

Use {fusen} to write or upgrade a package

Rmd-first Method; UseR 2021

Sébastien Rochette



This presentation on Github: [statnmap/prez](https://github.com/statnmap/prez)

Save some time

Write the code and the doc at the same time

Sébastien

Team leader, R expert, R instructor.

- ThinkR Website: <https://rtask.thinkr.fr>
- ThinkR GitHub: <https://github.com/ThinkR-open>
- ThinkR Twitter: [@Thinkr_FR](https://twitter.com/Thinkr_FR)
- Personal website: <https://statnmap.com>
- Personal Twitter: [@statnmap](https://twitter.com/statnmap)



What do you need to think about as developers?

Answer future users questions about your package

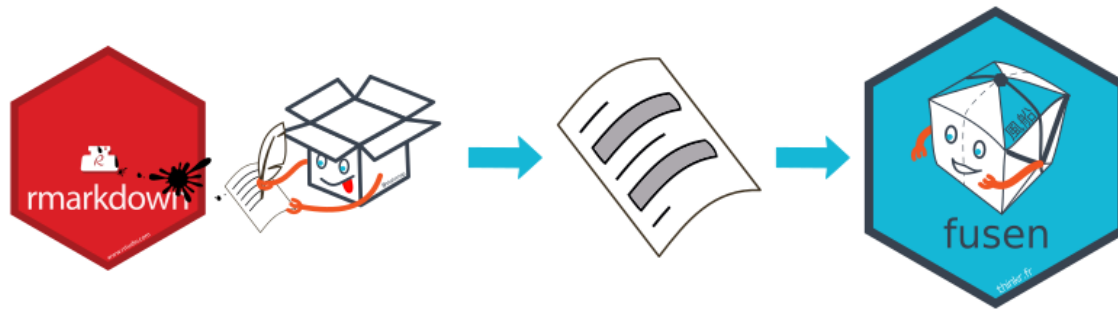
Questions	Answers
What does it do?	DESCRIPTION
How to install it with its dependencies?	DESCRIPTION
What are its function?	'R/' directory
How to fill parameters of this function?	'R/': Roxygen skeleton
Can I have an example on how to use this function?	'R/': <code>@examples</code>
Can I have an example on how to use the package as a whole?	'vignettes/' directory
Will it work with the last version of R and dependencies?	'tests/' directory and Continuous Integration

A minimum of 4 different places to store code and documentation

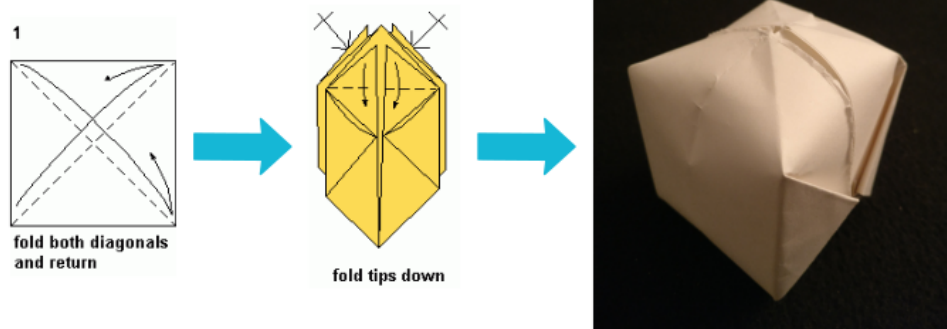
Why not preparing all your codes, examples, tests, and doc in the same file?

