Interactive document generation with R Markdown

Julia Schulte-Cloos
17 December 2019

slides: git.io/Je7dv

julia.schulte-cloos@gsi.uni-muenchen.de | jschultecloos | @jschultecloos

Contents

Why interactive documents?

Why interactive documents?

We are living in an interactive world

- interactive presentation slides are more engaging...
- ...and they allow the presenter to integrate the audience into the arguments

Show ▼ entries Search:

	Municipality	Election Type	Year	AfD Share	Turnout
	All	All	All	All	All
1	Brandenburg an der Havel	Federal	2013	4.43	60.72
2	Cottbus	Federal	2013	6.84	65.75
3	Frankfurt (Oder)	Federal	2013	6.25	64.88
4	Potsdam	Federal	2013	5.56	57.37
5	Ahrensfelde	Federal	2013	7.25	79.37

Showing 1 to 5 of 5,016 entries Previous 1 2 3 4 5 … 1004 Next

Some advantages of interactive presentations

Exploring and presenting data

- powerful tool for data exploration and for **descriptive presentation of data**
- **science communication** increasingly becoming important
- different interactive output formats within RMarkdown
    - most basic: html
    - ➞ focus of this workshop: **interactive presentations**
    - ...`htmlwidgets` can be embedded also in plain html documents or in `flexdashboards`

How does `xaringan` compare to other HTML5 presentation frameworks?

...e.g. reveal.js, impress.js, inspire.js, bespoke.js

`xaringan` relies on R Markdown

➞ presentations are **fully reproducible**

Getting started with xaringan

What is xaringan?

`xaringan` creates HTML5 slides with R Markdown and the JavaScript library 'remark.js'

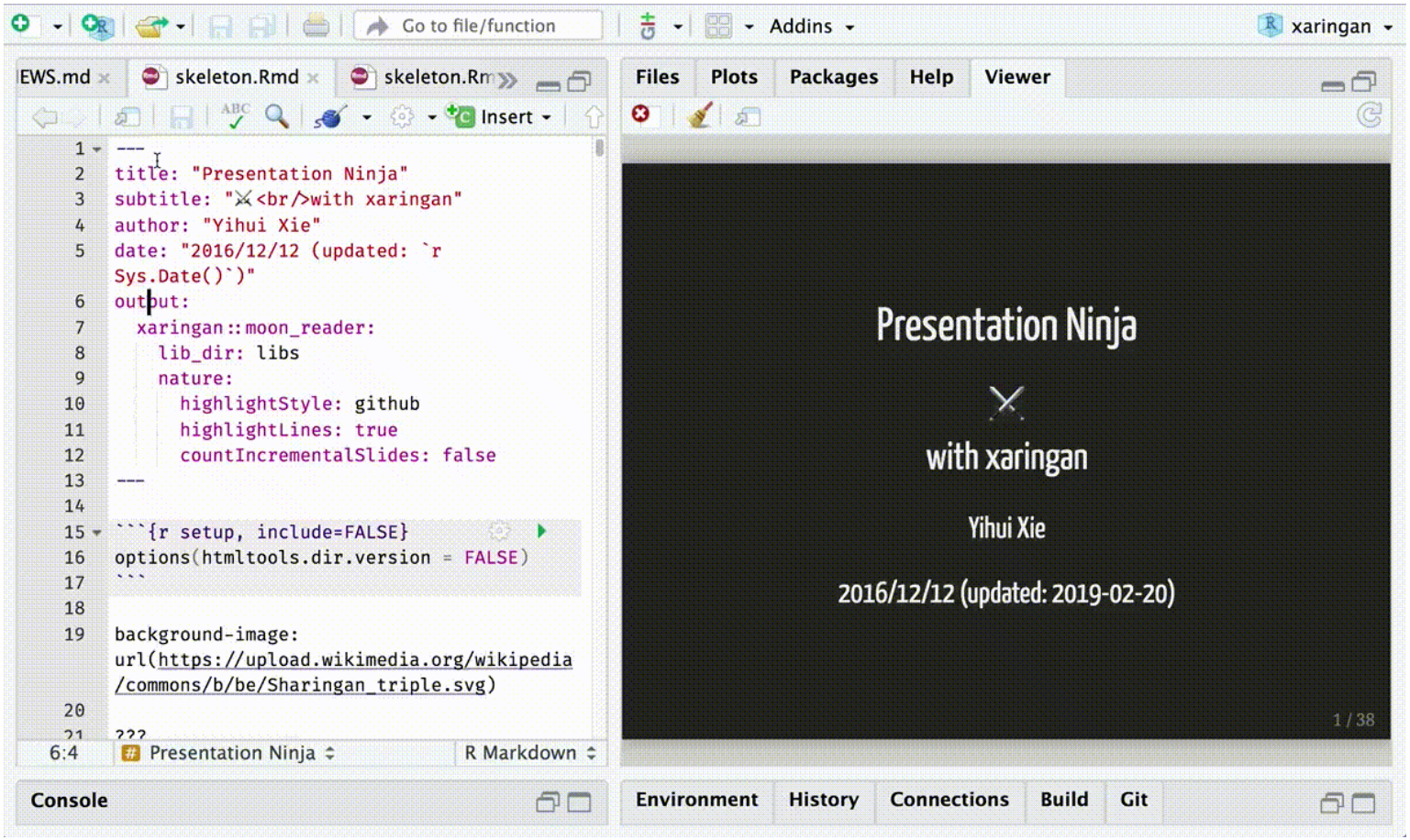
RMarkdown output format for interactive slide generation

