Storytelling with data

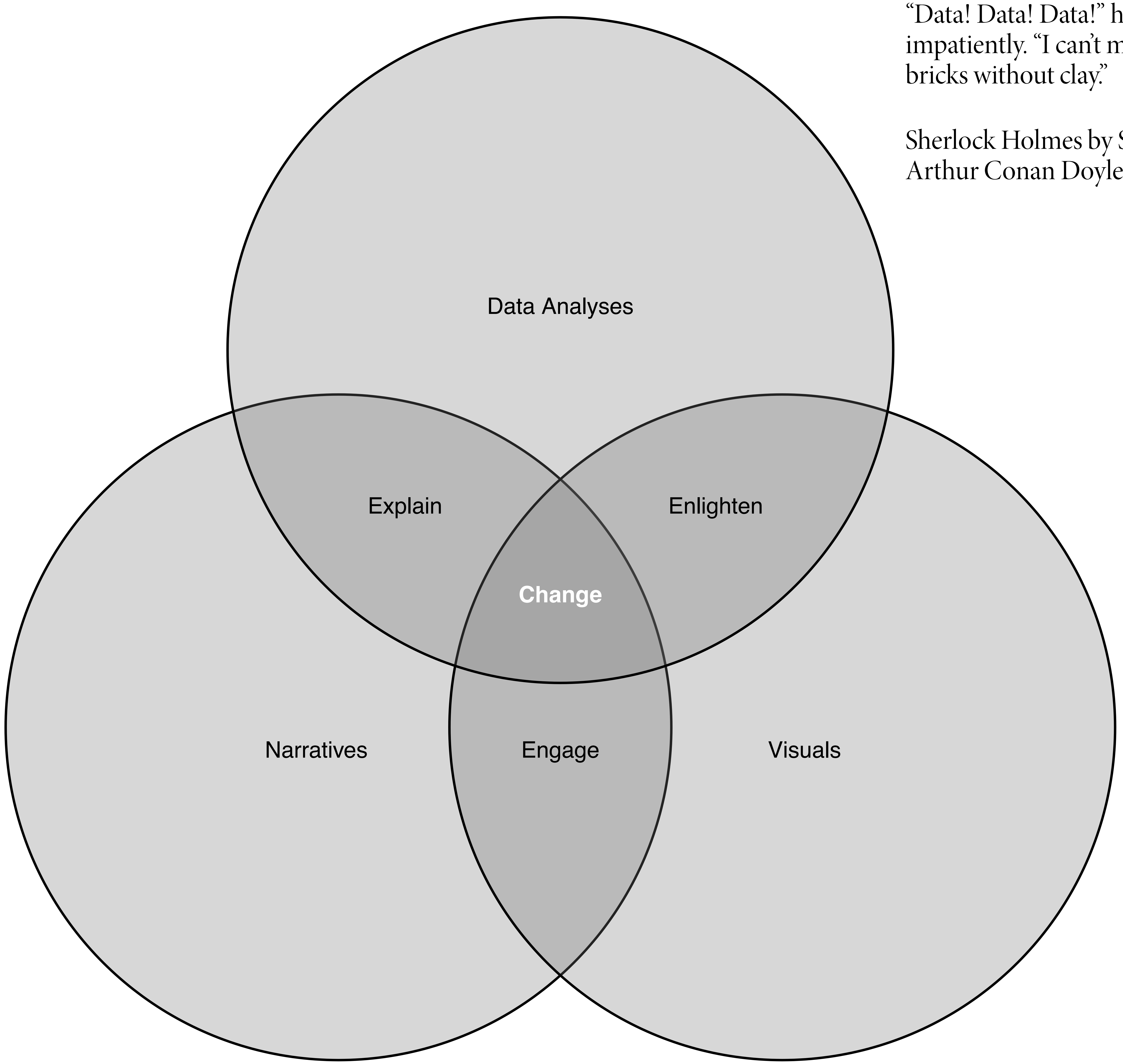
10 | Technologies and tools of interactive data-driven, visual design

Scott Spencer | Columbia University

course overview, learn to drive change using data visuals and narrative

“Data! Data! Data!” he cried impatiently. “I can’t make bricks without clay.”

Sherlock Holmes by Sir Arthur Conan Doyle, *author*



No one ever made a decision because of a number. They need a story.

Daniel Kahneman, *psychologist, behavioral economist, and author*

The greatest value of a picture is when it forces us to notice what we never expected to see.

John W Tukey, *mathematician*

general course deliverable timeline

Individual Work

For learning data visualization
and written narrative techniques

Group work

For building graphics and narrative
into interactive communications

Sept 30	Oct 14	Oct 28	Nov 18	Nov 18	Dec 11	Dec 13
Homework 1 graphics	Homework 2 graphics	Homework 3 writing	Homework 4 graphics	Proposal	Interactive Communication	Multimodal communication
10%	10%	10%	10%	15%	20%	15%

Participation 10%

next deliverables, **group interactive & multimodal communications**

Individual Work

For learning data visualization
and written narrative techniques

Group work

For building graphics and narrative
into interactive communications

Sept 30	Oct 14	Oct 28	Nov 18	Nov 18	Dec 11	Dec 13
Homework 1 graphics 10%	Homework 2 graphics 10%	Homework 3 writing 10%	Homework 4 graphics 10%	Proposal 15%	Interactive Communication 20%	Multimodal communication 15%
Participation 10%						

review of graphics practice

open-source technology stack for interactive, data-driven graphics

interactive technology stack, **components and relationships** — *click a technology below to learn more*

browser

event listeners

DOM

html
css
grid
svg
canvas

js

scrollama.js
d3.js
plotly.js
p5.js
react.js
jQuery.js
DataTables.js
three.js
...

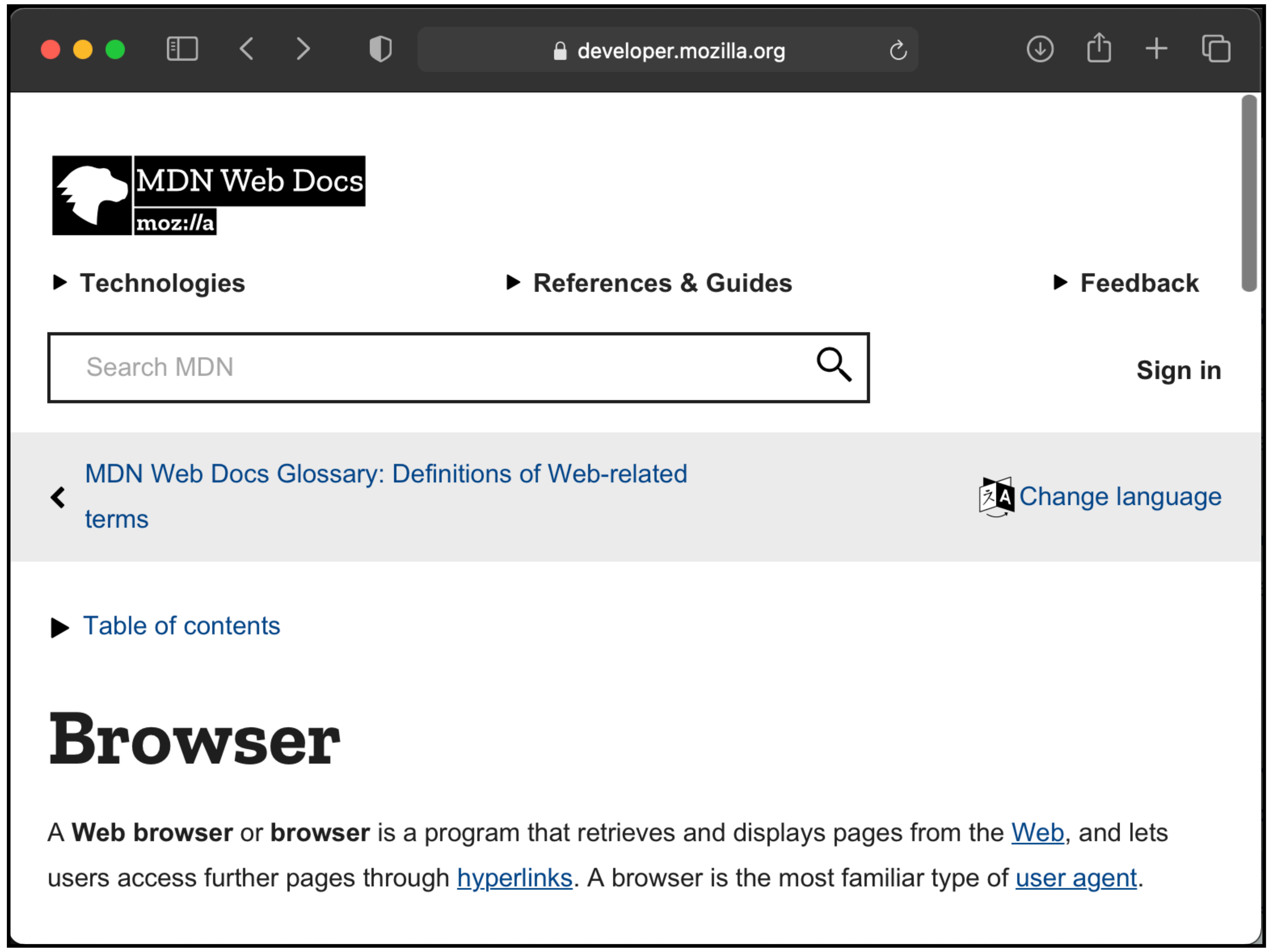
R

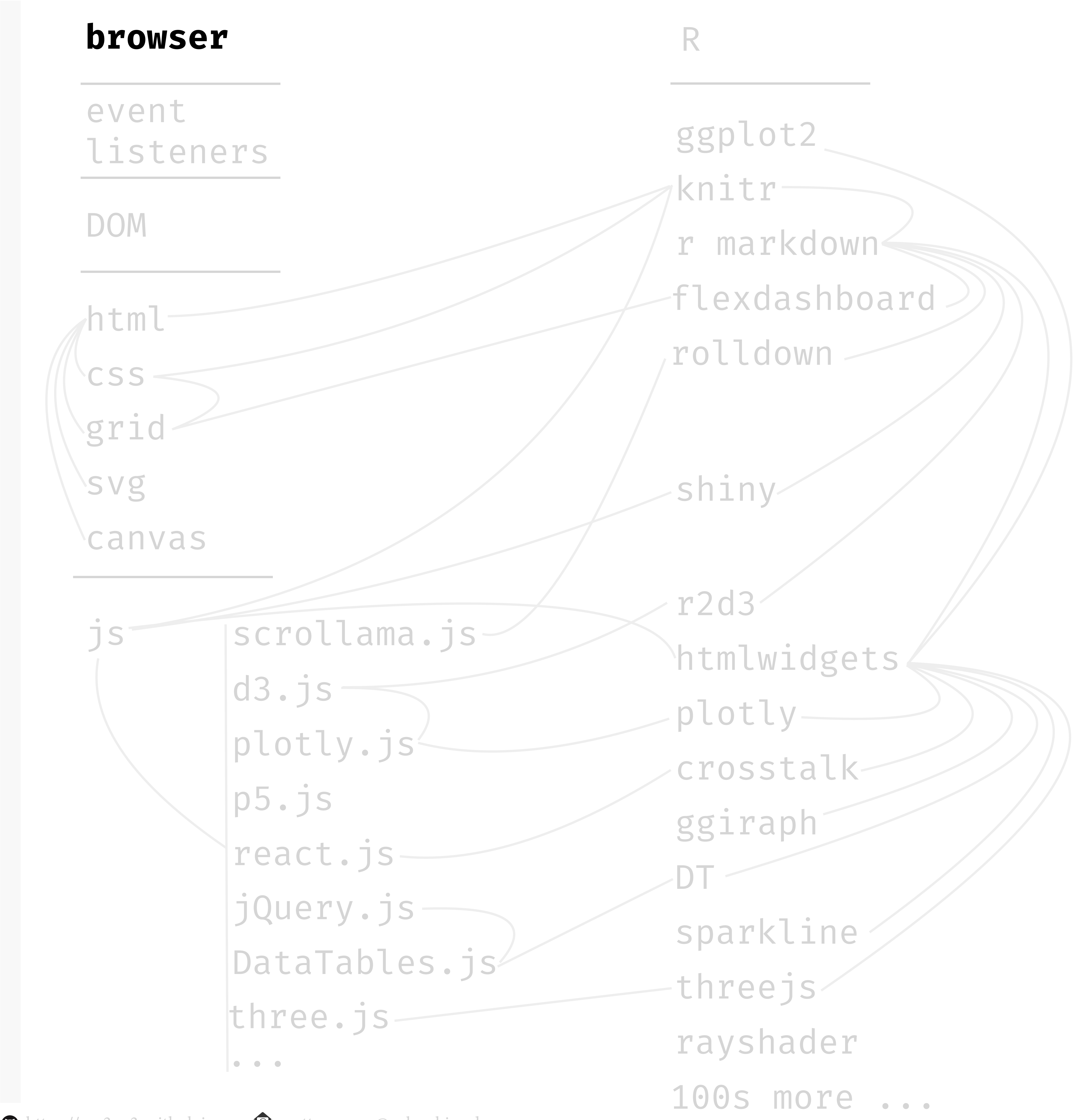
ggplot2
knitr
r markdown
flexdashboard
rolldown

shiny

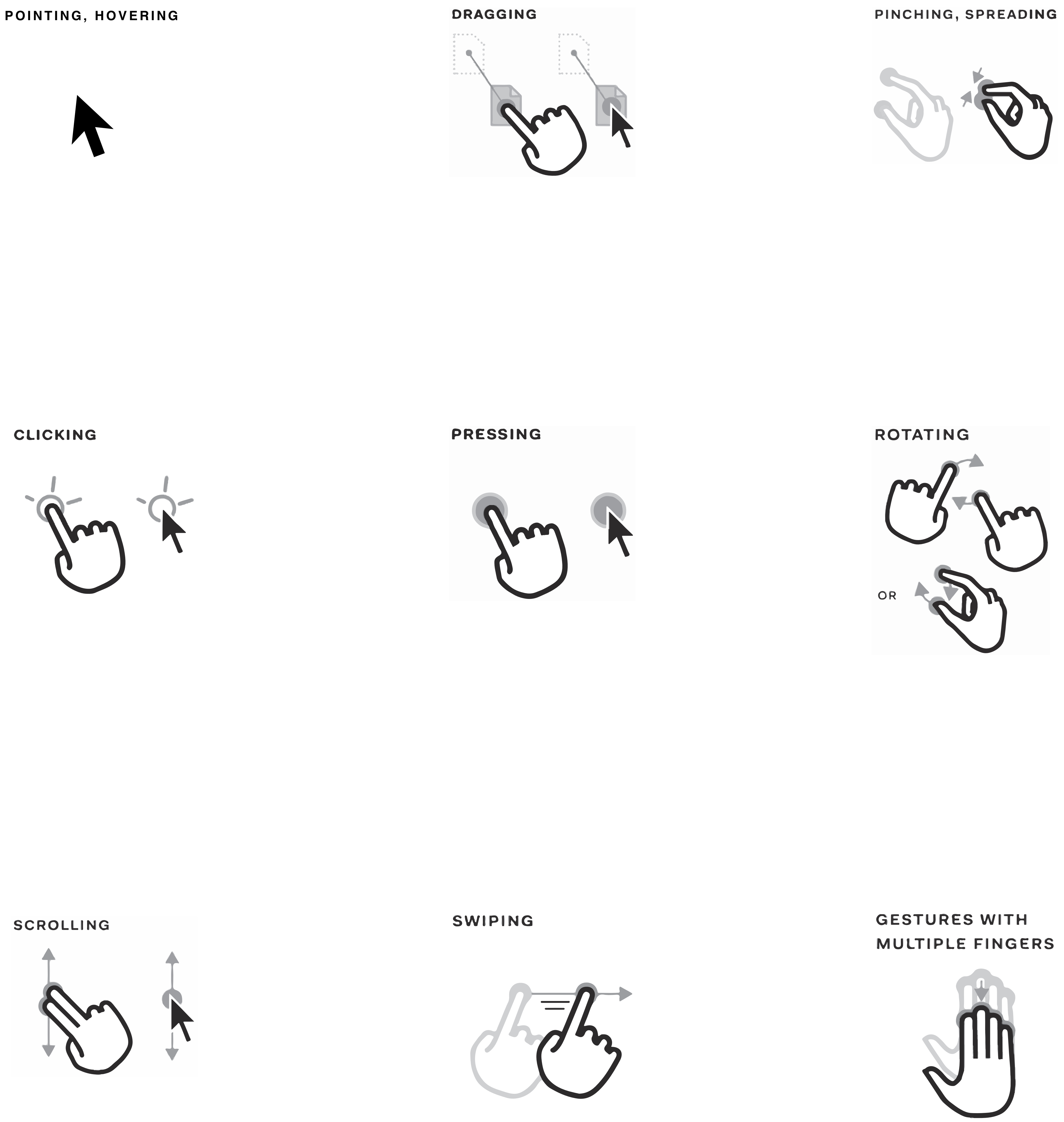
r2d3
htmlwidgets
plotly
crosstalk
ggiraph
DT
sparkline
threejs
rayshader
100s more ...

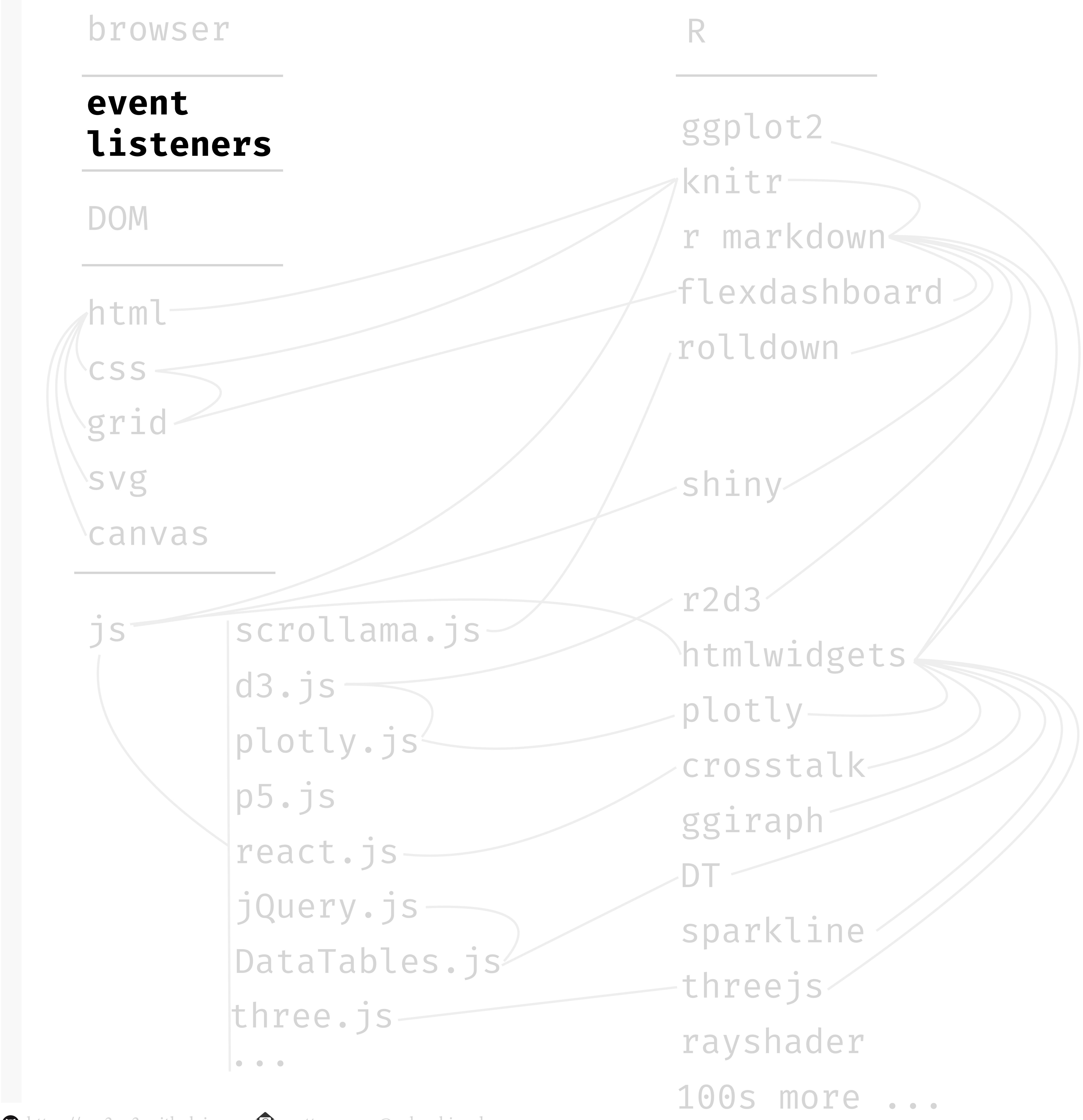
interactive technology stack, browsers parse various code to render content and respond to actions





interactive technology stack, **actions trigger events, for which page elements can be bound to listen**



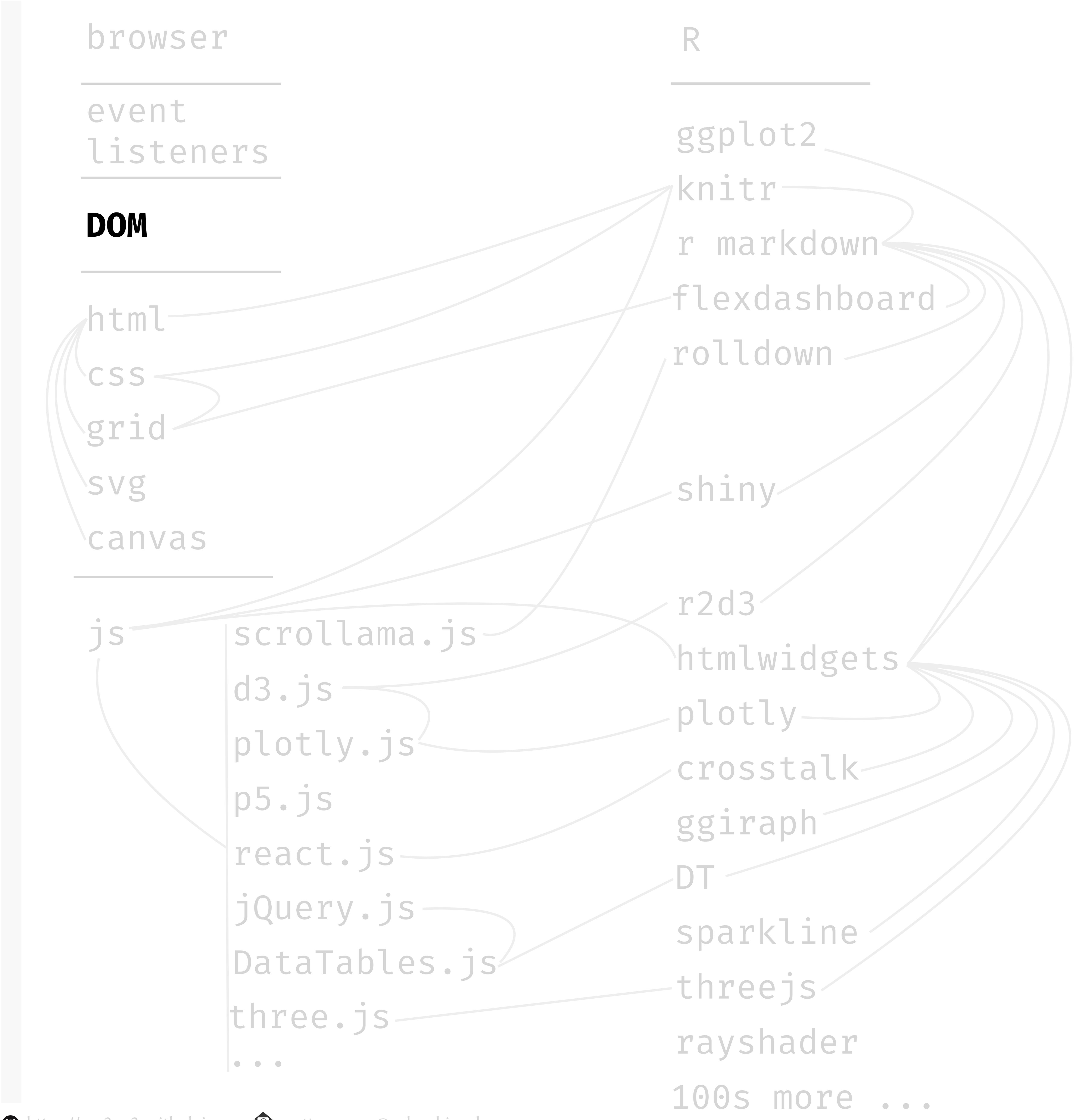


 a web page includes several languages, each has a purpose

web page structure

(Interactive) web **pages** all begin and end with `<html>` and `</html>` respectively. contain a **head** and **body**. Content between `<body>` and `</body>` is shown inside the main browser window Before the `<body>` element you will often see a `<head>` element. This contains information *about* the page, rather than information that is shown within the main part of the browser window. You will usually find a `<title>` element and `<script>` (not shown below) element(s) inside the `<head>` element. Notice how tag enclosures create a *tree-like structure* we can traverse — that's the Document Object Model, or **DOM**.

```
<html>
  <head>
    <title>This is the Title of the Page</title>
  </head>
  <body>
    <h1>This is in the Body of the Page</h1>
      <p>Anything within the body of a web page is
         displayed in the main browser window.</p>
  </body>
</html>
```



 place content in `html` elements, a content layer

`html` elements

Added to the content of a **page** to describe its structure. An element consists of an *opening* and *closing* **tag** and its **content**. Opening tags can carry **attributes**.

The `<p></p>` below instructs the **browser** to structure the content as a paragraph. There are **many** pre-defined **tag types** and attributes, and we can define our own.

