Notre système de revue en revue!

Présenter & analyser rOpenSci onboarding

Maëlle Salmon, éditrice pour rOpensci onboarding

ma_salmon maelle masalmon.eu

2018/06/23

Ma place dans la communauté R

- Travail pour rOpenSci et Locke Data
- *R-Ladies Global social master*
- Membre de l'équipe de R Weekly
- Éditrice bénévole pour rOpenSci

Qu'est-ce rOpenSci ?

Qu'est-ce rOpenSci ?

- Communautés de chercheur·se·s et de développeur·se·s de logiciels
- Paquets R pour la science ouverte et reproductible
- Contributions par le personnel et la communauté

Les paquets de rOpenSci

https://ropensci.org/packages/

Les paquets de rOpenSci

https://ropensci.org/packages/

FILTERS

- ○ All
- ○ Data Publication
- ◉ Visualization
- ○ Geospatial
- ○ Images Processing
- ○ Computing Infrastructure
- ○ Taxonomy
- ○ Altmetrics
- ○ Tools
- ○ Databases
- ○ Web
- ○ Literature
- ○ Security
- ☐ CRAN / BIOC available

Type to search...

PACKAGE	MAINTAINER	DESCRIPTION	DETAILS
iheatmapr	Alicia Schep	Interactive, Complex Heatmaps	CRAN
plotdap	Carson Sievert	Easily visualize data from 'ERDDAP' Servers via the rerddap package	CRAN
plotly	Carson Sievert	Programmatic interface to the Plot.ly API -construct plots	CRAN

Les paquets de rOpenSci

https://ropensci.org/packages/

FILTERS

- ○ All
- ○ Data Publication
- ○ Visualization
- ○ Geospatial
- ◉ Images Processing
- ○ Computing Infrastructure
- ○ Taxonomy
- ○ Altmetrics
- ○ Tools
- ○ Databases
- ○ Web
- ○ Literature
- ○ Security
- ☐ CRAN / BIOC available

Type to search...

PACKAGE	MAINTAINER	DESCRIPTION	DETAILS
ijtiff	Rory Nolan	TIFF I/O for ImageJ Users	CRAN
magick	Jeroen Ooms	Advanced Image-Processing in R, using the imagemagick library	CRAN
tesseract	Jeroen Ooms	Bindings to Tesseract OCR engine	CRAN

Les paquets de rOpenSci

https://ropensci.org/packages/

FILTERS

- ○ All
- ○ Data Publication
- ○ Visualization
- ○ Geospatial
- ○ Images Processing
- ○ Computing Infrastructure
- ◉ Taxonomy
- ○ Altmetrics
- ○ Tools
- ○ Databases
- ○ Web
- ○ Literature
- ○ Security
- ☐ CRAN / BIOC available

Type to search...

PACKAGE	MAINTAINER	DESCRIPTION	DETAILS
binomen	Scott Chamberlain	Taxonomic name classes and parsing methods	CRAN
ritis	Scott Chamberlain	Wrapper to the Integrated Taxonomic Information Service (ITIS) API	CRAN
taxa	Scott	Taxonomic Classes	CRAN

rOpenSci onboarding !

Comment assurer la qualité de tous les paquets ?

rOpenSci onboarding !

Comment assurer la qualité de tous les paquets ?

Revues de logiciels ouvertes. *Onboarding* = accueil et intégration.

rOpenSci onboarding !

Comment assurer la qualité de tous les paquets ?

Revues de logiciels ouvertes. *Onboarding* = accueil et intégration.

- Stimule l'adoption des meilleurs pratiques et standards.

rOpenSci onboarding !

Comment assurer la qualité de tous les paquets ?

Revues de logiciels ouvertes. *Onboarding* = accueil et intégration.

- Stimule l'adoption des meilleurs pratiques et standards.
- Construit une communauté de pratique.

rOpenSci onboarding !

Comment assurer la qualité de tous les paquets ?

Revues de logiciels ouvertes. *Onboarding* = accueil et intégration.

- Stimule l'adoption des meilleurs pratiques et standards.
- Construit une communauté de pratique.
- Partenariats avec le *Journal of Open Source Software* et *Methods in Ecology and Evolution*

Passer quoi en revue ?

Passer quoi en revue ?

Jeff Leek https://github.com/jtleek/rpackages

Documentation > Usability > Speed > Statistical superiority

Critères de revue

- Licence compatible avec l'*open-source initiative (OSI)*
- Documentation complète
- Haut taux de couverture de test
- Code lisible
- Utilisabilité

Critères de revue

- Licence compatible avec l'*open-source initiative (OSI)*
- Documentation complète
- Haut taux de couverture de test
- Code lisible
- Utilisabilité

https://ropensci.github.io/dev_guide/building.html

Critères de revue

- Licence compatible avec l'*open-source initiative (OSI)*
- Documentation complète
- Haut taux de couverture de test
- Code lisible
- Utilisabilité

https://ropensci.github.io/dev_guide/building.html

Tout un livre ! https://ropensci.github.io/dev_guide

Comment faire la revue ?

- Ouverte & non-accusatoire

Comment faire la revue ?

- Ouverte & non-accusatoire
- Pas de rejet

Comment faire la revue ?

- Ouverte & non-accusatoire
- Pas de rejet
- Un processus constructif pour tous les acteurs

Comment faire la revue ?

