

Don't reinvent the wheel: making use of shiny extension packages



Münster R User Group

Hi!

I'm Suthira Owlarn :)

- Biologist/postdoc at the Max Planck Institute
- R newbie (~1.5 years)
- Shiny enthusiast!

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

 [sowla](https://github.com/sowla)



[2019-MS-RUG-shiny-extensions](https://github.com/2019-MS-RUG-shiny-extensions)

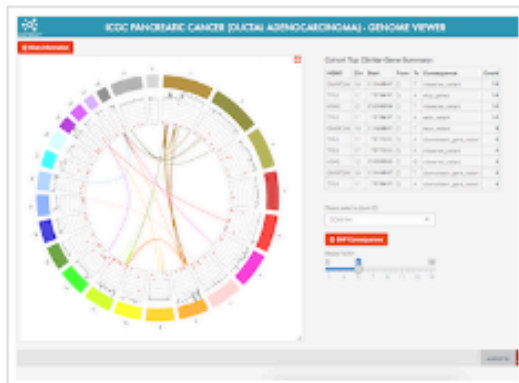


**USING
THE RIGHT TOOLS
SAVES
A LOT OF PAIN**

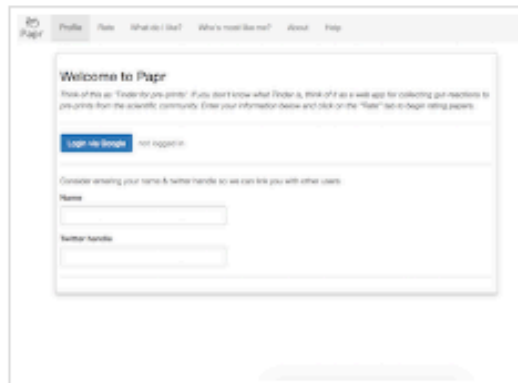
Shiny

Shiny User Showcase

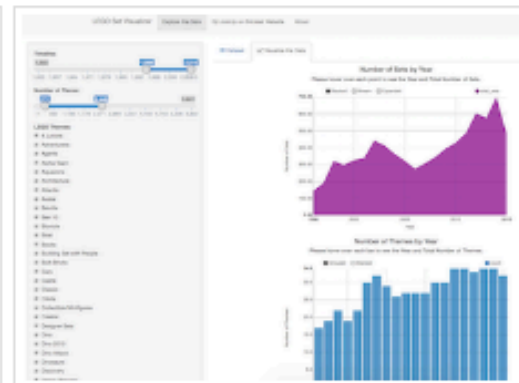
The [Shiny User Showcase](#) contains an inspiring set of sophisticated apps developed and contributed by Shiny users.



Genome browser



Papr



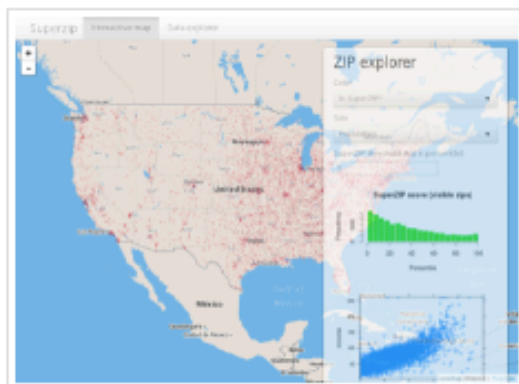
Lego Set Database Explorer



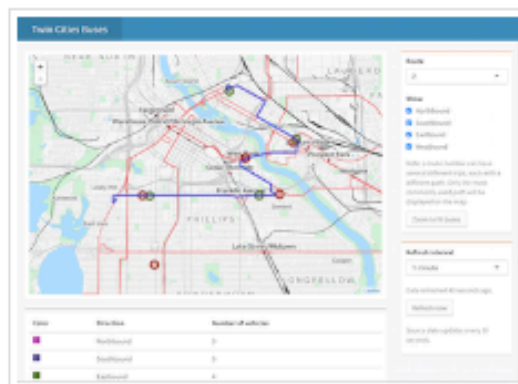
See more

Interactive visualizations

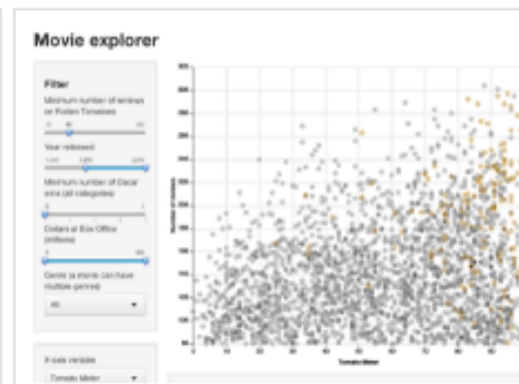
Shiny is designed for fully interactive visualization, using JavaScript libraries like [d3](#), [Leaflet](#), and [Google Charts](#).



