TSACE, October 26, 2021

rjd3modelling and documentation

Alain Quartier-la-Tente
Insee)
alain.quartier-la-tente@insee.fr

Sommaire

What is available in `rjd3modelling`?

Package available on GitHub:

```
remotes::install_github("palatej/rjd3toolkit")
remotes::install_github("palatej/rjd3modelling")
```

What is available in `rjd3modelling`?

Package available on GitHub:

```
remotes::install_github("palatej/rjd3toolkit")
remotes::install_github("palatej/rjd3modelling")
```

- create trading-days variables with a **user-defined calendar**: easter related days (`calendar.easter`), fixed days (`calendar.fixedday`) and from specific holidays `calendar.holiday` ➔ see `?calendar.new` for a complete example.

What is available in `rjd3modelling`?

Package available on GitHub:

```
remotes::install_github("palatej/rjd3toolkit")
remotes::install_github("palatej/rjd3modelling")
```

- create trading-days variables with a **user-defined calendar**: easter related days (`calendar.easter`), fixed days (`calendar.fixedday`) and from specific holidays `calendar.holiday` ➔ see `?calendar.new` for a complete example.
- create **common regressors**: stock trading days (`stock.td`), leap year regressors (`lp.variable`), easter regressors (`easter.variable`), outliers (`ao.variable`, `ls.variable`, `tc.variable`, `so.variable`), ramp (`ramp.variable`), intervention variables (`intervention.variable`), periodic dummies (`periodic.dummies`) and contrast (`periodic.contrasts`), trigonometric variables (`trigonometric.variables`)

What is available in rjd3modelling?

Package available on GitHub:

```
remotes::install_github("palatej/rjd3toolkit")
remotes::install_github("palatej/rjd3modelling")
```

- create trading-days variables with a **user-defined calendar**: easter related days (`calendar.easter`), fixed days (`calendar.fixedday`) and from specific holidays `calendar.holiday` ➔ see `?calendar.new` for a complete example.
- create **common regressors**: stock trading days (`stock.td`), leap year regressors (`lp.variable`), easter regressors (`easter.variable`), outliers (`ao.variable`, `ls.variable`, `tc.variable`, `so.variable`), ramp (`ramp.variable`), intervention variables (`intervention.variable`), periodic dummies (`periodic.dummies`) and contrast (`periodic.contrasts`), trigonometric variables (`trigonometric.variables`)

? How do you get started with these tools?

Sommaire

Vignette/Word/PDF documentation

Usual documentation, already available for JDemetra+

CALENDARS IN JDEMETRA+

DEFINITION OF CALENDAR EFFECTS

We consider in this document the regression variables that can be defined in JD+ to model calendar effects. We call them hereafter "calendar variables (CV)".

A natural way for modelling calendar effects consists in distributing the days of each period in different groups. The regression variable corresponding to a type of day (a group) is simply defined by the number of days it contains for each period. Usual classifications are:

- Trading days (7 groups): each day of the week defines a group (Mondays...Sundays)
- Working days (2 groups): week days and weekends

But we could also consider:

- 3 groups: week days (Mondays->Fridays), Saturdays, Sundays; that solution will be called TD3 hereafter
- ...

The definition of a group could involve partial days. For instance, we could consider that one half of Saturdays belong to week days and the second half to weekends

HTML/PDF tutorials with `unilur` (1)

Use `unilur` (github.com/koncina/unilur) to create tutorials/practicals or examination papers with `rmarkdown`

```
---
output:
  unilur::tutorial_html: default # without solutions
  unilur::tutorial_html_solution: default # with solutions
  unilur::tutorial_pdf: default
  unilur::tutorial_pdf_solutionn: default
---
```

````
```{r, solution = TRUE}
frenchCalendar <- calendar.new()
```
````

HTML/PDF tutorials with `unilur` (2)

2 - Correction des jours ouvrables

Formation - Désaisonnalisation avec JDemetra+ et RJDemetra

Alain Quartier-la-Tente

> L'objectif de ce TP est d'apprendre à créer des régresseurs jours ouvrables personnalisés, à les utiliser dans RJDemetra/JDemetra+ et à vérifier la qualité de la correction.

