



BUILDING A MODULARIZED SHINY APP


WITH THE GOLEM 
AND HTML WIDGETS


DR. CÉDRIC SCHERER

Freelancing Data Visualization Designer
Leibniz Institute for Zoo & Wildlife Research

 cedricscherer.netlify.com

 cedricphilippscherer@gmail.com

 @CedScherer

 @Z3tt

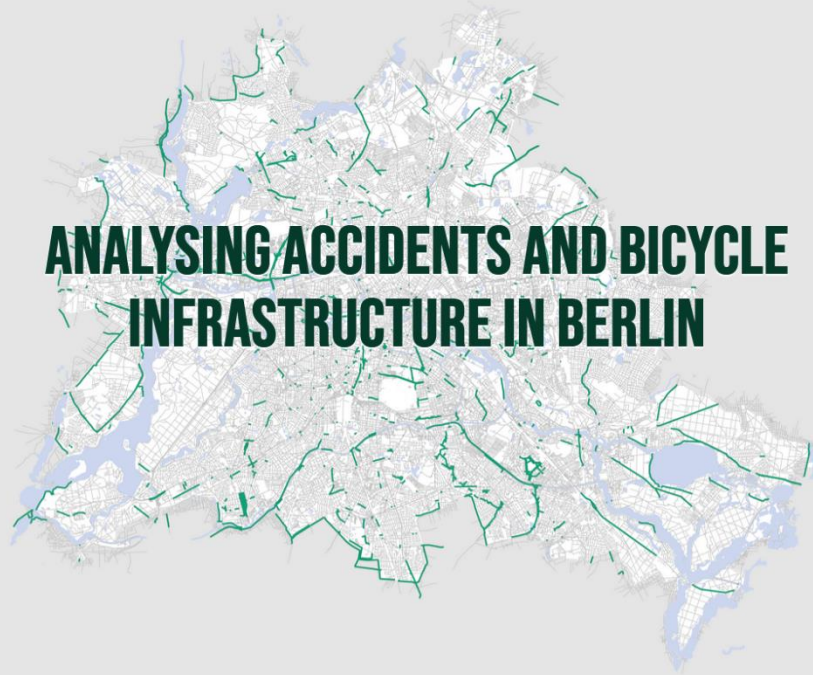
Artwork by Benjamin Lacombe



CORRELAID X CHALLENGE 2020



MOVING THROUGH BERLIN BY BIKE



ANALYSING ACCIDENTS AND BICYCLE INFRASTRUCTURE IN BERLIN



A CONTRIBUTION TO THE CORRELAID X CHALLENGE 2020



cedricscherer.netlify.com



cedricphilippscherer@gmail.com



[@CedScherer](https://twitter.com/CedScherer)



[@Z3tt](https://github.com/Z3tt)

CORRELAID X CHALLENGE 2020



MOVING THROUGH BERLIN BY BIKE

BERLINBIKES.CORRELAID.ORG



cedricscherer.netlify.com



cedricphilipscherer@gmail.com



@CedScherer



@Z3tt

CORRELAID X CHALLENGE 2020



ABOUT THE PROJECT

A Shiny app created with 🍷 and ❤️ for the CorrelAidX Challenge 2020
“Analysing and visualising German regional statistics with datenguidepy”
by Cédric Scherer, Andreas Neumann, Saleh Hamed & Steffen Reinhold



CorrelAidX Berlin, a local chapter of CorrelAid
Good Causes. Better Effects. Local Implementation.

CorrelAid is a network of young data analysts who want to change the world with a more inclusive, integrated and innovative approach to data analysis. The organization builds on three pillars: We take a pioneer role in analytics consulting for nonprofit organisations. We connect young, driven data scientists and offer them the possibility to apply and develop their skills on real-world problems. Last but not least, we start a dialogue on the potential of data and analytics for the civic society.

You want more?

[Learn more about CorrelAid](#)

[Read more about our methodology](#)

[Get the code from GitHub](#)



