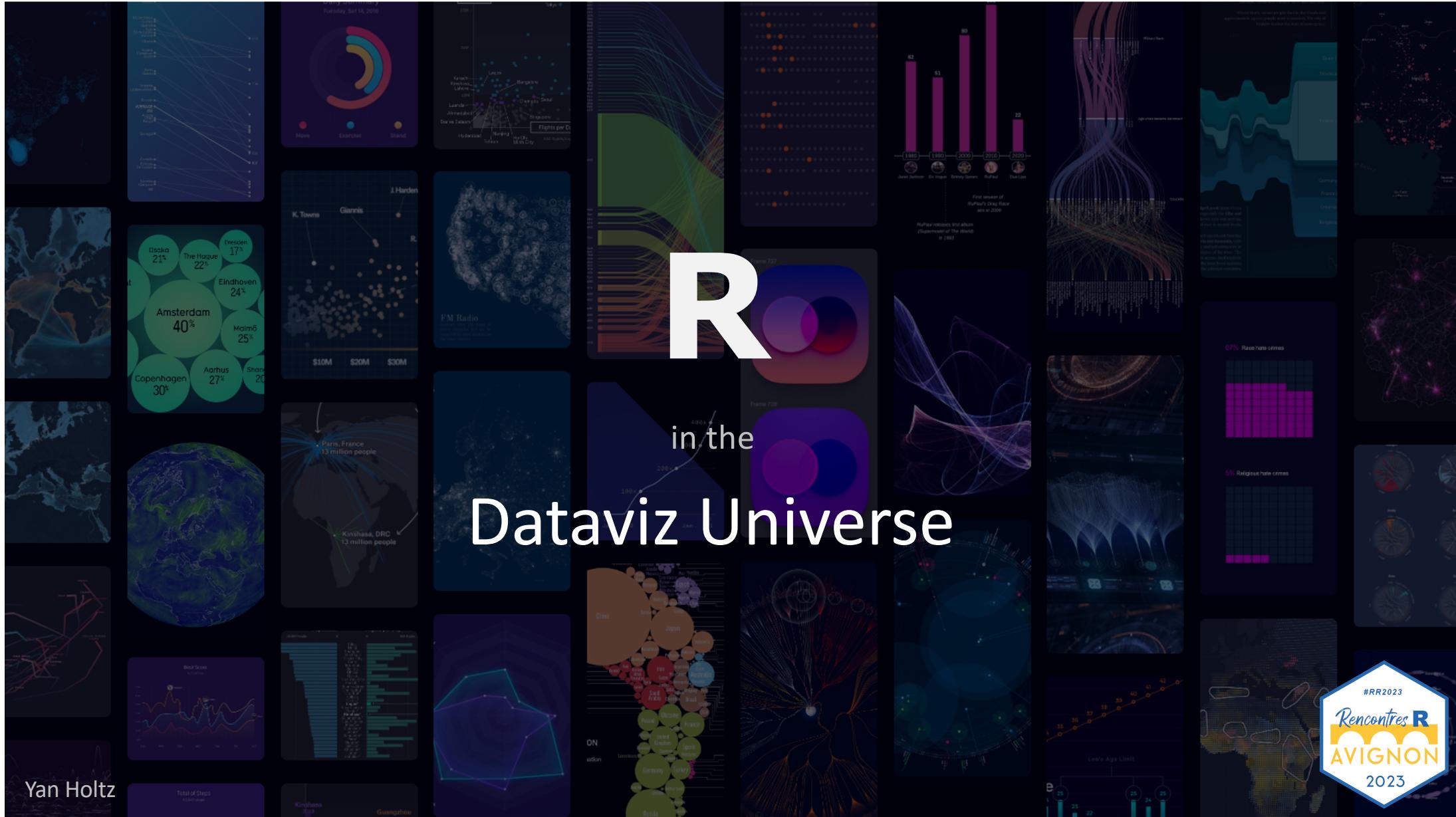


R in the Dataviz Universe



Yan Holtz

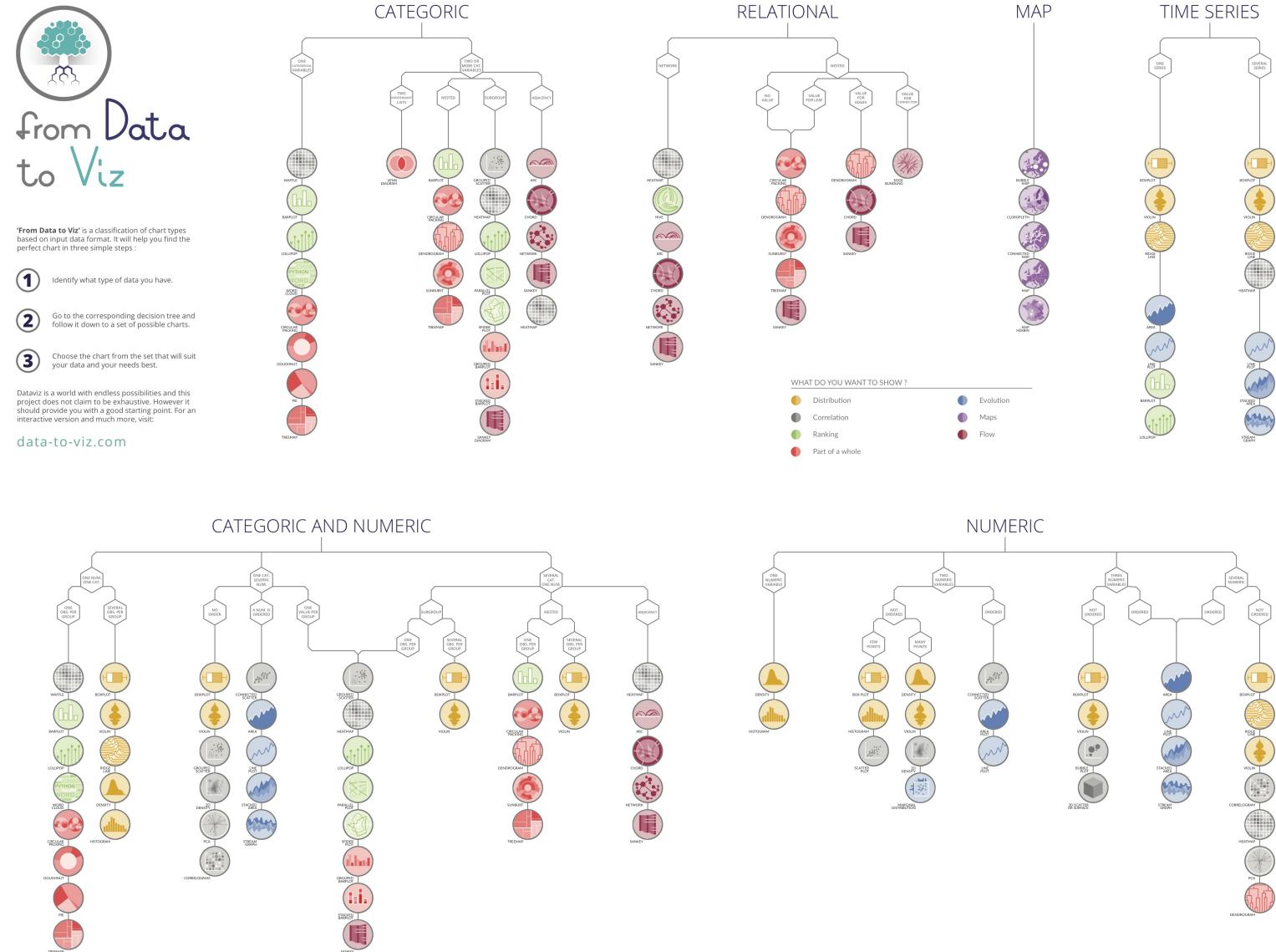
Hi, I'm Yan !

And I ❤️ data visualization



A universe of graph types

Data-to-viz.com





I ❤️ tech



R
Python
Data wrapper
Html
I ❤ Tech
D3.js
Shiny
React
Canvas

Vega

Tailwind

Leaflet

Plotly

Next.js

Illustrator

SVG

React-spring

R

Python

Data wrapper

Html

Tech

D3.js

Shiny

React

Canvas

Gatsby

ggplot2

Tidyverse

R Studio

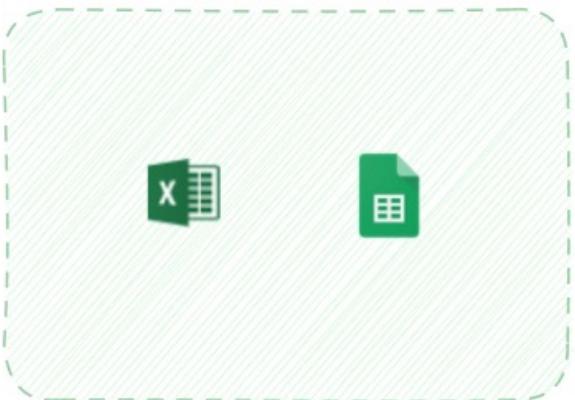
VS code

Github

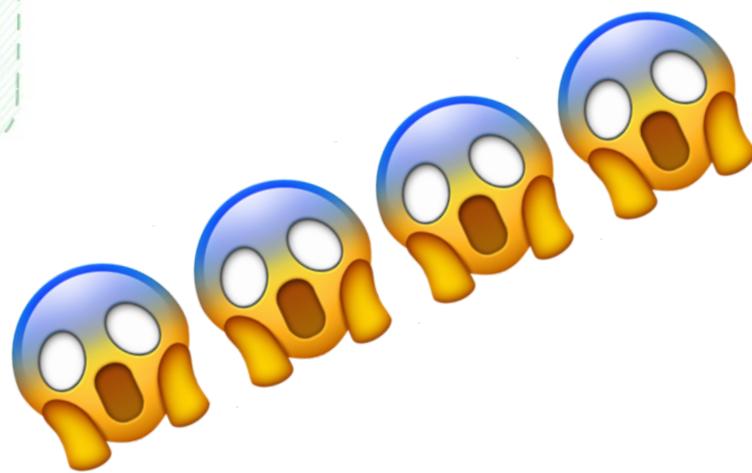
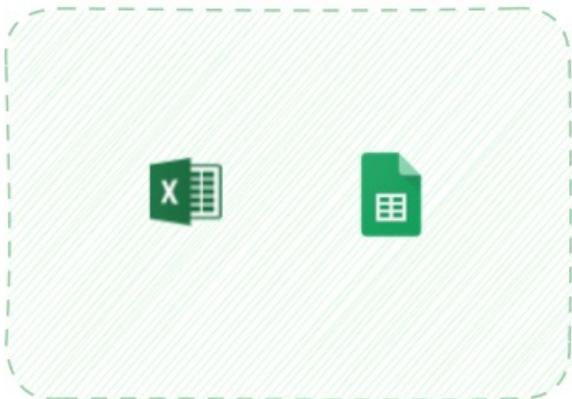
Bokeh

Seaborn

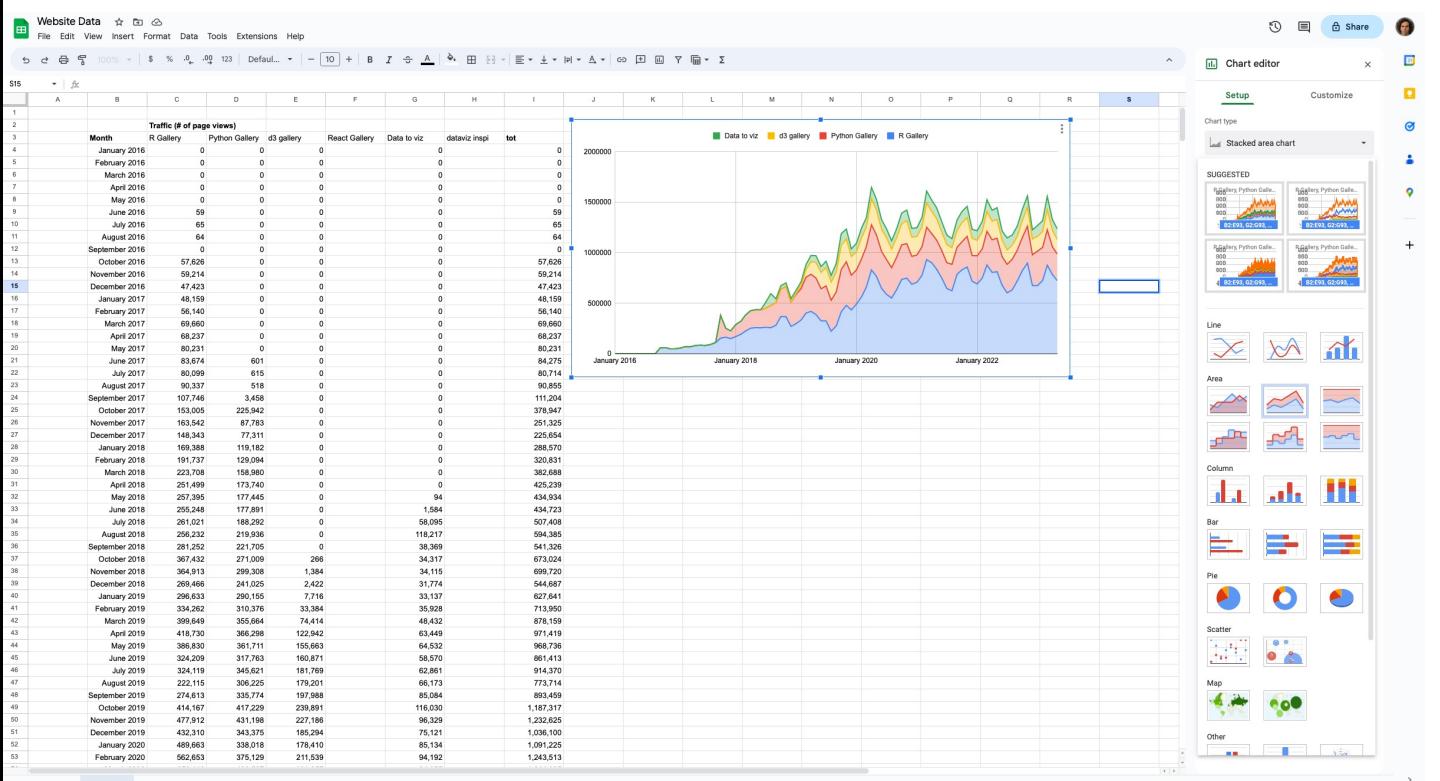
Spreadsheet



Spreadsheet



- Quick
- Easy



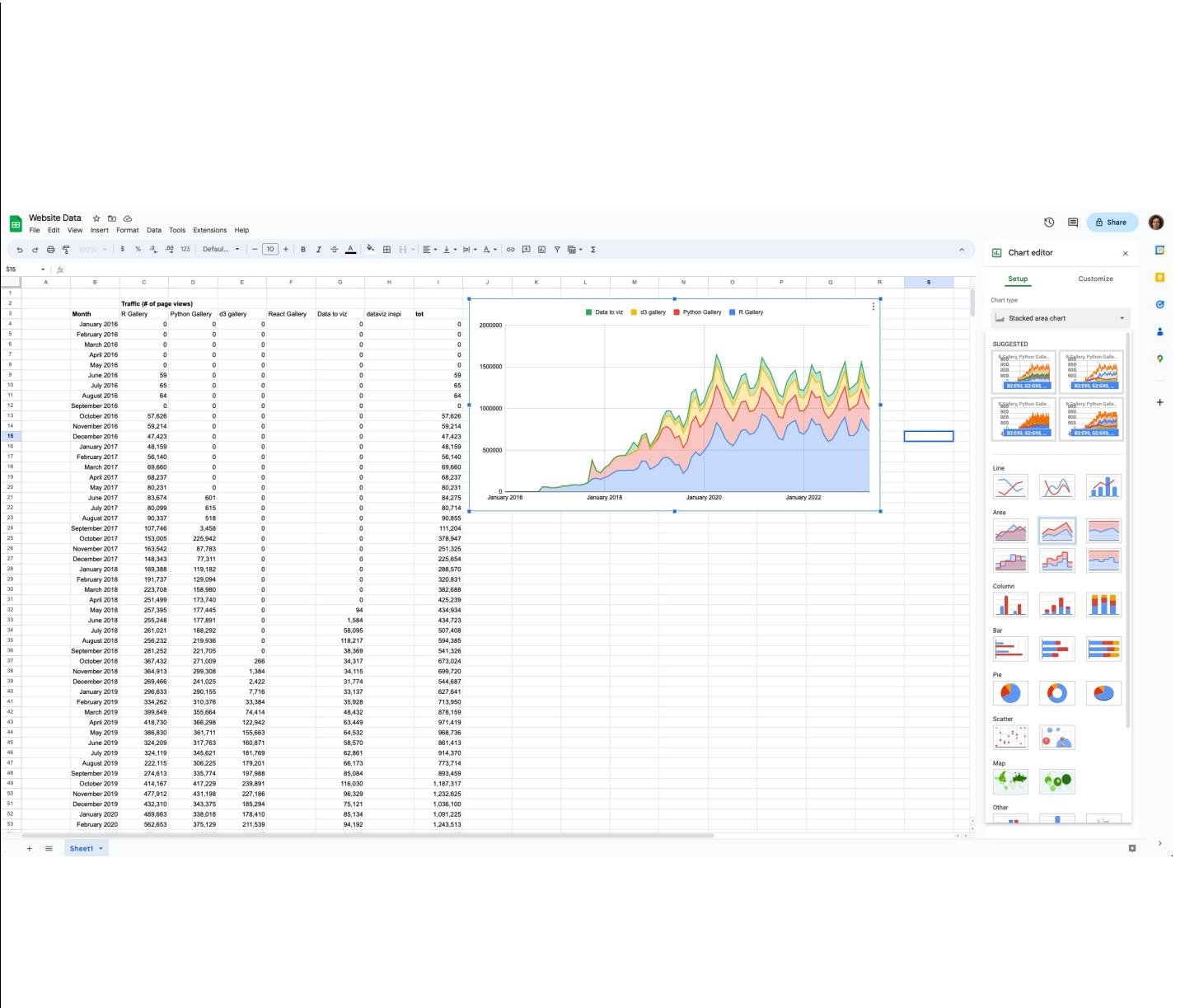
✓ Quick

✓ Easy

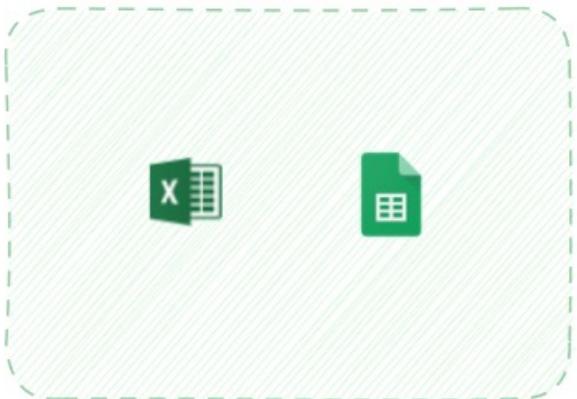
✗ Graph defaults
✗ Chart types
✗ Interactivity

✗ Data wrangling
✗ Stats
✗ Data size
✗ Pipeline automation

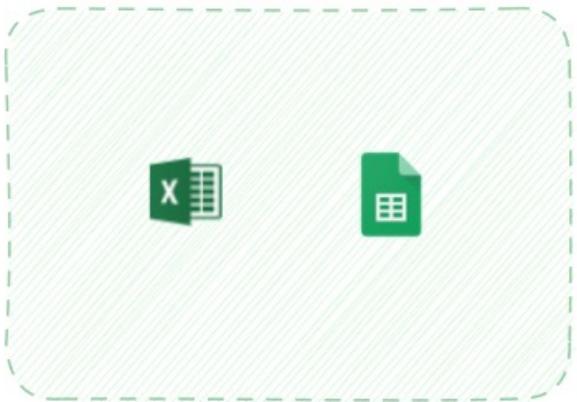
✗ Error prone
✗ Collaboration
✗ Versioning



Spreadsheet



Spreadsheet

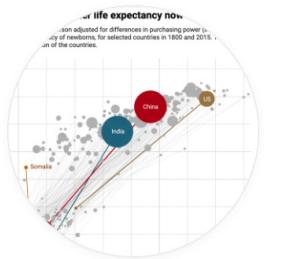


No Code tools



**Charts**

Use one of 19 interactive and responsive chart types ranging from bubble charts to treemaps.

