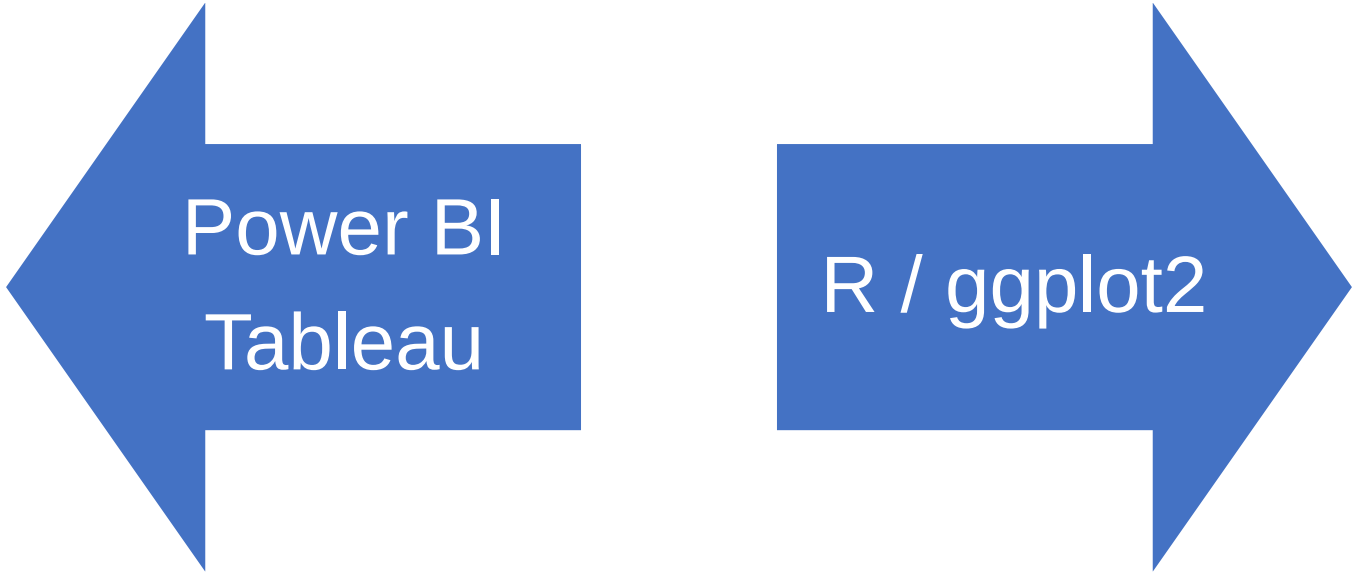
Data Visualisation and Dashboarding

Data visualisation platforms and useful resources

UNIVERSITY OF WESTMINSTER

What are the differences?



Attributes of a data visualisation tool/platform

Ease-of-use

Versatility / Possible charts

Interactivity

Local / server / cloud

Cost

Future-proof

Integration into other platforms

Supported data sources

Collaborative

Decision points

Does the solution have all the features?

Does it support the data source?

Can I afford it?

Analytics and BI platforms

Leaders:

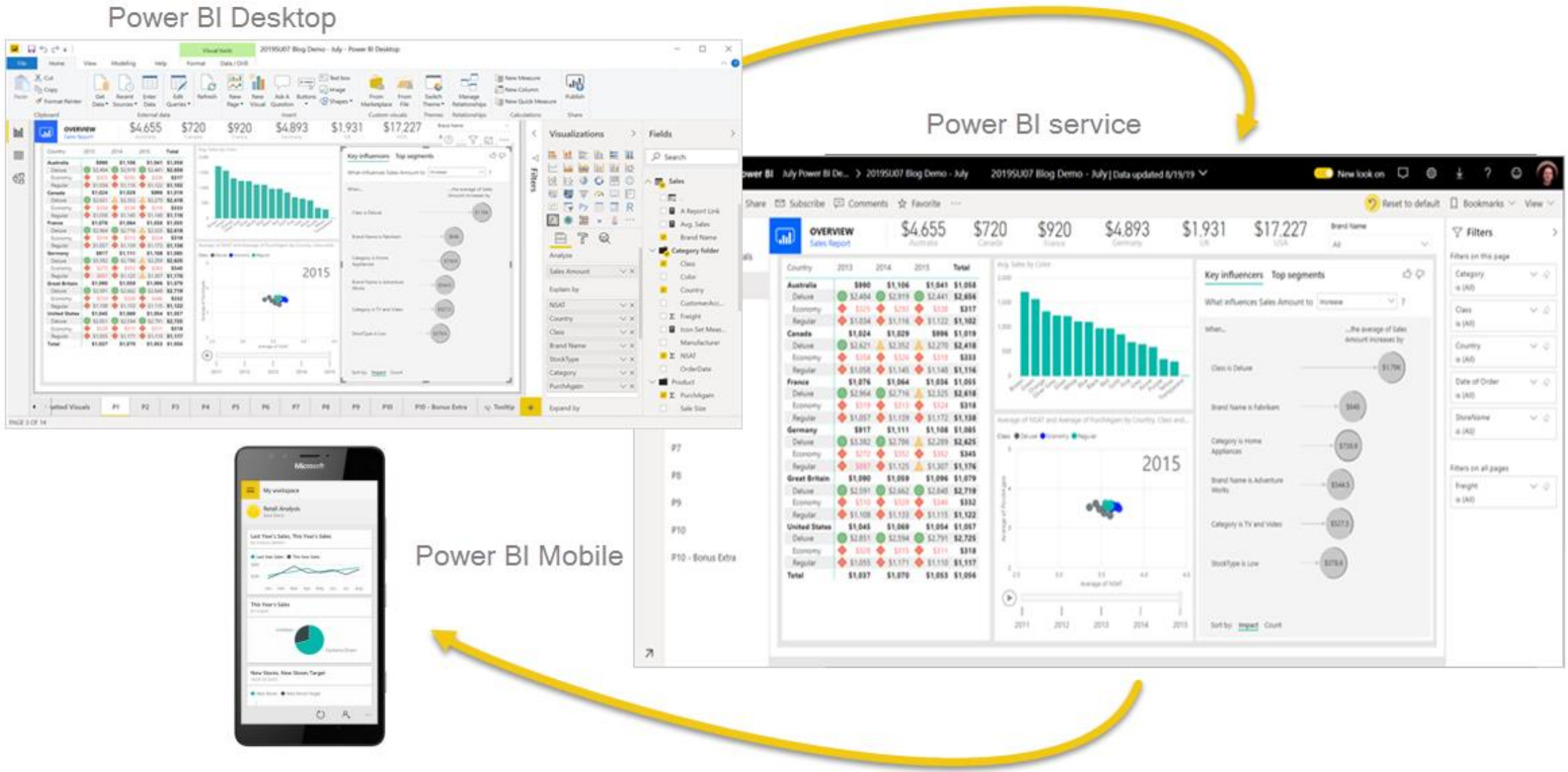
- Microsoft (PowerBI)
- Tableau
- Qlik (QlikView)