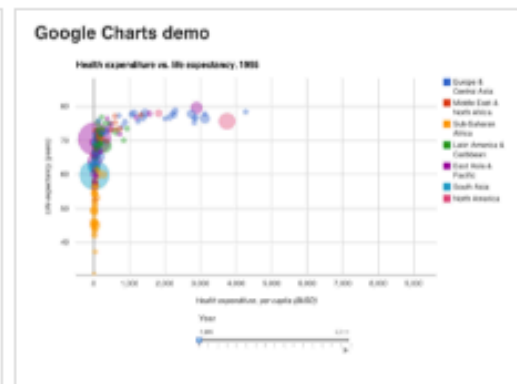
SuperZip example



Bus dashboard

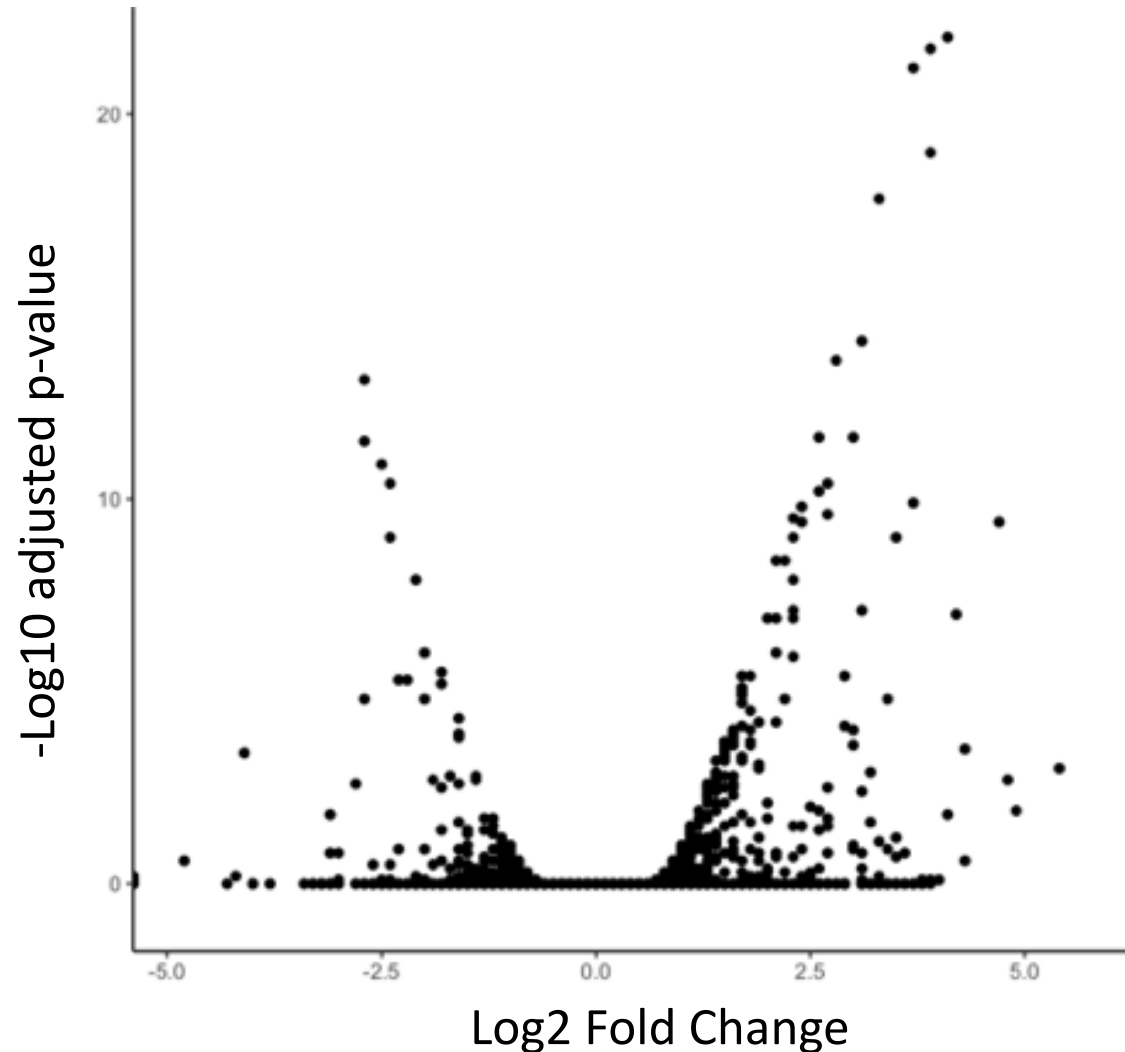


Movie explorer



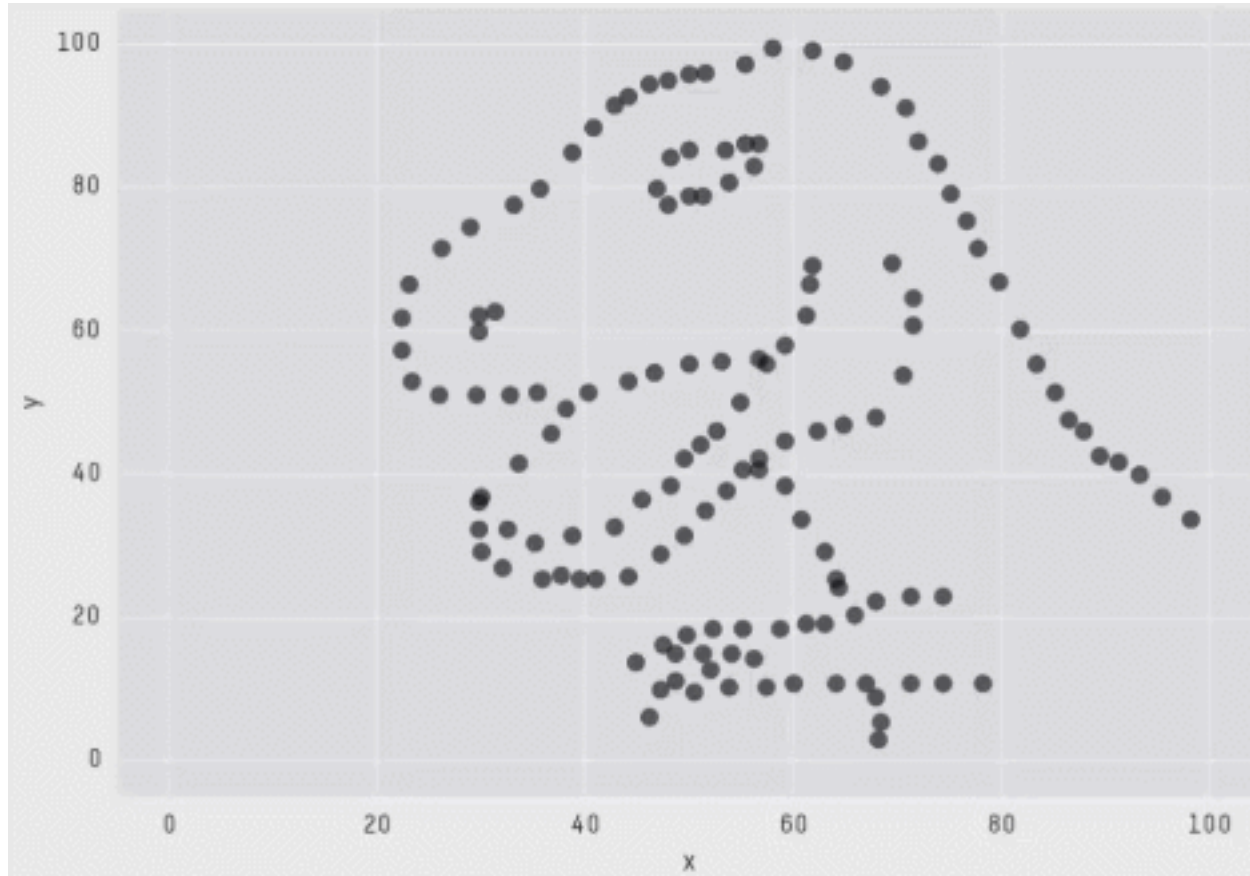
Google Charts

Why did I start using shiny?



