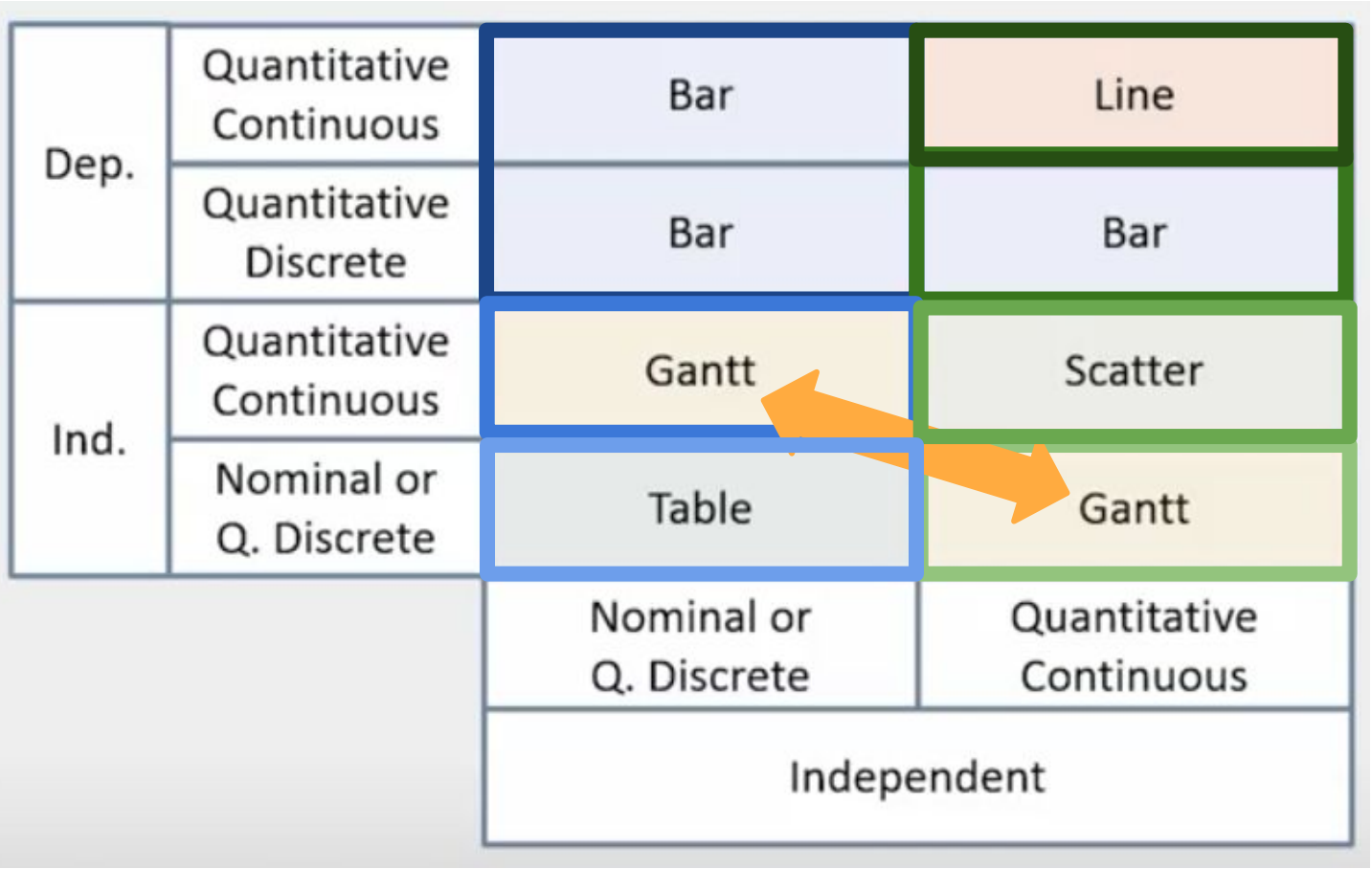
Visualization Pt. 2

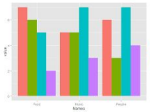
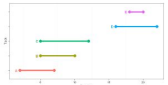
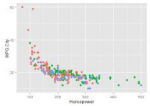
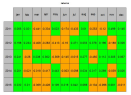
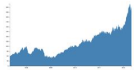
What and How

What to use - basics

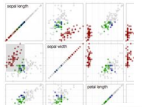
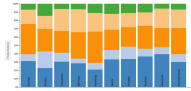
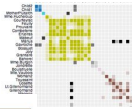
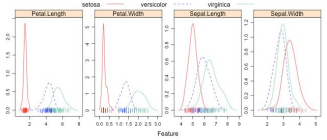


Summary

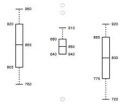
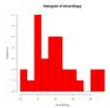
- Basic
  - Line
  - Bar Chart
  - Gantt
  - Scatter Plot
  - Table
  - Area

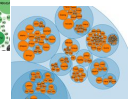
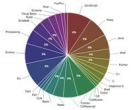
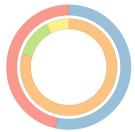
- Relationship
  - Scatter Plot Matrix
  - Stacked Bar
  - Co-occurence
  - Density Plots

- Summary Statistics
  - Box Plots
  - Histogram

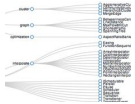



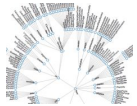
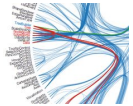
- Grouping / Proportion
  - Bubble Chart
  - Circle Packing
  -
  - Pie
  - Ring/Donut
  -
  - Radar

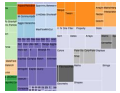





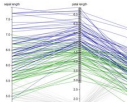

Summary

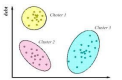
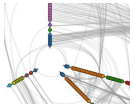
- Hierarchy / Graph
  - Dendrogram
  - Force-Directed Graph
  - Node-Link Tree
  - Hierarchical Edge Bundling
  - Sunbust
  - Treemap
  - Venn Diagram
  - Parallel Coordinates
  - Chord
  - Partition Layout
  - Cluster Layout
  - Hive Plot
  - Collatz Graph
  - Parallel Sets
- Surface
- Geospatial
  - Voronoi Diagram
- Time
  - Sankey Diagram
  - Time-Series Plot

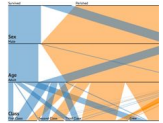


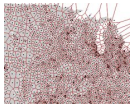
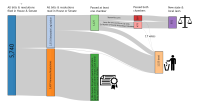
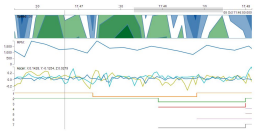









R Plotting

plot()

```
plot(x, hx, type="l", lty=2, xlab="x value",
ylab="Density",
main="Comparison of t Distributions")
```

featurePlot()

```
featurePlot(x = iris[, 1:4],
 y = iris$Species,
 plot = "pairs",
 ## Add a key at the top
 auto.key = list(columns = 3))
```

ggplot2()

```
ggplot(foo, aes(y=Average.Sepal.Width,
x=Species)) + geom_bar(stat="identity") +
coord_flip()
```

qplot()

```
qqplot(rt(1000,df=3), x, main="t(3) Q-Q Plot",
ylab="Sample Quantiles")
abline(0,1)
```

corrgram()

```
corrgram(mtcars, order=TRUE,
lower.panel=panel.shade,
upper.panel=panel.pie,
text.panel=panel.txt,
main="Car Milage Data in PC2/PC1 Order")
```

barplot()

```
barplot(table(mtcars$cyl), main="Plot of …")
```

title()

```
title("This is a title")
```

legend()

```
legend(xrange[1], yrange[2], 1:ntrees,
cex=0.8, col=colors, pch=plotchar,
lty=linetype, title="Tree")
```

D3.js

- https://d3js.org/
- https://github.com/mbostock/d3/wiki/Tutorials
- https://www.dashingd3js.com/table-of-contents
-

Bubbles

view-source:http://mbostock.github.io/d3/talk/20111116/pack-hierarchy.html

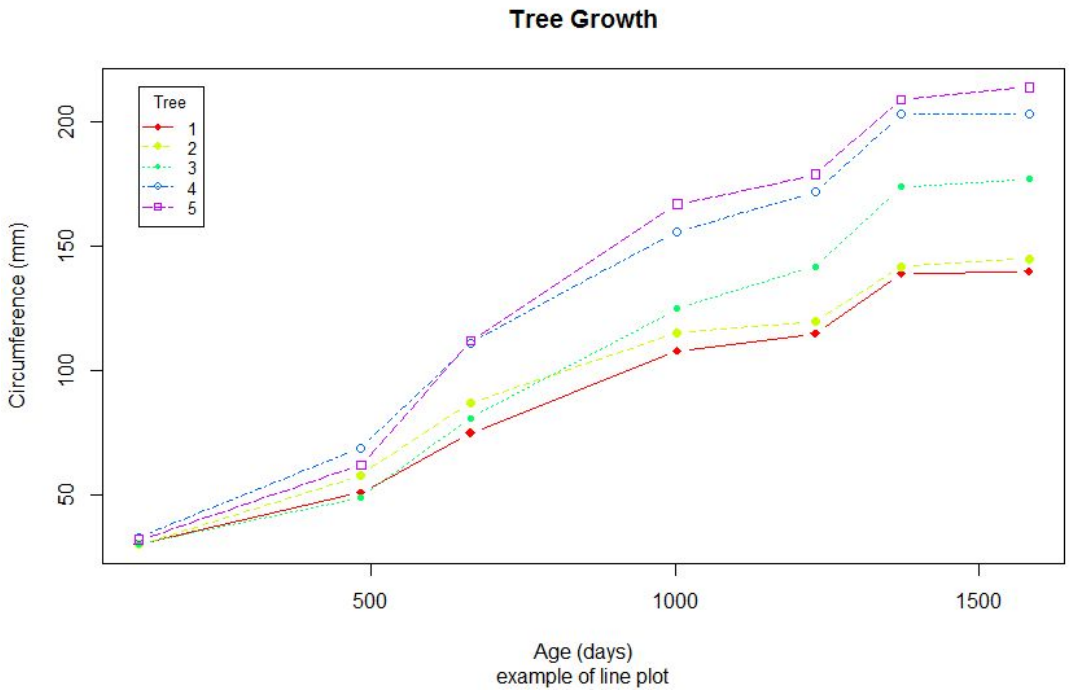
http://mbostock.github.io/d3/talk/20111116/flare.json

Line

Ind. Var (Quant. Cont.) x Dep. Var (Quant. Cont.)

R

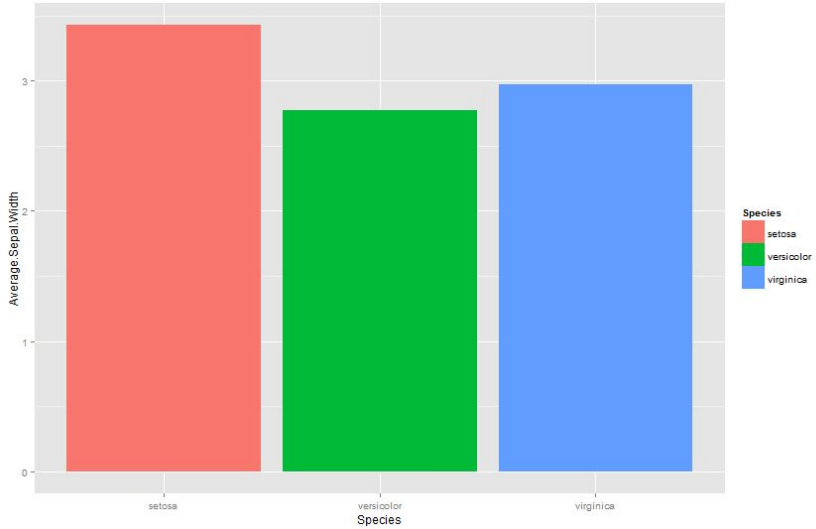
```
# convert factor to numeric for convenience
Orange$Tree <- as.numeric(Orange$Tree)
ntrees <- max(Orange$Tree)

# get the range for the x and y axis
xrange <- range(Orange$age)
yrange <- range(Orange$circumference)

# set up the plot
plot(xrange, yrange, type="n", xlab="Age (days)",
     ylab="Circumference (mm)" )
colors <- rainbow(ntrees)
linetype <- c(1:ntrees)
plotchar <- seq(18,18+ntrees,1)

# add lines
for (i in 1:ntrees) {
  tree <- subset(Orange, Tree==i)
  lines(tree$age, tree$circumference, type="b", lwd=1.5,
    lty=linetype[i], col=colors[i], pch=plotchar[i])
}

# add a title and subtitle
title("Tree Growth", "example of line plot")

# add a legend
legend(xrange[1], yrange[2], 1:ntrees, cex=0.8, col=colors,
     pch=plotchar, lty=linetype, title="Tree")
```