Magic Quadrant for Analytics and Business Intelligence Platforms. Gartner 2021.

Gartner: Capability areas

Security	• Capabilities that enable platform security, administering of users, auditing of platform access and authentication.
Manageability	• Capabilities that track usage of the ABI platform and manage how information is shared (and by whom).
Cloud analytics	• The ability to support building, deployment and management of analytics in the cloud, based on data stored both in the cloud and on-premises.
Data source connectivity	• Capabilities that enable users to connect to, query and ingest data, while optimizing for performance.
Data preparation	• Support for drag-and-drop, user-driven combination of data from different sources, and the creation of analytic models (such as user-defined measures, sets, groups and hierarchies).
Catalog	• The ability to automatically generate and curate a searchable catalog of analytic content, thus making it easier for analytic consumers to know what content is available.
Automated insights	• A core attribute of augmented analytics, this is the application of ML techniques to automatically generate findings for end users (for example, by identifying the most important attributes in a dataset).
Data visualization	• Support for highly interactive dashboards and exploration of data through manipulation of chart images.
Data storytelling	• The ability to combine interactive data visualization with narrative techniques in order to package and deliver analytic content in a compelling, easily understood form for presentation to decision makers.
Natural language query (NLQ)	• This enables users to ask questions and query data and analytic content using terms that are either typed into a search box or spoken.
Natural language generation (NLG)	• The automatic creation of linguistically rich descriptions of answers, data and analytic content. Within the analytics context, as the user interacts with data, the narrative changes dynamically to explain key findings or the meaning of charts or dashboards.
Reporting	• The ability to create and distribute (or “burst”) pixel-perfect, grid-layout, multipage reports to users on a scheduled basis.

Microsoft Power BI



Power BI

Workflow: Build dashboard with PowerBI Desktop and deploy to PowerBI service

Offers ETL capabilities with PowerQuery (M)

Ability to load custom visualisations

Strong R integration for visualisation and data preparation

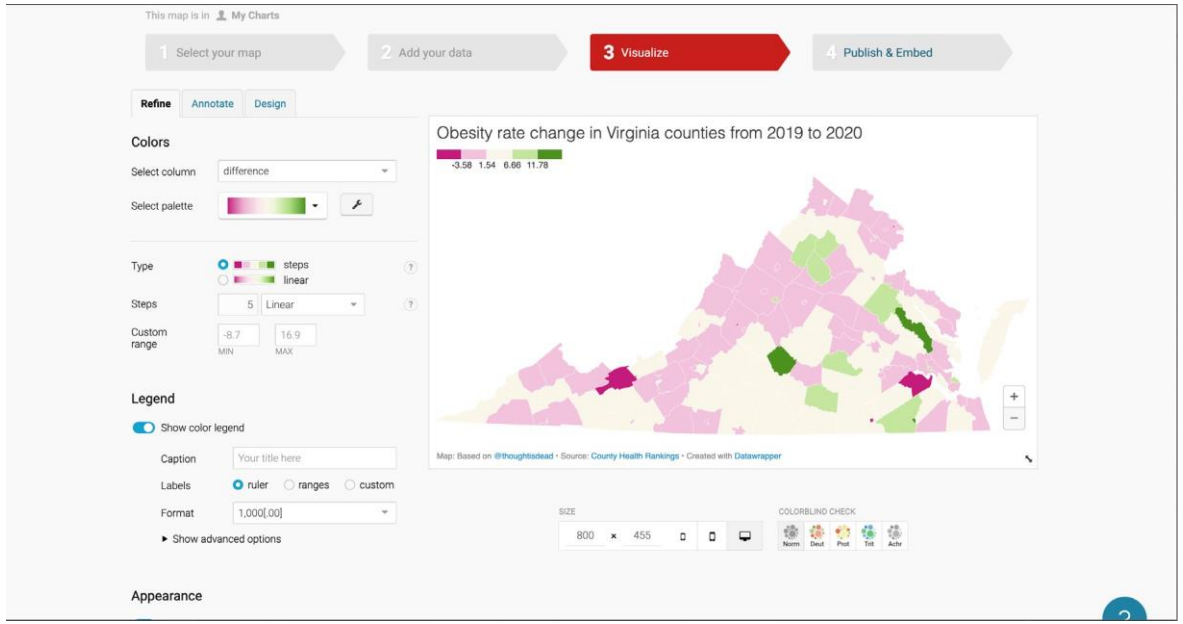
Datawrapper

Browser-based

Responsive designs for maps, charts & tables

Aimed at journalists – used by Thomson Reuters, New York Times, WIRED, Süddeutsche Zeitung, etc.