**Maps**

Three interactive and responsive map types let you create anything from choropleth maps to location-based dashboards.

**Tables**

Responsive data tables allow for lots of styling options and let you include bar, line, or gauge charts directly in the table.



Datawrapper

[+ Create new ...](#)[Sign in](#)

Welcome to Datawrapper 😊! Learn how to create your first chart.

1 Upload Data**2** Check & Describe**3** Visualize**4** Publish & Embed

How do you want to upload your data?

Copy & paste data table

XLS/CSV upload

Connect Google Sheet

Link external data

Copy & paste your data

Select your data (including header row/column) in Excel or LibreOffice and paste it in the text field. You can also upload a CSV or Excel file from your computer.

If you just want to try Datawrapper, here's a list of some example datasets you can use:

Charts

Use one of 19 interactive and responsive chart types ranging from simple bars and lines to complex dashboards.

Maps

Three interactive and responsive map types let you create anything from location maps to choropleth maps.

Tables

Responsive data tables allow for lots of styling options and let you include bar, volume & line charts, heatmaps,

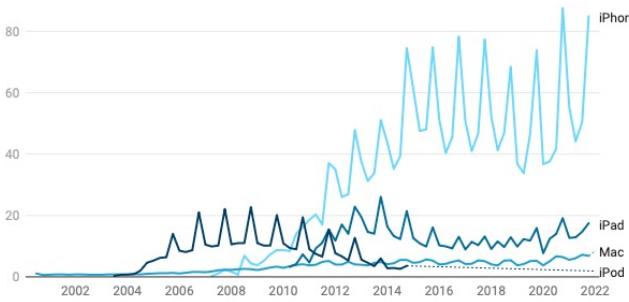
Welcome to Datawrapper! 😊 Learn how to create your first chart.

- 1 Upload Data**
- 2 Check & Describe**
- 3 Visualize**
- 4 Publish & Embed**

Chart type [Refine](#) [Annotate](#) [Layout](#)

 Bar Chart	 Stacked Bars	 Grouped Bars	 Split Bars
 Bullet Bars	 Column Chart	 Stacked Column Chart	 Grouped Column Chart
 Lines	 Area Chart	 Scatter Plot	 Table
 Dot Plot	 Range Plot	 Arrow Plot	 Pie Chart
 Donut Chart	 Multiple Pies	 Multiple Donuts	 Election Donut

Selling it!
Apple sales by product in shipped units, 2000 to 2017.

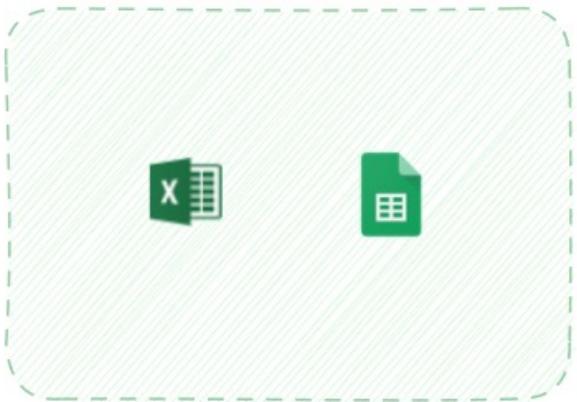


Apple stopped to report iPod sales at the end of 2014.
Source: [Apple - Get the data](#) • Created with [Datawrapper](#)

PREVIEW

Size (px) Colorblind check Dark Mode

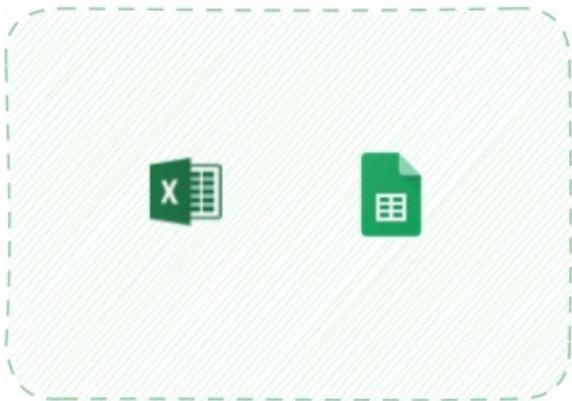
Spreadsheet



No Code tools



Spreadsheet



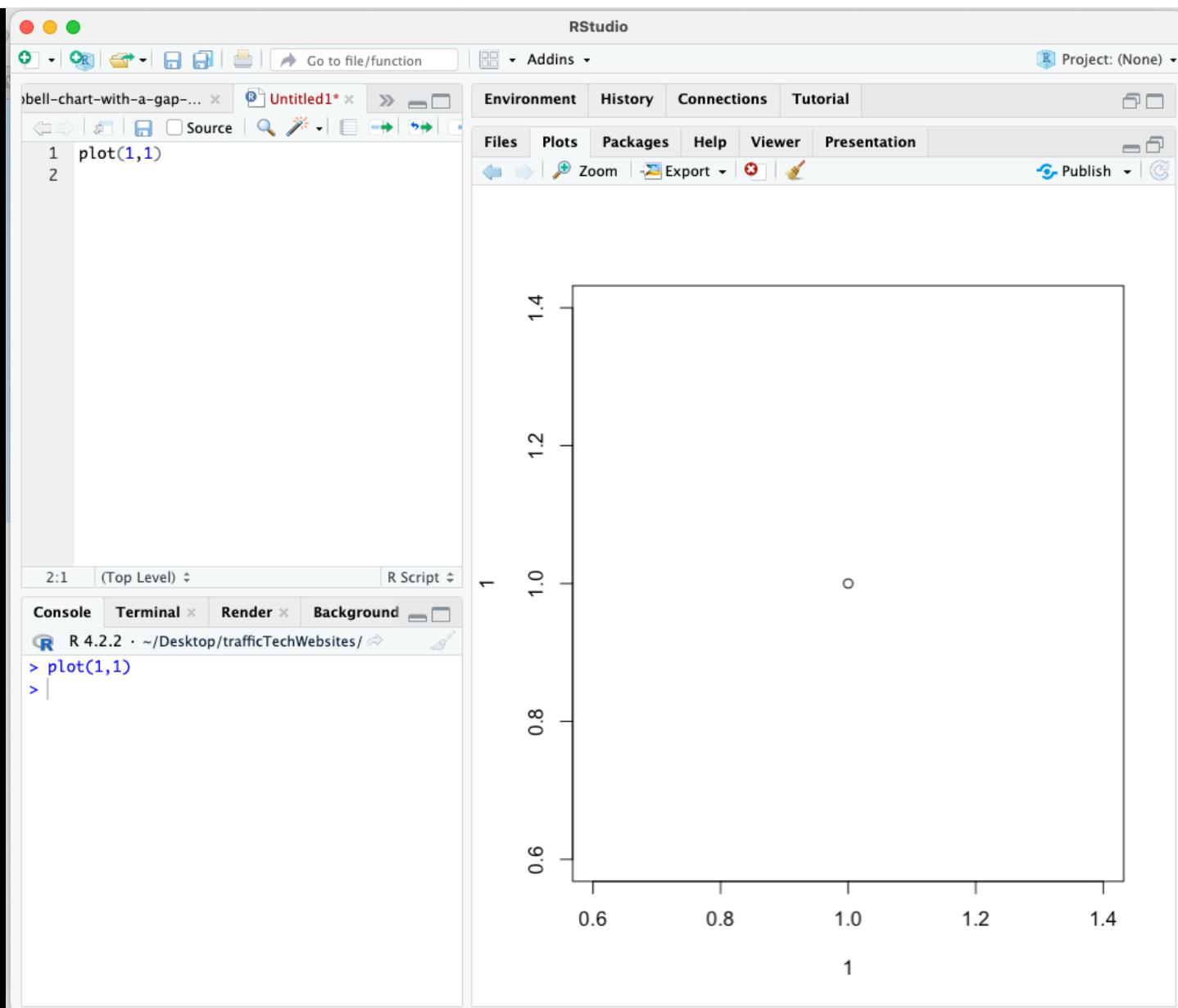
Data Analytics



No Code tools



1995



1995



```
x <- runif(300, min=-10, max=10)
y <- 0.1*x^3 - 0.5 * x^2 - x + 10 + rnorm(length(x),0,8)

# plot of x and y :
plot(x,y,col=rgb(0.4,0.4,0.8,0.6),pch=16 , cex=1.3)

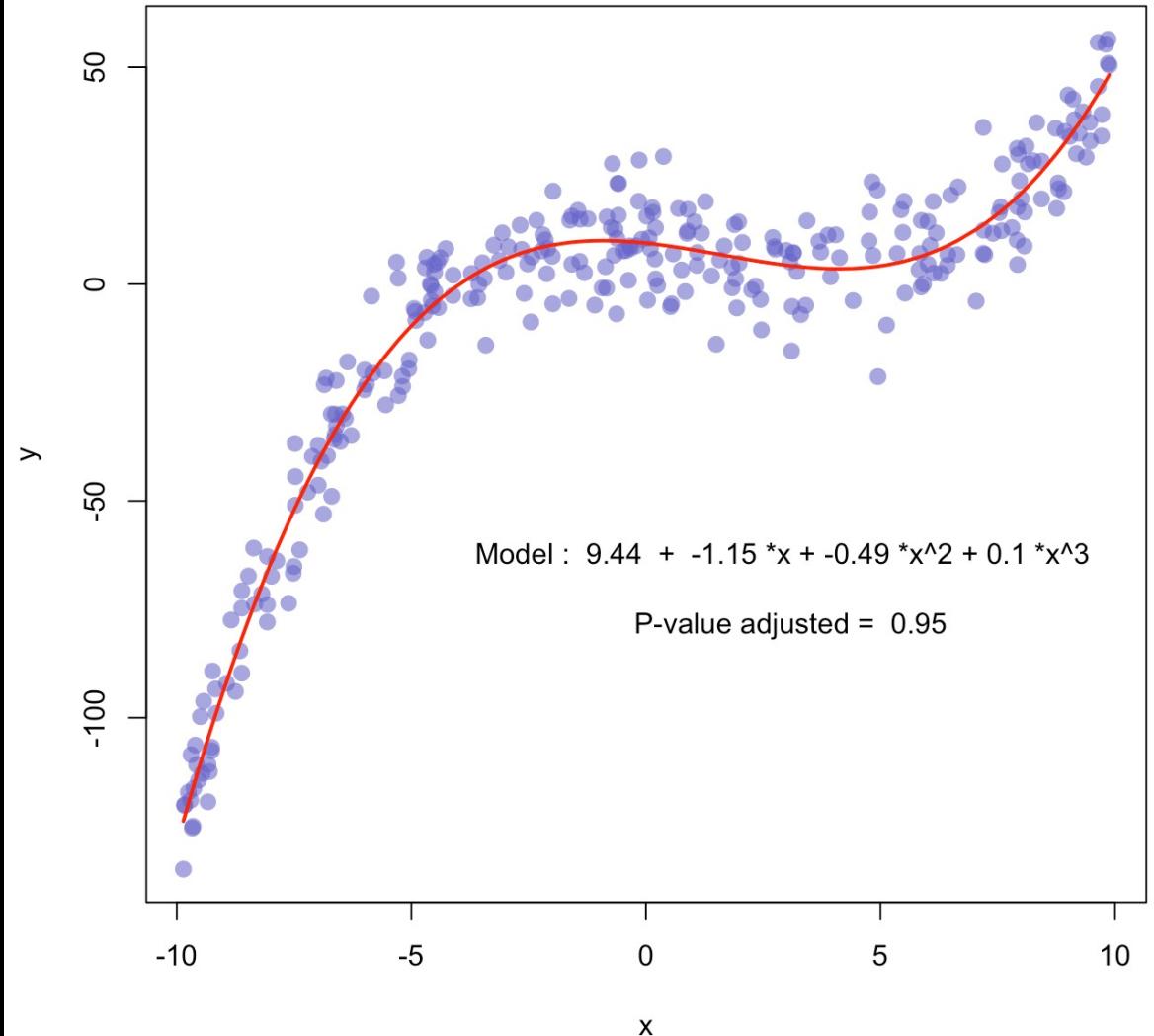
# Can we find a polynome that fit this function ?
model <- lm(y ~ x + I(x^2) + I(x^3))

# Prediction using model
myPredict <- predict( model )
ix <- sort(x,index.return=T)$ix
lines(x[ix], myPredict[ix], col=2, lwd=2 )

# I add the features of the model to the plot
coeff <- round(model$coefficients , 2)
text(3, -70 , paste("Model : ",coeff[1] , " + " , coeff[2] ,
" *x" , "+" , coeff[3] , " *x^2" , "+" , coeff[4] , " *x^3" ,
"\n\n" , "P-value adjusted =",
",round( summary(model)$adj.r.squared,2)))
```

Base R

😊 but 😞



● ● ●

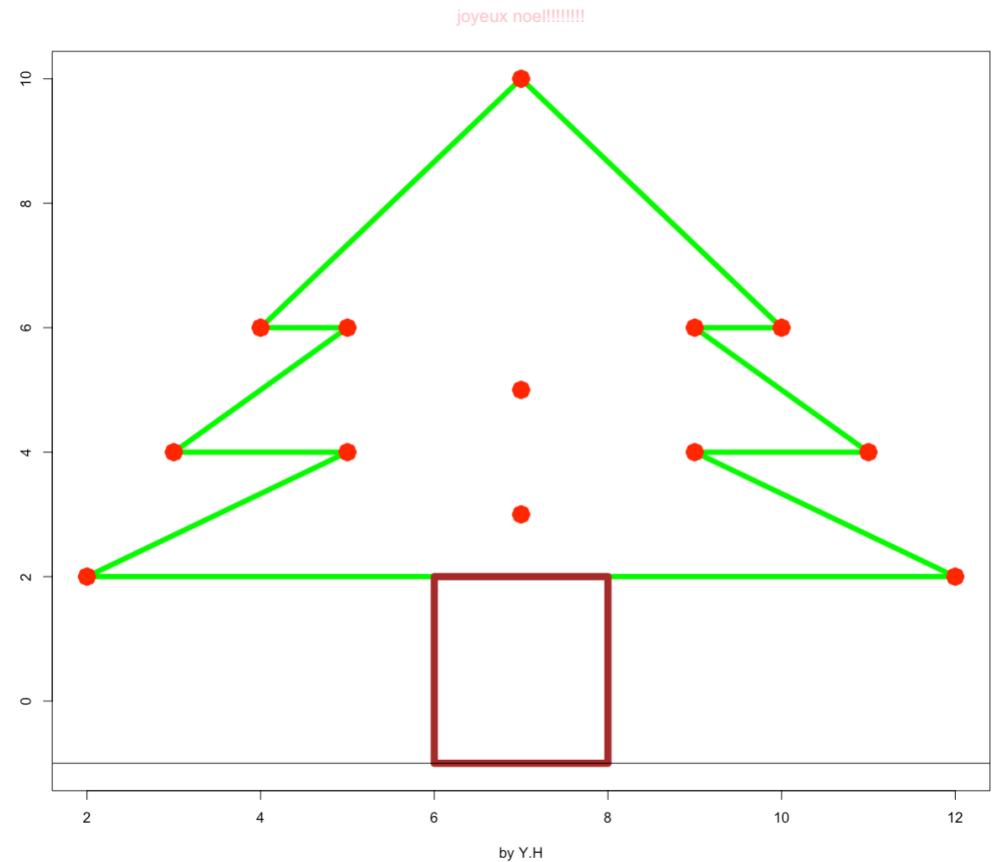
```
x=c(8,12,9,11,9,10,7,4,5,3,5,2,6,6)
y=c(2,2,4,4,6,6,10,6,6,4,4,2,2,-1)
z=c(6,8,8,6,6)
j=c(-1,-1,2,2,-1)

a=c(12,9,11,9,10,7,4,5,3,5,2,7,7)
b=c(2,4,4,6,6,10,6,6,4,4,2,3,5)

d=c(1,13)
e=c(-1,-1)

plot(x,y,type="l",col="green",xlab="by Y.H",lwd=6,main="joyeux
noel!!!!!!",col.main="pink",font.main=9,ylab="APIMET 2011" )

points(z,j,type="l",col="brown",lwd=8)
points(a,b,lwd=10,col="red",pch=8)
points(d,e,type="l")
```



r-graph-gallery.com

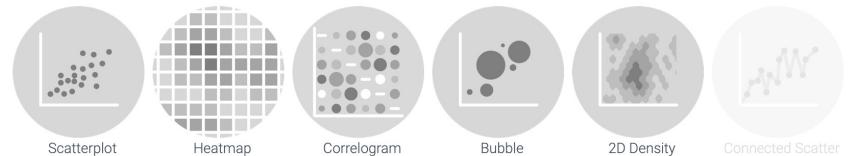
- ✓ 52 chart types
- ✓ ~ 400 chart examples (and growing)
- ✓ 800k visits / months

The R Graph Gallery

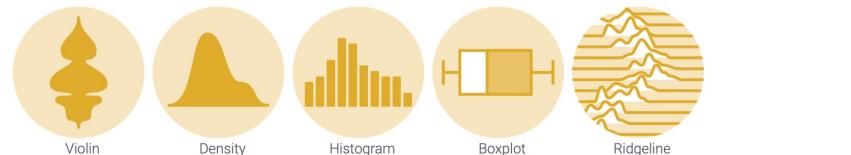


Welcome to the R graph gallery, a collection of charts made with the [R programming language](#). Hundreds of charts are displayed in several sections, always with their reproducible code available. The gallery makes a focus on the tidyverse and [ggplot2](#). Feel free to suggest a chart or [report a bug](#); any feedback is highly welcome! Stay in touch with the gallery by following it on [Twitter](#) or by subscribing to the [newsletter](#).