	A	B	C	D
1	Id	baseMean	log2Foldchange	padj
2	dd_Smed_v6_102_0_1	796.510123	-1.227211103	0.01294388
3	dd_Smed_v6_10302_0_1	369.428622	0.987704116	6.72E-08
4	dd_Smed_v6_1033_0_1	1855.65034	0.989061916	0.00671085
5	dd_Smed_v6_1054_0_1	5001.65104	-0.828711614	0.0009754
6	dd_Smed_v6_10849_0_1	202.860402	-0.500429771	0.04398802
7	dd_Smed_v6_1114_0_1	3545.29634	0.653791911	0.01336137
8	dd_Smed_v6_11501_0_1	63.8936551	-1.289230411	1.04E-06
9	dd_Smed_v6_1224_0_1	941.936102	0.622942178	0.01667194
10	dd_Smed_v6_12355_0_1	277.217068	-1.022192875	0.0162604
11	dd_Smed_v6_13291_0_1	56.5819099	-0.978517859	0.00789344
12	dd_Smed_v6_1350_0_1	117.100205	-1.160414409	0.01550004
13	dd_Smed_v6_14068_0_1	82.338354	-1.097416225	0.01971501
14	dd_Smed_v6_1509_0_1	1042.41145	-0.553030165	0.00980481
15	dd_Smed_v6_1519_0_1	955.358024	0.753083458	0.02762951
16	dd_Smed_v6_1531_0_1	652.722592	-1.148139028	1.39E-07
17	dd_Smed_v6_158_0_1	377.730819	-3.076838854	4.73E-20
18	dd_Smed_v6_1580_0_1	2953.09382	1.802740685	1.87E-13
19	dd_Smed_v6_1581_0_1	1773.91345	-0.524144223	0.00198497
20	dd_Smed_v6_1596_0_1	505.382348	-0.602958258	0.00268586
21	dd_Smed_v6_1753_0_1	66.3417388	1.73775701	0.0042826
22	dd_Smed_v6_1772_0_1	3602.39903	0.638196287	0.01017817
23	dd_Smed_v6_1774_0_1	366.528291	1.348691806	0.00078916
24	dd_Smed_v6_1778_0_1	551.331537	0.400441333	0.0071733

Why did I start using shiny?



```
X Mean: 54.2659224  
Y Mean: 47.8313999  
X SD   : 16.7649829  
Y SD   : 26.9342120  
Corr.  : -0.0642526
```

Demo



"OH, NO -- THAT'S JUST THE
PROTOTYPE."

Let's chat! :)

- Do you actually need shiny?
- Straight out of the shiny box
- Judging a book by its cover
- More ways to add to add to the UI/UX
- Tools to help your shiny workflow

Do you actually need shiny?

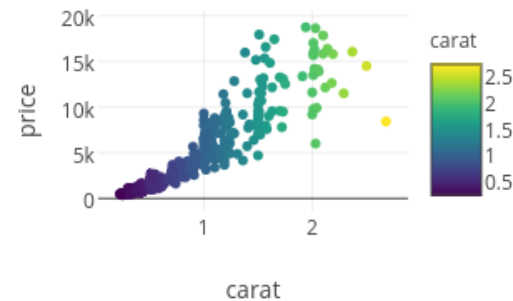
- [Crosstalk](#)

- HTML report demoed
- Linked data sets
- Filter/selection
- Some interactivity without needing shiny

- Some compatible widgets ->

- Make your own using [htmlwidgets](#) or [reactR](#)

[plotly](#)



[leaflet](#)



[summary widget](#)

You have selected 4 cars with an average mpg of **24.0**. The 2 cars with automatic transmission have an average mpg of **22.1**.

	mpg	cyl
Merc 230	22.8	4
Toyota Corona	21.5	4
Lotus Europa	30.4	4
Volvo 142E	21.4	4

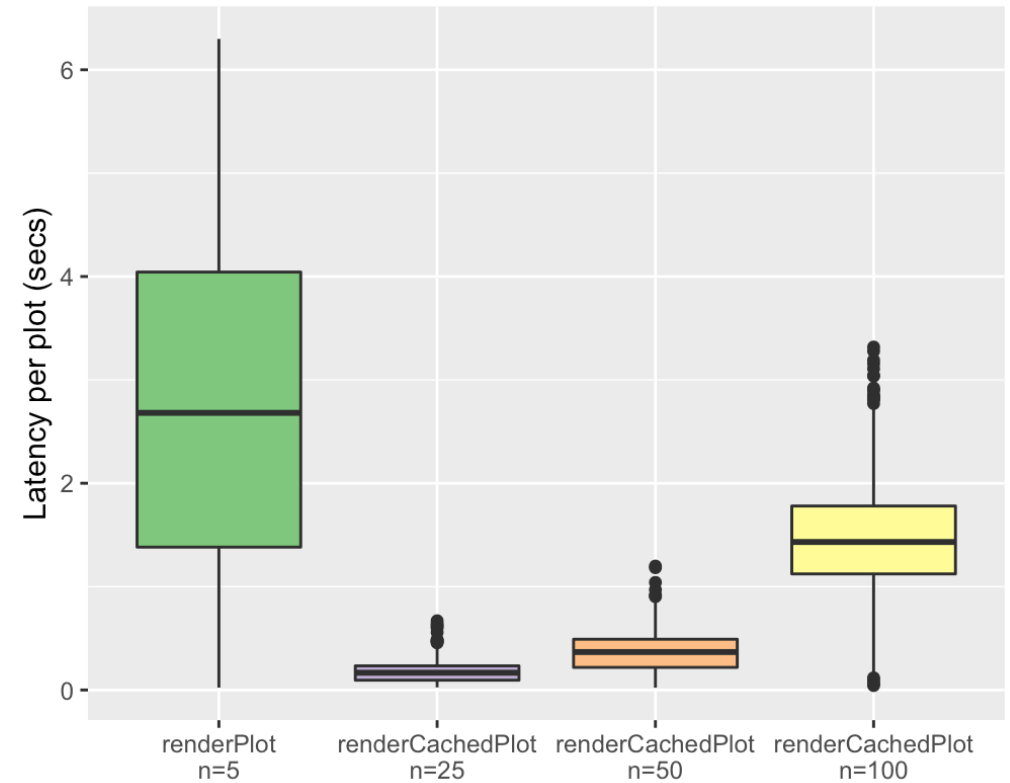
[DT](#)

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa

Showing 1 to 10 of 150 entries
Previous 1 2 3 4 5 ... 15 Next

Straight out of the shiny box

- Easy to make bootstrap-based web apps
 - Multiple inputs/outputs
 - Fine-tuned reactivity
 - [Dynamic UI](#)
 - [Modules](#)
 - [Bookmarking](#)
- [Shiny 1.2.0](#) (Nov 2018):
 - Plot caching



Judging a book by its cover

16 [shinythemes](#)

Flatly Navbar 1 Plot Table

File input:
Browse... No file se

Text input:
general

Slider input:
1 30 100
1 11 21 31 41 51 61 71 81 91 100

Default actionButton:
Search

actionButton with CSS class:
Action button

Tab 1 Tab 2 Tab 3

Table

speed	dist
4.00	2.00
4.00	10.00
7.00	4.00
7.00	22.00

Verbatim text output
general, 30, NULL

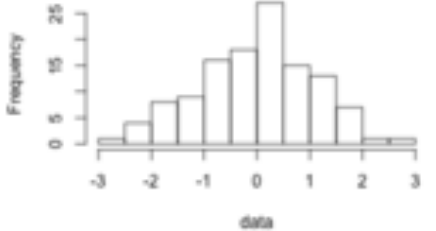
Header 1
Header 2
Header 3

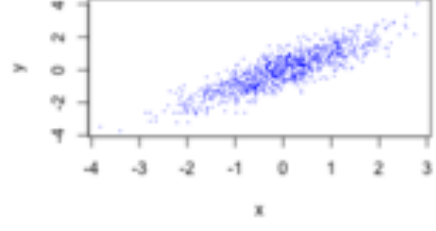
[shinydashboard](#)

My Dashboard

Search...