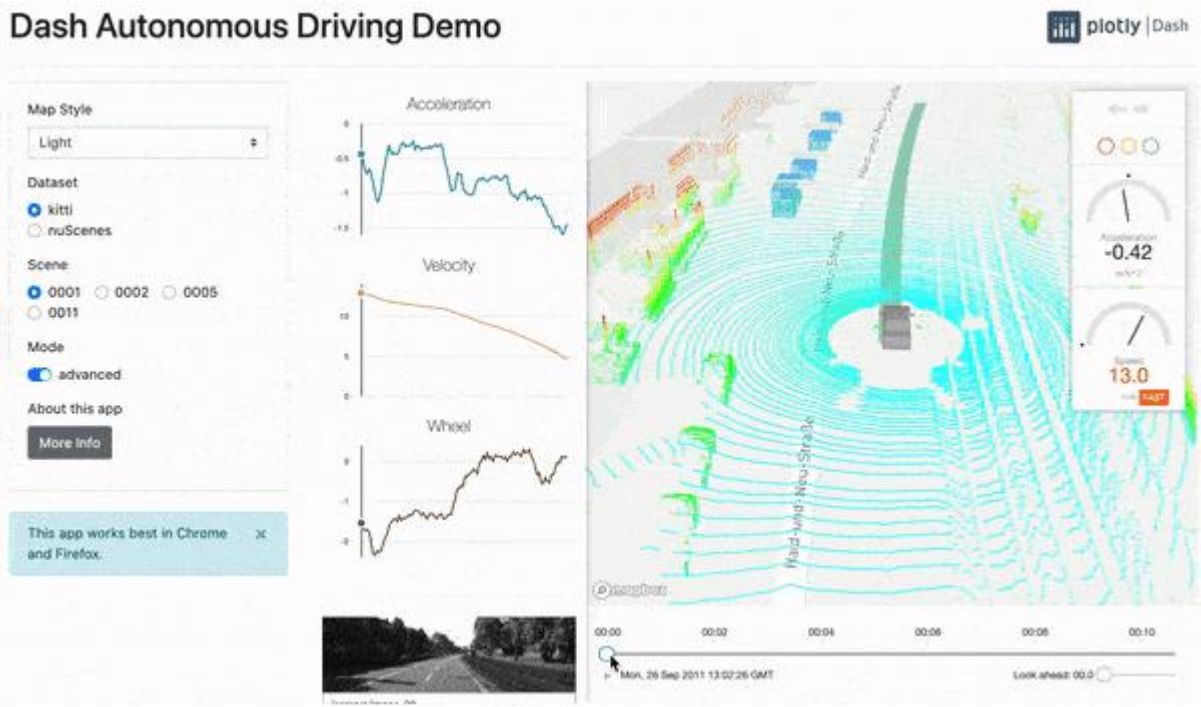
Free and premium plans available



Plotly

Open Source Graphing Libraries for R, Python JavaScript, Matlab, etc.

Dash: Python framework for creating data-centric interactive web apps



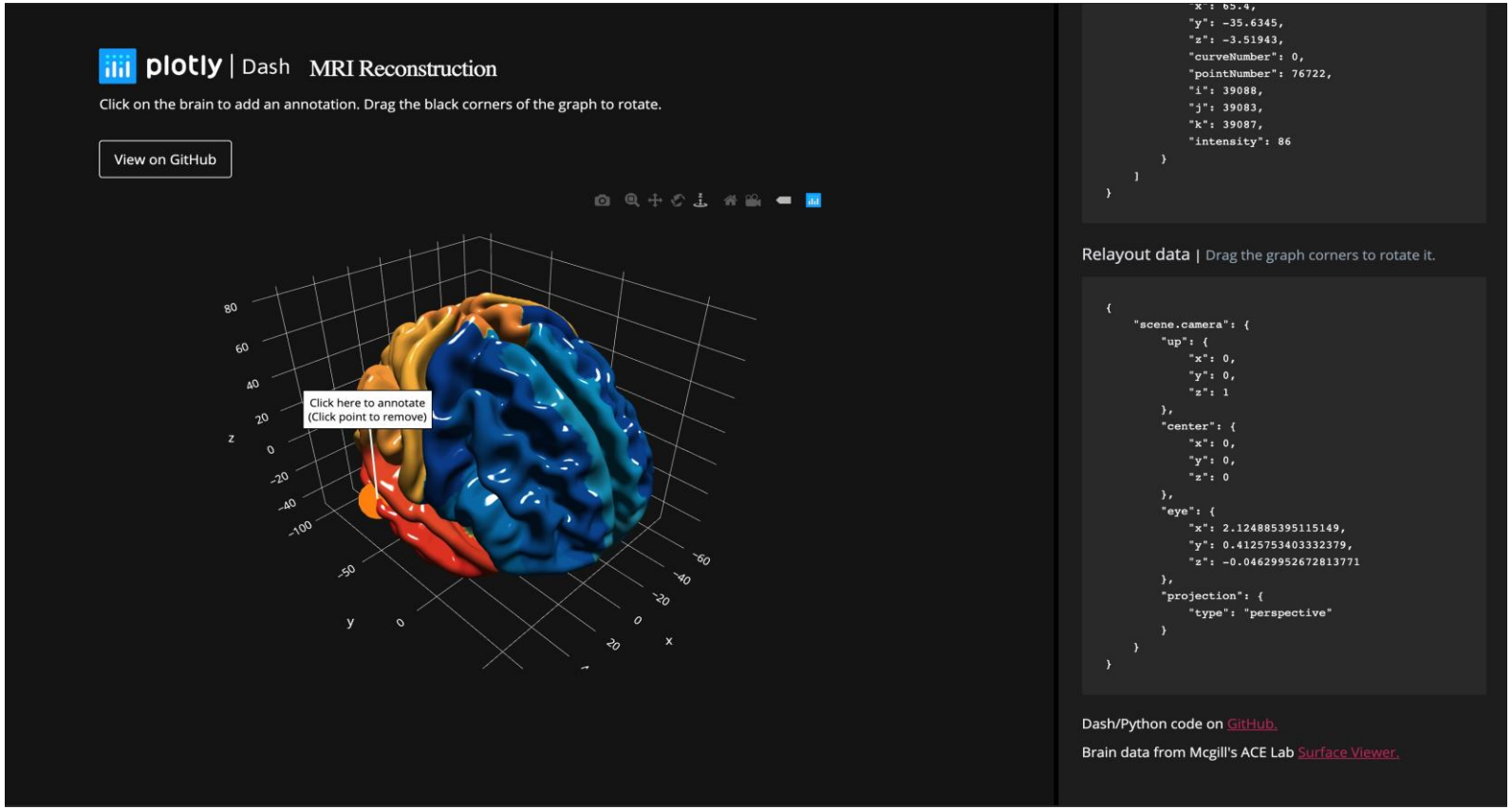
Plotly supports 3D graphics

3D graphics are often bad practice

Exception: Scatter plots and surface charts

Almost always needs interactivity to understand

[Brain surface viewer (dash.gallery)]