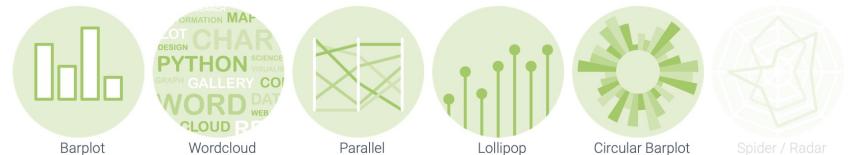
Correlation



Distribution



Ranking



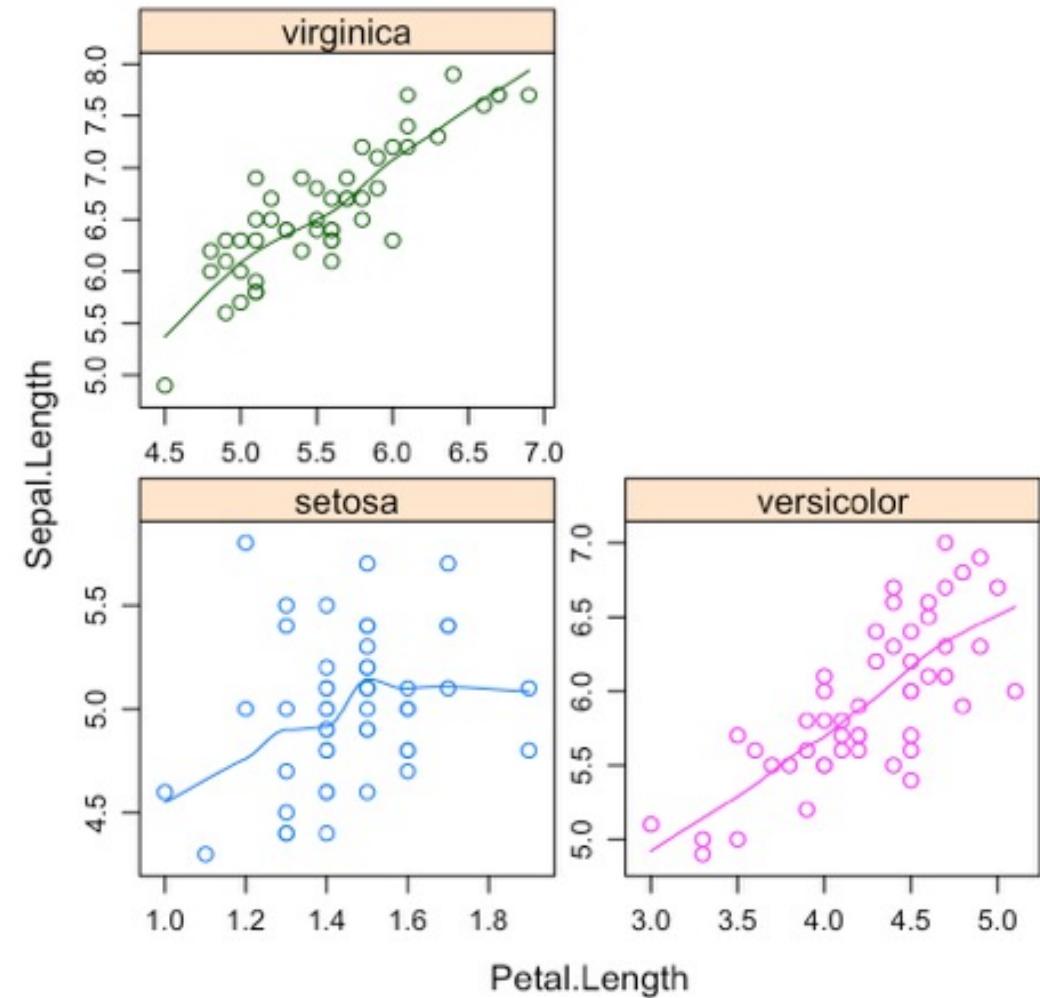
Part Of A Whole



2001

Lattice

New API
Easy small multiples

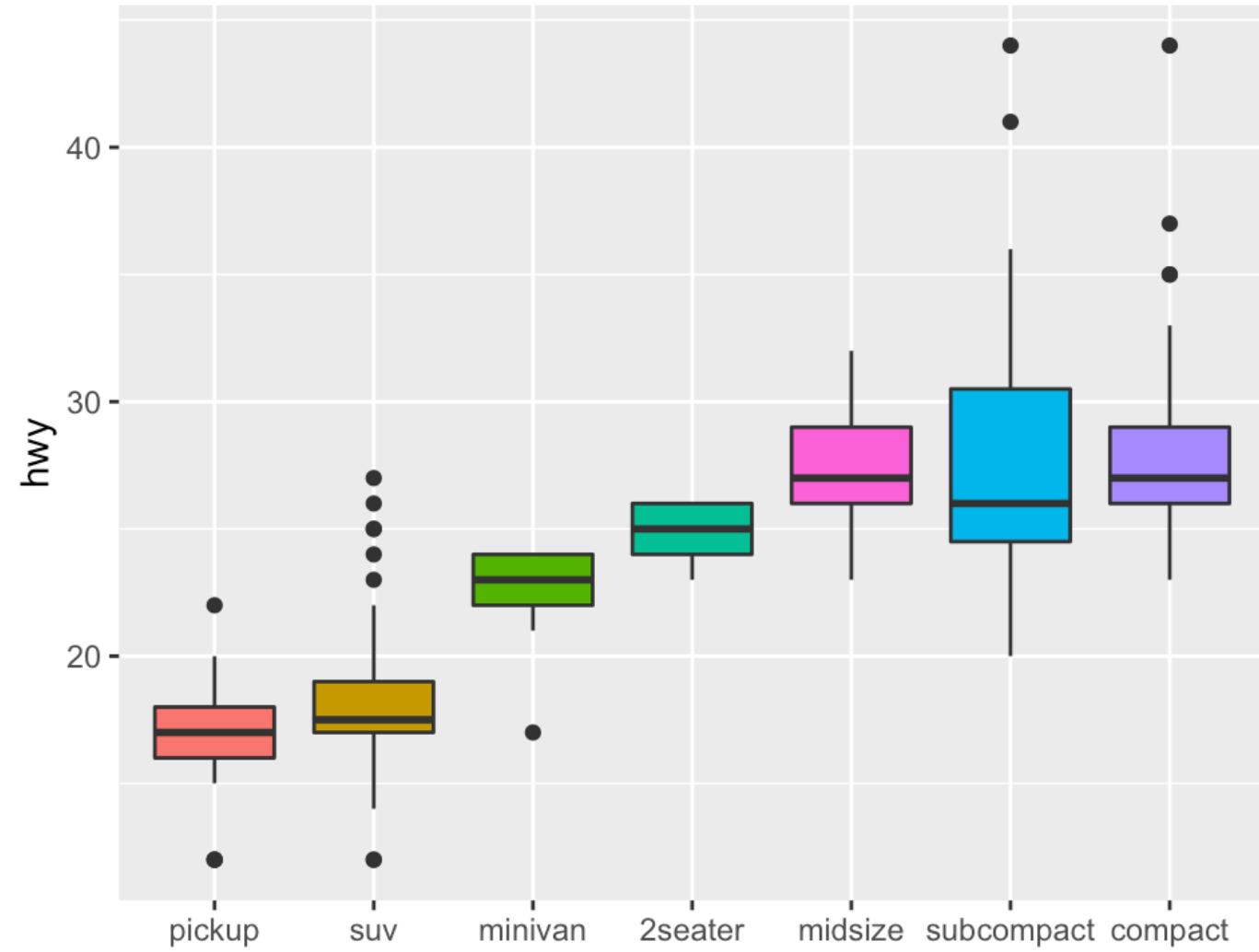


2007



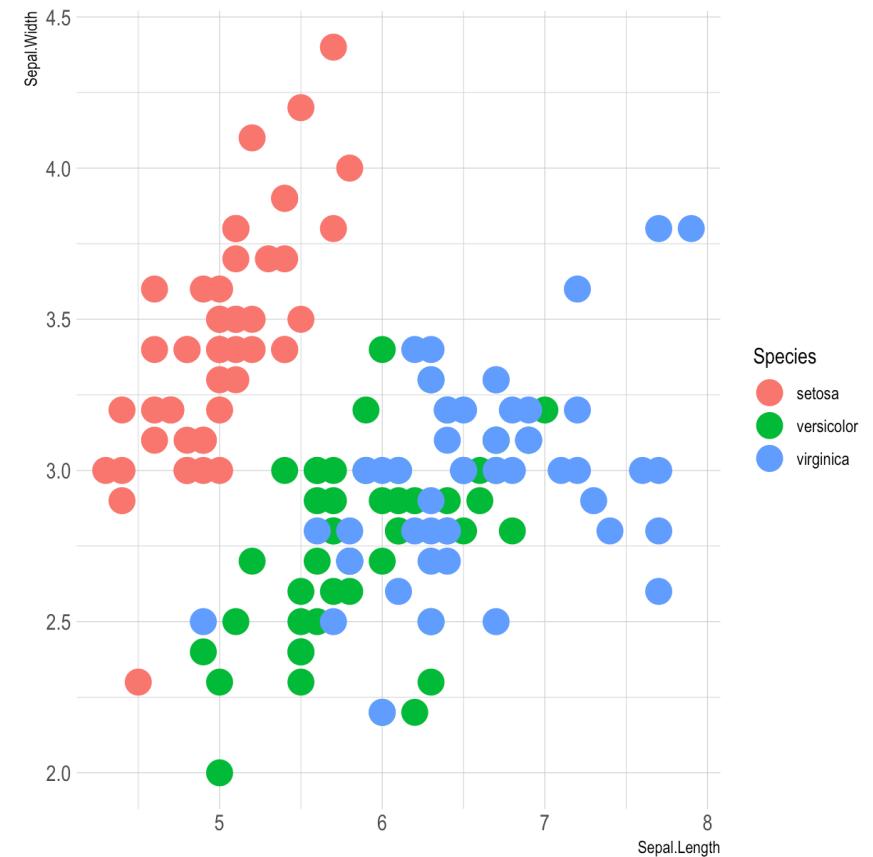
ggplot2

A revolution



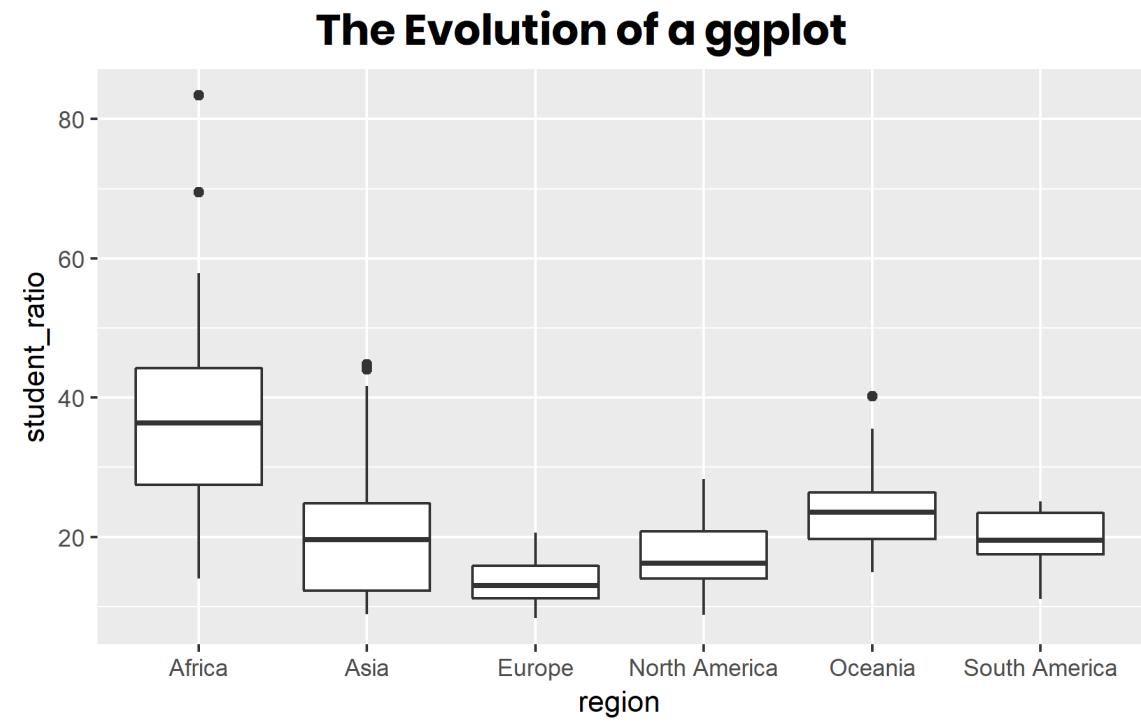
```
● ● ●  
# packages  
library(ggplot2)  
library(hrbrthemes)  
  
# A basic scatterplot with color depending on Species  
ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width, color=Species)) +  
  geom_point(size=6) +  
  theme_ipsum()
```

- ✓ Consistent Grammar
- ✓ Documentation
- ✓ Good default
- ✓ Concise
- ✓ Flexible
- ✓ Themes
- ✓ Community
- ✓ Small Multiples



Customization

Cedric Scherer



Ggplot2 extensions

exts.ggplot2.tidyverse.org

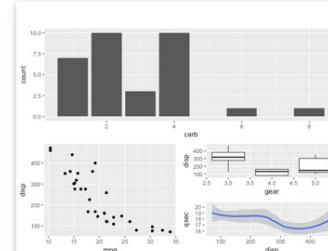
ggplot2 extensions - gallery

Add Your Extension! exts.ggplot2.tidyverse.org

126 registered extensions available to explore

Sort Text Filter Author Filter CRAN Only

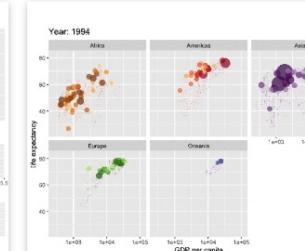
Showing 107 of 126



patchwork 2223

Easy composition of ggplot plots using arithmetic operators

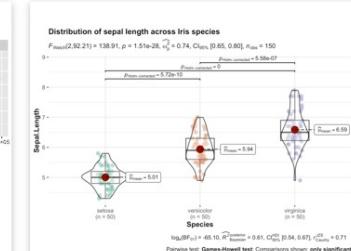
- **author:** thomasasp85
- **tags:** visualization, composition
- **js libraries:**



gganimate 1877

A Grammar of Animated Graphics.

- **author:** thomasasp85
- **tags:** visualization, general
- **js libraries:**



ggstatsplot 1717

'ggstatsplot' provides a collection of functions to enhance 'ggplot2' plots with results from statistical tests.

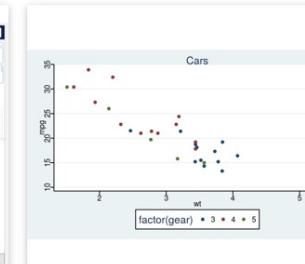
- **author:** IndrajeetPatil
- **tags:** visualization, statistics
- **js libraries:**



esquisse 1649

Explore and Visualize Your Data Interactively with ggplot2

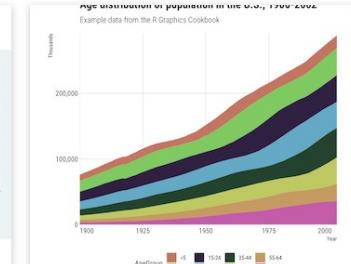
- **author:** dreamsrs
- **tags:** visualization, interface
- **js libraries:**



ggthemes 1242

Some extra geoms, scales, and themes for ggplot.

