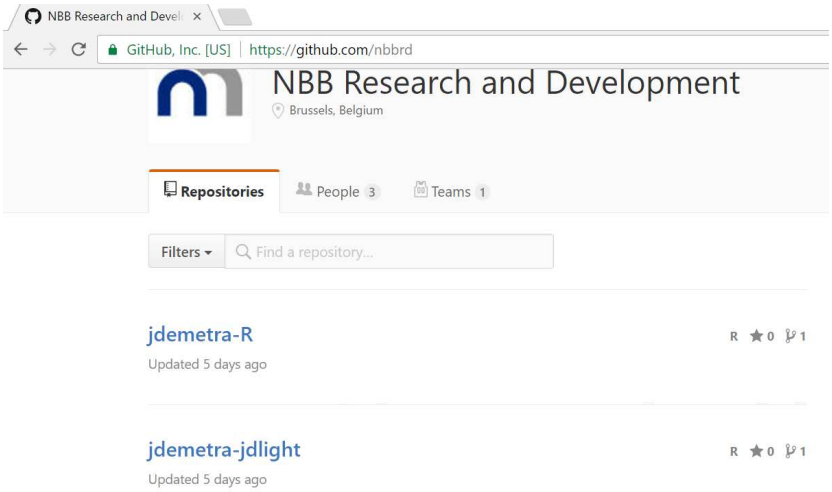
JD+ and R

Paris, 24 October 2016

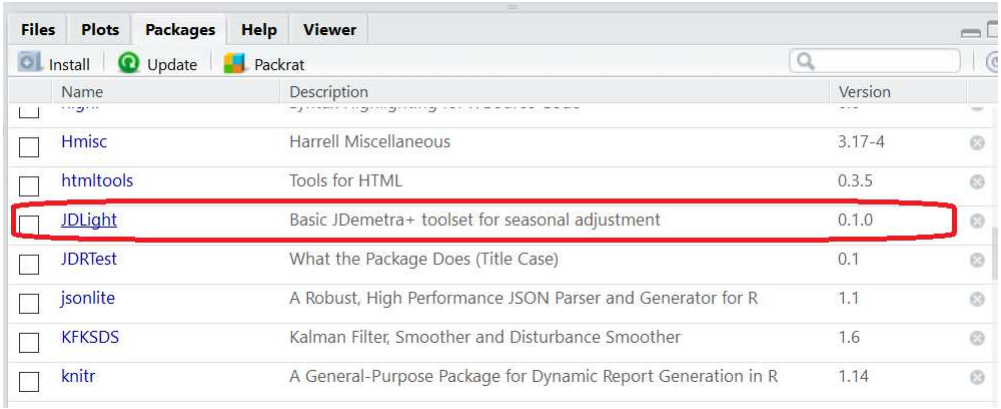
Current and planned projects

1. JDLight package (in test)
2. R files to call JD+
3. [JD full package (2017)]

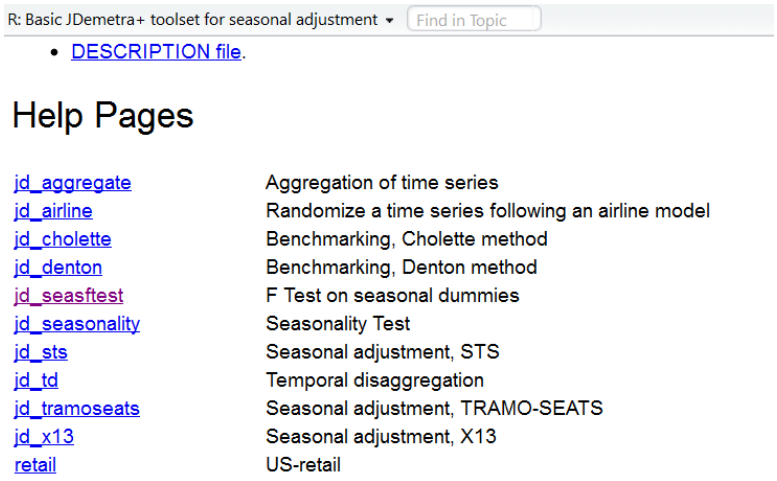
JD+ an R on github



1. JDLight



1.1 JDLight - Contents

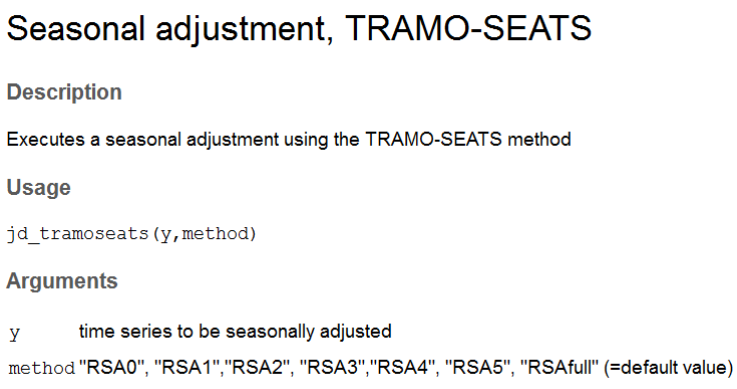

R: Basic JDemetra+ toolset for seasonal adjustment ▾ Find in Topic

- DESCRIPTION file.

Help Pages

jd_aggregate	Aggregation of time series
jd_airline	Randomize a time series following an airline model
jd_cholette	Benchmarking, Cholette method
jd_denton	Benchmarking, Denton method
jd_seasftest	F Test on seasonal dummies
jd_seasonality	Seasonality Test
jd_sts	Seasonal adjustment, STS
jd_td	Temporal disaggregation
jd_tramoseats	Seasonal adjustment, TRAMO-SEATS
jd_x13	Seasonal adjustment, X13
retail	US-retail

1.2 Seasonal adjustment


Seasonal adjustment, TRAMO-SEATS

Description

Executes a seasonal adjustment using the TRAMO-SEATS method

Usage

```
jd_tramoseats(y,method)