- R Markdown output format `xaringan::moon_reader`
- Wrapper for the JavaScript library remark.js

Useful resources for getting started with R Markdown

- **Cheatsheet** for RMarkdown
- Official reference guide

Workflow with RStudio and Moonreader



➞ read more on Yihui Xie's blog post

The YAML header

- Sets **global parameters** of the document
- E.g. `output, title, author, date`
- YAML is a syntax (YAML Ain't Markup Language, YAML)
- Tag-value pairs separated by colons

```
---
title: "My first interactive presentation"
output:
  xaringan::moon_reader:
    encoding: 'UTF-8'
    css: [default, xaringan-themer.css, extra.css]
    lib_dir: libs
    nature:
      ratio: '16:9'
      highlightStyle: github
      slideNumberFormat: "%current%/%total%"
---
```

❓ In doubt about YAML validitiy?

➞ use a YAML linter

- e.g. this online tool, a text-editor optimised for linting YAML (or a plugin)
- or a python-based application ➞ help page `?xaringan::moon_reader` for all possible options that can be specified in the YAML header

Getting started with xaringan

➞ Install latest version of the package in `R`

```
remotes::install_github('yihui/xaringan')
```

➞ In RStudio: start from a template

- Menu `File -> New File -> R Markdown -> From Template -> Ninja Presentation`
- `Knit` button compiles the document
- alternatively, use the Addin `Infinite Moon Reader` to live preview the slides (update of the live preview after each save of the Rmd)
- `xaringan::inf_mr()`

Global style of the slide deck

Option `nature` defines important **global** arguments

```
---
output:
  xaringan::moon_reader:
    nature:
      ratio: '16:9'
      slideNumberFormat: '%current%'
      highlightStyle: ir-black
      highlightLines: true
      countIncrementalSlides: false
---
```

- **nature**: `xaringan` format option that is passed to `remark.create()`
- different options, e.g.:
  - `ratio = '16:9'`
  - `countdown`: includes a countdown timer on each slide (number of milliseconds, e.g. countdown: 60000 for one minute)

➞ for more details, see full documentation of remark.create()

Styling code

Code syntax highlighting with highlight.js

```
output:
  xaringan::moon_reader:
    lib_dir: libs
    css: xaringan-themer.css
    nature:
       highlightStyle: github
```

- different code highlighting options
- `arta`, `ascetic`, `dark`, `default`, `far`, `github`, `googlecode`, `idea`, `ir-black`, `magula`, `monokai`, `rainbow`, `solarized-dark`, `solarized-light`, `sunburst`, `tomorrow`, `tomorrow-night-blue`, `tomorrow-night-bright`, `tomorrow-night`, `tomorrow-night-eighties`, `vs`, `zenburn`

→ for a full list, see the demo of `highlight.js`

Text formatting

- Headings start with a #
- Number of # specifies the heading level

Level 3 heading

Level 4 heading

Level 5 heading

Level 6 heading

Fully customizable in the CSS

```
h1 {
  font-weight: 800;
  font-size: 200% !important;
  margin-top: 0.9em;
  margin-bottom: 0.9em;
}
```

Lists

Markdown syntax to create lists

Bulleted List

- Bullet 1
- Bullet 2
    - Sub-bullet 2.1
    - Sub-bullet 2.2

Incremental Bullets

- Incremental Bullet 1
- Incremental Bullet 2

```
- Bullet 2
    + Sub-bullet 2.1
    + Sub-bullet 2.2

- Incremental Bullet 1
--

- Incremental Bullet 2
```

⚠ **no spaces** after two short dashes!

Slide formatting

Useful resource: wiki of remark.js remark.js Wiki

Alignment of a slide

- defined in the `class` argument just at the beginning of a new slide

```
---
name: results
class: center, middle, inverse

# Results
```

→ classes for **vertical** alignment: `top`*, `middle`, `bottom`

→ classes for **horizontal** alignment: `left`*, `center`, `right`

Useful slide properties

- **hiding a slide**: set the property of a slide to `exclude: true`

```
---
name: excludedslide
exclude: true

# A slide that should be excluded

- some content that we decided to exclude
- we do not need to delete all text
- instead, we simple set the exclude slide property to true
```

- `count: false` prevents a slide from being included in the slide count

Background images

`background-image: url(./figs/munich.jpg)`

- `background-size: cover` ➞ rescales + crops with no empty space
- `background-size: contain` ➞ rescales only

Useful slide properties

- **hiding a slide**: set the property of a slide to `exclude: true`

```
---
name: excludedslide
exclude: true

# A slide that should be excluded

- some content that we decided to exclude
- we do not need to delete all text
- instead, we simple set the exclude slide property to true
```

- **count**: `count: false` prevents a slide from being included in the slide count
- **background image**: `background-image: url(./figs/munich.jpg)`
    - `background-size: cover` → rescales + crops with no empty space
    - `background-size: contain` → rescales only

→ see the complete list of available slide properties in remark.js

Styling content

You can style and full customize content in your CSS

Text alignment

.left[Left-aligned text] .center[Centered text] .right[Right-aligned text]

Footnote

.footnote[]

Icons

```
devtools::install_github("ropenscilabs/icon")
icon::fa("spinner", size = 2,
         animate = "spin")
```

Spacing

```
.spaced {
    line-height: 150%;
}
```

Sidebar Layout

Point 1

Here is some information about why we need the first point, which brings us to...