- **author:** jrnold
- **tags:** visualization, general, themes
- **js libraries:**



hrbrthemes 1130

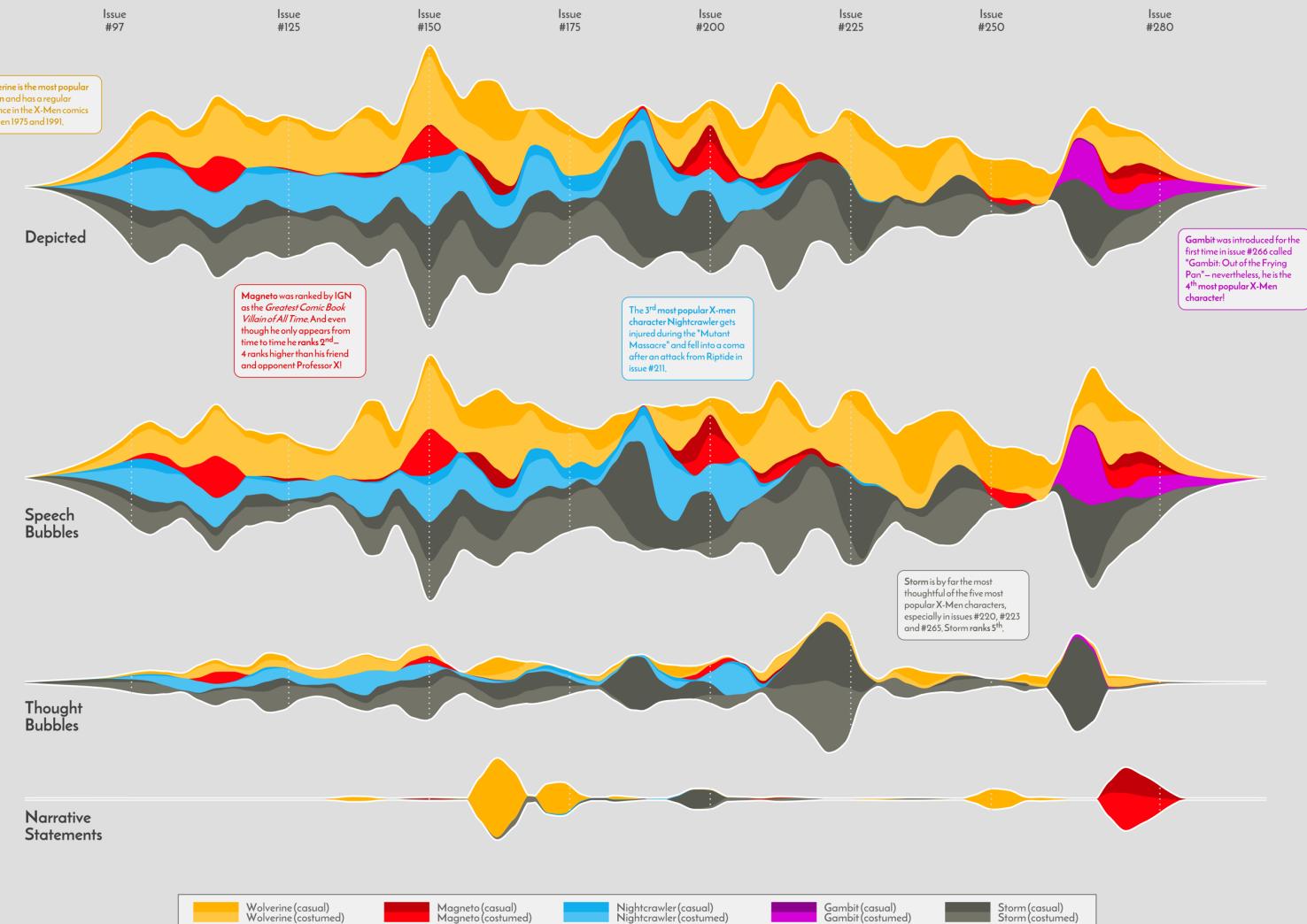
A compilation of extra (ggplot2) themes, scales and utilities, including a spell check function for plot label fields and an overall emphasis on typography.

- **author:** hrbrmstr
- **tags:** theme, typography
- **js libraries:**

Cedric Scherer

X-Men Characters

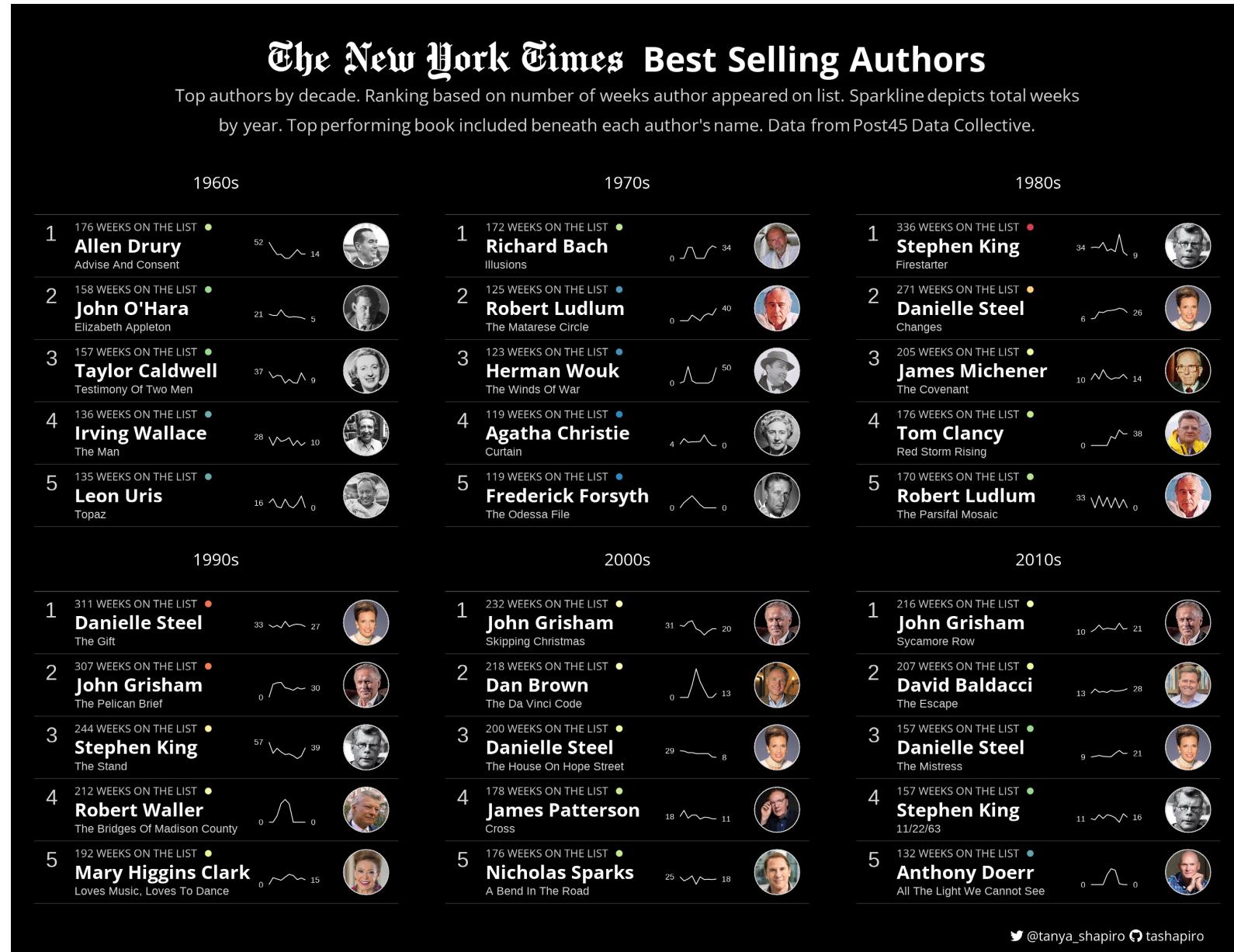
Appearance of the Five Most Popular X-Men Characters in Chris Claremont's **X-MEN** Comics

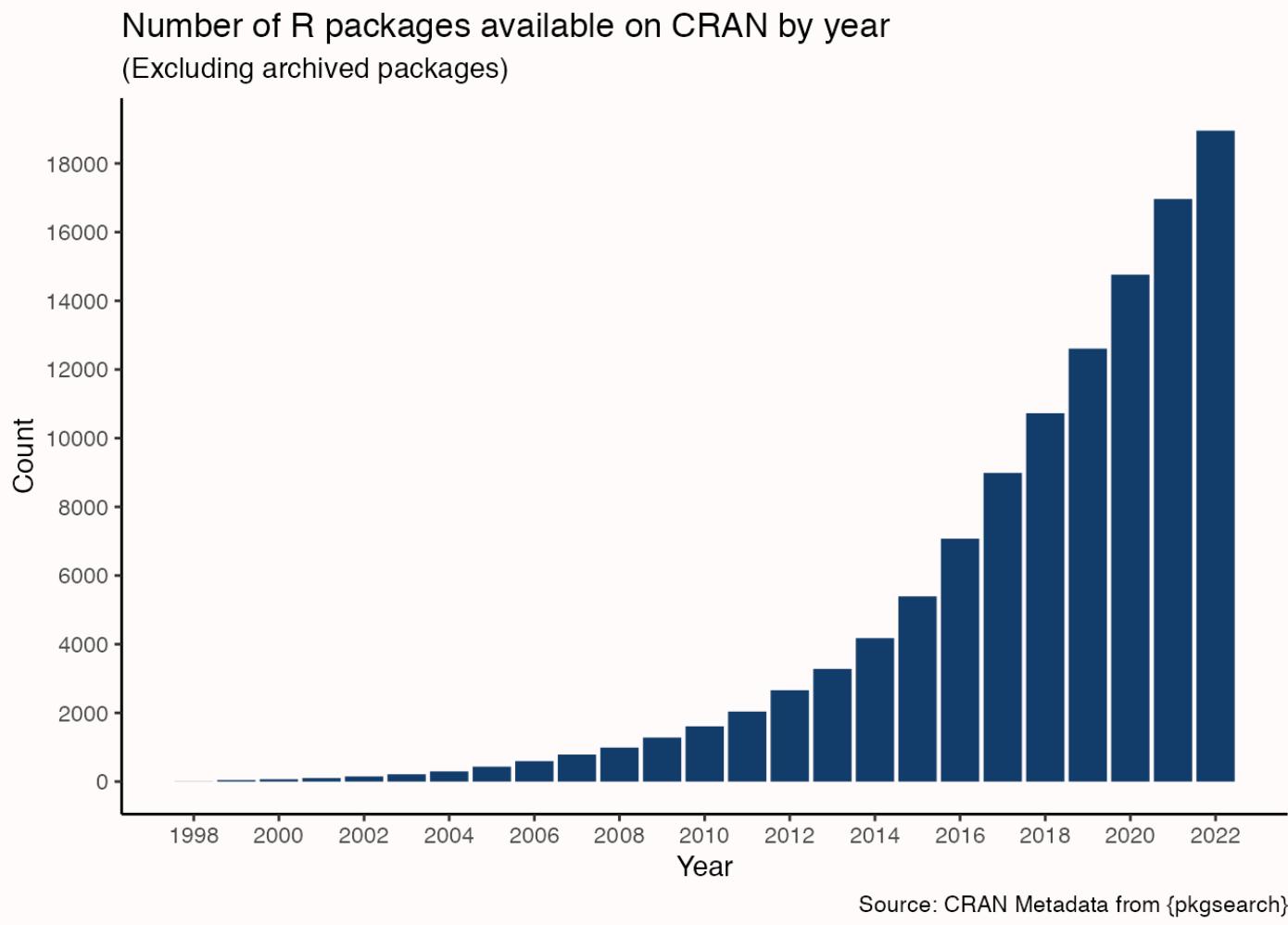


Visualization by Cédric Scherer • Data by Claremont Run Project via Malcom Barret • Popularity Scores by ranker.com • Logo by Comicraft

Tania Shapiro

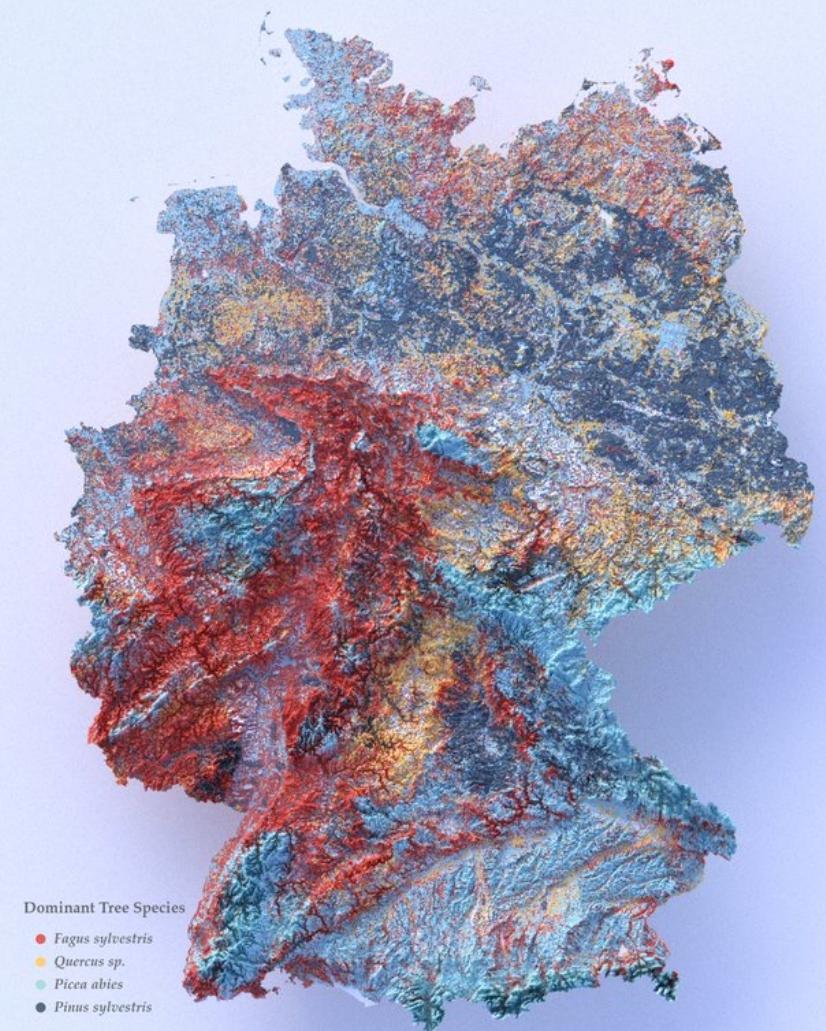
Best selling Authors





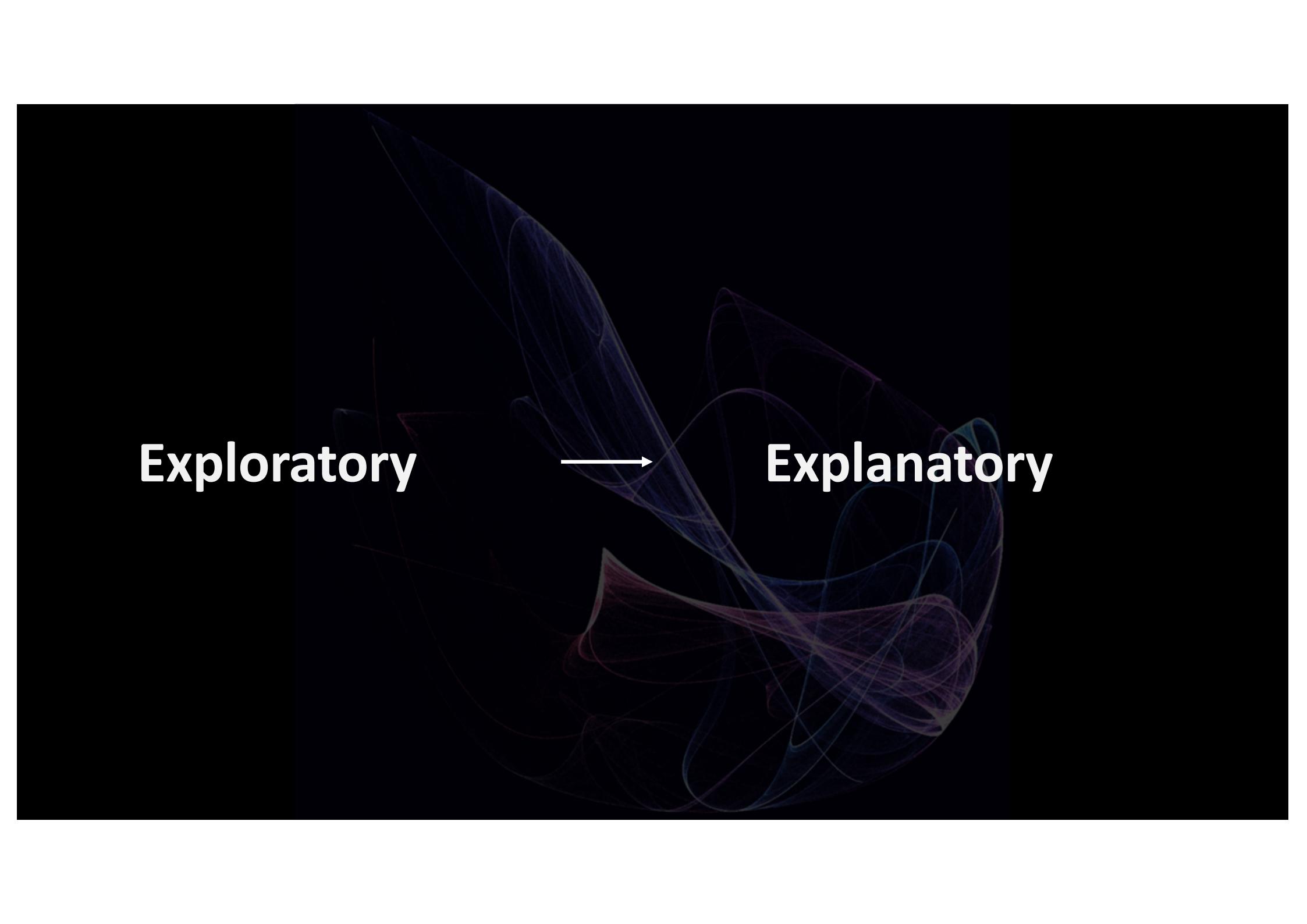
Jens Wiesehahn

Species Map (RayShader)