- Ouverte & non-accusatoire
- Pas de rejet
- Un processus constructif pour tous les acteurs
- En pratique, utilisation de l'infrastructure de GitHub

rOpenSci onboarding

ropensci / onboarding

Watch 25 | Star 123 | Fork 63

Code | Issues 30 | Pull requests 0 | Projects 0 | Wiki | Insights

rOpenSci Onboarding: peer-review of community-contributed packages

r | rstats | r-packages | peer-reviews | peer-review | software-quality

244 commits | 9 branches | 0 releases | Fetching contributors

Branch: master | New pull request | Create new file | Upload files | Find file

Clone or download

Maëlle Salmon Update names

Latest commit f3c381d Jun 25, 2018

CNAME	added a CNAME	Aug 21, 2017
README.Rmd	Update names	Jun 25, 2018
README.md	Update names	Jun 25, 2018
editors_guide.md	Update editors_guide.md	Jun 14, 2018

Onglet de suivi des "problèmes"

ropensci / onboarding

Watch 25 | Star 123 | Fork 63

Code | Issues 30 | Pull requests 0 | Projects 0 | Wiki | Insights

is:issue is:open | Labels | Milestones | New issue

30 Open | 153 Closed | Author | Labels | Projects | Milestones | Assignee | Sort

- Submission: smapr [1/editor-checks] [package] [topic:data-access] [topic:geospatial] — 1
  #231 opened Jun 20, 2018 by mbjoseph — 0 of 8
- DataPackageR submission [1/editor-checks] [package] — 3
  #230 opened Jun 18, 2018 by gfinak — 11 of 19
- Pre-submission inquiry: repvar [package] — 2
  #229 opened Jun 18, 2018 by zkamvar
- Submission: tiler [2/seeking-reviewer(s)] [package] — 5
  #226 opened Jun 12, 2018 by leonawicz — 11 of 19
- submission: arkdb [3/reviewer(s)-assigned] [package] [topic:databases] — 4
  #224 opened Jun 8, 2018 by cboettig — 11 of 19

Soumettre un paquet

ropensci / **bowerbird**

Watch 4 | Star 14 | Fork 2

Code | Issues 5 | Pull requests 0 | Projects 0 | Wiki | Insights

Keep a collection of sparkly data resources https://ropensci.github.io/bowerbird/

r | rstats | r-package

566 commits | **8** branches | **0** releases | Fetching contributors

Branch: **master** | **New pull request** | **Create new file** | **Upload files** | **Find file**

Clone or download

Fetching latest commit...

R	add bb_modify_source function	Feb 24, 2018
docs		
inst/extdata		
man	add bb_modify_source function	Feb 24, 2018
tests		

Soumettre un paquet

ropensci / onboarding

Watch 25 | Star 123 | Fork 63

Code | Issues 30 | Pull requests 0 | Projects 0 | Wiki | Insights

is:issue is:open | Labels | Milestones | New issue

30 Open | 153 Closed | Author | Labels | Projects | Milestones | Assignee | Sort

- Submission: smapr [1/editor-checks] [package] [topic:data-access] [topic:geospatial] — 1
  #231 opened Jun 20, 2018 by mbjoseph — 0 of 8
- DataPackageR submission [1/editor-checks] [package] — 3
  #230 opened Jun 18, 2018 by gfinak — 11 of 19
- Pre-submission inquiry: repvar [package] — 2
  #229 opened Jun 18, 2018 by zkamvar
- Submission: tiler [2/seeking-reviewer(s)] [package] — 5
  #226 opened Jun 12, 2018 by leonawicz — 11 of 19
- submission: arkdb [3/reviewer(s)-assigned] [package] [topic:databases] — 4
  #224 opened Jun 8, 2018 by cboettig — 11 of 19

Soumettre un paquet

ropensci / onboarding

Watch 25 | Star 123 | Fork 63

Code | Issues 30 | Pull requests 0 | Projects 0 | Wiki | Insights

Title

Write | Preview

Summary

- What does this package do? (explain in 50 words or less):

- Paste the full DESCRIPTION file inside a code block below:

```

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Styling with Markdown is supported

Submit new issue
```

Pas de rejets mais...

Paquets hors du domaine visé non accueillis.

https://ropensci.github.io/dev_guide/policies.html

Pas de rejets mais...

Paquets hors du domaine visé non accueillis.

https://ropensci.github.io/dev_guide/policies.html

- nos catégories: accès à des données, extraction de données, accès à des formats de base de données, manipulation de données, dépôts de données,reproducibilité, données géospatialies, analyse de texte.

Pas de rejets mais...

Paquets hors du domaine visé non accueillis.

https://ropensci.github.io/dev_guide/policies.html

- nos catégories: accès à des données, extraction de données, accès à des formats de base de données, manipulation de données, dépôts de données,reproducibilité, données géospatialies, analyse de texte.
- application *scientifique*

Pas de rejets mais...

Paquets hors du domaine visé non accueillis.

https://ropensci.github.io/dev_guide/policies.html

- nos catégories: accès à des données, extraction de données, accès à des formats de base de données, manipulation de données, dépôts de données,reproducibilité, données géospatialies, analyse de texte.
- application *scientifique*
- meilleur que les paquets similaires

Pas de rejets mais...

