mapview

Tim Appelhans

Christoph Reudenbach & Florian Detsch

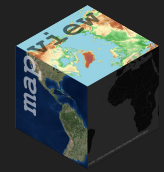
Who am I virtually?

tim-salabim (r-spatial)

@TimSalabim3

tim.appelhans@gmail.com

Who am I historically?



Since 2019: Director Data Science at METER Group https://www.metergroup.com/

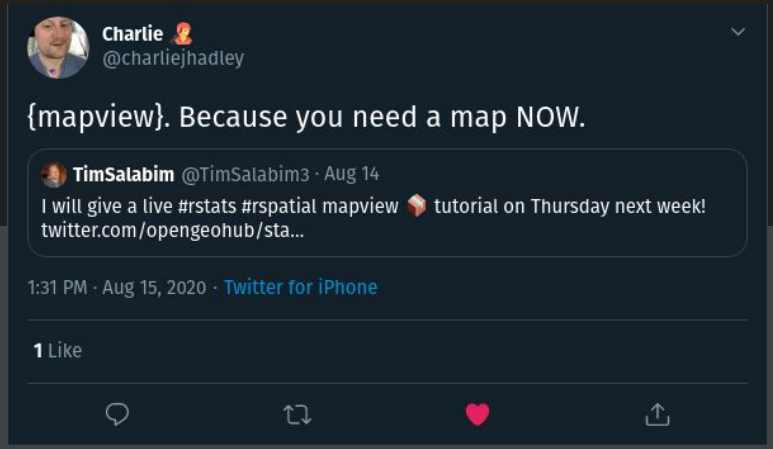
2017 - 2018: Geodata Scientist at GfK

2010 - 2016: Lecturer at Environmental Informatics, University of Marburg

2005 - 2010: PhD Geography, University of Canterbury

1998 - 2005: Dipl. Geography, FAU University Erlangen-Nürnberg

Why mapview?



https://twitter.com/charliejhadley/status/1294597569105068032

- Because R has no built-in interactive plotting functionality
- Because I was annoyed by having to switch between R & QGIS
- Because RStudio released {leaflet}
- Because {leaflet} is too verbose
- Because we need to be able to visualise things quickly and conveniently
- Because background maps can be a great visual aid
- Because creating plotting packages is fun!

mapview then

2015 - Immediately after UseR! Conference Aalborg:

First commit of a mapview-esque function:

https://github.com/environmentalinformatics-marburg/Rsenal/commit/58422a3a217cf0adaebe2c6634b0ecedf4e23a16#diff-7aa7de3809372fb3a13719a428a2c274

[R-sig-Geo] mapView: basic interactive viewing of spatial data in R

Tim Appelhans tim.appelhans at gmail.com
Fri Jul 24 11:39:46 CEST 2015

- Previous message: [R-sig-Geo] Split polygon by line
- Next message: [R-sig-Geo] mapView: basic interactive viewing of spatial data in R
- **Messages sorted by:** [date] [thread] [subject] [author]

Dear list members,

I would like to draw your attention to a little 'project' I've been
working on over the past few weeks.

Using leaflet for R, I have defined some methods for spatial data (e.g.
Raster*, Spatial* objects) to quickly visualise them in either the
RStudio Viewer pan or the default web browser. The function I created
for this is called mapView().

Think of it as an interactive version of spplot()/plot() for spatial
data (though less versatile). It enables zooming, paning and basic layer
queries (i.e. printing of the attributes in the @data slot of Spatial*
objects + the x/y location of the feature). For Raster* objects queries
are currently not available as leaflet translates the data into RGB
values for display. Furthermore, background maps can be defined and
multiple different spatial object layers can be overlaid.
At the moment mapView() lives in our Rsenal package on github
(https://github.com/environmentalinformatics-marburg/Rsenal) but this is
likely going to change at some point in the not too distant future. This
also depends on whether there is active interest in developping this
sort of thing further to provide more than the current admittedly rather
limited functionality. This, however, would involve JavaScript coding
which I do not have any experience with.

A quick (non-interactive) intorduction can be found here:

https://metvurst.wordpress.com/

The full introductory article including interactive examples is
published here:

http://environmentalinformatics-marburg.github.io/web-presentations/20150723_mapView.html

I hope this may prove useful for some of you.
Also, if anyone is keen to get involved in taking this further, please
let me know and we will see how to best proceed.