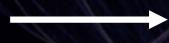


Visualization by:
Jens Wiesehahn, 2022

Data sources:
Elevation data from Copernicus EU-DEM v1.1 Tree species map by Jens Wiesehahn, 2021

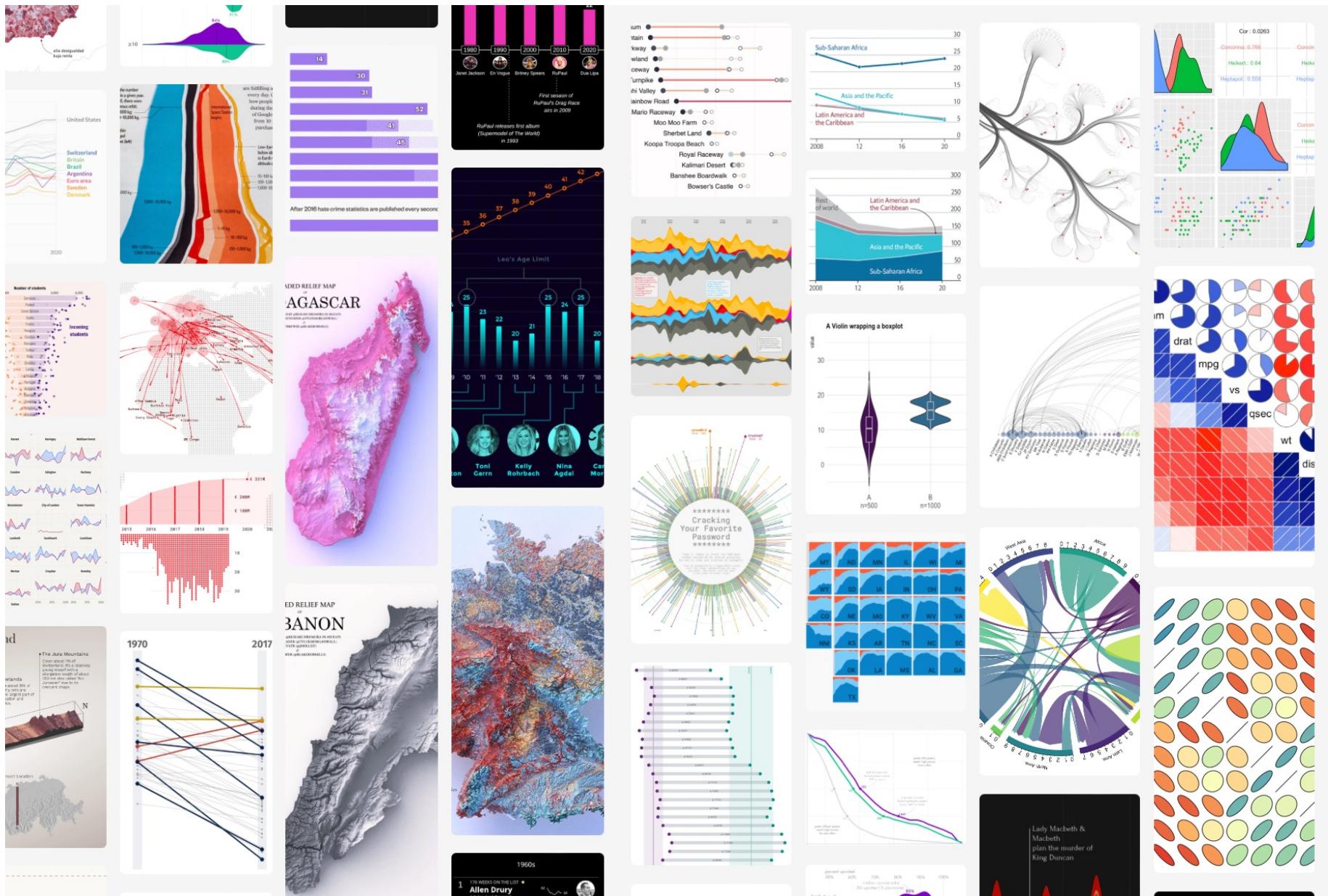
The background of the slide features a dark, abstract design composed of numerous thin, glowing blue and purple lines. These lines form a complex, organic, and somewhat chaotic pattern that resembles a celestial body or a microscopic organism. They are concentrated in two main clusters: one on the left side and another on the right side, with many lines radiating outwards between them.

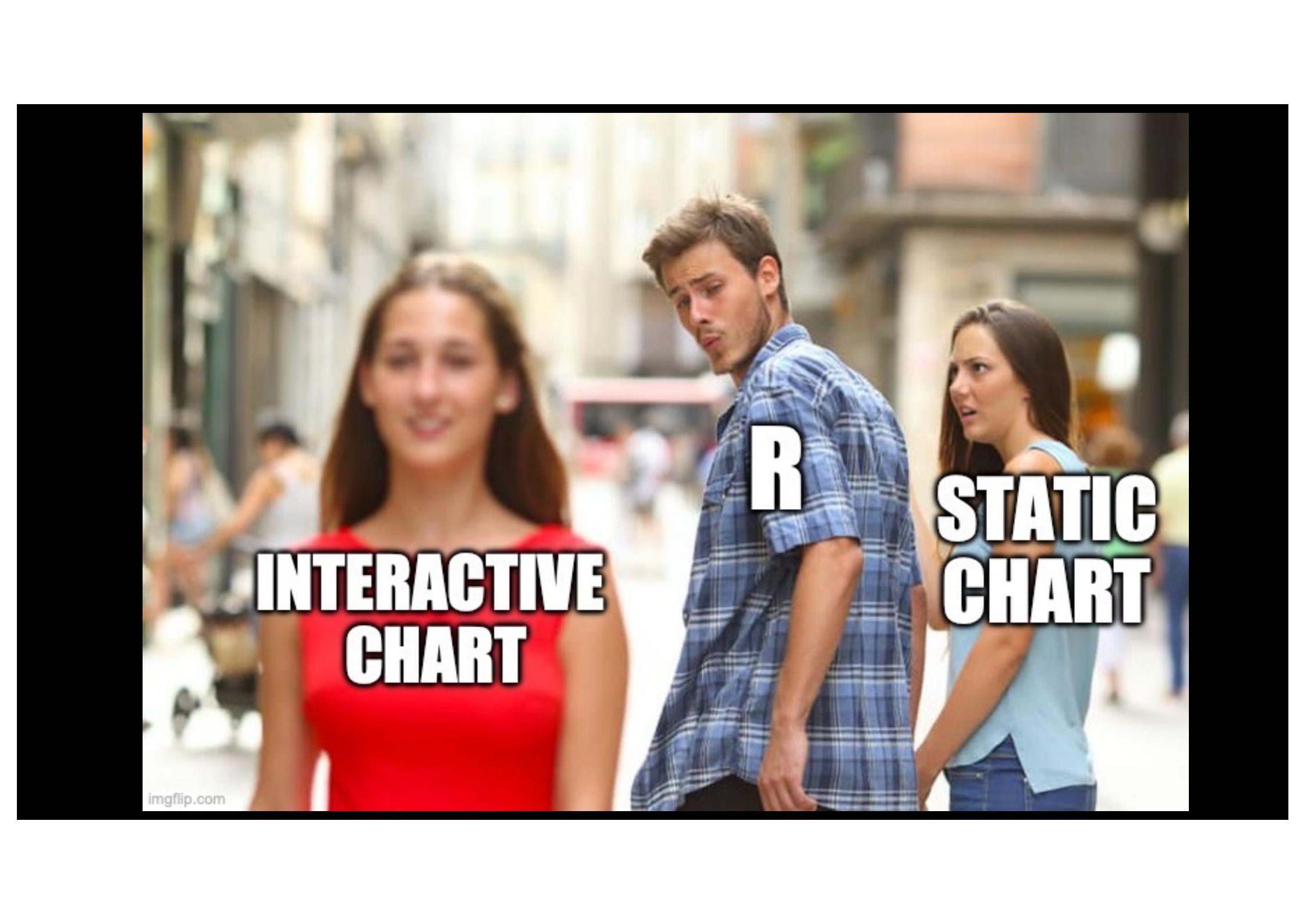
Exploratory



Explanatory

Dataviz-inspiration.com





**INTERACTIVE
CHART**

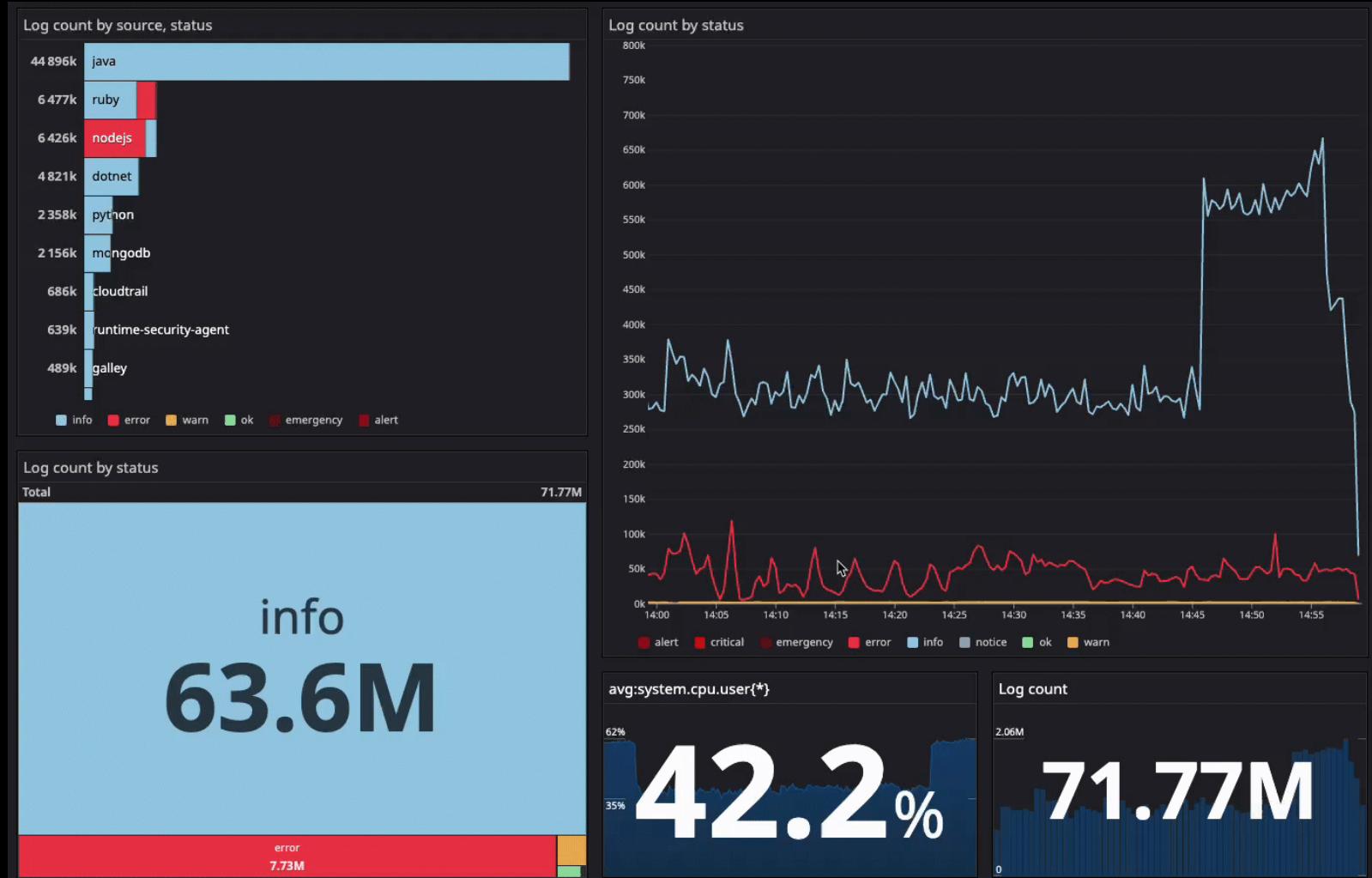
R

**STATIC
CHART**

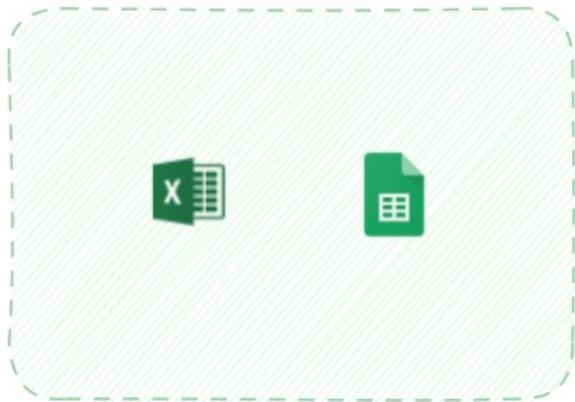


Hover effect
Tooltip
Zoom
Button
Scroll
Click

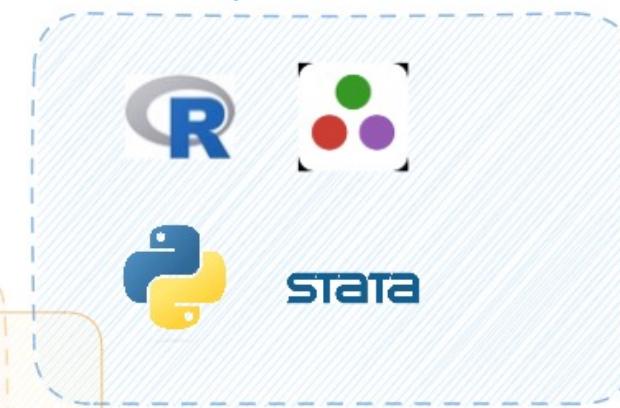
Dynamic data



Spreadsheet



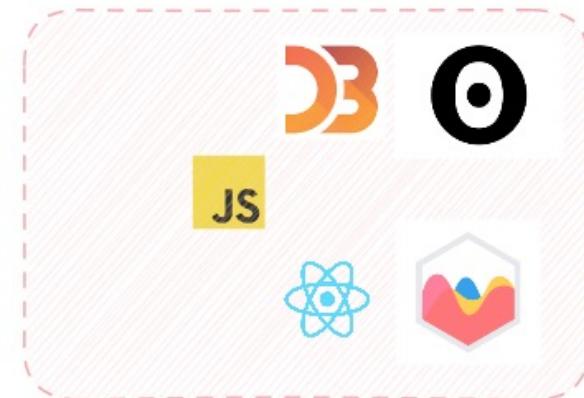
Data Analytics



No Code tools

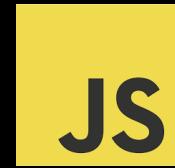


The web





+



browser javascript



```
<!DOCTYPE html>  
<html>
```

```
<body>  
  <div id="rectangle"></div>  
</body>
```

```
</html>
```

HTML

A very simple website made in HTML





```
<!DOCTYPE html>
<html>

<head>
  <style>
    body {
      background-color: black;
    }
    #rectangle {
      width: 100px;
      height: 50px;
      margin-top: 40px;
      background-color: blue;
      position: absolute;
      left: 0;
      top: 0;
    }
  </style>
</head>

<body>
  <div id="rectangle"></div>
</body>
```

CSS

HTML

Styled with CSS





```
<!DOCTYPE html>
<html>

<head>
  <style>
    body {
      background-color: black;
    }
    #rectangle {
      width: 100px;
      height: 50px;
      margin-top: 40px;
      background-color: blue;
      position: absolute;
      left: 0;
      top: 0;
    }
  </style>
</head>

<body>
  <div id="rectangle"></div>
</body>

<script>
  var rectangle = document.getElementById("rectangle");
  var position = 0;

  function animate() {
    position += 5;
    rectangle.style.left = position + "px";
    requestAnimationFrame(animate);
  }

  animate();
</script>

</html>
```

