TSACE Webinar, Wednesday December 14th 2022



# Using JDemetra+ in R: from version 2 to version 3
## Presentation 2: Seasonal adjustment in R

Anna Smyk and Tanguy Barthelemy
With the collaboration of Alain Quartier-la-tente

# Contents

# Outline table

# Data formats

here, no workspace structure - assets - shortcomings

# SA process

- testing for seasonality
- pre treatement
- decomposition
- output series
- diagnostics
- customize parameters
- repeat..

comp with GUI main panels ?

# rjd3 suite of packages for SA

in v2 :

in v3: more tools (tests,. . . )

# Contents

# testing for seasonality

Data format TS object (num vector for HF) ### Normality test

# Autocorrelation

# Contents

# Quick Launch with default specifications (1)

specifications - x13 - regarima - x11 (one less spec in default x13)

- Specification: created with `spec_x11_default()`, `spec_x13_default()`, `spec_regarima_default()`

spec_regarima_default(name = c("rg4", "rg0", "rg1", "rg2c", "rg3", "rg5c"))

spec_x13_default(name = c("rsa4", "rsa0", "rsa1", "rsa2c", "rsa3", "rsa5c"))

spec_x11_default()

# Quick Launch with default specifications (2)

- Apply model with `x11()`, `x13()`, `fast.x13()`, `regarima()`, `fast.regarima()`

# Running SA estimation process

in version 2

```
# X13
sa_x13_v2<-RJDemetra::x13(y_raw, spec ="RSA5c")

#Tramo-Seats
sa_ts_v2<-RJDemetra::tramoseats(y_raw, spec ="RSAfull")
```

in version 3

```
#X13
sa_x13_v3 <- rjd3x13::x13(y_raw, spec= "RSA5")

#Tramo seats
sa_ts_v3 <- rjd3tramoseats::tramoseats(y_raw, spec= "RSAfull")
```

affichage output ?

# Running only pre-adjustment

in version 2

```
# Reg-Arima part from X13 only (different default spec names, cf
regA_v2<-RJDemetra::regarima_x13(y_raw, spec ="RG5c")

# Tramo only
tramo_v2<-RJDemetra::regarima_tramoseats(y_raw,
  spec = "TRfull")
```

in version 3

```
#X13
sa_regarima_v3 <- rjd3x13::regarima(y_raw)

#Tramo seats (if no spec indicated what is the defalut ?)
# retrouver en
sa_tramo_v3 <- rjd3tramoseats::tramo(y_raw)

# "fast." versions...(cf output structure)
```

# Running only decomposition

in version 2

```
# X11 (spec option)
X11_v2<-RJDemetra::x13(y_raw, spec ="X11")

#Tramo-Seats ? you
#sa_ts_v2<-RJDemetra::tramoseats(y_raw, spec ="RSAfull")
```

in version 3

```
#X11
x11_v3 <- rjd3x13::x11(y_raw)

#Tramo seats
#sa_ts_v3 <- rjd3tramoseats::seats.decompose(y_raw)
```

affichage output ?

# Output structure v2

show the list of lists do a new version

# Output structure v3 (cf txt file)

show the NEW list of lists

# Differences from v2 to v3

highlight deifferences: - specs - specs direct accessible + 2 concepts (spec in v12 was point spec;, more about this in refresh section)

# Retrieve output series

input and output = TS objects (not when using specific extensions for HF data) - final series : different

```
# Version 2 (affichage main, d tables in user def output)
sa_x13_v2$final$series
```

```
##                    y        sa         t         s         i
## Jan 1990  74.93056  60.11430  58.51831 1.2464681 1.0272733
## Feb 1990  67.27349  58.94740  58.61627 1.1412462 1.0056490
## Mar 1990  71.60221  57.49983  58.74645 1.2452595 0.9787797
## Apr 1990  54.76262  58.27019  58.85384 0.9398051 0.9900831
## May 1990  50.01400  57.65493  58.98080 0.8674714 0.9775203
## Jun 1990  56.43779  59.44801  59.14528 0.9493639 1.0051185
## Jul 1990  58.72544  61.02377  59.34659 0.9623372 1.0282607
## Aug 1990  60.09017  58.91973  59.54885 1.0198650 0.9894351
## Sep 1990  56.82430  59.69652  59.72004 0.9518864 0.9996061
## Oct 1990  57.86107  59.44146  59.80266 0.9734127 0.9939601
## Nov 1990  54.82622  60.01217  59.73370 0.9135851 1.0046617
## Dec 1990  49.32696  60.92179  59.49030 0.8096769 1         5
## Jan 1991  72.89074  59.85221  59.07795 1.2178454 1.0131058
```