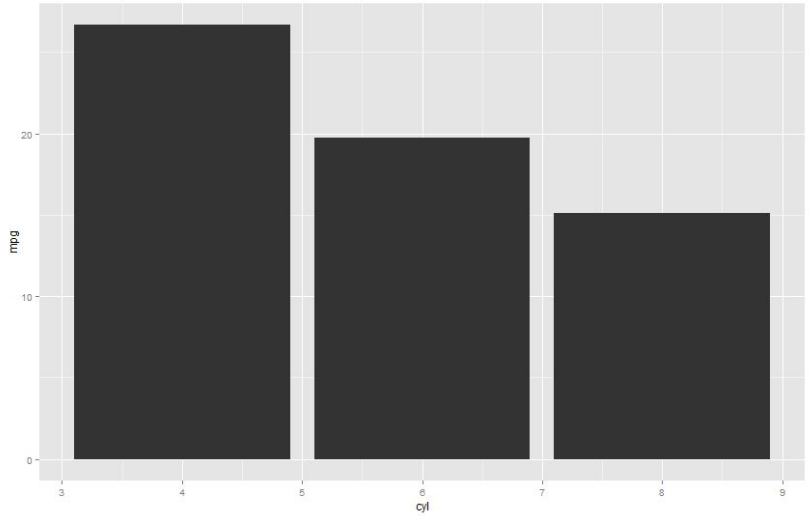


Bar Chart

Ind Var (Nominal, Quant. Discrete) x Dep. Quant. or Ind Var (Quant. Cont.) x Dep. Quant. (Discrete - NOT Cont.)

```
# RIGHT - Nominal vs. Quantitative Continuous
library(ggplot2)
library(dplyr)

foobar <- aggregate(iris$Sepal.Width, list(iris$Species), mean)

foo <- rename(foobar, Species=Group.1, "Average.Sepal.Width"=x)

ggplot(foo, aes(x = Species, y=Average.Sepal.Width, fill=Species)) +
geom_bar(stat ="identity")
```



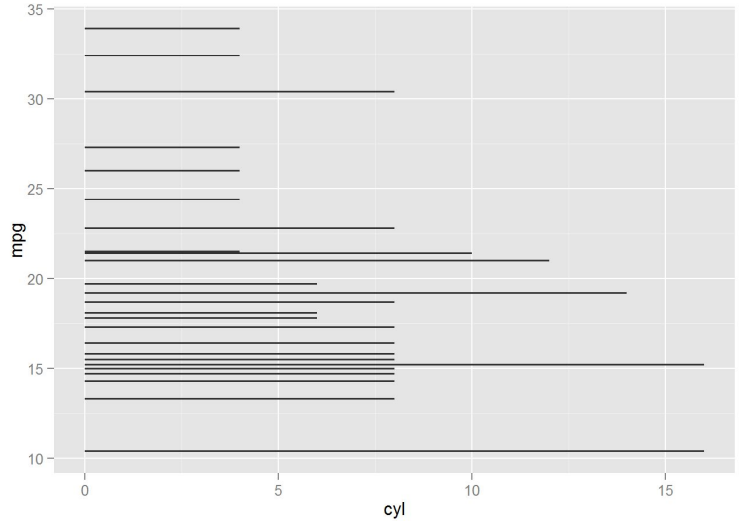



```
# LEFT -  Discrete vs. Continuous
foobar <- mtcars[c("cyl","mpg")]

row.names(foobar) <- NULL

foo <- aggregate(foobar, list(foobar$cyl), mean)

ggplot(foo, aes(x=cyl, y=mpg)) + geom_bar(stat="identity")
```

Gantt

Ind. Var (Nominal or Quant. Disc.) x Ind. Var (Quant. Cont.) or Ind. Var (Quant. Cont.) x Ind. Var (Nominal or Quant. Disc.)

```
# RIGHT - Continuous vs. Nominal

foobar <- aggregate(iris$Sepal.Width, list(iris$Species), mean)
foo <- rename(foobar, Species=Group.1, "Average.Sepal.Width"=x)

# NOTE THAT YOU NEED TO FLIP WHAT WILL BE HORIZONTAL and
VERTICAL AXIS SERIES when YOU USE coord_flip()
ggplot(foo, aes(y=Average.Sepal.Width, x=Species)) + geom_bar(stat="
identity") + coord_flip()
```




```
## LEFT - Continuous vs. Discrete
foobar <- mtcars[c("mpg", "cyl")]
foo <- aggregate(foobar, list(foobar$cyl), mean)

ggplot(foobar, aes(x=mpg, y=cyl)) + geom_bar(stat="identity") +
coord_flip()
```

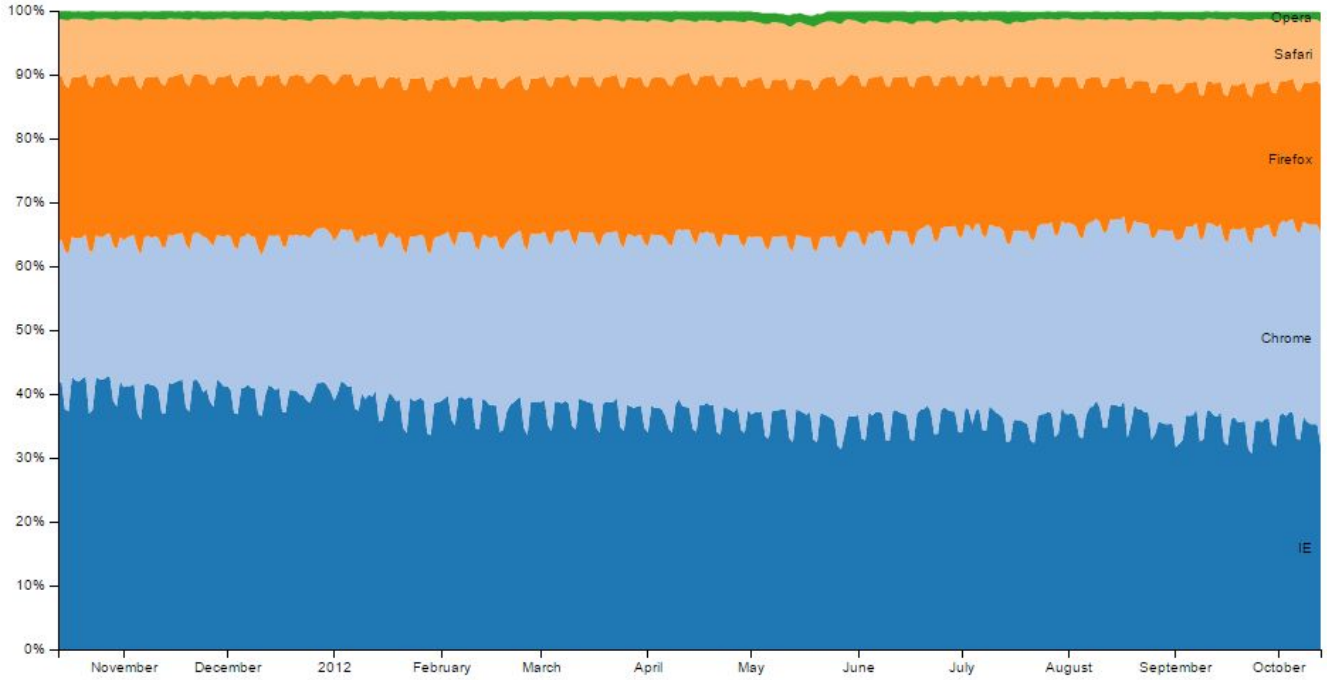
Table

Ind. Var (Nominal, Quant. Disc.) x Ind. Var (Nominal, Quant. Disc.)

	2008				2009				2010				2011			
	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
Active Cases Reported	0	0	0	0	5	5	4	8	2	4	4	13	6	8	4	4
Active Therapy Starts	30	32	36	25	16	24	21	29	12	20	18	32	15	28	23	22
LTBI Therapy Starts	168	204	233	208	204	170	212	145	145	157	229	167	164	195	163	173
Visits	3,146	3,394	3,270	3,603	3,445	3,484	3,337	3,076	2,646	2,552	2,623	2,686	2,689	2,710	2,874	2,518

http://bl.ocks.org/mmparker/3670696

Area

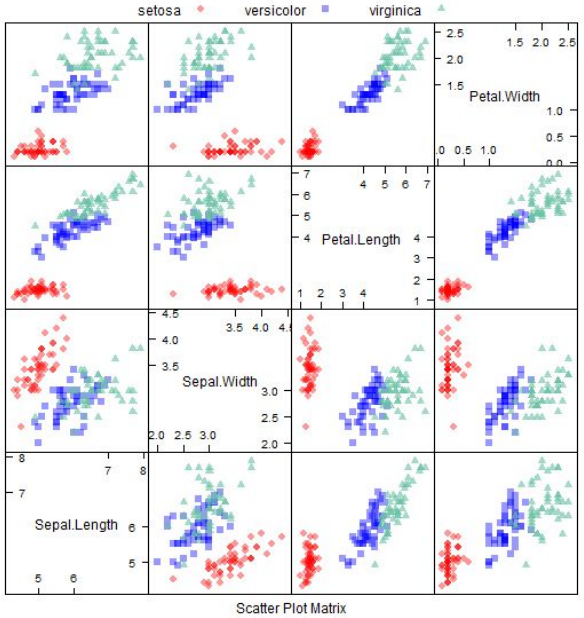


http://bl.ocks.org/mbostock/3885211

Scatter Plot

Ind. Var (Quant. Cont.) x Ind. Var (Quant. Cont.)

```
# RIGHT - Scatter Matrix
library(AppliedPredictiveModeling)
transparentTheme(trans = .4)

library(caret)
featurePlot(x = iris[, 1:4],
            y = iris$Species,
            plot = "pairs",
            ## Add a key at the top
            auto.key = list(columns = 3))
```

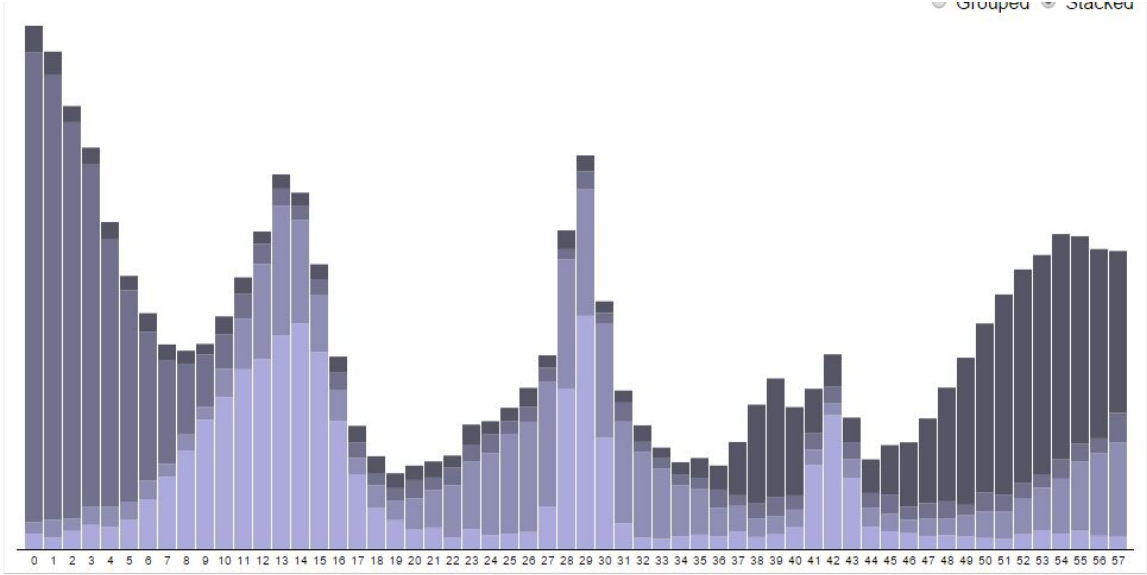



```
# LEFT - Single Scatter Plot
ggplot(mtcars, aes(disp, mpg)) + geom_point(aes(color=cyl))
```

Stacked Bar

Bar graphs can also be used for more complex comparisons of data with grouped bar charts and stacked bar charts.[4] In a **grouped bar chart**, for each categorical group there are two or more bars. These bars are color-coded to represent a particular grouping. For example, a business owner with two stores might make a grouped bar chart with different colored bars to represent each store: the horizontal axis would show the months of the year and the vertical axis would show the revenue. Alternatively, a **stacked bar chart** could be used. The stacked bar chart stacks bars that represent different groups on top of each other. The height of the resulting bar shows the combined result of the groups. However, stacked bar charts are not suited to datasets where some groups have negative values. In such cases, grouped bar charts are preferable.

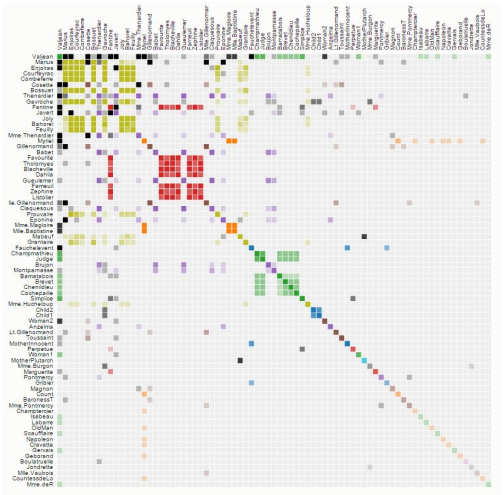
Grouped bar graphs usually present the information in the same order in each grouping. Stacked bar graphs present the information in the same sequence on each bar.