cedricscherer.netlify.com



cedricphilippscherer@gmail.com

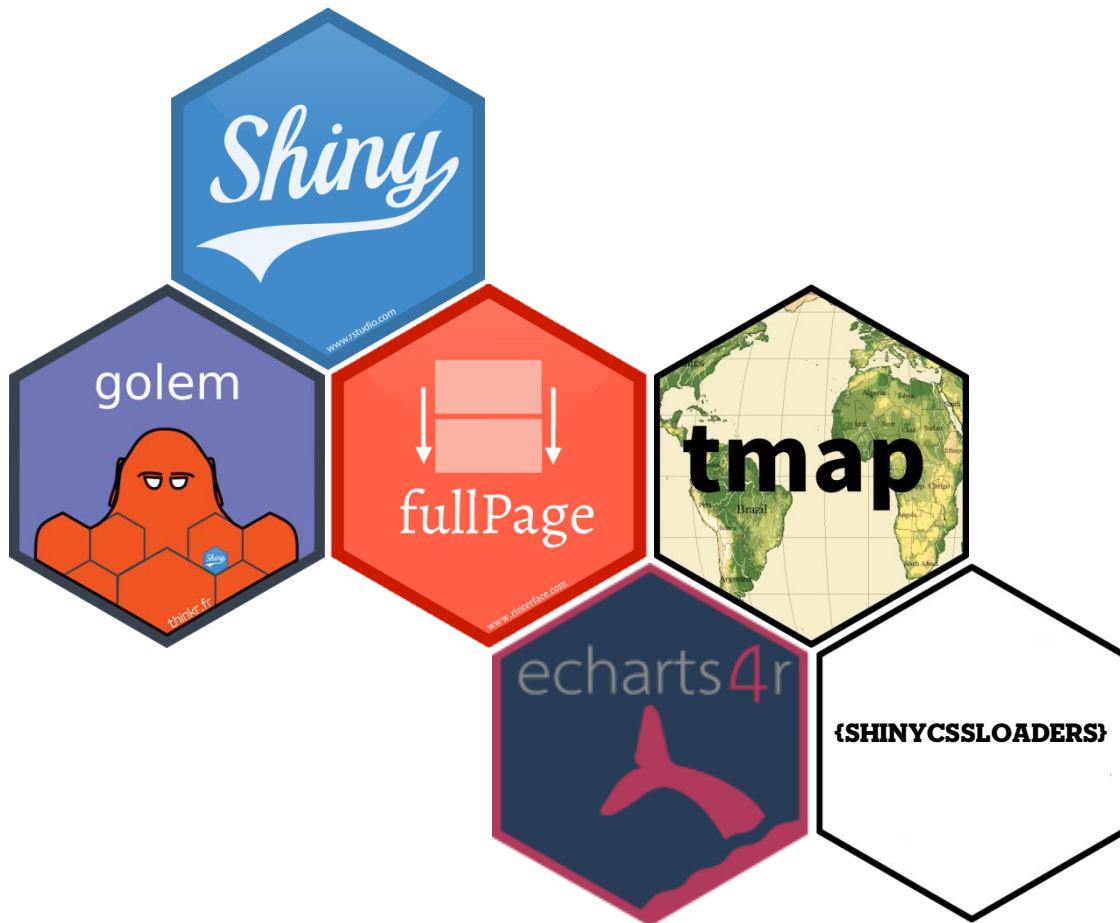


@CedScherer

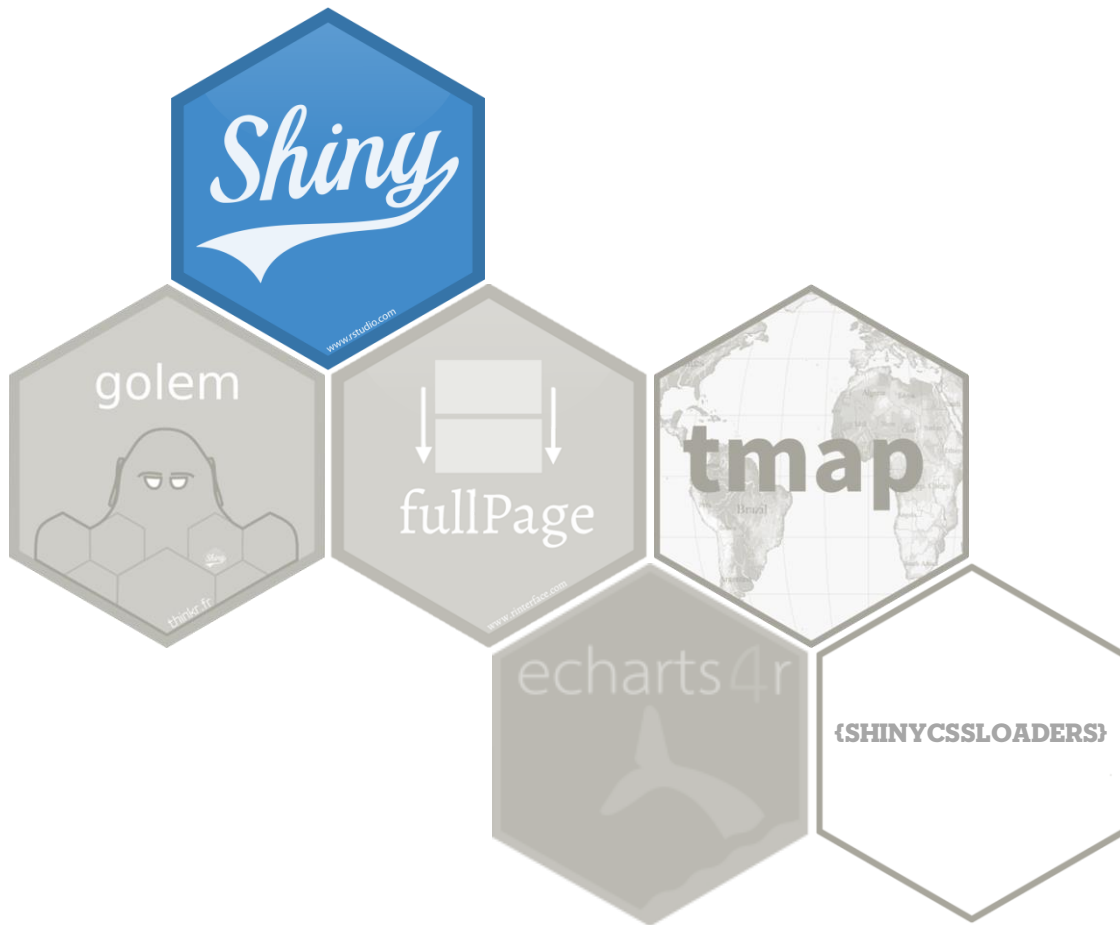


@Z3tt

CORRELAID X CHALLENGE 2020

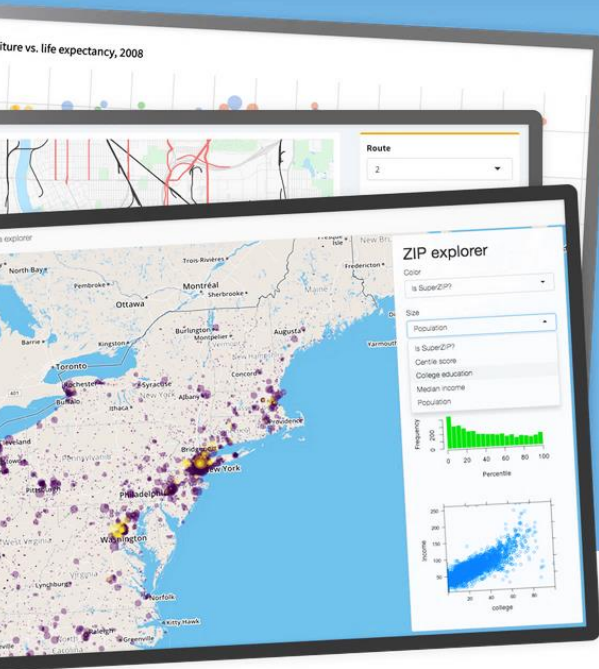


CORRELAID X CHALLENGE 2020





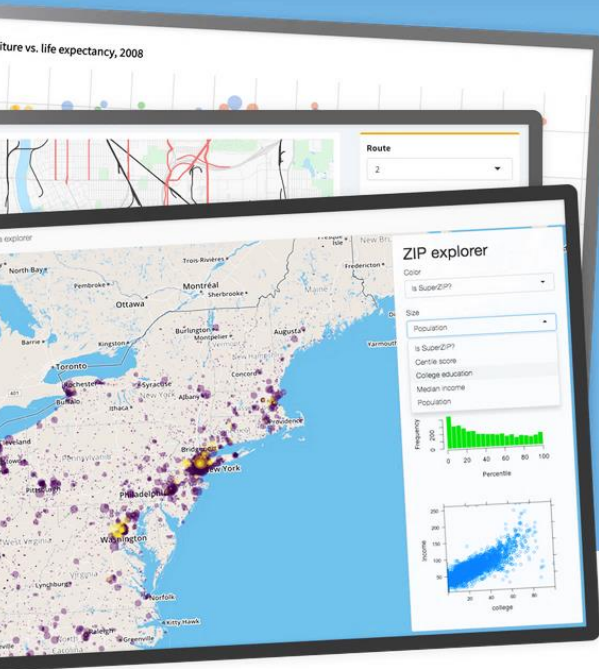
Shiny is an R package that makes it easy to build interactive web apps straight from R.





Shiny is an R package that makes it easy to build interactive web apps straight from R.

You can host standalone apps on a webpage, build dashboards or embed them in R Markdown documents.

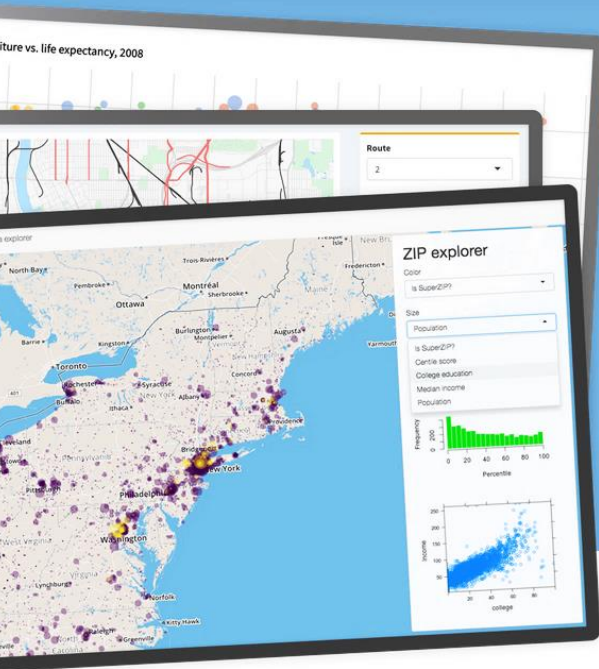




Shiny is an R package that makes it easy to build interactive web apps straight from R.


You can host standalone apps on a webpage, build dashboards or embed them in R Markdown documents.

You can extend your Shiny apps with CSS themes, htmlwidgets, and JavaScript actions.



MASTERING SHINY

Hadley Wickham

 mastering-shiny.org

O'REILLY®

Mastering Shiny

Build Interactive Apps, Reports & Dashboards
Powered by R



Hadley Wickham



cedricscherer.netlify.com



cedricphilipscherer@gmail.com



[@CedScherer](https://twitter.com/CedScherer)



[@Z3tt](https://github.com/Z3tt)

Create a new directory and in this directory a file called `app.R` with the following content:

```
library(shiny)
ui <- fluidPage(
  "Hello, world!"
)
server <- function(input, output, session) {
}
shinyApp(ui, server)
```

The screenshot shows the RStudio IDE interface. The top menu bar includes File, Edit, Code, View, Plot, Session, Build, Debug, Profile, Tools, and Help. The top toolbar contains icons for file operations and a search bar. The main window is divided into four panes:

- Console:** Displays the R version 4.0.2 (2020-06-22) -- "Taking Off Again" and copyright information. It also shows the R license text in German and English, and instructions for using the help system. The prompt is > |.
- Environment:** Shows the Global Environment, which is currently empty.
- Files:** Shows the file explorer with a search bar.
- Viewer:** Displays the R Documentation for the runApp function, titled "Run Shiny Application".

The documentation in the Viewer pane includes the following sections:

- Description:** Runs a Shiny application. This function normally does not return; interrupt R to stop the application (usually by pressing Ctrl+C or Esc).
- Usage:** runApp(
 appDir = getwd(),
 port = getOption("shiny.port"),
 launch.browser = getOption("shiny.launch.browser", interactive()),
 host = getOption("shiny.host", "127.0.0.1"),
 workerId = "",
 quiet = FALSE,
 display.mode = c("auto", "normal", "showcase"),
 test.shiny = getOption("shiny.test.shiny", FALSE)

Modularizing Shiny app code

Last Updated: 17 Apr 2020

By: Winston Chang

As Shiny applications grow larger and more complicated, we use modules to manage the growing complexity of Shiny application code.

Modularizing Shiny app code

Last Updated: 17 Apr 2020

By: Winston Chang

As Shiny applications grow larger and more complicated, **we use modules to manage the growing complexity** of Shiny application code.

MOD.R

```
create_plot <- function(input, output, session, data) { ... }
```

```
create_plot_ui <- function(id) { ... }
```

SERVER.R

```
callModule(create_plot, id = 'plotA', data = dataA)  
callModule(create_plot, id = 'plotB', data = dataB)  
callModule(create_plot, id = 'plotC', data = dataC)
```

UI.R

```
create_plot_ui(id = 'plotA')  
create_plot_ui(id = 'plotB')  
create_plot_ui(id = 'plotC')
```

MOD.R

```
create_plot <- function(input, output, session, data) { ... }  
create_plot_ui <- function(id) { ... }
```



SERVER.R

```
callModule(create_plot, id = 'plotA', data = dataA)  
callModule(create_plot, id = 'plotB', data = dataB)  
callModule(create_plot, id = 'plotC', data = dataC)
```

UI.R

```
create_plot_ui(id = 'plotA')  
create_plot_ui(id = 'plotB')  
create_plot_ui(id = 'plotC')
```

MOD.R

```
create_plot <- function(input, output, session, data) { ... }  
  
create_plot_ui <- function(id) { ... }
```



SERVER.R

```
callModule(create_plot, id = 'plotA', data = dataA)  
callModule(create_plot, id = 'plotB', data = dataB)  
callModule(create_plot, id = 'plotC', data = dataC)
```

UI.R

```
create_plot_ui(id = 'plotA')  
create_plot_ui(id = 'plotB')  
create_plot_ui(id = 'plotC')
```

MOD.R

```
create_plot <- function(input, output, session, data) { ... }  
  
create_plot_ui <- function(id) { ... }
```



SERVER.R

```
callModule(create_plot, id = 'plot', data = data)  
callModule(create_map, id = 'map', data = data)
```

UI.R

```
create_plot_ui(id = 'plot')  
create_map_ui(id = 'map')
```

MOD.R

```
create_plot <- function(input, output, session, data) { ... }  
create_plot_ui <- function(id) { ... }  
  
create_map <- function(input, output, session, data) { ... }  
create_map_ui <- function(id) { ... }
```



SERVER.R

```
callModule(create_plot, id = 'plot', data = data)  
callModule(create_map, id = 'map', data = data)
```

UI.R

```
create_plot_ui(id = 'plot')  
create_map_ui(id = 'map')
```

MOD_PLOT.R

```
create_plot <- function(input, output, session, data) { ... }  
create_plot_ui <- function(id) { ... }
```