Best,
Tim

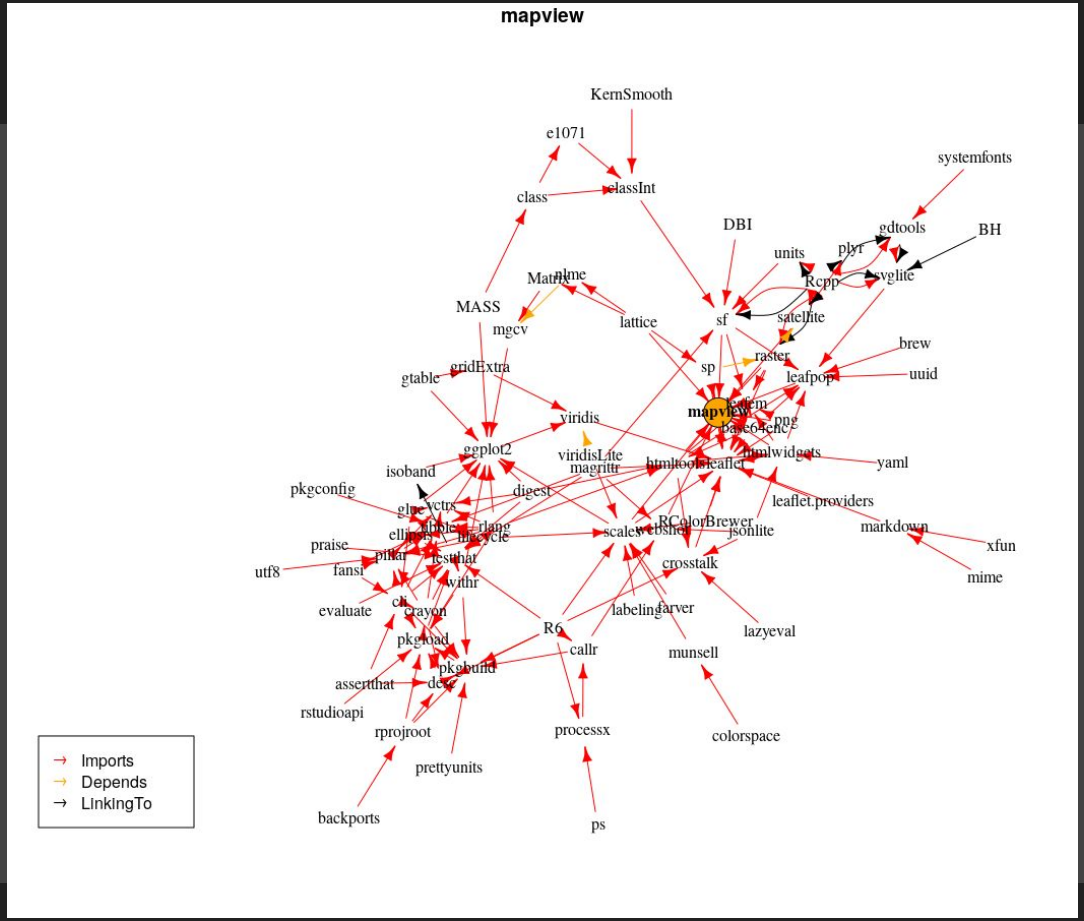
--
###################################
Tim Appelhans
Department of Geography
Environmental Informatics
Philipps Universität Marburg
Deutschhausstraße 12
35032 Marburg (Paketpost: 35037 Marburg)
Germany

mapview now

2020 - a full-blown package with lots of dependencies (probably too many).