Paquets hors du domaine visé non accueillis.

https://ropensci.github.io/dev_guide/policies.html

- nos catégories: accès à des données, extraction de données, accès à des formats de base de données, manipulation de données, dépôts de données,reproducibilité, données géospatialies, analyse de texte.
- application *scientifique*
- meilleur que les paquets similaires

En cas de doute, demande de pré-soumission !

Demande de pré-soumission ?

ropensci / onboarding

Watch 25 | Star 123 | Fork 63

Code | Issues 30 | Pull requests 0 | Projects 0 | Wiki | Insights

Doctr: Pre-Submission Inquiry #110

New issue

Closed ctlente opened this issue Apr 13, 2017 · 2 comments

ctlente commented Apr 13, 2017

This is a pre-submission of the `doctr` package. I'd just like to know if I should consider going for the formal submission or whether the package doesn't fit rOpenSci's guidelines.

Summary

- What does this package do? (explain in 50 words or less):
  `doctr` fits in rOpenSci's "reproducibility" category and its main goal is to automate EDA workflows while at the same time making them reproducible, predictable and consistent. Like a real doctor, it has functions for examining, diagnosing and assessing the progress of its "patients".
- URL for the package (the development repository, not a stylized html page):
  https://github.com/ctlente/doctr

Assignees

No one assigned

Labels

0/presubmission

out-of-scope

Projects

None yet

Milestone

No milestone

Demande de pré-soumission ?

ropensci / onboarding

Watch 25 | Star 123 | Fork 63

Code | Issues 30 | Pull requests 0 | Projects 0 | Wiki | Insights

Presub enquiry: bowerbird #129

New issue

Closed **raymondben** opened this issue Jul 2, 2017 · 7 comments

raymondben commented Jul 2, 2017 • edited — Member

See https://github.com/AustralianAntarcticDivision/bowerbird

This is a data retrieval package, but I'm enquiring as to whether this fits within your desired scope for such packages. Specifically, it facilitates downloading and maintaining a local data repository from one or more third-party data providers. It comes with (at latest count) 50-odd predefined data sources, generally themed around marine and Antarctic environmental data. The basic usage is to define a configuration of one or more sources that you want mirrored locally, then run the synchronisation function on daily or other regular intervals to keep your local copy in sync with the remote masters. Most of the download functionality is achieved by passing calls to `wget` and asking it to mirror the remote source, although a few of the pre-defined sources have more elaborate handlers.

It seems to me that this provides a nice complement to existing ropensci packages that provide on-demand

Assignees

No one assigned

Labels

0/presubmission

package

Projects

None yet

Milestone

No milestone

Procédé de revue

Vérifications par l'éditeur·rice

maelle commented Aug 4, 2017 • edited by sckott — Member

Editor checks:

- [x] **Fit:** The package meets criteria for fit and overlap
- [x] **Automated tests:** Package has a testing suite and is tested via Travis-CI or another CI service.
- [x] **License:** The package has a CRAN or OSI accepted license
- [x] **Repository:** The repository link resolves correctly
- [ ] **Archive** (JOSS only, may be post-review): The repository DOI resolves correctly
- [ ] **Version** (JOSS only, may be post-review): Does the release version given match the GitHub release (v1.0.0)?

Editor comments

Thanks for your submission **@raymondben**! I'm currently seeking one reviewer (already found one)!

Here is the output of `goodpractice::gp()` for you and the reviewers to consider (note: we don't expect you to have 100% unit test code coverage!)

```
It is good practice to

  ✖ write unit tests for all functions, and all package code in general. 79% of
    code lines are covered by test cases.

    R/aadc_eds_handler.R:14:NA
    R/aadc_eds_handler.R:15:NA
    R/aadc_eds_handler.R:16:NA
    R/aadc_eds_handler.R:17:NA
    R/aadc_eds_handler.R:27:NA
    ... and 385 more lines

  ✖ write short and simple functions. These functions have high cyclomatic
    complexity:bb_source (56).
  ✖ avoid long code lines, it is bad for readability. Also, many people prefer
    editor windows that are about 80 characters wide. Try make your lines
    shorter than 80 characters

    R\aadc_eds_handler.R:28:1
    R\aadc_eds_handler.R:32:1
    R\aadc_eds_handler.R:36:1
    R\aadc_eds_handler.R:38:1
    R\aadc_eds_handler.R:40:1
    ... and 480 more lines

  ✖ avoid calling setwd(), it changes the global environment. If you need it,
    consider using on.exit() to restore the working directory.

    R\aadc_eds_handler.R:44:5
    R\amps_handler.R:44:9
    R\provenance.R:50:5
    R\sync.R:63:5
    R\utils.R:77:5
    ... and 2 more lines
```

Reviewers: **@MilesMcBain @lwasser**
Due date: 2017-11-07

Revues

Seasonx commented Nov 7, 2017 • edited ▾

wonderful, thanks **@agreenbaum** Ok well I did focus a lot on #2 above, first if my feedback is helpful :) - I'm happy to continue to help (things as the package developer) :) . My apologies that this took longer than I knew you all needed it for.

Hi **@mdcramer @raymondben @ckohrt @mcefle**

I want to first and foremost thank the developers for putting so much time and effort into this package, and I'd like to thank the reviewer for asking me to do a review. My reviewer perspective is from that of a neuroscientist.
Specifically because I spend so much of my time teaching scientists, post docs, grads, undergrads, etc etc. I think a lot about what a typical scientists knows about data and APIs. Please take my review as a set of suggestions as I greatly appreciate the effort need / did not run all of the spell check, etc tests - just including spelling and such. Instead I spent a good bit of time downloading data and thinking about own points for users of various skill and knowledge levels.