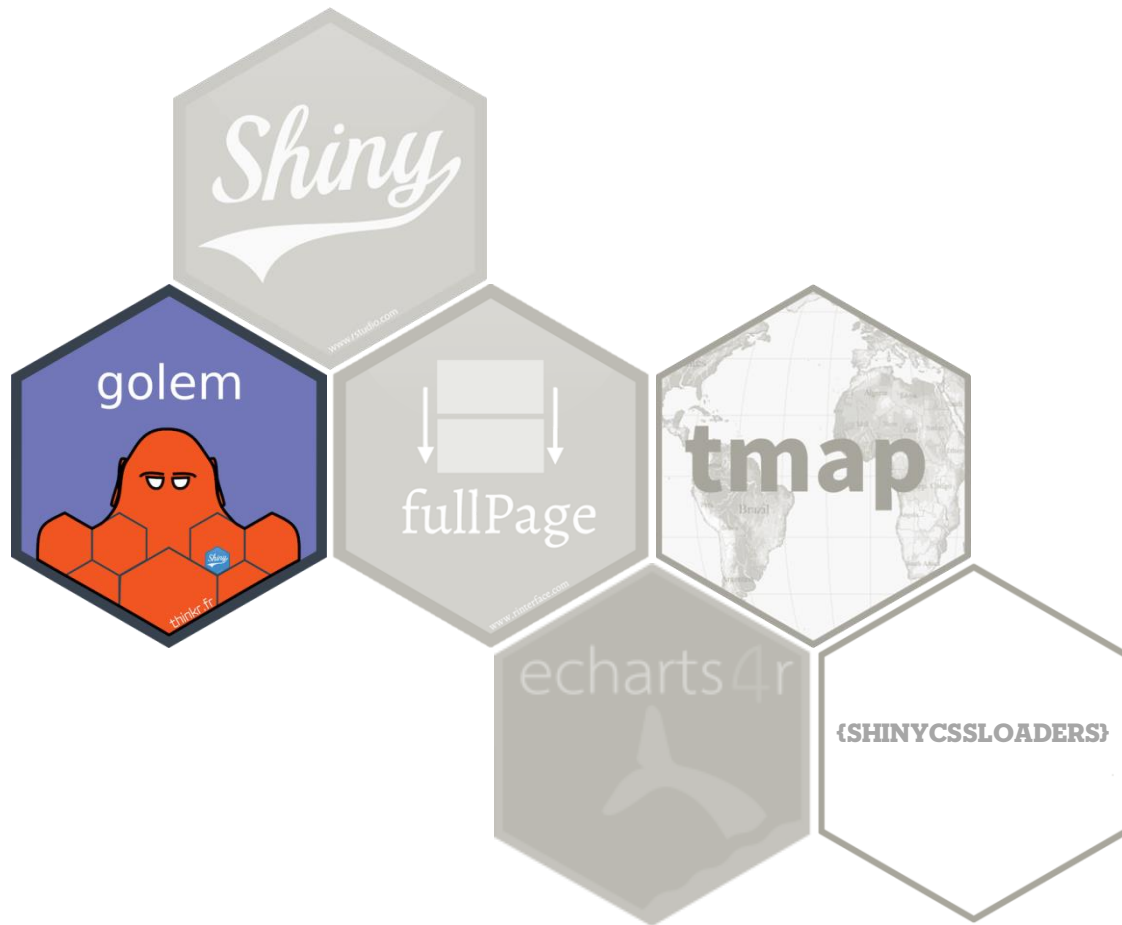
MOD_MAP.R

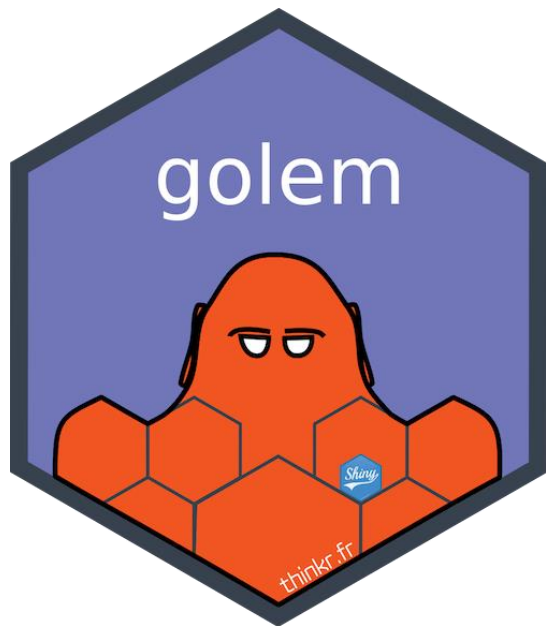
```
create_map <- function(input, output, session, data) { ... }  
create_map_ui <- function(id) { ... }
```



MOD_PLOT.R

```
create_plot_server <- function(input, output, session, data) {  
  ns <- session$ns  
  ...  
}  
  
create_plot_ui <- function(id) {  
  ns <- NS(id)  
  ...  
}
```




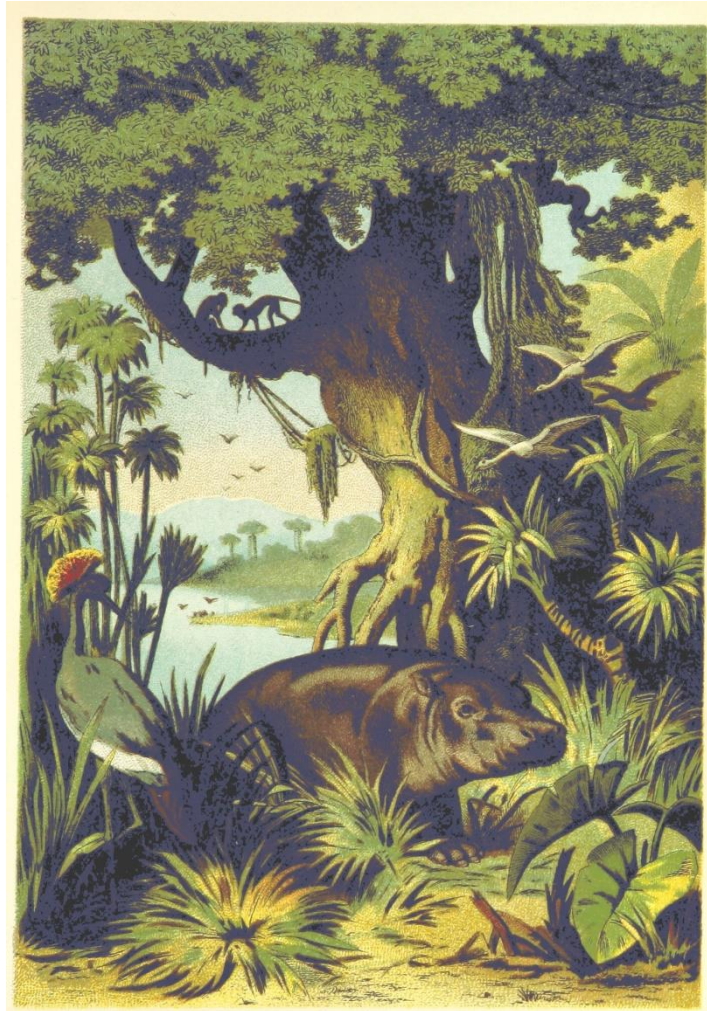


The `{golem}` package is an opinionated framework for building production-grade shiny applications.

ENGINEERING PRODUCTION-GRADE SHINY APPS

Colin Fay, Sébastien Rochette,
Vincent Guyader, Cervan Girard

 engineering-shiny.org



#CORRELCON2020



cedricscherer.netlify.com



cedricphilipscherer@gmail.com



[@CedScherer](https://twitter.com/CedScherer)



[@Z3tt](https://github.com/Z3tt)

If you haven't already installed {golem}:

```
install.packages("golem")  
remotes::install_github("Thinkr-open/golem")
```

New Project

Back

Project Type

R Package using RcppEigen



R Package using RcppParallel



Book Project using bookdown



R Package using devtools



Package for Shiny App using golem



New Plumber API Project



Simple R Markdown Website



Create a new
Package for Shiny
App using golem

Cancel



Building a Prod-Ready, Robust Shiny Application.

#

README: each step of the dev files is optional, and you don't have to
fill every dev scripts before getting started.

01_start.R should be filled at start.

02_dev.R should be used to keep track of your development during the project.

03_deploy.R should be used once you need to deploy your app.



01_START.R

```
#####
#### CURRENT FILE: ON START SCRIPT ####
#####
```

Fill the DESCRIPTION ----

Add meta data about your application

```
golem::fill_desc(
  pkg_name = "golex", # The Name of the package containing the App
  pkg_title = "PKG_TITLE", # The Title of the package containing the App
  pkg_description = "PKG_DESC.", # The Description of the package containing
    the App
  author_first_name = "AUTHOR_FIRST", # Your First Name
  author_last_name = "AUTHOR_LAST", # Your Last Name
  author_email = "AUTHOR@MAIL.COM", # Your Email
  repo_url = NULL # The URL of the GitHub Repo (optional)
)
```

02_DEV.R

03_DEPLOY.R

Codes on GitHub:

Z3TT/CORRELCON2020_GOLEM_HTML_WIDGETS

01_START.R

```
#####
#### CURRENT FILE: ON START SCRIPT ####
#####
```

Fill the DESCRIPTION ----

Add meta data about your application

```
golem::fill_desc(
  pkg_name = "correlcon", # The Name of the package containing the App
  pkg_title = "A tutorial on the golem package and html widgets for nice looking
  Shiny apps", # The Title of the package containing the App
  pkg_description = "This Shiny app is build with the help of the {golem} framework
  and exemplary uses the html widgets {echarts4R}, {tmap}, and
  {shinycssloader}.", # The Description of the package containing the App
  author_first_name = "Cédric", # Your First Name
  author_last_name = "Scherer", # Your Last Name
  author_email = "cedricphilippscherer@gmail.com", # Your Email
  repo_url = NULL # The URL of the GitHub Repo (optional)
)
```

02_DEV.R

03_DEPLOY.R



01_START.R

Set {golem} options ----

```
golem::set_golem_options()
```

Create Common Files ----

See ?usethis for more information

```
usethis::use_mit_license(name = "Cédric Scherer") # You can set another license here
```

```
usethis::use_readme_rmd(open = FALSE)
```

```
usethis::use_code_of_conduct()
```

```
usethis::use_lifecycle_badge("Experimental")
```

```
usethis::use_news_md(open = FALSE)
```

Use git ----

```
usethis::use_git()
```

Init Testing Infrastructure ----

Create a template for tests

```
golem::use_recommended_tests()
```

02_DEV.R

03_DEPLOY.R



01_START.R

Use Recommended Packages ----

```
golem::use_recommended_deps()
```

Favicon ----

```
# If you want to change the favicon (default is golem's one)
```

```
golem::remove_favicon()
```

```
golem::use_favicon("https://raw.githubusercontent.com/CorrelAid/xberlin/  
master/inst/app/www/favicon.ico") # path = "path/to/ico". Can be an online file.
```

Add helper functions ----

```
golem::use_utils_ui()
```

```
golem::use_utils_server()
```