Dashboard Widgets Charts Source code for app

Distribution


View 1 View 2


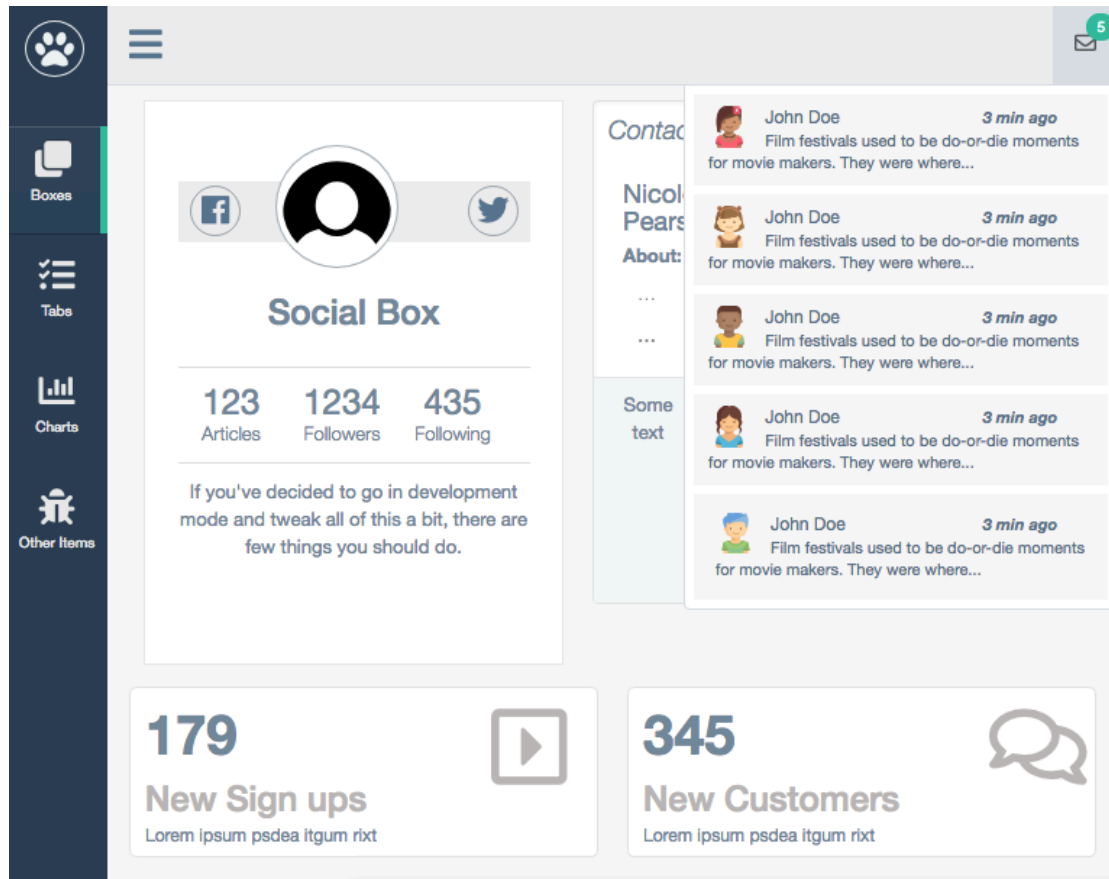
Histogram control
Count
1 120 500
1 50 100 150 200 250 300 350 400 450 500