{fusen} allows to write everything in the same place

- Write everything in a Rmarkdown
- Follow the folding lines
- Inflate



It is like origami



Follow {fusen} folding lines

Four different places to store code and documentation

- **DESCRIPTION**: package documentation
- **'R/'** directory: functions and examples
- **'tests/'** directory: unit tests
- **'vignettes/'** directory: documentation

{fusen} needs to distinguish these places to be able to correctly distribute

Follow {fusen} folding lines

- Create a new classical project / directory
- Run `fusen::add_dev_history()`
 - Use it as a template for your {fusen}
 - [Follow the guide](#)

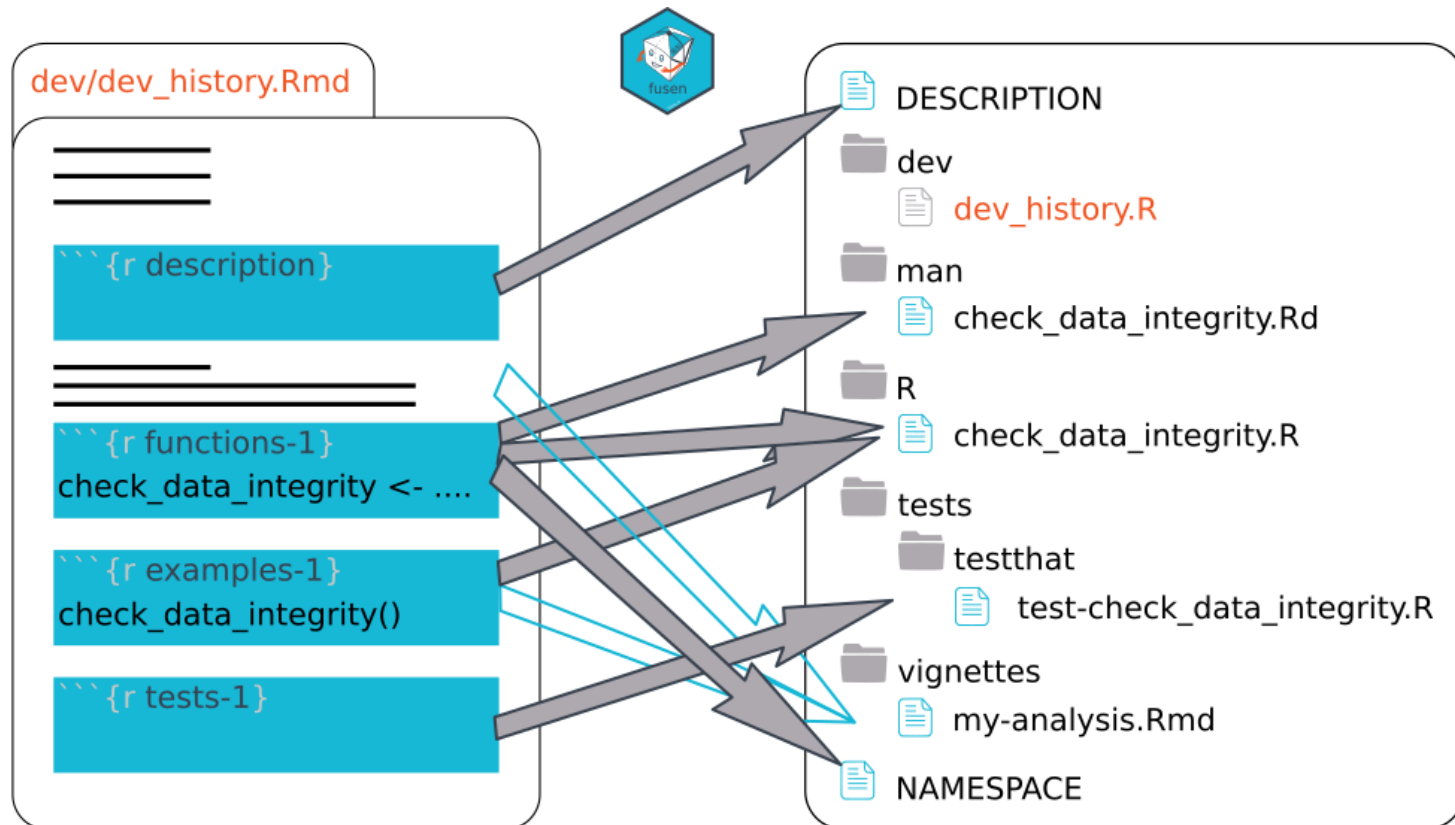
```
--
21 ▾ ```{r description}
22
23 ▴ |``
24
25 ▾ # My function
26
27 ▾ ```{r function-1}
28
29 ▴ ```
30
31 ▾ ```{r examples-1}
32
33 ▴ ```
34
35 ▾ ```{r tests-1}
36 ▾ test_that("my_function works properly", {
37
38 ▴ })
39 ▴ ```
40
```