Sidebar Layout

Point 1

... the second point.

Point 2

➞ incremental effects by using different pages

- left column is 20%
- right column is 75%
- if you use headings in the left column (`## level two` or `### level three`) ➞ last heading **highlighted**, previous headings **grayed out**

Including images

- via **simple Markdown** syntax

```
![Image description](figs/elephant.jpg)
```

- via the `knitr package` for captions, sizing, aligning

```
{r elephant-chunk, out.width='20%', fig.align='center', fig.cap='Elephant in the room'}
knitr::include_graphics('figs/elephant.jpg')
```

Elephant in the room

- directly via HTML:

```
<img src="./figs/elephant.jpg" height="150">
```

Presentation mode & keyboard shortcuts

Cross-linking slides

- add a slide name individual slides, then cross link to them too from other slides in your deck:

```
[go back to slide on formatting](#slideformatting)
```

→ go back to slide on formatting

Presenter notes

- add notes and comments to yourself that are shown in presenter mode
- keyboard shortcut p while presentation is on
- writte under three question marks ??? after a slide
- presentation modes also shows time that has passed since the presentation started

Keyboard shortcuts

- press key h (help) or ? while presentation is on to see all possible keyboard shortcuts

Customization

Xaringan Themer

adds customization to the slides

➞ you do not need to be an expert in CSS

```
# install.packages("devtools")
devtools::install_github("gadenbuie/xaringanthemer")
```

- works with *tab completion*

````
35
36 ```{r xaringan-themer, include=FALSE}
37 library(xaringanthemer)
38
39 ```
40
41 # Hello World
42
43 Install the **xaringanthemer** package from [Github](https://github.com/gadenbuie/xaringanthemer):
44
45 ```{r eval=FALSE, tidy=FALSE}
46 devtools::install_github("gadenbuie/xaringanthemer")
47 ```
48
````

Interactive presentation of data and results

Tables | Datatable

R library DT as interface to powerful JavaScript library DataTables

→ advantages for slides: e.g. **pagination**, **filtering**, **search function** ...

```
datatable = DT::datatable(data, filter = "top", options = list(pageLength = 5, lengthMenu = c(
    10)), colnames = c("Municipality", "Election Type", "Year", "AfD Share", "Turnout"))
datatable
```

Show 5 entries Search:

	Municipality	Election Type	Year	AfD Share	Turnout
	All	All	All	All	All
1	Brandenburg an der Havel	Federal	2013	4.43	60.72
2	Cottbus	Federal	2013	6.84	65.75
3	Frankfurt (Oder)	Federal	2013	6.25	64.88
4	Potsdam	Federal	2013	5.56	57.37
5	Ahrensfelde	Federal	2013	7.25	79.37

Showing 1 to 5 of 5,016 entries Previous 1 2 3 4 5 … 1004 Next

Tables | Datatable

Datatable: Styling table cells

```
datatable %>%
  formatStyle(
    'turnout',
    backgroundColor = styleInterval(50, c('red', 'lightgrey'))
)
```

Show 5 ▼ entries Search:

	Municipality	Election Type	Year	AfD Share	Turnout
	All	All	All	All	All
1	Brandenburg an der Havel	Federal	2013	4.43	60.72
2	Cottbus	Federal	2013	6.84	65.75
3	Frankfurt (Oder)	Federal	2013	6.25	64.88
4	Potsdam	Federal	2013	5.56	57.37
5	Ahrensfelde	Federal	2013	7.25	79.37

Showing 1 to 5 of 5,016 entries Previous 1 2 3 4 5 … 1004 Next

Tables | Datatable

Further customization and options

- extensive documentation online
- fonts can be adjusted in the `css`

```
/* DT table font size */
.dataTables_wrapper{
  font-size: 0.7em;
}
```

Tables | Kable

```
library(kableExtra)

kable.tab = knitr::kable(data,
  col.names = c('Municipality', 'Election Type', 'Year', 'AfD Share', 'Turnout'),
) %>%
  kable_styling() %>%
  scroll_box(width = "500px", height = "300px")
kable.tab
```