# Series from preadjustement

```
# Version 2 (affichage main, d tables in user def output)
sa_x13_v2$regarima$model$effects # data frame
```

```
##                y_lin          tde           ee omhe out_t
## Jan 1990 4.281143  0.0354192655  0.000000000    0     0
## Feb 1990 4.217655 -0.0088889474  0.000000000    0     0
## Mar 1990 4.257676 -0.0039103497  0.017360471    0     0
## Apr 1990 4.035447 -0.0150790633 -0.017360471    0     0
## May 1990 3.896360  0.0159430184  0.000000000    0     0
## Jun 1990 4.034003 -0.0008639551  0.000000000    0     0
## Jul 1990 4.075459 -0.0025860708  0.000000000    0     0
## Aug 1990 4.072815  0.0230308669  0.000000000    0     0
## Sep 1990 4.091918 -0.0519537119  0.000000000    0     0
## Oct 1990 4.022626  0.0354192655  0.000000000    0     0
## Nov 1990 3.987634  0.0165344464  0.000000000    0     0
## Dec 1990 3.933995 -0.0355238594  0.000000000    0     0
## Jan 1991 4.273019  0.0159430184  0.000000000    0     0
## Feb 1991 4.205955 -0.0088889474  0.000000000    0     0
## Mar 1991 4.346519 -0.0323728709 -0.028085787    0     0
```

# Series from decompostion

```
# Version 2 (affichage main, d tables in user def output)

# D tables accessible via user-defined output,

# forecast accessible only via user defined output (cf below)

# Version 3: "x11 names" : preadjustement effets as stored in the
# add doc on names
sa_x13_v3$result$decomposition$d5 # tables from D1 to D13
```

```
##              Jan       Feb       Mar       Apr       May
## 1990 1.1923683 1.1475372 1.2360235 0.9704338 0.8556732
## 1991 1.1923683 1.1475372 1.2360235 0.9704338 0.8556732
## 1992 1.1849743 1.1510465 1.2435127 0.9749012 0.8612511
## 1993 1.1807324 1.1566905 1.2605805 0.9885448 0.8669907
## 1994 1.1922639 1.1523619 1.2757969 1.0068935 0.8695447
## 1995 1.2166191 1.1249847 1.2721669 1.0121162 0.8688322
## 1996 1.2450179 1.0834820 1.2454913 1.0023317 0.8656
## 1997 1.2555405 1.0495401 1.2271165 0.9904221 0.8486
```

# Retrieve Diagnostics

Just fetch the needed objects in the relevant par of the output structure or print the whole "model"

```
# Version 2
print(sa_x13_v2)
```

```
## ^^[[4m^^[[1mRegARIMA^^[[22m^^[[24m
## y = regression model + arima (0, 1, 1, 0, 1, 1)
## Log-transformation: yes
## Coefficients:
##           Estimate Std. Error
## Theta(1)   -0.7247      0.038
## BTheta(1)  -0.5637      0.047
##
##             Estimate Std. Error
## Monday      0.016430      0.009
## Tuesday     0.012493      0.009
## Wednesday   0.006496      0.009
## Thursday   -0.003046      0.009
## Friday      0.019581      0.009
```

# Retrieve user defined-output (1/2)

In v2 or v3 : define a vector of objects your wish to add

Lists of avaible diagnostics or series

```
# Version 2
user_defined_variables("X13-ARIMA")
user_defined_variables("TRAMO-SEATS")
# Version 3: more specific functions
userdefined_variables_tramoseats("tramoseats")
userdefined_variables_tramoseats("tramo") # restriction

userdefined_variables_x13("regarima") #restrction
userdefined_variables_x13()
```