UNIVERSITY OF WESTMINSTER

D3.js

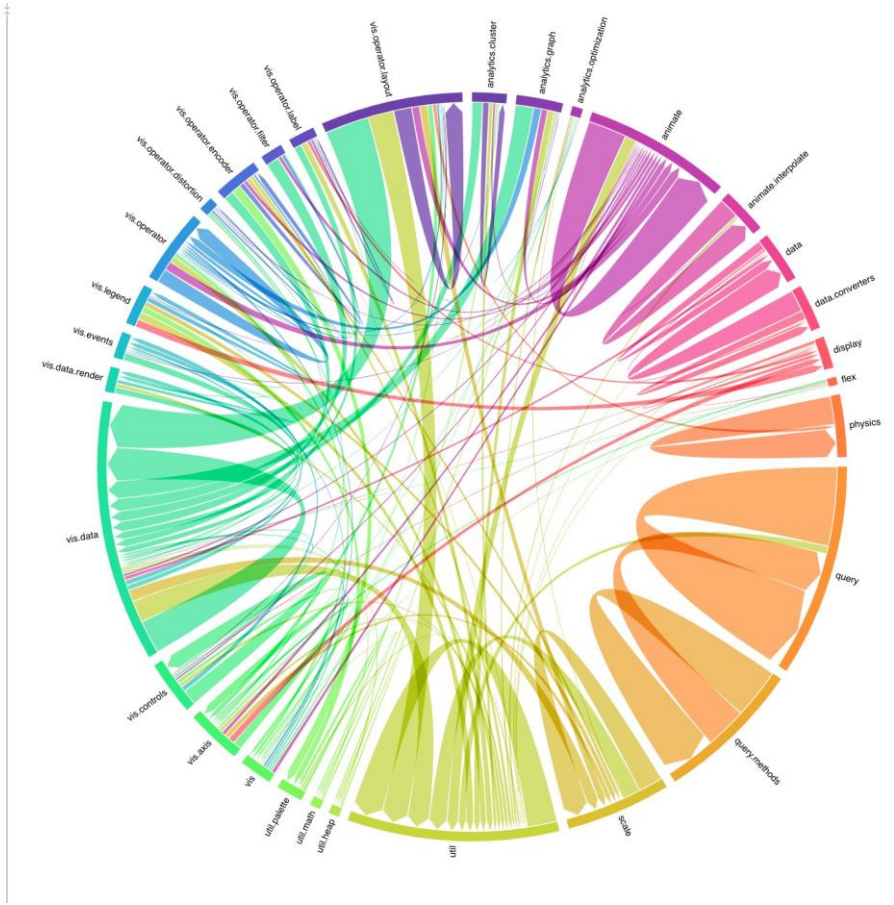
Open Source JavaScript library for data-driven documents