Appearance
Fill
☒ None
☐ Blue
☐ Black
☐ red

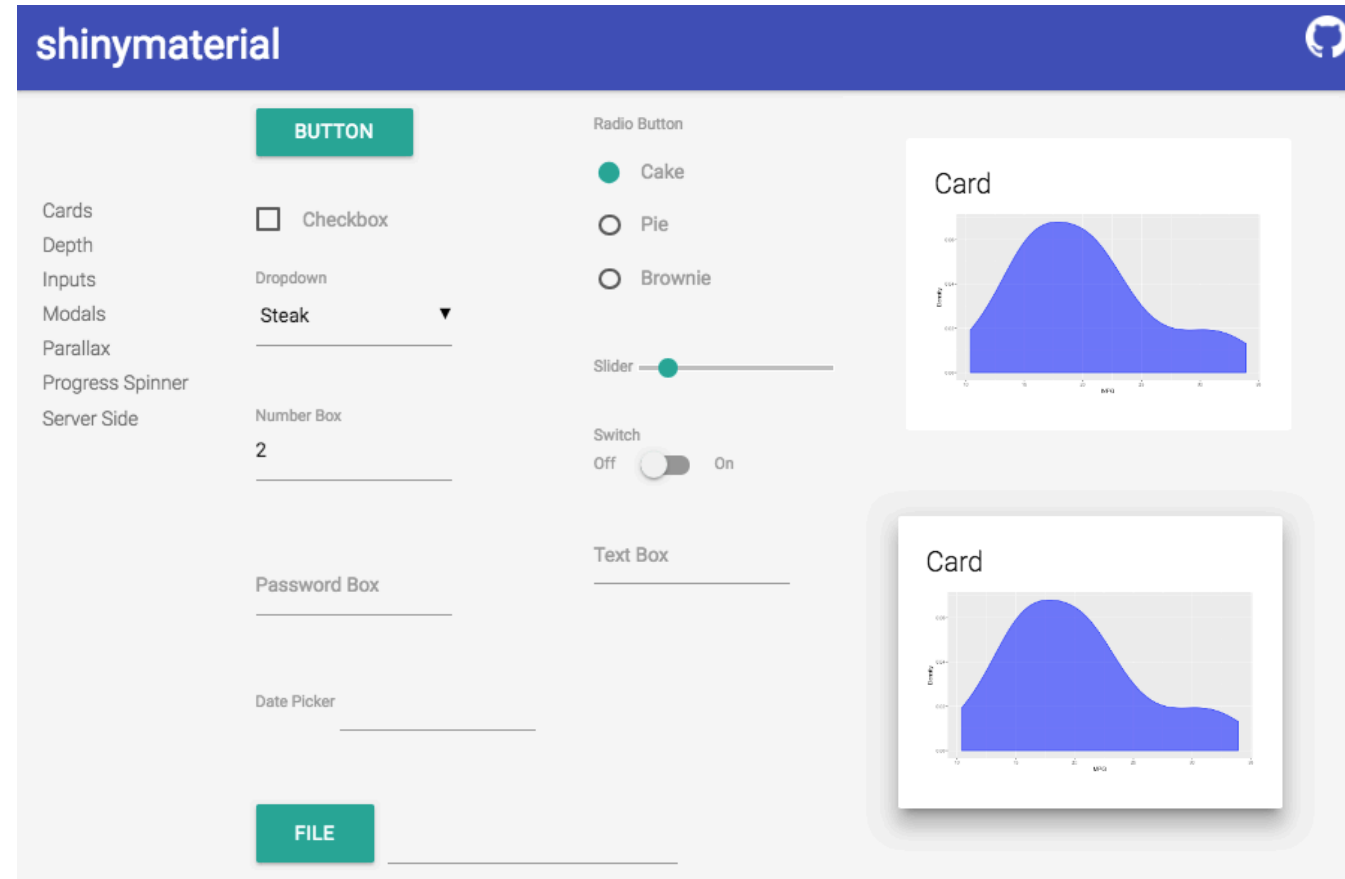
Scatterplot control
Spread
60%

Judging a book by its cover

[gentelellaShiny](#)



[shinymaterial](#)



Judging a book by its cover

[wired](#) (+ [xkcd](#))

Wired

Distribution:

☒ Normal

☐ Uniform

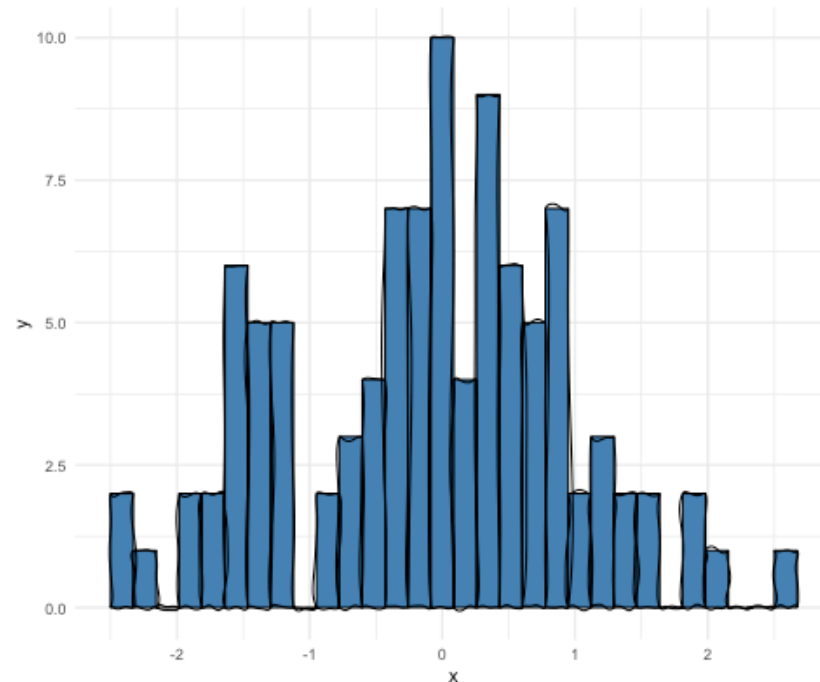
☐ Log-normal

☐ Exponential

Number of observations:

Use log scale ☐

Made with wired.js library



Plot made with the [xkcd](#) package.

[nessy](#)

{nessy}

NES-style CSS Framework for Shiny.

Buttons

Normal Primary hit Warning Error

Radio

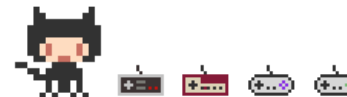
yes
no

Balloons

Hi!

Thanks for coming to today's talk!

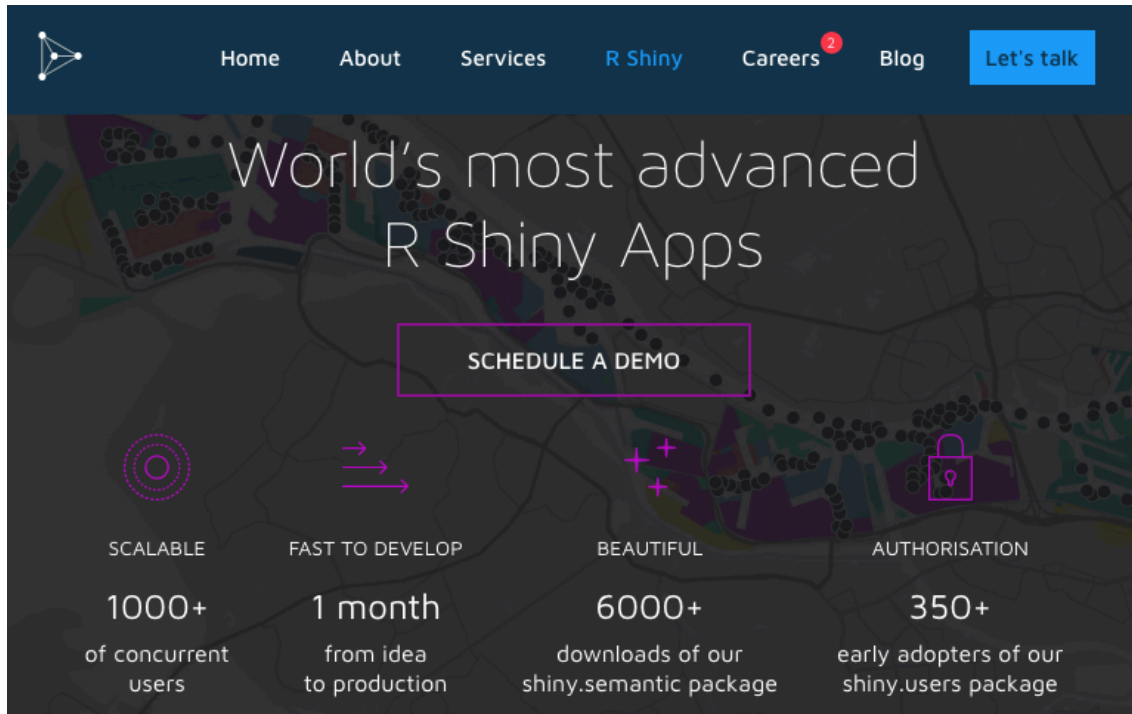
Icons



{nessy}, based on NES.css

Judging a book by its cover

[appsilon](#)



