



R news and tutorials contributed by (750) R bloggers

- [Home](#)
- [About](#)
- [RSS](#)
- [add your blog!](#)
- [Learn R](#)
- [R jobs](#) ♦ ♦ ♦
- [Contact us](#)

Welcome!

Follow @rbloggers { 61.9K

Here you will find daily **news and tutorials about R**, contributed by over 750 bloggers.

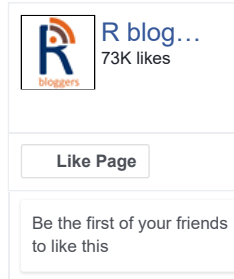
There are many ways to **follow us** -

[By e-mail:](#)

49273 readers

BY FEEDBURNER

[On Facebook:](#)



If you are an R blogger yourself you are invited to [add your own R content feed to this site](#) (Non-English R bloggers should add themselves- [here](#))

[Jobs for R-users](#)

- [R programmer & population data scientist](#)
- [Data Analytics Instructor](#)
- [Financial Systems Analyst](#)
- [Data Scientist](#)
- [Shiny-R-Developer](#)

Popular Searches



- [Boost the speed of R calls from Rcpp](#)
- [Rcpp 0.12.18: Another batch of updates](#)
- [Antipredator behavior with R \(or why wildebeest should stay together\)](#)
- [EARL London interviews – Willem Ligtenberg, CZ](#)
- [Real-time data visualization using R and data extracting from SQL Server](#)
- [DALEX and H2O: Machine Learning Model Interpretability And Feature Explanation](#)
- [REST APIs and Plumber](#)
- [future 1.9.0 – Output from The Future](#)
- [Gifski on CRAN: the fastest GIF encoder in the universe](#)
- [Amazon's Hanging Cable Problem \(Golden Gate Edition\)](#)
- [Evaluation of the 2018 FIFA World Cup Forecast](#)
- [How to add Trend Lines in R Using Plotly](#)
- [Where to get help with your R question?](#)
- [simmer 4.0.0](#)
- [RStudio:addins part 4 – Unit testing coverage investigation and improvement, made easy.](#)

Other sites

- [Jobs for R-users](#)
- [SAS blogs](#)

Boost the speed of R calls from Rcpp

July 24, 2018

By [Iñaki Úcar](#)

Like 21 Share

(This article was first published on [R – Enchufa2](#), and kindly contributed to [R-bloggers](#)).



If you are a user who needs to work with Rcpp-based packages, or you are a maintainer of one of such packages, you may be interested in the recent development of the unwind API, which can be leveraged to boost performance since [the last Rcpp update](#). In a nutshell, until R 3.5.0, every R call from C++ code was executed inside a try-catch, which is really slow, to avoid breaking things apart. From v3.5.0 on, this API provides a new and safe fast evaluation path for such calls.

Some motivation

Here is a small comparison of the old and the new APIs. The following toy example just calls an R function N times from C++. A pure R for loop is also provided as a reference.

```
Rcpp::cppFunction('
  void old_api(Function func, int n) {
    for (int i=0; i
Rcpp::cppFunction(plugins = "unwindProtect", '
  void new_api(Function func, int n) {
    for (int i=0; i

reference <- function(func, N) {
  for (i in 1:N) func()
}

func <- function() 1
N <- 1e6

system.time(old_api(func, N))

##    user    system elapsed
## 17.863    0.006   17.950

system.time(new_api(func, N))

##    user    system elapsed
##  0.289    0.000    0.290

system.time(reference(func, N))

##    user    system elapsed
##  0.216    0.000    0.217
```

Obviously, there is still some penalty compared to not switching between domains, but the performance gain with respect to the old API is outstanding.

A real-world example

This is a quite heavy simulation of an M/M/1 system using simmer:

```
library(simmer)

system.time({
  mm1 <- trajectory() %>%
    seize("server", 1) %>%
    timeout(function() rexp(1, 66)) %>%
    release("server", 1)

  env <- simmer() %>%
    add_resource("server", 1) %>%
    add_generator("customer", mm1, function() rexp(50, 60), mon=F) %>%
    run(10000, progress=progress::progress_bar$new()$update)
})
```

In my system, it takes around **17 seconds** with the old API. The new API makes it in **less than 5 seconds**. As a reference, if we avoid R calls in the timeout activity and precompute all the arrivals instead of defining a dynamic generator, i.e.:

```
system.time({
  input <- data.frame(
    time = rexp(10000*60, 60),
    service = rexp(10000*60, 66)
  )

  mm1 <- trajectory() %>%
    seize("server", 1) %>%
    timeout_from_attribute("service") %>%
    release("server", 1)

  env <- simmer() %>%
```



```
run(10000, progress=progress::progress_bar$new()$update)
})
```

then the simulation takes **around 2.5 seconds**.