Package Review

Please check off boxes as applicable, and elaborate in comments below. Your review is not limited to these topics, as described in the reviewer guide

- [x] As the reviewer I confirm that there are no conflicts of interest for me to review this work (If you are unsure whether you are in conflict, please speak to your editor before starting your review).

Documentation:

The package includes all the following forms of documentation:

- **A statement of need** clearly stating problems the software is designed to solve and its target audience in README
  - **I think I generally understand the need but the target audience isn't clear to me. (specifically the knowledge / skill of the target audience) and thus I think that if the need as stated is for a more general science audience, it is not quite addressed as of yet but could be!**
- **Installation instructions**: for the development version of package and any non standard dependencies in README
  - **see notes below - it installed well but I had issues with wget on MAC and the errors were not helpful.**
- **Vignette(s)** demonstrating major functionality that runs successfully locally
  - **see notes below - they need further development to be useful to a broader audience**
- **Function Documentation**: for all exported functions in R help
  - **see below - again it's a great start but needs more development!**
- **Examples**: for all exported functions in R Help that run successfully locally
  - **see notes below - they did once I got w_get working! so this could be checked if wget works**
- **Community guidelines**: including contribution guidelines in the README or CONTRIBUTING, and DESCRIPTION with `URL`, `BugReports`, and `Maintainer` (which may be autogenerated via `Authors@R`).
  - ?? may have missed these files in the repo? I don't see all of them ??
  - Description: https://github.com/hackathon-demo/dataforsciencedemo.dfsdata/issues/DESCRIPTION I didn't see a contributing file or description of how contributing works for this package. I also don't see `inst/AUTHORS`.

For packages co-submitting to JOSS

- [ ] The package has an **obvious research application** according to JOSS's definition

The package contains a `paper.md` matching JOSS's requirements with:

- [ ] A short summary describing the high-level functionality of the software
- [ ] Authors: A list of authors with their affiliations
- [ ] **A statement of need** clearly stating problems the software is designed to solve and its target audience.
- [ ] References: with DOIs for all those that have one (e.g. papers, datasets, software).

Is this being submitted to JOSS? How do I know that?

Functionality:

- **Installation**: Installation succeeds as documented.
  - I didn't check this because wget didn't install right away and it's not well documented.
- **Functionality**: Any functional claims of the software been confirmed.
  - I think functionality is generally there but I had issues with running this against different API's. I'll note in comments below where I had issues.
- **Performance**: Any performance claims of the software been confirmed.
  - I did not test performance.
- **Automated tests**: Unit tests cover essential functions of the package
  - There are unit tests in the package functions however I suggested on related to a directory setup below. My review in this regard was not comprehensive.
  and a reasonable range of inputs and conditions. All tests pass on the local machine.
- **Packaging guidelines**: The package conforms to the rOpenSci packaging guidelines.
  - Package naming I don't know how this name relates to what the package actually does!
  - NEWS is missing
  - I made some comments on style below
  - I didn't have time to go through the guidelines associated with testing and Travis.

Final approval (post-review)

- [x] **The author has responded to my review and made changes to my satisfaction. I recommend approving this package.**

Estimated hours spent reviewing so far I think I've spent about 5 hours reviewing with a few more playing with it when there was talk of it being split.

Review Comments

Overall Concept

In general I think this effort is a great one. I like the idea of a data download package. I think it's important to have a standardized way to download and downloaded data. It also supports increased usability and reproducibility of a project on the "left side" of the spectrum - ie getting the data initially. I support and applaud this effort!

However I have a few general concerns that I think could be addressed. I'd like to better understand the use cases for this effort and the users technical knowledge. In general, most people don't fully understand how large some of these data are. And often the starting place is to "download everything" without recognizing that for instance their hard drive is quickly filling up.

Also these days things are moving away from local processing and storage and towards cloud based activities. Thus, I think this package could provide some guidance for users BEFORE they download and attempt to archive everything.

For instance,

- Perhaps there are suggestions that ask a user to consider their hard drive space and storage and computation options before they download things over a certain size.
- Perhaps there are some guidelines and warning associated with downloading data and the space availability on the drive were the data are being downloaded.
- Perhaps there is some guidance associated with the types of data that might be appropriate to use with this package

I think some of these warning points are important to a successful package because so many users could quickly make huge API calls and fill up their hard drives unknowingly.

Use Cases

I'd like to see several clear use cases for this package that include the user personas who are accessing the data - a few much does the audience know for each use case? particularly as it relates to remote sensing data, I see many potential challenges for users who are less well versed in working with these data - particularly in the data size realm and dealing with API calls and associated arguments.

I'd also like to see a bit more documentation and warnings associated with potentially large requests and associated resource availability on a users machine.

Package Documentation

In general, the readme and introductory package text is written for someone with more technical experience and who has probably used API's before even though the audience as described is "researchers". If the audience is anyone who is looking to access data and build metadata around it, I think more clear documentation is required up front about what the package does and how to use it. I provide some specifics in this regard below

I also think that all of the function documentation needs to be developed further

Setup Feedback

First I installed the package from github. No issues with the installation.