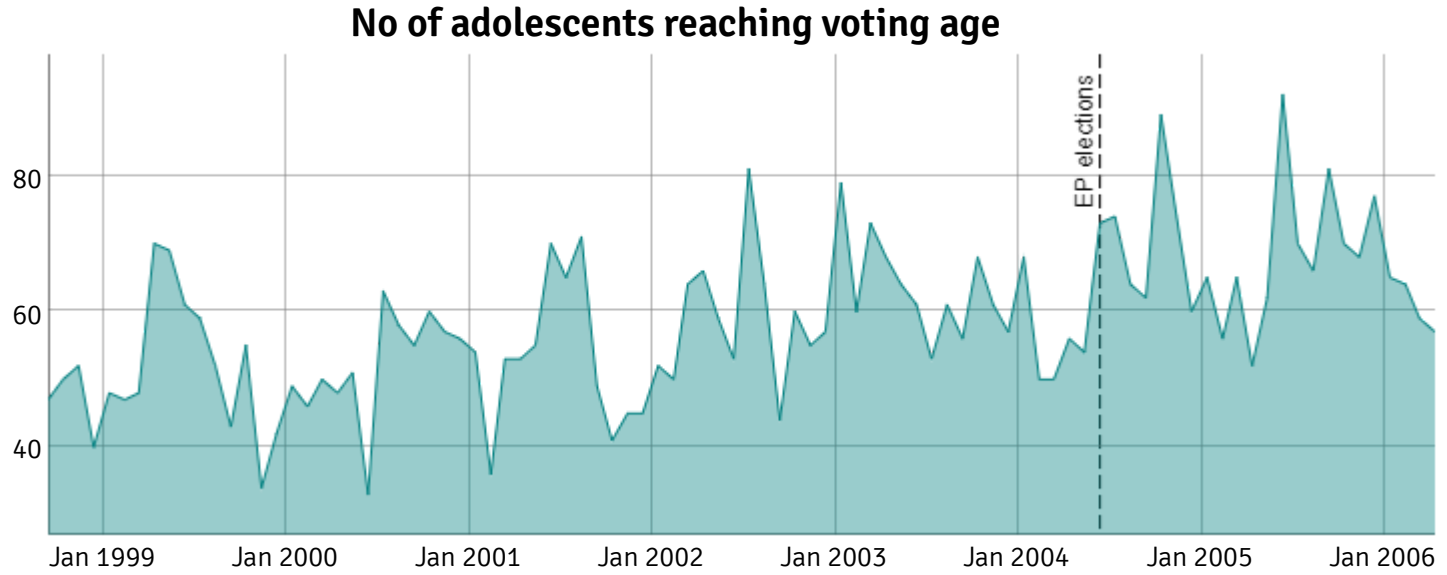
Municipality	Election Type	Year	AfD Share	Turnout
Brandenburg an der Havel	Federal	2013	4.43	60.72
Cottbus	Federal	2013	6.84	65.75
Frankfurt (Oder)	Federal	2013	6.25	64.88
Potsdam	Federal	2013	5.56	57.37
Ahrensfelde	Federal	2013	7.25	79.37
Bernau bei Berlin	Federal	2013	7.02	68.88
Eberswalde	Federal	2013	5.14	58.32
Panketal	Federal	2013	6.78	77.74

Timelines | Dygraphs

dygraphs: open source JavaScript charting library

```
pacman::p_load(dygraphs)

dygraph(birthdates.data,
        main = "No of adolescents reaching voting age") %>%
  dyEvent("2004-06-15", "EP elections", labelLoc = "top") %>%
  dyOptions(fillGraph = TRUE, fillAlpha = 0.4)
```