OPENING TAG

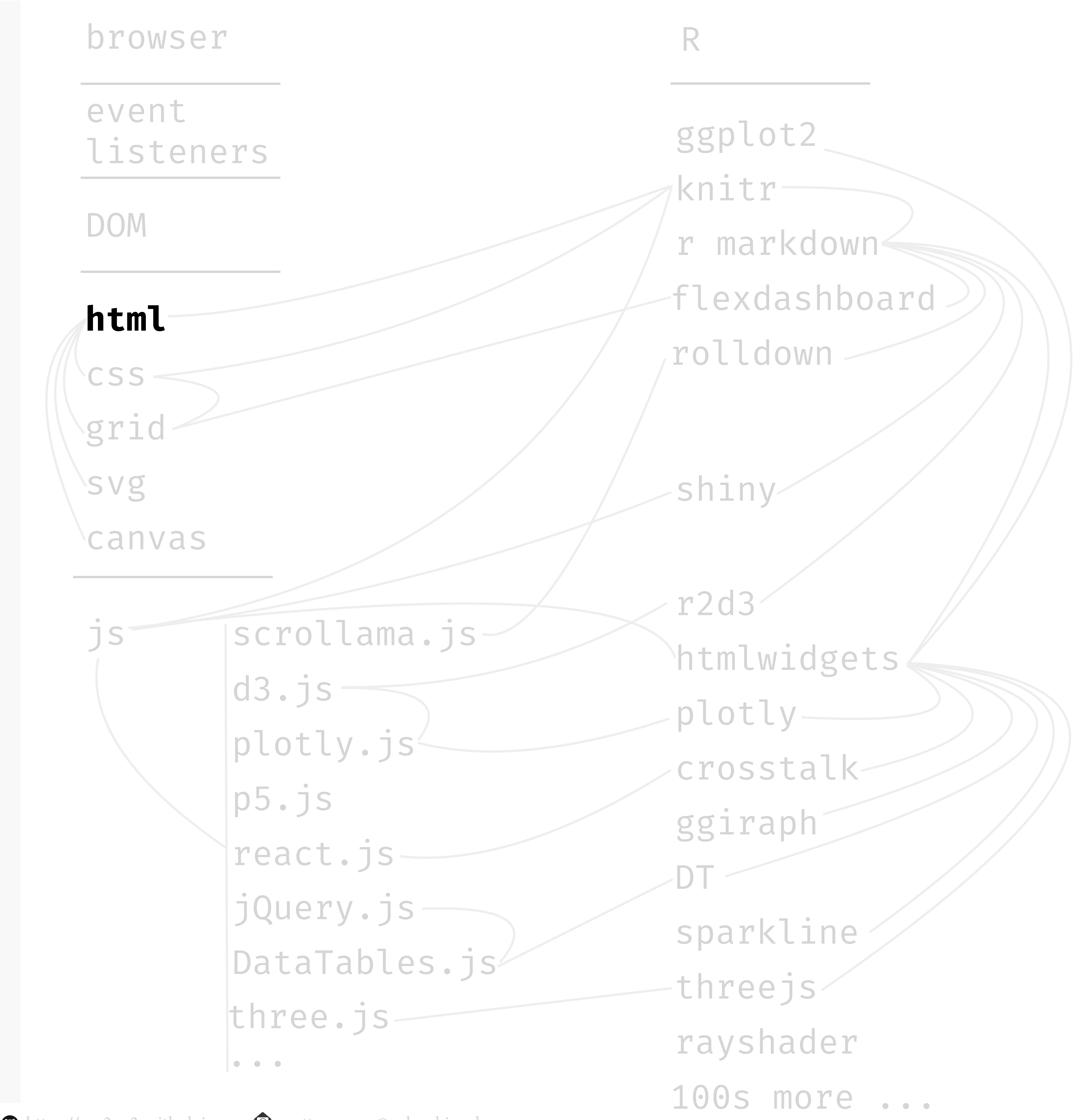
CLOSING TAG

```
<p class="cycling_team">Education First</p>
```

ATTRIBUTE NAME

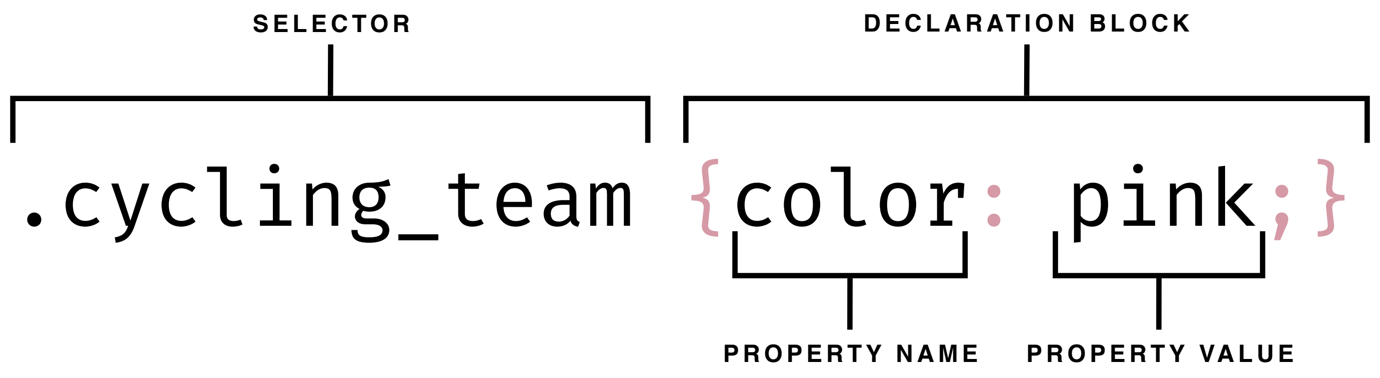
ATTRIBUTE VALUE

CONTENT

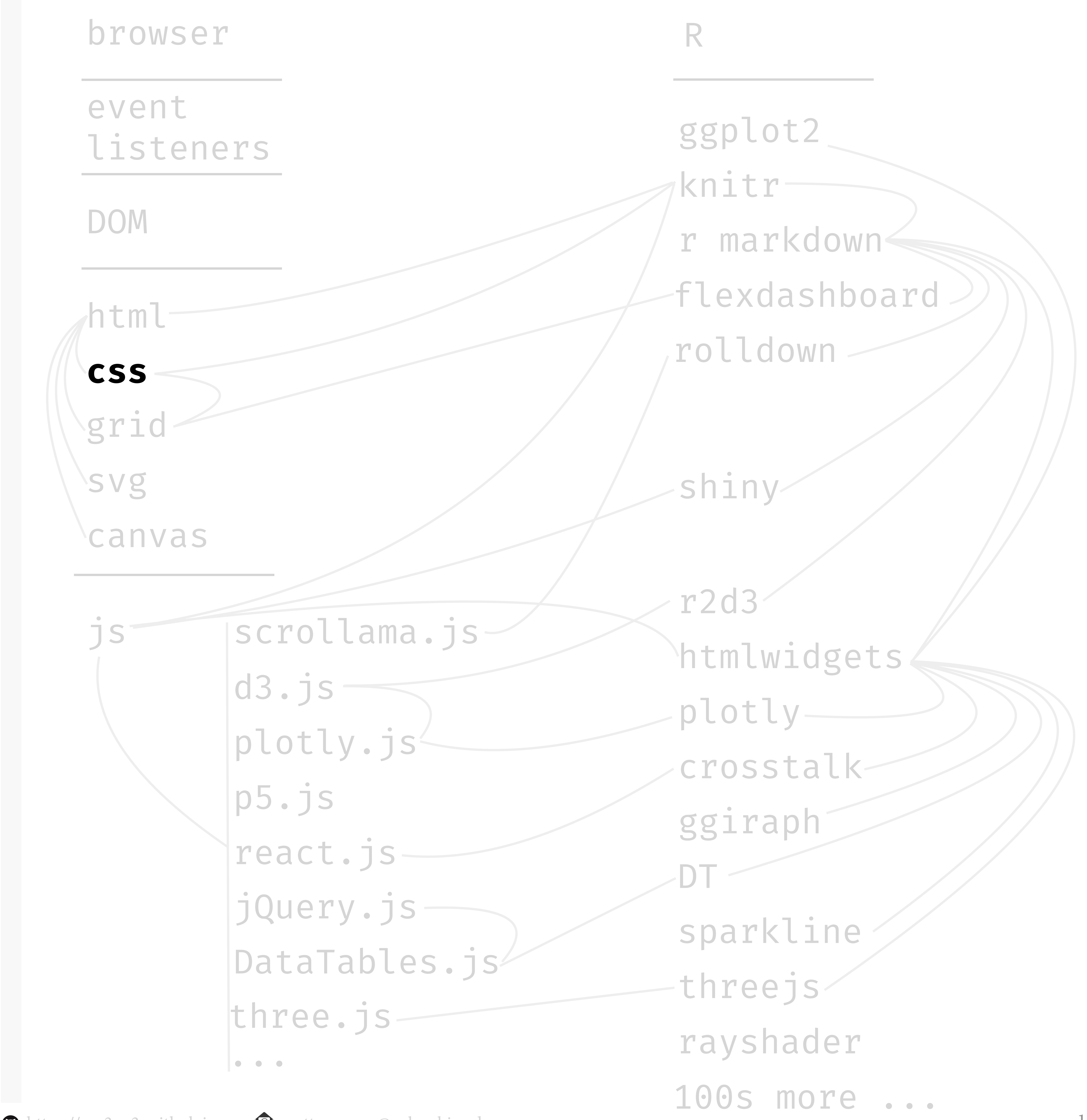


css rules

Indicates how the contents of one or more elements should be displayed in the browser. Each rule has a selector and a declaration block. The **selector** indicates to which element(s) the rule applies. Each **declaration** block specifies one or more properties and corresponding values.Below, applying the **class** `.cycling_team` to a tag as an attribute, it will **color** the text a **pink** hue. CSS rules are specified within `<style>` tags.




```
<style>
  .cycling_team {
    color: pink;
  }
</style>
```

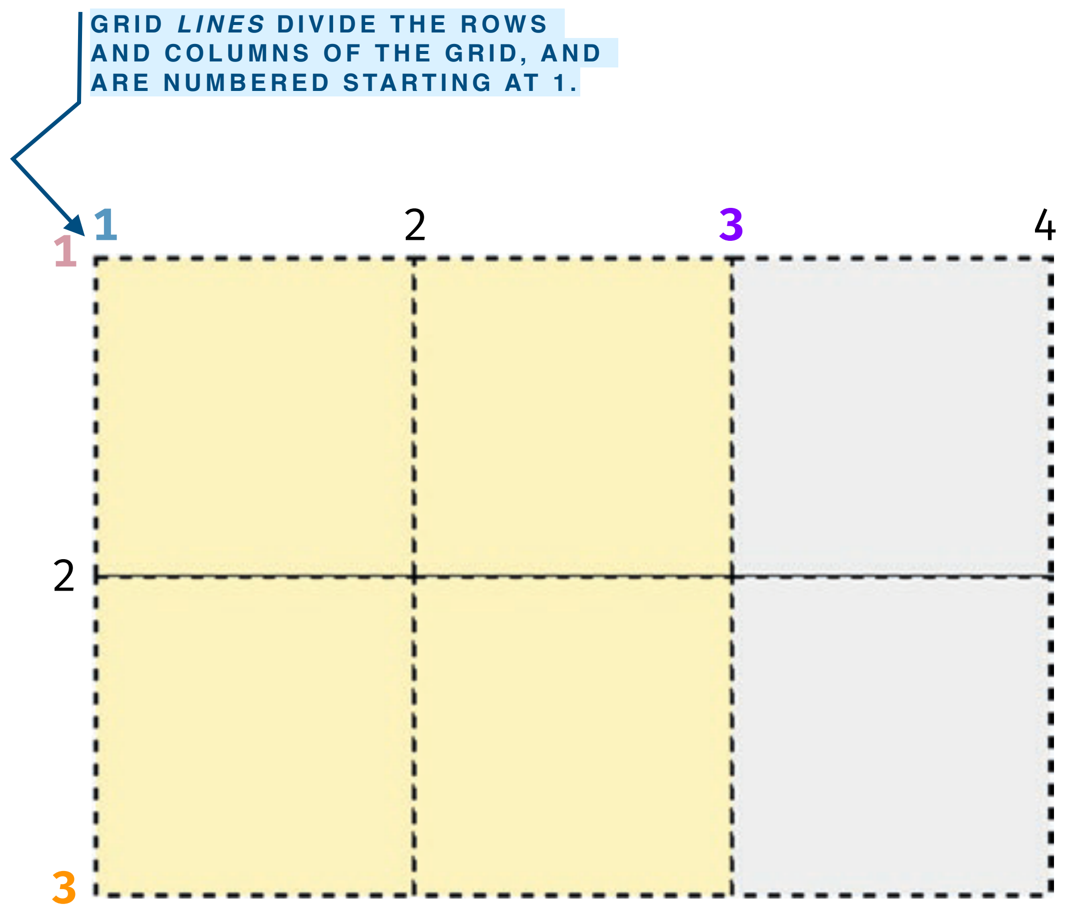


`css` grid

We've discussed and practiced using **grids** earlier in the semester to help us organize text and data graphics for memos, proposals, and information graphics. The `html` language includes grids we can specify using **tags**. Below, we define a **class** `.gridlayout` and in that specify `{display: grid;}` and related properties. Then, we use our `class` attributes in **divider** tags `<div></div>` to format the content. The example below displays a 2 x 3 grid of **cells**, each with a size specified and placed in row major order.

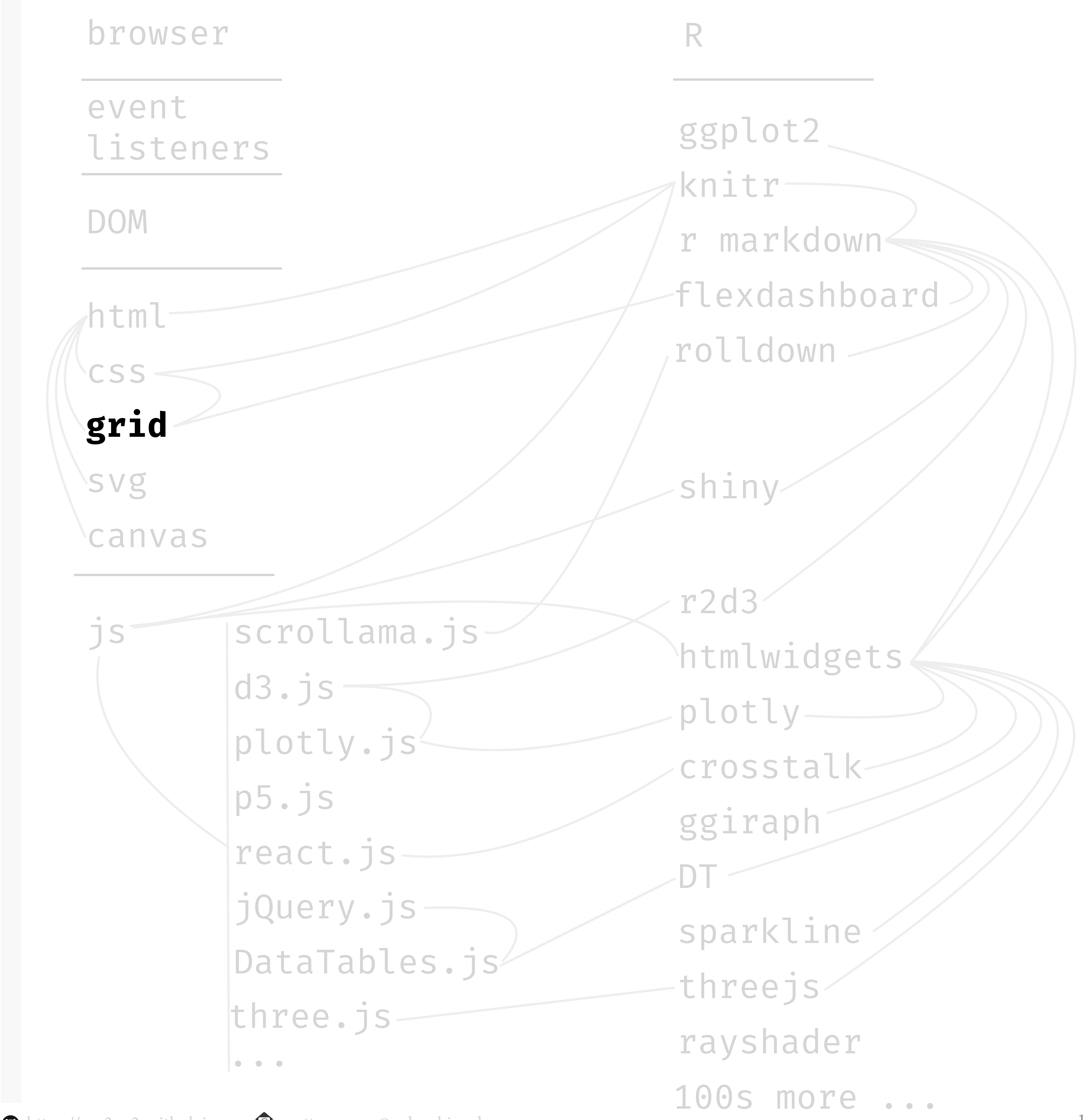
WE PLACE OUR CLASSES FOR THE GRID BETWEEN THE TWO CSS <style> TAGS.

```
<style>
  .gridlayout {
    display: grid;
    grid-template-columns: 1fr 1fr;
    grid-template-rows: 5rem 5rem;
    gap: 5px;
  }
  .item {
    background: lightgray;
    text-align: center;
  }
  .area {
    grid-column: 1 / 3;
    grid-row: 1 / 3;
    background: lightyellow;
    text-align: center;
  }
</style>
```



TO ADD CONTENT, WE PLACE OUR CONTENT BETWEEN <div> TAGS, AND FORMAT USING OUR CLASSES WE DEFINED.

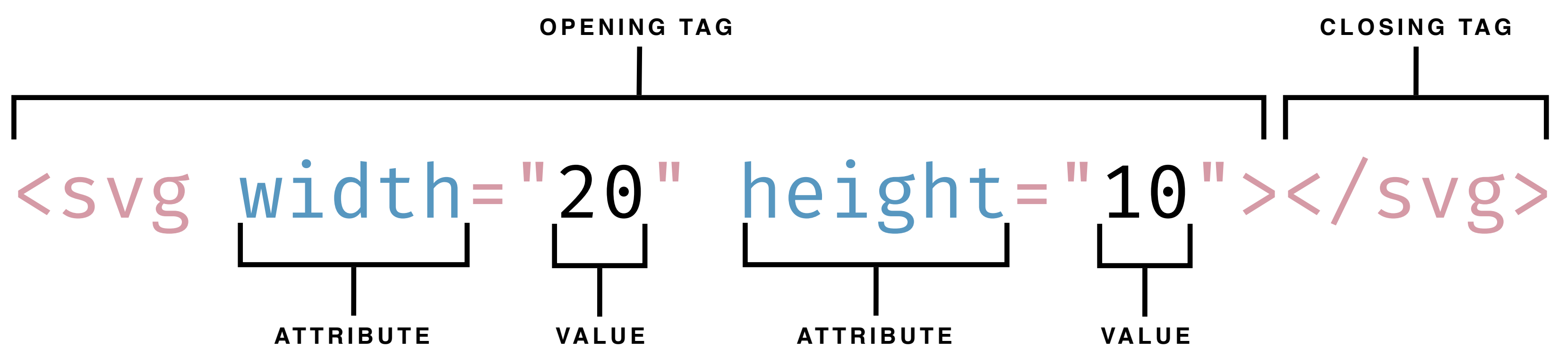
```
<div class="gridlayout">
  <div class="area"></div>
  <div class="item"></div>
  <div class="item"></div>
</div>
```

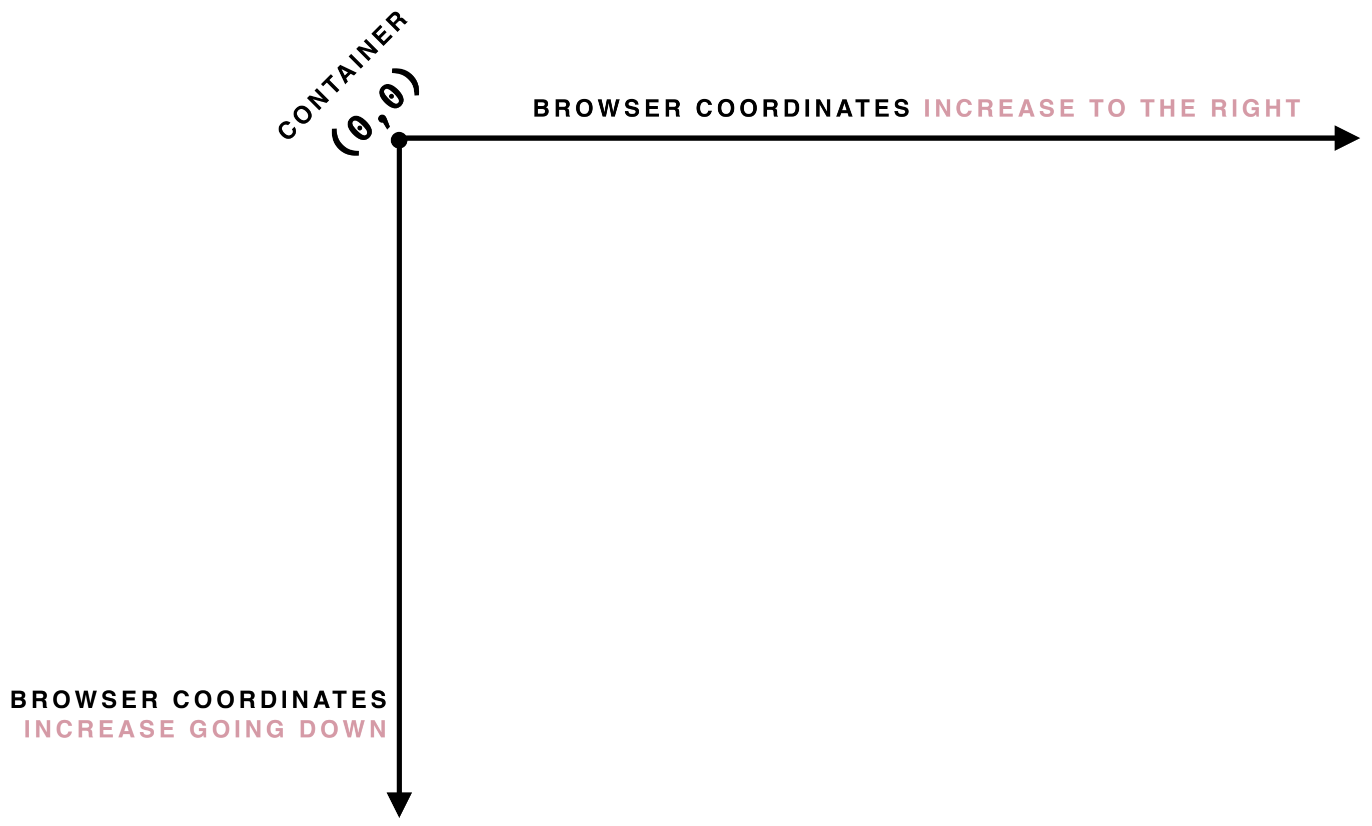


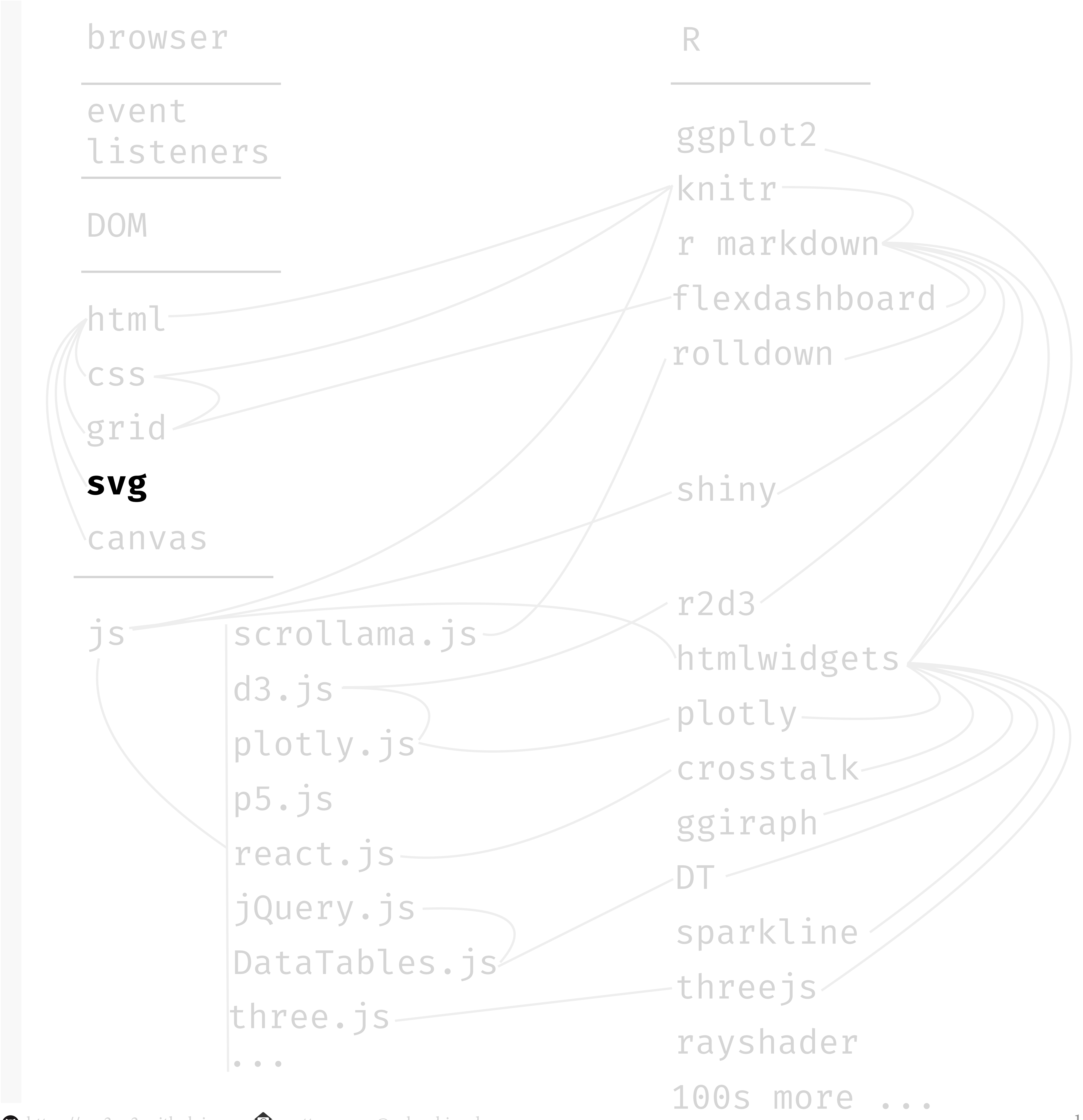
svg

Scalable vector graphics — **svg** — are human-readable descriptions of **shapes** or **paths** that the browser can display. As we've discussed, *enlarging* **vector** graphics, unlike raster-based graphics, will not reduce **resolution**. Together these paths and shapes comprise a graphic.