# Retrieve user defined-output (2/2)

Select the objects and customize estimation function

```
# version 3
ud<-userdefined_variables_x13()[15:17] # b series
ud
```

```
## [1] "decomposition.b1"  "decomposition.b10"
## [3] "decomposition.b11"
```

```
sa_x13_v3_UD<-rjd3x13::x13(y_raw,"RSA5c",userdefined=ud)
sa_x13_v3_UD$user_defined # remainder of the names
```

```
## Names of additional variables (3):
## decomposition.b1, decomposition.b10, decomposition.b11
```

```
# retrieve the object
sa_x13_v3_UD$user_defined$decomposition.b1
```

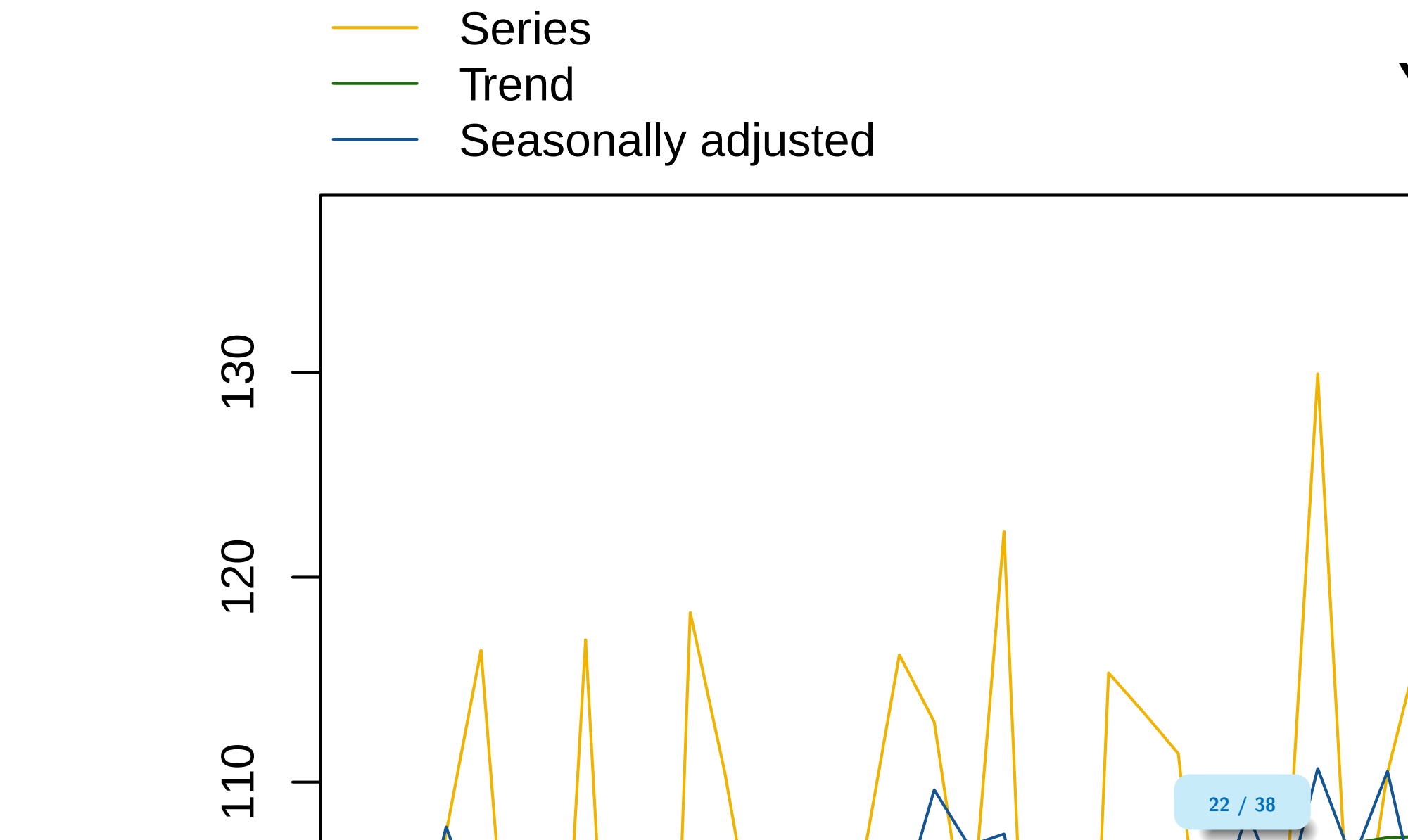
```
##              Jan       Feb       Mar       Apr       May
## 1990   72.32302  67.87415  70.64560  56.56822  49.22
```