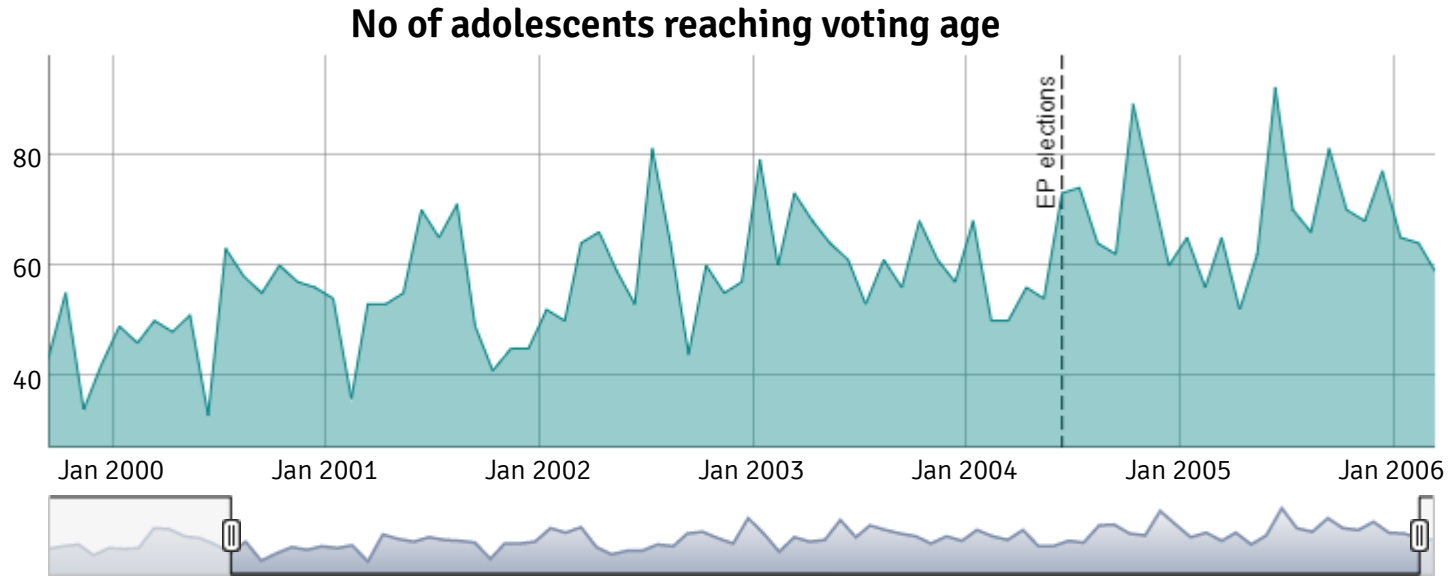


Timelines | Dygraphs

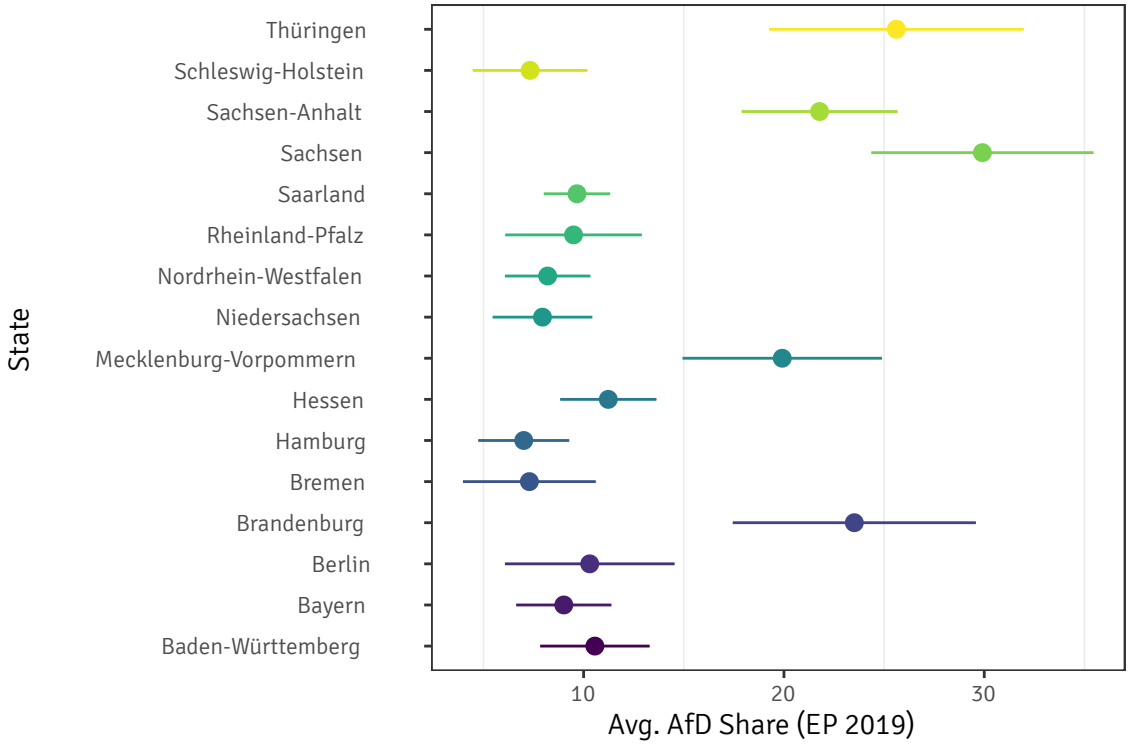
Range selector with dygraphs

➞ specify range and pre-chosen window

```
dygraph(birthdates.data,
        main = "No of adolescents reaching voting age") %>%
  dyRangeSelector(dateWindow = c("1999-09-15", "2006-03-15")) %>%
  dyEvent("2004-06-15", "EP elections", labelLoc = "top") %>%
  dyOptions(fillGraph = TRUE, fillAlpha = 0.4)
```



Animating ggplot2 | ggiraph




```
pacman::p_load("ggiraph")
afd_ggplot = afd_ggplot +
  geom_point_interactive(aes(tooltip = paste0("<b>AfD Share: </b>",
                                              round(mean_afd, digits=2))),
                         size=1)
girafe(afd_girafe = afd_ggplot)
```

Animating ggplot2 | ggiraph

ggiraph: htmlwidget and ggplot2 extension

⊕ easily add interactivity to existing ggplot objects (e.g. from a manuscript written with RMarkdown using bookdown)

⊖ interactivity options (thus far) limited

Interactive layers

- add layers to ggplot
- options based ggplot
  - `geom_point_interactive()`,
  - `geom_col_interactive()`
  - `geom_tile_interactive()`
  - `scale_fill_manual_interactive()`
  - `scale_discrete_manual_interactive()`
  - `guide_legend_interactive()`