How to start using this feature

First of all, you need R $\geq 3.5.0$ and Rcpp $\geq 0.12.18$ installed. Then, if you are a user, the easiest way to enable this globally is to add `CPPFLAGS += -DRCPP_USE_UNWIND_PROTECT` to your `~/R/Makevars`. Packages installed or re-installed, as well as functions compiled with `Rcpp::sourceCpp` and `Rcpp::cppFunction`, will benefit from this performance gains. If you are a package maintainer, you can add `-DRCPP_USE_UNWIND_PROTECT` to your package's `PKG_CPPFLAGS` in `src/Makevars`. Alternatively, there is a plugin available, so this flag can be enabled by adding `[[Rcpp::plugins(unwindProtect)]]` to one of your source files.

Note that this is fairly safe according to reverse dependency checks, but there might be still issues in some packages. But the sooner we start testing this feature and reporting possible issues, the sooner it will be enabled by default in Rcpp.

Article originally published in Enchufa2.es: [Boost the speed of R calls from Rcpp](#).



Share



Tweet

To leave a comment for the author, please follow the link and comment on their blog:

[R – Enchufa2](#).

R-bloggers.com offers [daily e-mail updates](#) about [R](#) news and [tutorials](#) on topics such as: [Data science](#), [Big Data](#), [R jobs](#), visualization ([ggplot2](#), [Boxplots](#), [maps](#), [animation](#)), programming ([RStudio](#), [Sweave](#), [LaTeX](#), [SQL](#), [Eclipse](#), [git](#), [hadoop](#), [Web Scraping](#)) statistics ([regression](#), [PCA](#), [time series](#), [trading](#)) and more...

If you got this far, why not [subscribe for updates](#) from the site?
Choose your flavor: [e-mail](#), [twitter](#), [RSS](#), or [facebook](#)...

Like 21

Share

Share

Comments are closed.

Search R-bloggers

Custom Search

Recent popular posts

- [Real-time data visualization using R and data extracting from SQL Server](#)
- [Evaluation of the 2018 FIFA World Cup Forecast](#)
- [DALEX and H2O: Machine Learning Model Interpretability And Feature Explanation](#)

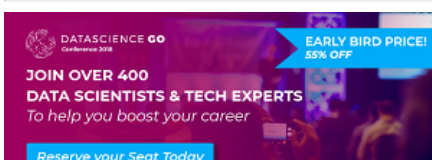
Most visited articles of the week

1. [Machine Learning Results in R: one plot to rule them all!](#)
2. [How to write the first for loop in R](#)
3. [Installing R packages](#)
4. [Using apply, sapply, lapply in R](#)
5. [How to perform a Logistic Regression in R](#)
6. [Tutorials for learning R](#)



7. [R – Sorting a data frame by the contents of a column](#)
8. [Benchmarking Feature Selection Algorithms with Xy\(\)](#)
9. [How to Make a Histogram with ggplot2](#)

Sponsors





Quantide: statistical consulting and training

ODSC West
The Future of Data Science Gathers Here.
San Francisco | Oct 31 - Nov 3

Don't Miss
Europe's Leading
Data Science Conference

Save 40%
Limited time offer

ODSC Europe 2018
London | Sept 19 - Sept 22

R Courses for Professionals
Download R templates for machine learning

The fastest way to learn
data science!

VIEW SNEAK PEAK

DATA SOCIETY

INNOVATION
ANALYTICS
COLLABORATION

eoda | data science environment

CLICK HERE ...

R for Machine Learning:
A Hands-On Introduction
with Max Kuhn

Predictive Analytics
WORKSHOP
Business

June 4, 2018
Las Vegas

LEARN MORE

Vriendenprijs.

HP ENVY X360 15-CN0088NB MET
INTEL® CORE™ I7-8550U PROCESSOR

15.6" 1920 x 1080

1,099,-
999,-

Meer informatie

cool blue

jumping rivers

TRAINING: R, SCALA, STAN





Our ads respect your privacy. Read our [Privacy Policy page](#) to learn more.

[Contact us](#) if you wish to help support R-bloggers, and place **your banner here**.

[Jobs for R users](#)

- [R programmer & population data scientist](#)
- [Data Analytics Instructor](#)
- [Financial Systems Analyst](#)
- [Data Scientist](#)
- [Shiny-R-Developer](#)
- [Senior Data Scientist](#)
- [Behavioral Risk Research Associate @ New York](#)

[Full list of contributing R-bloggers](#)

[R-bloggers](#) was founded by [Tal Galili](#), with gratitude to the [R](#) community.

Is powered by [WordPress](#) using a [bavotasan.com](#) design.

Copyright © 2018 **R-bloggers**. All Rights Reserved. [Terms and Conditions](#) for this website