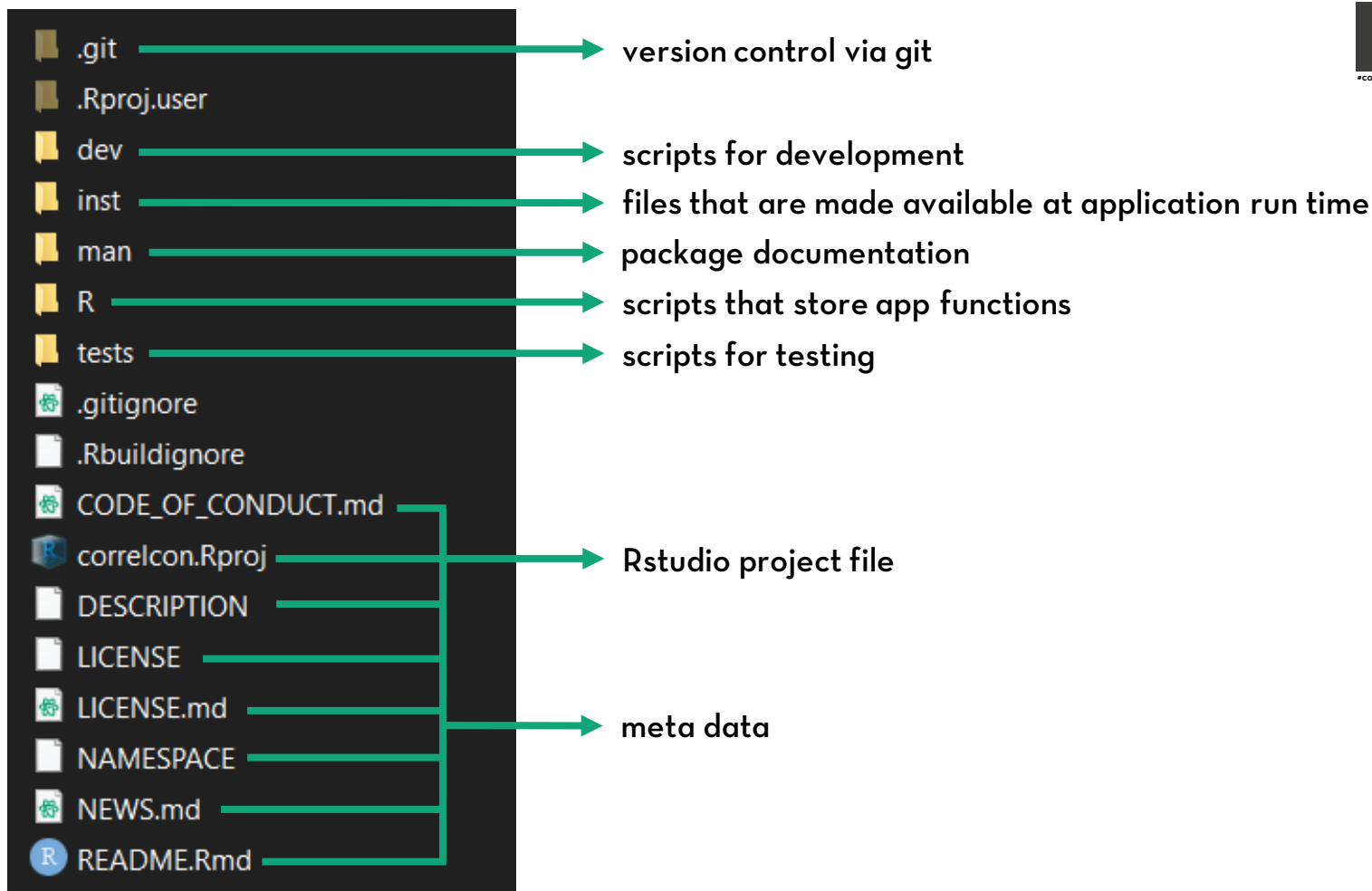
You're now set! ----

```
# go to dev/02_dev.R
```

```
rstudioapi::navigateToFile( "dev/02_dev.R" )
```

02_DEV.R

03_DEPLOY.R



```
#####
#### CURRENT FILE: DEV SCRIPT ####
#####
```

Engineering

Dependencies ----

```
## Add one line by package you want to add as dependency
usethis::use_package( "echarts4r" )
usethis::use_package( "tmap" )
```

Add modules ----

```
## Create a module infrastructure in R/
golem::add_module( name = "bars_echarts" ) # Name of the module
golem::add_module( name = "map_tmap" ) # Name of the module
```

01_START.R

02_DEV.R

03_DEPLOY.R

```
#####
#### CURRENT FILE: DEV SCRIPT ####
#####
```

Engineering

Dependencies ----

```
## Add one line by package you want to add as dependency
usethis::use_package( "echarts4r" )
usethis::use_package( "tmap" )
```

Add modules ----

```
## Create a module infrastructure in R/
golem::add_module( name = "bars_echarts" ) # Name of the module
golem::add_module( name = "map_tmap" ) # Name of the module
```



```
#' bars_echarts UI Function
#
#' @description A shiny Module.
#
#' @param id,input,output,session Internal parameters for {shiny}.
#
#' @noRd
#
#' @importFrom shiny NS tagList
mod_bars_echarts_ui <- function(id){
  ns <- NS(id)
  tagList(

  )
}
```

#' bars_echarts Server Function

```
#'
#' @noRd
mod_bars_echarts_server <- function(input, output, session){
  ns <- session$ns

}
```

To be copied in the UI

```
# mod_bars_echarts_ui("bars_echarts_ui_1")
```

To be copied in the server

```
# callModule(mod_bars_echarts_server, "bars_echarts_ui_1")
```

SERVER.R

```
callModule(mod_bars_echarts_server, "bars_echarts_ui_1")  
callModule(mod_map_tmap_server, "map_tmap_ui_1")
```

UI.R

```
mod_bars_echarts_ui("bars_echarts_ui_1")  
mod_map_tmap_ui("map_tmap_ui_1")
```

R/MOD_BARS_ECHARTS.R

```
mod_bars_echarts_ui <- function(id){  
  ns <- NS(id)  
  tagList()  
}  
mod_bars_echarts_server <- function(input, output, sess  
  ns <- session$ns  
}
```

R/MOD_MAP_TMAP.R

```
mod_map_tmap_ui <- function(id){  
  ns <- NS(id)  
  tagList()  
}  
mod_map_tmap_server <- function(input, output, session){  
  ns <- session$ns  
}
```



Add helper functions ----

```
## Creates ftc_* and utils_*  
golem::add_fct( "helpers" )  
golem::add_utils( "helpers" )
```

External resources ----

```
## Creates .js and .css files at inst/app/www  
golem::add_js_file( "script" )  
golem::add_js_handler( "handlers" )  
golem::add_css_file( "custom" )
```

Add internal datasets ----

```
## If you have data in your package  
usethis::use_data_raw( name = "my_dataset", open = FALSE )
```

Tests ----

```
## Add one line by test you want to create  
usethis::use_test( "app" )
```

01_START.R

02_DEV.R

03_DEPLOY.R

Documentation

Vignette ----

```
usethis::use_vignette("correlcon")  
devtools::build_vignettes()
```

Code coverage ----

(You'll need GitHub there)

```
usethis::use_github()  
usethis::use_travis()  
usethis::use_appveyor()
```

You're now set! ----

```
# go to dev/03_deploy.R  
rstudioapi::navigateToFile("dev/03_deploy.R")
```

01_START.R

02_DEV.R

03_DEPLOY.R

```
#####
#### CURRENT FILE: DEPLOY SCRIPT ####
#####
```

```
# Test your app
```

```
## Run checks ----
```

```
## Check the package before sending to prod
```

```
devtools::check()
```

```
rhub::check_for_cran()
```

```
# Deploy
```

```
## RStudio ----
```

```
## If you want to deploy on RStudio related platforms
```

```
golem::add_rstudioconnect_file()
```

```
golem::add_shinyappsio_file()
```

```
golem::add_shinyserver_file()
```



01_START.R

02_DEV.R

03_DEPLOY.R

Docker ----

If you want to deploy via a generic Dockerfile

```
golem::add_dockerfile()
```

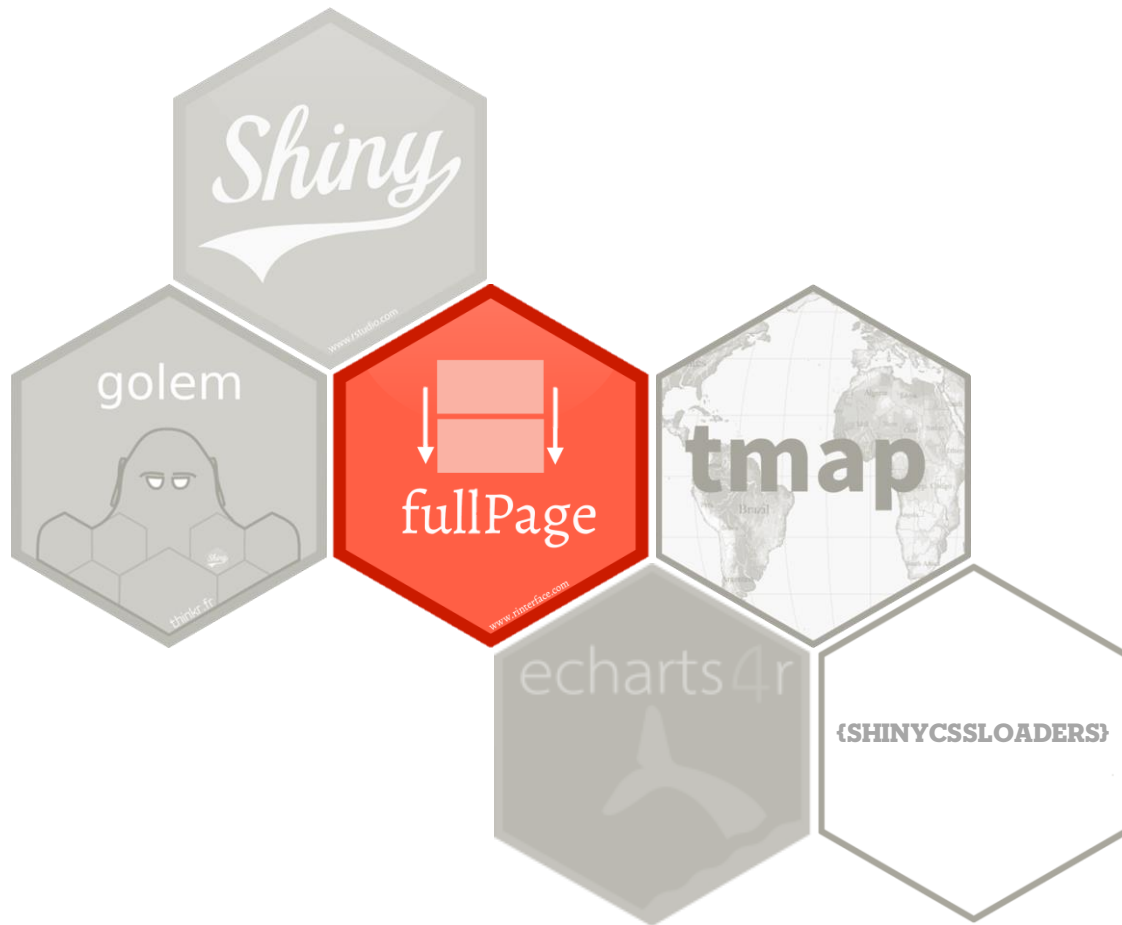
If you want to deploy to ShinyProxy

```
golem::add_dockerfile_shinyproxy()
```

If you want to deploy to Heroku

```
golem::add_dockerfile_heroku()
```





[Piling](#)[Grid](#)[Plots](#)[Images](#)[Themes](#)[Buttons](#)[sisters](#)

pagePiling.js meets Shiny!

cedricscherer.netlify.comcedricphilippscherer@gmail.com[@CedScherer](https://twitter.com/CedScherer)[@Z3tt](https://github.com/Z3tt)

#' The application server-side

```
#'
#' @param input,output,session Internal parameters for {shiny}. #' @import shiny
#' @noRd
app_server <- function( input, output, session ) {
  # List the first level callModules here
  callModule(modBarsEchartsServer, "barsEcharts_ui_1")
  callModule(modMapTmapServer, "map_tmap_ui_1")
}
```