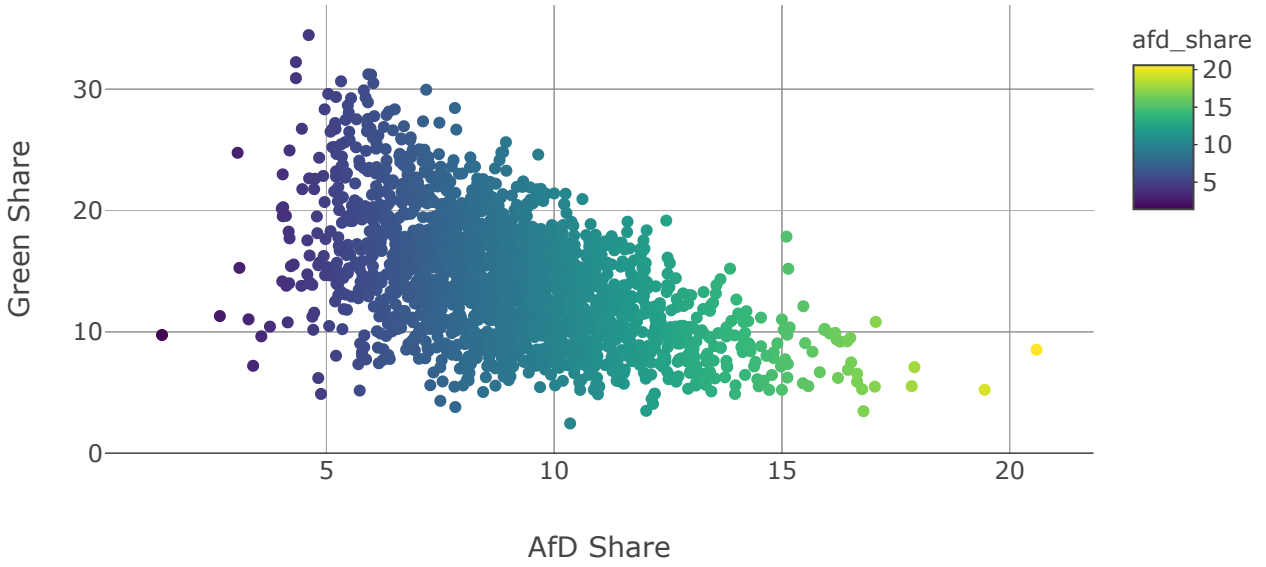
On-click option

- `onclick`: JavaScript function to be executed when elements are clicked

→ Full documentation online

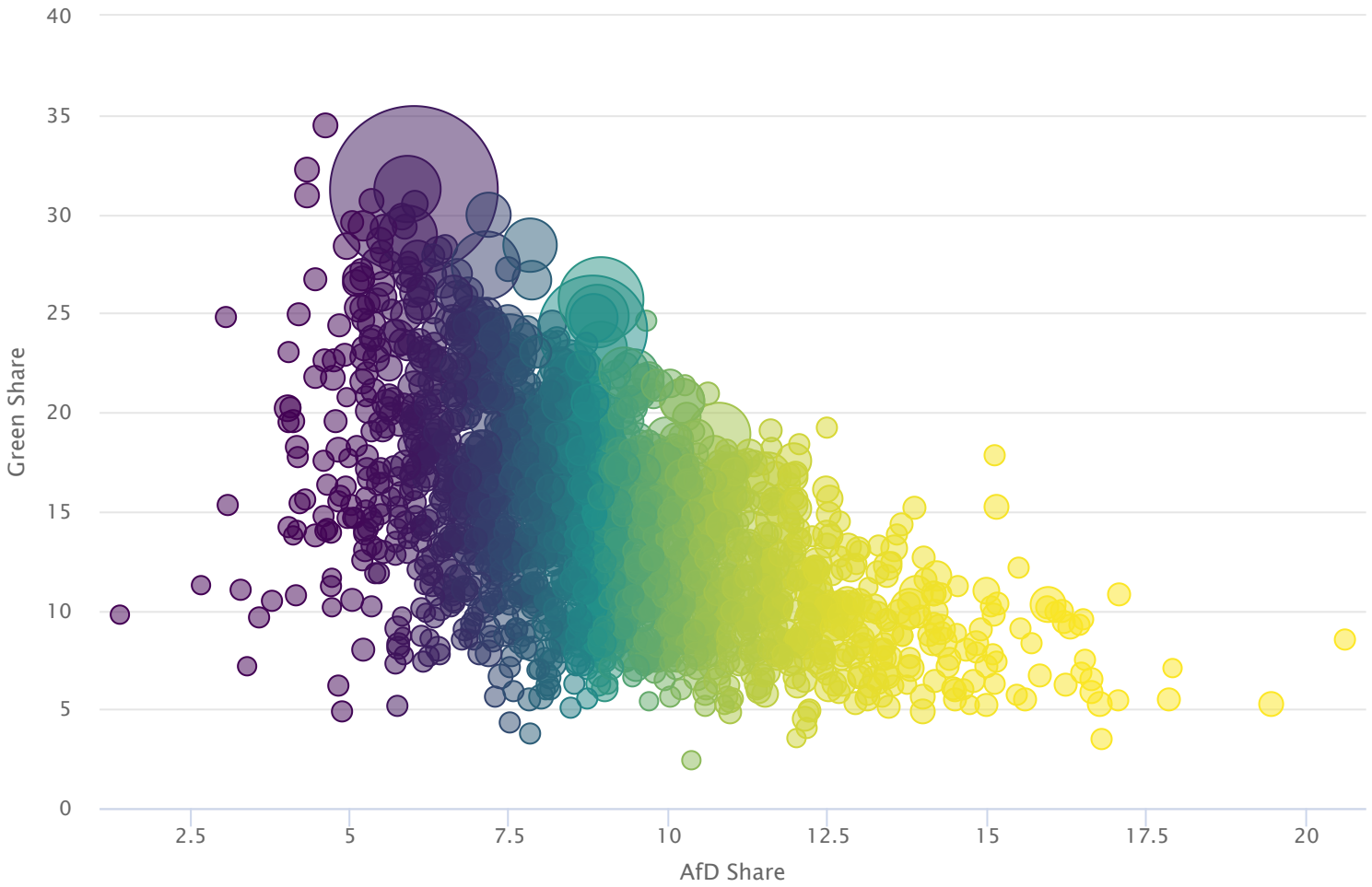
Plotly for R | Plotly

```
library(plotly)
plotly = plot_ly(data = ep19 %>% filter(region == "Bayern"), x = ~afd_share, y = ~green_share,
    color = ~afd_share, text = ~paste(municipality, "<br>AfD Share: ", afd_share,
        "<br>Green Share:", green_share)) %>% layout(xaxis = list(title = "AfD Share"),
    yaxis = list(title = "Green Share"), autosize = F, width = 750, height = 400)
plotly
```



Highcharter | Tooltips

R package `highcharter` as wrapper around Highcharts JS, JavaScript charting library



Highcharter | Tooltips

⚠ professional tool, check license options!