```

Arguments

y	time series to be seasonally adjusted
method	"RSA0", "RSA1","RSA2", "RSA3","RSA4", "RSA5", "RSAfull" (=default value)

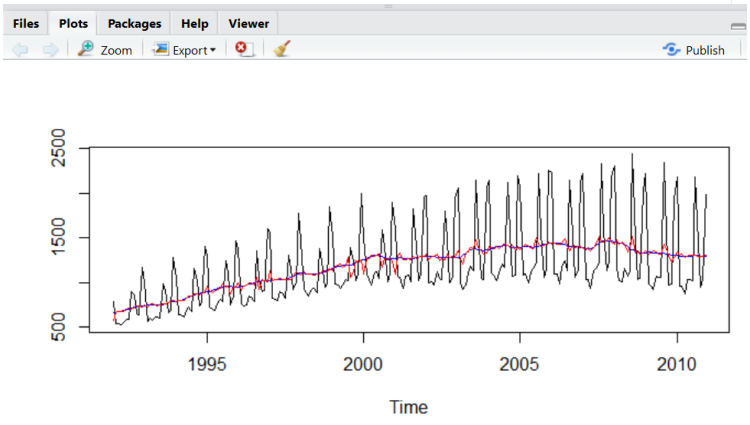
1.3 Execution of Tramo-Seats/X13…

Console ~/JD+/jdemetra-jdlight/

```
> library(JDLight)
> jd_tramoseats(y=retail$BookStores, method = "RSA4")
               y         t         s         i         sa
Jan 1992   790  660.2117 1.2558392 0.9528178  629.0614
Feb 1992   540  666.9574 0.8068509 1.0034654  669.2687
Mar 1992   536  673.9434 0.7926410 1.0033785  676.2204
Apr 1992   524  680.8838 0.7688820 1.0009182  681.5090
May 1992   553  687.8636 0.8062496 0.9971335  685.8918
Jun 1992   589  694.8755 0.8397672 1.0093677  701.3848
Jul 1992   593  701.7326 0.8312430 1.0166116  713.3894
Aug 1992   895  708.3828 1.2668549 0.9973053  706.4740
Sep 1992   863  714.9391 1.1916353 1.0129742  724.2149
Oct 1992   647  721.2821 0.8818630 1.0171805  733.6741
Nov 1992   642  727.3031 0.8762928 1.0073267  732.6319
Dec 1992  1166  733.1512 1.5845386 1.0036960  735.8609
```