We put them in the `html` document body between **svg** `<svg>` and `</svg>` tags. Shapes I commonly use include the circle `<circle>`, rectangle `<rect>`, text `<text>`, path `<path>`, and group `<g>`. We can edit vector graphic shapes using software like Adobe Illustrator or Inkscape, too.



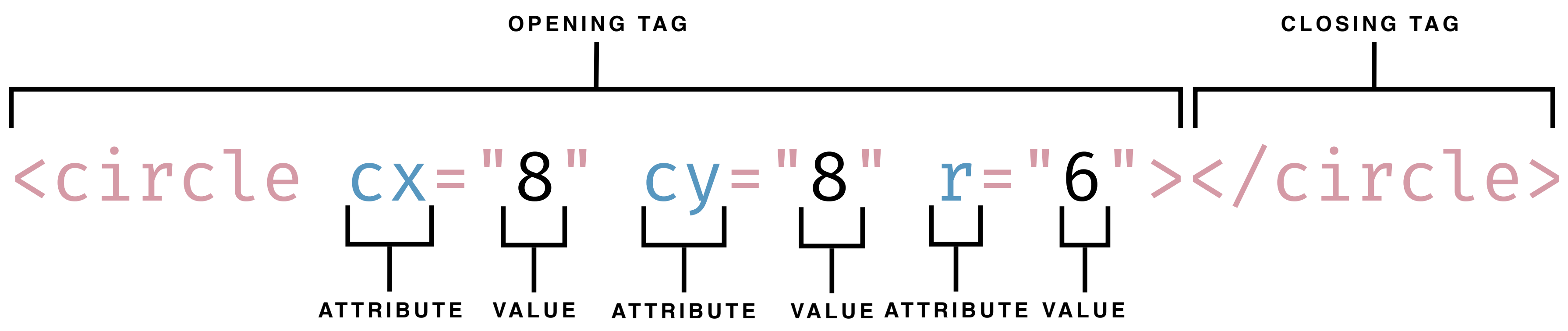


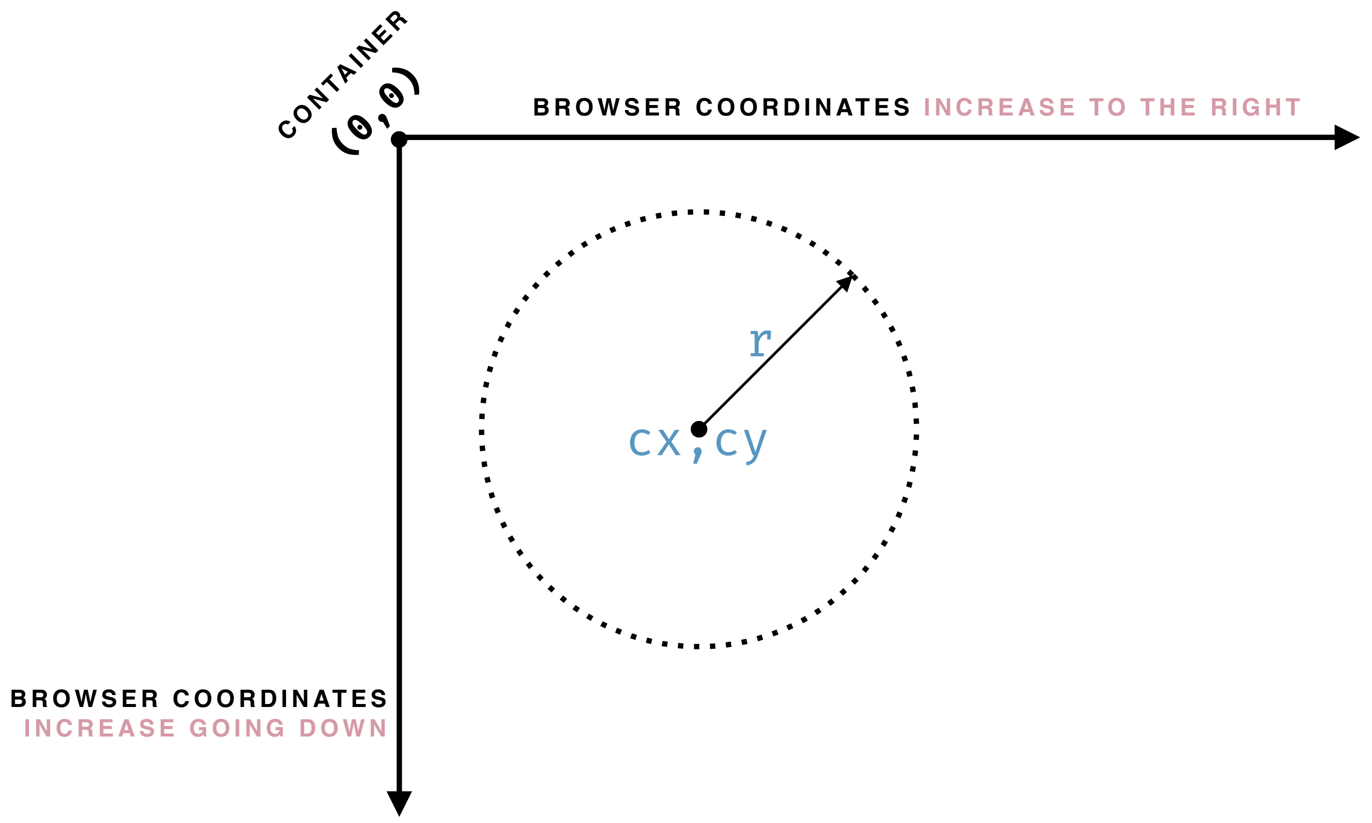


svg

Scalable vector graphics — **svg** — are human-readable descriptions of **shapes** or **paths** that the browser can display. As we've discussed, *enlarging* **vector** graphics, unlike raster-based graphics, will not reduce **resolution**. Together these paths and shapes comprise a graphic.

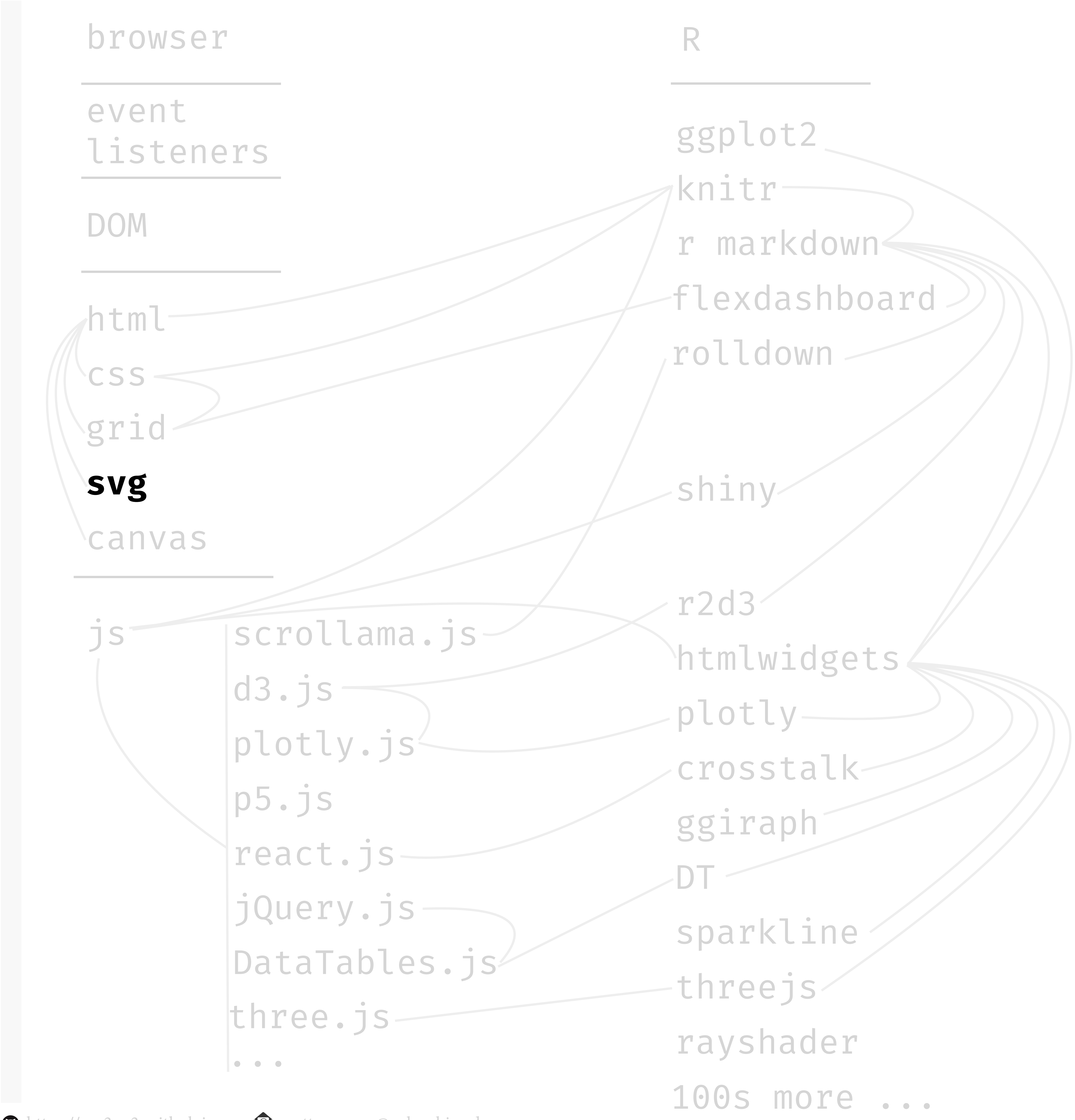
We put them in the `html` document body between svg `<svg>` and `</svg>` tags. Shapes I commonly use include the **circle** `<circle>`, rectangle `<rect>`, text `<text>`, path `<path>`, and group `<g>`. We can edit vector graphic shapes using software like Adobe Illustrator or Inkscape, too.





OTHER COMMON SHAPE ATTRIBUTES

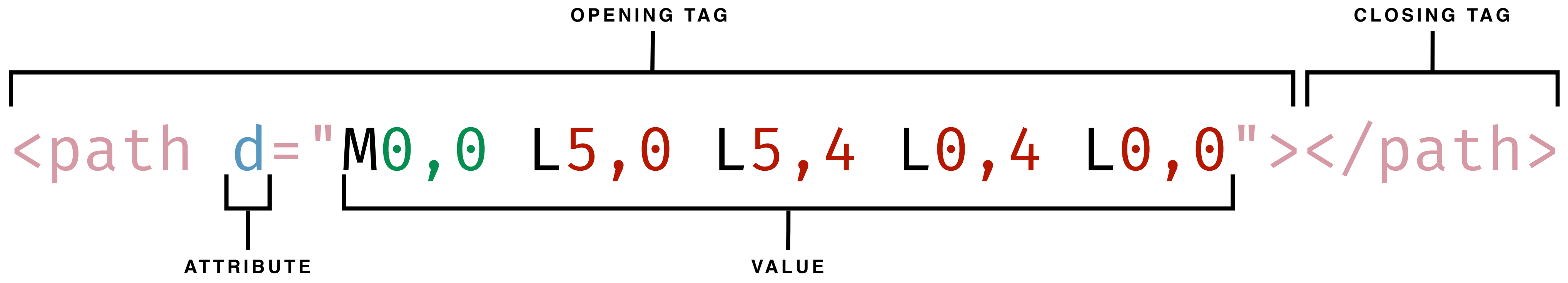
```
stroke
stroke-width
stroke-opacity

fill
fill-color
fill-opacity
```



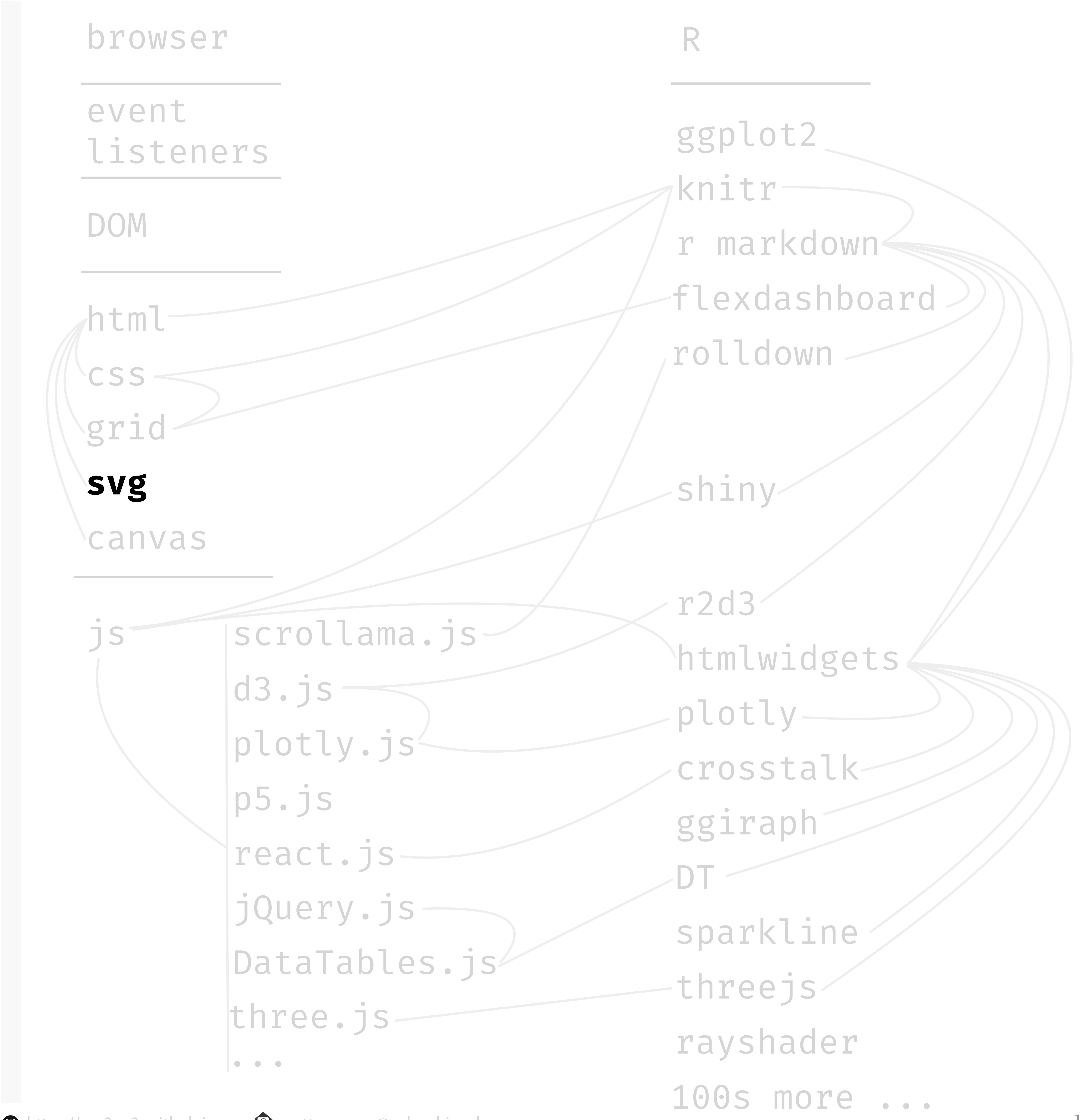
svg

Scalable vector graphics — **svg** — are human-readable descriptions of **shapes** or **paths** that the browser can display. As we've discussed, *enlarging* **vector** graphics, unlike raster-based graphics, will not reduce **resolution**. Together these paths and shapes comprise a graphic.

We put them in the html document body between svg <svg> and </svg> tags. Shapes I commonly use include the circle <circle>, rectangle <rect>, text <text>, **path** <path>, and group <g>. We can edit vector graphic shapes using software like Adobe Illustrator or Inkscape, too.



COMMAND	SYNTAX	MEANING
MOVE TO	Mx,y	location coordinate x, y where the drawing starts.
LINE TO	Lx,y	draw straight path from previous coordinate x, y to this coordinate x, y.
CURVE TO	Cx,y x,y x,y	draw curve path from previous coordinate x, y using two control points x, y and x,y to this coordinate x, y.



interactive technology stack, **draw pixels within `canvas` tags, a content layer**

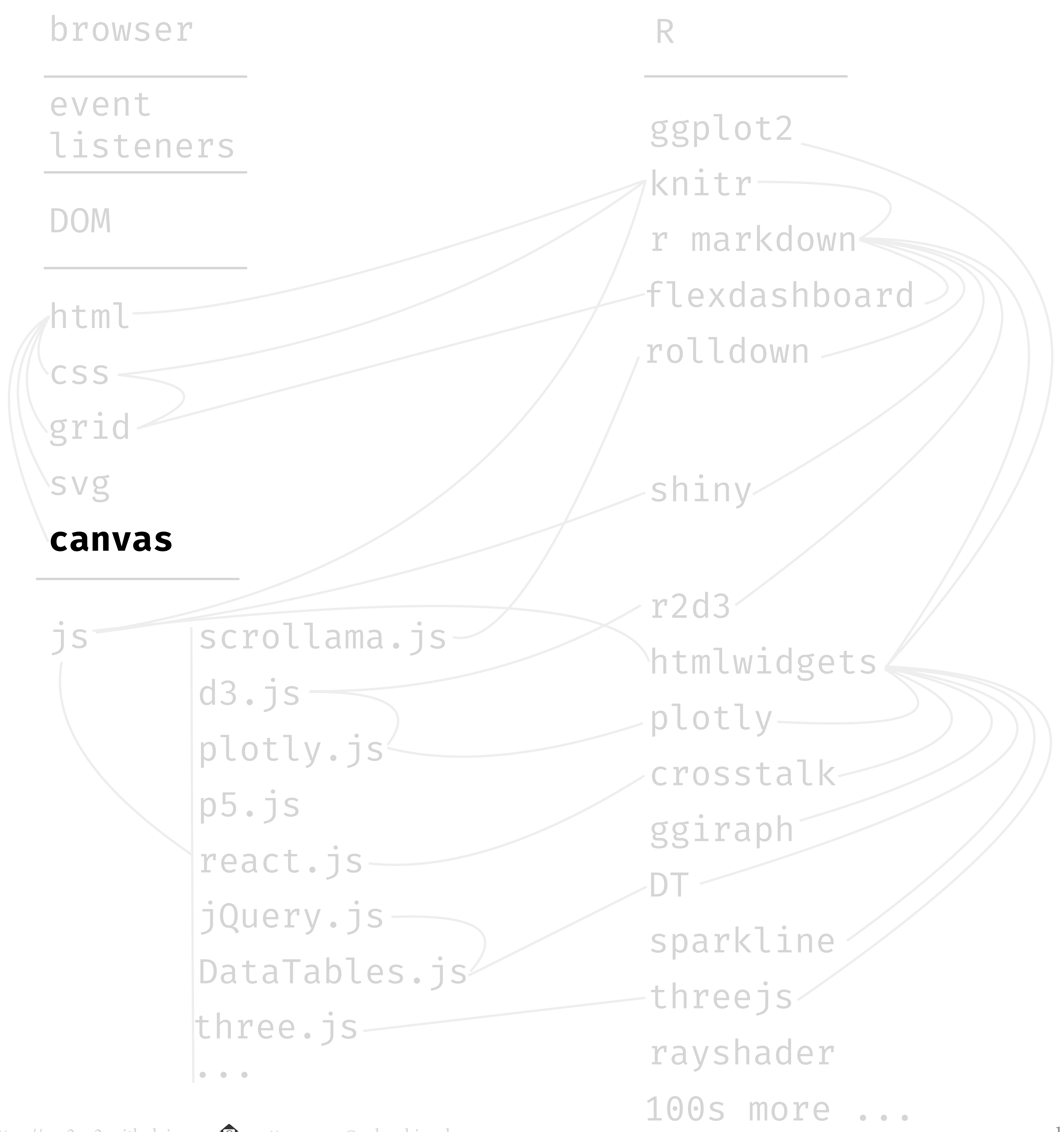
`canvas`

When **performance** drawing `svg` shapes becomes an issue—which may occur on slower computers with 1,000 to 10,000 shapes, more with today's computers—we gain performance by switching to **raster** graphics. For raster graphics, we draw **pixels** on **canvas**, which we specify within `html` using the `<canvas></canvas>` tag. From pixels, we cannot select shapes or paths like we can with `svg` graphics, and **resolution drops** upon **zooming** into the canvas. To edit rasters, we're better off using something like Photoshop.

OPENING TAG — CLOSING TAG

```
<canvas id="c1" width="20" height="10"></canvas>
```

ATTRIBUTE — VALUE — ATTRIBUTE — VALUE — ATTRIBUTE — VALUE



 respond to events by changing content or style with `js`, a behavior layer

JavaScript

We can ***bind* elements** to **events** that, upon happening, ***trigger javascript code***, which in turn can modify content: html elements and attributes, svg or canvas, or css styles. Really it **can modify anything in the DOM**. As with R packages that abstract and ease our application of specialized functionality, easing the burden of writing code, many **javascript libraries** are available to do the same. Those listed to the right are particularly important for interactive data visualization, but many more not listed are also available.

```
element.onevent = functionName;
```

- ELEMENT — DOM element node to target
- EVENT — Event bound to node(s) preceded by word "on".
- CODE — Name of function to call with no parenthesis following it.

```
function functionName() {
  // code to do something
}
```

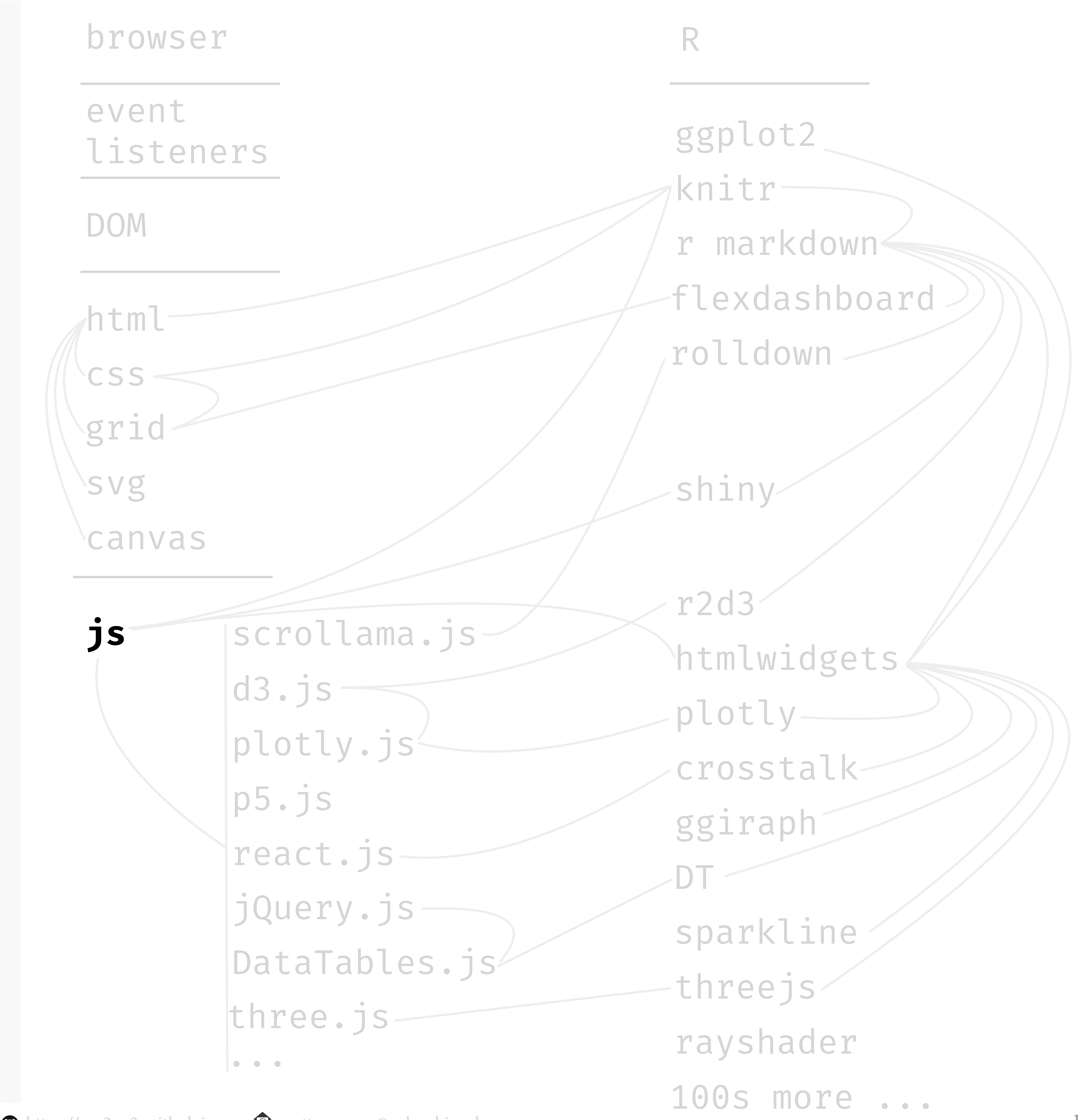
THE CODE STARTS BY DEFINING THE NAMED FUNCTION.

A REFERENCE TO THE DOM ELEMENT IS OFTEN STORED IN A VARIABLE.

```
var el = document.getElementById('element');
el.on_______ = functionName;
```

THE EVENT NAME IS PRECEDED BY THE WORD "ON".

THE FUNCTION IS CALLED BY THE EVENT HANDLER ON THE LAST LINE, BUT THE PARENTHESIS ARE OMITTED.