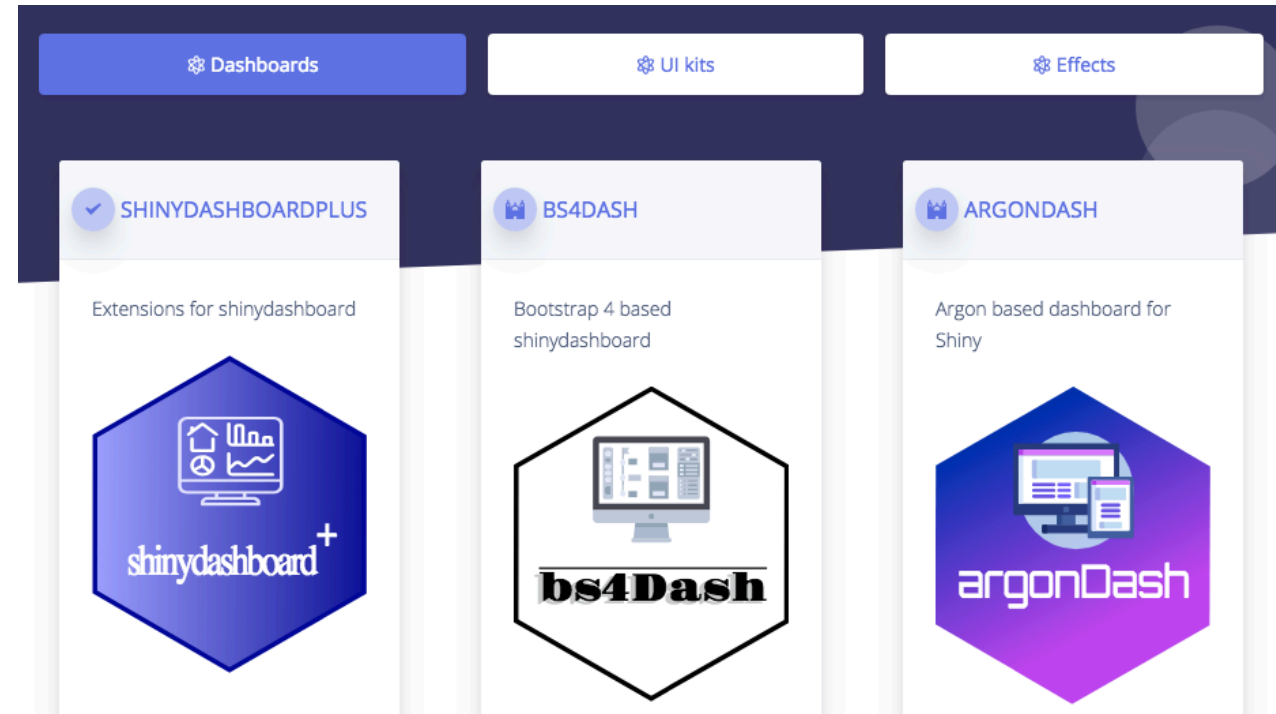
The screenshot shows the appsilon website with a dark blue header containing navigation links: Home, About, Services, R Shiny, Careers (with a red notification badge), Blog, and a blue 'Let's talk' button. The main content area features a map background with the text 'World's most advanced R Shiny Apps' and a 'SCHEDULE A DEMO' button. Below this, four features are listed in a grid:

SCALABLE	FAST TO DEVELOP	BEAUTIFUL	AUTHORISATION
1000+ of concurrent users	1 month from idea to production	6000+ downloads of our shiny.semantic package	350+ early adopters of our shiny.users package

[shiny.semantic](#) + [semantic.dashboard](#)

Also: shiny supports [HTML templates](#)

[rinterface](#)



The screenshot shows the rinterview website with a dark blue header containing navigation links: Dashboards, UI kits, and Effects. The main content area features three product cards:

- SHINYDASHBOARDPLUS**: Extensions for shinydashboard. The card shows a blue hexagon icon with a monitor and the text 'shinydashboard+'.
- BS4DASH**: Bootstrap 4 based shinydashboard. The card shows a black hexagon icon with a monitor and the text 'bs4Dash'.
- ARGONDASH**: Argon based dashboard for Shiny. The card shows a purple hexagon icon with a monitor and the text 'argonDash'.

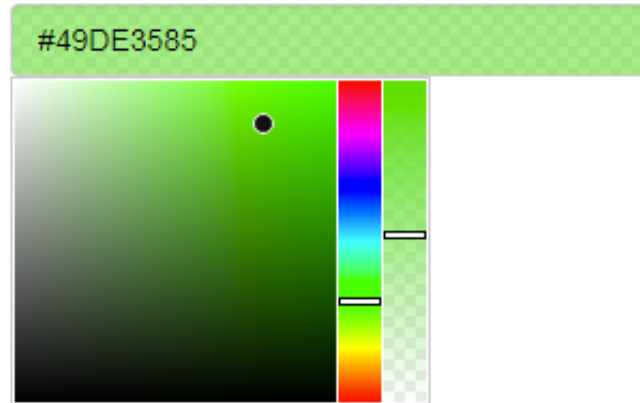
[shinydashboardPlus](#), [bs4Dash](#),
[argonDash](#) + [argonR](#), [shinybulma](#)

More ways to add to the UI/UX

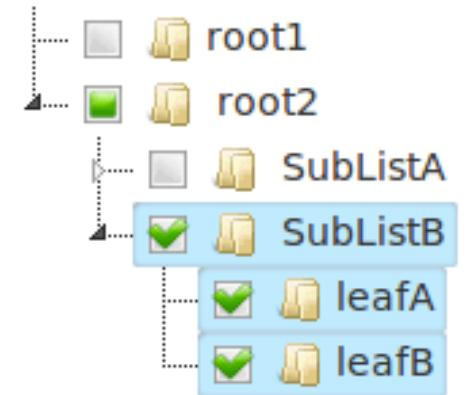
[shinyAce](#)

```
1 library(shiny)
2 library(shinyAce)
3
4 modes <- dir(system.file('www/ace', package='shinyAce'), "^mode-.*.js$")
5 modes <- sub("^mode-(.*).js$", "\\1", modes)
6
7 themes <- dir(system.file('www/ace', package='shinyAce'), "^theme-.*.js$")
8 themes <- sub("^theme-(.*).js$", "\\1", themes)
9
10 #' Define UI for application that demonstrates a simple Ace editor
11 #' @author Jeff Allen \email{jeff@trestletech.com}
12 shinyUI<
13   pageWithSidebar(
14     # Application title
15     headerPanel("Simple Shiny Ace!"),
16
17     sidebarPanel(
18       selectInput("mode", "Mode: ", choices=modes, selected="plain_text"),
19       selectInput("theme", "Theme: ", choices=themes, selected="textmate"),
20       actionButton("reset", "Reset Text"),
21       HTML("<hr />"),
22       helpText(HTML("A simple Shiny Ace editor.
23 | | | | <p>Created using <a href = \"http://github.com/trestle
24 | | | |
25
26 # Show the simple table
```

[colourpicker](#)



[shinyTree](#)




more UI elements:

[shinyWidgets](#), [shinyBS](#), [bsplus](#), [TileMaker](#)

More ways to add to the UI/UX

[shinyFeedback](#)

Warn if > 100

Ye be warned

Danger if < 0

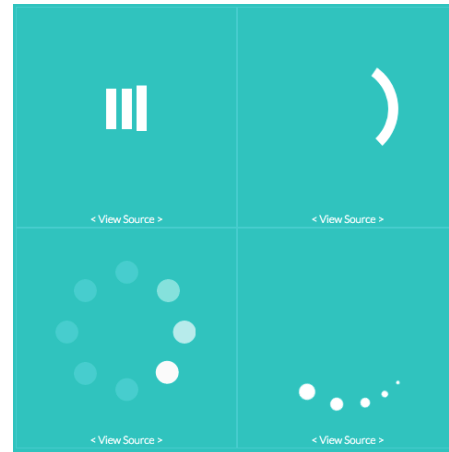
 

Danger, turn back!