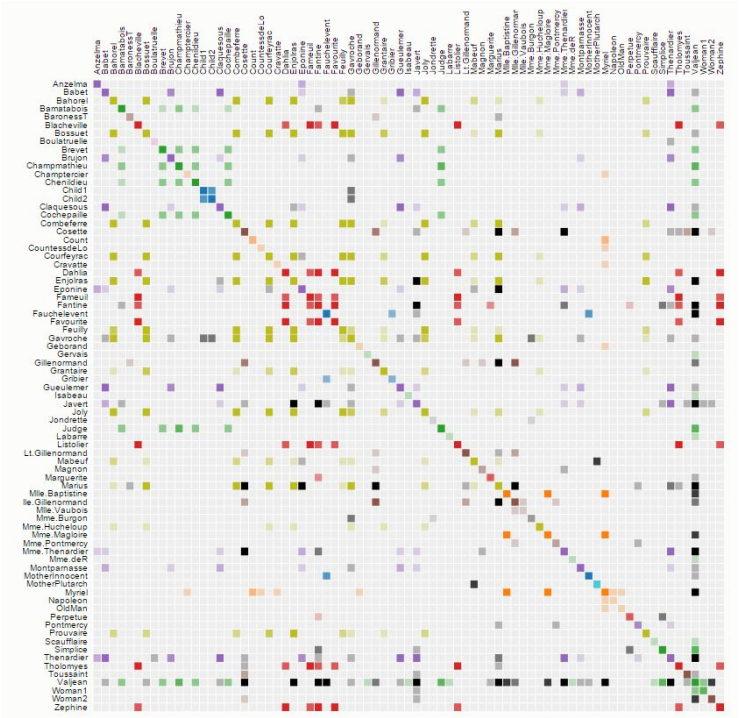
http://bl.ocks.org/mbostock/3943967

https://en.wikipedia.org/wiki/Bar_chart

Co-occurrence



This matrix diagram visualizes character co-occurrences in Victor Hugo's *Les Misérables*.
Each colored cell represents two characters that appeared in the same chapter; darker cells indicate characters that co-occurred more frequently.
Use the drop-down menu to reorder the matrix and explore the data.



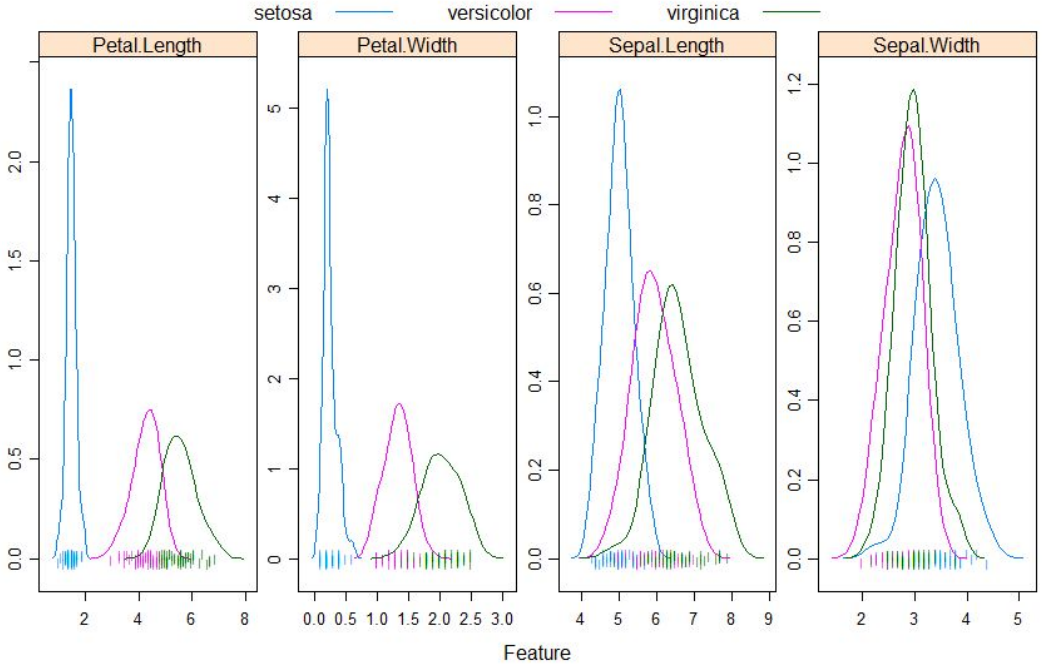
Co-occurrence or **cooccurrence** is a linguistics term that can either mean concurrence / coincidence or, in a more specific sense, the above-chance frequent occurrence of two terms from a text corpus alongside each other in a certain order. Co-occurrence in thislinguistic sense can be interpreted as an indicator of semantic proximity or an idiomatic expression. In contrast to collocation, co-occurrence assumes interdependency of the two terms. A *co-occurrence restriction* is identified when linguistic elements never occur together. Analysis of these restrictions can lead to discoveries about the structure and development of a language.

https://bost.ocks.org/mike/miserables/

https://en.wikipedia.org/wiki/Co-occurrence

Density Plots

```
library(caret)
transparentTheme(trans = .9)
featurePlot(x = iris[, 1:4],
            y = iris$Species,
            plot = "density",
            ## Pass in options to xyplot() to
            ## make it prettier
            scales = list(x = list(relation="free")
                          y = list(relation="free")),
            adjust = 1.5,
            pch = "|",
            layout = c(4, 1),
            auto.key = list(columns = 3))
```

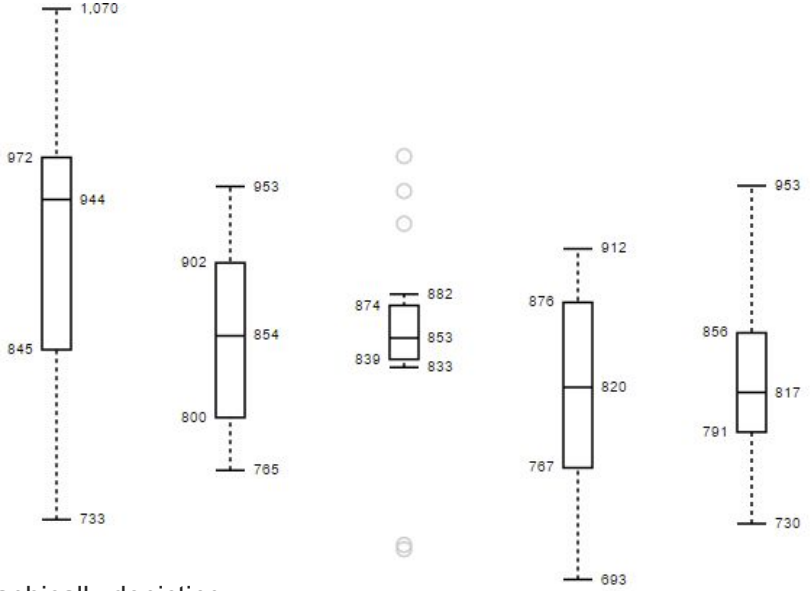


http://topepo.github.io/caret/visualizations.html

R Caret Package using featurePlot using plot=”density”

Box Plots

A box-and-whisker plot uses simple glyphs that summarize a quantitative distribution with five standard statistics: the smallest value, lower quartile, median, upper quartile, and largest value. This summary approach allows the viewer to easily recognize differences between distributions. Data from the Michelson–Morley experiment. Implementation contributed by Jason Davies. This example periodically randomizes the values to demonstrate transitions.



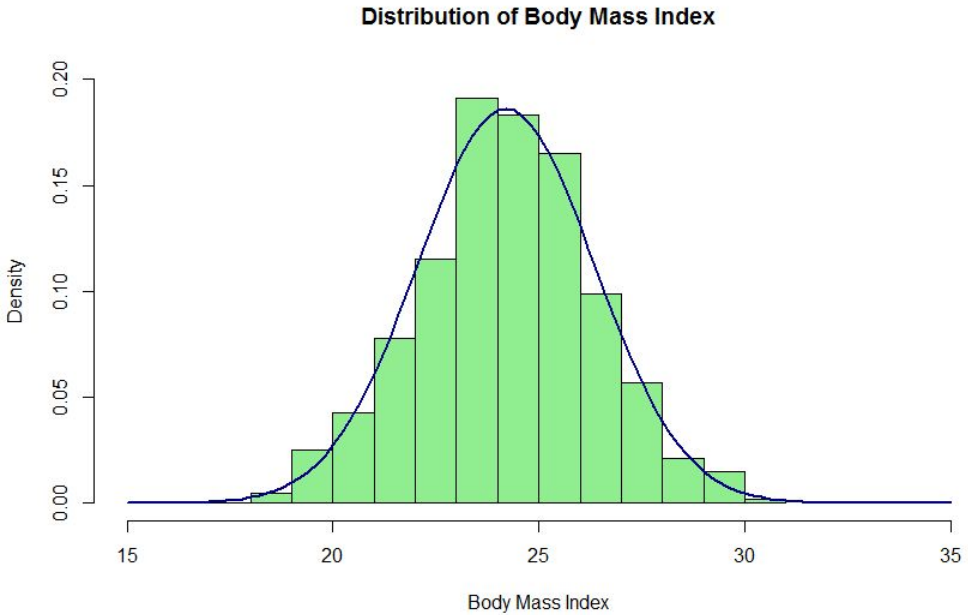
In descriptive statistics, a **box plot** or **boxplot** is a convenient way of graphically depicting groups of numerical data through their quartiles. Box plots may also have lines extending vertically from the boxes (*whiskers*) indicating variability outside the upper and lower quartiles, hence the terms **box-and-whisker plot** and **box-and-whisker diagram**. Outliers may be plotted as individual points. Box plots are non-parametric: they display variation in samples of a statistical population without making any assumptions of the underlying statistical distribution. The spacings between the different parts of the box indicate the degree of dispersion (spread) and skewness in the data, and show outliers. In addition to the points themselves, they allow one to visually estimate various L-estimators, notably the interquartile range, midhinge,range, mid-range, and trimean. Boxplots can be drawn either horizontally or vertically.

https://en.wikipedia.org/wiki/Box_plot

http://bl.ocks.org/mbostock/4061502

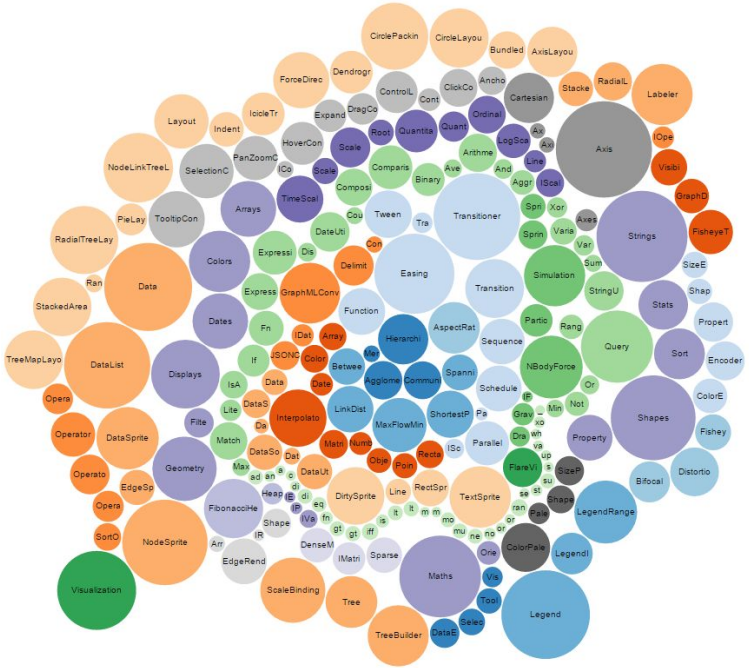
Histograms

```
library(ggplot2)

BMI<-rnorm(n=1000, m=24.2, sd=2.2)

hist(BMI)

hist(BMI, freq=FALSE,
     xlab="Body Mass Index",
     main="Distribution of Body Mass Index",
     col="lightgreen",
     xlim=c(15,35),
     ylim=c(0, .20)
     )
curve(dnorm(x, mean=mean(BMI), sd=sd(BMI)),,
add=TRUE, col="darkblue", lwd=2)
```



Bubble Chart

Bubble charts encode data in the area of circles. Although less perceptually-accurate than bar charts, they can pack hundreds of values into a small space. Implementation based on work by Jeff Heer. Data shows the Flare class hierarchy, also courtesy Jeff Heer.