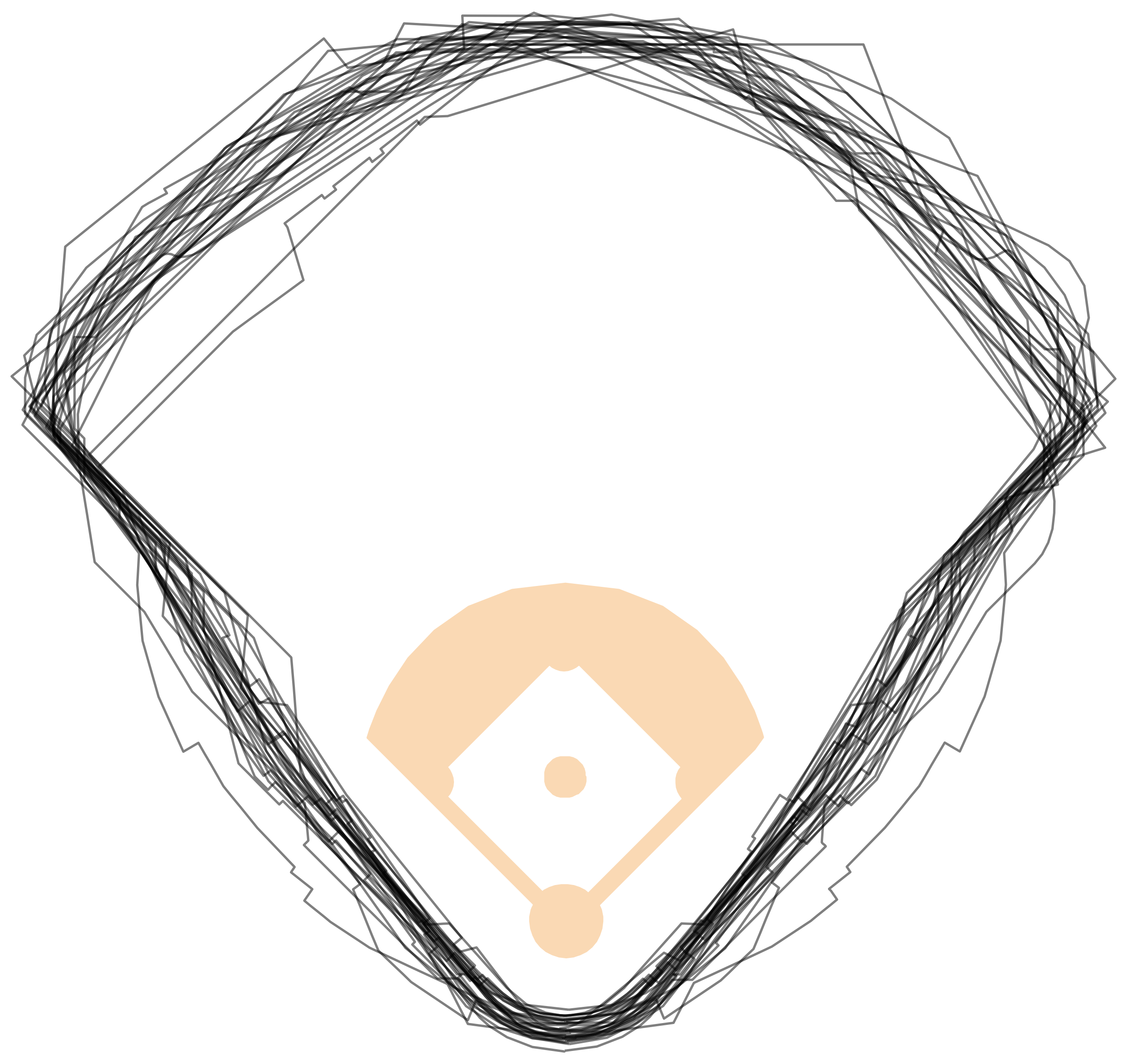
content creation for this interactive technology stack

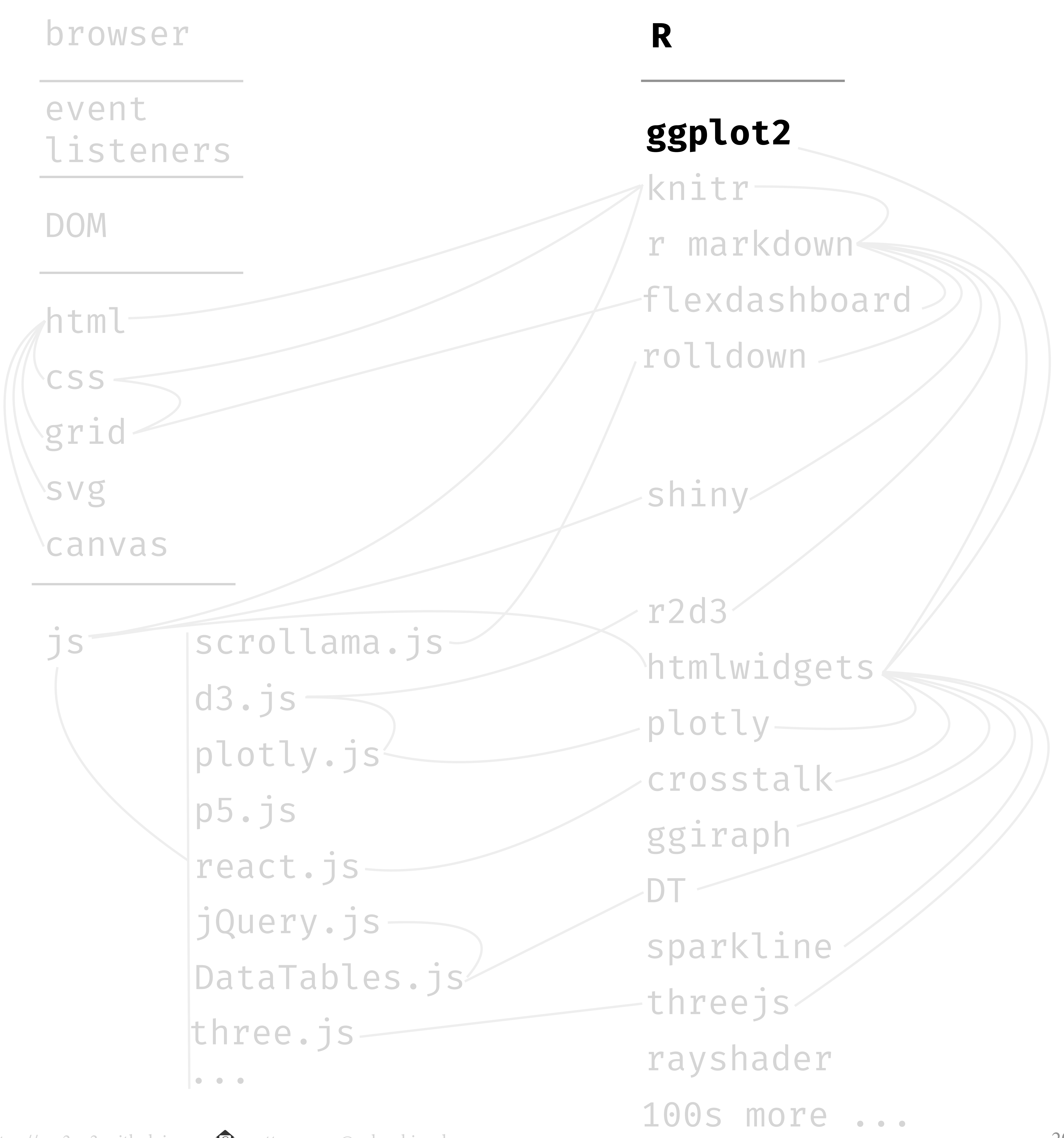
tools for interactive content, **several R packages transform ggplot2 into interactive graphics**

ggplot2

The **grammar of graphics** — implemented in R as ggplot2 — is among the most powerful coding libraries for creating static graphics. We've already seen how to use a complementary package with ggplot2 to add animation: gganimate, a grammar of animated graphics. With similar complementary packages, we can specify **interactivity**. Let's see a static version of a class example, the 30 baseball outfields, then make it interactive using ggiraph.

30 baseball outfields — *static* version





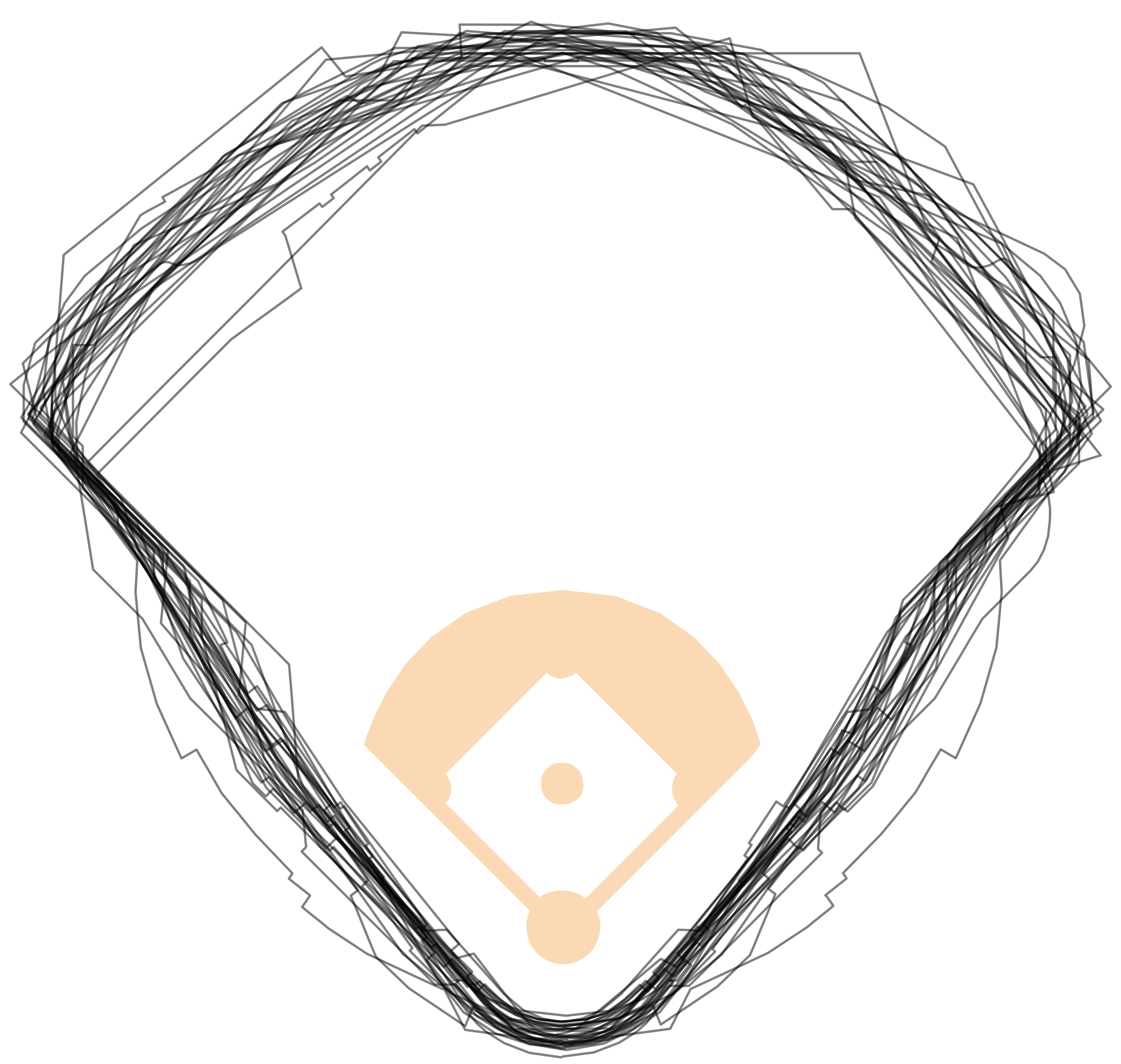
 several R packages transform ggplot2 into interactive graphics

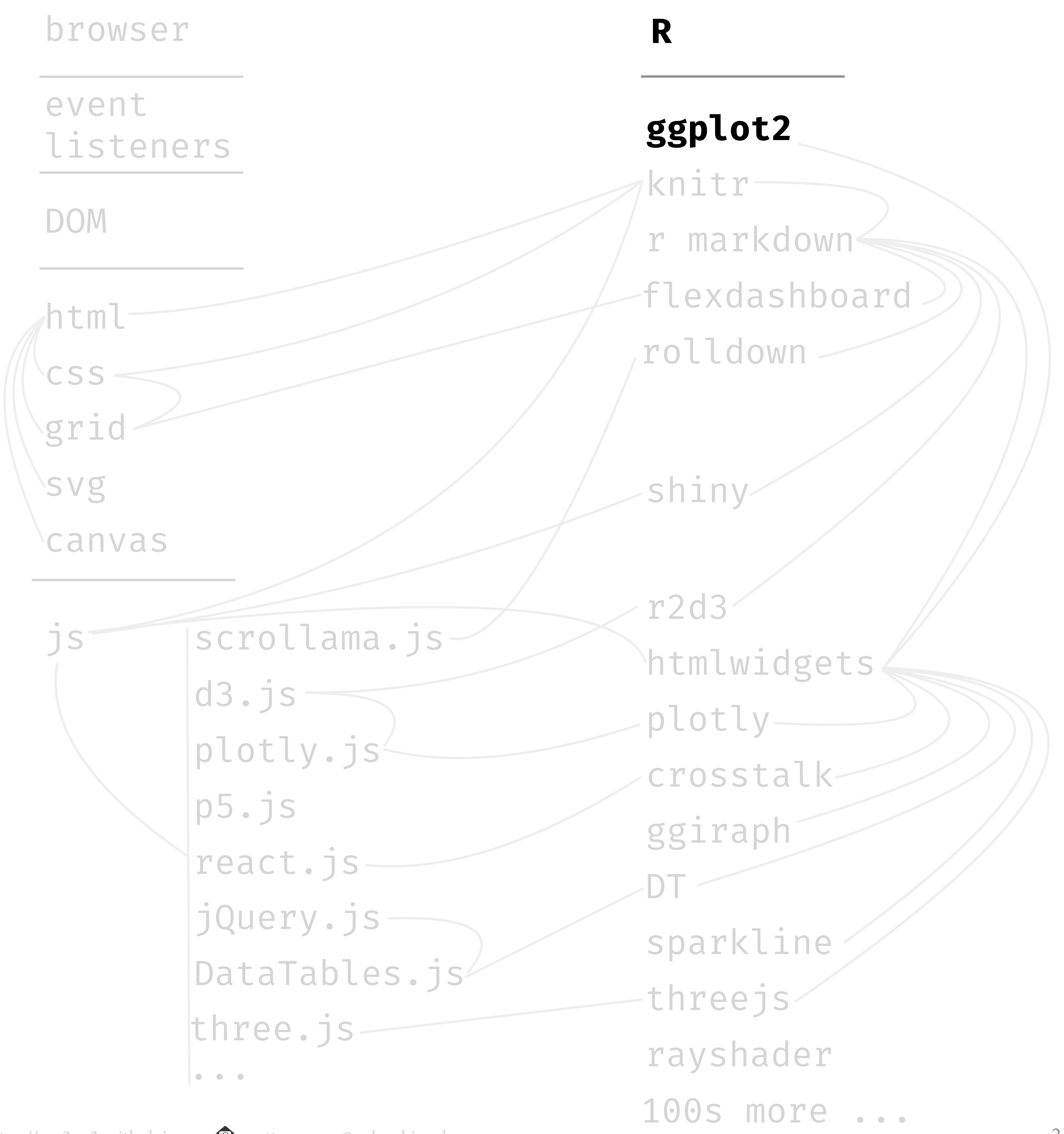
ggplot2

The **grammar of graphics** — implemented in R as ggplot2 — is among the most powerful coding libraries for creating static graphics. We've already seen how to use a complementary package with ggplot2 to add animation: gganimate, a grammar of animated graphics. With similar complementary packages, we can specify **interactivity**. Let's see a static version of a class example, the 30 baseball outfields, then make it interactive using ggiraph.

```
gg_boundaries <-
  ggplot() +
  theme_void() +
  coord_equal() +
  geom_path(
    data = subset(
      fields,
      is_infield == FALSE),
    mapping = aes(
      x = xsh,
      y = ysh,
      group = id),
    color = '#000000',
    alpha = 0.5) +

  geom_polygon(
    data = subset(
      fields,
      is_infield == TRUE),
    mapping = aes(
      x = xsh,
      y = ysh,
      group = id),
    fill = '#FAD9B4',
    color = '#FAD9B4')
```

30 baseball outfields — *static* version





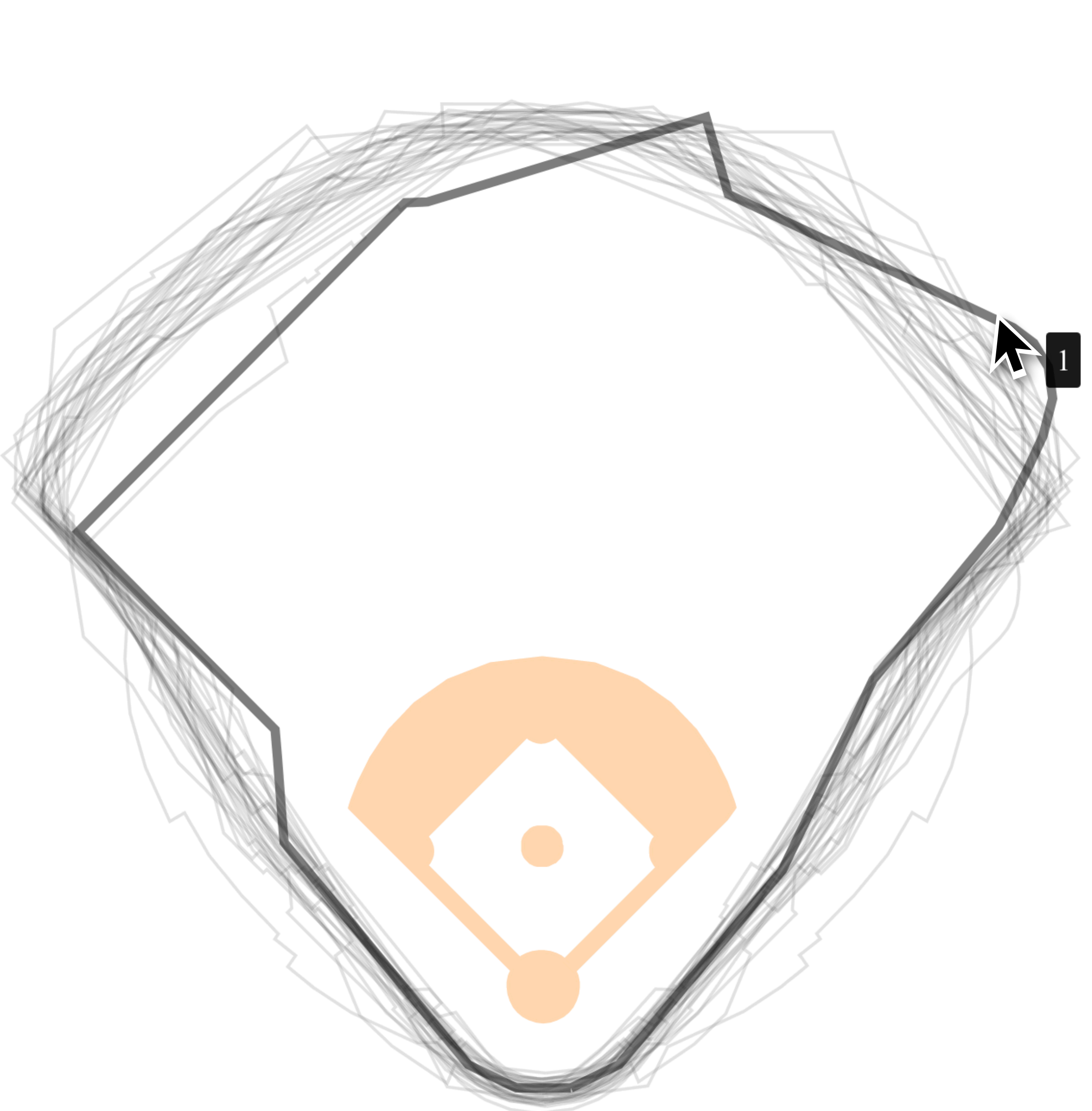
tools for interactive content, several R packages transform ggplot2 into interactive graphics

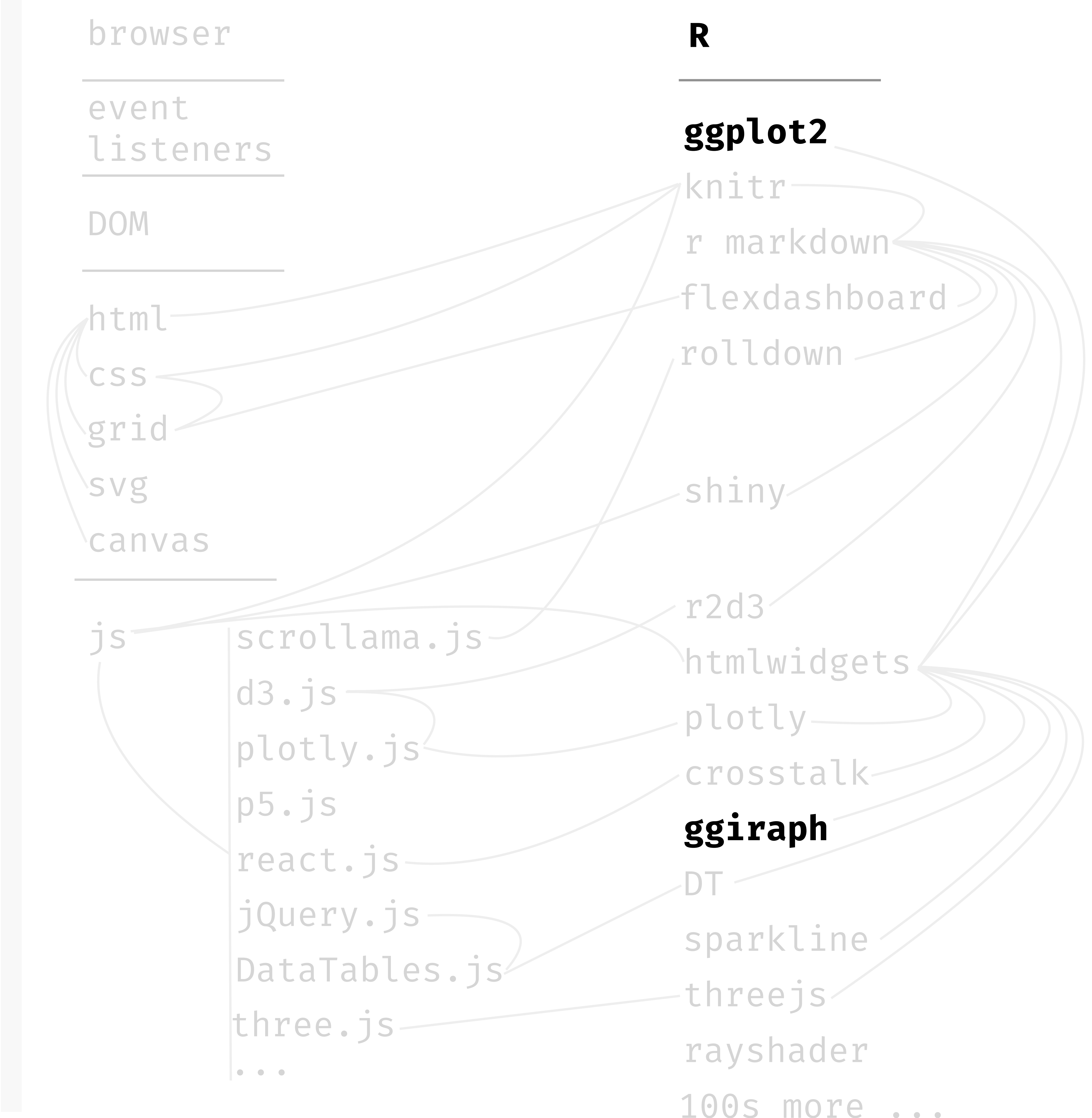
ggplot2 + ggiraph

The **grammar of graphics** — implemented in R as ggplot2 — is among the most powerful coding libraries for creating static graphics. We've already seen how to use a complementary package with ggplot2 to add animation: gganimate, a grammar of animated graphics. With similar complementary packages, we can specify **interactivity**. Let's see a static version of a class example, the 30 baseball outfields, then make it interactive using ggiraph.

```
gg_boundaries <-
  ggplot() +
  theme_void() +
  coord_equal() +
  geom_path_interactive(
    data = subset(
      fields,
      is_infield == FALSE),
    mapping = aes(
      x = xsh,
      y = ysh,
      group = id,
      tooltip = id,
      data_id = id
      ),
    color = '#000000',
    alpha = 0.5) +

  geom_polygon(
    data = subset(
      fields,
      is_infield == TRUE),
    mapping = aes(
      x = xsh,
      y = ysh,
      group = id),
    fill = '#FAD9B4',
    color = '#FAD9B4')

girafe(
  code = print(gg_boundaries),
  options = list(
    opts_hover(
      css = 'stroke-width:3;'),
    opts_hover_inv(
      css = 'stroke-opacity:0.1;')
    )
  )
```

30 baseball outfields — an *interactive* version





 several R packages transform ggplot2 into interactive graphics

ggplot2 + ggiraph

The **grammar of graphics** — implemented in R as ggplot2 — is among the most powerful coding libraries for creating static graphics. We've already seen how to use a complementary package with ggplot2 to add animation: gganimate, a grammar of animated graphics. With similar complementary packages, we can specify **interactivity**. Let's see a static version of a class example, the 30 baseball outfields, then make it interactive using ggiraph.

```
gg_boundaries <-
  ggplot() +
  theme_void() +
  coord_equal() +
  geom_path_interactive(
    data = subset(
      fields,
      is_infield == FALSE),
    mapping = aes(
      x = xsh,
      y = ysh,
      group = id,
      tooltip = id,
      data_id = id
    ),
    color = '#000000',
    alpha = 0.5) +

  geom_polygon(
    data = subset(
      fields,
      is_infield == TRUE),
    mapping = aes(
      x = xsh,
      y = ysh,
      group = id),
    fill = '#FAD9B4',
    color = '#FAD9B4')

girafe(
  code = print(gg_boundaries),
  options = list(
    opts_hover(
      css = 'stroke-width:3;'),
    opts_hover_inv(
      css = 'stroke-opacity:0.1;')
    )
  )
```