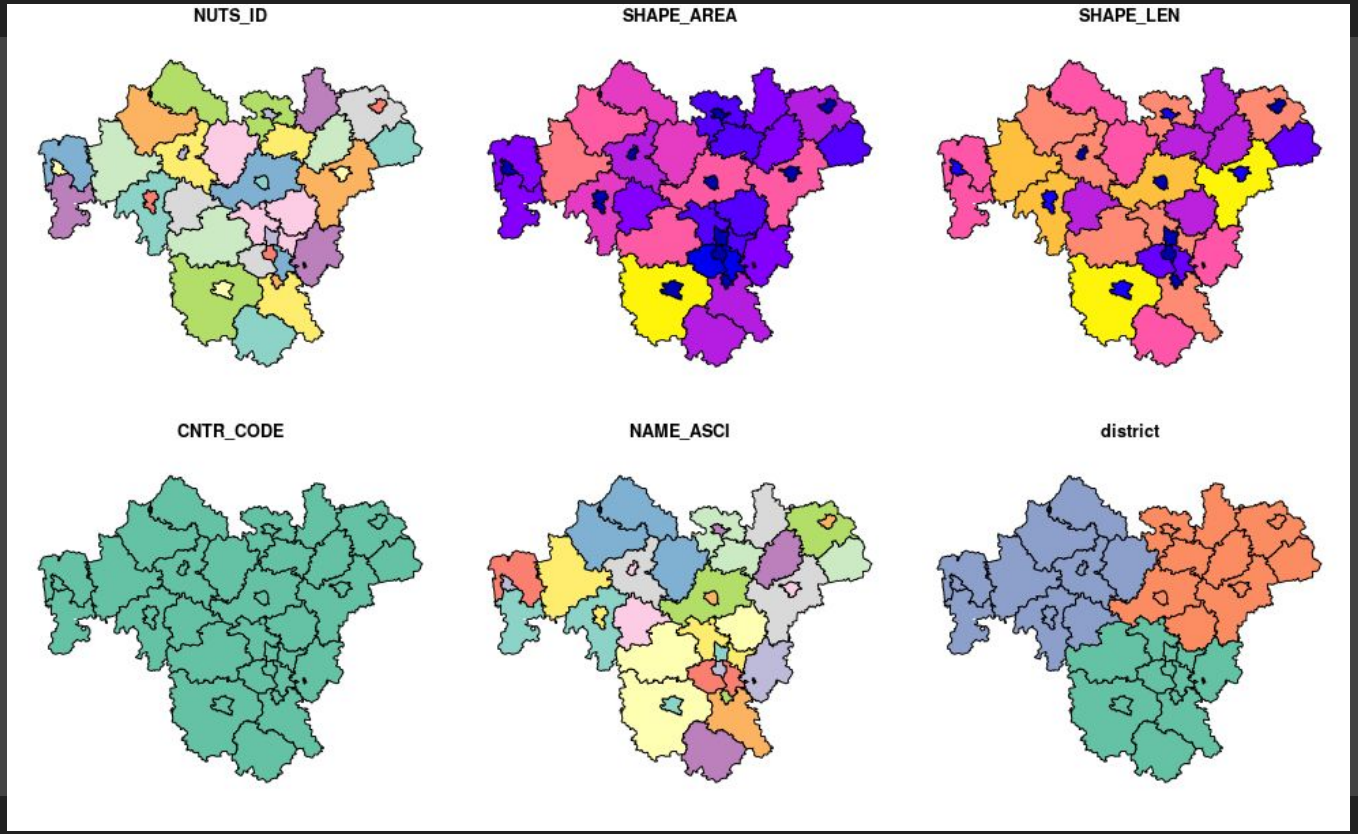
Most important:

* leafem
* leafpop
* leafgl
* (leaflet)