- [ ] Minor but I do suggest adding the `library(leaverdata)` call to the readme vignette

```
install.packages("devtools")
library(devtools)
# install dependencies missing space
install_github("hackathon-demo/dataforsciencedemo/leaverdata", build_vignettes = TRUE)

# add this to the vignette just for clarity sake!
library(leaverdata)
```

also minor:

this URL doesn't work for me http://oceanscience.gov/

WGET Issues

When I ran through the vignette, I immediately ran into issues with `wget()`.

```
dir: wget: command not found
Error: install.packages("wget")
  could not find the wget executable. Try the `BW_install_wget()` function, or `install.packages("wget")`
```

This error seems to be focused on Windows users whereas I am on a MAC.
After checking a few bears I tried to use brew.

`brew install wget`

This approach tried to force me to update ruby which I wasn't prepared to do given other coming tasks (better API entwrangeing is underway over here!). The code below, however worked. I actually don't know why. It must have been the work of a helpful magician make-things-install-right gnome.

`brew install wget --with-libressl`

Note:

Revues

MikeMcMillan commented Nov 3, 2017 • edited ▸

Package Review

Please check off boxes as applicable, and elaborate in comments below. Your review is not limited to those topics, as described in the reviewer guide

- [ ] As the reviewer I confirm that there are no conflicts of interest for me to review this work (If you are unsure whether you are in conflict, please speak to your editor before starting your review).

Documentation:

The package includes all the following forms of documentation:

- [x] **A statement of need** clearly stating problems the software is designed to solve and its target audience in README
- [x] **Installation Instructions:** for the development version of package and any non-standard dependencies in README
- [x] **Vignette(s)** demonstrating major functionality that runs successfully locally
- [x] **Function Documentation:** for all exported functions in R help
- [x] **Examples:** for all exported functions in R Help that run successfully locally
- [x] **Community guidelines** including contribution guidelines in the README or CONTRIBUTING, and DESCRIPTION with `URL`, `BugReports` and `Maintainer` (which may be autogenerated via `Authors@R`).

For packages co-submitting to JOSS

- [ ] The package has an **obvious research application** according to JOSS's definition

The package contains a `paper.md` matching JOSS's requirements with:

- [x] **A short summary** describing the high-level functionality of the software
- [x] **Authors:** A list of authors with their affiliations
- [x] **A statement of need** clearly stating problems the software is designed to solve and its target audience.
- [x] **References:** with DOIs for all those that have one (e.g. papers, datasets, software).

Functionality:

- [x] **Installation:** Installation succeeds as documented.
- [x] **Functionality:** Any functional claims of the software been confirmed.
- [x] **Performance:** Any performance claims of the software been confirmed.
- [x] **Automated tests:** Unit tests cover essential functions of the package and a reasonable range of inputs and conditions. All tests pass on the local machine.
- [x] **Packaging guidelines:** The package conforms to the rOpenSci packaging guidelines

Final approval (post-review):

- [x] **The author has responded to my review and made changes to my satisfaction. I recommend approving this package.**

Estimated hours spent reviewing: 15.

Review Comments:

Thanks for your hard work **@mdusmuner** and **@hrpinceville**! I applaud the changes you have made since my initial feedback. My first experience with downloading the electoral data could not have been a bigger contrast to last time. It was one of those "By what magic is this made possible?" moments. I can see `tuseries` as a powerful tool for managing sources and syncing data from the web. You obviously get great mileage out of it purposes and I think that is a solid foundation for a great package. The main question mark I have after my review is about how accessible this package will be to the wider community.

This concern arises from a number of areas: I feel the documentation is too minimal for many functions. There are still some unconventional R styles used that hindered my ability to understand the package. A level of comfort with `rlang` seems assumed in the package that did not hold for me and I am not sure holds widely.

In my comments below I detail my concerns. Please take them for what they are: my attempts to improve the wider appeal of your valuable work.

Vignettes

In the self-titled vignette the use of the `postprocess` argument is only mentioned in passing with respect to decompression. I'd argue it is major functionality with enough caveats that deserves its own subheading.

As I mention in the function documentation section, this package could benefit from a greater breadth of examples with commentary that explains how features of the datasource result in each choice of method flag and postprocessing function.

Improve function documentation

In general the function documentation is minimal to the point of being somewhat unhelpful. I learned the most from the examples and source code. But this involved jumping back and forth between many functions. Here are some comments by function:

- `ds_settings` - Documentation does not explain the usecase for this function. It is not clear why it would be needed in addition to `ds_config`. The `sosmedia` example made this clearer.
- `ds_validate` - Documentation does not explain why this function does or is used for.
- `ds_license` - The use of the `ds_config` argument is mystifying. The function is specified as part of the config!
- `ds_resources` - As above.
- `ds_fingerprint` - Unclear how this function is expected to be used. It gets a mentioned in the "Data Integrity" section of the vignette but not explain how the output should be used to detect changes.
- `ds_update_cpi` - Unclear on the effect or valid usecase for `local_dir_only` arg. Did not appear to affect the operation of `ds_sync`.
- `ds_source` - The documentation for this key function is not sufficient for me to be able to use it confidently. `access_function` sounds like it is important but does not appear in any examples and the explanation is vague. The example uses some method flags that are not explained. I think this needs more examples and more commentary of used examples. Explain the nature of the web page and data and how that resulted in the choices of both `method_flag` and `postprocess`. Also, Users would benefit from multiple examples of increasing complexity so they can get a sense of the usecases of `tuseries`.