CSS

HTML

JS

Alive with Javascript

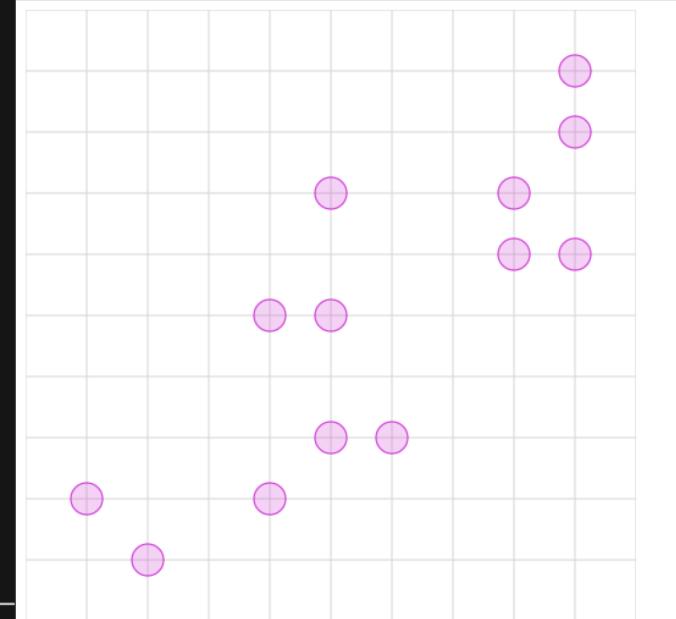


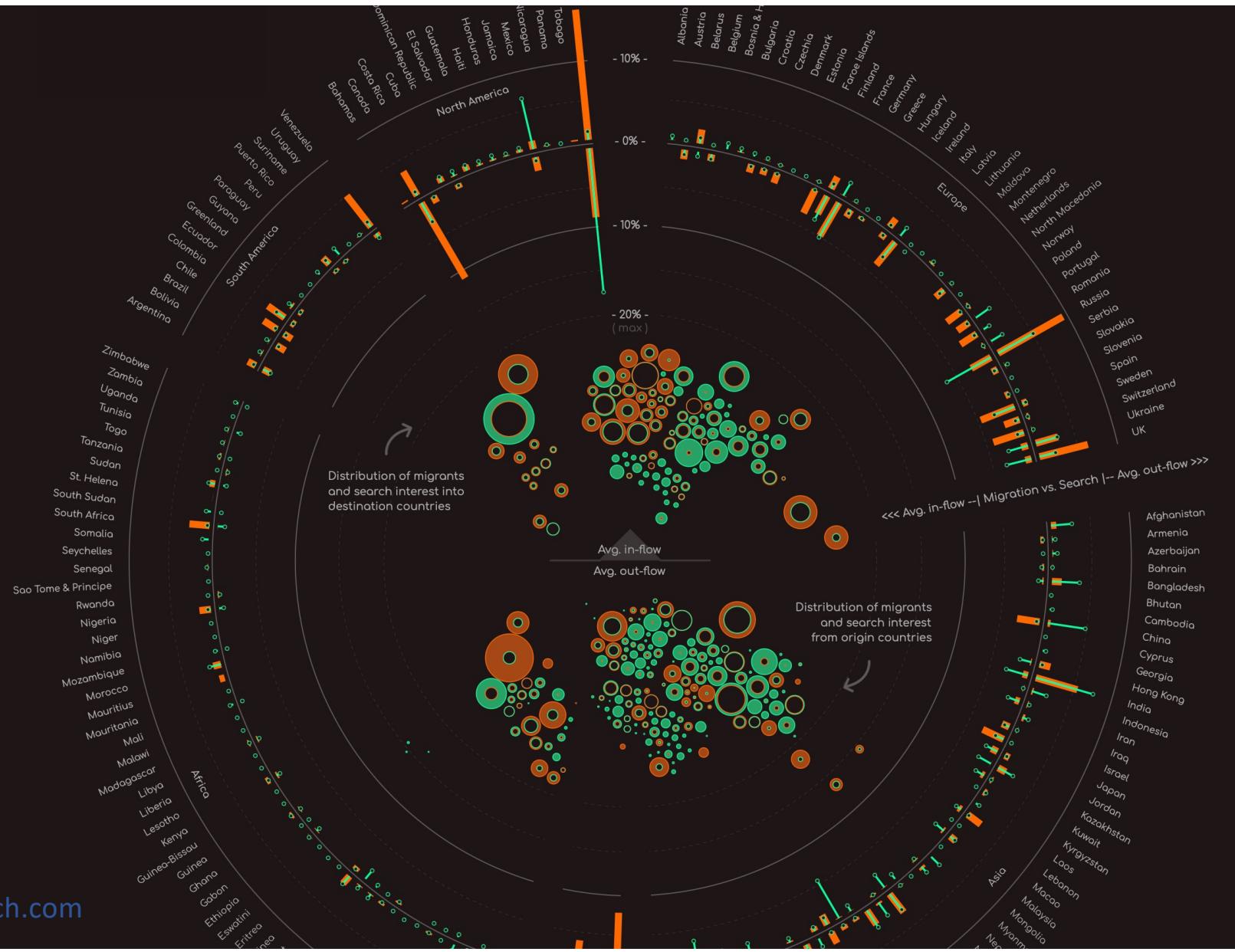
```
Scatterplot.tsx x
```

```
1 import * as d3 from "d3";
2 import { AxisLeft } from "./AxisLeft";
3 import { AxisBottom } from "./AxisBottom";
4
5 export const Scatterplot = ({ data }) => {
6   // Scales
7   const yScale = d3.scaleLinear().domain([0, 10]).range([500, 0]);
8   const xScale = d3.scaleLinear().domain([0, 10]).range([0, 500]);
9
10  // Build the shapes
11  const allShapes = data.map((d, i) => {
12    return (
13      <circle
14        key={i}
15        r={13}
16        cx={xScale(d.y)}
17        cy={yScale(d.x)}
18        opacity={1}
19        stroke="#cb1dd1"
20        fill="#cb1dd1"
21        fillOpacity={0.2}
22        strokeWidth={1}
23      />
24    );
25  });
26
27  return (
28    <div>
29      <svg width={500} height={500}>
30        <AxisLeft yScale={yScale} pixelsPerTick={40} width={500} />
31
32        <g transform={`translate(0, ${500})`}>
33          <AxisBottom xScale={xScale} pixelsPerTick={40} height={500} />
34        </g>
35
36        {allShapes}
37      </svg>
38    </div>
39  );
40}
```

```
Browser Tests Terminal
```

```
< > C https://ry9jzl.csb.app/
```



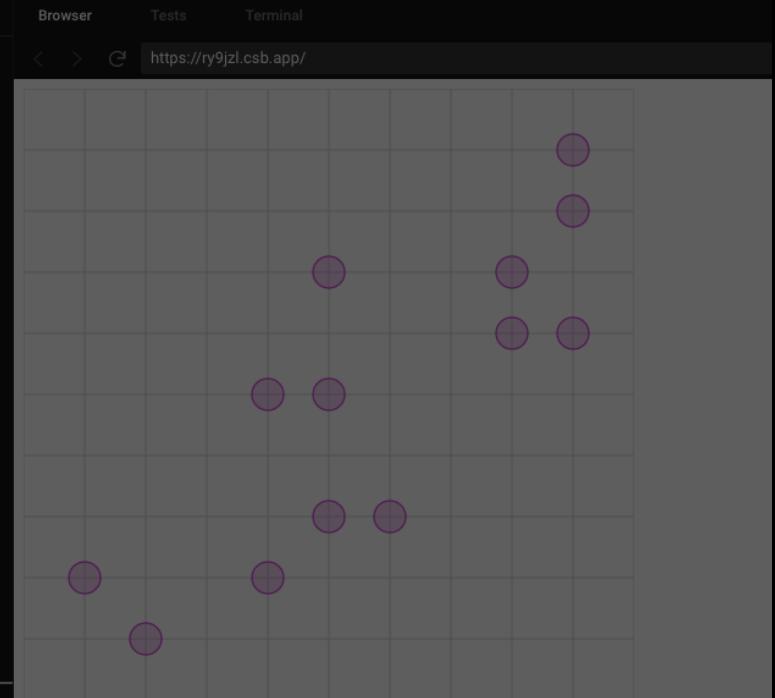


```
Scatterplot.tsx x
1 import * as d3 from "d3";
2 import { AxisLeft } from "./AxisLeft";
3 import { AxisBottom } from "./AxisBottom";
4
5 export const Scatterplot = ({ data }) => {
6   // Scales
7   const yScale = d3.scaleLinear().domain([0, 10]).range([500, 0]);
8   const xScale = d3.scaleLinear().domain([0, 10]).range([0, 500]);
9
10  // Build the shapes
11  const allShapes = data.map((d, i) => {
12    return (
13      <circle
14        key={i}
15        r={13}
16        cx={xScale(d.x)}
17        cy={yScale(d.y)}
18        opacity={1}
19        stroke="#cb1dd1"
20        fill="#cb1dd1"
21        fillOpacity={0.2}
22        strokeDash={[4, 4]}
23      );
24    });
25  }
26
27  return (
28    <div>
29      <svg width={500} height={500}>
30        <AxisLeft yScale={yScale} pixelsPerTick={40} width={500} />
31
32        <g transform={`translate(0, ${500})`}>
33          <AxisBottom xScale={xScale} pixelsPerTick={40} height={500} />
34        </g>
35
36        {allShapes}
37      </svg>
38    </div>
39  );
40};
```

Flexibility:



Complexity:



```
Scatterplot.tsx x
1 import * as d3 from "d3";
2 import { AxisLeft } from "./AxisLeft";
3 import { AxisBottom } from "./AxisBottom";
4
5 export const Scatterplot = ({ data }) => {
6   // Scales
7   const yScale = d3.scaleLinear().domain([0, 10]).range([500, 0]);
8   const xScale = d3.scaleLinear().domain([0, 10]).range([0, 500]);
9
10  // Build the shapes
11  const allShapes = data.map((d, i) => {
12    return (
13      <circle
14        key={i}
15        r={13}
16        cx={xScale(d.x)}
17        cy={yScale(d.y)}
18        opacity={1}
19        stroke="#cb1dd1"
20        fill="#cb1dd1"
21        fillOpacity={0.2}
22        strokeDash={[4, 4]}
23      );
24    });
25
26    return (
27      <div>
28        <svg width={500} height={500}>
29          <AxisLeft yScale={yScale} pixelsPerTick={40} width={500} />
30
31          <g transform={`translate(0, ${500})`}>
32            <AxisBottom xScale={xScale} pixelsPerTick={40} height={500} />
33          </g>
34
35          {allShapes}
36        </svg>
37      </div>
38    );
39  );
40};
```

Flexibility:



Complexity:



Browser Tests Terminal

< > C https://ry9jzl.csb.app/



Abstraction 1 : D3.js

Abstraction 2 : Other Chart libraries

1 D3.js

The JavaScript library
for bespoke data
visualization

Create custom dynamic visualizations
with unparalleled flexibility

Get started What is D3? Examples



D3 by Observable

Search 36 K

Home Examples Community Plot ...

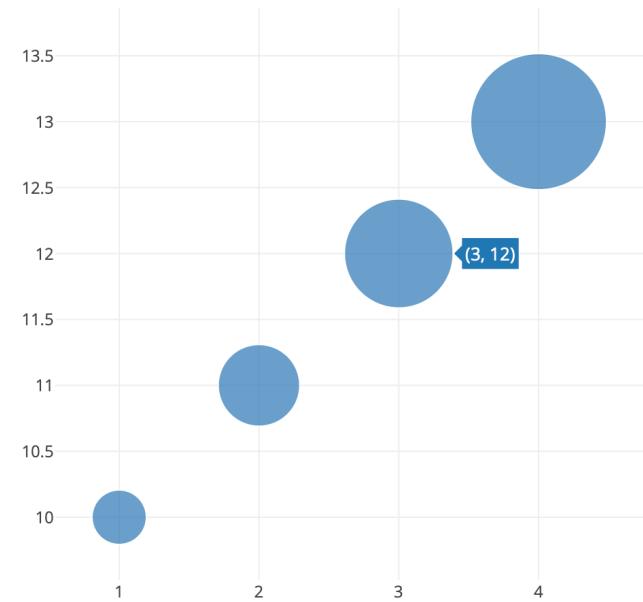


2

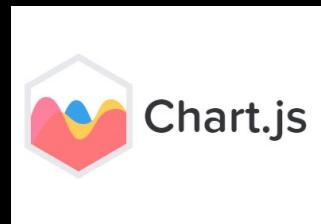
Other chart libraries



```
var trace1 = {  
    x: [1, 2, 3, 4],  
    y: [10, 11, 12, 13],  
    mode: 'markers',  
    marker: {  
        size: [40, 60, 80, 100]  
    }  
};  
  
var data = [trace1];  
  
var layout = {  
    title: 'Marker Size',  
    showlegend: false,  
    height: 600,  
    width: 600  
};  
  
Plotly.newPlot('myDiv', data, layout);
```



Vega-Lite

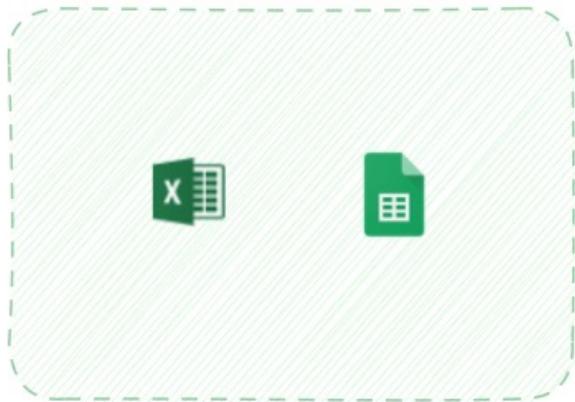


three.js

dygraphs

And many many many
many more !

Spreadsheet



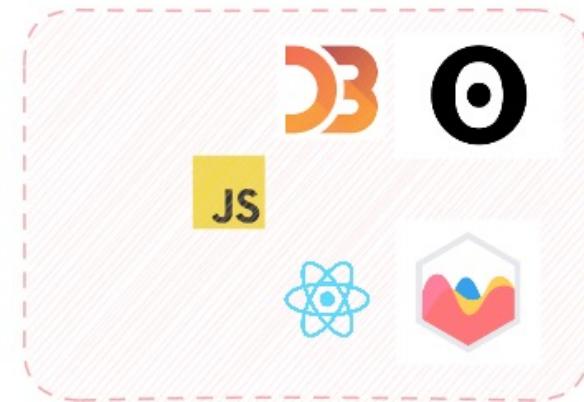
Data Analytics

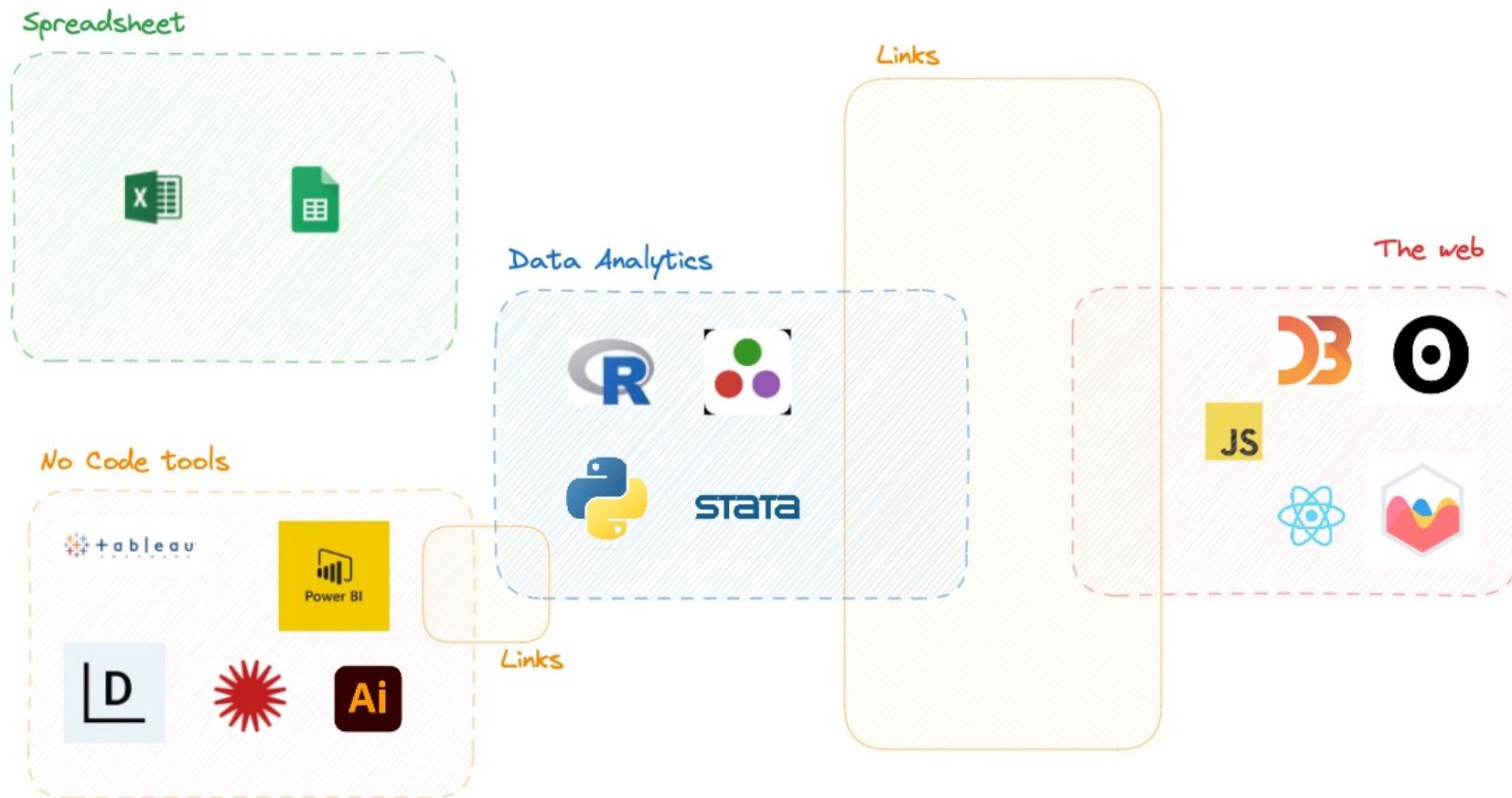


No Code tools



The web





htmlwidgets.org

create R bindings to
JavaScript libraries

htmlwidgets for R

Home

Showcase

Develop ▾

Flexdashboard

Crosstalk

Gallery

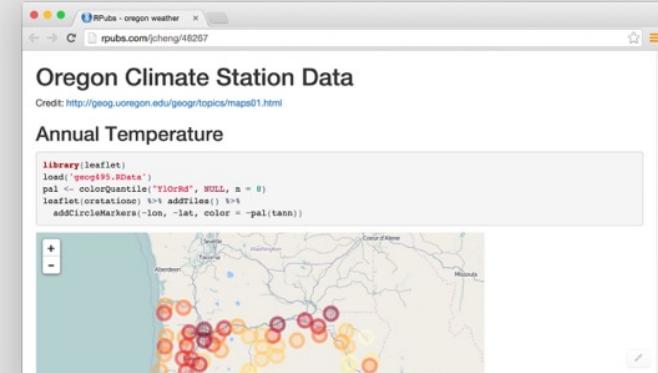
GitHub

Bring the best of JavaScript data visualization to R

Use JavaScript visualization libraries at the R console, just like plots

Embed widgets in **R Markdown** documents and Shiny web applications

Develop new widgets using a framework that seamlessly bridges R and JavaScript

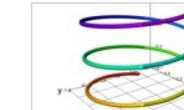
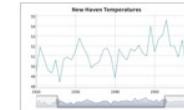


At the R console

In R Markdown docs

In Shiny apps

Widgets in action



Just a line or two of R code can be used to create interactive visualizations. See the featured widgets in the [showcase](#) and browse over 50 available widgets in the [gallery](#).