Supports interactivity and animation

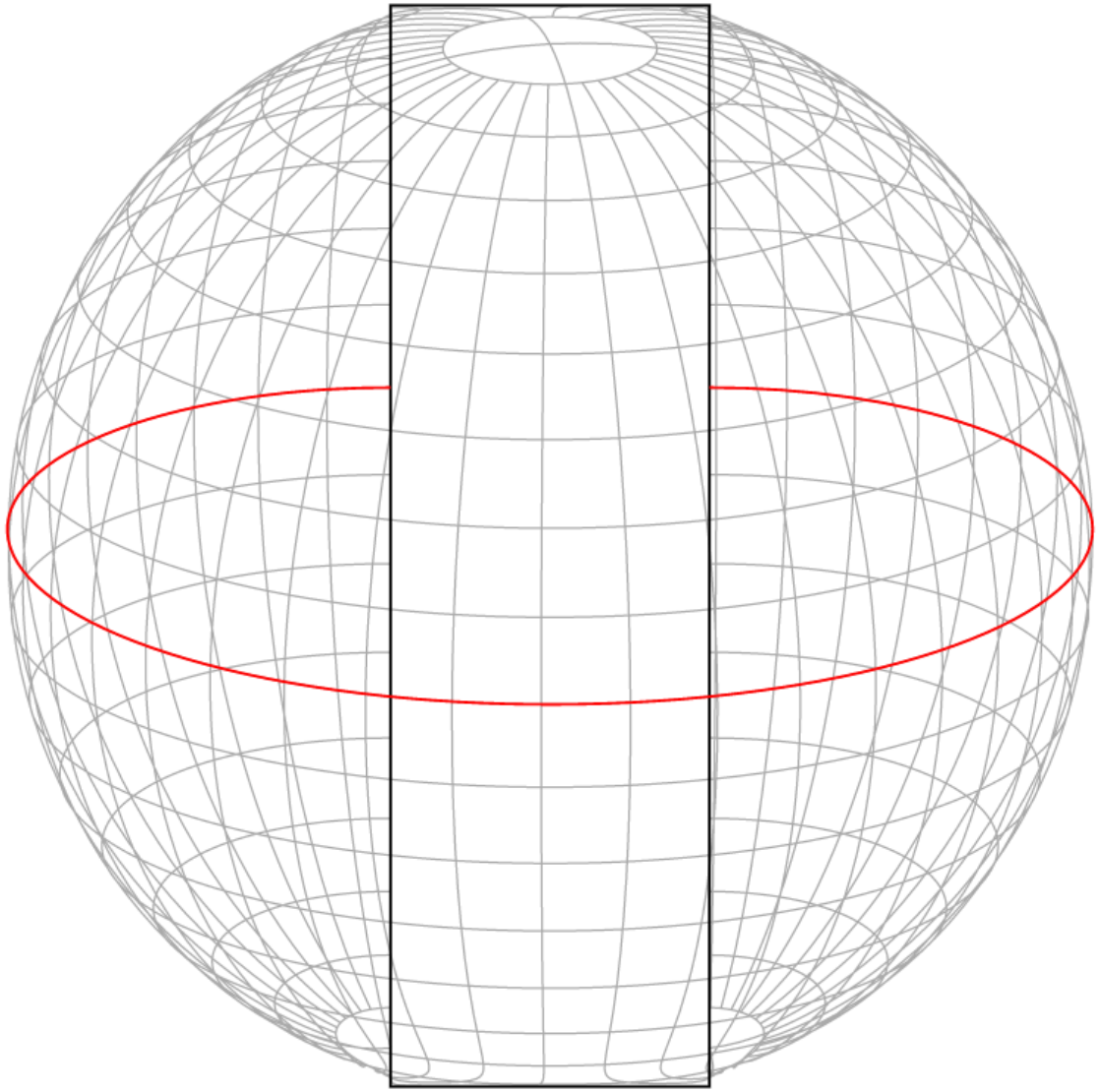
Supports vast number of visualisations

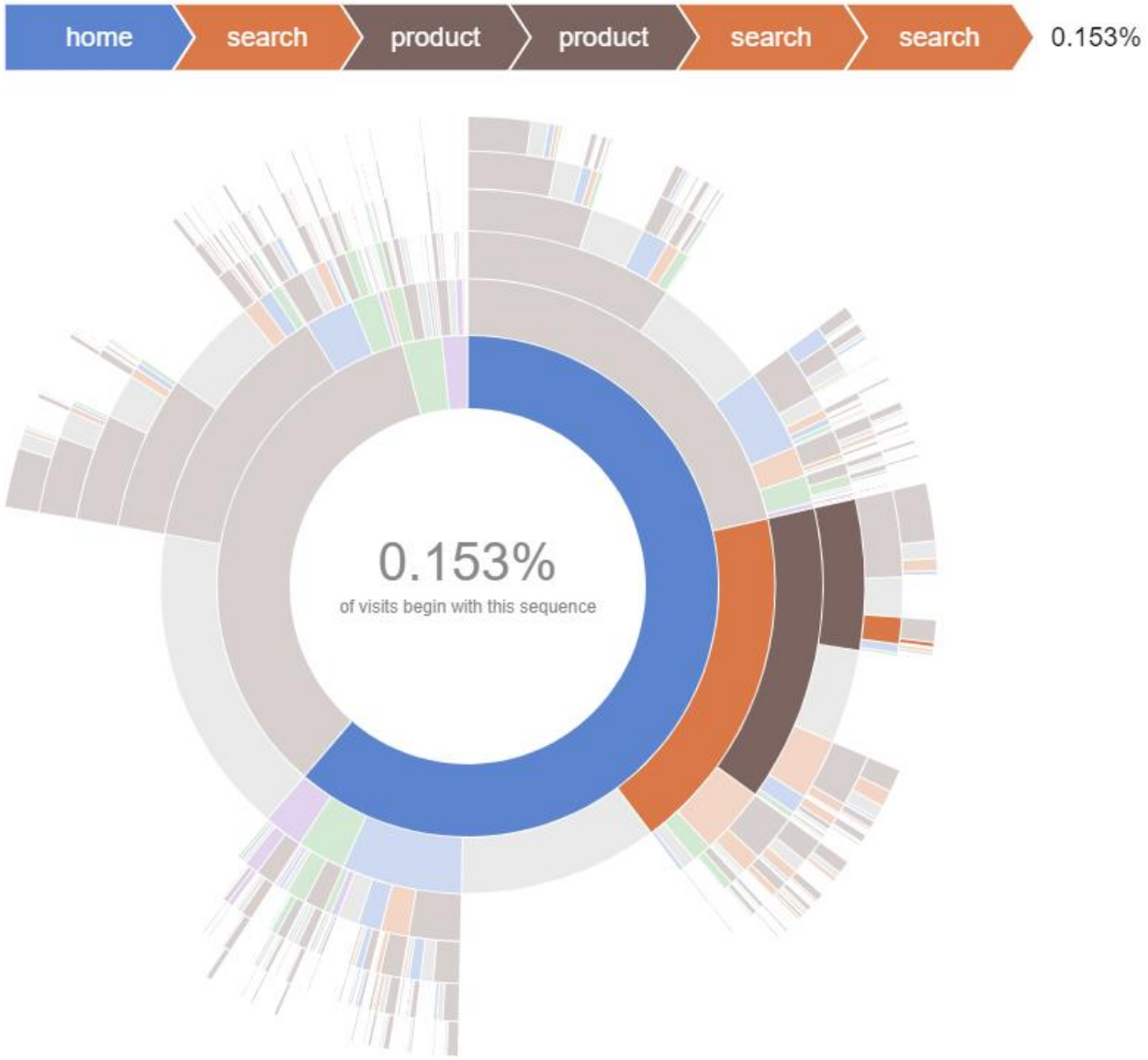
Uses HTML, SVG and CSS

Created by Observable, which also make interactive JavaScript notebooks



UNIVERSITY OF WESTMINSTER





UNIVERSITY OF WESTMINSTER

Useful resources

—

What visualisations can you name?

https://www.menti.com/al3f7cu48hjz

UNIVERSITY OF WESTMINSTER

Let’s ask Tableau

Change over time (line chart)

Correlation (Scatter plot)

Magnitude (Bar chart)

Deviation (Bullet chart)

Distribution (Histogram)

Ranking (Ordered bar chart)

Part-to-whole (Treemap)

Spatial (Symbol map)

Flow (Connected symbol map)

[Choose the Right Chart Type for Your Data – Tableau]

Ggplot2

52 geometries…

UNIVERSITY OF WESTMINSTER

Geoms

Use a geom function to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

ggplot2

GRAPHICAL PRIMITIVES

a <- ggplot(economics, aes(date, unemploy))
b <- ggplot(seals, aes(x = long, y = lat))

a + geom_blank() and **a + expand_limits()**
Ensure limits include values across all plots.

b + geom_curve(aes(yend = lat + 1, xend = long + 1), curvature = 1) - x, xend, y, yend, alpha, angle, color, curvature, linetype, size

a + geom_path(lineend = "butt", linejoin = "round", linemitre = 1)
x, y, alpha, color, group, linetype, size

a + geom_polygon(aes(alpha = 50)) - x, y, alpha, color, fill, group, subgroup, linetype, size

b + geom_rect(aes(xmin = long, ymin = lat, xmax = long + 1, ymax = lat + 1)) - xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size

a + geom_ribbon(aes(ymin = unemploy - 900, ymax = unemploy + 900)) - x, ymax, ymin, alpha, color, fill, group, linetype, size