Explain `config` first argument

Either the help or vignettes or both need to clarify the mysterious first argument of `ds_handler_get`, the `ds_decompress` family and `ds_clean` - the vignette examples and the built-in example data sources use them to omit or without specifying this first argument even though it has no default. It was only after digging into the code of `ds_init` that I saw why this argument exists and how it is populated.

To make this clear you could mention in the help for the arg that it will typically be populated by `ds_init` and does not need to be specified in `postprocessing` or `method`.

On a related but more general note, you might also cover in a vignette that anyone needing to implement their own handler needs to write a function that must take the special arguments `config`, `verbose`, and `local_dir_only` in addition to any context specific args.

Ditch `dplyr` or update to tidyeval

`dplyr` is used very little within this package. So I question its worth as a dependency given the amount of installation time and number of recursive dependencies it adds. I can imagine `tuseries` being used with automated reproducible pipelines where this may be an issue.

Another issue is that the deprecated 'underscore' verbs (`select_`, `group_by_`, `select_`, `rename_`) are sprinkled around this package. For the sake of future package maintenance, these would need to be updated to the new 'tidyeval' style or removed. Given their very light usage I would recommend just replacing with `base::subset` with a view to removing `dplyr` altogether.

In the case of `dplyr::bind_rows()` can be used quite free for the small dataframes you are manipulating here.

Unusual usage of `quote`

At first I was confused as to the vignette example `quote` the functions supplied to the `method` and `postprocess`.

After viewing the examples in various function help files, my understanding now is that you intend `quote` to be the mechanism with which users pass arguments with their specified function. Is this correct? For me, this is quite unusual and not in keeping with the conventions I am familiar with in both base R or the `Tidyverse`. I found a few in `ds_license` examples that supplied the arguments to `postprocess` in the style of "happy family. I find this telling about how unexpected this pattern is.

I am far from an expert on R internals, but I am not sure if `quote` is safe way to do this since it may produce strange results when variable names are passed instead of literals due to lexical scoping. I think this needs mechanism needs to be reconsidered. If it is not changed to be more in line with conventional approaches, then it at least needs to be prominently flagged in the documentation.

Make some `wget` method flags into arguments

In `ds_config` you did a really nice thing where you created a local function argument (`clobber`) to provide a user API to the `wget` clobber arguments. The help for this argument is also quite detailed giving me confidence I could use it correctly if need be. I think this pattern is the bar that the rest of the package should aspire to.

In the self-titled vignette you state:

> You will probably want to specify `--timestamp` as part of the `method_flags`, even if the data source doesn't require a recursive download.

Before going on to explain the technical reason for this. So now `--recursive` is effectively toolsplate I will dutifully replicate in all my data sources until I get a error - because you said I should. This is not how you create a friendly API. Instead you should provide an argument `recursive` in `ds_config` and then allow the user to create an argument like `recursive = TRUE` with an explanation of the rare case I need to think about changing it in the function documentation.

Another candidate would be `--level=1` which seems to pop up frequently in the examples. If you made it a `recursive` argument you could explain the rare case where it might need to be different in the documentation. Yet another candidate would be `--delete` - a VERY useful argument.

I am not saying you need to provide a mapping to all `wget` args. From what I gather there are probably less than a half-dozen that will cover > 80% of usecase for your users. You could still have a `dwarg=wgs` for the remaining cases. But, in my view, the ease of use and documentation benefits of providing the common arguments would dramatically improve the appeal of this package. YOU are the experts on `wget`, and what you would be saying to us novices is: "Don't worry I've understood all the wget arguments so you don't have to. These are the ones you care about and this is how you use them". Would have a much more welcoming package that doesn't have to direct its users to a GNU manpage in the opening paras of its vignette!

The function for these args is line in is probably `ds_handler_wget()`. This might be a bit safer if you move away from the use of `quote` as I suggest above.

As things are there is also some weirdness with the fact that some `wget` flags are handled automatically by the handler eg: `rec if method=the_sdoc | method=ds_file_name=sosmedia` - `-progress=dot:giga` while others are expected to be supplied. What happens if they clash?

Misc Comments

- I did not find top-level documentation using `NewsMd` as per rOpenSci packaging guidelines
- Widespread usage of `cat(paste0(...))` for errors/warnings etc. Should replace with `message()`/`stop()` as per guidelines.
- Test coverage is good overall but skewed mostly to 'happy path'. Untested error paths appear in `sync.R`, `utils.R`, `wget.R`, `postprocess.R` etc. I feel like a gem for bringing this up. I know how unfun it is to write test cases for these.
- No `NeedsCompiler` in DESCRIPTION.

Procédé de revue

Discussion jusqu'à acceptation et transfert du dépôt

Procédé de revue

Discussion jusqu'à acceptation et transfert du dépôt

Souvent un article de blogue

https://ropensci.org/tags/review/

Un portrait par les données ?

Comment faire une *analyse de données* de *rOpenSci onboarding*?

Un portrait par les données ?

Comment faire une *analyse de données* de *rOpenSci onboarding*?