# Plots and data visualisation (1/2) (1)

In version 2 3 kinds of plots : final (2 types), regarima and SI ratios

```
# version 2
# for class 'final' : 2 types
# syntaxe min
plot(sa_x13_v2, type_chart = "sa-trend", first_date = c(2015, 1))
```

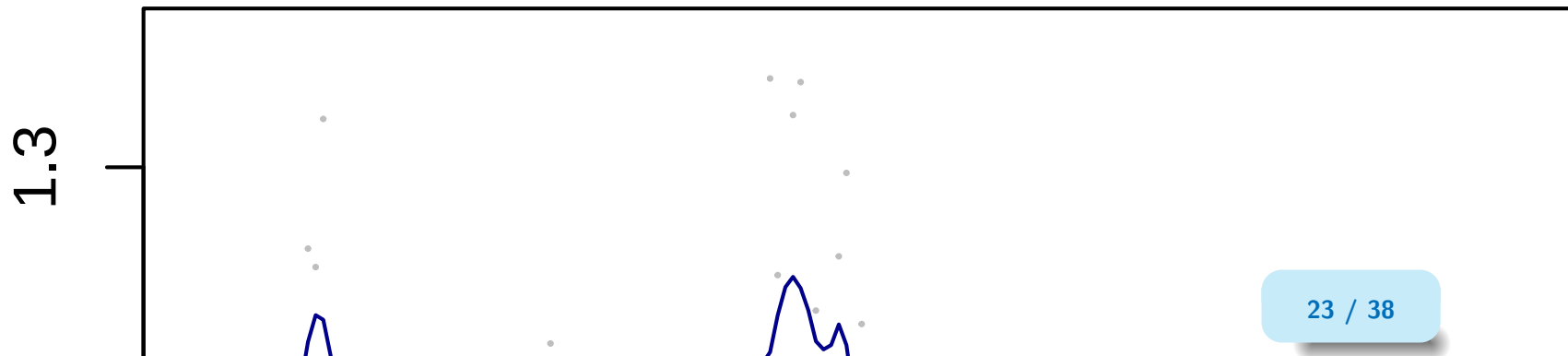
# Plots and data visualisation (1/2) (2)



# Plots and data visualisation (2/2)

In version 3 - final + NEW "autoplot" layout - regarima not available (yet ?) - SI ratios + NEW ggplot layout

```
# version 3
# remotes::install_github("AQLT/ggdemetra3",INSTALL_opts = "--no-
# library(ggdemetra3)
ggdemetra3::siratioplot(sa_x13_v3)
```



# Customizing specifications: general steps

To customize a specification you must - start with a valid specification, usually one of the defaut specs (equivalent to cloning a spec in GUI) - create a new spec - apply the new spec to a series

Some differences between v2 and v3

# Customizing specifications: in version 2

Direct parameter modification as arguments of the spec function

```
# version 2
# changing estimation span, imposing additive model and adding us
# first create a new spec modifying the previous one
spec_1<- x13_spec(sa_x13_v2)
spec_2<- x13_spec(spec_1, estimate.from = "2004-01-01",
                  usrdef.outliersEnabled = TRUE,
                              usrdef.outliersType = c("LS", "AO"),
                              usrdef.outliersDate = c("2008-10-01"
                              transform.function = "None") # addit
# here the reg-arima model will be estimated from  "2004-01-01"
# the decomposition will be run on the whole span

# new sa processing
sa_x13_v2_2<-RJDemetra::x13(serie_brute,spec_2)
sa_x13_v2_2$final$series
```

# Customizing specifications: in version 3

Use direct and specific `set_` functions - for the preprocessing step (functions defined in `rjd3modelling`):

`set_arima()`, `set_automodel()`, `set_basic()`, `set_easter()`, `set_estimate()`, `set_outlier()`, `set_tradingdays()`, `set_transform()`, `add_outlier()` and `remove_outlier()`, `add_ramp()` and `remove_ramp()`, `add_usrdefvar()`

- for the decomposition step in X13 (function defined in `rjd3x13`): `set_x11()`
- for the decomposition step in Tramo-Seats (function defined in `rjd3tramoseats`): `set_seats()`
- for the benchmarking step (function defined in `rjd3modelling`): `set_benchmarking()`

New v3 feature, same options available as in GUI.