NOTICE THE ONE-TO-ONE MATCHUP WITH GGPLOT2 GEOMETRIES

DETAILS-ON-DEMAND: ADD INFORMATION TO SHOW IN THE TOOLTIP, SHOWN WHEN HOVERING

USED TO BIND ELEMENTS TO EVENT LISTENERS

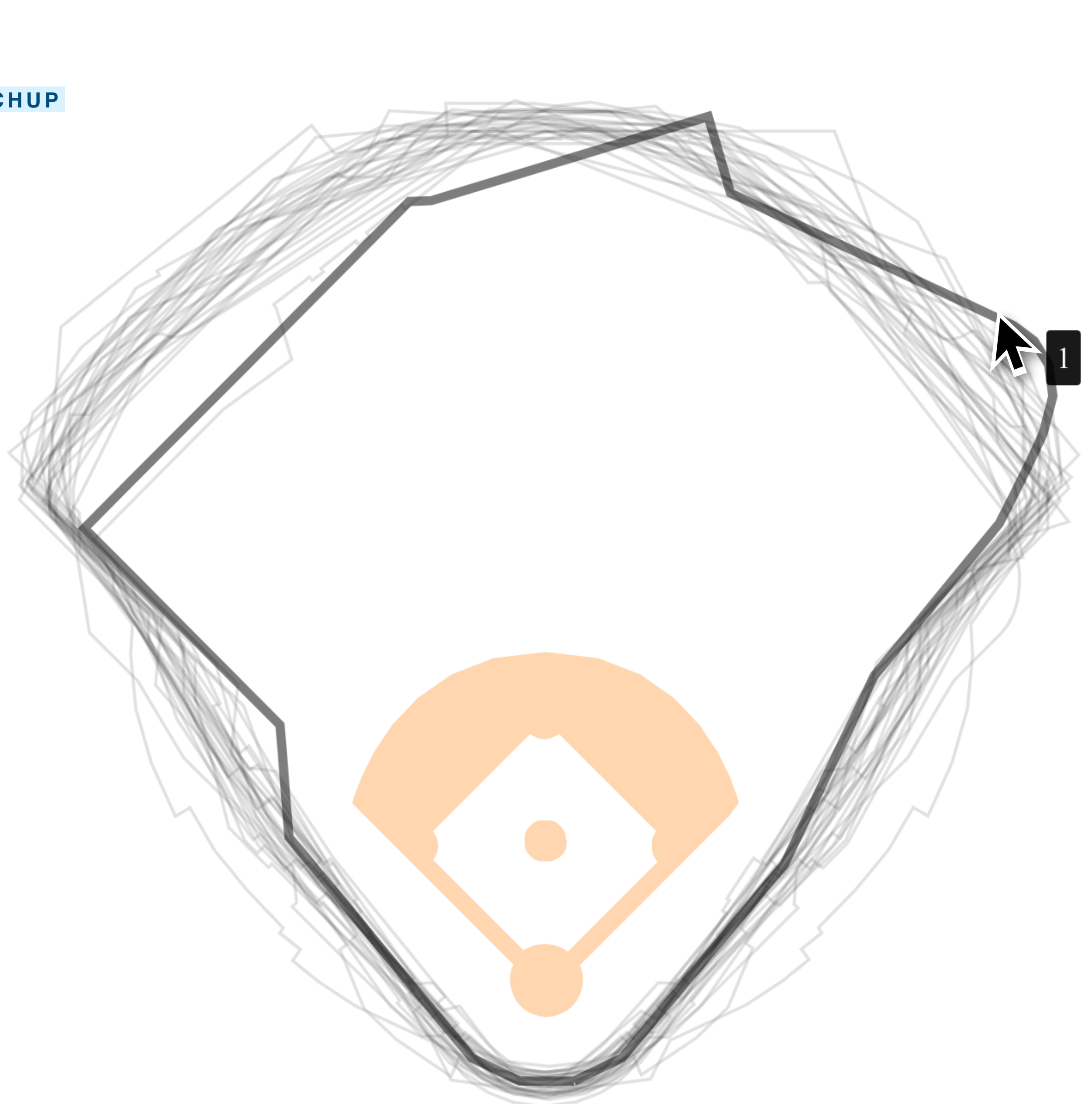
CHANGE PROPERTIES OF THINGS OF SHAPE REACTING TO HOVER

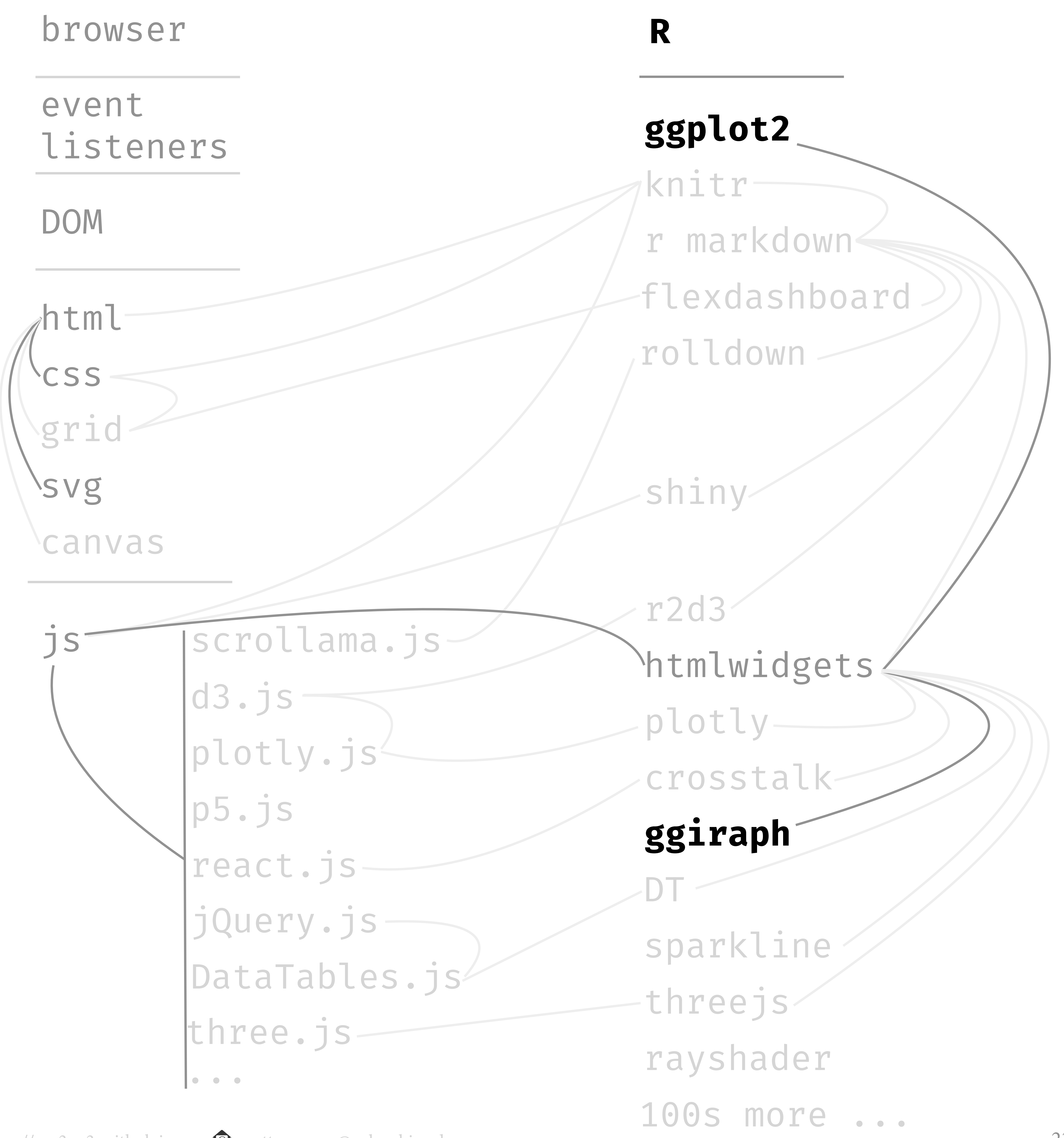
SET CSS PROPERTIES OF SHAPE; HERE STROKE-WIDTH

CHANGE PROPERTIES OF THINGS ELEMENT *NOT* REACTING TO HOVER

SET CSS PROPERTIES OF SHAPE; HERE STROKE-OPACITY

30 baseball outfields — an *interactive* version





ggplot2 + ggiraph

The **grammar of graphics** — implemented in R as ggplot2 — is among the most powerful coding libraries for creating static graphics. We've already seen how to use a complementary package with ggplot2 to add animation: gganimate, a grammar of animated graphics. With similar complementary packages, we can specify **interactivity**. Let's see a static version of a class example, the 30 baseball outfields, then make it interactive using ggiraph.

```
gg_fences <-
  ggplot() +
  theme_void() +
  theme( axis.text.x = element_text() ) +
  coord_equal() +
  scale_x_continuous(
    breaks = c(100, 300, 500),
    labels = c("Left Field",
               "Center Field",
               "Right Field")) +
  geom_path_interactive(
    data = fences,
    mapping = aes(
      x = xsh,
      y = -ysh,
      group = id,
      tooltip = id,
      data_id = id),
    color = 'black',
    alpha = 0.5)

girafe(
  code = print(gg_boundaries / gg_fences),
  options = list(
    opts_hover(
      css = 'stroke-width:3;'),
    opts_hover_inv(
      css = 'stroke-opacity:0.1;')
    )
  )
```

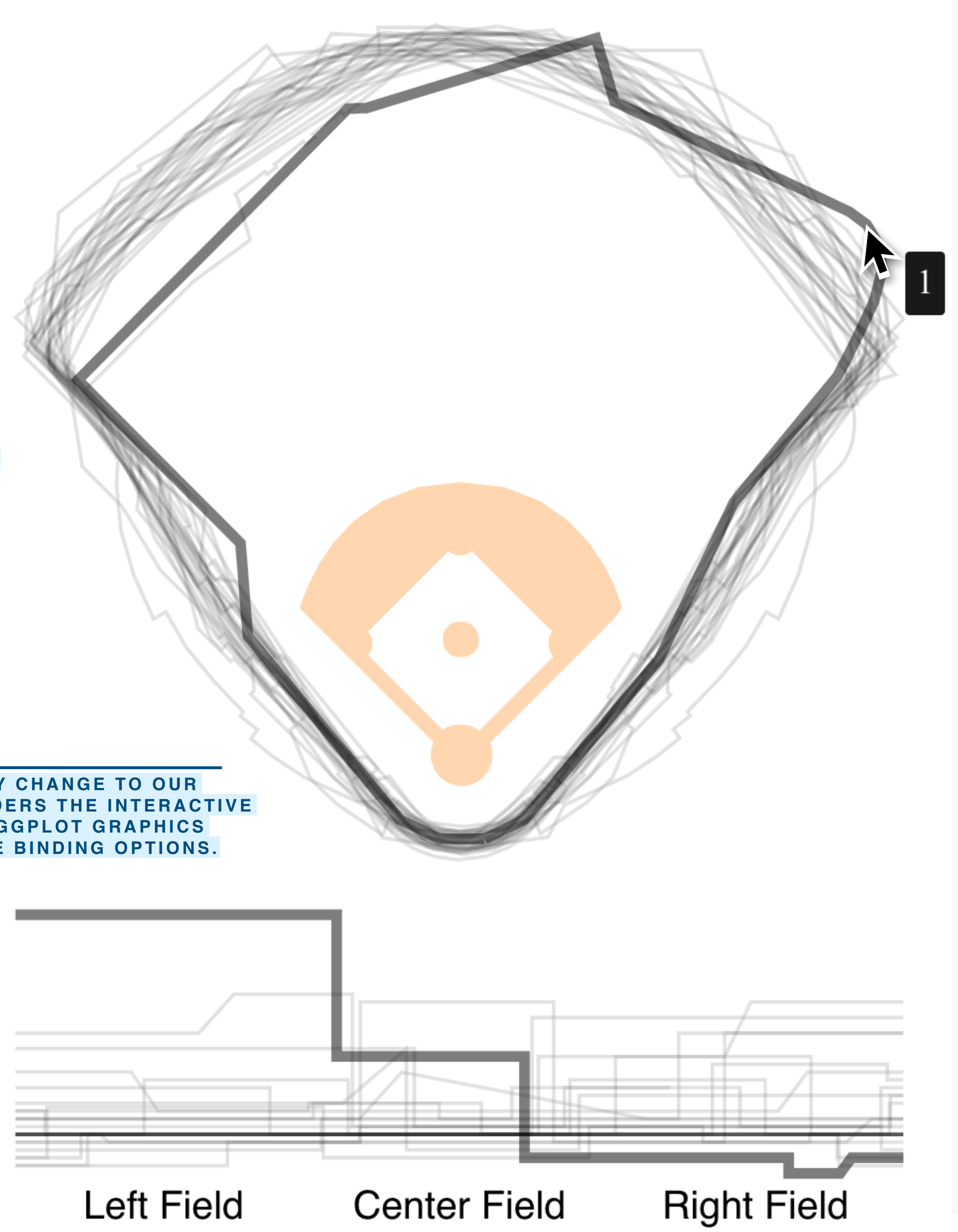
NOTICE THE SAME DATA ID USED HERE AS FOR THE OUTFIELD BOUNDARIES

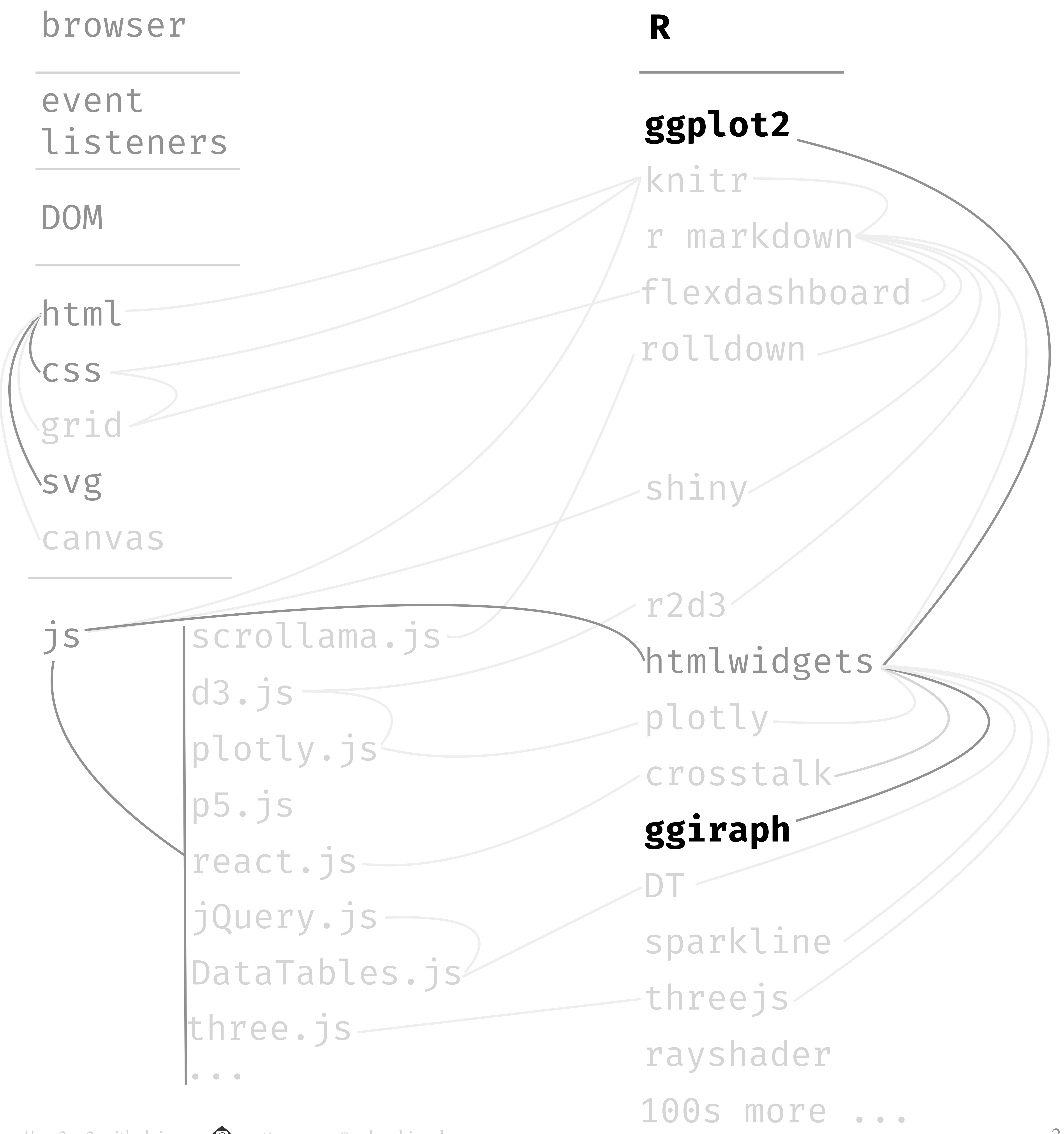
THIS IS THE ONLY CHANGE TO OUR CODE THAT RENDERS THE INTERACTIVE GRAPHIC. BOTH GGPLOT GRAPHICS SHARE THE SAME BINDING OPTIONS.

IN RSTUDIO, WE CAN SAVE THE INTERACTIVE GRAPHIC AS A STAND-ALONE WEB PAGE.

OR WE CAN INCLUDE THE CODE IN AN R MARKDOWN CHUNK OR FLEXDASHBOARD AND KNIT TO HTML.

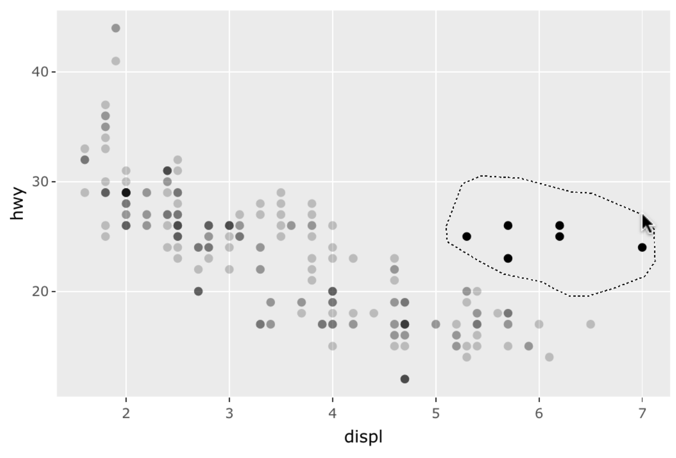
30 baseball outfields — an *interactive* version





`ggplot2 + plotly + crosstalk + DT`

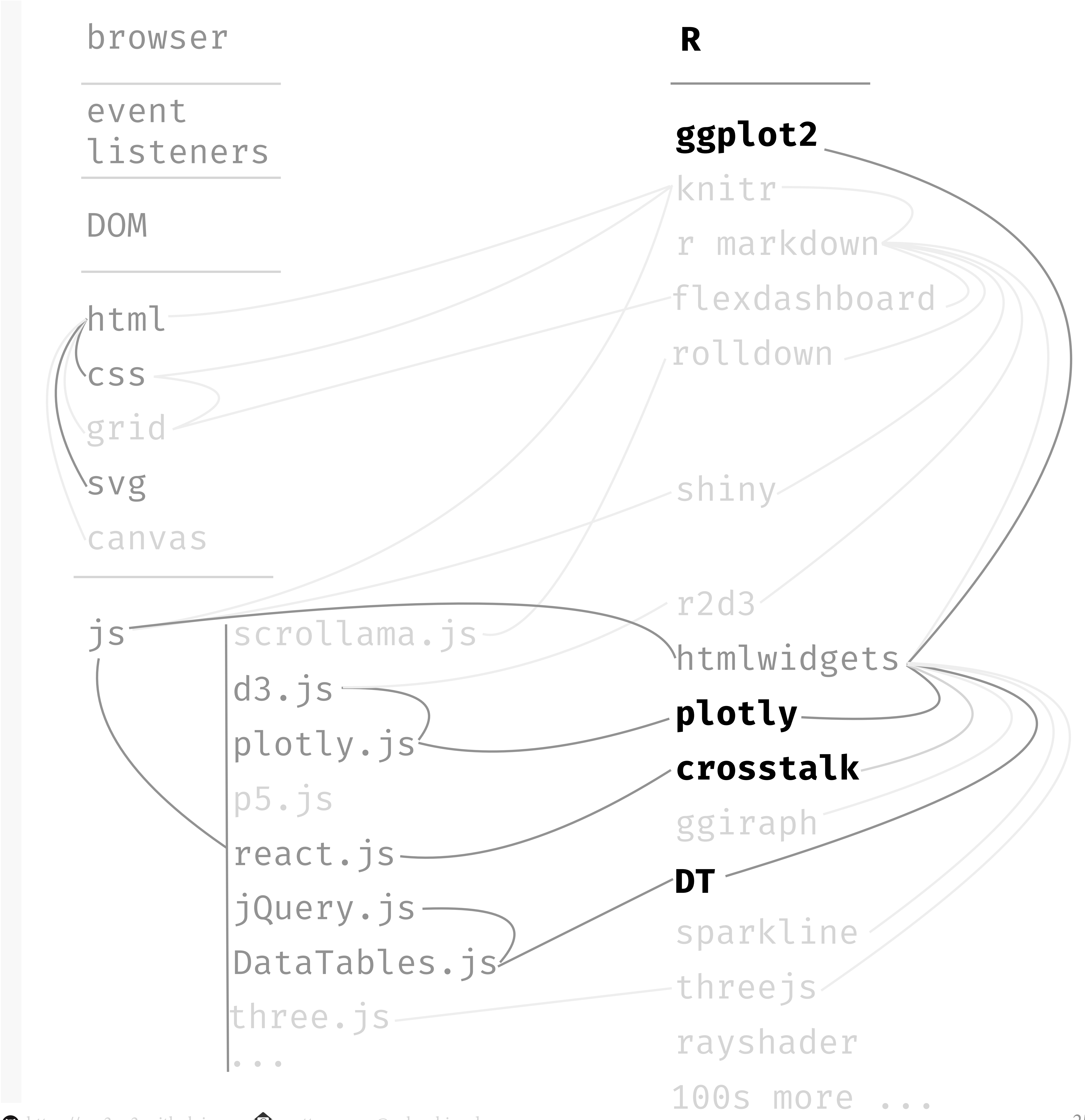
`plotly` is an R package for creating interactive graphics, and interfaces with the same-named javascript library, `plotly.js`, which in turn is based on `d3.js`. R's `plotly` has several helpful features. The first of these are it, like, `ggiraph`, allows **easy integration** with `ggplot2`. The first function, perhaps, to learn, is `ggplotly` which takes as a parameter a `ggplot` object and makes it interactive. And combined with another package, `crosstalk`, it a `plotly` graphic **can link or bind with other `htmlwidgets`**. Here's an example:



Show 10 entries Search:

	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
1	audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
2	audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
3	audi	a4	2	2008	4	manual(m6)	f	20	31	p	compact
4	audi	a4	2	2008	4	auto(av)	f	21	30	p	compact
5	audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
6	audi	a4	2.8	1999	6	manual(m5)	f	18	26	p	compact
7	audi	a4	3.1	2008	6	auto(av)	f	18	27	p	compact
8	audi	a4 quattro	1.8	1999	4	manual(m5)	4	18	26	p	compact
9	audi	a4 quattro	1.8	1999	4	auto(l5)	4	16	25	p	compact
10	audi	a4 quattro	2	2008	4	manual(m6)	4	20	28	p	compact

Showing 1 to 10 of 234 entries Previous 1 2 3 4 5 … 24 Next



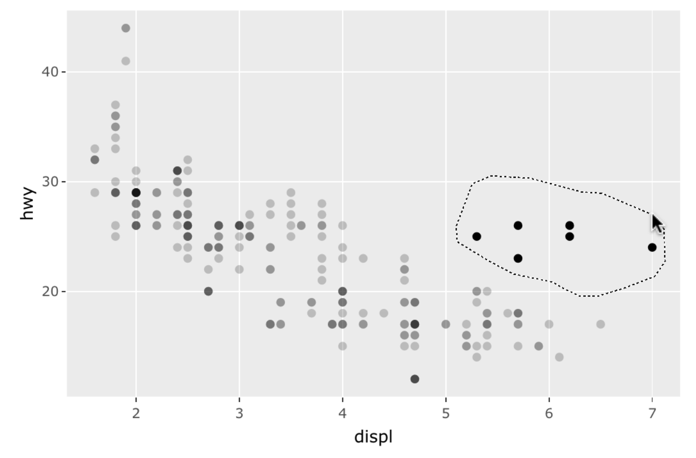
tools for interactive content, `plotly` is a charting library that can bind with other `htmlwidgets`

ggplot2 + plotly + crosstalk + DT

`plotly` is an R package for creating interactive graphics, and interfaces with the same-named javascript library, `plotly.js`, which in turn is based on `d3.js`. R's `plotly` has several helpful features. The first of these are it, like, `ggiraph`, allows **easy integration** with `ggplot2`. The first function, perhaps, to learn, is `ggplotly` which takes as a parameter a `ggplot` object and makes it interactive. And combined with another package, `crosstalk`, it a `plotly` graphic **can link or bind with other `htmlwidgets`**. Here's an example:

```r
library(ggplot2)
library(plotly)    # FUNCTIONS FOR INTERACTIVE CHARTS
library(crosstalk) # FUNCTIONS ENABLING HTMLWIDGETS TO SHARE INTERACTIVITY
library(DT)        # FUNCTIONS FOR INTERACTIVE TABLES

m <- highlight_key(mpg)
```