Follow {fusen} folding lines

- Inflate !

```
fusen::inflate(rmd = "dev/dev_history.Rmd")
```

functions



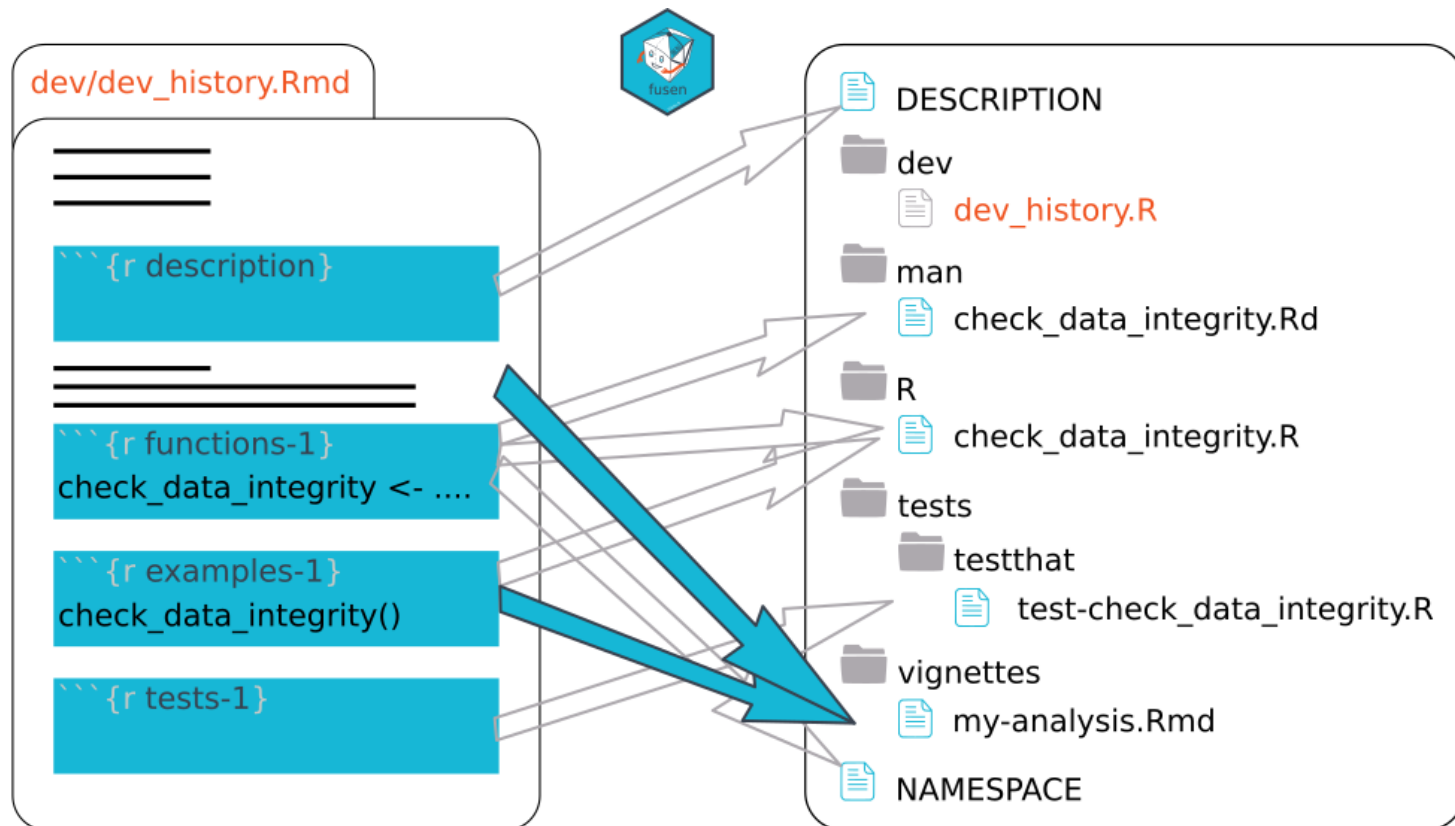
@statnmap

Follow {fusen} folding lines

- Inflate !

```
fusen::inflate(rmd = "dev/dev_history.Rmd")
```

vignette



@statnmap

Code and document in the same place

Reduce the number of copy pasting, files opened, files switch

Code and document in the same place

Reduce the number of copy pasting, files opened, files switch

- Do not let unit testing for when your package is finished!

Code and document in the same place

Reduce the number of copy pasting, files opened, files switch

- Do not let unit testing for when your package is finished!

Write your documentation as you code

Code and document in the same place

Reduce the number of copy pasting, files opened, files switch

- Do not let unit testing for when your package is finished!

Write your documentation as you code

- Do not let vignette writing for when your package is finished!



How do I add new fonctionnalités ?

- **Add new functions to an existing package**, even not built with {fusen}
 - See vignette [Maintain-packages-with-fusen](#)

How do I add new fonctionnalités ?

- **Add new functions to an existing package**, even not built with {fusen}
 - See vignette [Maintain-packages-with-fusen](#)
- **Re-inflate as many times as needed**

How do I add new functionalities ?

- **Add new functions to an existing package**, even not built with {fusen}
 - See vignette [Maintain-packages-with-fusen](#)
- **Re-inflate as many times as needed**
- **Add new sections in the Rmd for new functions**
 - Use `fusen::add_dev_history(name = "full")` for a full working example package

How do I add new functionalities ?

- **Add new functions to an existing package**, even not built with {fusen}
 - See vignette [Maintain-packages-with-fusen](#)
- **Re-inflate as many times as needed**
- **Add new sections in the Rmd for new functions**