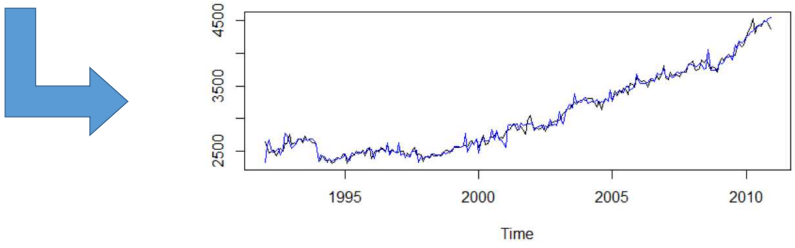
1.4 Displaying SA using R functions

```
> ts.plot(jd_x13(y=retail$BookStores)[,c(1,2,5)], col=c("black", "blue", "red"))
>
```



1.5 Mixing different functions

```
Console ~/JD+/jdemetra-jdlight/
> # Aggregation of a time series
> t_q<-jd_aggregate(retail$AllOtherGenMerchandiseStores, newfreq = 4)
> # Temporal disaggregation (Chow-Lin, could be done with tempdisagg)
> x_m<-retail$BookStores
> t_m<-jd_td(t_q~1+x_m)
> # display the seasonally adjusted interpolated and original series
> ts.plot(jd_x13(retail$AllOtherGenMerchandiseStores)[,5], jd_x13(t_m)[,5], co
l=c("black", "blue"))
```



2 R files

- Basis for future advanced packages
- Main features
  - Jd_test.r
    - SA processing
    - Results retrieval
  - Jd_testsa.r
    - SA processing with user-define specifications
  - Jd_testcalendar.r
    - Definition and use of calendars
  - Jd_testws.r
    - Reading an existing workspace of JD+ and retrieving results

2.1 Reading existing JD+ workspaces

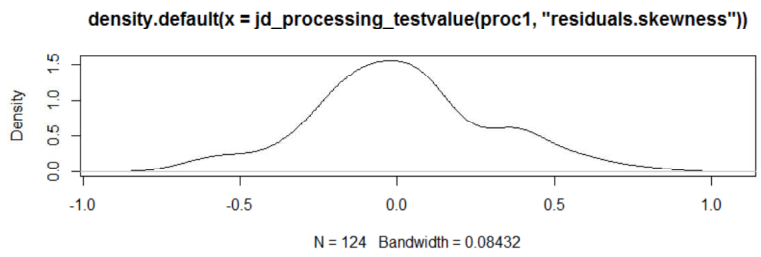
```
# reads a workspace of JD+
ws<-jd_ws(file = "../Data/test.xml")

# gets the multi-processing names
jd_processingNames(ws)

# read a multi-processing
proc1<-jd_processing(ws,"SAProcessing-5")
```

Retrieving results from an existing workspace

```
# displays some results
jd_processing_numeric(proc1, "likelihood.bicc")
# use R facilities
plot(density(jd_processing_testvalue(proc1, "residuals.lb")))
```



Final remarks

- Promising solution
- Can be extended by any R user (the code is often cumbersome but straightforward)

```
jd_ws<-function(file){
  jd_monitor<-.jnew("ec/demetra/jcruncher/Monitor")
  if (FALSE == .jcall(jd_monitor, "Z", "load", file))
    return(NULL)
  jd_monitor
}
```

- Probably more for research than for production