

RUGS application for ‘ggtedious’

Evangeline Reynolds

9/12/2023

Dear RUGS committee members,

The following is a proposal for \$900 for a RUGS sponsored workshop: ‘ggtedious’, <https://github.com/EvaMaeRey/ggtedious>. The objective is to work on package building skills, especially the ‘tedious’ tasks of writing tests and user-facing messages (warnings and error). The functions to be packaged are toy ggplot2 extensions, and so the tests and messages will be tailored to that case. See the appendix for details on those functions. Workshop participants will gain experience with these skills, and a *video* recording will be posted so that a wider audience can benefit. The ‘ggtedious’ github repo that we will add to during the workshop is intended to serve as a future reference point for participants and the video audience. It will house a fully-working, ggplot2 extension package that embraces best practices. To our knowledge, there are no workshop resources that focus specifically packaging ggplot2 extensions.

The nature of the project and its goals

New developers may not be motivated to test and write error and warning messages because the rewards aren’t as tangible as writing functionality. “Look my test didn’t fail” feels magnitudes less exciting than “look at this awesome thing my function accomplishes — in one line of code!”.

But the promise of testing is peace of mind; the promise is contentment and confidence not being powerful. This is acknowledged nicely by the testthat introduction:

‘Testing your code can be painful and tedious, but it greatly increases the quality of your code.’ -
{testthat} introduction

So, testing and messaging might not sound like much fun. To get going in this space, we invite people to tie their hands and sign up for the {ggtedious} workshop — committing to diving into this space of best practices! We’ll try to make it as fun and memorable as possible: A goal of the workshop is that, when applying best practices in their future projects, ggtedious participants will be accompanied by warm-and-fuzzy memories from ggtedious workshop!

The workshop will have in-person and remote contingents. \$150 is requested for a professional Zoom account purchase, which will allow the meeting to be recorded and span uninterrupted.

The remaining \$750 honorarium funds will be divided among 2 or 3 consultants. The first consultant will guide the package building itself — tentatively Andrew Heiss. Dr. Heiss has great experience communicating code and methods and bringing energy and warmth to virtual formats. The other consultant(s), to be finalized, will lend their expertise in the business of making the package safe for users beyond the package author (testing, messaging, etc.). In-person hosting venues will be donated.

{ggtedious} is meant to complement other educational materials that focus *only* on building ggplot2 extension functionality (e.g. <https://github.com/EvaMaeRey/easy-geom-recipes>) — and not packaging.

The beginning and end dates for the project

We are working on scheduling the workshop for early 2024, accommodating the schedules of our team and in-person-location sponsors.

A description of who will benefit from the project

Attendees will benefit from the project. To keep the workshop intimate and allow for brief introductions, we'll limit attendees to 20 or fewer online participants. In person locations will accommodate attendees according to the physical limits of the venues and will include the University of Denver (Ritchie School of Engineering and Computer Science).

Furthermore, ggtedious workshop video will be posted about a month after the meeting and the a github repository will be maintained as reference for other new-to-ggplot2 extension folks for reference.

A list of detailed activities along with relevant timelines

Pre Workshop

0. Apply for RUGS funding January, 2024.
1. Secure technical participation, Feb, 2024.
2. Schedule meeting (early March 2024), coordinating with in person hosts and technical experts, and in-person locations.
3. 1 months prior to meeting. Announce at gathering for in-person locations & announce. opportunity for virtual participation and launch application form to participate.
4. 3 weeks prior select virtual participants (if pool is too large), and notify virtual participants.
5. Week prior to workshop, send reminders to workshop participants and experts.

Workshop

1. Introductions (10 minutes) and functionality (30 minutes)
2. Build out package file structure, files (1.25 hours) detail <https://evamaerey.github.io/ggtedious/>

Post Workshop:

1. Post-workshop survey collecting feedback
2. Zoom video posting on YouTube ~ 1 month after meeting

Your measure of success for the project

Provide positive and informative first-experiences in package writing, testing, and error/warning message writing is the primary goal of the workshop, evaluated with a brief, voluntary post-workshop survey.

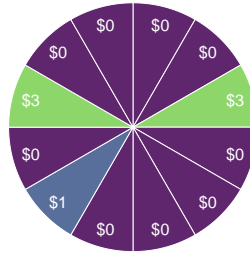
A detailed financial plan and list of expenses

\$1050 total

- \$150 1X professional Zoom membership,
- \$750 divided among 2 or 3 consultants
- Donated: In person meeting space at the University of Denver

Appendix: toy project proposed functions for the ggtedious workshop

To motivate our project we can think about data viz for a discrete probability problem. For a prize spin wheel, we'd like to calculate the 'expected value' for a single spin of the wheel. As a pre-step, we'd like to visualize the distribution of outcomes. Some work on this has already been done in the exploratory package `[{ma206distributions}]` found here: <https://evamaerey.github.io/ma206distributions/>.

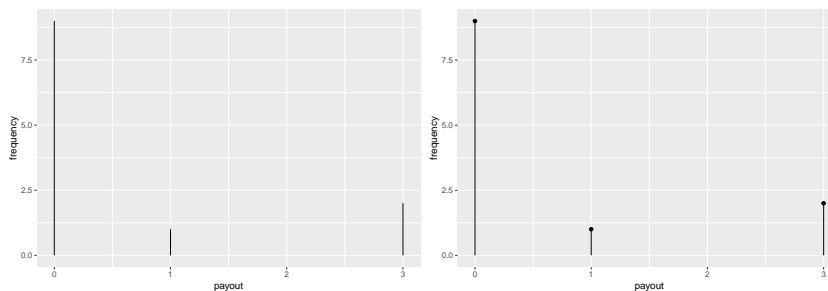


Status quo code and output

```
prize_wheel <- data.frame(sector_type = c("No Prize", "Win $1", "Win $3"),
                          frequency = c(9, 1, 2),
                          payout = c(0, 1, 3))

library(ggplot2)
ggplot(prize_wheel,
       aes(x = payout, y = frequency, xend = payout, yend = 0)) +
  geom_segment()

last_plot() +
  geom_point()
```



Proposed code that package should enable

```
library(ggplot2)
ggplot(prize_wheel, aes(x = payout, y = frequency)) +
  geom_post()

ggplot(prize_wheel, aes(x = payout, y = frequency)) +
  geom_lollipop()
```