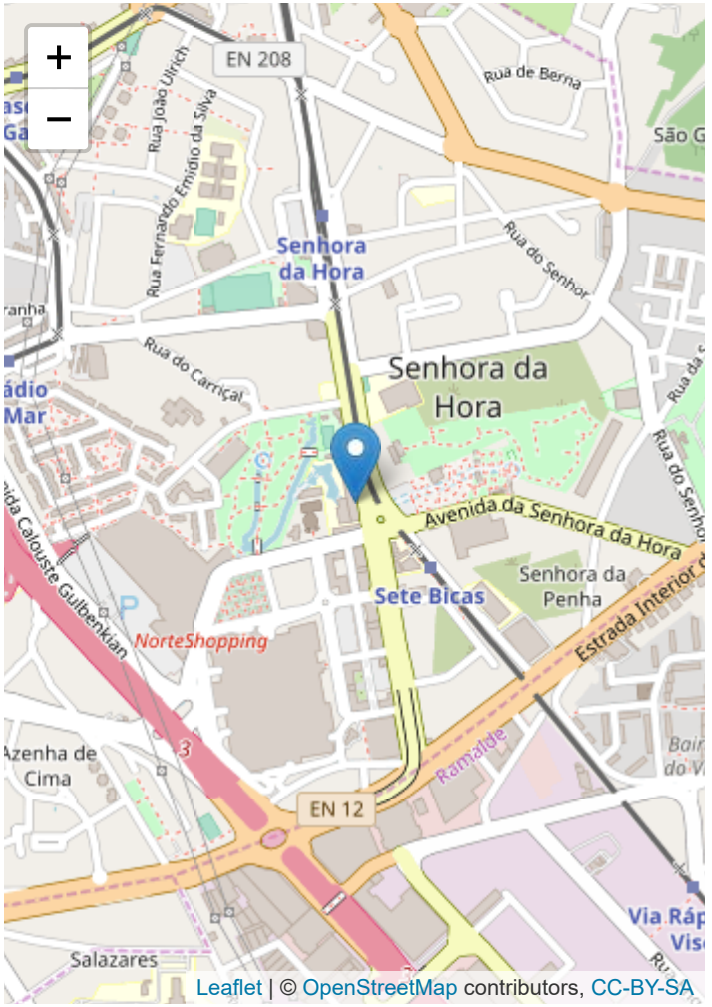
```
afd_plot_bavaria_nested = data_bavaria_long %>%
  select(municipality, year, afd_share, ags) %>%
  nest(-ags) %>%
  mutate(
    data = map(data, mutate_mapping, hcaes(x = year, y = afd_share), drop = TRUE),
    data = map(data, list_parse)
    ) %>%
  rename(ttdata = data)
# ags is the unique municipality identifier used in Germany
afd_plot_bavaria_join = left_join(afd_plot_bavaria, afd_plot_bavaria_nested, by = "ags")

# highcharter plot
afdhighchart = hchart(afd_plot_bavaria_join, "point",
  hcaes(x = afd_share,
        y = green_share,
        size = eligible,
        color = afd_share,
        name = municipality)
  ) %>%
  hc_yAxis(title = list(text = "Green Share")) %>% hc_xAxis(title = list(text = "AfD Share"))
  hc_tooltip(
    useHTML = TRUE,
    headerFormat = "<b>{point.key}</b>",
    pointFormatter = tooltip_chart(accesor = "ttdata")
    )


htmlwidgets::saveWidget(afdhighchart, "afdhighchart.html")
```