APP_SERVER.R

```
#' The application server-side
#
#' @param input,output,session Internal parameters for {shiny}. #' @import shiny
#' @noRd
app_server <- function( input, output, session ) {
  # List the first level callModules here
  callModule(modBarsEchartsServer, "barsEcharts_ui_1")
  callModule(modMapTmapServer, "map_tmap_ui_1")
}
```

APP_UI.R

SERVER.R

```
callModule(modBarsEchartsServer, "barsEcharts_ui_1")
callModule(modMapTmapServer, "map_tmap_ui_1")
```

#' The application User-Interface

```
#' @param request Internal parameter for `{shiny}`.
```

```
#' @import shiny fullPage
```

```
#' @noRd
```

```
app_ui <- function(request) {
  tagList(
    # Leave this function for adding external resources
    golem_add_external_resources(),
    # List the first level UI elements here
    fullPage::pagePiling(
      sections.color = c('#f1f1f1', '#dddddd'),
      opts = list(easing = "linear", keyboardScrolling = TRUE),
      menu = c("interactive bar chart" = "bar",
               "interactive map" = "map"),
      fullPage::pageSection(center = TRUE,
                            menu = "bar",
                            mod_bars_echarts_ui("bars_echarts_ui_1")),
      fullPage::pageSection(center = TRUE,
                            menu = "map",
                            mod_map_tmap_ui("map_tmap_ui_1"))
    )
  )
}
```

#' The application User-Interface

```
#' @param request Internal parameter for `{shiny}`.
```

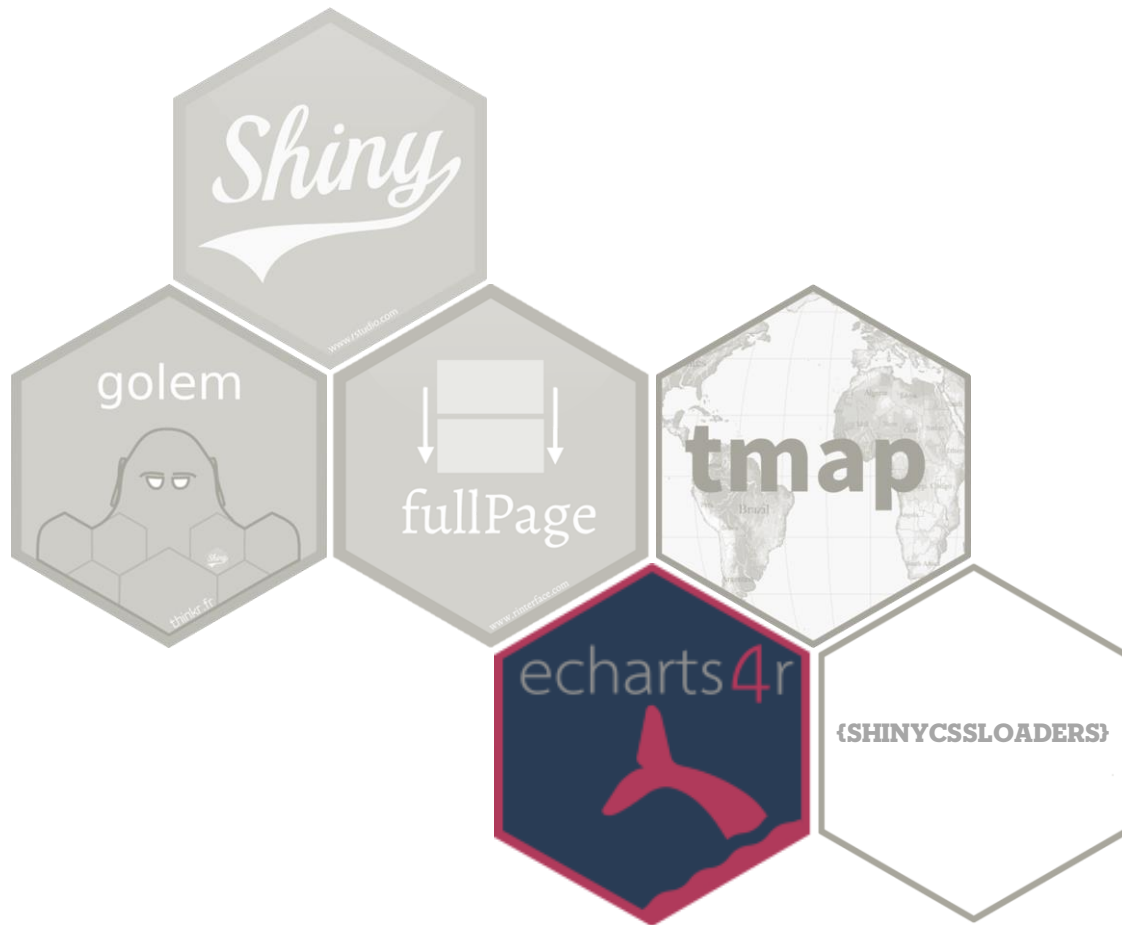
```
#' @import shiny fullPage
```

```
#' @noRd
```

```
app_ui <- function(request) {
  tagList(
    # Leave this function for adding external resources
    golem_add_external_resources(),
    # List the first level UI elements here
    fullPage::pagePiling(
      sections.color = c("#f1f1f1", "#dddddd"),
      opts = list(easing = "linear", keyboardScrolling = TRUE),
      menu = c("interactive bar chart" = "bar",
               "interactive map" = "map"),
      fullPage::pageSection(center = TRUE,
                             menu = "bar",
                             mod_bars_echarts_ui("bars_echarts_ui_1")),
      fullPage::pageSection(center = TRUE,
                             menu = "map",
                             mod_map_tmap_ui("map_tmap_ui_1"))
    )
  )
}
```

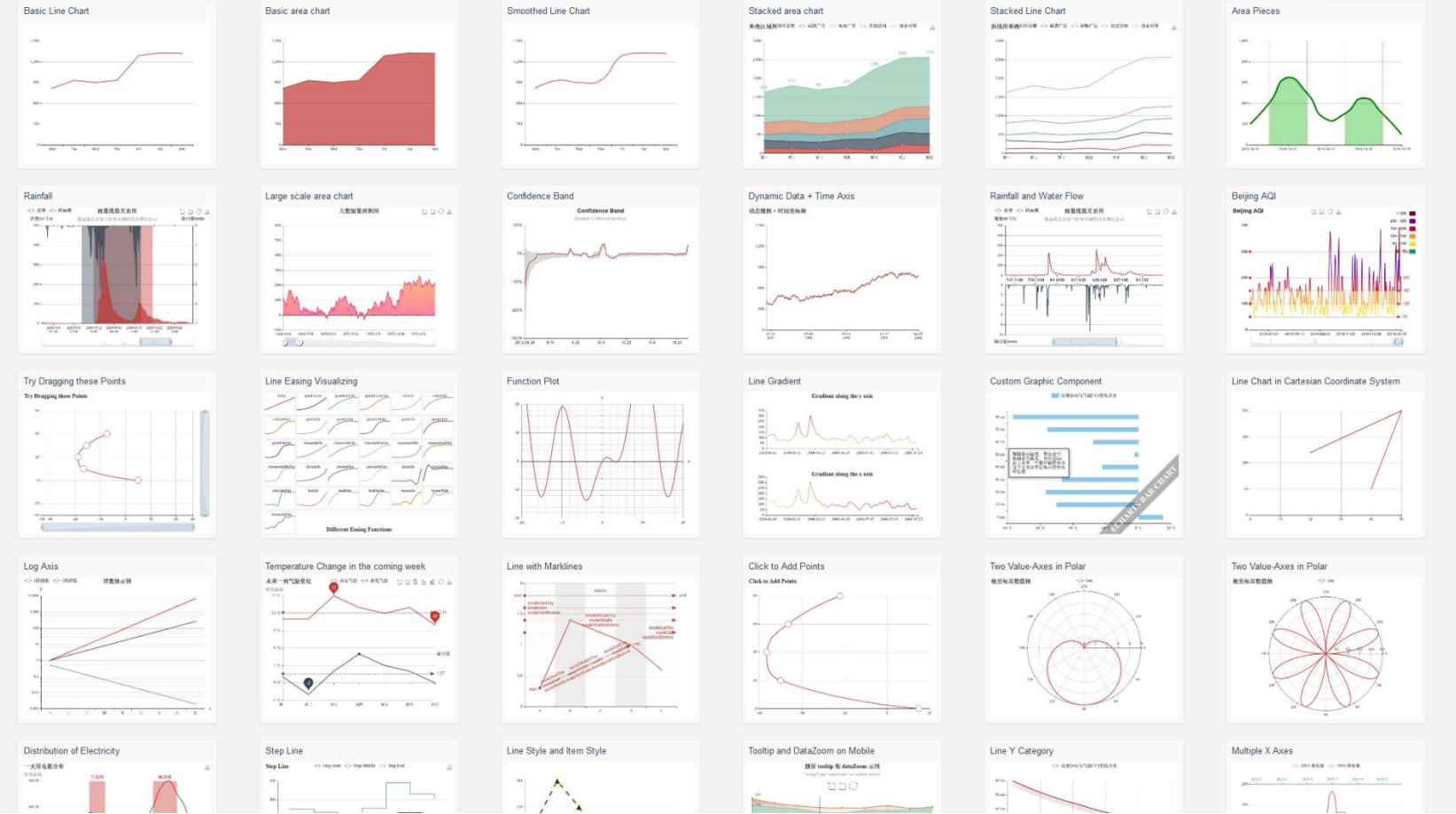
UI.R

```
mod_bars_echarts_ui("bars_echarts_ui_1")
mod_map_tmap_ui("map_tmap_ui_1")
```



- Line
- Bar
- Pie
- Scatter
- GEOMap
- Candlestick
- Radar
- Boxplot
- Heatmap
- Graph
- Lines
- Tree
- Treemap
- Sunburst
- Parallel
- Sankey
- Funnel
- Gauge
- PictorialBar
- ThemeRiver
- Calendar
- Custom
- Dataset
- DataZoom
- Drag
- Rich Text
- 3D Globe
- 3D Bar

Line



Rainfall



Large scale area chart



Confidence Band



Dynamic Data + Time Axis



Rainfall and Water Flow



Beijing AQI



Try Dragging these Points



Line Easing Visualizing



Function Plot



Line Gradient



Custom Graphic Component



Line Chart in Cartesian Coordinate System



Log Axis



Temperature Change in the coming week



Line with Marklines



Click to Add Points



Two Value-Axes in Polar



Two Value-Axes in Polar



Distribution of Electricity



Step Line



Line Style and Item Style



Tooltip and DataZoom on Mobile



Line Y Category



Multiple X Axes



echarts4r



Interactive visualisations for R via [Apache ECharts](#) (incubating)

[GET STARTED](#)[REFERENCE](#)[TIMELINE](#)[SHINY DEMO](#)

New Features

"Use (almost) any chart serie as chart proxy, now on CRAN!"