LINE SEGMENTS

common aesthetics: x, y, alpha, color, linetype, size

b + geom_abline(aes(intercept = 0, slope = 1))
b + geom_hline(aes(yintercept = lat))
b + geom_vline(aes(xintercept = long))

b + geom_segment(aes(yend = lat + 1, xend = long + 1))
b + geom_spoke(aes(angle = 1:1155, radius = 1))

ONE VARIABLE continuous

c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)

c + geom_area(stat = "bin")
x, y, alpha, color, fill, linetype, size

c + geom_density(kernel = "gaussian")
x, y, alpha, color, fill, group, linetype, size, weight

c + geom_dotplot()
x, y, alpha, color, fill

c + geom_freqpoly()
x, y, alpha, color, group, linetype, size

c + geom_histogram(binwidth = 5)
x, y, alpha, color, fill, linetype, size, weight

c2 + geom_qq(aes(sample = hwy))
x, y, alpha, color, fill, linetype, size, weight

discrete

d <- ggplot(mpg, aes(fl))

d + geom_bar()
x, alpha, color, fill, linetype, size, weight

TWO VARIABLES

both continuous

e <- ggplot(mpg, aes(cty, hwy))

e + geom_label(aes(label = cty), nudge_x = 1, nudge_y = 1) - x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

e + geom_point()
x, y, alpha, color, fill, shape, size, stroke

e + geom_quantile()
x, y, alpha, color, group, linetype, size, weight

e + geom_rug(sides = "bl")
x, y, alpha, color, linetype, size

e + geom_smooth(method = lm)
x, y, alpha, color, fill, group, linetype, size, weight

e + geom_text(aes(label = cty), nudge_x = 1, nudge_y = 1) - x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

continuous bivariate distribution

h <- ggplot(diamonds, aes(carat, price))

h + geom_bin2d(binwidth = c(0.25, 500))
x, y, alpha, color, fill, linetype, size, weight

h + geom_density_2d()
x, y, alpha, color, group, linetype, size

h + geom_hex()
x, y, alpha, color, fill, size

continuous function

i <- ggplot(economics, aes(date, unemploy))

i + geom_area()
x, y, alpha, color, fill, linetype, size

i + geom_line()
x, y, alpha, color, group, linetype, size

i + geom_step(direction = "hv")
x, y, alpha, color, group, linetype, size

one discrete, one continuous

f <- ggplot(mpg, aes(class, hwy))

f + geom_col()
x, y, alpha, color, fill, group, linetype, size

f + geom_boxplot()
x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight

f + geom_dotplot(binaxis = "y", stackdir = "center")
x, y, alpha, color, fill, group

f + geom_violin(scale = "area")
x, y, alpha, color, fill, group, linetype, size, weight

visualizing error

df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)
j <- ggplot(df, aes(grp, fit, ymin = fit - se, ymax = fit + se))

j + geom_crossbar(fatten = 2) - x, y, ymax, ymin, alpha, color, fill, group, linetype, size

j + geom_errorbar() - x, ymax, ymin, alpha, color, group, linetype, size, width
Also **geom_errorbarh()**.

j + geom_linerange()
x, ymin, ymax, alpha, color, group, linetype, size

j + geom_pointrange() - x, y, ymin, ymax, alpha, color, fill, group, linetype, shape, size

both discrete

g <- ggplot(diamonds, aes(cut, color))

g + geom_count()
x, y, alpha, color, fill, shape, size, stroke

e + geom_jitter(height = 2, width = 2)
x, y, alpha, color, fill, shape, size

maps

data <- data.frame(murder = USArrests$Murder, state = tolower(rownames(USArrests)))
map <- map_data("state")
k <- ggplot(data, aes(fill = murder))

k + geom_map(aes(map_id = state), map = map) + **expand_limits**(x = map$long, y = map$lat)
map_id, alpha, color, fill, linetype, size

THREE VARIABLES

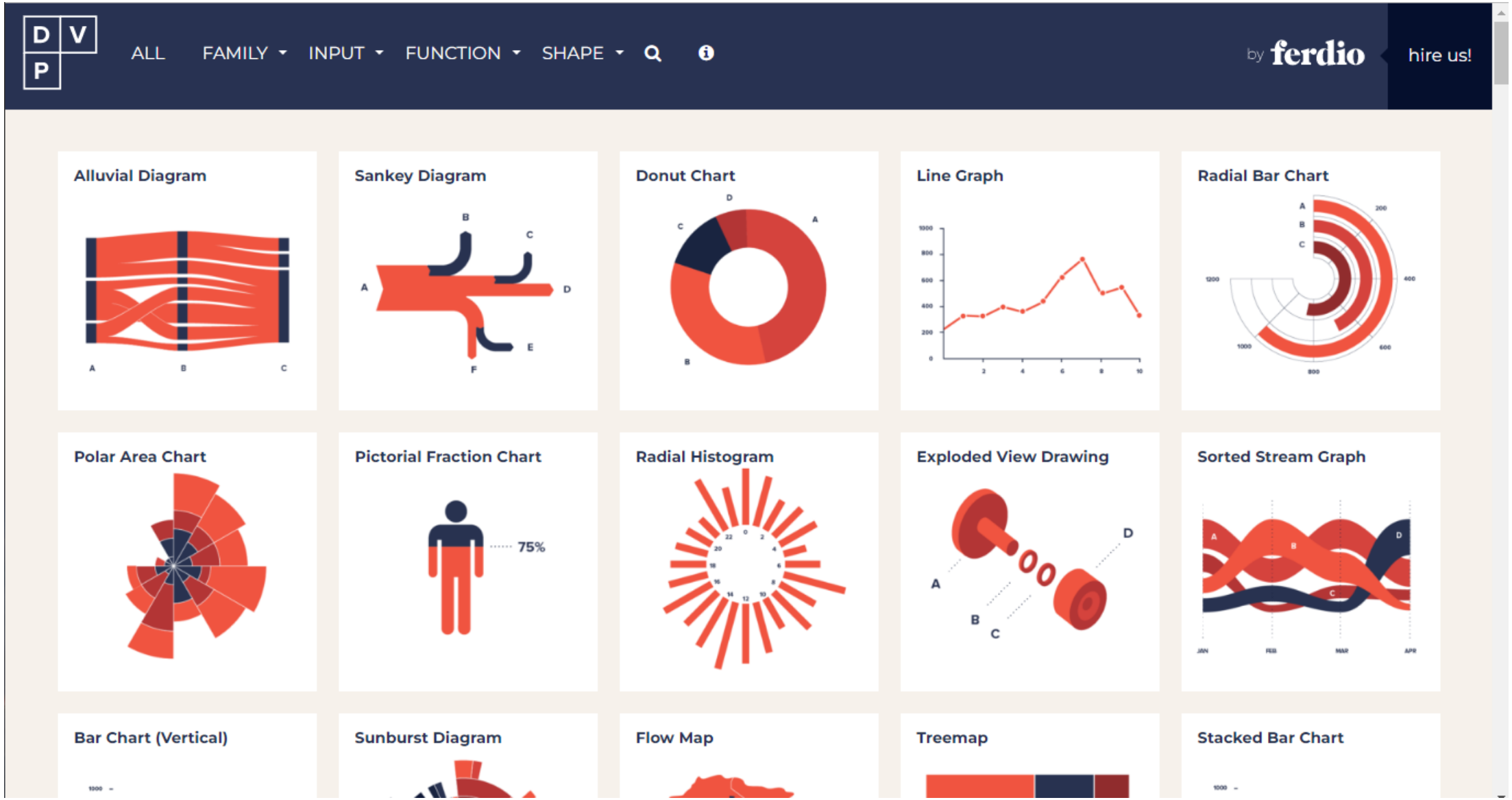
seals$z <- with(seals, sqrt(delta_long^2 + delta_lat^2)); l <- ggplot(seals, aes(long, lat))