[See the showcase »](#)

Interactive dashboards

You can easily compose multiple htmlwidgets into a dashboard using the [flexdashboard](#) package, which features flexible and easy to specify row and column-based layouts for multiple widgets.

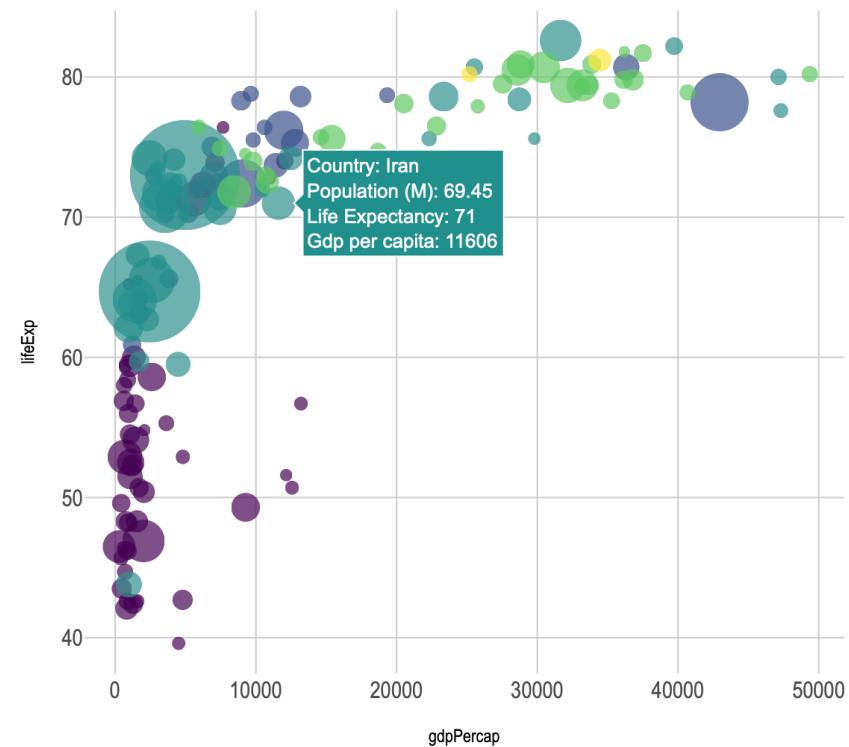
[Create a dashboard »](#)



The R package !

```
# Classic ggplot
ggplot(
  aes(x=gdpPercap, y=lifeExp, size = pop, color =
    continent, text=text)) +
  geom_point(alpha=0.7) +
  scale_size(range = c(1.4, 19), name="Population
    (M)") +
  scale_color_viridis(discrete=TRUE, guide=FALSE) +
  theme_ipsum() +
  theme(legend.position="none")

# turn ggplot interactive with plotly
ggplotly(p, tooltip="text")
```



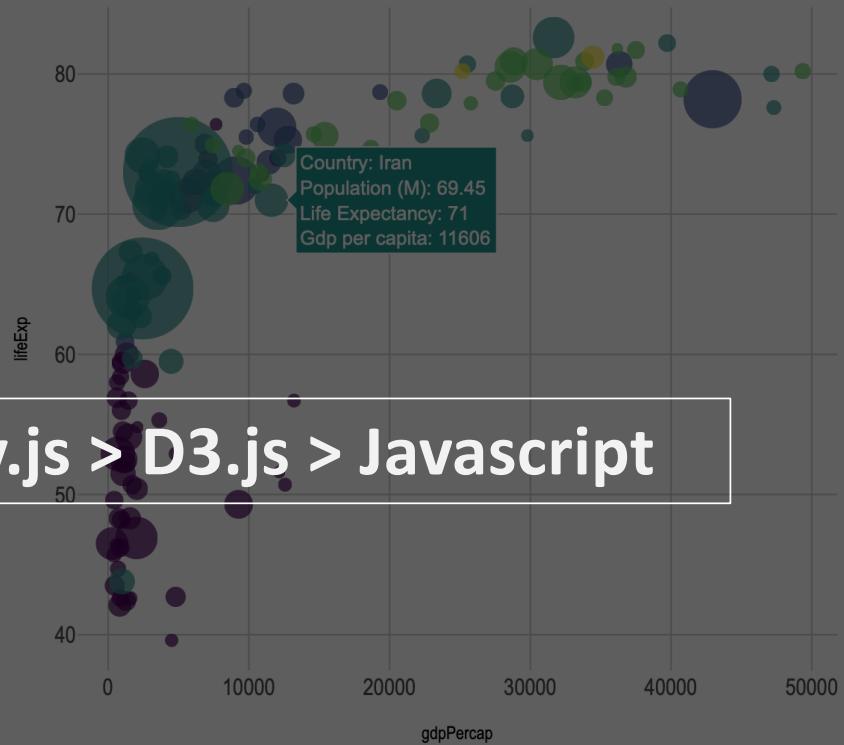


```
# Classic ggplot
ggplot(
  aes(x=gdpPercap, y=lifeExp, size = pop, color =
    continent, text=text)) +
  geom_point(alpha=0.7) +
  scale_size(range = c(1.4, 19), name="Population
    (M)") +
  scale_color_viridis(discrete=TRUE, limit=c(5))
  theme_ipsum() +
  theme(legend.position="none")

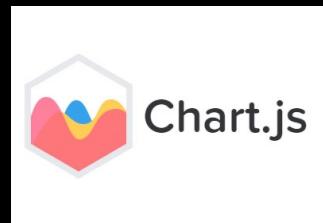
# turn ggplot interactive with plotly
ggplotly(p, tooltip="text")
```



R Plotly > ggplot2 > Plotly.js > D3.js > Javascript



Vega-Lite



Leaflet

Vega-Altair

Tutuchan / chartjs

leaflet

three.js

dygraphs



three.js

dygraphs

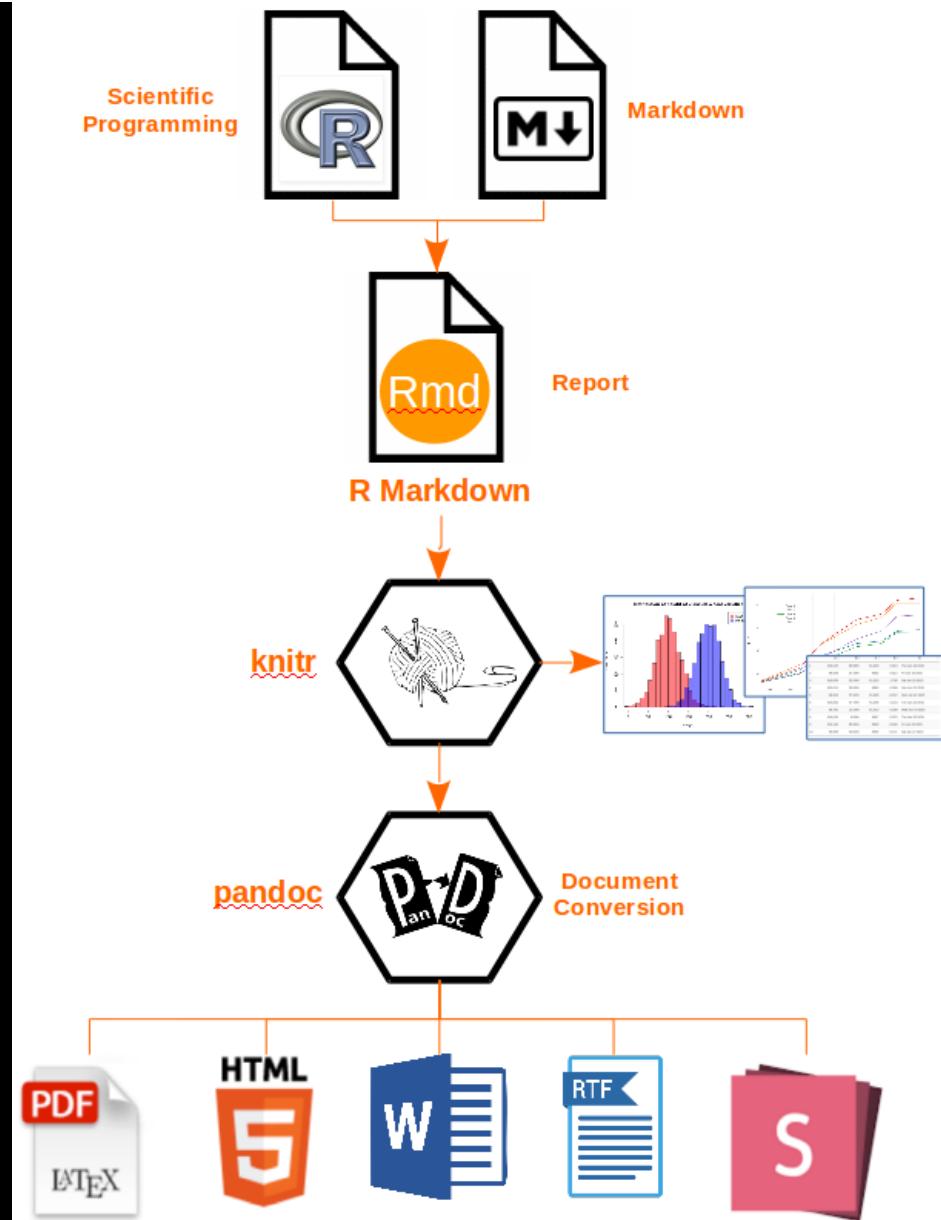
networkD3:
D3 JavaScript Network Graphs from R



- Reporting
- Reproducibility

- RENDER HTML !

- Export a static website



Dream Pipeline 😊

data



.rmd

Hello

2023-06-22

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

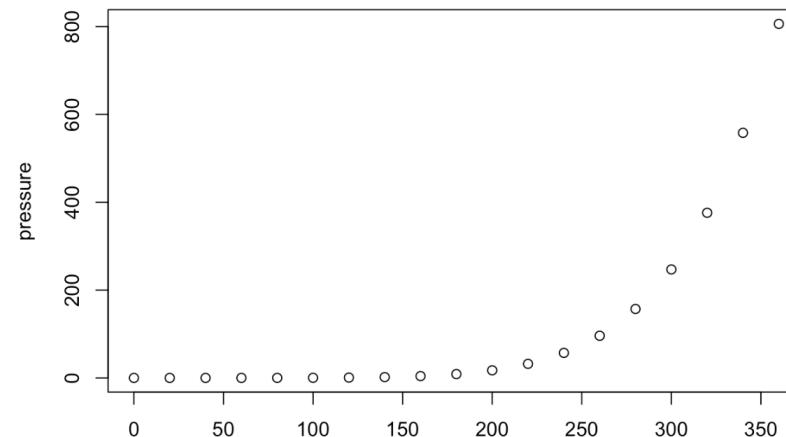
```
summary(cars)
```



```
##      speed         dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean    :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.    :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Dream Pipeline 😊

data



.rmd

pretty template



interactive charts



Libraries

Data wrangling

Let's build a chart

Let's keep data for 2007 only

HIDE

```
data <- gapminder %>% filter(year=="2007") %>% select(-year)
```

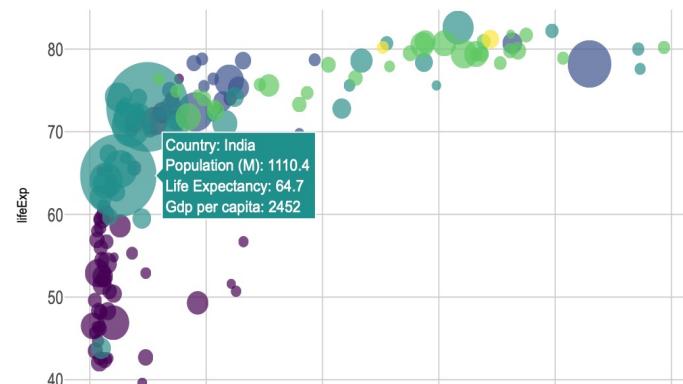
Let's build a chart

Build the chart with ggplot2, make it interactive with plotly.

HIDE

```
# Interactive version
p <- data %>%
  mutate(gdpPercap=round(gdpPercap,0)) %>%
  mutate(pop=round(pop/1000000,2)) %>%
  mutate(lifeExp=round(lifeExp,1)) %>%
  arrange(desc(pop)) %>%
  mutate(country = factor(country, country)) %>%
  mutate(text = paste("Country: ", country, "\nPopulation (M): ", pop, "\nLife Expectancy: ", lifeExp,
                     ggpplot( aes(x=gdpPercap, y=lifeExp, size = pop, color = continent, text=text)) +
  geom_point(alpha=0.7) +
  scale_size(range = c(1.4, 19), name="Population (M)") +
  scale_color_viridis(discrete=TRUE, guide=FALSE) +
  theme_ipsum() +
  theme(legend.position="none")

ggplotly(p, tooltip="text")
```



Dream Pipeline

data



.rmd

pretty template



interactive charts



Send to Github

Projects Packages Stars 72

Type to search

Pinned

Customize your pins

data_to_viz Public
Leading to the dataviz you need
HTML ⚡ 799 ⚡ 263

D3-graph-gallery Public
A collection of simple graphics made with D3.js
HTML ⚡ 698 ⚡ 227

The-Python-Graph-Gallery Public
A website displaying hundreds of charts made with Python
Jupyter Notebook ⚡ 1.5k ⚡ 330

R-graph-gallery Public
A website that displays hundreds of R charts with their code
HTML ⚡ 560 ⚡ 208

dataviz-inspiration Public
A collection of viz I love
TypeScript ⚡ 62 ⚡ 7

react-graph-gallery Public
A set of graph examples showing how to make react and d3.js work together
TypeScript ⚡ 49 ⚡ 8

Single sign-on to see contributions for organizations within the Datadog, Inc. enterprise.

682 contributions in the last year

Contribution settings ▾

Less More

Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun

Mon Wed Fri

@d3

Activity overview

Contributed to [holtzy/react-graph-gallery](#), [holtzy/dataviz-corner](#), [holtzy/The-Python-Graph-Gall...](#) and 13 other repositories

1% Code review

92% Commits

7% Issues

Pull requests

2023

2022

2021

2020

2019

2018

2017

2016

2015

2014

Dream Pipeline

data
↓
.rmd
pretty template
↓
interactive charts
↓
Send to Github
↓
Serve online
Share URL

The screenshot shows a GitHub repository settings page for the repository `holtzy / data_analysis_website`. The main navigation bar includes links for Projects, Packages, Stars (72), and a search bar. Below the navigation, there are several tabs: Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The Settings tab is highlighted with a red underline and a red vertical bar on the right side of the page.

The left sidebar contains sections for General, Access, Collaborators, Moderation options, Code and automation (Branches, Tags, Rules, Actions, Webhooks, Environments, Codespaces, Pages), and Security. The Pages section is highlighted with a red vertical bar.

The main content area is titled "GitHub Pages". It states: "GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository." A message indicates that the site is live at https://holtzy.github.io/data_analysis_website/, which is also highlighted with a red bar. Buttons for "Visit site" and "..." are present.

The "Build and deployment" section includes a "Source" dropdown set to "Deploy from a branch", a "Branch" dropdown set to "main" (root), and a "Save" button. It also provides instructions to add a Jekyll theme.

The "Custom domain" section lists "holtzy/The-Python-Graph-Gall..." and "and 13 other repositories".

A timeline chart at the bottom shows activity from 2014, with data points for 92% Commits, 7% Issues, and Pull requests.



✓ Export a dynamic website

✓ Quick prototyping of web app

✓ Statistical power of R on the web

Easy web apps for data science
without the compromises

No web development skills required

Get started in R

Get started in Python



Here is a Shiny app

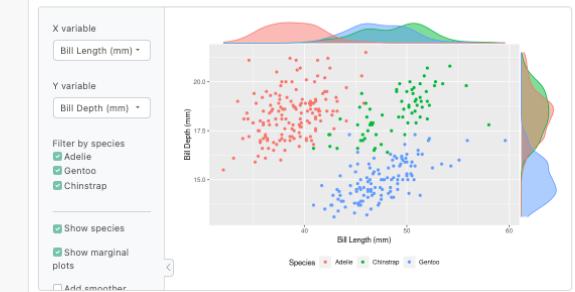
Shiny apps are easy to write. Let users interact with your data and your analysis, all with R or Python:

R Python

```
library(shiny)
library(bslib)
library(dplyr)
library(ggplot2)