Pour créer des régresseurs jours ouvrables personnalisés, deux solutions :

Create interactive tutorials with `learnr` (1)

Tutorials includes in a R package (`remotes::install_github("AQLT/rjd3tutorials")`)

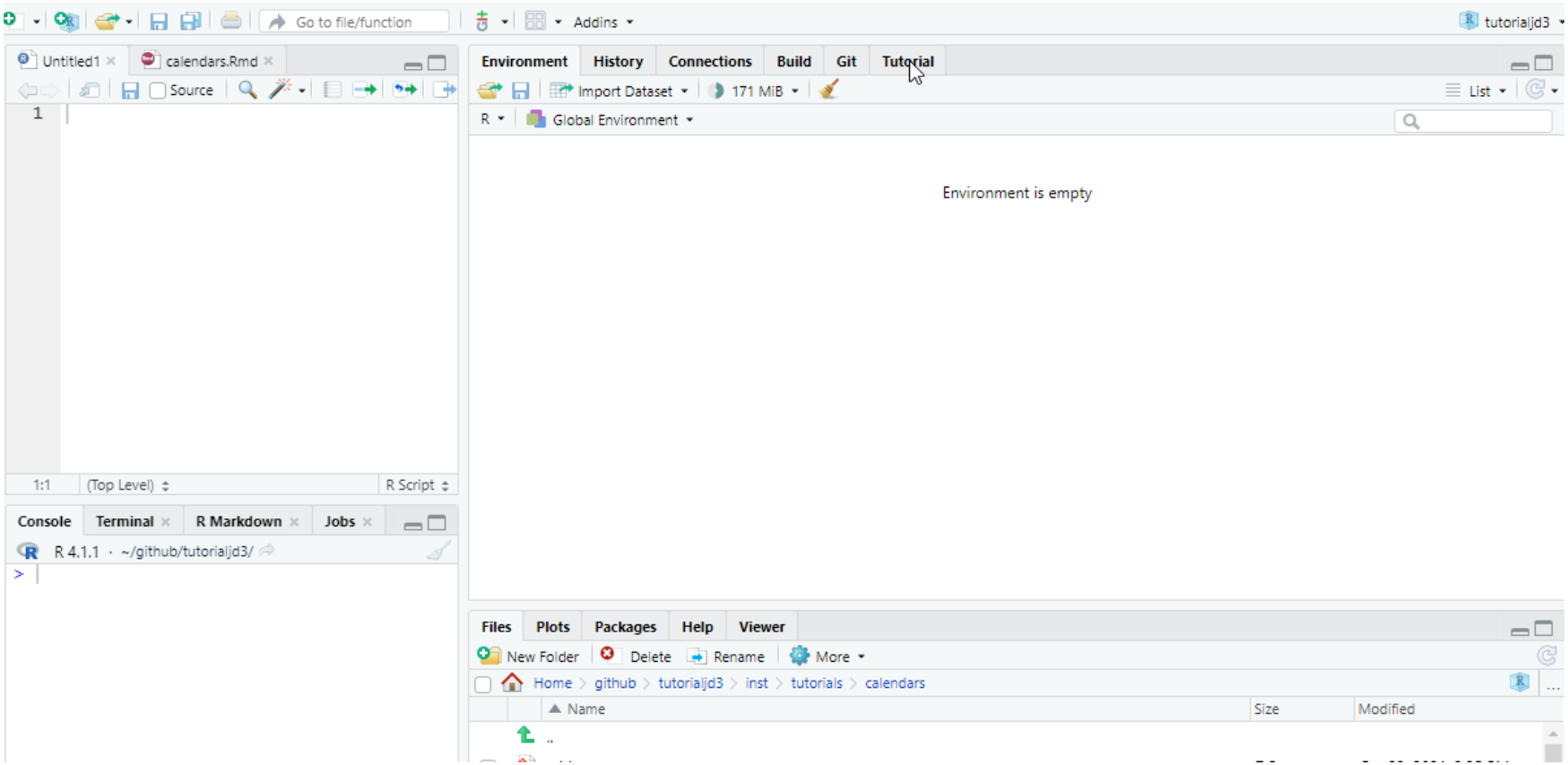
````
```{r regressors, exercise = TRUE}
frenchCalendar <- calendar.new()
```


```{r regressors-hint}
# define Saturday and Sunday as contrast
groups <- c(1, 1, 1, 1, 1, 0, 0)
```


```{r regressors-solution}

```
````

Create interactive tutorials with `learnr` (2)



Thank you for your attention

R packages:

- palatej/rjd3toolkit
- palatej/rjd3modelling

Tutorials:

- AQLT/rjd3tutorials
- example with unilur: https://aqlt-formation-rte.netlify.app/TP/Enonces/R-2-CJO_solution.html

Contact: alain.quartier-la-tente [at] insee.fr