 - Use `fusen::add_dev_history(name = "full")` for a full working example package
- **Add new "dev_history_*.Rmd" each time you need**
 - `fusen::add_dev_history(name = "additional")`

How do I add new functionalities ?

- **Add new functions to an existing package**, even not built with {fusen}
 - See vignette [Maintain-packages-with-fusen](#)
- **Re-inflate as many times as needed**
- **Add new sections in the Rmd for new functions**
 - Use `fusen::add_dev_history(name = "full")` for a full working example package
- **Add new "dev_history_*.Rmd" each time you need**
 - `fusen::add_dev_history(name = "additional")`
- **Bonus: Do not think too much about dependencies** in DESCRIPTION

How do I add new functionalities ?

- **Add new functions to an existing package**, even not built with {fusen}
 - See vignette [Maintain-packages-with-fusen](#)
- **Re-inflate as many times as needed**
- **Add new sections in the Rmd for new functions**
 - Use `fusen::add_dev_history(name = "full")` for a full working example package
- **Add new "dev_history_*.Rmd" each time you need**
 - `fusen::add_dev_history(name = "additional")`
- **Bonus: Do not think too much about dependencies** in DESCRIPTION
- **{fusen} itself is created and updated using {fusen}**
 - Multiple "dev_history.Rmd" files as source: <https://github.com/ThinkR-open/fusen/tree/master/dev>

Common questions about {fusen}

- How about **reorganising my existing vignettes**?
 - Go back to "dev_history.Rmd", re-organise and re-inflate
- How about **reorganising between vignettes**?
 - You can lock the "dev_history.Rmd" and start developing as usual
- How about **debugging my code**?
 - You can still debug in the global environment with "dev_history.Rmd"
 - Be careful where RStudio leads you...
- What about **not {fusen} existing functions**?
 - There is no `deflate()` function yet...
 - You can create back the Rmd skeleton if you want

Can I teach {fusen} to new developers?