# Customizing specifications in version 3: example

```
######### spec custo in v3
# start with default spec
spec_1 = spec_x13_default("RSA3")
# or start with existing spec (no extraction function needed)
spec_1<-sa_x13_v3_UD$estimation_spec

# set a new spec
## add outliers
spec_2 = rjd3modelling::add_outlier(spec_1,
                    type = c("AO"), c("2015-01-01", "2010-01-01"))
## set trading days
spec_2<-rjd3modelling::set_tradingdays(spec_2,
    option = "workingdays" )
# set x11 options
spec_2<-set_x11(spec_2,henderson.filter = 13)
# apply with `fast.x13` (results only)
fast.x13(y,spec_2)
```

# Adding user-defined regressors

In version 2: regressors added directly to the specification

In version 3: new notion of “context” an additionnal concept designed to - -

# Adding user-defined regressors in v2

```
# defining user defined trading days
spec_4 <- x13_spec(spec_1,
tradingdays.option = "UserDefined",
tradingdays.test ="None",
usrdef.varEnabled = TRUE,
# the user defined variable will be assigned to the calendar comp
usrdef.varType="Calendar",
usrdef.var=td_regs ) # regressors have to be a single or multiple
# new sa processing
sa_x13_v2_4<-x13(serie_brute,spec_4)
# user defined intervention variable
spec_5 <- x13_spec(spec_1,
                   usrdef.varEnabled = TRUE,
                   # the user defined variable will be assigned t
                   usrdef.varType="Trend",
                   usrdef.var=x ) # x has to to be a single or mu
# new sa processing
sa_x13_v2_5<-x13(serie_brute,spec_5)
```

# Adding user-defined regressors in version 3

```
# defining user defined trading days
td_reg1<- rjd3modelling::td(12, start=start(y_raw),length = lengt

context<-rjd3modelling::modelling_context(variables=list(a=xvar))

spec <- rjd3x13::spec_regarima_default(name = "rg3") |>
  rjd3modelling::add_usrdefvar(id = "r.a")

reg_a_estimation<-rjd3x13::regarima(window(ts, start=1985, end=20

# if user_def variable to tren d? how to chose component ?
# regressors have to be added one by one
```

# Estimation spec vs result_spec

Possibility of refreshing data is new feature of version 3.

The “sa_model” generated with an estimation: - new handling of spec (no extraction needed) - notion of - estimation spec (domain spec): set of customizable constraints

```
sa_x13_v3$estimation_spec$regarima$arima
```

```
- result spec (or point spec)
```

- in v2 could only retrieve point spec
- in v3 your are able to restimate the result spec inside the domain (estimation spec) freeing constriants on some parameters : just like in GUI

# Steps for refreshing data

```
current_result_spec<-sa_x13_v3$result_spec
current_domain_spec<-sa_x13_v3$estimation_spec

# generate NEW spec for refresh
refreshed_spec<-x13.refresh(current_spec, # point spec to be refr
            current_domain_spec, #domain spec (set of constraints
            policy = "Outliers",
            period = 12, # monthly series
            start = "2017-01-01",
            end = NULL)

# apply the new spec on new data : y_new= y_raw + 3 months
sa_x13_v3_refresh<-x13(y_new,refreshed_spec)
# what will be the domain spec here ?
# domain spec = point spec ?
```

- Outliers identiication : more flaxible the last ouliers or all outleirs in v2, here the span can be customized (Warning: x13.refresh hasn't b
thouroughly tested yet)

# Refresh Policies

- “FreeParameters”,
- “Complete”:
- “Outliers_StochasticComponent”,
- “Outliers”,
- “FixedParameters”,
- “FixedAutoRegressiveParameters”,
- “Fixed”,

# User-defined parameters: summary

- what's new ?
- whats's missing ?

# Contents

# SA of High-Frequency data

High-frequency data can

In github repo full presnetations about {rjd3highfreq}

# Contents

# calendars

here new functionnality of v3, rjd3modelling pacakage

# outliers and intervention variables

(using this variables already presented, now focus on generation)

intervention bug in rj3modelling ?

# Contents

# Conclusion on SA in R

What has v3 brought to the table ?