Rectanglons *onboarding*! *Rectangler, mot inventé par Jenny Bryan*

Image par Allie Brosh

Données dans le suivi des problèmes

ropensci / onboarding

Watch 25 | Star 123 | Fork 63

Code | Issues 30 | Pull requests 0 | Projects 0 | Wiki | Insights

is:issue is:open | Labels | Milestones | New issue

30 Open | 153 Closed | Author | Labels | Projects | Milestones | Assignee | Sort

- Submission: smapr [1/editor-checks] [package] [topic:data-access] [topic:geospatial] — 1
  #231 opened Jun 20, 2018 by mbjoseph — 0 of 8
- DataPackageR submission [1/editor-checks] [package] — 3
  #230 opened Jun 18, 2018 by gfinak — 11 of 19
- Pre-submission inquiry: repvar [package] — 2
  #229 opened Jun 18, 2018 by zkamvar
- Submission: tiler [2/seeking-reviewer(s)] [package] — 5
  #226 opened Jun 12, 2018 by leonawicz — 11 of 19
- submission: arkdb [3/reviewer(s)-assigned] [package] [topic:databases] — 4
  #224 opened Jun 8, 2018 by cboettig — 11 of 19

Tisser les fils de "problèmes" GitHub

Tisser les fils de "problèmes" GitHub

GitHub GraphQL API v4. Mieux que v3? Obtenez seulement les données voulues.

Tisser les fils de "problèmes" GitHub

GitHub GraphQL API v4. Mieux que v3? Obtenez seulement les données voulues.

Mon expérience

Tisser les fils de "problèmes" GitHub

GitHub GraphQL API v4. Mieux que v3? Obtenez seulement les données voulues.

Mon expérience

- Explorateur en ligne de l'API
- `ghql` https://github.com/ropensci/ghql
- jq play pour écrire des commandes jq pour dompter le JSON
- `jqr` https://github.com/ropensci/jqr

Résultat : fils de GitHub tissés

Magie de GitHub V4 + un peu de domptage de données...

Résultat : fils de GitHub tissés

Magie de GitHub V4 + un peu de domptage de données…

```
## # A tibble: 2,521 x 10
##    title created_at          closed_at           body  user  issue package
##    <chr> <dttm>              <dttm>              <chr> <chr> <int> <chr>
##  1 rrli~ 2015-03-31 00:25:14 2015-04-13 23:26:38 "- 1~ rich~     6 rrlite
##  2 rrli~ 2015-04-01 17:30:51 2015-04-13 23:26:38 "hey~ scko~     6 rrlite
##  3 rrli~ 2015-04-01 17:36:03 2015-04-13 23:26:38 "@sc~ kart~     6 rrlite
##  4 rrli~ 2015-04-02 03:36:09 2015-04-13 23:26:38 "Sur~ jero~     6 rrlite
##  5 rrli~ 2015-04-02 03:50:43 2015-04-13 23:26:38 "IMO~ gabo~     6 rrlite
##  6 rrli~ 2015-04-02 03:53:57 2015-04-13 23:26:38 "Ide~ rich~     6 rrlite
##  7 rrli~ 2015-04-02 18:58:53 2015-04-13 23:26:38 "> H~ kart~     6 rrlite
##  8 stpl~ 2015-04-08 23:56:17 2015-10-29 14:14:35 "- 1~ Robi~    10 stplanr
##  9 rrli~ 2015-04-10 21:52:39 2015-04-13 23:26:38 "@ri~ stew~     6 rrlite
## 10 rrli~ 2015-04-10 22:10:48 2015-04-13 23:26:38 "Tha~ rich~     6 rrlite
## #  … with 2,511 more rows, and 3 more variables: is_review <lgl>,
## #   commenter <chr>, role <chr>
```

Résultat : fils de GitHub tissés

```
## [1] 70
```

Données dans les dépôts accueillis

Données dans les dépôts accueillis

ropensci / **bowerbird**

Watch 4 | Star 14 | Fork 2

Code | Issues 5 | Pull requests 0 | Projects 0 | Wiki | Insights

Branch: **master**

Commits on Jun 7, 2018

try apt update for trusty travis build

raymondben committed Jun 7, 2018

0f277c6

Merge pull request #20 from maelle/master ...

raymondben committed Jun 7, 2018

Verified | fe2f8fb

Update GitHub links

maelle committed Jun 7, 2018

Verified | 68aeb56

Commits on Mar 5, 2018

Données dans les dépôts accueillis

- Clone via `git2r` https://github.com/ropensci/git2r

Données dans les dépôts accueillis

- Clone via `git2r` https://github.com/ropensci/git2r
- Remise à leur état avant la soumission grâce à `git2r`. `git reset hard`… Machine à remonter le temps !

