Review of `gtsummary` for R Journal

Overview

Thank you very much for giving me the opportunity to review the `gtsummary` submission to the R journal.

I have read the article, played with the package, visited the website, and skimmed the code.

In my view, this looks like an excellent package. There are a lot of competitors in this "space", but `gtsummary` does offer some interesting features. For instance, I especially like the support for `survey` objects and the theming functionality, two features I have not seen elsewhere. The inline reporting function is super clever. The package is very well documented, and the vignettes are detailed and useful. Finally, I really love the complex Table 1 example in the paper. Powerful stuff!

Below, I raise one semi-serious issue related to testing, and a list of very minor points that the authors could consider in the future (but not necessarily for this publication).

Article

The R Journal submission is very clear and well-written. The examples are useful and adequately described. I was able to reproduce them easily on my computer. This submission would not require much (if any) copy editing before publication.

The article reads quite well as-is, but if the authors (or editors) are looking to streamline it, here are two possibilities.

1. The article is very heavy in tables, which is understandable given the nature of the package. But maybe there is a way to rationalize? For instance, the `tbl_svysummary_1` table is essentially identical to the previous one. Do we need to see it in such a short exposition, or is it sufficient to discuss `survey` support?
2. I'm not sure I understand the motivation for the table customization functions (e.g., `bold_levels`). Isn't the whole point of building a package on top of `gt` that we can leverage its functions directly?

Package: Testing

The `gtsummary` developers use `testthat` as their testing suite, and `codecov` to measure and report their test coverage. On Github, they report a 93% test coverage which, at first glance, is very impressive. Unfortunately, I do not think that those tests follow best practices, and I am not convinced that the 93% tells us much.

After looking at a few tests I did a quick `grep` and found that about 75% of expectations in the test folder were either `expect_message`, `expect_error`, or `expect_warning` (please correct me if this is wrong!). Most of the time, the tests only seem to make sure that no error is produced.

Of course, a package can misbehave in *many* ways without producing an actual error. This is especially true in cases like this for packages like this one, which relies on a lot of external dependencies (15 direct dependencies, but over 85 sub-dependencies). Given the nature of the test suite, users have little guarantee that their output won't change in non-trivial ways if one of those 85 packages changes.

Package: Misc

Here are a few minor questions related to documentation improvement or future versions of the package. I do not think these have to be solved or answered before publication.

- tbl_summary
  - the continuous/categorical cutoff seems to be 10. can this be changed?
  - Is there a way to set the number of digits for all variables at the same time? This can be onerous when there are many variables.
  - Can we give a manual sort order?
  - Is it possible to use custom/arbitrary functions in the `statistic` argument, or are we limited to those explicitly defined by the package?
- tbl_regression
  - How do I manually reorder terms?