- Yes, you can.
- Teaching template: `fusen::add_dev_history(name = "teaching")`
- [{fusen.squirrels}](#) on my statnmap GitHub

master

Commits on Mar 2, 2021

- Add Actions for GitHub**
statnmap committed on 2 Mar ✓ [c45be22](#) <>
- Present the package in a website**
statnmap committed on 2 Mar [ea31a1a](#) <>
- Present the package with README**
statnmap committed on 2 Mar [66ae0f8](#) <>

Commits on Mar 1, 2021

- Inflate the package**
statnmap committed on 1 Mar [6314121](#) <>
- Add unit tests**
statnmap committed on 1 Mar [cbd16b0](#) <>
- Add Roxygen Skeleton**
statnmap committed on 1 Mar [c599989](#) <>
- Show reproducible example**
statnmap committed on 1 Mar [e489b55](#) <>
- Transform as generic function**
statnmap committed on 1 Mar [caa30ef](#) <>
- Write regular R code**
statnmap committed on 1 Mar [b3f1f7b](#) <>
- Add sample data in the package**
statnmap committed on 1 Mar [12c3173](#) <>
- add DESCRIPTION and LICENSE**
statnmap committed on 1 Mar [c7151ce](#) <>
- add dev/dev_history.Rmd**
statnmap committed on 1 Mar [373efdb](#) <>

Give {fusen} a try for your next new
awesome fonctionnnality or package

And tell me...

'Rmd first' method for every project using {fusen}

Documentation and tests is the key for your work to be trusted and used Documentation reduces calls, mails and tweets

Code everything in the same place, avoid multiple copy-paste

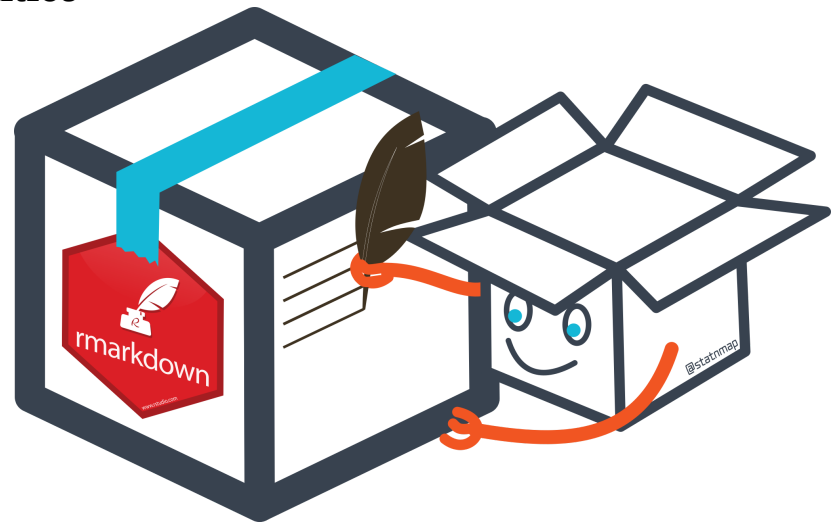
- Write everything you have in mind in the Rmd
- Write what you are about to code as plain text as a first draft
- Document `function` parameters as you add them
- Write `examples` as you try them
- Write unit `tests` as you add new possibilities

THINK Package with {fusen} !

THANK YOU for your attention

See more:

- rtask.thinkr.fr
- <https://thinkr-open.github.io/fusen>



This presentation on Github: [statnmap/prez](https://github.com/statnmap/prez)