[EXPLORE](#)[CHANGELOG](#)

Installation

The package is available on [CRAN](#). The full installation can be obtained with:

```
install.packages("echarts4r")
```

However, if you only want a *lite* version you can simply do, this is useful for a lighter version that installs faster if you do not want to use any of the geospatial features of the package:

```
install.packages("echarts4r", dependencies = c("Depends", "Imports"))
```

You can also install the *unstable* development version of echarts4r with [remotes](#) from Github, see [changes](#).

```
# install.packages("remotes")
remotes::install_github("JohnCoene/echarts4r")
```

如果您位于中国，请安装

```
# install.packages("remotes")
remotes::install_git("https://gitee.com/JohnCoene/echarts4r")
```

Links

Download from CRAN at
<https://cloud.r-project.org/package=echarts4r>

Browse source code at
<https://github.com/JohnCoene/echarts4r/>

Report a bug at
<https://github.com/JohnCoene/echarts4r/issues>

Original library at
<https://echarts.apache.org/>

License

[Full license](#)

Apache License (>= 2.0)

Community

[Code of conduct](#)

Developers

[John Coene](#)

Author, maintainer, copyright holder

[All authors...](#)

Support

Sponsor the project

[💖 Sponsor](#)


```
mod_bars_echarts_server <- function(input, output, session){  
  ns <- session$ns  
  
  output$bars <- echarts4r::renderEcharts4r({  
    accidents_sum_bikes %>%  
    dplyr::group_by(type) %>%  
    echarts4r::e_charts(Gemeinde_name) %>%  
    echarts4r::e_bar(accidents) })  
}
```

```
mod_bars_echarts_server <- function(input, output, session){  
  ns <- session$ns  
  
  output$bars <- echarts4r::renderEcharts4r({  
    accidents_sum_bikes %>%  
    dplyr::group_by(type) %>%  
    echarts4r::e_charts(Gemeinde_name) %>%  
    echarts4r::e_bar(accidents)  
  })  
}
```

RENDER CHART FOR UI

```
modBarsEchartsServer <- function(input, output, session){  
  ns <- session$ns  
  
  output$bars <- echarts4r::renderEcharts4r({  
    accidents_sum_bikes %>%  
    dplyr::group_by(type) %>%  
    echarts4r::e_charts(Gemeinde_name) %>%  
    echarts4r::e_bar(accidents)  
  })  
}
```

BAR CHART

ACCIDENTS VS GEMEINDE_NAME GROUPED BY TYPE

R/MOD_BARS_ECHARTS.R

```
mod_bars_echarts_server <- function(input, output, session){
  ns <- session$ns

  output$bars <- echarts4r::renderEcharts4r({
    accidents_sum_bikes %>%
    dplyr::group_by(type) %>%
    echarts4r::e_charts(Gemeinde_name) %>%
    echarts4r::e_bar(accidents) %>%
    echarts4r::e_x_axis(axisTick = list(interval = 0),
                        axisLabel = list(rotate = 30),
                        nameGap = 35) %>%
    echarts4r::e_grid(bottom = 100, left = 150)
  })
}
```

ROTATE LABELS
TO SHOW ALL GEMEINDE_NAMES

R/MOD_BARS_ECHARTS.R

```
mod_bars_echarts_server <- function(input, output, session){
  ns <- session$ns

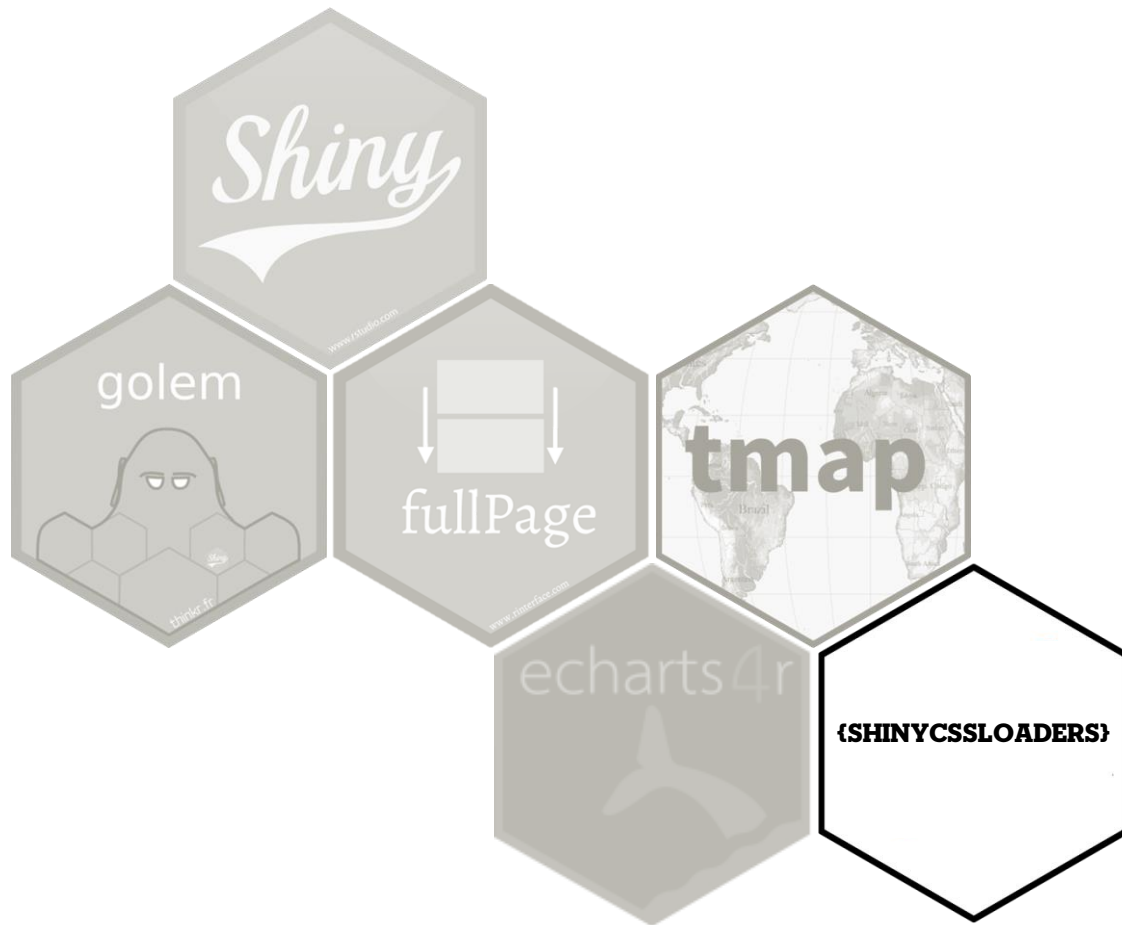
  output$bars <- echarts4r::renderEcharts4r({
    accidents_sum_bikes %>%
    dplyr::group_by(type) %>%
    echarts4r::e_charts(Gemeinde_name) %>%
    echarts4r::e_bar(accidents) %>%
    echarts4r::e_x_axis(axisTick = list(interval = 0),
                        axisLabel = list(rotate = 30),
                        nameGap = 35) %>%
    echarts4r::e_grid(bottom = 100, left = 150) %>%
    echarts4r::e_toolbox_feature(feature = "dataZoom") %>%
    echarts4r::e_toolbox_feature(feature = "dataView") %>%
    echarts4r::e_toolbox(bottom = 0)
  })
}
```

ADD TOOLBOX

```
mod_bars_echarts_ui <- function(id){  
  ns <- NS(id)  
  fullPage::pageContainer(  
    h1("An interactive bar chart"),  
    br(),  
    echarts4r::echarts4rOutput(ns("bars"), height = "50vh")  
    br(), br(),  
    p("Source: Statistische Ämter des Bundes und der Länder via",  
      tags$a(href="https://unfallatlas.statistikportal.de/_opendata2020.html", "Unfallatlas"))  
  )  
}
```

```
mod_bars_echarts_ui <- function(id){  
  ns <- NS(id)  
  fullPage::pageContainer(  
    h1("An interactive bar chart"),  
    br(),  
    echarts4r::echarts4rOutput(ns("bars"), height = "50vh")  
    br(), br(),  
    p("Source: Statistische Ämter des Bundes und der Länder via",  
      tags$a(href="https://unfallatlas.statistikportal.de/_opendata2020.html", "Unfallatlas"))  
  )  
}
```

```
mod_bars_echarts_ui <- function(id){  
  ns <- NS(id)  
  fullPage::pageContainer(  
    h1("An interactive bar chart"),  
    br(),  
    echarts4r::echarts4rOutput(ns("bars"), height = "50vh")  
    br(), br(),  
    p("Source: Statistische Ämter des Bundes und der Länder via",  
      tags$a(href="https://unfallatlas.statistikportal.de/_opendata2020.html", "Unfallatlas"))  
  )  
}
```

{shinycssloaders} - Add loading animations to a Shiny output while it's recalculating

CRAN 1.0.0 build passing

When a Shiny output (such as a plot, table, map, etc.) is recalculating, it remains visible but gets greyed out. Using {shinycssloaders}, you can add a loading animation ("spinner") to outputs instead of greying them out. By wrapping a Shiny output in `withSpinner()`, a spinner will automatically appear while the output is recalculating.

You can choose from one of 8 built-in animation types, and customize the colour/size. You can also use your own image instead of the built-in animations. See the [demo Shiny app](#) online for examples.

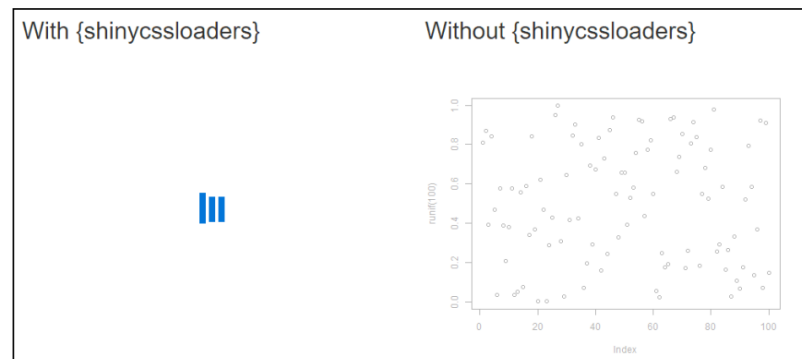
Table of contents

- [Example](#)
- [How to use](#)
- [Installation](#)
- [Features](#)
- [Sponsors](#) 🏆

Example

For interactive examples and to see some of the features, [check out the demo app](#).