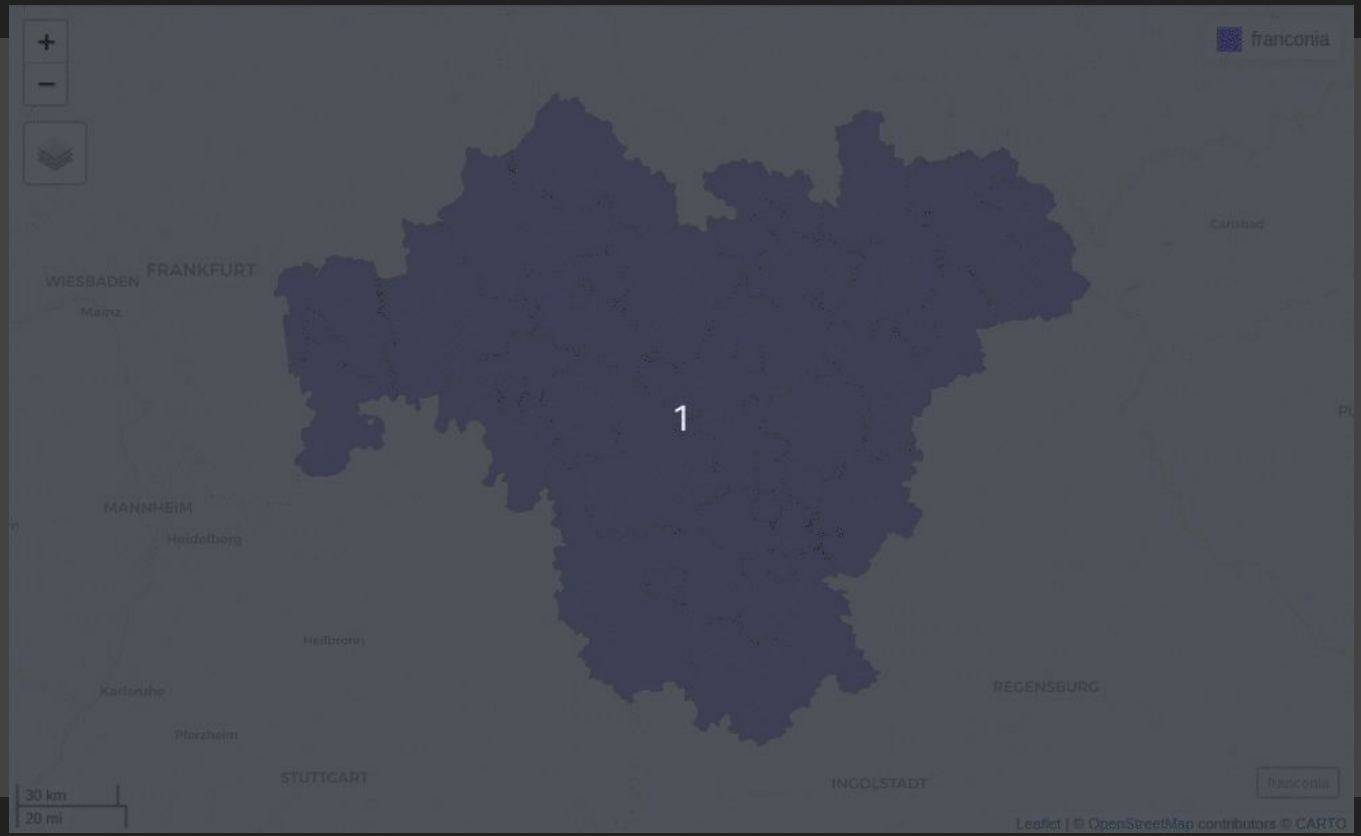
What's in the box?

plot(franconia)



What's in the box?

mapview(franconia)



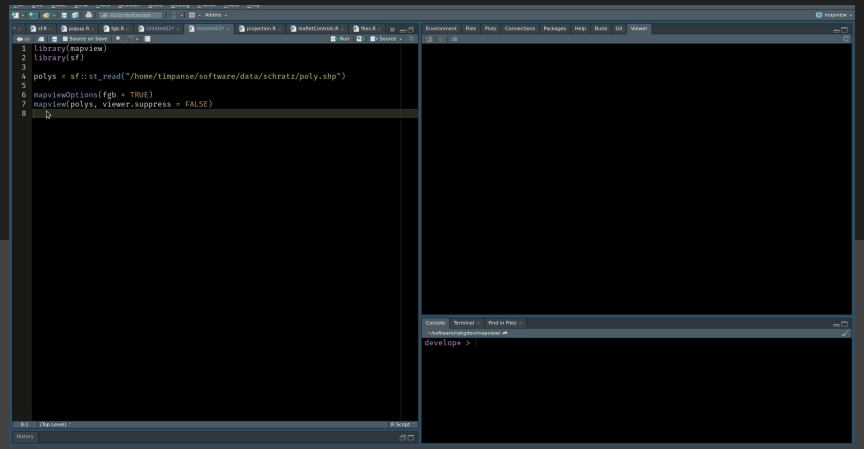
Should be able to throw pretty much anything at mapview...

What's in the box?

- Data-driven leaflet API
- One liner, yet customisable
- Add layers/maps with + operator
- Supports many spatial data types
- Customisable via global options

More recent developments:

- Data streaming (using flatgeobuf)
- Scalability (leafgl & mapdeck)
- Support for large rasters (still in alpha)
- Compare maps with | operator



What next?

- JavaScript!!
  - flatgeobuf colors (chroma.js)
  - stars adaptive rendering (more layers, pass JS functions)
  - make leafgl work together with flatgeobuf
  - iron out issues related to data attachments
  - popupTables!
- Drop raster dependency -> convert internal code to stars
- General speed up -> @_davecooley has some awesome stuff in the pipeline!!
- Enhance automagic renderer (browseURL -> servr::httw)
- leaftime integration for space-time data?
- Clean code base https://xkcd.com/1695/
- Write more tests (testthat -> tinytest)
- Update pkgdown site
- Feature requests...

Thank you!

Code & slides available at:
https://github.com/tim-salabim/2020_opengeohub