Données dans les dépôts accueillis

- Clone via `git2r` https://github.com/ropensci/git2r
- Remise à leur état avant la soumission grâce à `git2r`. `git reset hard`... Machine à remonter le temps !
- Obtension de l'historique des modifications via `gitsum`. https://github.com/lorenzwalthert/gitsum

Données dans les dépôts accueillis

- Clone via `git2r` https://github.com/ropensci/git2r
- Remise à leur état avant la soumission grâce à `git2r`. `git reset hard`… Machine à remonter le temps !
- Obtension de l'historique des modifications via `gitsum`. https://github.com/lorenzwalthert/gitsum
- Comptage des lignes de code via `cloc`. https://github.com/hrbrmstr/cloc

Données dans les dépôts accueillis

- Clone via `git2r` https://github.com/ropensci/git2r
- Remise à leur état avant la soumission grâce à `git2r`. `git reset hard`… Machine à remonter le temps !
- Obtension de l'historique des modifications via `gitsum`. https://github.com/lorenzwalthert/gitsum
- Comptage des lignes de code via `cloc`. https://github.com/hrbrmstr/cloc
- Comptage des fonctions et classes exportées par `devtools::parse_ns_file`

Regardons les données

- Le code est dispo dans https://ropensci.org/blog/2018/04/26/a-satrday-ct-series/
- Paquets utilisés pour les visualisations : `ggplot2`, `hrbrthemes`, `viridis`.

Combien de travail pour ce système ?

Travail dans les dépôts

Âge à la soumission

Âge à la soumission

Temps de revue

Quelle taille font les paquets ?

Travail des auteurs

Travails pour les réviseur·se·s

Temps de revue vs. nombre d'exports

Notes sur la quantification du travail

Notes sur la quantification du travail

- Difficile de définir des mesures.

Notes sur la quantification du travail

- Difficile de définir des mesures.
- Travail fait par des *volontaires* !

Notes sur la quantification du travail

- Difficile de définir des mesures.
- Travail fait par des *volontaires* !
- Diminution du temps par automatisation.

Notes sur la quantification du travail

- Difficile de définir des mesures.
- Travail fait par des *volontaires* !
- Diminution du temps par automatisation.
- Et les éditeur·rice·s alors ? Selon Tim Trice (traduit), *"anges gardiens du début à la fin du procédé d'accueil"*. Un paquet tous les un ou deux mois.

Procédé de qualité… et sympa ?

Procédé de qualité… et sympa ?

Météo sociale de *rOpenSci onboarding*

Essai d'Ann Gentle dans http://open-advice.org/

Impressions, exemples... approche plus générale ?

Météo sociale de *rOpenSci onboarding*

Essai d'Ann Gentle dans http://open-advice.org/

Impressions, exemples... approche plus générale ?

Analyse de texte !

Météo sociale de *rOpenSci onboarding*

Essai d'Ann Gentle dans http://open-advice.org/

Impressions, exemples... approche plus générale ?

Analyse de texte !

- `tidytext` et `sentimentr`
- https://www.tidytextmining.com/ de Julia Silge et David Robinson

Mots les plus communs

Bigrammes les plus communs

Corrélations par paire

Sentiment

Sentiment

Mots des lignes négatives

Lignes les plus négatives

```
## # A tibble: 15 x 2
##    line                                                          sentiment
##    <chr>                                                             <dbl>
##  1 @ultinomics no more things, although do make sure to add mor~    -1.63
##  2 not sure what you mean, but i'll use different object names ~    -1.20
##  3 error in .local(.object, ...) :                                  -1
##  4 error:                                                           -1
##  5 #### miscellaneous                                               -1
##  6 error: command failed (1)                                        -0.866
##  7 - get_plate_size_from_number_of_columns: maybe throwing an e~    -0.786
##  8 "this code returns an error, which is good, but it would be ~    -0.744
##  9 0 errors | 0 warnings | 0 notes                                  -0.722
## 10 once i get to use this package more, i'm sure i'll have more~    -0.721
## 11 - i now realize i've pasted the spelling mistakes without th~    -0.707
## 12 minor issues:                                                    -0.707
## 13 ## minor issues                                                  -0.707
## 14 replicates issue                                                 -0.707
## 15 visualization issue                                              -0.707
```

Lignes les plus positives !

```
## # A tibble: 15 x 2
##    line                                                          sentiment
##    <chr>                                                             <dbl>
##  1 absolutely - it's really important to ensure it really has b~      1.84
##  2 overall, really easy to use and really nicely done.                1.73
##  3 this package is a great and lightweight addition to working ~      1.46
##  4 i am very grateful for your approval and i very much look fo~      1.26
##  5 thank you very much for the constructive thoughts.                 1.24
##  6 thanks for the approval, all in all a very helpful and educa~      1.22
##  7 - really good use of helper functions                              1.14
##  8 - i believe the utf note is handled correctly and this is ju~      1.13
##  9 seem more unified and consistent.                                  1.13
## 10 very much appreciated!                                             1.13
## 11 - well organized, readable code                                    1.1
## 12 - wow very extensive testing! well done, very thorough             1.1
## 13 - i'm delighted that you find my work interesting and i'm ve~      1.08
## 14 thank you very much for your thorough and thoughtful review,~      1.08
## 15 great, thank you very much for accepting this package. i am ~      1.07
```

Engagez-vous, qu'il·elle·s disaient !

rOpenSci

Dépôt officiel https://github.com/ropensci/onboarding

Blogue https://ropensci.org/tags/onboarding/ en particulier https://ropensci.org/blog/2018/04/26/a-satrday-ct-series/

Lisez notre livre https://ropensci.github.io/dev_guide/

Portez-vous volontaire pour être réviseu·r·se https://ropensci.org/onboarding/

Éditeur·rice·s

```
thank(c("karthik", "noamross", "sckott"))
```

```
thank(c("lmullen", "annakrystalli", "ropensci"))
```

Merci !

bit.do/rr2018 | ma_salmon | maelle | masalmon.eu