Below is a simple example of what {shinycssloaders} looks like:



Contributors 3



daattali Dean Attali



andrewsali



sTeamTraen Nick Brown

Languages



● CSS 65.6% ● R 28.5%

● JavaScript 5.9%



#CORNELCON2020



cedricscherer.netlify.com



cedricphilippscherer@gmail.com










@CedScherer



@Z3tt

Single Element CSS Spinners

BG  FG 

			
View Source	View Source	View Source	View Source
			
View Source	View Source	View Source	View Source



Created by @luketuan

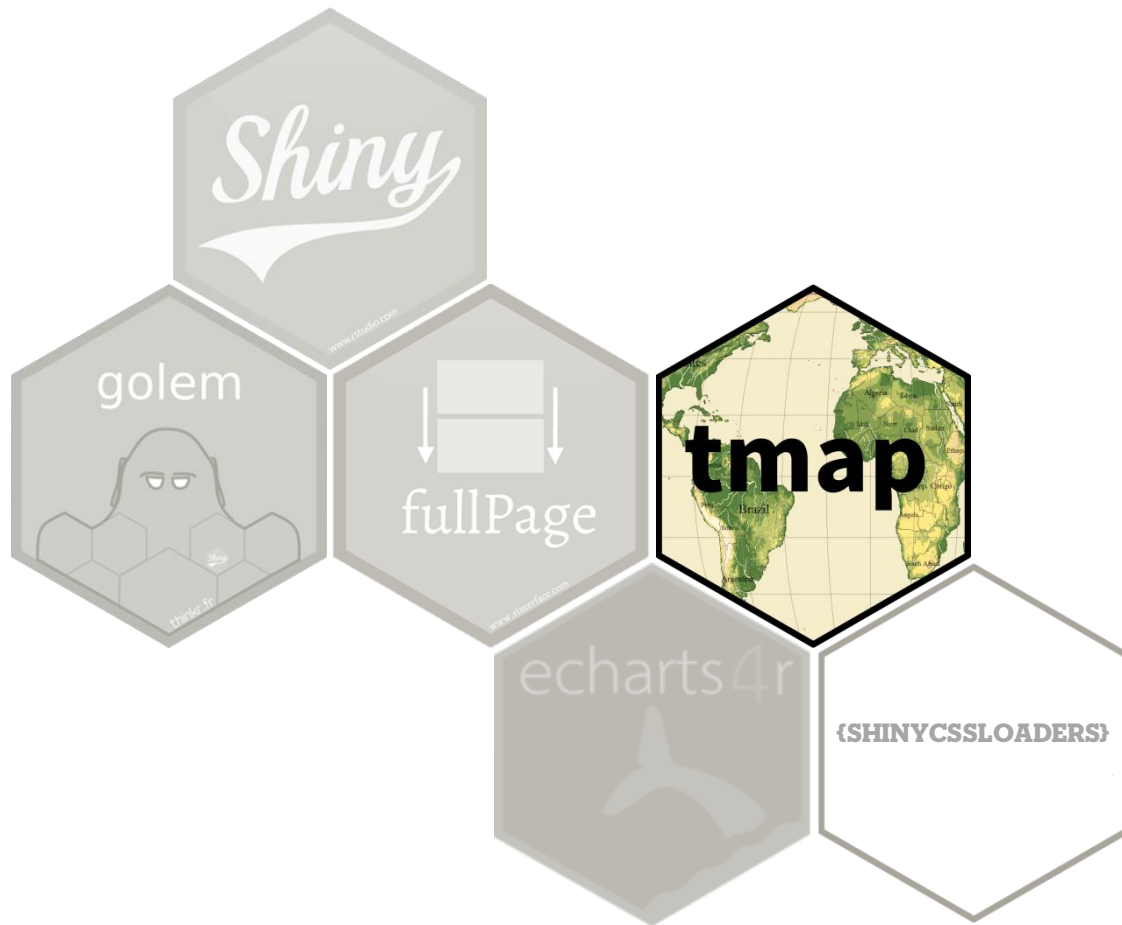
```
mod_bars_echarts_ui <- function(id){  
  ns <- NS(id)  
  fullPage::pageContainer(  
    pageContainer(  
      hl("An interactive bar chart"),  
      br(),  
      shinycssloaders::withSpinner(  
        echarts4r::echarts4rOutput(ns("bars"), height = "50vh")  
      )  
    ),  
    br(), br(),  
    p("Source: Statistische Ämter des Bundes und der Länder via",  
      tags$a(href="https://unfallatlas.statistikportal.de/_opendata2020.html", "Unfallatlas"))  
  )  
}
```

R/MOD_BARS_ECHARTS.R

```
mod_bars_echarts_ui <- function(id){
  ns <- NS(id)
  fullPage::pageContainer(
    pageContainer(
      hl("An interactive bar chart"),
      br(),
      shinycssloaders::withSpinner(
        echarts4r::echarts4rOutput(ns("bars"), height = "50vh")
      )
    ),
    br(), br(),
    p("Source: Statistische Ämter des Bundes und der Länder via",
      tags$a(href="https://unfallatlas.statistikportal.de/_opendata2020.html", "Unfallatlas"))
  )
}
```

APP_UI.R

```
options(
  spinner.type = 7,
  spinner.color="#1a579",
  spinner.size = 1
)
```



tmap: get started!

- Hello World!
- Interactive maps
- Multiple shapes and layers
- Facets
- Basemaps and overlay tile maps
- Options and styles
- Exporting maps
- Shiny integration
- Quick thematic map
- Tips n Tricks

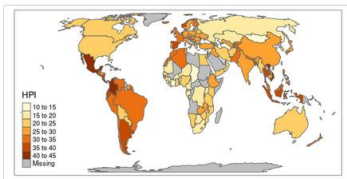
With the tmap package, thematic maps can be generated with great flexibility. The syntax for creating plots is similar to that of `ggplot2`, but tailored to maps. This vignette is for those who want to get started with tmap within a couple of minutes. A more detailed description of tmap can be found in an [article](#) published in the *Journal of Statistical Software* [33]. However, that article describes tmap version 1.1.1-2, which is out-of-date. Some major changes have been made since then, which are described in [vignettes](#) ("tmap-changes"). For more context on R's geographic capabilities we recommend the online version of the book [Geocomputation with R](#). The [Making maps with R](#) chapter of the book provides many more context and abundant code examples of map making with `sf` and other packages. Other good resources are the vignettes of the `sf` package, and the website [spatial.org](#).

Hello World!

A good place to start is to create a map of the world. After [installing](#) tmap, the following lines of code should create the map shown below:

```
library(tmap)
data("iso1a")

tm_shape(iso1a) +
  tm_polygons("HPI")
```

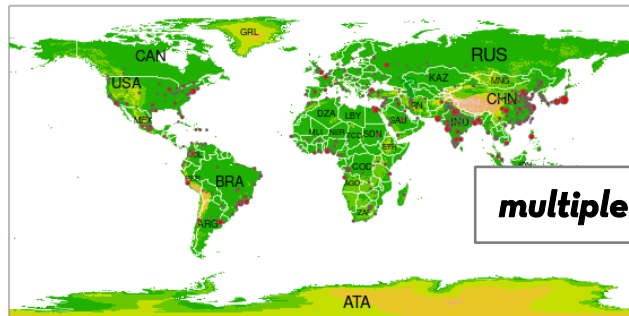
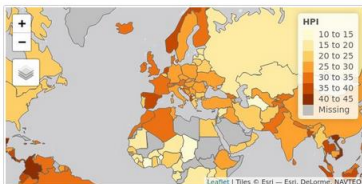


The object `iso1a` is a spatial object of class `sf` from the [sf](#) package. It is a `data.frame` with a special column that contains a geometry for each row, in this case polygons. In order to plot it in tmap, you first need to specify it with `tm_shape(iso1a)`. Layers can be added with the `+` operator, in this case `tm_polygons`. There are many layer functions in tmap, which can easily be found in the documentation by their `sf_` prefix. See also ["tmap-elements"](#).

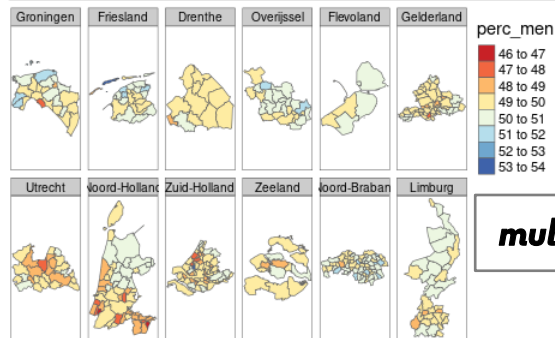
Interactive maps

Each map can be plotted as a static image or viewed interactively using `"static"` and `"viewer"` modes, respectively. The mode can be set with the function `tm_map_mode`, and toggling between the modes can be done with the `switch_viewer()` (which stands for toggle thematic map).

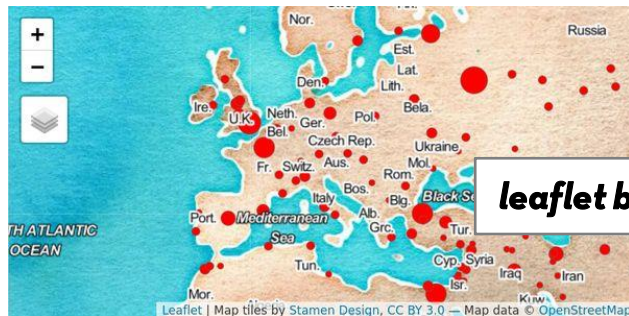
```
tm_map_mode("viewer")
tm_shape(iso1a) +
  tm_polygons("HPI")
```



multiple shapes and layers



multiple maps via facets



leaflet base- and tilemaps



```
mod_map_tmap_server <- function(input, output, session){  
  ns <- session$ns  
  
  output$map <- renderTmap({  
    tmap::tm_shape(traffic_summary_int, name = "Ratio Map") +  
      tmap::tm_polygons(  
        id = "NAME.x",  
        col = "bike"  
      )  
  })  
}
```



```
mod_map_tmap_server <- function(input, output, session){  
  ns <- session$ns  
  
  output$map <- renderTmap({  
    tmap::tm_shape(traffic_summary_int, name = "Ratio Map") +  
      tmap::tm_polygons(  
        id = "NAME.x",  
        col = "bike"  
      )  
  })  
}
```

RENDER MAP FOR UI

```
mod_map_tmap_server <- function(input, output, session){
  ns <- session$ns

  output$map <- renderTmap({
    tmap::tm_shape(traffic_summary_int, name = "Ratio Map") +
      tmap::tm_polygons(
        id = "NAME.x",
        col = "bike"
      )
  })
}
```

CHLOROPLETH MAP
 POLYGONS NAME.X
 COLORED BY BIKE

```
mod_map_tmap_server <- function(input, output, session){
  ns <- session$ns

  output$map <- renderTmap({
    tmap::tm_shape(traffic_summary_int, name = "Ratio Map") +
      tmap::tm_polygons(
        id = "NAME.x",
        col = "bike", alpha = .75, border.col = "white",
        palette = rev(rcartocolor::carto_pal(n = 5, "ag_Sunset")),
        breaks = c(1, 5, 10, 15, 20, 25, 30, Inf),
        legend.reverse = TRUE,
        textNA = "No Accidents in 2019",
        title = "Number of reported\nbike accidents",
        popup.vars = c("District:" = "Gemeinde_name",
                      "Total number of bike accidents:" = "n_total",
                      "Accidents on bicycle infrastructure:" = "n_bike",
                      "Proportion:" = "perc_bike")
      )
  })
}
```