# https://raw.githubusercontent.com/jcheng5/simplepenguins.R/main/penguins.csv
df <- readr::read_csv("penguins.csv")
# Find subset of columns that are suitable for scatter plot
df_num <- df %> select(where(is.numeric), -Year)

ui <- page_fillable(theme = bs_theme(bootswatch = "minty"),
  layout_sidebar(fillable = TRUE,
    sidebar(
      varSelectInput("xvar", "X variable", df_num, selected = "Bill Length (mm)"),
      varSelectInput("yvar", "Y variable", df_num, selected = "Bill Depth (mm)"),
      checkboxInput("filter_by_species", "Filter by species",
        c("Adelie", "Gentoo", "Chinstrap")),
      checkboxInput("show_marginal_plots", "Show marginal plots"),
      checkboxInput("add_smoother", "Add smoother")
    ),
    mainPanel(
      ggplot(df, aes(x = get(xvar), y = get(yvar))) +
        geom_point() +
        geom_smooth()
    )
  )
)
```



Get started in R

Get started in Python

I ❤️ Shiny !

LOCATION COMPARISON

You have already identified two locations of interest, but can't decide which one to invest in? Let us help make your final decision. Enter the addresses of two properties below and click the button to compare the locations in terms of gentrification potential.

Click the button to compare two sample locations or enter search addresses of your own!

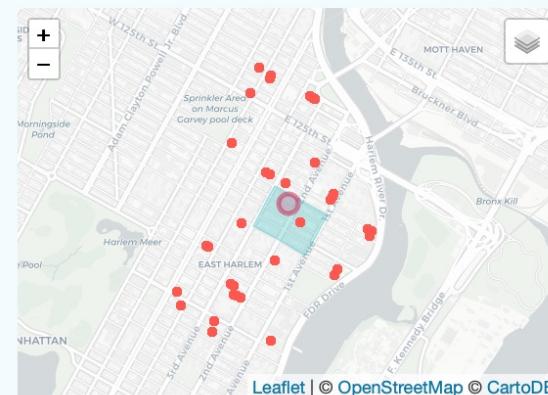
Using the buttons above the map, you can also show the location of nearby schools, subway stations and places on Yelp.

243-245 E 118th St, New York

« Compare locations »

20 Gerry St, New York

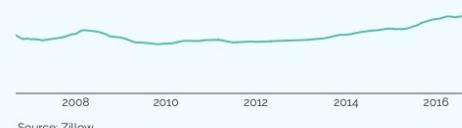
Schools Subway Yelp



EAST HARLEM NORTH

Intelligentsia predicts the census tract of this location to gentrify with a probability of 48.0%.

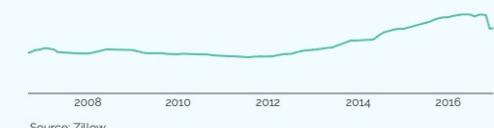
Monthly development of real estate prices



BEDFORD

Intelligentsia predicts the census tract of this location to gentrify with a probability of 99.1%.

Monthly development of real estate prices



**Limits ?
Complexity**

DATADOG

Views Network Performance + Save Learn More 1h Past 1 Hour

Overview Analytics Map DNS Dashboards

Search for Q client and server tags (e.g. "client_service:foo") </> Filter Traffic

View clients as Auto-grouped traffic View servers as Auto-grouped traffic

Client Server Hide Controls Top 100 results found Limit 100 Last updated: Wed, Jun 21, 5:00:30 pm Customize

Summary Graphs

Volume Sent TCP Retransmits RTT

Showing 94 of 94

NETWORK

tcp udp

IP Type

private other loopback link_local

Domain

auth-dotnet.default.svc.c... fraud-prevention-api.def... ad-server.default.svc.clus... email-api-py.default.svc.c... user-db.default.svc.clust... mongo.default.svc.cluste...

CLIENT SERVER ↓ VOLUME

CLIENT	SERVER	VOLUME
fluent-bit	fluent-bit-gke-exporter	35.1 GB / 9.11 MB/s
fluent-bit-gke-exporter	gcp.logging	27.5 GB / 7.29 MB/s
fluent-bit-gke-exporter	logging.googleapis.com	9.80 GB / 2.14 MB/s
fluent-bit-gke-exporter	cloudtrace.googleapis.com	9.73 GB / 2.14 MB/s
gke-demo-dpn-us-west-default-pool-cb40...	intake.profile.datadoghq.com	9.11 GB / 2.53 MB/s
gke-demo-dpn-us-west-default-pool-1bd6...	intake.profile.datadoghq.com	8.79 GB / 2.44 MB/s

FORMULA sum:network.rtt[*] / sum:network.rtt_count[*] / 1000000

Isolate this series Copy tags to clipboard View related hosts View related containers View related processes No related logs View related traces View related profiles

Privacy | Cache | Units | Access right | Scalability | Performance | Loading state | Error notifs

Design System with reusable components

Query parser

State management

Synced graphs

Progressive loading

Context Menu

Table with windowing

The screenshot shows the Datadog Network Performance Analytics interface. At the top, there are tabs for Views, Network Performance (selected), and Save. Below that is a search bar and a dashboard navigation section. The main area features three stacked charts: 'Volume Sent' (line graph), 'TCP Retransmits' (bar chart), and 'RTT' (line graph). A sidebar on the left contains a navigation menu with links like Service Migration, Dashboards, Infrastructure, Monitors, Metrics, Integrations, APM, CI, Notebooks, Logs, Security, UX Monitoring, Live Chat, Help, and Auth. The 'Infrastructure' section is expanded, showing 'NETWORK' and 'Network Transport' facets, with 'tcp' and 'udp' selected. The 'Logs' section is also expanded, showing 'Domain' and a list of log sources. A 'Search facets' input field is present. On the right, a context menu is open over a table row, listing options such as 'View related hosts', 'View related containers', 'View related processes', 'No related logs', 'View related traces', 'View related profiles', and 'View related services'. The table itself has columns for CLIENT, SERVER, and VOLUME, showing data for various log sources.

CLIENT	SERVER	VOLUME
fluent-bit	fluent-bit-gke-exporter	35.1 GB / 9.1
fluent-bit-gke-exporter	gcp.logging	27.5 GB / 7.1
fluent-bit-gke-exporter	logging.googleapis.com	9.80 GB / 2.1
fluent-bit-gke-exporter	cloudtrace.googleapis.com	9.73 GB / 2.1
gke-demo-dpn-us-west-default-pool-cb40...	intake.profile.datadoghq.com	9.11 GB / 2.53 MB/s 175 MB / 48.6 KB/s 39 3
gke-demo-dpn-us-west-default-pool-1bd6...	intake.profile.datadoghq.com	8.79 GB / 2.44 MB/s 189 MB / 52.4 KB/s 66 3

Limits ?
Do you need a server ?



server

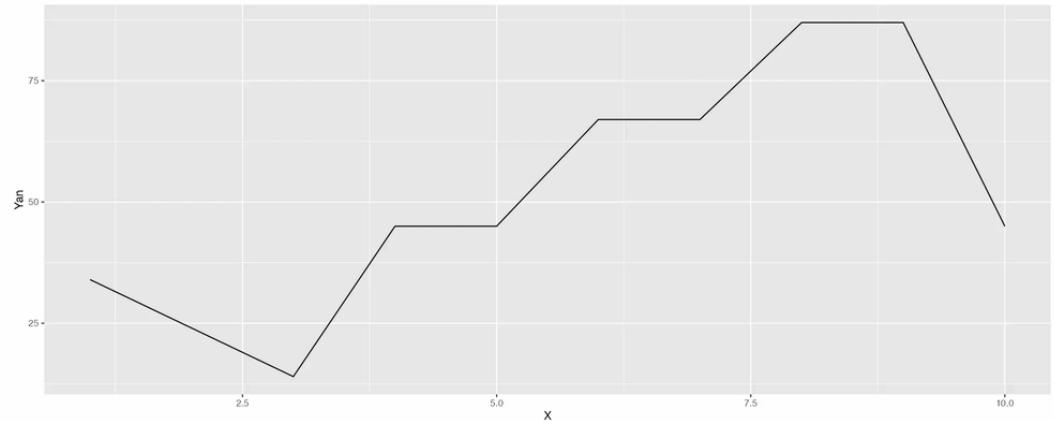


browser

Yan and Melanie Data

Select a variable:

Yan





server

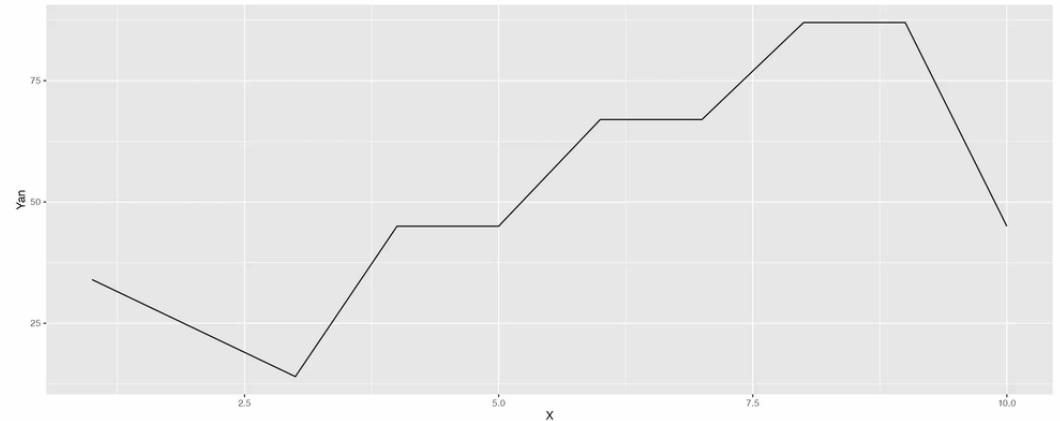


browser

Yan and Melanie Data

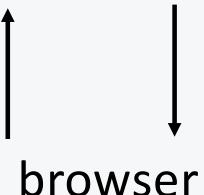
Select a variable:

Yan



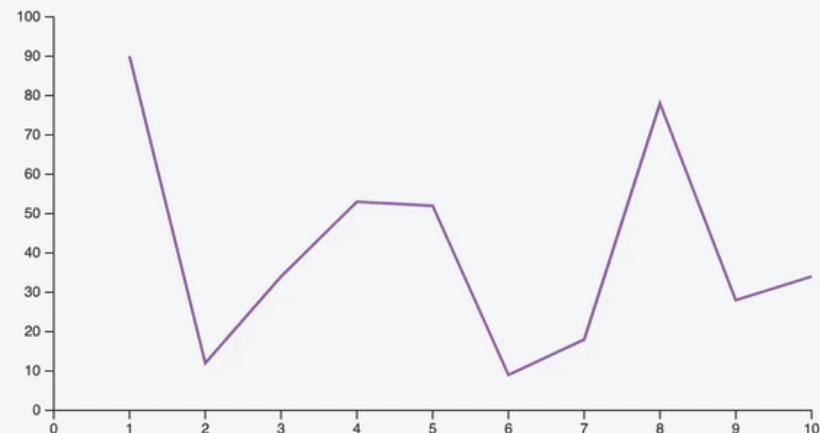
JS

server

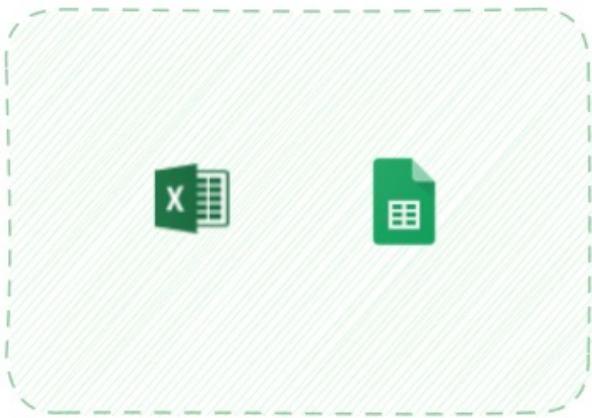


browser

Melanie Yan



Spreadsheet



No Code tools



Data Analytics



Links

HTML Widgets

plotly



The web



JS

Exciting time for dataviz



R is a solid choice
But there are many others!





Exciting time for dataviz

Charts are everywhere



 Exciting time for dataviz
AI is gonna make our life easier

RTutor Home Data Report EDA Ask About

Dataset: mpg [Reset](#)

Example requests:

Boxplot, ggplot2

Use ggplot2 to create a boxplot of highway vs. type. Color by type. Add jitter points.

[Submit](#) [Settings](#) Python

R1: 35 tokens, 2 second(s)
Cumulative API Cost: 0.1¢
Temperature: 0.1

Comments & questions

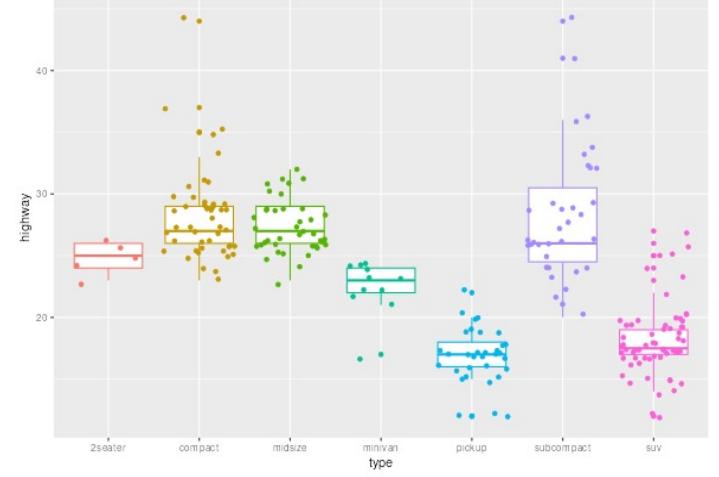
AI generated code:

Chunk #1 Continue from this chunk

```
library(ggplot2)

ggplot(df, aes(x = type, y = highway, color = type)) +
  geom_boxplot() +
  geom_jitter()
```

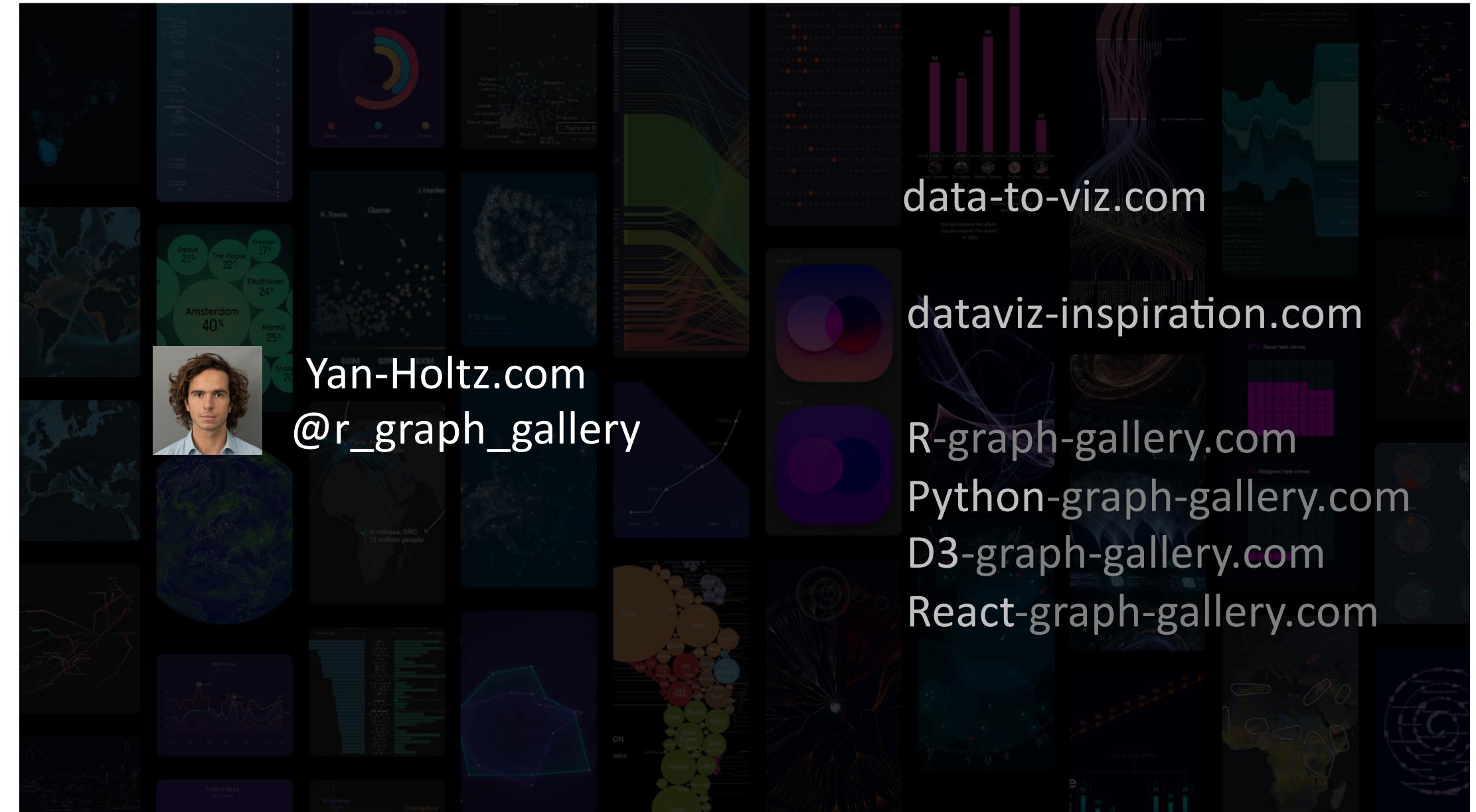
Results:



Interactive via plotly
 Interactive via CanvasXpress

Data Types Description

```
'data.frame': 234 obs. of 11 variables:
 $ maker      : chr "audi" "audi" "audi" "audi" ...
 $ model      : chr "a4" "a4" "a4" "a4" ...
 $ dis         : num  1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
 $ year        : Factor w/ 2 levels "1999","2008": 1 1 2 2 1 1 2 1 2 ...
```



Yan-Holtz.com
@r_graph_gallery

data-to-viz.com

dataviz-inspiration.com

R-graph-gallery.com

Python-graph-gallery.com

D3-graph-gallery.com

React-graph-gallery.com