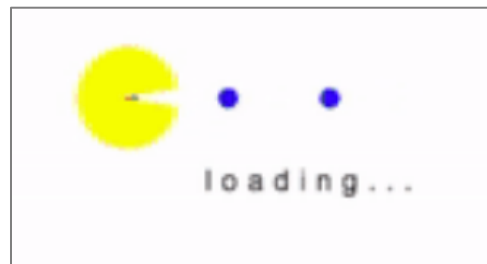
Success if > 3 characters

[shinycssloaders](#)



[shinycustomloader](#)



[rintrojs](#)

step-by-step introductions,
clickable hints (from intro.js)

[shinyjs](#)

Easy ways to eg.

- Show/hide
- Enable/disable

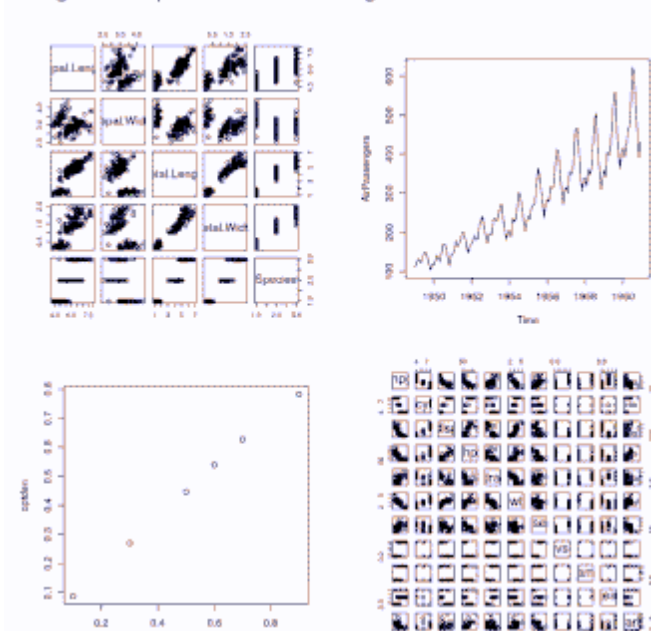
[shinyEffects](#)

CSS effects

More ways to add to the UI/UX

Interact with/edit elements

Drag and drop elements with dragulaR



[shinyjquery](#), [dragulaR](#),
[shinydnd](#)

[shinysense](#) – "help shiny sense the world around it"

- [shinyswipr](#) - cards that can be swiped left/right
- [shinydrawr](#), [shinyearr](#), [shinyviewr](#), [shinymovr](#)

[shiny.router](#) – helps create paged URLs

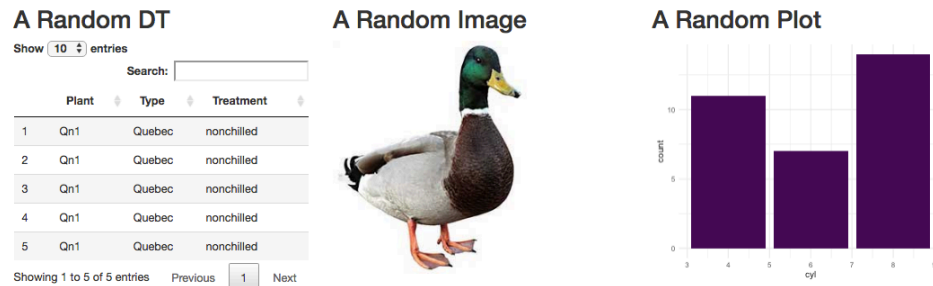
[shiny.i18n](#) – helps with internationalisation

[shiny.collections](#) – "Google Docs-like live collaboration in Shiny"

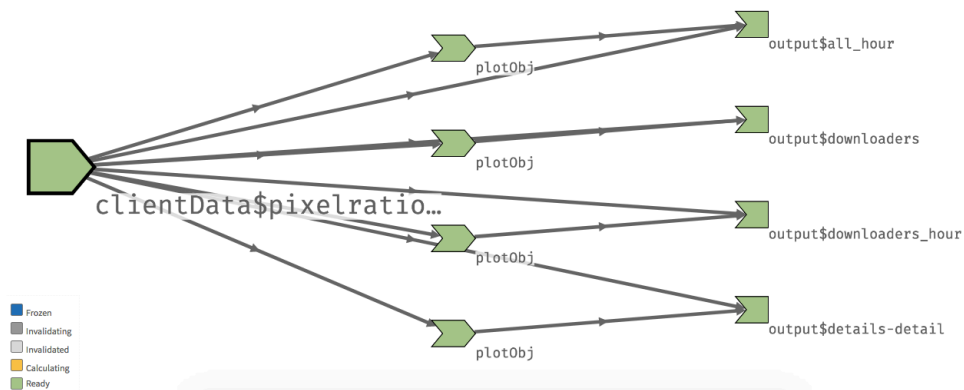
[fullPage](#) – fullscreen scrolling websites

Tools to help your shiny workflow

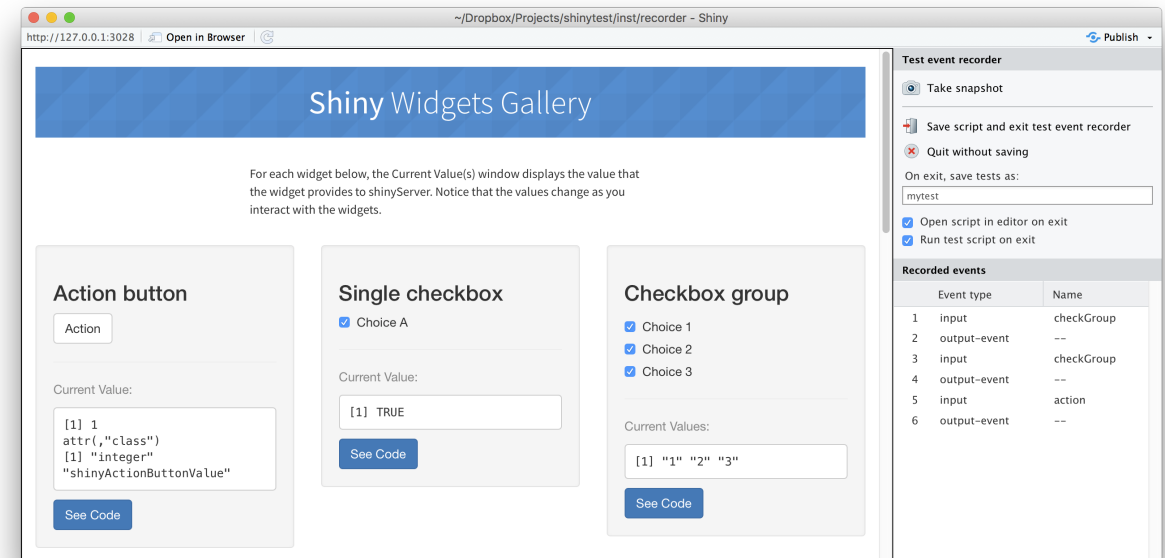
[shinipsum](#) (for prototyping)