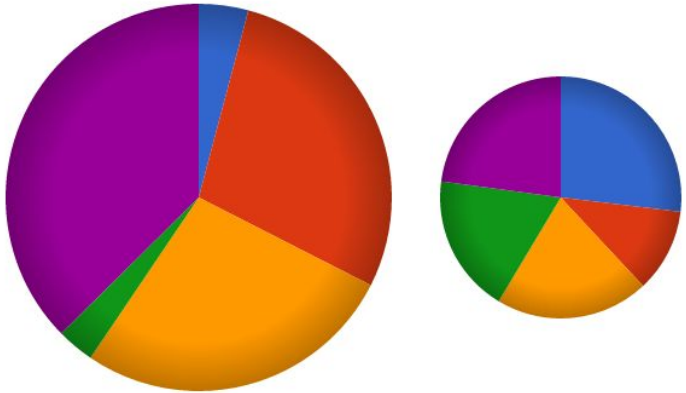
http://bl.ocks.org/mbostock/4063269

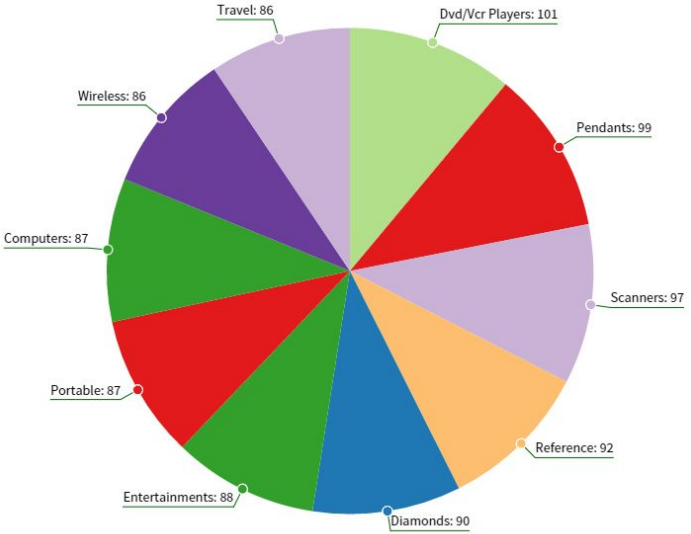
Circle Packing

Enclosure diagrams use containment to represent the hierarchy. Although circle packing is not as space-efficient as a treemap, it better reveals the hierarchy. Implementation based on work by Jeff Heer. Data shows the Flare class hierarchy, also courtesy Jeff Heer.

http://bl.ocks.org/mbostock/4063530

Pie

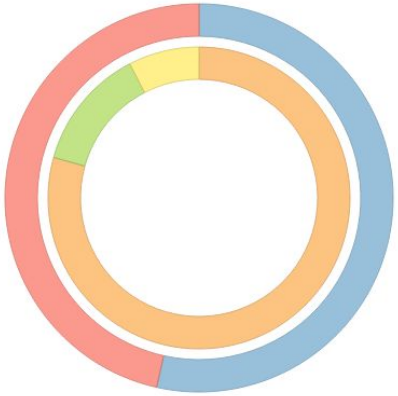




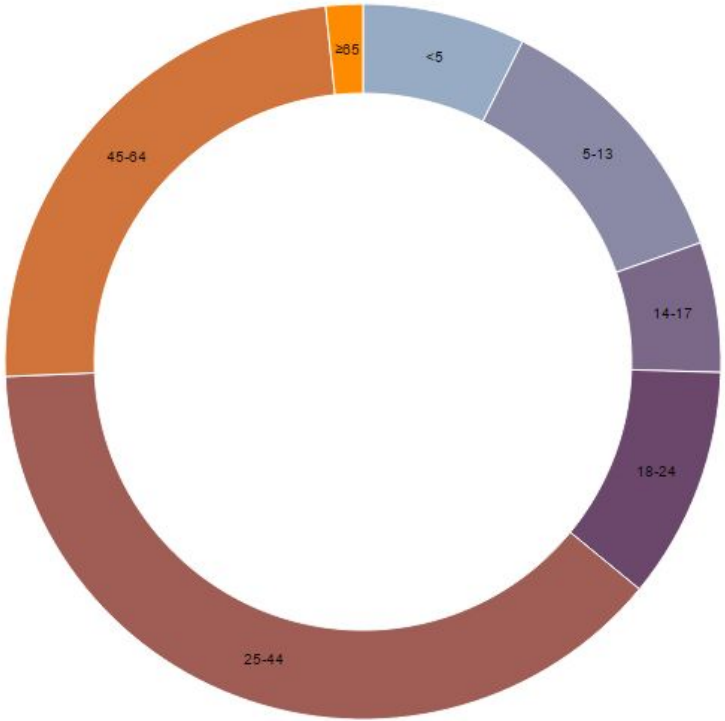
http://bl.ocks.org/NPashaP/9999786

https://en.wikipedia.org/wiki/Pie_chart#Ring_chart_.2F_Sunburst_chart_.2F_Multilevel_pie_chart

Ring / Donut



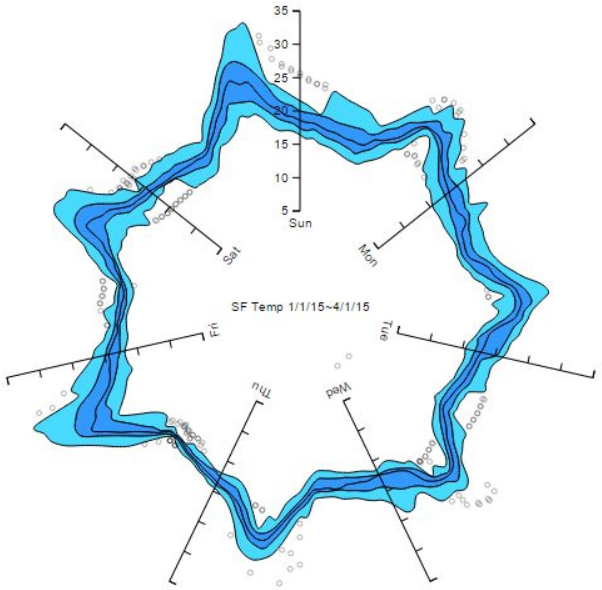
http://dimplejs.org/examples_viewer.html?id=ring_multiple

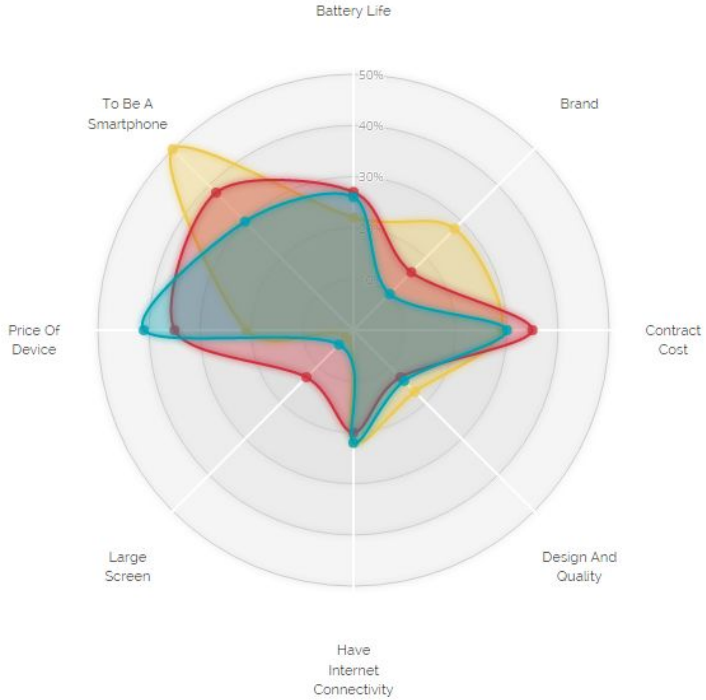


http://bl.ocks.org/mbostock/3887193

https://en.wikipedia.org/wiki/Pie_chart#Ring_chart_.2F_Sunburst_chart_.2F_Multilevel_pie_chart

Radar



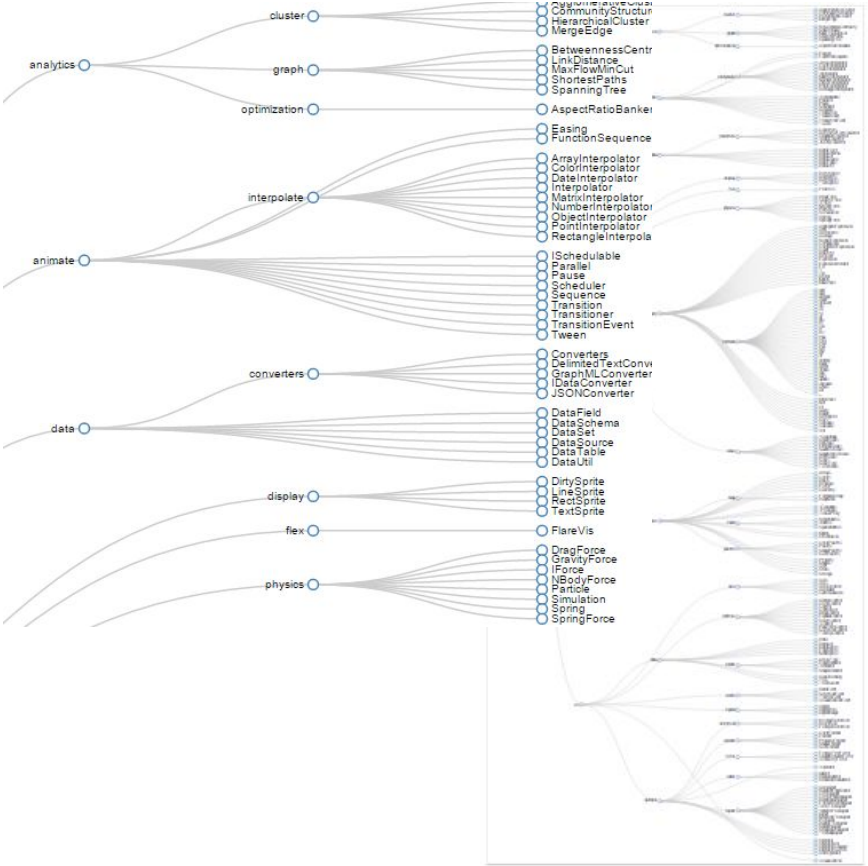


http://bl.ocks.org/davidwclin/ad5d13db260caeffe9b3

http://www.visualcinnamon.com/2015/10/different-look-d3-radar-chart.html

Dendrogram

A dendrogram is a node-link diagram that places leaf nodes of the tree at the same depth. In this example, the classes (leaf nodes) are aligned on the right edge, with the packages (internal nodes) to the left. Data shows the Flare class hierarchy, courtesy Jeff Heer. Compare to this radial layout.



A **dendrogram** (from Greek *dendro* "tree" and *gramma* "drawing") is a tree diagram frequently used to illustrate the arrangement of the clusters produced by hierarchical clustering.[1] Dendrograms are often used in computational biology to illustrate the clustering ofgenes or samples.

https://en.wikipedia.org/wiki/Dendrogram

http://bl.ocks.org/mbostock/4063570

Force-Directed Graph

This simple force-directed graph shows character co-occurence in *Les Misérables*. A physical simulation of charged particles and springs places related characters in closer proximity, while unrelated characters are farther apart. Layout algorithm inspired by Tim Dwyer and Thomas Jakobsen. Data based on character coappearence in Victor Hugo's *Les Misérables*, compiled byDonald Knuth.
Compare this display to a force layout with curved links, a force layout with fisheye distortion and a matrix diagram.

http://bl.ocks.org/mbostock/4062045

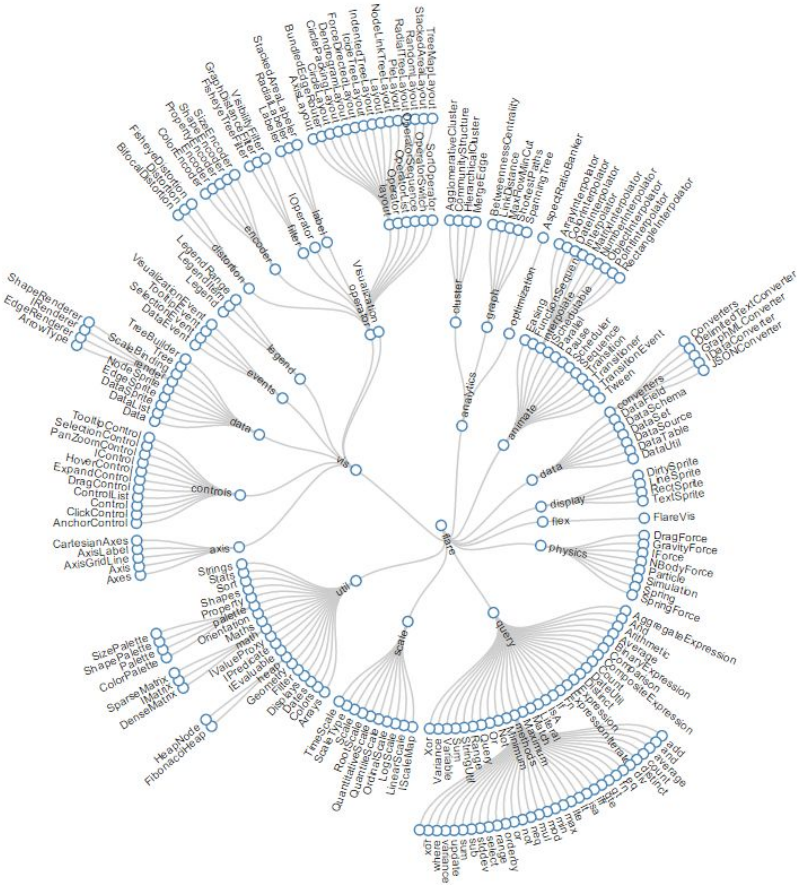
Force-directed graph drawing algorithms are a class of algorithms for drawing graphs in an aesthetically pleasing way. Their purpose is to position the nodes of a graph in two-dimensional or three-dimensional space so that all the edges are of more or less equal length and there are as few crossing edges as possible, by assigning forces among the set of edges and the set of nodes, based on their relative positions, and then using these forces either to simulate the motion of the edges and nodes or to minimize their energy.[1]