l + geom_contour(aes(z = z))
x, y, z, alpha, color, group, linetype, size, weight

l + geom_contour_filled(aes(fill = z))
x, y, alpha, color, fill, group, linetype, size, subgroup

l + geom_raster(aes(fill = z), hjust = 0.5, vjust = 0.5, interpolate = FALSE)
x, y, alpha, fill

l + geom_tile(aes(fill = z))
x, y, alpha, color, fill, linetype, size, width



[Data Viz Project (https://datavizproject.com/)](https://datavizproject.com/)

[The Data Visualisation Catalogue (datavizcatalogue.com)]

from Data to Viz

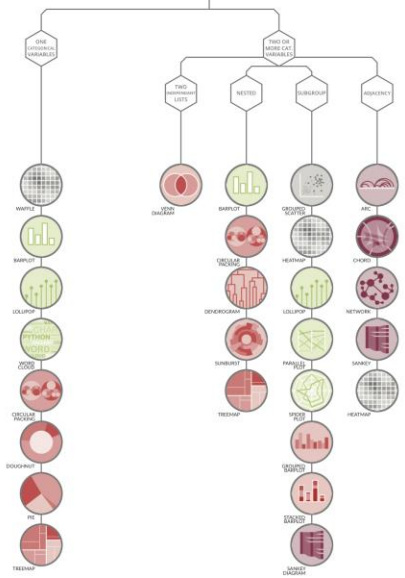
'From Data to Viz' is a classification of chart types based on input data format. It will help you find the perfect chart in three simple steps :

1. Identify what type of data you have.
2. Go to the corresponding decision tree and follow it down to a set of possible charts.
3. Choose the chart from the set that will suit your data and your needs best.

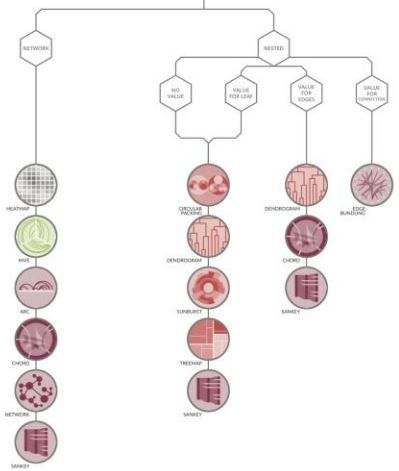
Dataviz is a world with endless possibilities and this project does not claim to be exhaustive. However it should provide you with a good starting point. For an interactive version and much more, visit:

data-to-viz.com

CATEGORIC

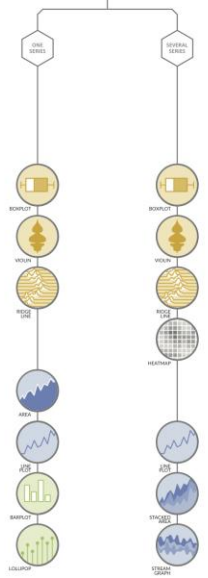


RELATIONAL



MAP

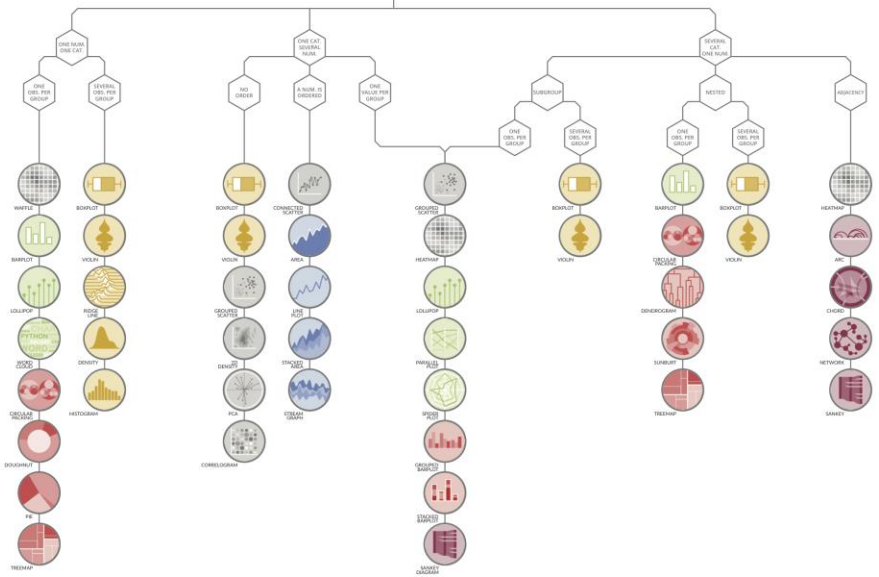
TIME SERIES



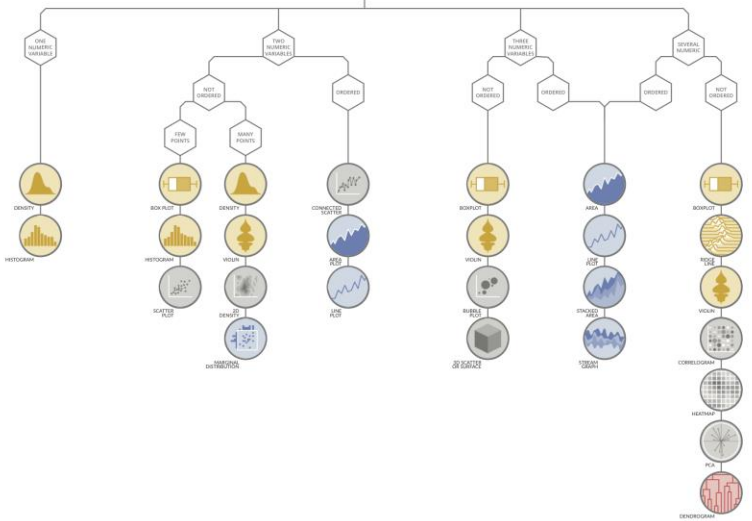
WHAT DO YOU WANT TO SHOW ?

- Distribution
- Correlation
- Ranking
- Part of a whole
- Evolution
- Maps
- Flow

CATEGORIC AND NUMERIC



NUMERIC



[From data to Viz (data-to-viz.com)]

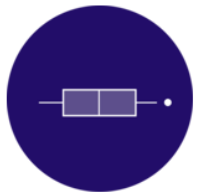
Caveats (or: what not to do and why)

To cut or not to cut?

Cutting the Y-axis is one of the most controversial practice in data viz. See why.

Pie chart

The human eye is bad at reading angles. See how to replace the most criticized chart ever.



Do boxplots hide information?

Boxplots are a great way to summarize a distribution but hide the sample size and their distribution.

The problem with error bars

Barplots with error bars must be used with great care. See why and how to replace them.

The problem with dual axes

Using dual axes is a good way to manipulate the history behind your data. Avoid it. (blog by datawrapper)

The Simpson paradox

When a trend appears in several different groups of data but reverses when these groups are combined

Choropleth and normalization

If you don't scale your data, your choropleth will basically look like a population map.

[Data to Viz | A collection of graphic pitfalls (data-to-viz.com)]

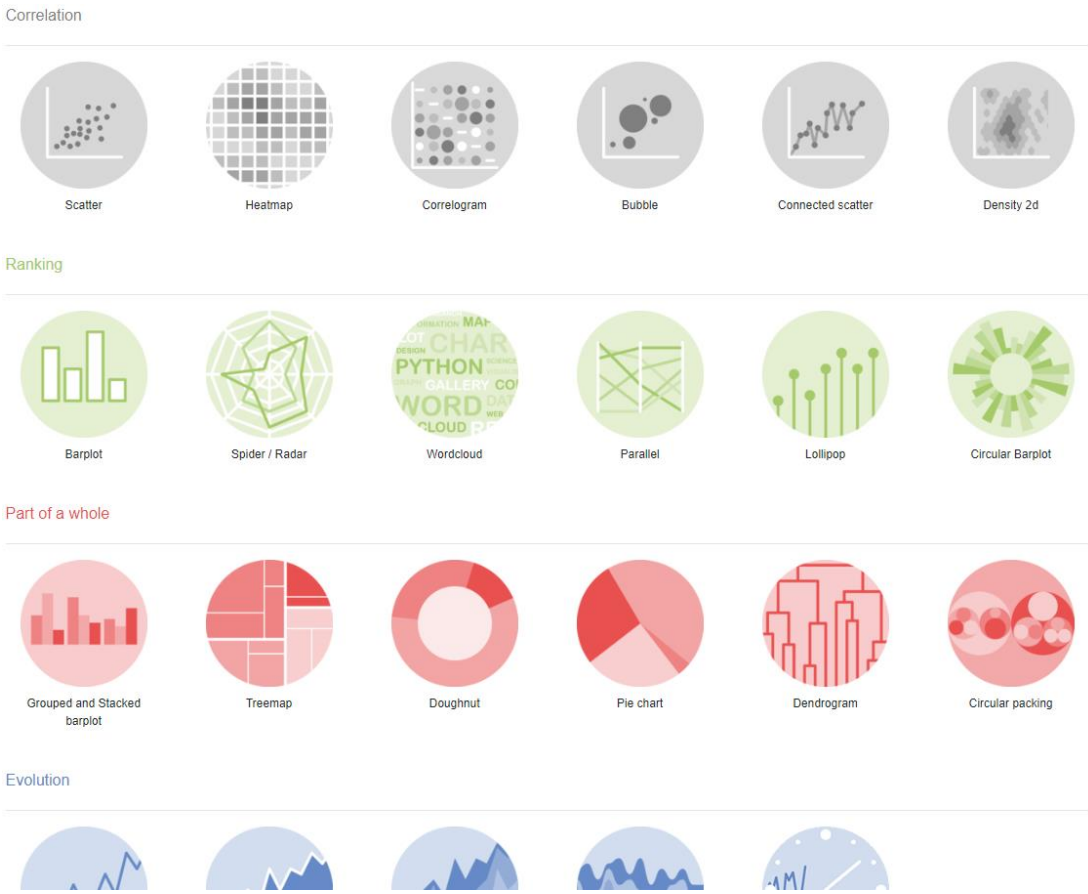
The R graph gallery

[The R Graph Gallery – Help and inspiration for R charts (r-graph-gallery.com)]

Over 400 examples with code

Focus on ggplot2, but covers other packages where appropriate.

Also available for D3.js and Python



UNIVERSITY OF WESTMINSTER

UNIVERSITY OF
WESTMINSTER