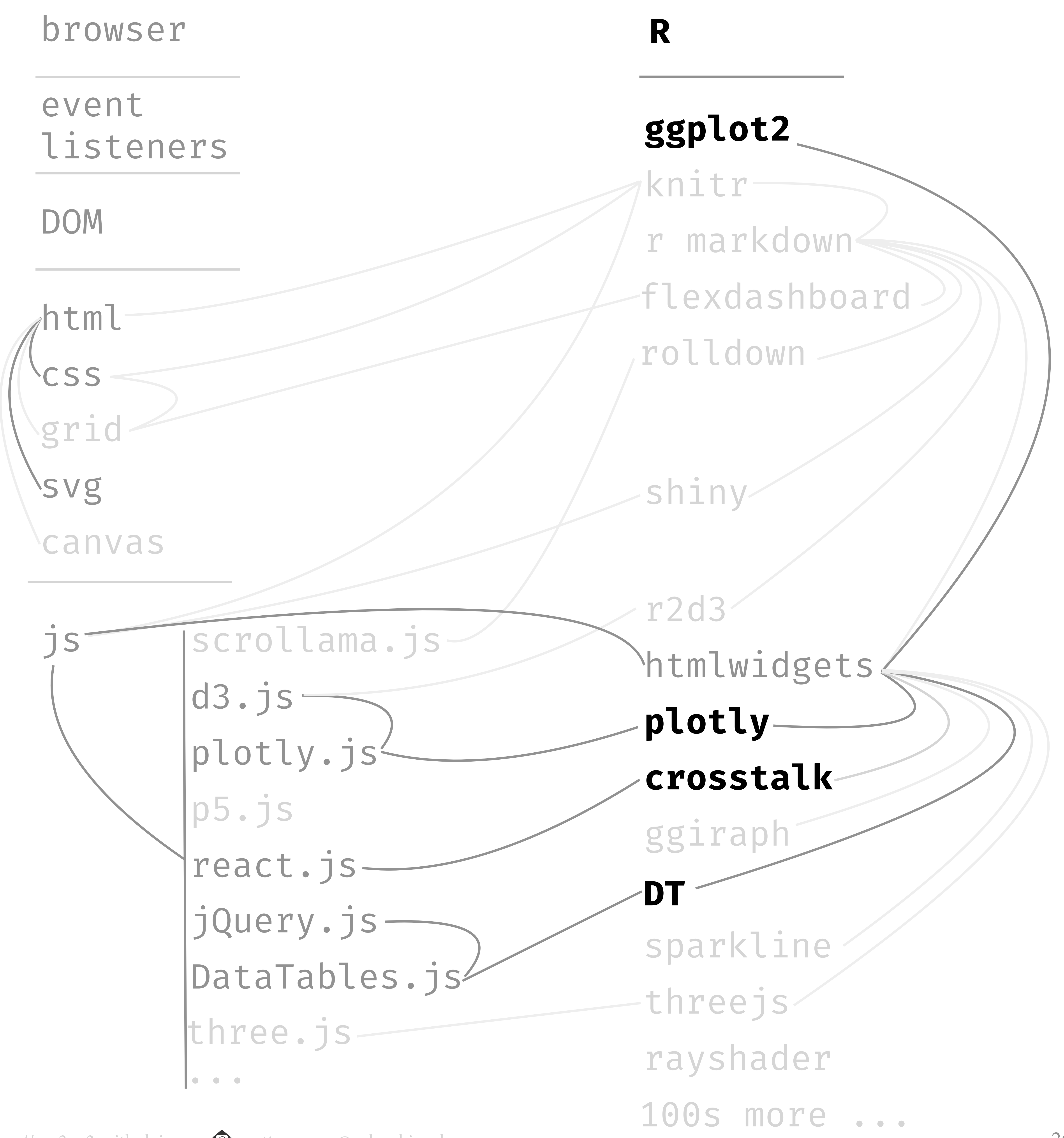
```r
p <- ggplot(           # HERE'S OUR GGPLOT OBJECT
  data = m,
  mapping = aes(
    x = displ,
    y = hwy)) +
  geom_point()

gg <- highlight(       # NOW WE MAKE THE GGPLOT OBJECT INTERACTIVE
  p = ggplotly(p),
  on = "plotly_selected")

bscols(gg, datatable(m))   # FINALLY WE LINK BOTH INTERACTIVE WIDGETS.
```



Show 10 entries   Search:

	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
1	audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
2	audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
3	audi	a4	2	2008	4	manual(m6)	f	20	31	p	compact
4	audi	a4	2	2008	4	auto(av)	f	21	30	p	compact
5	audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
6	audi	a4	2.8	1999	6	manual(m5)	f	18	26	p	compact
7	audi	a4	3.1	2008	6	auto(av)	f	18	27	p	compact
8	audi	a4 quattro	1.8	1999	4	manual(m5)	4	18	26	p	compact
9	audi	a4 quattro	1.8	1999	4	auto(l5)	4	16	25	p	compact
10	audi	a4 quattro	2	2008	4	manual(m6)	4	20	28	p	compact

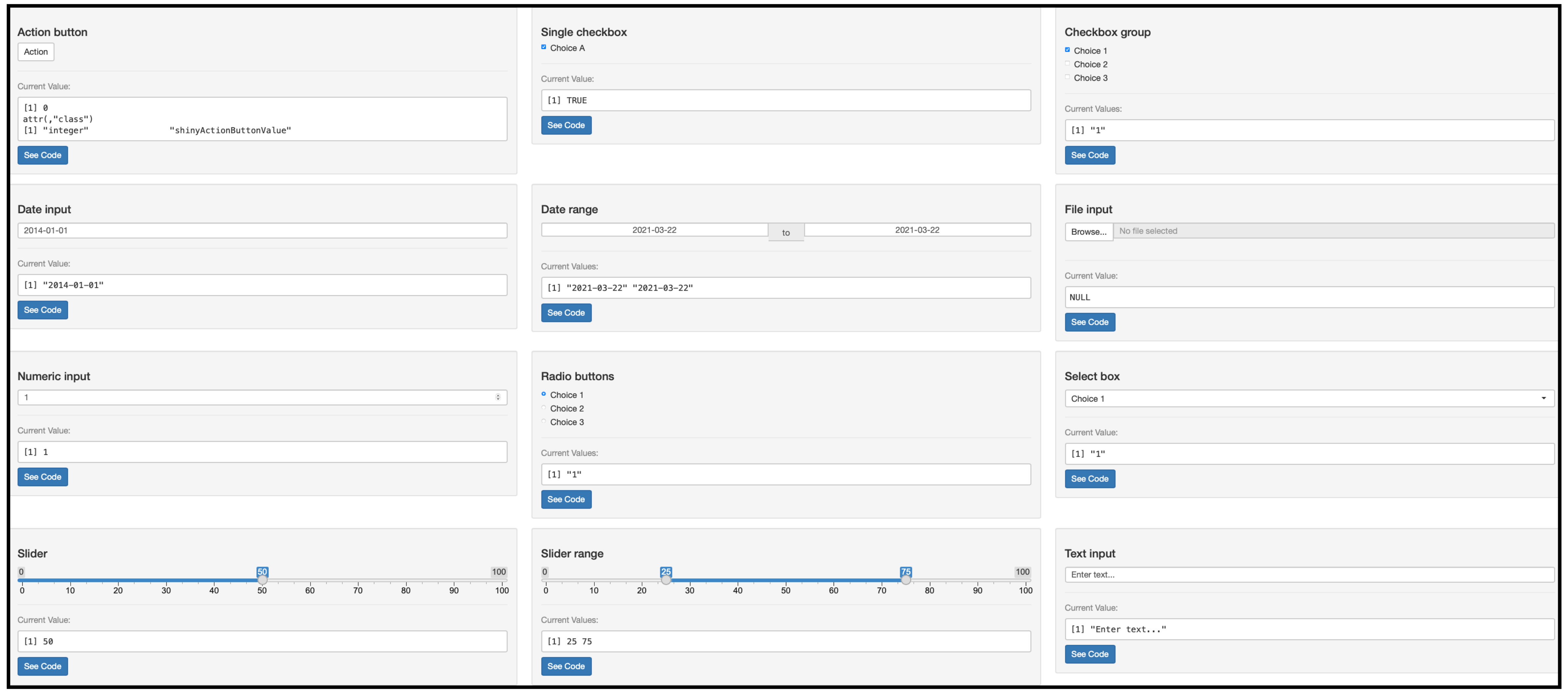
Showing 1 to 10 of 234 entries   Previous 1 2 3 4 5 … 24 Next

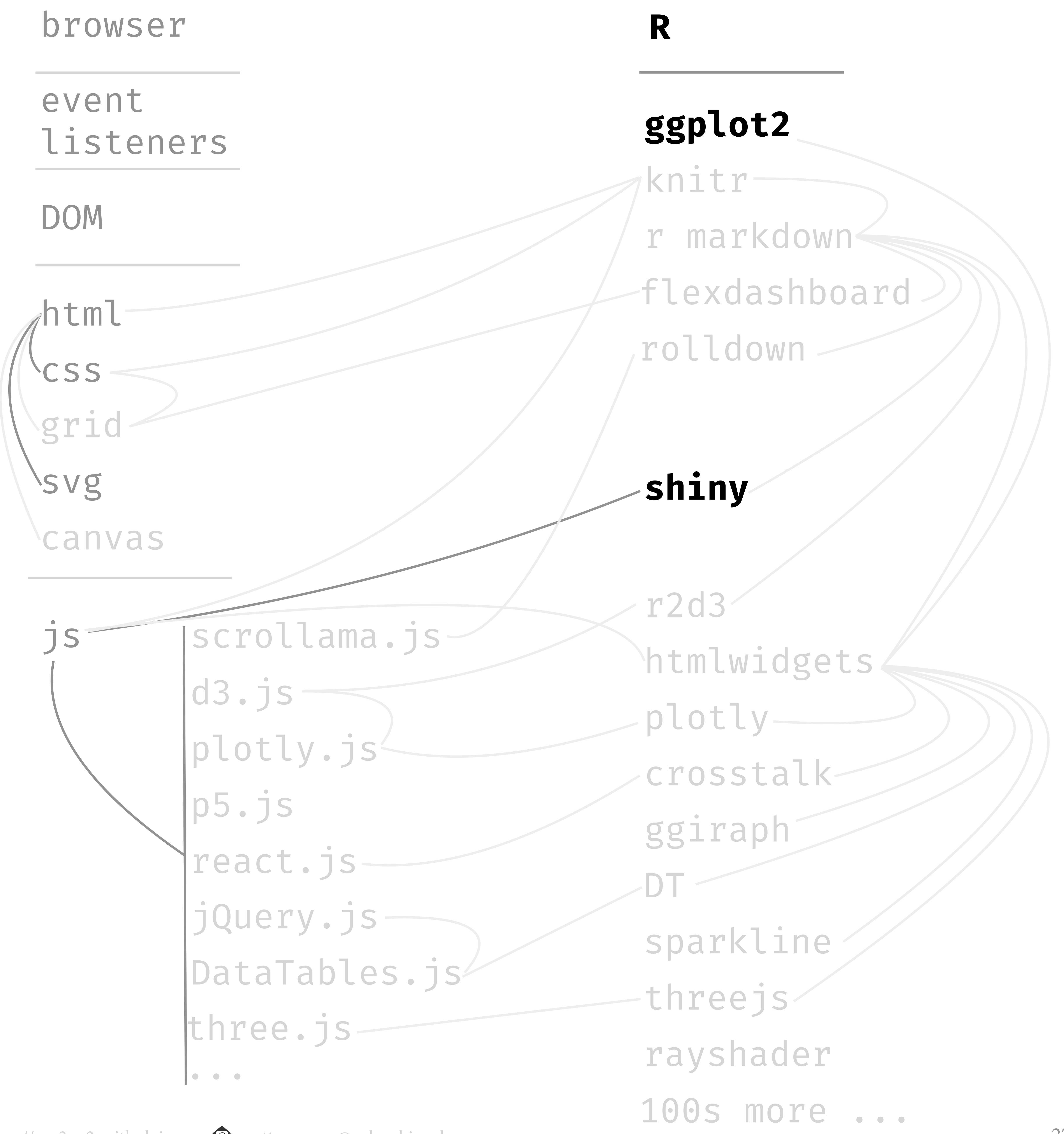


 web application tools are more complex but allow more sophisticated interactions

ggplot2 + shiny + ...

Shiny is for developing **web applications**. This means it runs on a **web server** to enable user interface widgets on a **webpage**. Further, it **requires linking to an active R session**. Thus, unlike the previous software, we cannot share a single, standalone html file. The closest we get is to share an r markdown file with shiny code that someone can open in RStudio and click "run" to start a server. Below are examples of various widgets we can use to create these interactive, web applications.





tools for interactive content, **web application tools are more complex but allow more sophisticated interactions**

R + r2d3 + d3.js

We can also pass data objects directly from an R environment to the industry standard d3 javascript library using the R package r2d3. This allows us to combine the strengths and flexibility of both languages.

We can either run the d3 script directly from R, or we can embed the d3 script within an R markdown document as a d3 code chunk in whatever your choice of R markdown format: html document, distill, flex dashboard, …

R markdown partial file, toy example

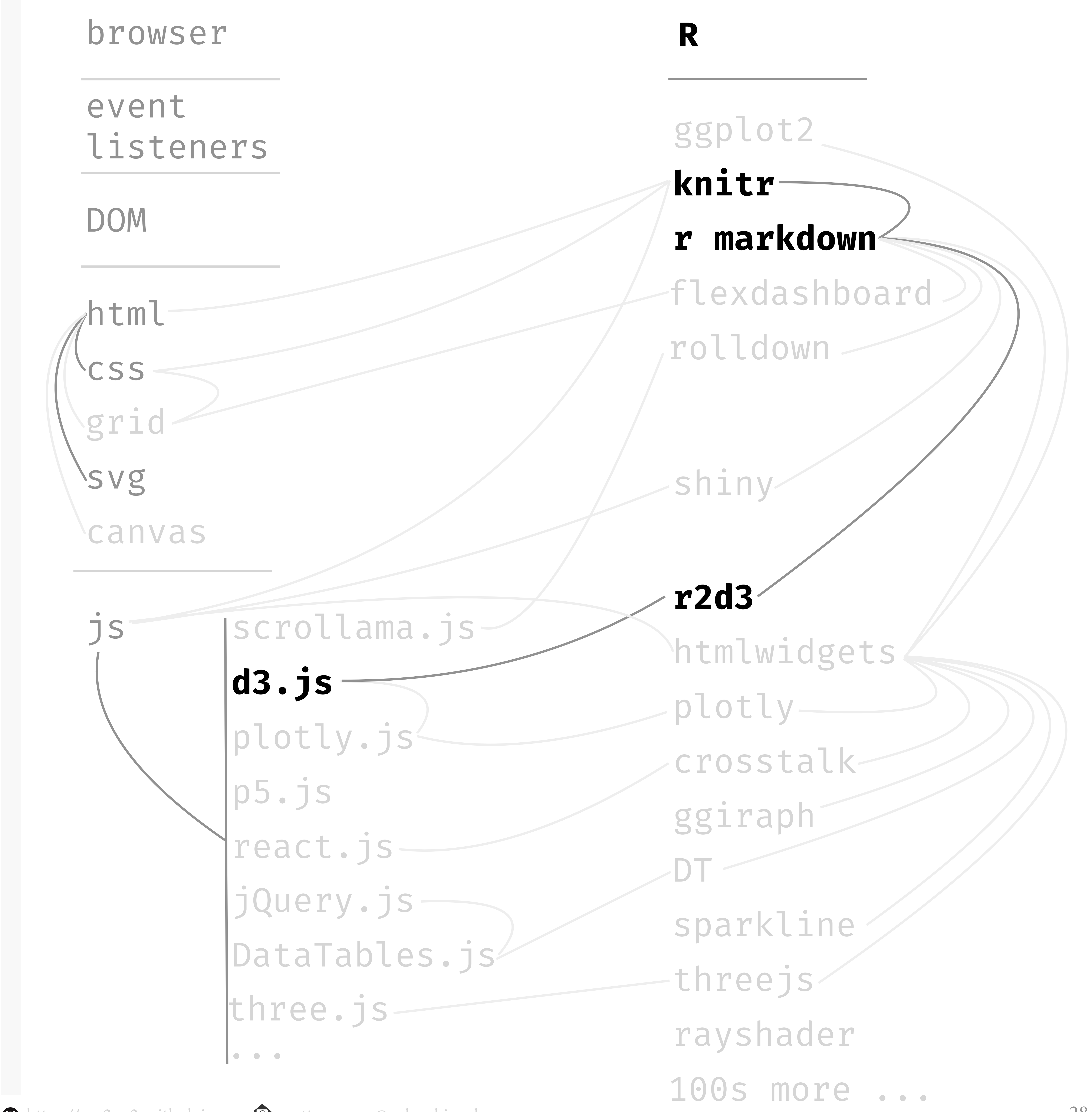
````
```{r}
library(r2d3)
bars <- c(10, 20, 30)
```

```{d3 data = bars}
svg.selectAll('rect')
  .data(data)
  .enter()
    .append('rect')
      .attr('width', function(d) { return d * 10; })
      .attr('height', '20px')
      .attr('y', function(d, i) { return i * 22; })
      .attr('fill', 'orange');
```
````

https://rstudio.github.io/r2d3/

Resulting svg embedded in knitted html file

```
<svg ...>

<style ...></style>

<rect width="100" height="20px" y="0"
fill="orange"></rect>
<rect width="200" height="20px" y="22"
fill="orange"></rect>
<rect width="300" height="20px" y="44"
fill="orange"></rect>
</svg>
```

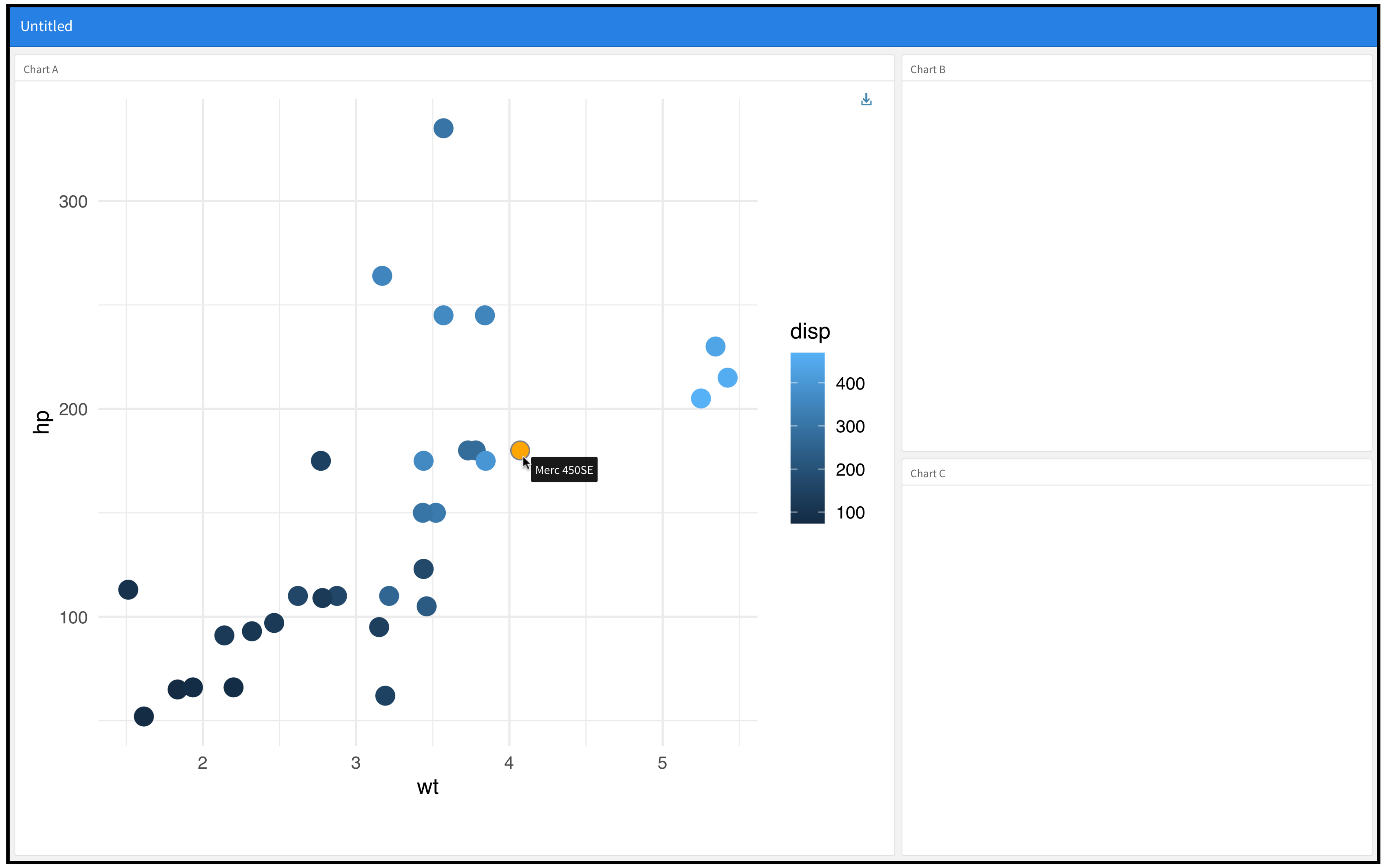
What we see viewing the html file

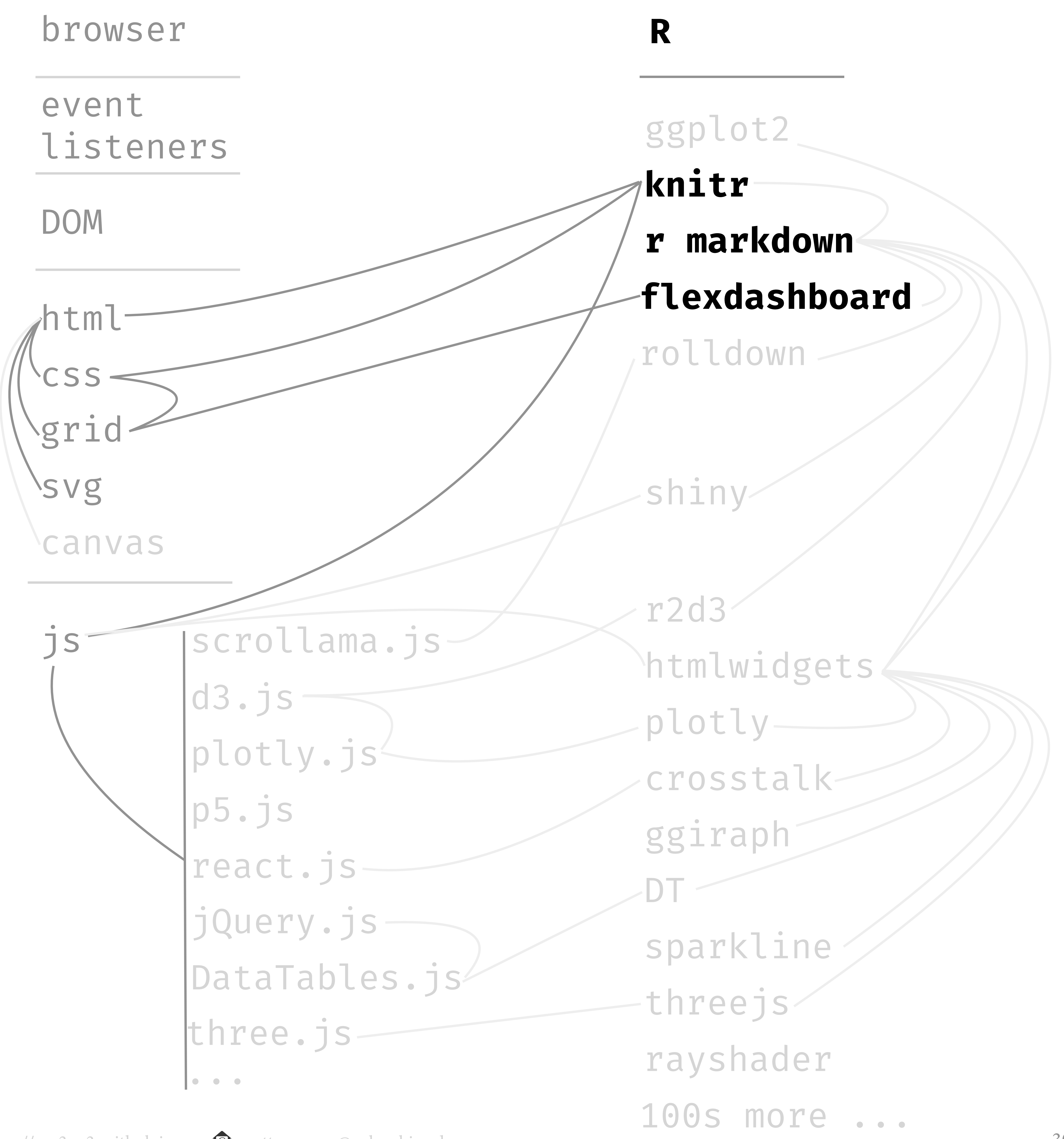


organizing interactive graphics with web technologies — (for dashboards)

knitr + rmarkdown + flexdashboard

We can **organize** various widgets and enable their communication through web technologies, all placed inside an `html` file. Perhaps my favorite way to bring these technologies together is using **`r markdown` templates** like **flexdashboard** that **`knitr`** and **`RStudio`** uses to weave together text, image, code and results. Along with markdown templates, we can roll our own with **`css grid`**, adding code chunks between `<div class="">` and `</div>` where we define our own `css` classes. Here's a screenshot of an example below:

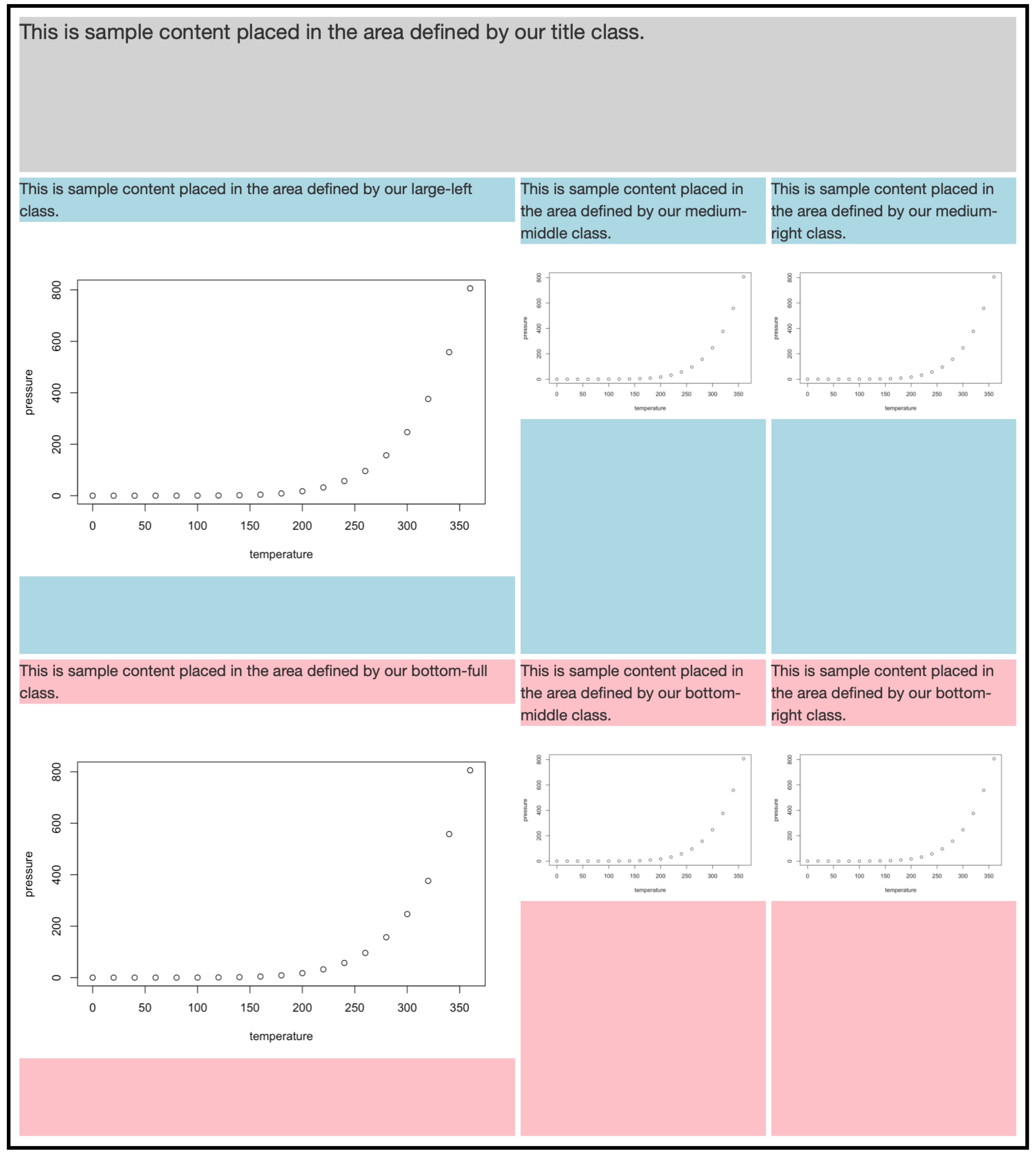


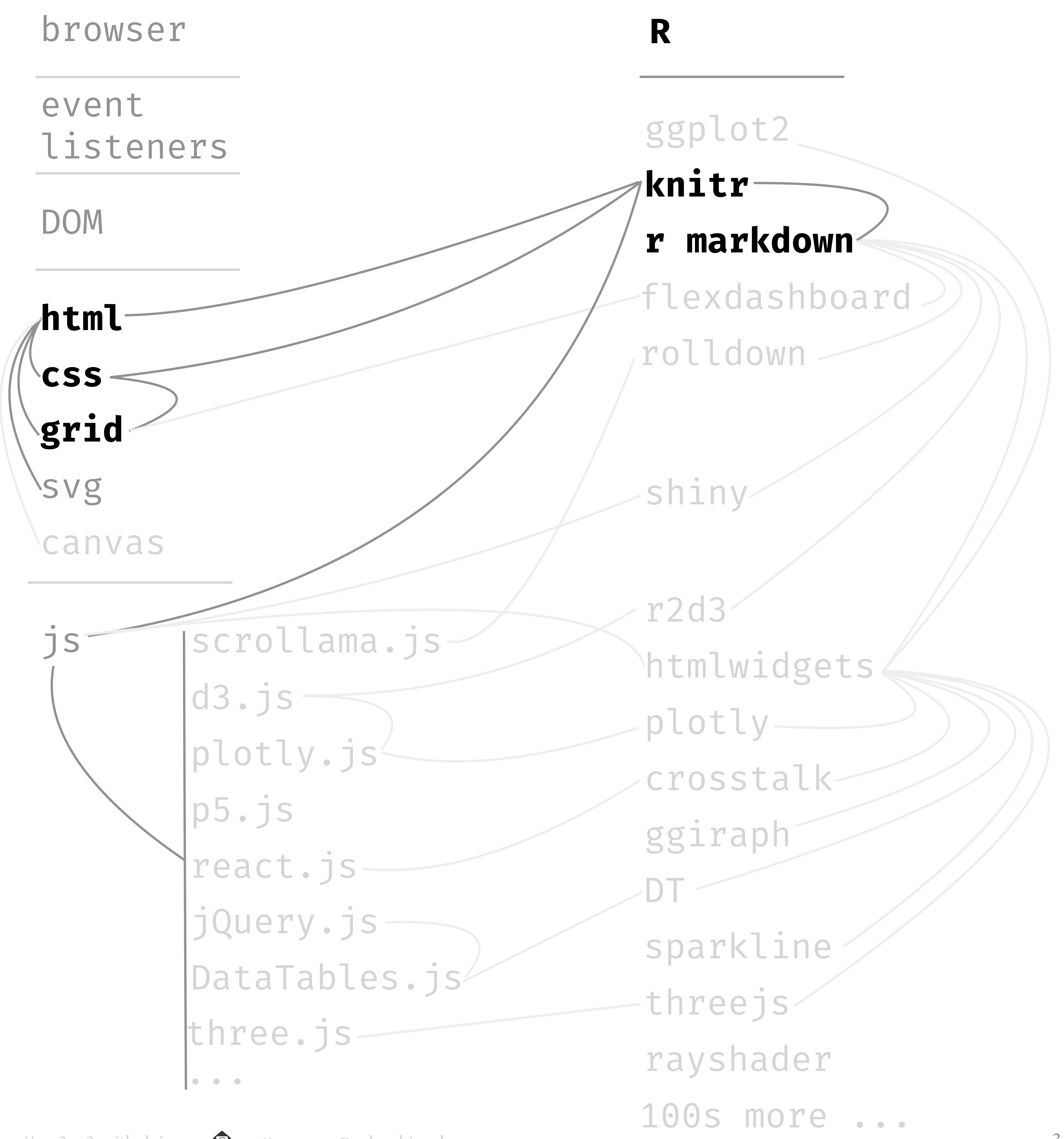


knitr + rmarkdown + css grid + html

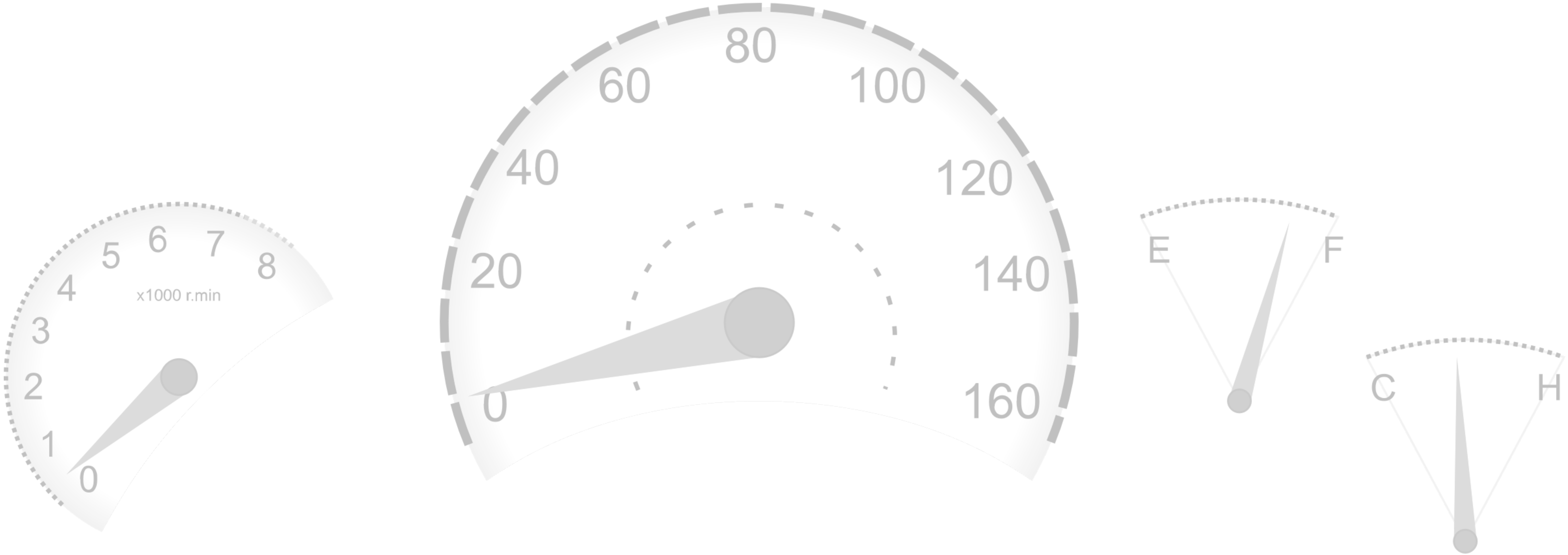
We can **organize** various widgets and enable their communication through web technologies, all placed inside an `html` file. Perhaps my favorite way to bring these technologies together is using **`r markdown` templates** like **flexdashboard** that **`knitr`** and **`RStudio`** uses to weave together text, image, code and results. Along with markdown templates, we can roll our own with **`css grid`**, adding code chunks between `<div class="">` and `</div>` where we define our own css classes. Here's a screenshot of an example below:

```
<style>

.main-container {
  min-width: 950px;
  max-width: 950px;

}

.gridlayout {
  display: grid;
  position: relative;
  margin: 10px;
  gap: 5px;
  grid-template-columns:
    repeat(8, 1fr);
  grid-template-rows:
    repeat(8, 140px);
}

.gridlayout * {
  max-width: 100%;
  object-fit: contain;

}

.title {
  background: lightgray;
  grid-column: 1 / 9;
  grid-row: 1 / 2;
  font-size: 14pt;
}

.large-left {
  background: lightblue;
  grid-column: 1 / 5;
  grid-row: 2 / 5;
}
```





visual narrative flow



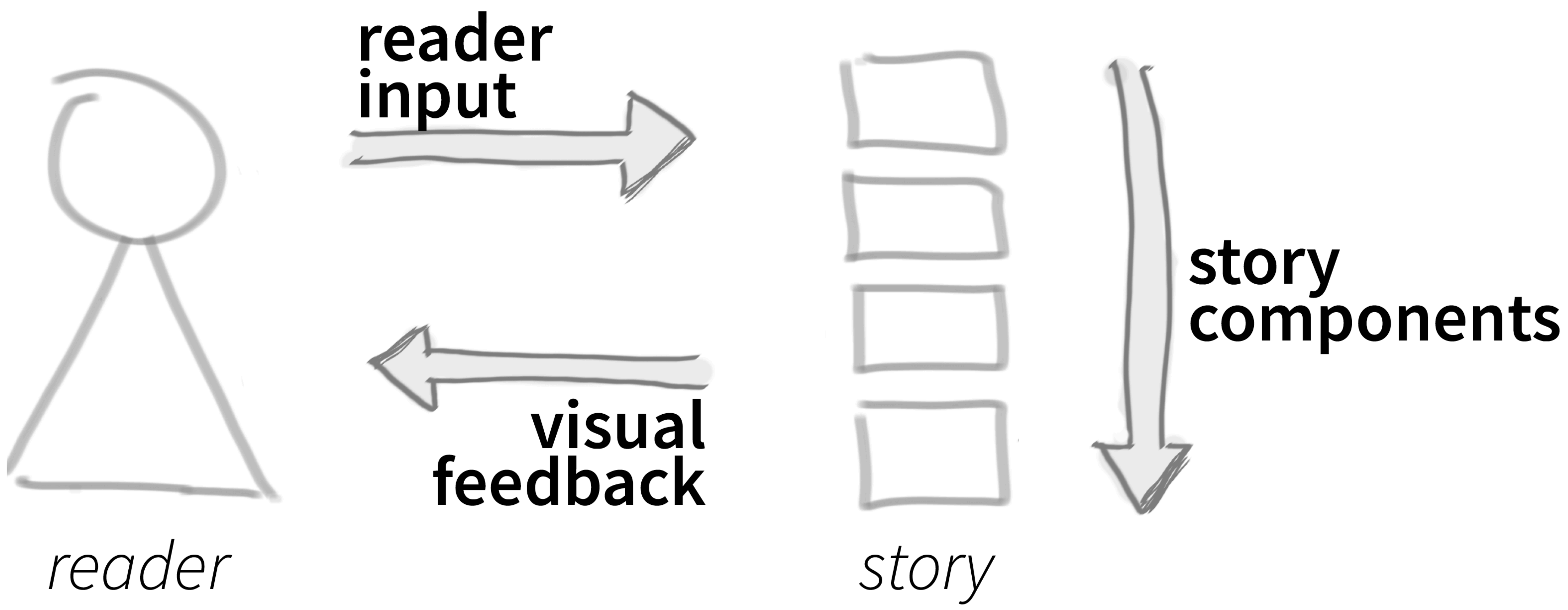
For a vehicle dashboard, who’s its audience? What’s its purpose? Needs words? — **Audience and purpose drive design.**

visual narrative flow, *if* a dashboard, the need for *guided* dashboards

An issue of communication is related to storytelling ability. Dashboards are increasingly used for decision making and communication across contexts: top-down, within departments, and across the organization. **Dashboards that capture only the data and not the semantics of the data, or what was done in response to the data, can be *insufficient* for communication purposes**. In BI, people often take screenshots of dashboards and put them into slide presentations in order to annotate them with contextual information, suggesting **a *need* for more powerful storytelling features**.

— Sarikaya, *What Do We Talk About When We Talk About Dashboards?* January 2019.

visual narrative flow, **characteristics that affect experience**

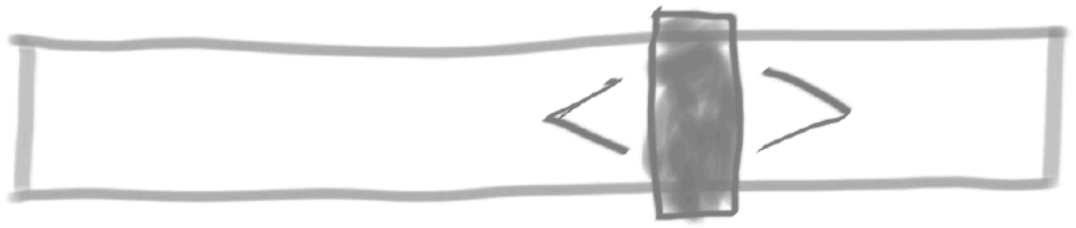


visual narrative flow | the congruence between ***flow-factors***, i.e., 1) the way a reader navigates the story, 2) the visual components of the story, and 3) the type of visual feedback the reader receives; along with the nature of the data and facts that the author wants to communicate.

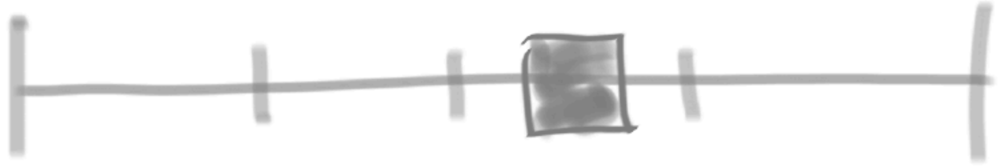
— McKenna, *Visual Narrative Flow: Exploring Factors Shaping Data Visualization Story Reading Experiences*. June 2017.

design space for *flow factors*, navigation input · level of control · navigation progress · story layout · role of visualization · story progression · navigation feedback

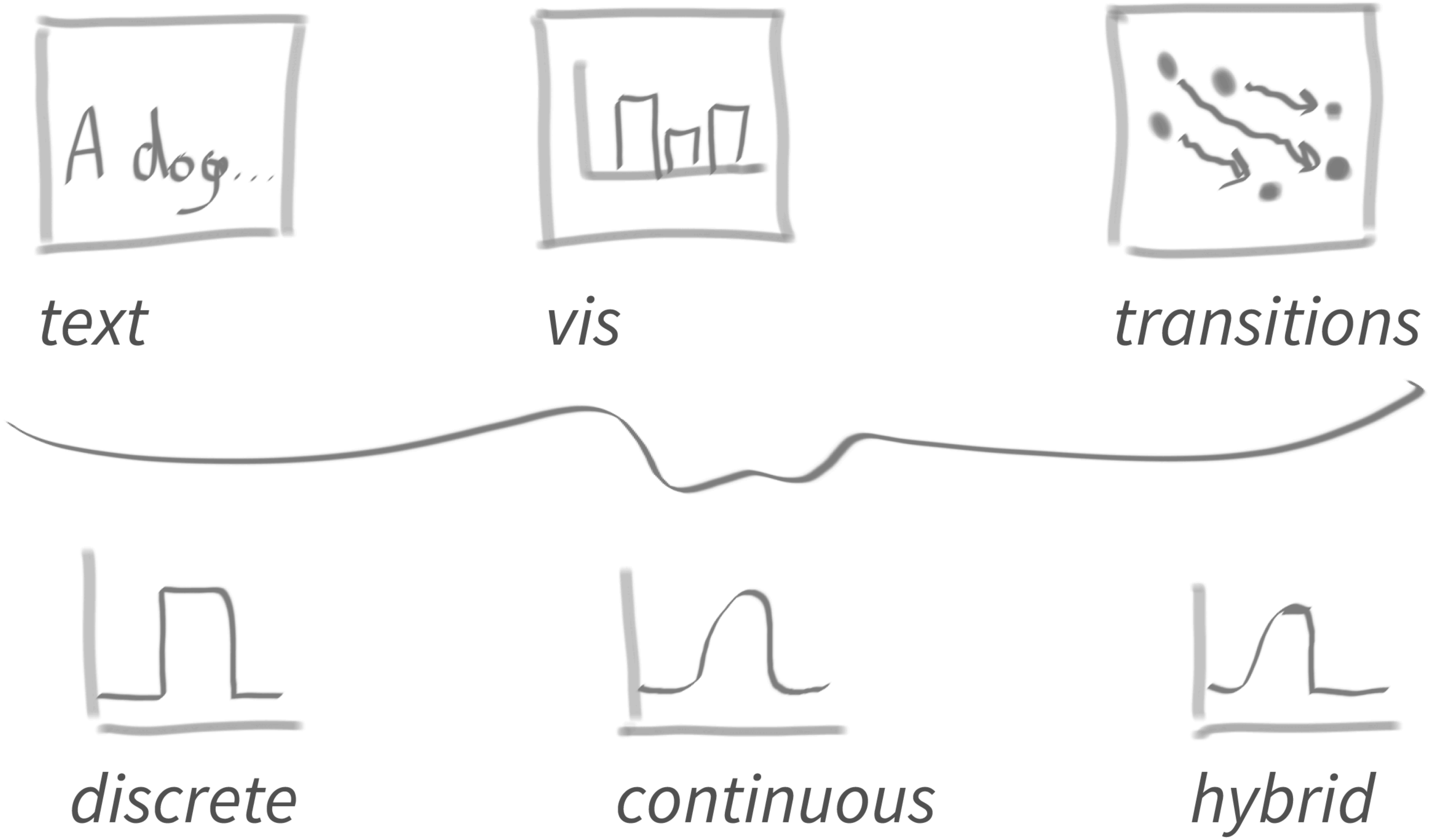
button



scroll

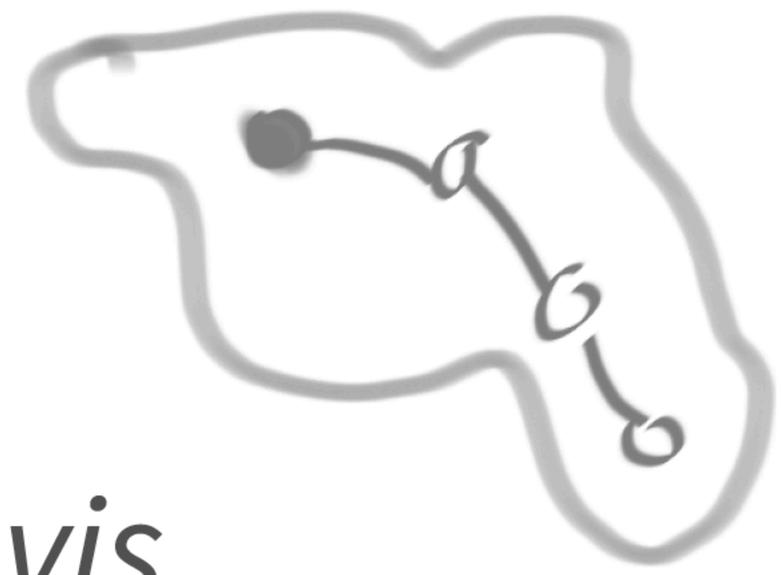


slider


A dog...
text
vis
transitions
discrete
continuous
hybrid

text

dots



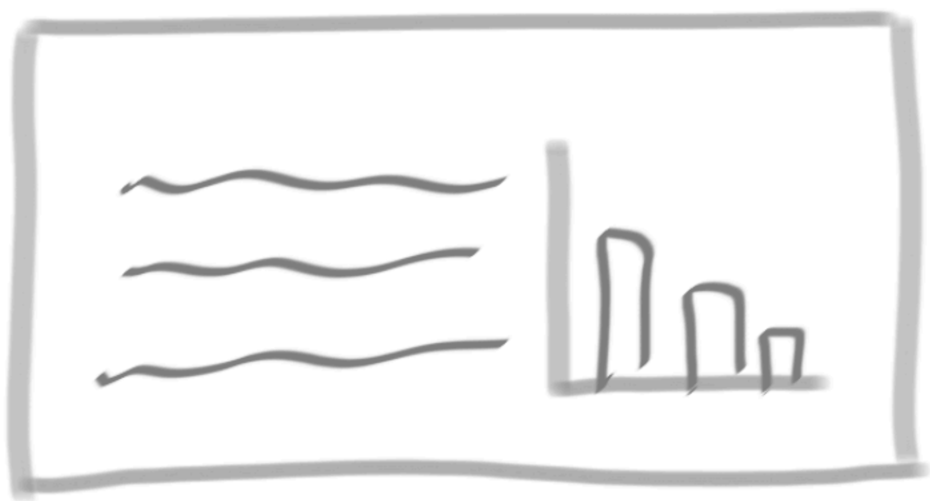
vis

design space for *flow factors,* navigation input · level of control · navigation progress · **story layout** · role of visualization · story progression · navigation feedback

document

slideshow

hybrid

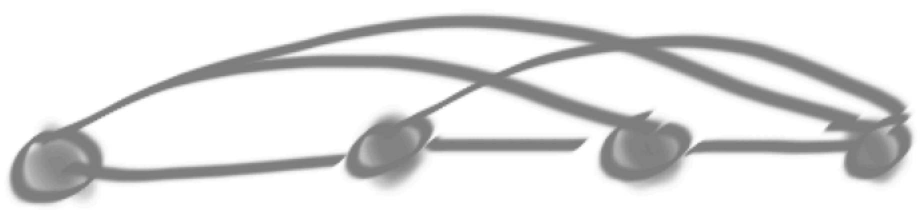


equal

figure

annotated

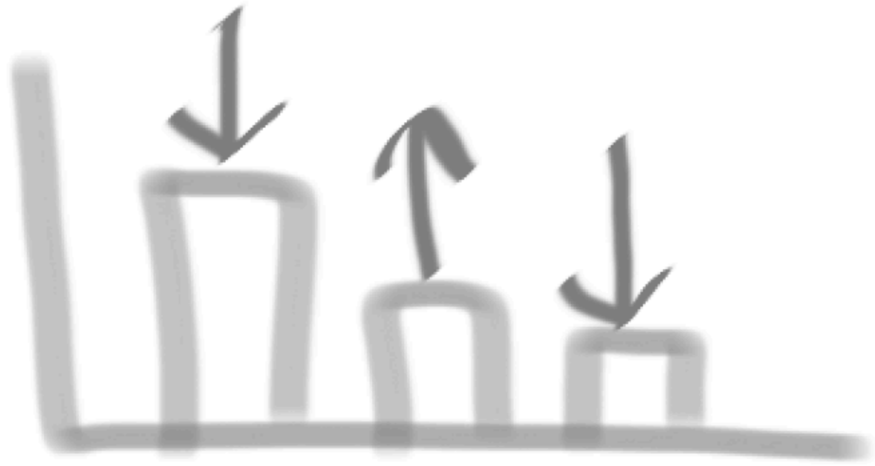
linear



linear skip

tree/graph

text



vis

widget

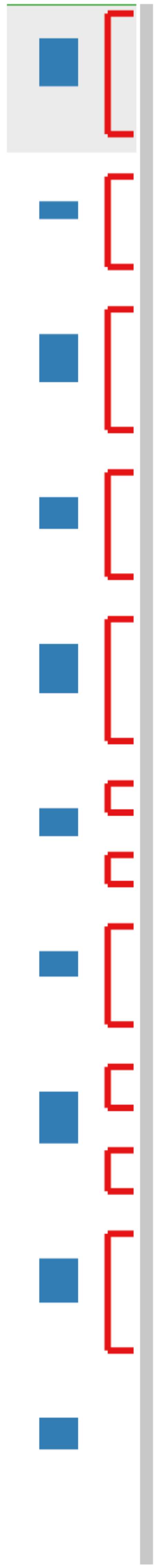
 https://ssp3nc3r.github.io scott.spencer@columbia.edu

design space for *flow factors,* **taxonomies like theirs can be helpful in seeing many example variations of these techniques**