While graph drawing can be a difficult problem, force-directed algorithms, being physical simulations, usually require no special knowledge about graph theory such as planarity.

https://en.wikipedia.org/wiki/Force-directed_graph_drawing

Node-Link Tree

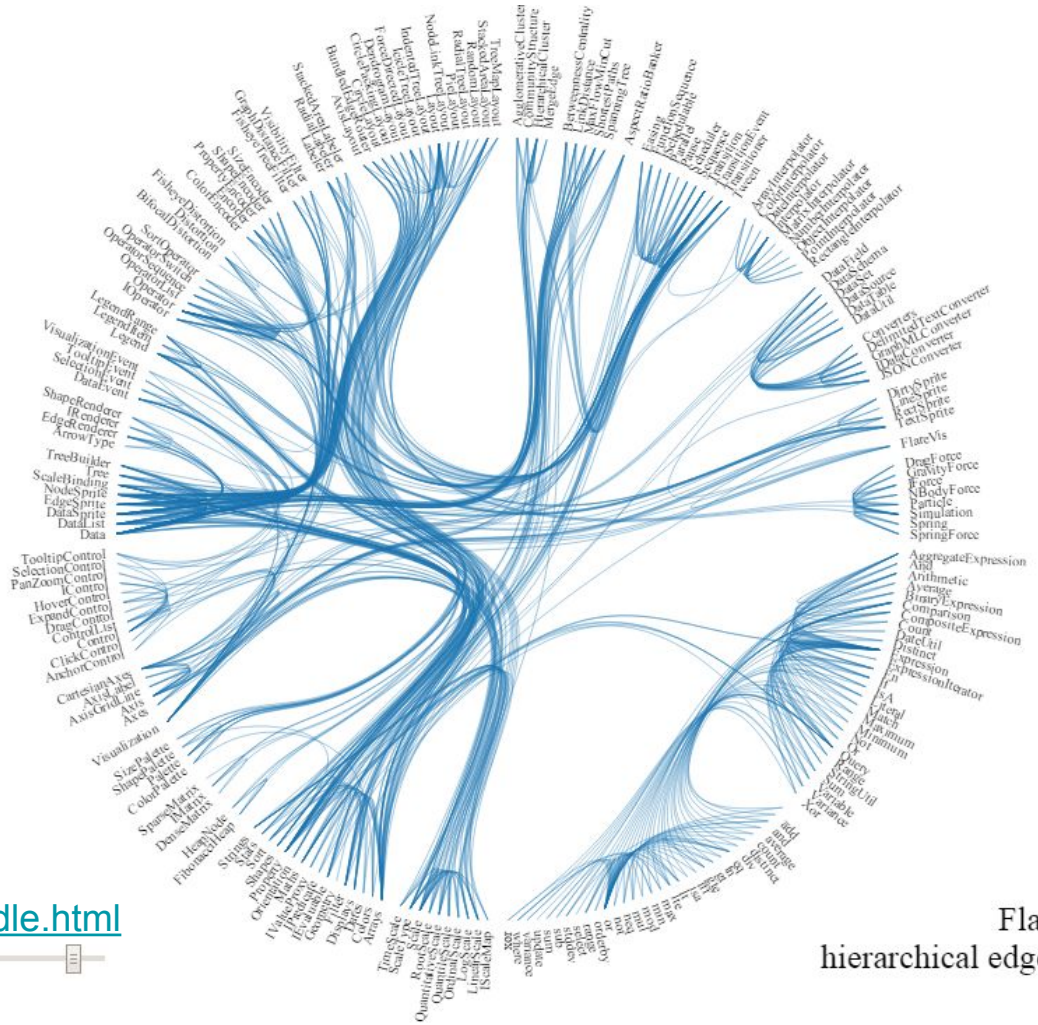
The tree layout implements the Reingold-Tilford algorithm for efficient, tidy arrangement of layered nodes. The depth of nodes is computed by distance from the root, leading to a ragged appearance. Cartesian orientations are also supported. Implementation based on work by Jeff Heer and Jason Davies using Buchheim et al.'s linear-time variant of the Reingold-Tilford algorithm. Data shows theFlare class hierarchy, also courtesy Jeff Heer.



http://bl.ocks.org/mbostock/4063550

Hierarchical Edge Bundling

Similar to Chord but links between pairings show hierarchy of associations - not just point to point.



http://mbostock.github.io/d3/talk/20111116/bundle.html

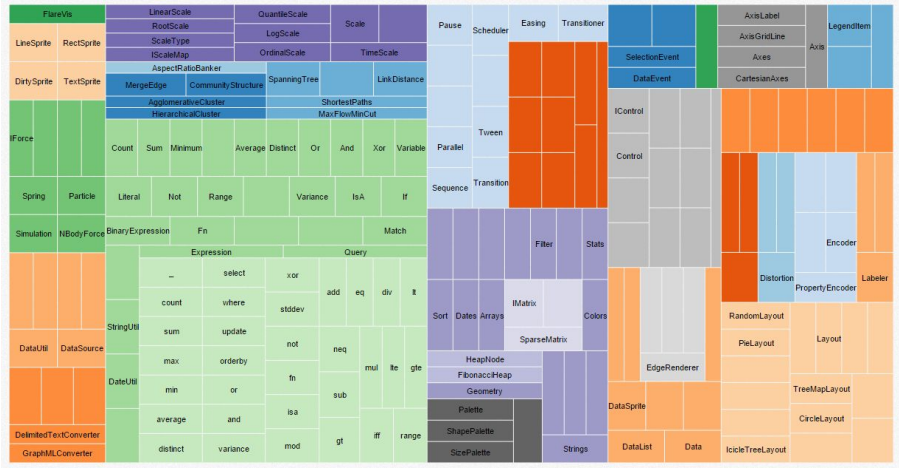
Flar
hierarchical edge

Sunbust

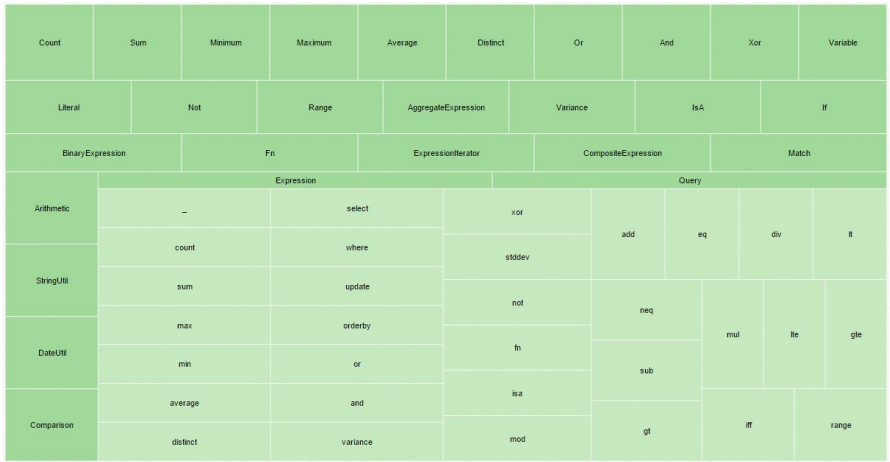
A ring chart, also known as a sunburst chart or a multilevel pie chart, is used to visualize hierarchical data, depicted by concentric circles.[16] The circle in the centre represents the root node, with the hierarchy moving outward from the center. A segment of the inner circle bears a hierarchical relationship to those segments of the outer circle which lie within the angular sweep of the parent segment.[17]

http://bl.ocks.org/kerryrodden/477c1bfb081b783f80ad

Treemap

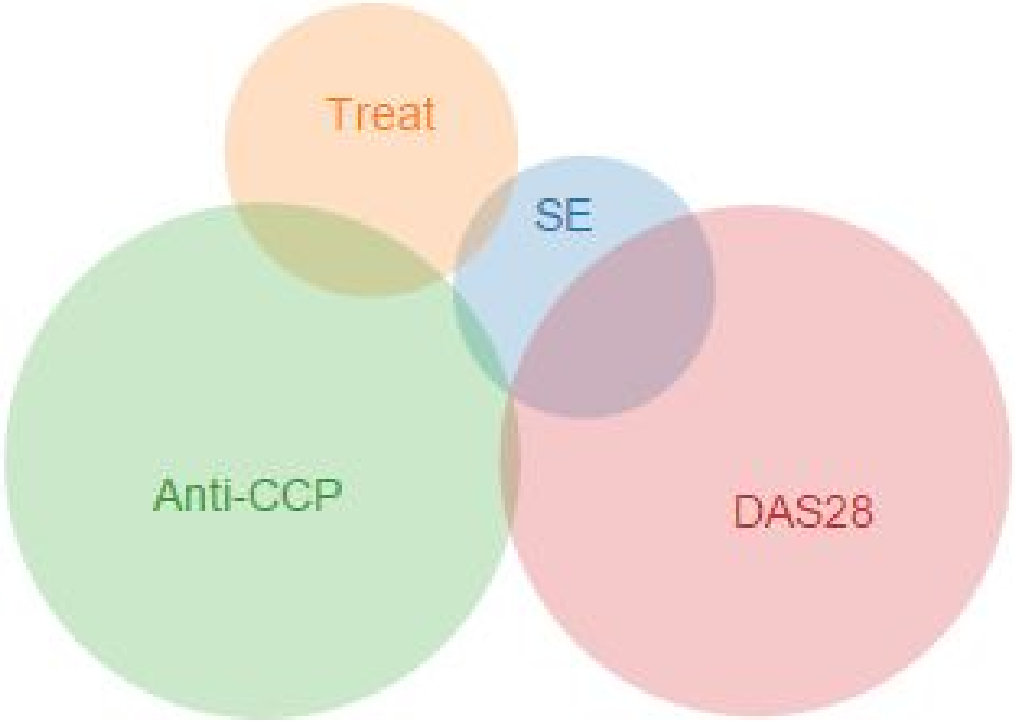


http://bl.ocks.org/mbostock/4063582



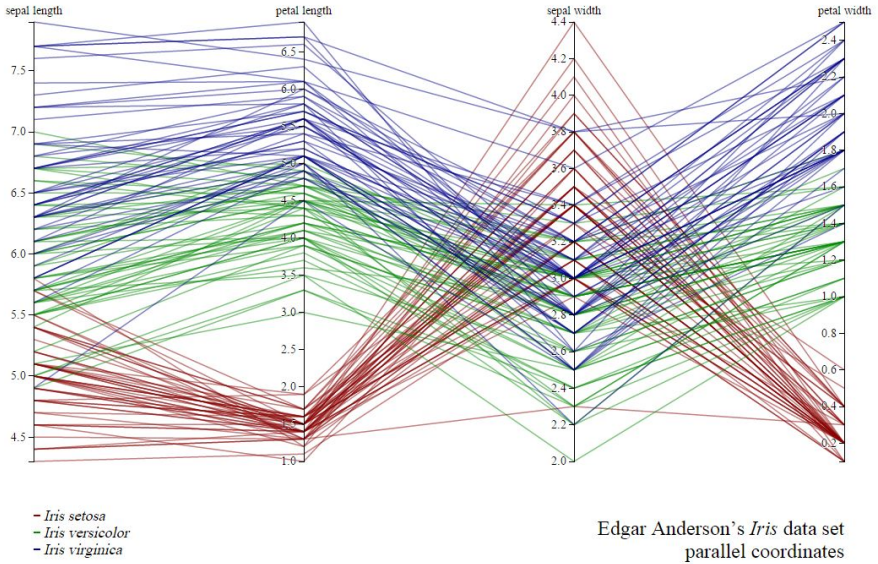
In information visualization and computing, **treemapping** is a method for displaying hierarchical data by using nested rectangles.