[reactlog](#) ([talk](#), [demo](#))



[shinytest](#) (test overall function)



Shiny Widgets Gallery

For each widget below, the Current Value(s) window displays the value that the widget provides to shinyServer. Notice that the values change as you interact with the widgets.

Action button

Current Value:

```
[1] 1  
attr(,"class")  
[1] "integer"  
"shinyActionButtonValue"
```

See Code

Single checkbox

Choice A

Current Value:

```
[1] TRUE
```

See Code

Checkbox group

Choice 1
Choice 2
Choice 3

Current Values:

```
[1] "1" "2" "3"
```

See Code

Test event recorder

Take snapshot

Save script and exit test event recorder

Quit without saving

On exit, save tests as:

mytest

Open script in editor on exit

Run test script on exit

Recorded events

	Event type	Name
1	input	checkGroup
2	output-event	--
3	input	checkGroup
4	output-event	--
5	input	action
6	output-event	--

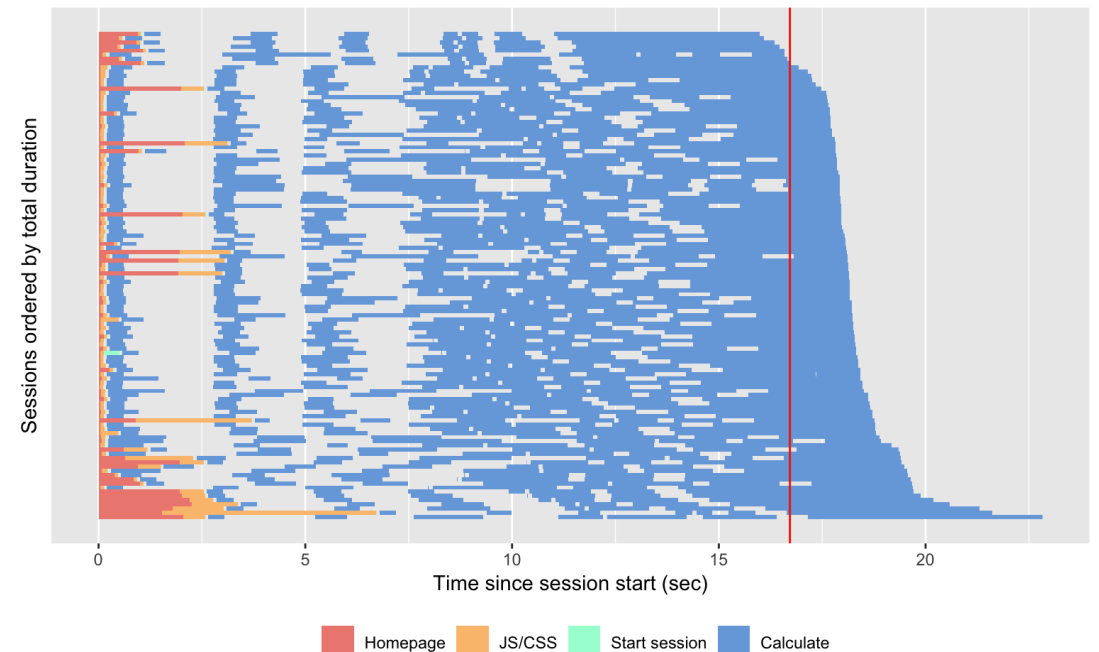
- alternatively, [RSelenium](#)
- unit test underlying functions, eg. [testthat](#)
- [frankenstein](#) to “resurrect” your app
- [shinysnippets](#) to save you some typing

Tools to help your shiny workflow

Performance workflow

1. Use **shinyloadtest** to see if it's fast enough
2. If not, use **profvis** to see what's making it slow
3. Optimize
 1. Move work out of Shiny (very often)
 2. Make code faster (very often)
 3. Use caching (sometimes)
 4. Use async (occasionally)
4. Repeat!

[shinyloadtest](#)
[profvis](#) (+ eg. [microbenchmark](#))



Joe Cheng, [rstudio::conf 2019 keynote](#)

Tools to help your shiny workflow

Performance workflow

1. Use **shinyloadtest** to see if it's fast enough
 2. If not, use **profvis** to see what's making it slow
 3. Optimize
 1. Move work out of Shiny (very often)
 2. Make code faster (very often)
 3. Use caching (sometimes)
 4. Use async (occasionally)
 4. Repeat!
- Asynchronous programming
 - [future](#), [promises](#)
 - [ipc](#)
 - [furrr](#)
- Joe Cheng, [rstudio::conf 2019 keynote](#)

Tools to help your shiny workflow

- Shiny + [Electron](#)
 - Alternative to [shinyapps.io/Shiny Server/RStudio Connect](#)
 - Katie Sasso's UseR 2018 [talk](#)
(Shiny meets Electron: Turn your Shiny app into a **standalone desktop app** in no time)
 - Columbus Collaboratory's electron-quick-start github [repo](#)
- More materials:
 - Shiny in production: Joe Cheng's [keynote](#) and Kelly O'Briant's (WIP) [book](#)
 - Building Big Shiny Apps - A Workflow [blog post](#)
 - [shinytemplate](#) “an RStudio project template for building prod-ready shiny apps”

A few sources for shiny updates/advice

- [@ ColinFay/ @Thinkr FR](#)
 - eg. various [advice](#) on working with shiny, especially shiny in production
- [@ pvictorr/ @dreamRs fr](#)
 - eg. [esquisse](#): a package and RStudio add-in for ggplot2 GUI
- [@nic crane](#)
 - eg. a [javascript snippet](#) to get confirmation when clicking back/forward buttons
 - inspired by her, I added [a bit of code](#) for the refresh button as well
- Rstudio also has shiny [tutorials](#), [videos](#) and a [cheatsheet](#)
- Joe Cheng is writing a “Mastering Shiny” [book](#)

Don't reinvent the wheel: making use of shiny extension packages

Münster R User Group

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

 [sowla](https://github.com/sowla)



[2019-MS-RUG-shiny-extensions](https://github.com/2019-MS-RUG-shiny-extensions)