Turning R objects into Pandoc's markdown

Content of this proposal is not editable at this moment, because student application period is over. If you still need to make some changes, please send a comment to the organization. They can enable modifications for this proposal.

Organization: R Project for Statistical Computing

Abstract: Pander is an R package for rendering R objects on markdown. It provides extensive support for different R classes combined with a lot of rendering options. During this GSoC session I want to improve on 4 things about Pander package: 1. Improve current test suite for pander package 2. Refactor pandoc.table function 3. Add rendering support for not yet supported R classes 4. Create a use-case specific vignettes and add more examples

Additional info: https://github.com/rstats-gsoc/gsoc2015/wiki/pander

Public URL: http://www.google-melange.com/gsoc/proposal/public/google/gsoc2015/roman...

Bio of Student

I'm a Fulbright Graduate student from Ukraine, now pursuing master's degree in Purdue university, USA. My main interests lay in the area of programming language implementation and statistics. Since last year, I have been working on fastR project (https://bitbucket.org/allr/fastr/) with Purdue university and Oracle, which is focused on implementing R on top of Graal compiler and Truffle framework. Besides implementation of some parts of fastR, I have also been working on a testing framework for R which focuses on automatic test case generation and filtering test based on coverage (https://github.com/allr/testr). I made a presentation on work done on R in Purdue at useR2014 (http://user2014.stat.ucla.edu/abstracts/talks/129_Tsegelskyi.pdf).

Last year, I have successfully participated in GSoC working on Pander (https://www.google-melange.com/gsoc/project/details/google/gsoc2014/romantsegelskyi/5724160613416960). We had a really productive collaboration, during which I was able to fulfill most of my goal for previous summer.

I think my background qualifies for this project because I have good background working with R (both internals and scripts), I have good understanding of it's semantics, limitations and best usages.

Student affiliation

Institution: Purdue University

Program: Master of Science (MS)

Stage of completion: graduation in May 2015

Schedule Conflicts:

I will be traveling to Ukraine last 2 weeks of May. While I will be able to work during that time, my availability will be limited. Also I might be attending ECOOP conference (http://2015.ecoop.org/) in Prague in July.

MENTORS

Mentor names: Gergely Daróczi, László Szakács

Have you been in touch with the mentors? When and how?

We collaborated with Gergely last year, and we maintained contact since then. Also we were fortunate to meet in person last year during useR2014 conference.

CODING PLAN & METHODS

During this GSoC session I have want to achieve such objectives:

1. Objective: improve current test suite for pander package

Recently pander started using https://coveralls.io/ for continuos integration and it revealed that only 55% of current codebase is covered by existing tests. Considering that in later stages of Google Summer of Code, I am planning to change and refactor certain integral parts of the existing functionality, I feel that it will be more beneficial to improve the test suite first, to have a more stable ground for future changes and improvements.

So during that I want to work on such major aspects:

1. Increase package coverage to at least 80%, preferably 90%+. Main focus will be functions in **pandoc.R** and S3 methods since I am planning to improve and refactor that functionality later.

2. Work on robust way of testing correctness of S3 methods of pander that are concerned with rendering table. Just plain testing on output equivalence is not entirely correct in this case.

3. Use my previous work with test case generation to produce more unit tests from larger scripts.

2. Objective: refactor **pandoc.table** function

Large chunk of functionality of **Pander** package relies on pandoc.table function. This function seems to be somewhat overly complicated with a not very straightforward flow. I want to try to refactor pandoc.table as much as possible while reducing the number of checks by having a more standardized form of input. This might also require to adjust S3 methods that call pandoc.table accordingly, but I feel that general clarity of package and future extensibility will improve greately. If time permits I also want to look into parallelization of a rendering of large table.

3. Objective: create new **pander** and **broom** methods for not yet supported R classes

Starting list of classes to implement markdown rendering for:

• ivreg (AER package)

• tobit (AER package)

• lmer (lme4 package)

• nmer (lme4 package)

• glmer (lme4 package)

• survreg (MASS package)

• polr(MASS package)

Currently those classes are not supported by neither **Pander** or **Broom** packages. For most of them default methods does not produce any reasonable output or fails with an error, so it is a reasonable addition. As I will be going through internal of listed classes I feel that is reasonable to add the support for those classes to **Broom**(https://github.com/dgrtwo/broom) package also.

From my experience, one obstacle might be that some classes really require deeper understanding of it's internals and extra treatment, so they might take more time that expected.

This is only a list of classes that come from the top of my head, I also plan to add more if time permits.

4. Objective: create a use-case specific vignettes and add more examples

Currently **Pander** has a mature function-level documentation. All package functions are rather well documented. However, function documentation is great if you know the name of the function you need, but it's useless otherwise. I feel that **Pander** lacks the use-case specific documentation. And for this I want to add vignettes for most common use-cases of **Pander.** A vignette is like a book chapter or an academic paper: it can describe the problem that your package is designed to solve, and then show the reader how to solve it.

In my opinion it would be beneficial to add vignettes to make **Pander** more user-friendly:

1. Introduction to Pander

2. Generating markdown documents using Pander.brew.

3. Integration between Pander and knitr.

TIMELINE

May 25 - June 7 - Objective 1. Extending existing test suite and provide more documentation.

June 8 - July 1 - Objective 2. Refactor pandoc.table function.

July 1 - July 6. Objective 2. Documentation and testing.

July 7 - August 2. Objective 3. Create new pander methods for not yet supported R classes.

August 3 - August 17. Objective 4. Create a use-case specific vignettes and add more example

August 17 - August 21. Additional documentation. Preparation for evaluation.

What is your contingency plan for things not going to schedule?

Contingency plan - I tried dividing each of my 4 high level goals into smaller subgoals. For example for objective 3, I have a draft of objects that I want to add support for in Pander, so depending on complexity of each particular class, list can be extended with more objects or shrunk down to most important one. Same for objective 4 - depending on complexity of improving documentation, plans can be easily adjusted. This proposal is a general framework to give a high level idea what I want to achieve during GSoC, while exact plans may vary depending on different factors.

Test Task

Test task was to add rendering support for tabular class from tables package.

tabular class has specific structure and can support complex tables, so basically task was to extract this information in a table, add appropriate row and column names and then supplying this to a pandoc.table function which produces markdown rendering for tables. Also my implementation relies on idea of multiline cells, so rendering produced by pander can be nicely converted using pandoc command line tool to other format.

Exact implementation can be found in my pull request - https://github.com/Rapporter/pander/pull/161