Venn Diagram

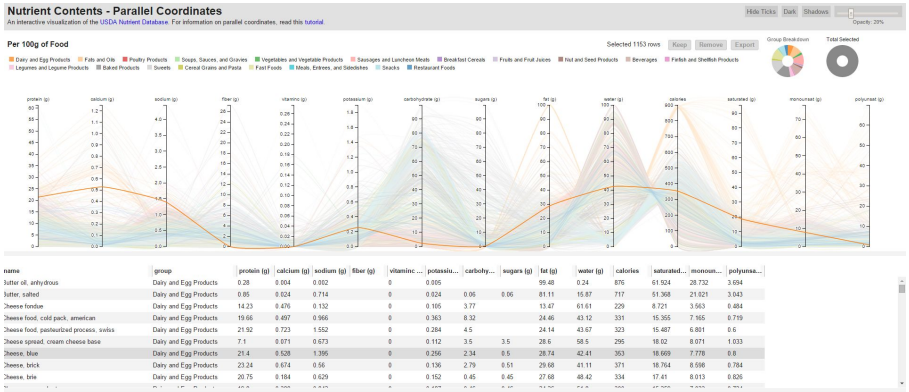


http://www.benfrederickson.com/venn-diagrams-with-d3.js/

Parallel Coordinates



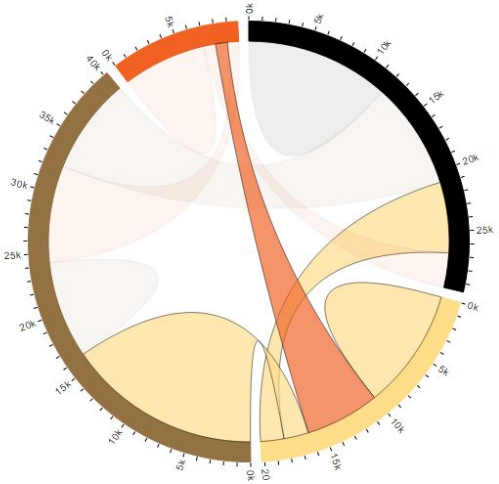
Edgar Anderson's *Iris* data set
parallel coordinates



http://mbostock.github.io/d3/talk/20111116/iris-parallel.html

http://exposedata.com/parallel/

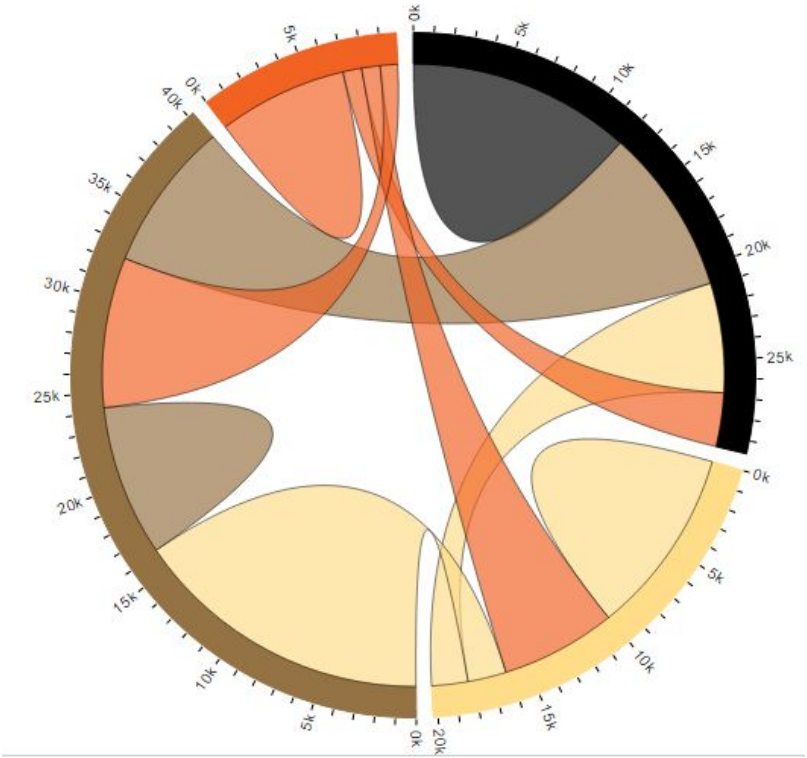
Chord



A **chord diagram** is a graphical method of displaying the inter-relationships between data in a matrix. The data is arranged radially around a circle with the relationships between the points typically drawn as arcs connecting the data together.

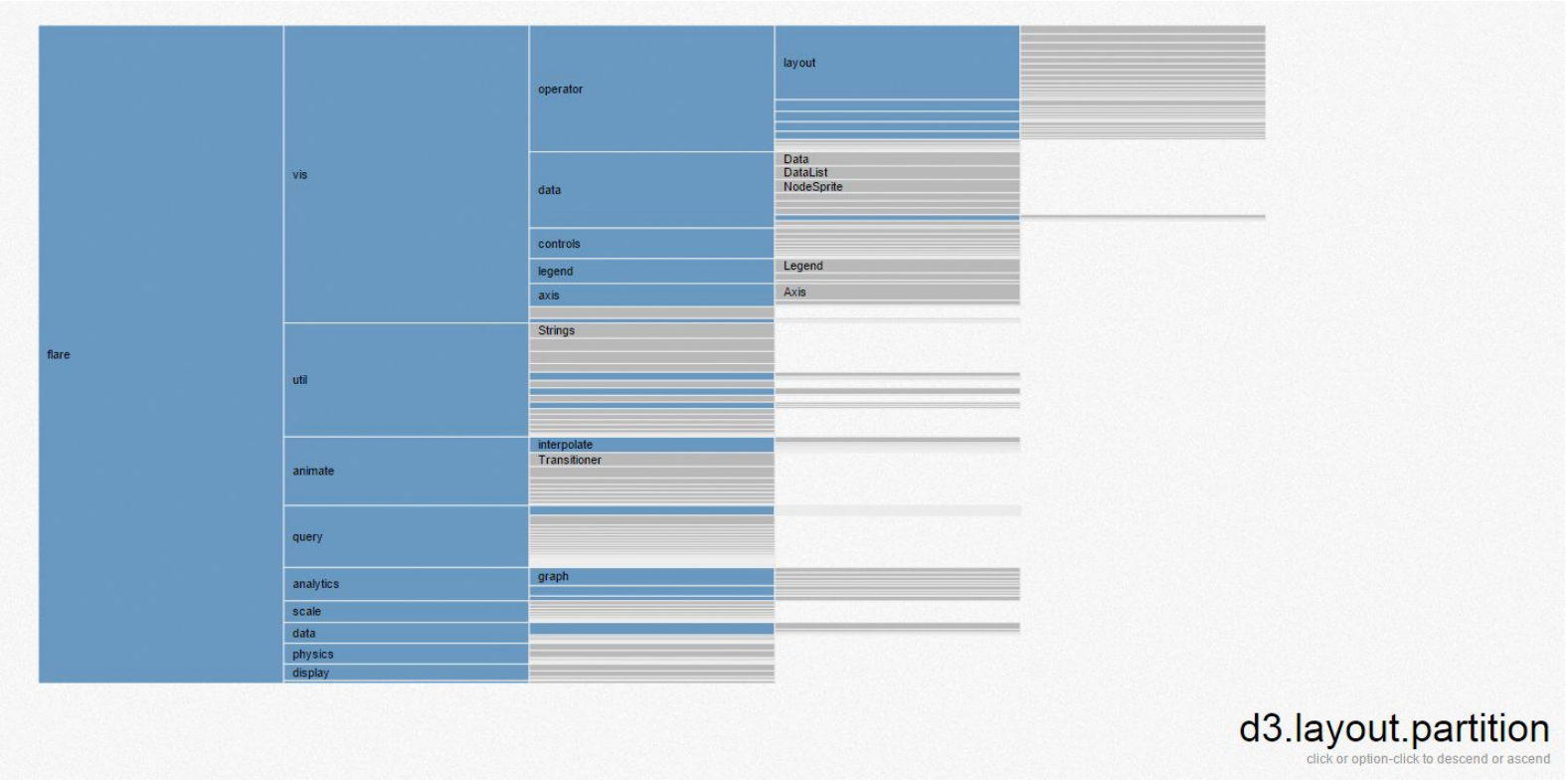
The format can be aesthetically pleasing, making it a popular choice in the world of data visualization.

https://en.wikipedia.org/wiki/Chord_diagram



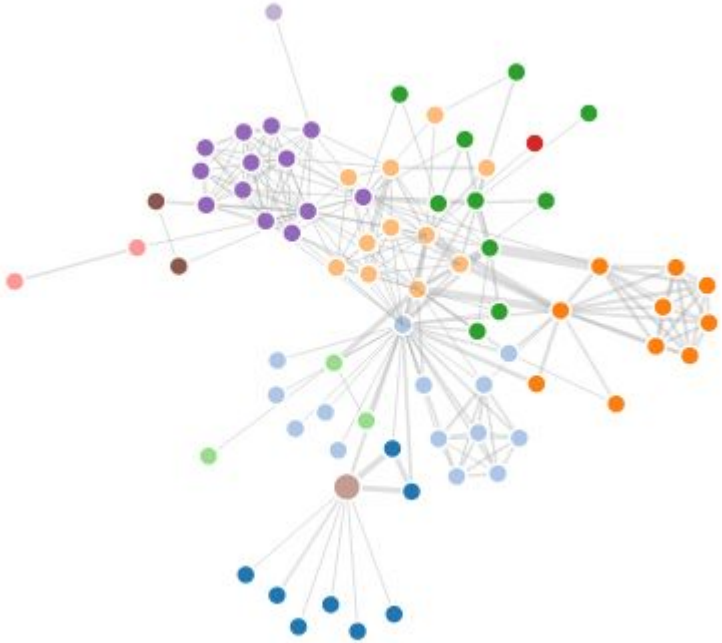
http://bl.ocks.org/mbostock/4062006

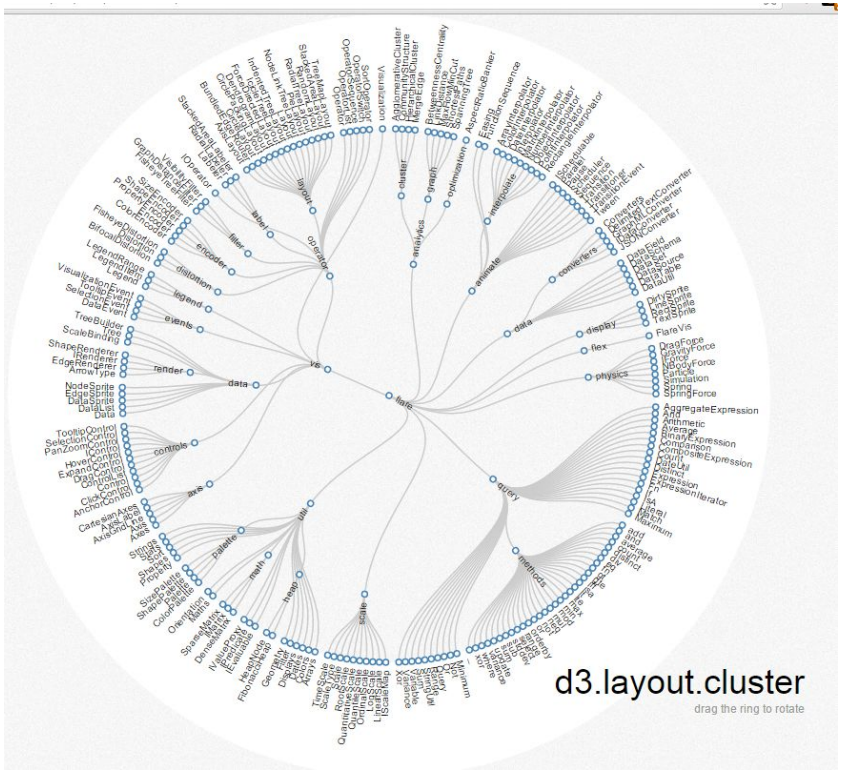
Partition Layout



http://mbostock.github.io/d3/talk/20111018/partition.html

Cluster Layout





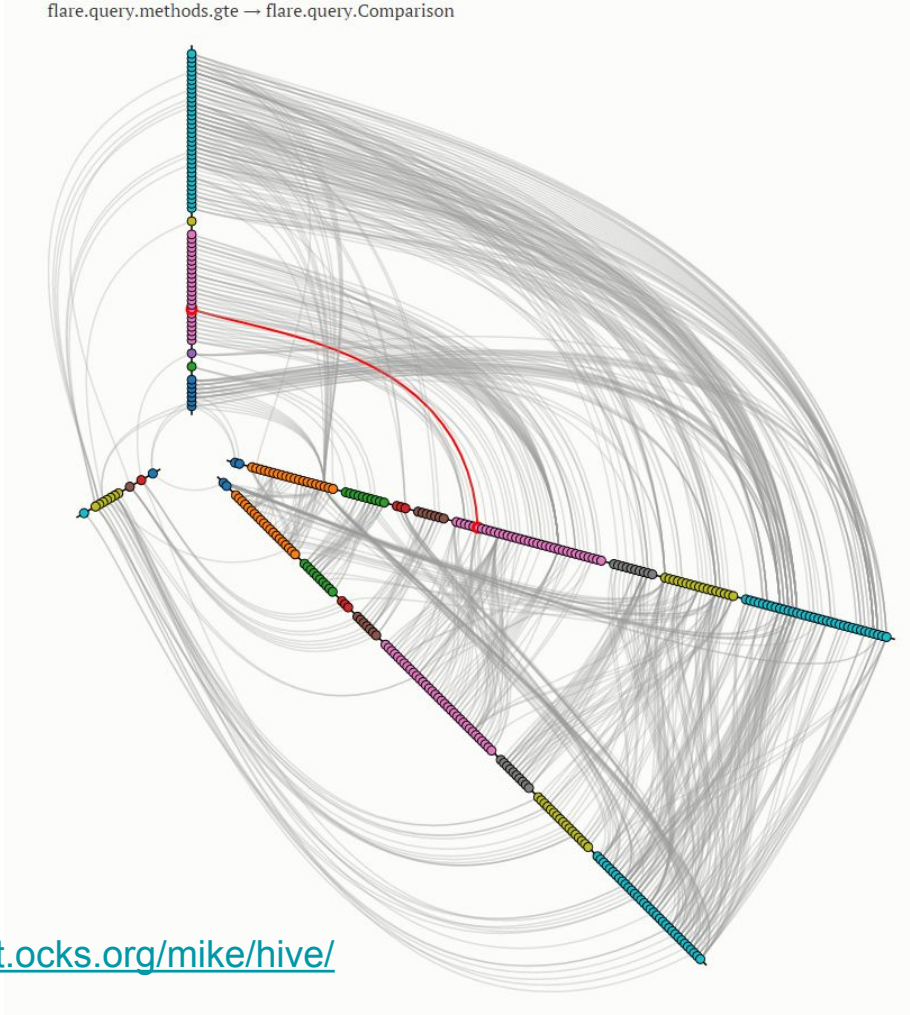
http://bl.ocks.org/4ndr01d3/2686646672a92aa89a0e