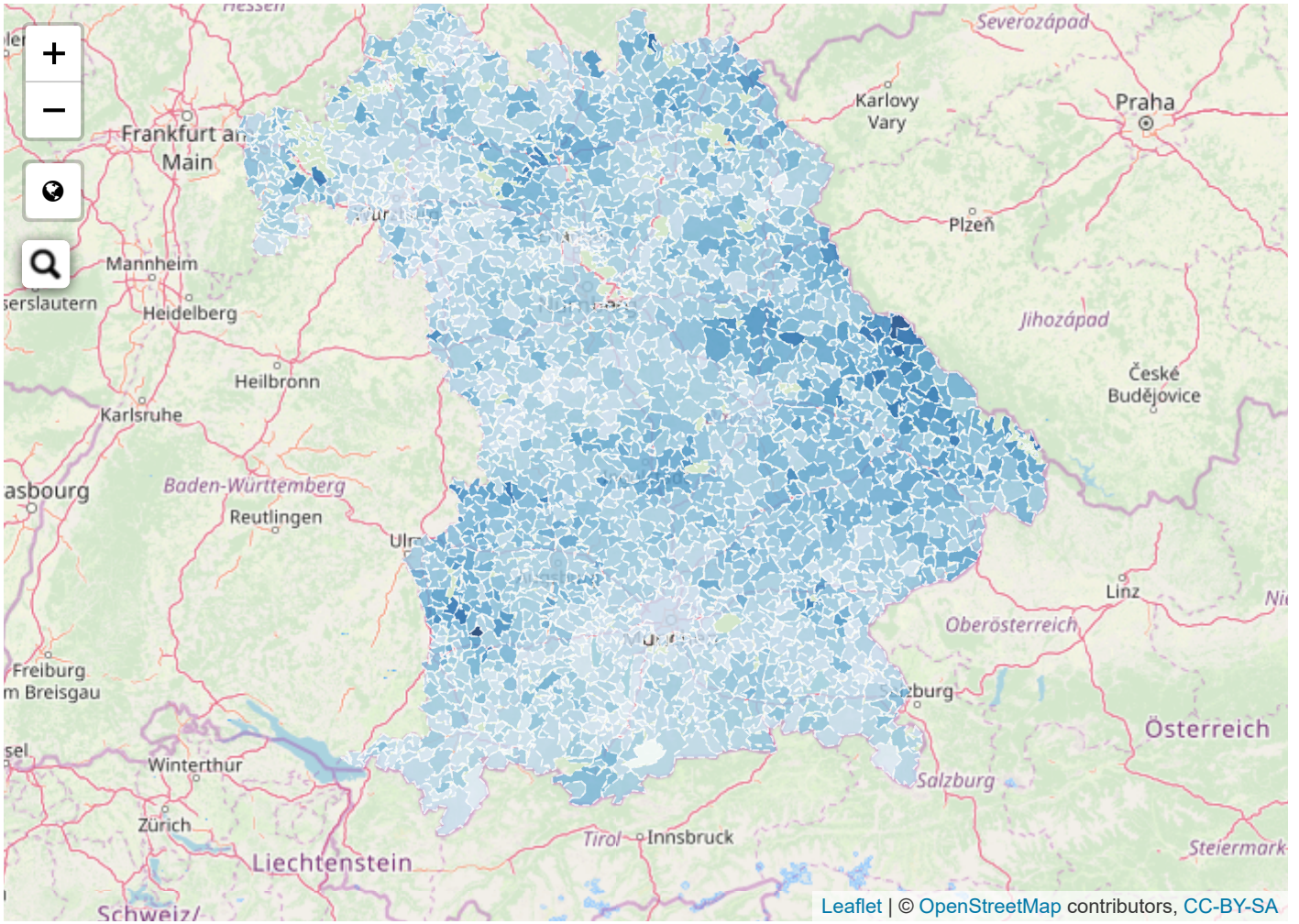
Interactive maps | leaflet

R package `leaflet` provides R bindings for javascript mapping library leaflet.js




```
leaflet(height=500,
        width=350) %>%
  addTiles() %>%
  addMarkers(lat=41.18343,
             lng=-8.65379,
             popup="Porto Business School")
  setView(lat=41.18343,
          lng=-8.65379, zoom=15)
```

Interactive spatial polygons | Leaflet



Online vs. offline - dynamic vs. static

Rendering slides offline and static slides

- **xaringan** uses the latest version of remark.js by default (online connection neccessary)
- to render slides offline have a local copy of `remark-latest.min.js` in your folder

→ `xaringan::summon_remark()` to download & store most recent version in local directory

- specify the YAML header to

```
output:
  xaringan::moon_reader:
    chakra: libs/remark-latest.min.js
```

Using Google fonts online and offline

- google fonts work whenever device has online access
- to use google fonts either modify directly the css

```
@import url("https://fonts.googleapis.com/css?family=Alegreya|Alegreya+Sans|Palanquin")
body, p {
  font-family: "Palanquin", sans-serif;
}

h1, h2, h3, h4, h5, h6 {
  font-family: "Alegreya Sans", serif;
}
```

- alternatively, rely on `xaringanthemer`

```
xaringanthemer::duo_accent(
  header_font_google = google_font("Signika", "600"),
  text_font_google   = google_font("Signika", "300", "300i")
  )
```

- **full integration of plots and graphs: `showtext` R package**

```
library(showtext)
font_add_google("Amiri", "amiri")
font_add_google("Signika", "signika")
```

- without online access: download the fonts and install them locally on your machine, e.g. via google-webfonts-helper

PDF Export

- via Decktape.js: PDF exporter for HTML presentation frameworks
- via Google Chrome's or Chromium printing function
- currently printing of HTML presentation frameworks not supported within Mozilla or Edge
- manual PDF printing or from within RMarkdown through the pagedown package 

```
install.packages(c("pagedown", "xaringan"))
# make sure you have pagedown >= 0.2 and xaringan >= 0.9; if not, run
# remotes::install_github(c('rstudio/pagedown', 'yihui/xaringan'))

pagedown::chrome_print("path/to/your/slides.Rmd")
# or just pass the HTML output file path to chrome_print()
pagedown::chrome_print("path/to/your/slides.html")
```

Thank you for your attention.

R version 3.6.1 (2019-07-05)

Platform: x86_64-w64-mingw32/x64 (64-bit)

OS: Windows 10 x64 (build 17763)

Built on : 17-Dez-2019 at 10:40:51

2019 • Julia Schulte-Cloos • jschultecloos