#	title	navigation input			level of control			navigation progress				story layout			role of visualization			story progression			navigation feedback			
		scroll	button	slider	text	vis	anim	text	dots	vis	other	doc	slide	cols	equal	figure	annot.	linear	skip	other	text	vis	widget	order
1	A Visual Introduction to Machine Learning	●			C	C	C			○		●		2	●			●			●			sync
2	Scientific Proof that Americans are Completely	●			C	C	D					●		2	●			●			●			sync
3	Fewer Helmets, More Deaths	●	●		C	C	D		●			○	○	2	●			●			●		●	vis
4	A 3-D View of a Chart That Predicts The Econ		●		D	D	D		●				●	1			●		●				●	sync
5	A Visual Analysis of Battle at the Berrics	●			C	C	D					●		1		●		●			●			sync
6	Budget Forecasts, Compared With Reality		●	●	D	D	D	●		○			●	1			●		●				●	sync
7	Human Development Trends, 2005		●		D	D	D	●					●	1			●		●				●	hyb
8	Diary of a Food Tracker	●	●		H	H	H			●		○	○	1					●		●	●		vis
9	How Americans Die		●		D	D	D		●				●	1			●		●				●	text
10	Visualizing MBTA Data: An Interactive Explora	●			C	C	C					●		1		●		●			●			vis
11	The World According to China	●			C	C	D					●		1	●			●			●			swap
12	How the U.S. and OPEC Drive Oil Prices	●			C	H	D		●			○	○	1	●				●		●		●	sync
13	Scaling Mt. Everest: A Scroll Up the Icy Path	●		●	C	C	C			●		●		3	●				●		●		●	sync
14	Snow Fall: The Descent Begins	●			C	C	D			○		●		2	●			●			●			sync
15	The Story of Jess & Russ	●			C	C	C					●		1	●			●			●			sync
16	2014 Was the Hottest Year on Record	●			C	C	D			○		●		1			●	●				●		sync
17	The World's Ball	●	●		C	C	D					●		1		●		●			●			swap
18	The Russia Left Behind	●	●		C	C	C			●		●		2		●			●		●	●		sync
19	The Water We Eat	●			C	H	C			○		○	○	1-2	●	●	●	●			●	○		sync
20	Ski Jumping	●	●		C	H	C		●			○	○	1			●		●		●		●	swap
21	The Dawn Wall: El Capitan’s Most Unwelcomi	●			C	C	C			○		●		2	●		●	●			●	○		sync
22	Russia’s Endgame in Ukraine	●			C	C	-					●		1		●	●	●			●	○		swap
23	At Top Colleges, an Admissions Gap for Mino		●		D	D	D	●					●	1			●		●			●		text
24	Greenland Is Melting Away	●			C	C	C			○		●		2			●	●			●	○		sync
25	How Different Groups Spend Their Day		●		D	D	D	●					●	1			●			graph		●		sync
26	Deconstructing the Past: A New Look at Histo	●	●		D	D	D				block		●	1			●		●		●		●	sync
27	Dollar-a-Day Schools	●	●		D	D	D				image		●	1			●		●		●		●	sync
28	ChopTainer	●			H	C	-					○	○	1		●	●	●			●	○		sync
29	Neurotic Neurons: An Interactive Explanation		●		D	D	D			○			●	1			●			tree				hyb
30	The Year Ahead 2016: 50 Companies to Watc	●	●		C	C	-	●				●		2	●		●		●		●	●		sync
31	The Museum of the World	●	●		-	C	C			●		○	○	1		●	●		●			●		sync
32	Bloomberg Carbon Clock	●			D	D	D						●	1			●	●						vis
33	Interactive: Global Emission		●		-	D	D	●					●	1			●		●			●		vis
34	A Map of Olympic Medals		●	●	-	D	D	●					●	1			●		●			●	●	sync
35	Shaun White's Double McTwist		●		D	D	D			○			●	1			●	●				●	●	vis
36	Bubble to Bust to Recovery		●		D	D	D		●				●	1			●		●		●	●		vis
37	A Nation Divided	●	●		D	C	D		●			○	○	1		●		●			●		●	sync
38	342,000 Swings Later, Derek Jeter Calls It a C	●			C	D	C					○	○	1	●			●			●			sync
39	52 Places to Go in 2015	●			C	C	-					●		2		●		●			●	●		sync
40	A Walk Through the Gallery	●			D	D	C					●		2			●	●				●		text
41	Illuminating North Korea	●			C	C	-					●		1		●		●			●	●		sync
42	Walking New York	●	●		C	D	-					●		2	●	●		●			●	●		vis
43	Why Infectious Bacteria Are Winning	●			C	D	D					●		1	●	●		●			●			text
44	Hell and High Water	●	●		H	D	D			○	time	○	○	1-2	●		●	●			●		●	text
45	Eigenvectors and Eigenvalues	●			C	C	-					●		1		●		●			●	●		sync
46	Film Dialogue from 2,000 Screenplays, Broken	●			C	C	D					●		1	●	●		●			●			sync
47	What's Really Warming the World?	●	●		H	D	D		●			○	○	1			●		●		●		●	sync
48	If the Moon Were Only One Pixel	●			C	C	-					●		1	●			●			●	●		sync
49	State of the Gadget Union	●			C	C	-					●		1		●		●			●	●		text
50	Why Pinellas County is the Worst Place in Flo		●		D	D	D		●				●	1			●	●			●	●		vis
51	The Dark Side of Guardian Comments	●			C	C	D	●				○	○	1		●	●	●			●			text
52	Trolls of the West	●			H	H	C					○	○	1		●	●	●			●	●		sync
53	Make Your Money Matter	●			H	H	C					○	○	1		●	●	●			●	●		sync
54	Bond: License to Drive	●	●		D	C	C		●			○	○	1			●		●		●	●		sync
55	Every Last Drop - Water Saving Website	●			H	C	C					○	○	1			●	●			●	●		sync
56	Green Honey	●	●		C	D	D		●			●		2	●		●		●		●		●	sync
57	The Clubs that Connect The World Cup	●			C	C	D					●		2	●			●			●	●		vis
58	Gestalt Principles for Data Visualization	●			C	C	D					●		2	●	●		●			●			text
59	Money Wins Elections	●			C	C	C					○	○	1	●		●	●			●	●		sync
60	The Air We Breathe	●	●		H	C	D			●		●		1		●		●			●	●		text
61	Most Unlikely Comebacks: Using Historical Da	●			C	C	C			●		●		2	●			●			●	●		sync
62	Started From The Bottom	●			C	C	D					●		2	●			●			●			text
63	A Game of Shark and Minnow	●			C	H	D		●			●		1-2		●	●	●			●	○		text
64	Fleeing Syria for Europe: Safaa's fatal journey	●			C	C	C			●		●		2		●	●	●			●	●		sync
65	New Energy Outlook 2016	●			C	C	D					●		2		●	●	●			●	●		text
66	Introducing Serio Verify	●			H	C	C		●		slider	○	○	1			●	●			●	●		sync
67	Im Zentrum Des Geschehens	●			H	C	D					●		2			●	●			●			sync
68	Das Tunnelsystem der Rekorde	●			C	C	-			●		●		2		●	●	●			●	●		vis
69	These Memories Won't Last	●			C	C	-					○	○	1		●		●			●	●		sync
70	Fuglefjellet	●			C	D	D					○	○	3	●		●	●			●			sync
71	Gun Deaths In America		●		D	D	D	●					●	1		●	●		●		●	●		vis
72	A Trail of Terror in Nice, Block by Block		●		D	D	D			○			●	1	●		●	●				●		vis
73	The Sieve of Eratosthenes	●	●		C	D	D			○		○	○	2	●	●		●			●	●		text
74	The Wild Path: An Icelandic Adventure	●			C	D	C			●		●		2	●	●		●			●	●		sync
75	How Fed Rates Move Markets	●			C	C	C			○		○	○	1	●			●			●	●		sync
76	What ECB Stimulus Has Done	●			H	C	C			●		○	○	1	●	●		●				●		vis
77	Sizing Up The Olympics	●			C	C	C			●	slider	○	○	1			●		●			●	●	vis
78	The Internet of Things		●		D	D	D						●	1			●	●			●	●		sync
79	Setting the Pace: The Fed Acts, Markets Move		●	●	D	D	H		●	●			●	1			●		●			●		vis
80	What I Saw in Syria	●	●		D	D	-				slider		●	1			●		●		●	●	●	sync

design space for *flow factors,* experimental results — visualizations and animated transitions help audience engagement

scroll | press/swipe | continuous | discrete | moving text | 600 transition duration | sticky text



Teaching Bar Charts through Data Visualization

showing the raw data

Data enables us to better understand the world around us.

Take this list of a few characters from the TV show, The Simpsons. It includes their names, genders, and their ages.

Let's start with just the 5 main Simpson family characters.

	Gender	Adult
Lisa	F	N
Bart	M	N
Homer	M	Y
Marge	F	Y
Maggie	F	N

design space for *flow factors,* ***general preferred approach of "business intelligence experts", one study***

Interactivity. …*When creators were asked if they want the visualizations in the reports to be completely interactive and encourage readers to interact with them (e.g. using drill down/up, filter, link & brush)*, four of our **experts prefer to have interactive visualizations that permit linking and brushing (*i.e.* data selection)**.

But **they would limit the more advanced interactions** such as drill down/up or filtering.

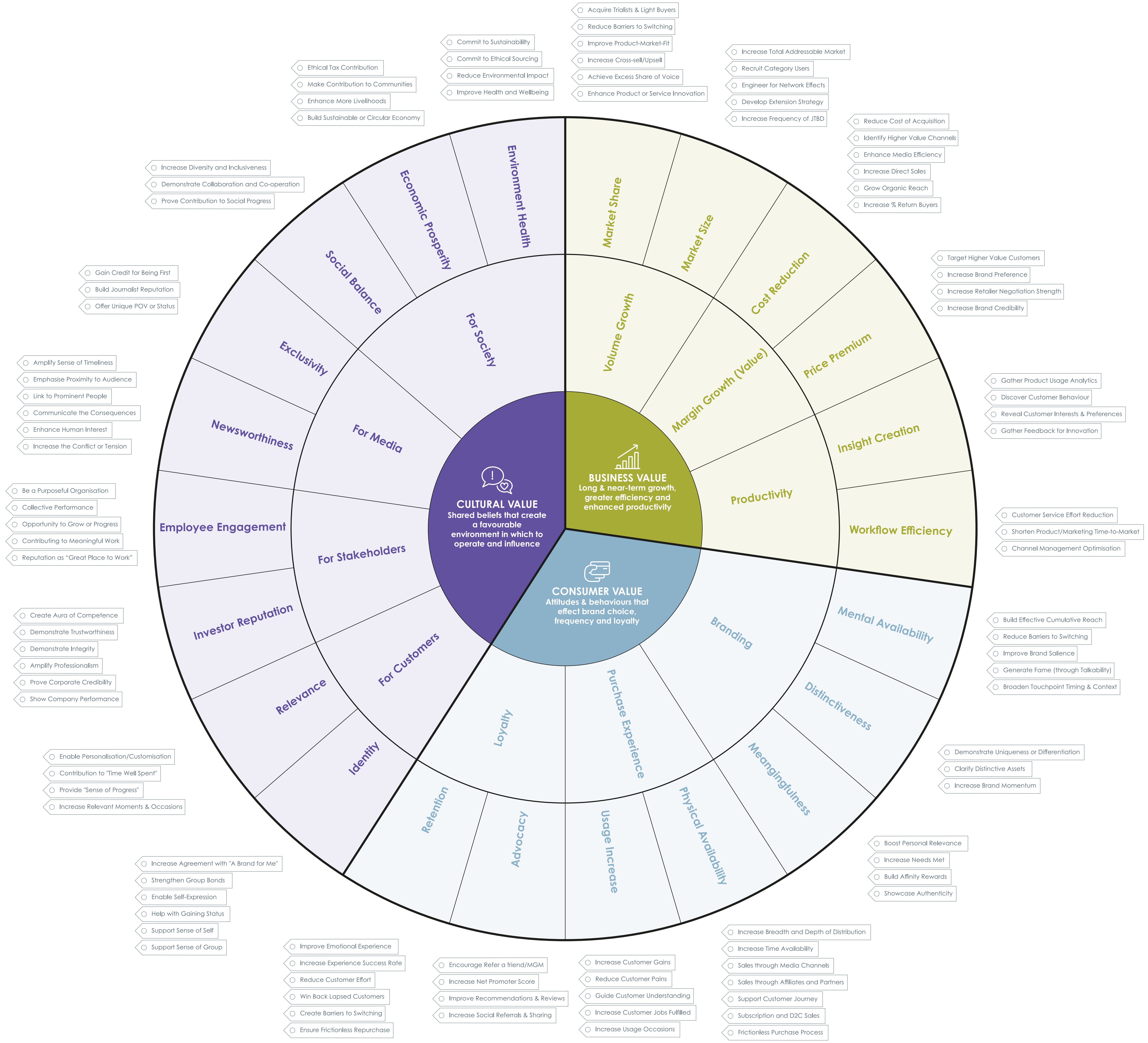
They felt that **all the data needed to tell the story should be displayed clearly in the report without the need to explore the data further**.… Thus authors feel business stories should be mostly author-driven and constraint, known to work best when the goal is storytelling or efficient communication.

Agree? Disagree? Explain.

— Kotzé, *Storytelling in Visual Analytics Tools for Business Intelligence*. 2013.

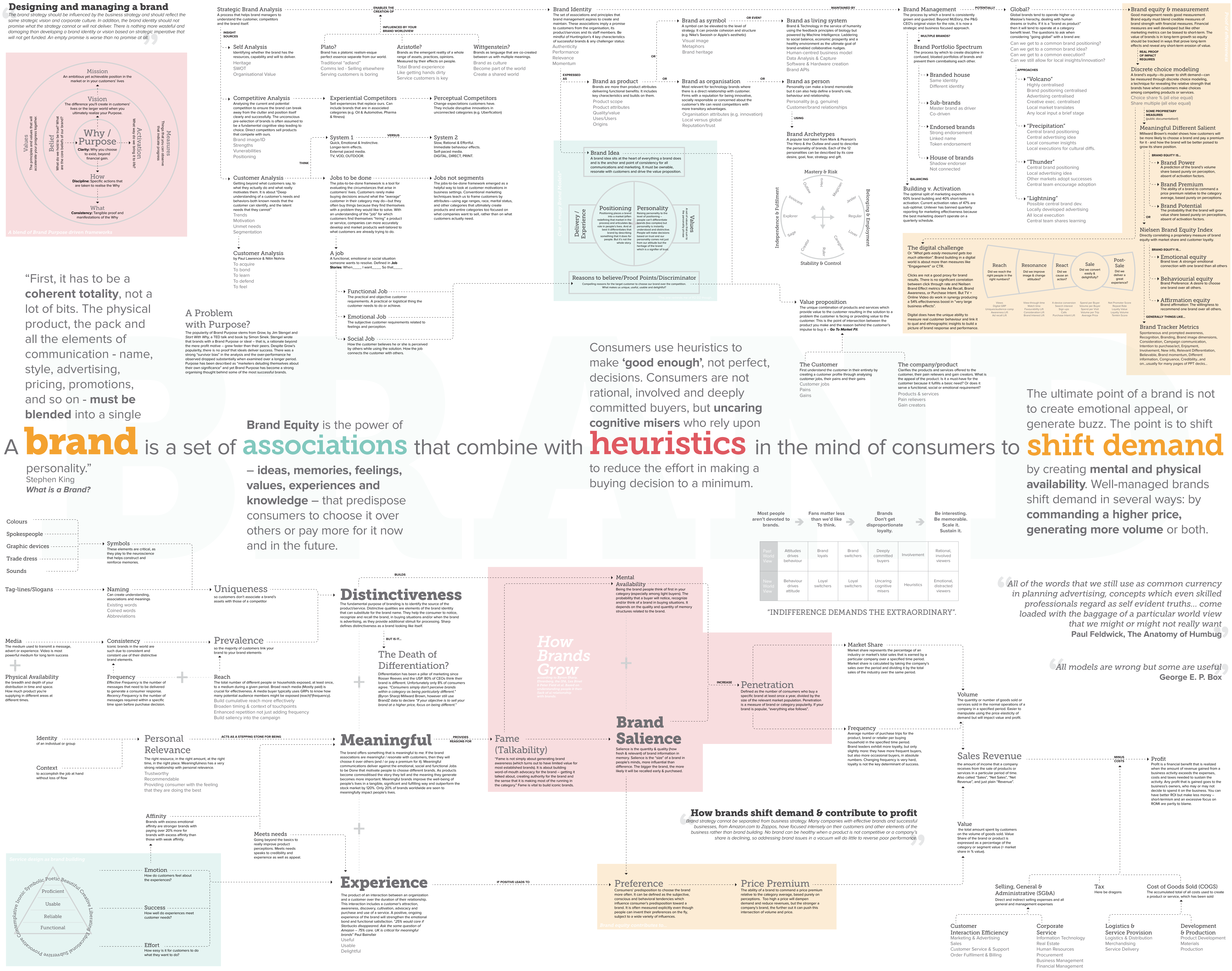
minimal example — interactive, exploratory communication for Lyft's marketing executive

minimal example, *for what things are a marketing executive responsible?*



— Carr, David J. *What Value Do You Create? Marketings 3 Types of Value.* 2019.

minimal example, *for what things are a marketing executive responsible?*



— Carr, David J. *A map of modern brand building*. 2016.

Data drives marketing, can reveal biases

This marketing director knows that marketing is data-driven. Further, "Data can often show the basis for our biases and intuition."

Limitations in data need to be understood, addressed

He also understands issues with use of data:

Sources of unique data can be limited.

Data is often corrupted, unhygienic, or mistransformed when converting to information.

Data is often guesstimated, panel-skewed, inaccurate, and not proven, but at the same time "treated as gospel."

Measured data is only part of the story; things that go unmeasured are important and can change what the total information mean from a business standpoint.

Use of data is about truth and trust, requires openness about source and methodology

"The debate about the use of data in marketing and communications is really a debate about truth and trust, the two biggest issues in the world today."

— Carr, David J. *Data is the new oil*. 2018.

minimal example, *what's the background of the head marketing executive for bikes at Lyft (CitiBike)?*

Message ···

Azmat Ali · 3rd in

Head of Rider Product Marketing at Lyft

San Diego, California, United States · 500+ connections ·

Contact info

Lyft

Imperial College London

About

Results driven executive with over 25 years experience in leading start up, high growth and mature organizations through rapid growth and change worldwide. Consistently successful in identifying and developing growth opportunities, achieving operational results, building highly effective organizations and collaborating across organizational boundaries. Expertise includes management and diffusion of innovation, customer insights that drive action, consumer, SMB and enterprise customer segments, retail channel and international markets

Specialties: Strategic Marketing, Developing and delivering growth strategies, Management of Innovation, Consumer Marketing. Growth mindset. Innovation Funnel Management. New Category Creation. Excellent people and business management. Digital Marketing. PPC SEO and full funnel optimization. Data Analytics

Experience

Head of Rider Product Marketing
Lyft · Full-time
May 2020 – Present · 11 mos
San Francisco Bay Area

HP
3 yrs 11 mos

Head of Innovation and Incubation
Nov 2019 – May 2020 · 7 mos

Global Head, Consumer Product and Segment Marketing
Jul 2016 – Nov 2019 · 3 yrs 5 mos
Palo Alto

VP Brand and Marketing
Evernote
Feb 2016 – Jul 2016 · 6 mos
San Francisco Bay Area

Chief Marketing Officer
Avegant
Mar 2015 – Mar 2016 · 1 yr 1 mo
San Francisco Bay Area

Vice President Marketing
Lytro Inc.
Jan 2014 – Mar 2015 · 1 yr 3 mos
Mountain View, California

Show 5 more experiences

Education

Imperial College London
MBA, Marketing and Innovation
1990 – 1991

Kingston University
Bachelor of Engineering - BE, Electronic Systems Engineering , Honours
1986 – 1990

Explore conditions of January, CitiBike ridership for segmentation and targeting.

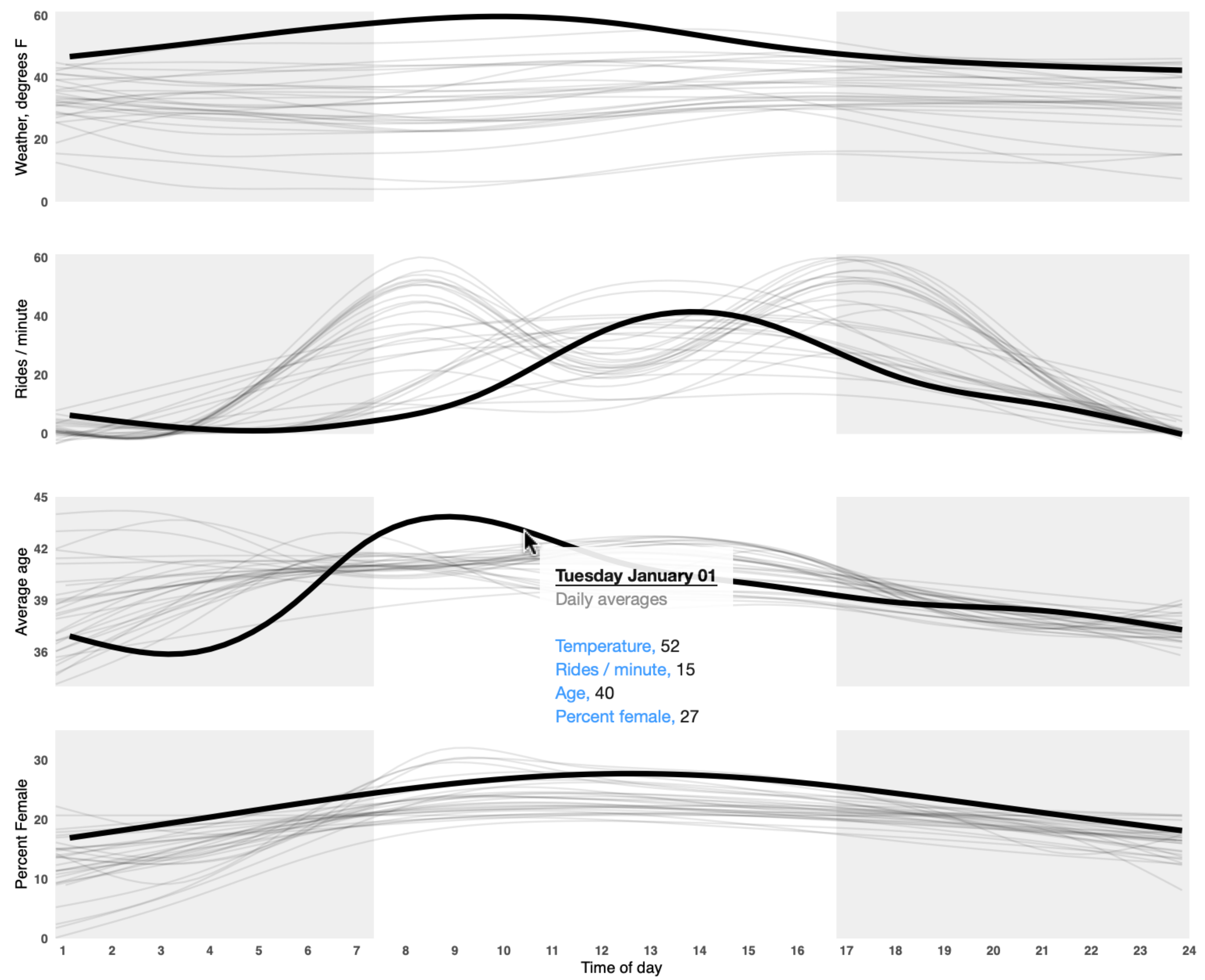
How to explore : **Hovering** over any line will link the four variables — *weather*, *rides per minute*, *average age*, and *percent female* — and identify the *date* and *weekday* selected. **Quick takes :** The morning and evening weekday peak commutes stand out from weekends, of course. But more interestingly, on New Year's Day, our warmest of the month, you'll find a significant swing in average age as night became morning; were our younger commuters out late, sleeping in? Below are ***smoothed functions*** of the data.

Do rider **attributes** correlate with lower usage? Are we missing key target audiences?

Are there better **temperatures** for us to trigger marketing messages to encourage rides?

How can we **segment** our audience to find opportunities for increasing ridership?

Are there better **times of day** for us to trigger marketing messages to encourage rides?



The lines show cubic splines, smoothing variation of each variable over the day. Sources: NYC Open Data, The Open Bus project, and Weather Underground. 2019 January 1-31. Design and code by Scott Spencer. 2021 March 31.

—Spencer, Scott. 2021 March 31. Interactive version, https://ssp3nc3r.github.io/publications/interactive_minimal_example.html

resources

References

Spencer, Scott. Sec. 3.2-3.3. "Interaction: technologies and tools of interactive data-driven, visual design," and "Interaction: Interactive communication with data-driven graphics." In *Data in Wonderland*. 2021. https://ssp3nc3r.github.io/data_in_wonderland.

Attardi, Joe. *Modern CSS: Master the Key Concepts of CSS for Modern Web Development*, 2020. https://doi.org/10.1007/978-1-4842-6294-8.

Bellamy-Royds, Amelia, Kurt Cagle, and Dudley Storey. *Using SVG with CSS3 and HTML5 : Vector Graphics for Web Design*. O'Reilly, 2018.

Duckett, Jon. *HTML & CSS. Design and Build Websites*. Wiley, 2011; *JavaScript & JQuery: Interactive Front-End Web Development*. Indianapolis, IN: Wiley, 2014.

Fay, Colin, Vincent Guyader, Sebastien Rochette, and Girard Cervan. *Engineering Production-Grade Shiny Apps*. First edition. R Series. Boca Raton: CRC Press, 2021. https://engineering-shiny.org.

Gohel, David, and Panagiotis Skintzos. "*Ggiraph: Make 'ggplot2' Graphics Interactive*." Manual, 2021. https://davidgohel.github.io/ggiraph.

Hohman, Fred, Matthew Conlen, Jeffrey Heer, and Duen Chau. "Communicating with Interactive Articles." *Distill* 5, no. 9 (September 11, 2020): 10.23915/distill.00028. https://doi.org/10.23915/distill.00028.

Hullman, Jessica, Steven Drucker, Nathalie Henry Riche, Bongshin Lee, Danyel Fisher, and Eytan Adar. "A Deeper Understanding of Sequence in Narrative Visualization." *IEEE Transactions on Visualization and Computer Graphics* 19, no. 12 (August 2013): 2406–15.

Hullman, Jessica, and Andrew Gelman. "Designing for Interactive Exploratory Data Analysis Requires Theories of Graphical Inference." *Harvard Data Science Review*, no. 3.3 (July 23, 2021). https://doi.org/10.1162/99608f92.3ab8a587.

Hullman, Jessica, and Andrew Gelman. "Challenges in Incorporating Exploratory Data Analysis into Statistical Workflow." *Harvard Data Science Review*, no. 3.3 (July 30, 2021). https://doi.org/10.1162/99608f92.9d108ee6.

Janert, Philipp K. *D3 for the Impatient: Interactive Graphics for Programmers and Scientists*. First edition. Sebastopol, CA: O'Reilly Media, Inc, 2019.

Kotzé, Paula, INTERACT, and International Federation for Information Processing, eds. "Storytelling in Visual Analytics Tools for Business Intelligence." In *Human-Computer Interaction: INTERACT 2013*; 14th IFIP TC 13 International Conference, Cape Town, South Africa, September 2 - 6, 2013; Proceedings. Pt. 3: ..., 280–97. Lecture Notes in Computer Science 8119. Heidelberg: Springer, 2013.

McKenna, S., N. Henry Riche, B. Lee, J. Boy, and M. Meyer. "Visual Narrative Flow: Exploring Factors Shaping Data Visualization Story Reading Experiences." *Computer Graphics Forum* 36, no. 3 (June 2017): 377–87. https://doi.org/10.1111/cgf.13195. Supplemental material: https://narrative-flow.github.io/

Murray, Scott. *Interactive Data Visualization for the Web*. Second. *An Introduction to Designing with D3*. O'Reilly, 2017.

Reas, Casey, and Ben Fry. *Processing A Programming Handbook for Visual Designers and Artists*. Second. The MIT Press, 2014.

Sarikaya, Alper, Michael Correll, Lyn Bartram, Melanie Tory, and Danyel Fisher. "What Do We Talk About When We Talk About Dashboards?" *IEEE Transactions on Visualization and Computer Graphics* 25, no. 1 (January 2019): 682–92. https://doi.org/10.1109/TVCG.2018.2864903.

Schneiders, Pascal. "What Remains in Mind? Effectiveness and Efficiency of Explainers at Conveying Information." *Media and Communication* 8, no. 1 (March 18, 2020): 218–31. https://doi.org/10.17645/mac.v8i1.2507.

Sievert, Carson. *Interactive Web-Based Data Visualization with R, Plotly, and Shiny*. Boca Raton, FL: CRC Press, Taylor and Francis Group, 2020. https://plotly-r.com.

Vaidyanathan, Ramnath, Yihui Xie, JJ Allaire, Joe Cheng, Carson Sievert, and Kenton Russell. "*Htmlwidgets: HTML Widgets for r.*" Manual, 2020. https://CRAN.R-project.org/package=htmlwidgets; main introduction: http://www.htmlwidgets.org.