http://mbostock.github.io/d3/talk/20111018/cluster.html

Hive Plot

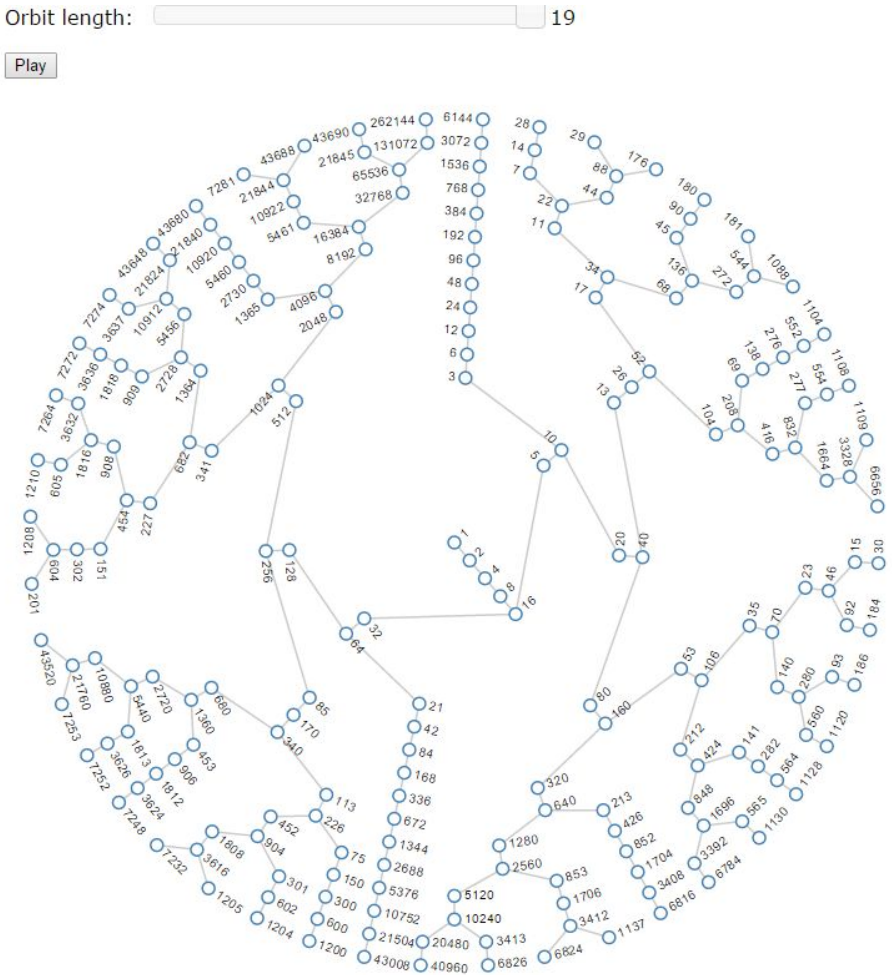
Hive plots define a linear layout for nodes, grouping nodes by type and arranging them along radial axes based on some property of data. The explicit position encoding has the potential to better reveal the network structure while communicating additional information. Hive plots can also be extended to show aggregate relationships.

https://bl.ocks.org/mbostock/2066421



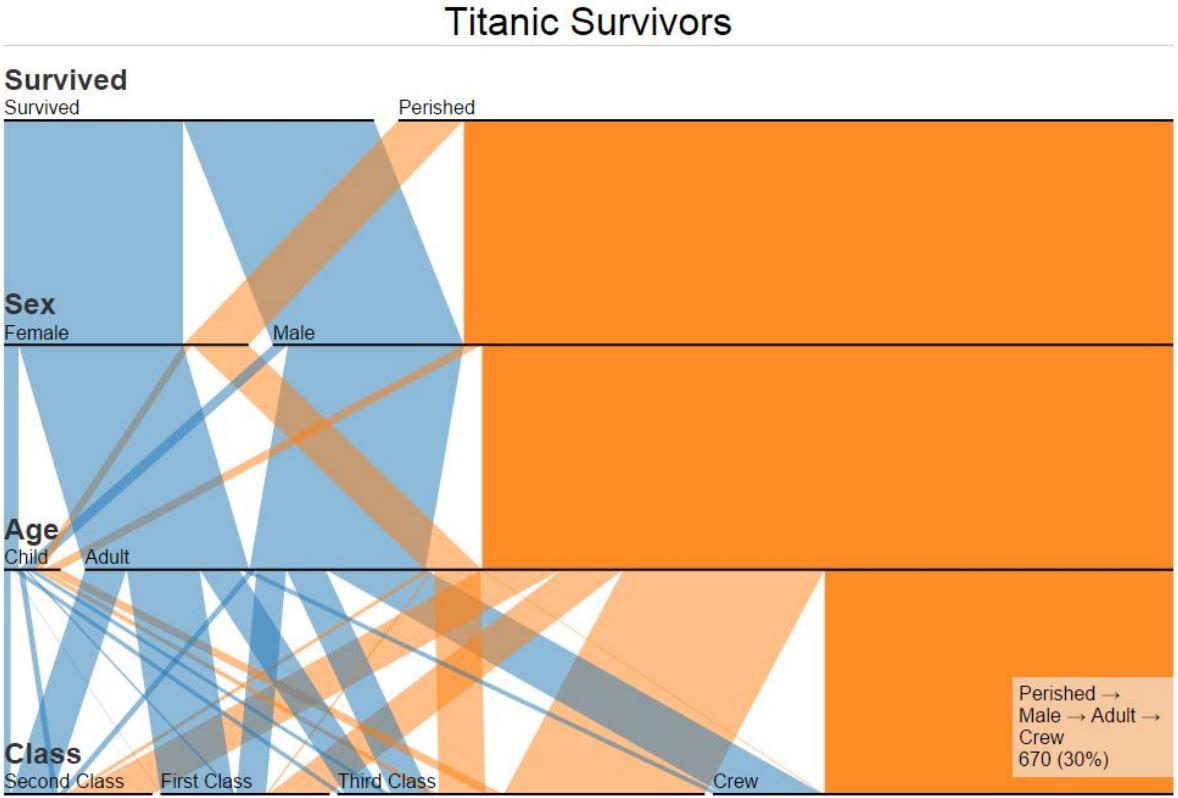
https://bost.ocks.org/mike/hive/

Collatz Graph



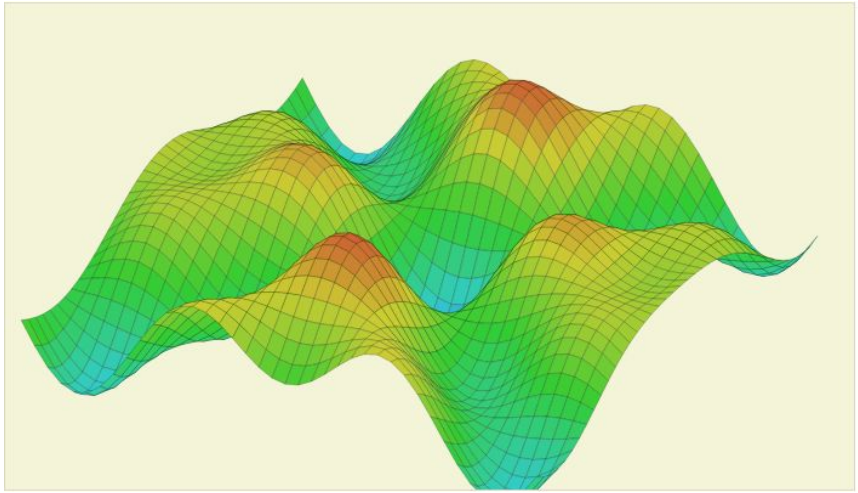
https://www.jasondavies.com/collatz-graph/

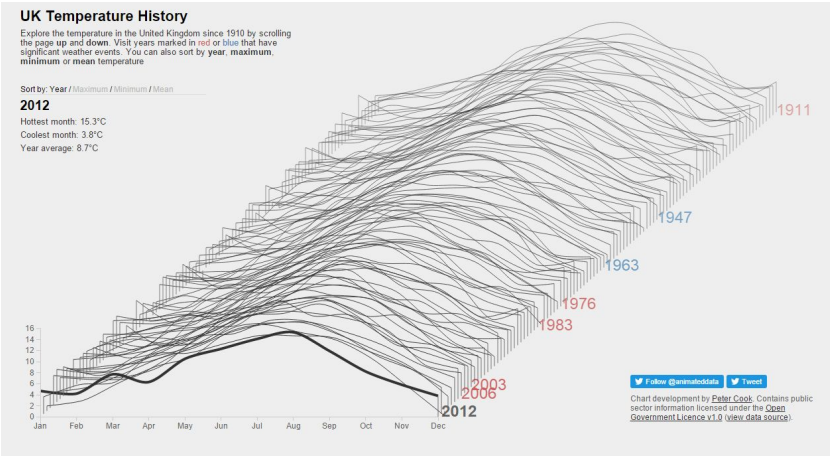
Parallel Sets



https://www.jasondavies.com/parallel-sets/

Surface

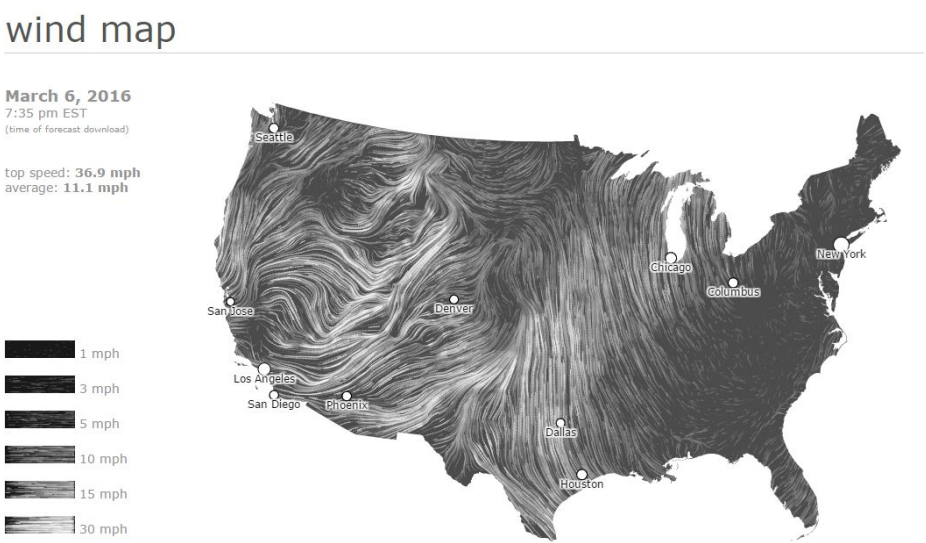




http://bl.ocks.org/supereggbert/aff58196188816576af0

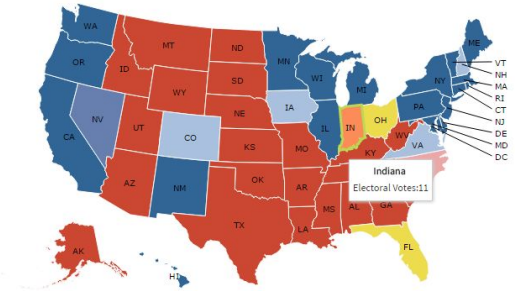
http://charts.animateddata.co.uk/uktemperaturelines/