MODIFY COLORS + LEGEND

```
mod_map_tmap_server <- function(input, output, session){
  ns <- session$ns

  output$map <- renderTmap({
    tmap::tm_shape(traffic_summary_int, name = "Ratio Map") +
      tmap::tm_polygons(
        id = "NAME.x",
        col = "bike", alpha = .75, border.col = "white",
        palette = rev(rcartocolor::carto_pal(n = 5, "ag_Sunset")),
        breaks = c(1, 5, 10, 15, 20, 25, 30, Inf),
        legend.reverse = TRUE,
        textNA = "No Accidents in 2019",
        title = "Number of reported\nbike accidents",
        popup.vars = c("District:" = "Gemeinde_name",
                      "Total number of bike accidents:" = "n_total",
                      "Accidents on bicycle infrastructure:" = "n_bike",
                      "Proportion:" = "perc_bike")
      )
  })
}
```

MODIFY TOOLTIPS

```

mod_map_tmap_ui <- function(id){
  ns <- NS(id)
  fullPage::pageContainer(
    tags$style(
      type = "text/css",
      "div.info.legend.leaflet-control {text-align:left; }
      div.leaflet-control-layers-expanded {text-align:left;}"
    ),
    hl("An interactive map"), br(),
    shinycssloaders::withSpinner(
      tmap::tmapOutput(ns("map"), height = 530)
    ),
    br(), br(),
    p("Source: Statistische Ämter des Bundes und der Länder via",
      tags$a(href="https://unfallatlas.statistikportal.de/_opendata2020.html", "Unfallatlas"),
      "• Geoportal Berlin via",
      tags$a(href="https://data.technologiestiftung-berlin.de/", "Technologiestiftung Berlin"))
  )
}

```

```

mod_map_tmap_ui <- function(id){
  ns <- NS(id)
  fullPage::pageContainer(
    tags$style(
      type = "text/css",
      "div.info.legend.leaflet-control {text-align:left; }
      div.leaflet-control-layers-expanded {text-align:left;}"
    ),
    hl("An interactive map"), br(),
    shinycssloaders::withSpinner(
      tmap::tmapOutput(ns("map"), height = 530)
    ),
    br(), br(),
    p("Source: Statistische Ämter des Bundes und der Länder via",
      tags$a(href="https://unfallatlas.statistikportal.de/_opendata2020.html", "Unfallatlas"),
      "• Geoportal Berlin via",
      tags$a(href="https://data.technologiestiftung-berlin.de/", "Technologiestiftung Berlin"))
  )
}

```

```

mod_map_tmap_ui <- function(id){
  ns <- NS(id)
  fullPage::pageContainer(
    tags$style(
      type = "text/css",
      "div.info.legend.leaflet-control {text-align:left; }
      div.leaflet-control-layers-expanded {text-align:left;}"
    ),
    hl("An interactive map"), br(),
    shinycssloaders::withSpinner(
      tmap::tmapOutput(ns("map"), height = 530)
    ),
    br(), br(),
    p("Source: Statistische Ämter des Bundes und der Länder via",
      tags$a(href="https://unfallatlas.statistikportal.de/_opendata2020.html", "Unfallatlas"),
      "• Geoportal Berlin via",
      tags$a(href="https://data.technologiestiftung-berlin.de/", "Technologiestiftung Berlin"))
  )
}

```

```

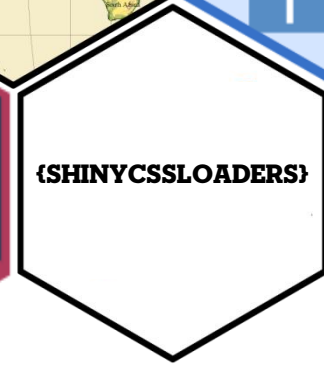
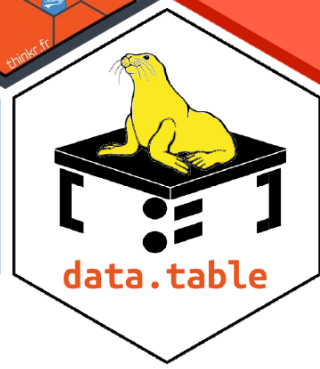
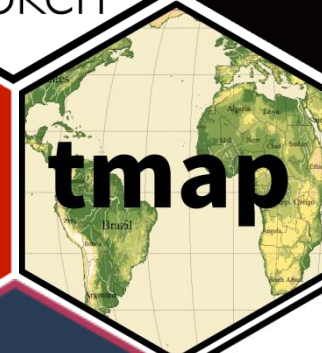
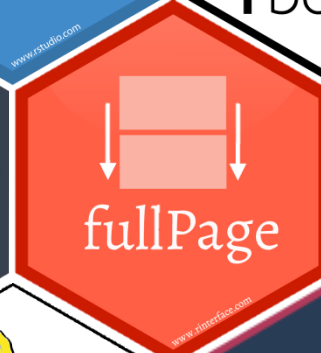
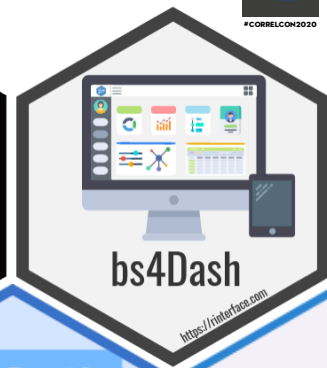
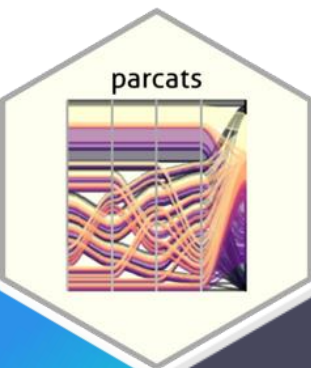
mod_map_tmap_ui <- function(id){
  ns <- NS(id)
  fullPage::pageContainer(
    tags$style(
      type = "text/css",
      "div.info.legend.leaflet-control {text-align:left; }
      div.leaflet-control-layers-expanded {text-align:left;}"
    ),
    hl("An interactive map"), br(),
    shinycssloaders::withSpinner(
      tmap::tmapOutput(ns("map"), height = 530)
    ),
    br(), br(),
    p("Source: Statistische Ämter des Bundes und der Länder via",
      tags$a(href="https://unfallatlas.statistikportal.de/_opendata2020.html", "Unfallatlas"),
      "• Geoportal Berlin via",
      tags$a(href="https://data.technologiestiftung-berlin.de/", "Technologiestiftung Berlin"))
  )
}

```


Codes on GitHub:

Z3TT/CORRELCON2020_GOLEM_HTML_WIDGETS

Additional Code for adding button in branch “buttons”



RESOURCES



Books

- „Mastering Shiny“ by Hadley Wickham (mastering-shiny.org)
- „Engineering Production-Grade Shiny Apps“ by Colin Fay *et al.* (engineering-shiny.org)
- „JavaScript for R“ by Joene Coene (javascript-for-r.com)

Packages for charts

- {plotly} (plot.ly/r)
- {echarts4r} (echarts4r.john-coene.com)
- {ggiraph} (davidgohel.github.io/ggiraph)
- {highcharter} (jkunst.com/highcharter)
- {dygraphs} (rstudio.github.io/dygraphs)
- {charter} (github.com/johncoene/charter)
- {rbokeh} (hafen.github.io/rbokeh)
- {metricsgraphics} (hrbrmstr.github.io/metricsgraphics)
- {rthreejs} (github.com/bwlewis/rthreejs)
- {visnetwork} (dataknowledge.github.io/visNetwork)
- {networkD3} (christophergandrud.github.io/networkD3/)
- {DiagrammeR} (rich-iannone.github.io/DiagrammeR)

Packages for tables

- {DT} (rstudio.github.io/DT)
- {formattable} (renkun-ken.github.io/formattable)

Packages for maps

- {leaflet} (rstudio.github.io/leaflet)
- {tmap} (github.com/mtennekes/tmap)
- {mapdeck} (symbolixau.github.io/mapdeck)
- {echarts4r} (echarts4r.john-coene.com)

Packages for theming

- {shinydashboard} (rstudio.github.io/shinydashboard/)
- {shinydashboardPlus} (rinterface.github.io/shinydashboardPlus)
- {fullPage} (rinterface.github.io/fullPage)
- {bs4Dash} (rinterface.github.io/bs4Dash)
- {shinybulma} (rinterface.github.io/shinybulma)

RESOURCES



Books

- „Mastering Shiny“ by Hadley Wickham (mastering-shiny.org)
- „Engineering Production-Grade Shiny Apps“ by Colin Fay *et al.* (engineering-shiny.org)
- „JavaScript for R“ by Joene Coene (javascript-for-r.com)

Packages for charts

- {plotly} (plot.ly/r)
- {echarts4r} (echarts4r.john-coene.com)
- {ggiraph} (davidgohel.github.io/ggiraph)
- {highcharter} (jkunst.com/highcharter)
- {dygraphs} (rstudio.github.io/dygraphs)
- {charter} (github.com/johncoene/charter)
- {rbokeh} (hafen.github.io/rbokeh)
- {metricsgraphics} (hrbrmstr.github.io/metricsgraphics)
- {rthreejs} (github.com/bwlewis/rthreejs)
- {visnetwork} (dataknowledge.github.io/visNetwork)
- {networkD3} (christophergandrud.github.io/networkD3/)
- {DiagrammeR} (rich-iannone.github.io/DiagrammeR)

gallery.htmlwidgets.org
nanxstats/awesome-shiny-extensions

Packages for tables

- {DT} (rstudio.github.io/DT)
- {formattable} (renkun-ken.github.io/formattable)

Packages for maps

- {leaflet} (rstudio.github.io/leaflet)
- {tmap} (github.com/mtennekes/tmap)
- {mapdeck} (symbolixau.github.io/mapdeck)
- {echarts4r} (echarts4r.john-coene.com)

Packages for theming

- {shinydashboard} (rstudio.github.io/shinydashboard/)
- {shinydashboardPlus} (rinterface.github.io/shinydashboardPlus)
- {fullPage} (rinterface.github.io/fullPage)
- {bs4Dash} (rinterface.github.io/bs4Dash)
- {shinybulma} (rinterface.github.io/shinybulma)

THANK YOU!



cedricscherer.netlify.com



cedricphilippscherer@gmail.com



[@CedScherer](https://twitter.com/CedScherer)



[@Z3tt](https://github.com/Z3tt)