Geospatial - Data Maps

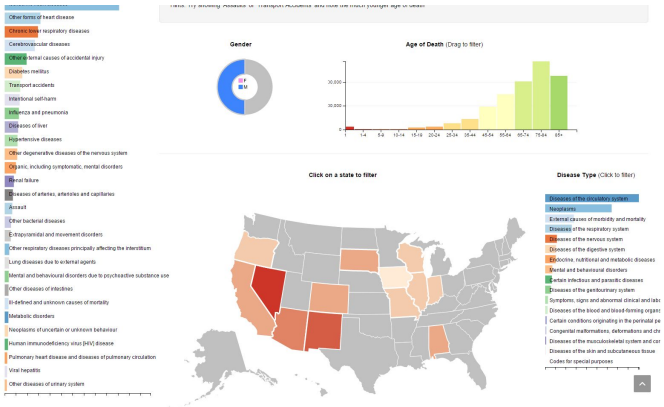


http://charts.animateddata.co.uk/ukwind/

http://hint.fm/wind/

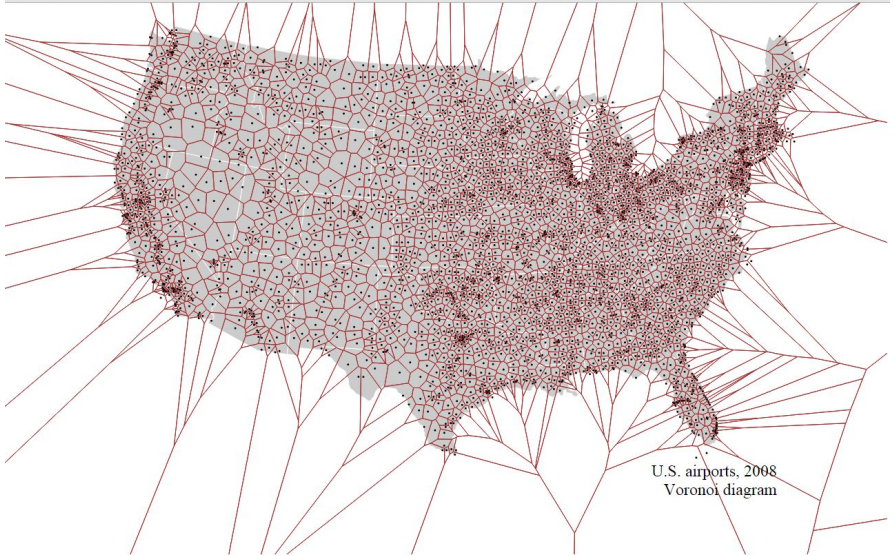


http://datamaps.github.io/

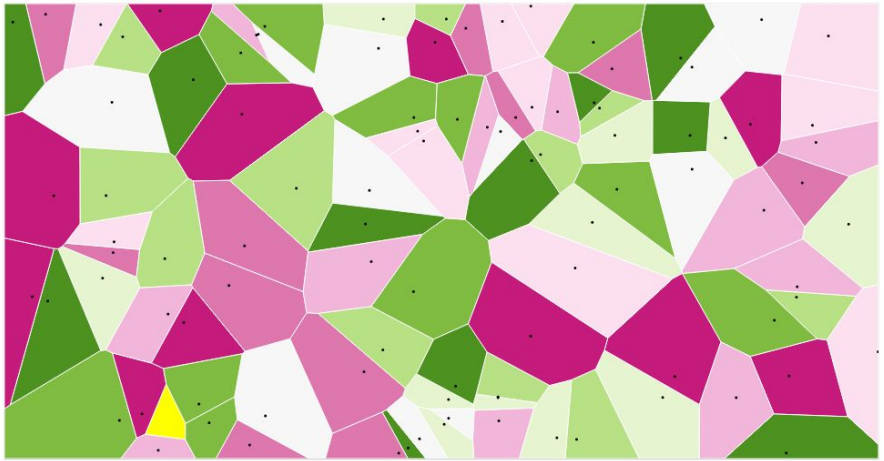


https://www.infino.me/mortality/usmap

Voronoi Diagram



http://mbostock.github.io/d3/talk/20111116/airports-all.html

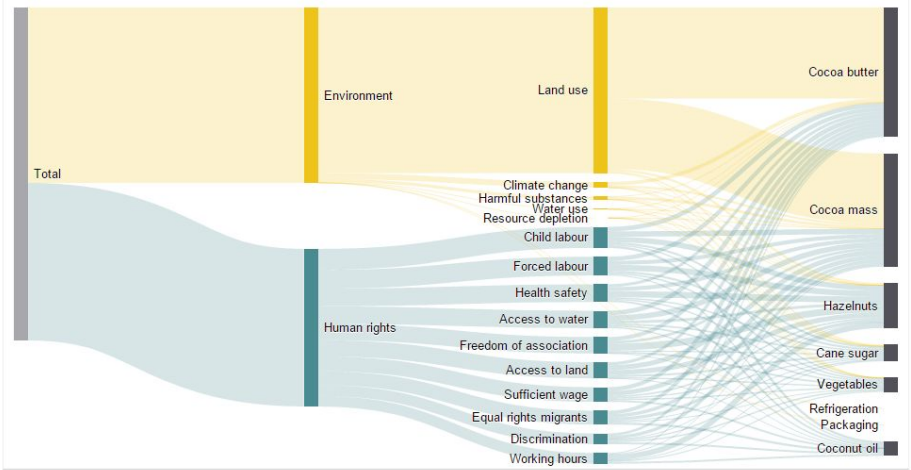


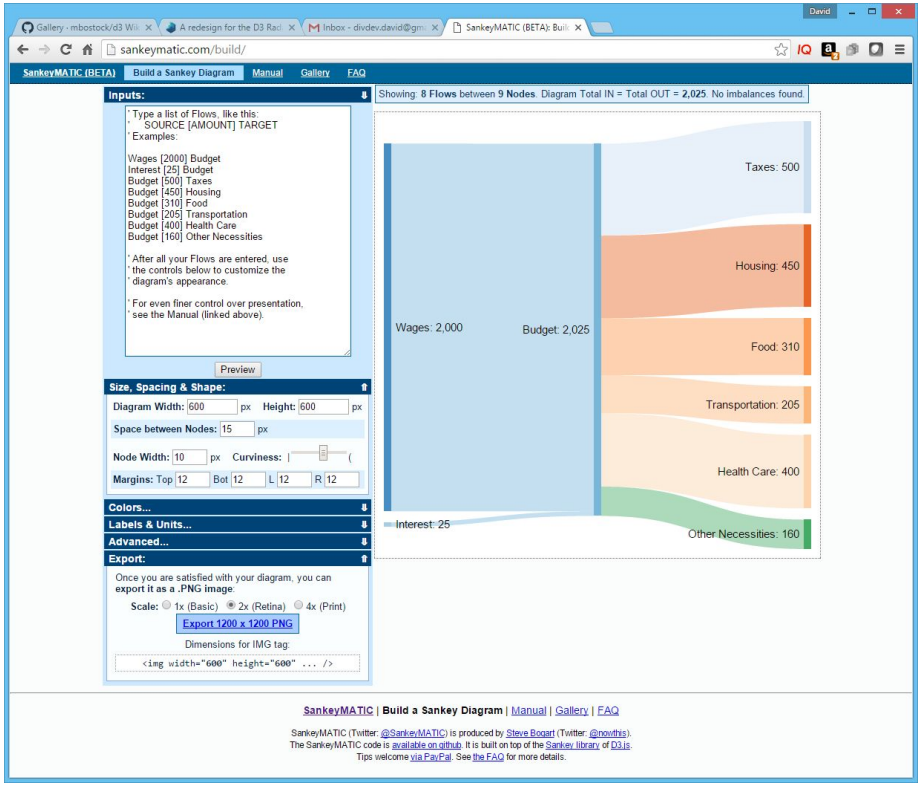
http://bl.ocks.org/mbostock/4060366

In mathematics, a **Voronoi diagram** is a partitioning of a plane into regions based on distance to points in a specific subset of the plane. That set of points (called seeds, sites, or generators) is specified beforehand, and for each seed there is a corresponding region consisting of all points closer to that seed than to any other. These regions are called Voronoi cells. The Voronoi diagram of a set of points is dual to its Delaunay triangulation. It is named after Georgy Voronoi, and is also called a **Voronoi tessellation**, a **Voronoi decomposition**, a **Voronoi partition**, or a **Dirichlet tessellation** (after Peter Gustav Lejeune Dirichlet). Voronoi diagrams have practical and theoretical applications to a large number of fields, mainly in science and technology but also including visual art.[

https://en.wikipedia.org/wiki/Voronoi_diagram

Sankey Diagram

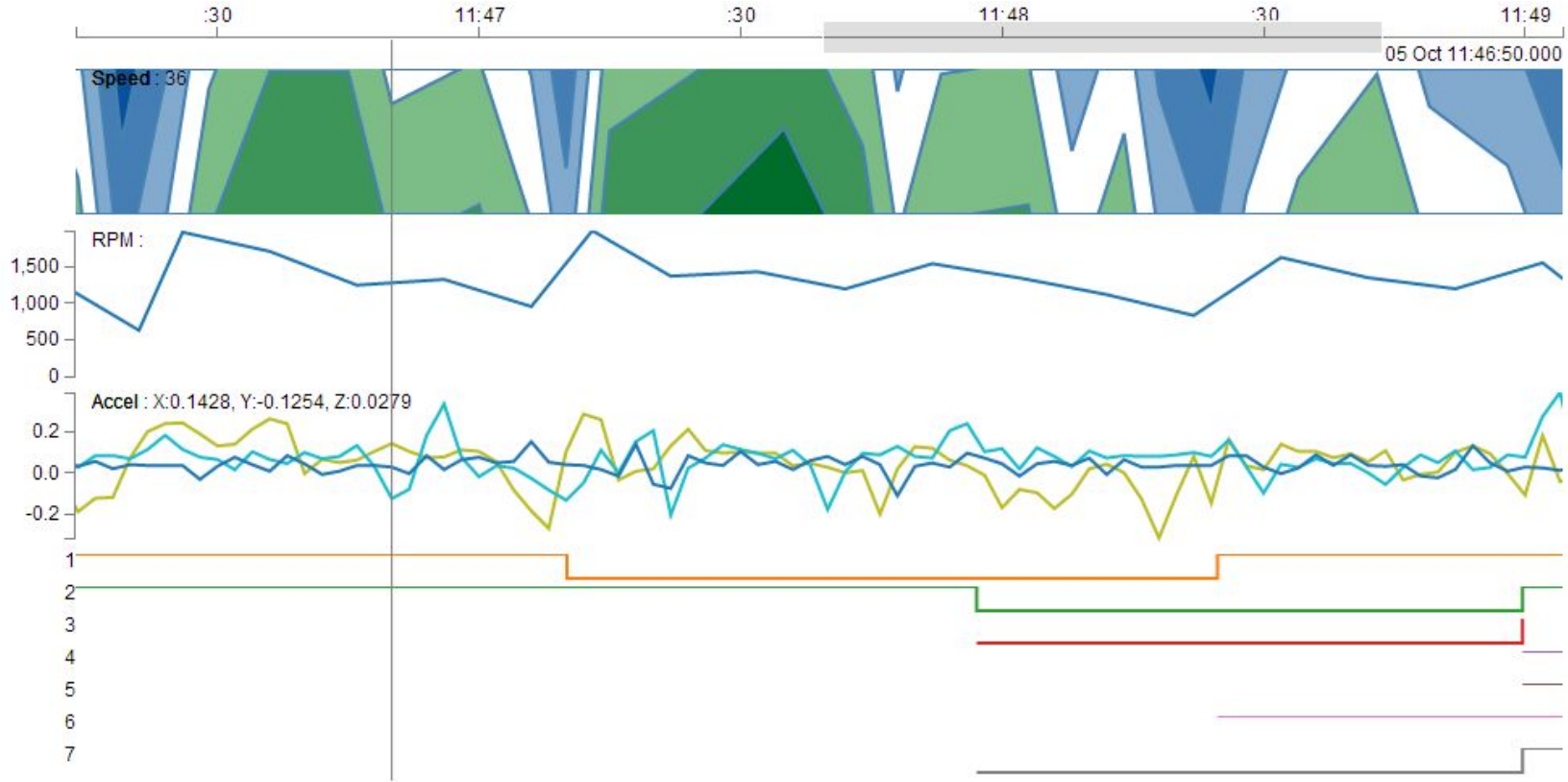




http://bl.ocks.org/wvengen/cab9b01816490edb7083

http://sankeymatic.com/

Time-series - Irregular Intervals



http://bl.ocks.org/marufbd/7191340

R Visualization Resources

- http://www.r-bloggers.com/ - A number of blog / tutorial articles about different tasks and libs
- http://tutorials.iq.harvard.edu/R/Rgraphics/Rgraphics.html - tutorial on R based visualizations
- http://www.r-bloggers.com/a-small-introduction-to-the-rocr-package/ - Tutorial on ROC package
- https://www.rstudio.com/resources/cheatsheets/ - PDF Cheatsheet I handed out on ggplot2
- https://www.rstudio.com/online-learning/ - Information on on-line resources for learning R and DS
- http://data.princeton.edu/R/gettingStarted.html - Good starting R tutorial
- http://www.inside-r.org/r-doc/graphics/plot - Covers the plot package
- http://shiny.rstudio.com/gallery/ - Shows what can be done with RStudio's Shiny dashboard service
- https://www.rstudio.com/ - RStudio's main site - Community Edition R IDE - Best / most commonly used
- http://swirlstats.com/ - R package (swirl) provides guided R exercies to learn R inside R - Great Way to Learn R!!!
- https://www.kaggle.com/scripts - Posted scripts by various users - forum + code
- https://www.kaggle.com/datasets - Sample datasets to learn with
- http://www.htmlwidgets.org/ - html widgets to use with Shiny + R Scripts to make it easy to create interactive visualizations

Additional Favorite R Packages

- dplyr
- plotly - https://plot.ly/r/

Thank you for coming tonight