

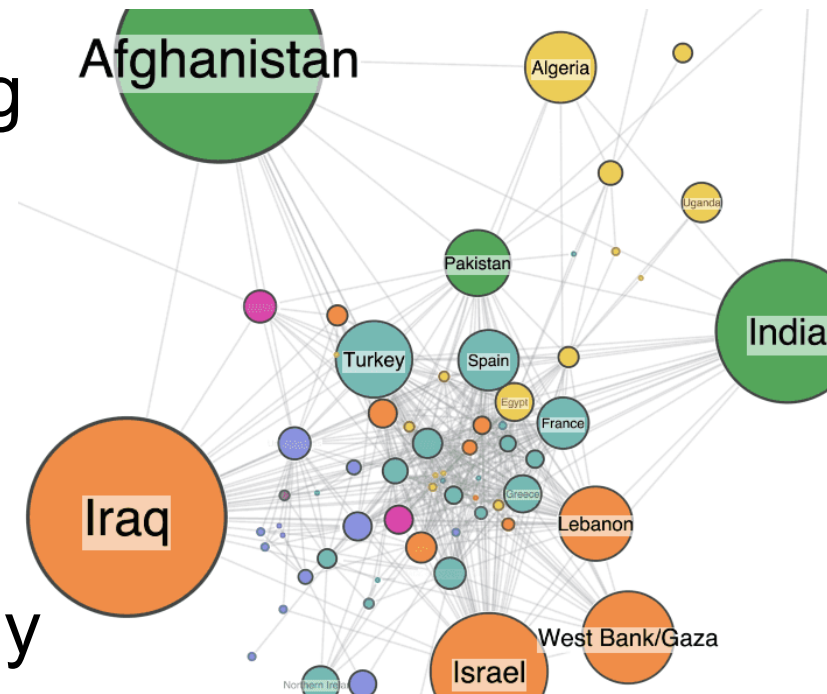
Types of Visualizations (Beyond Numeric/Statistical)

John Keyser

CONNECTION AND FLOW

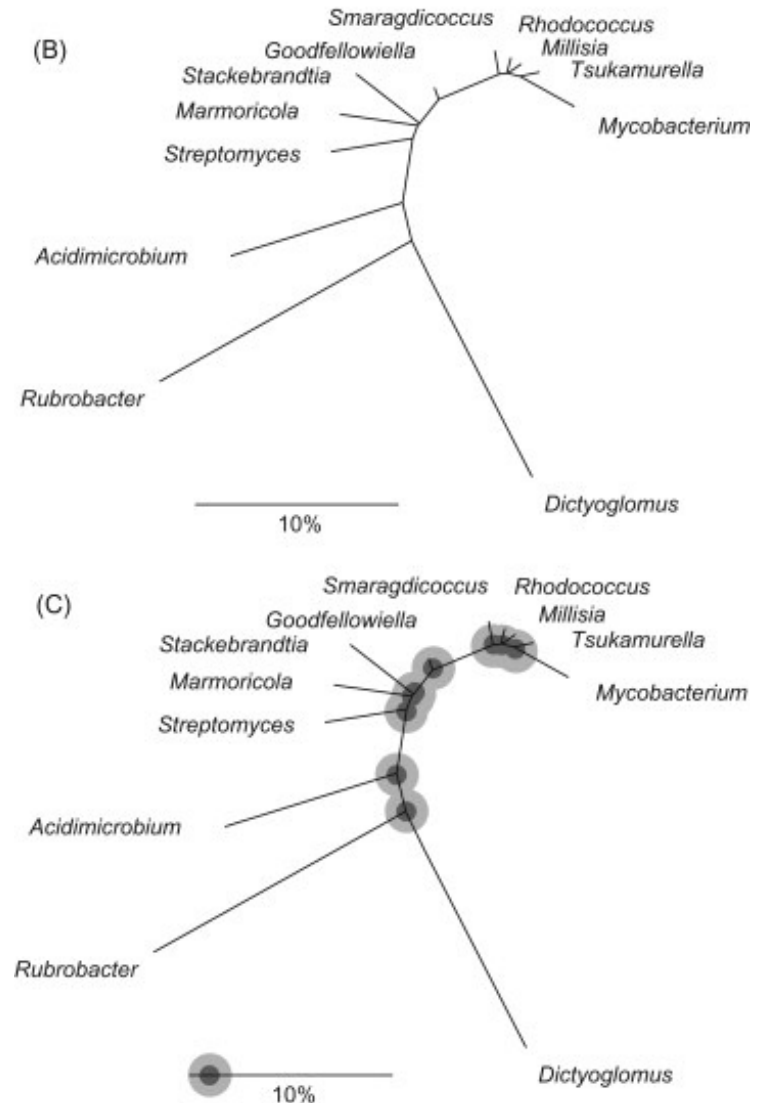
Graph Drawing

- Graph Drawing is a whole research area of its own
 - Often just understanding the adjacency relationships is the goal
 - The way to lay out and draw the nodes, edges, etc. can vary significantly



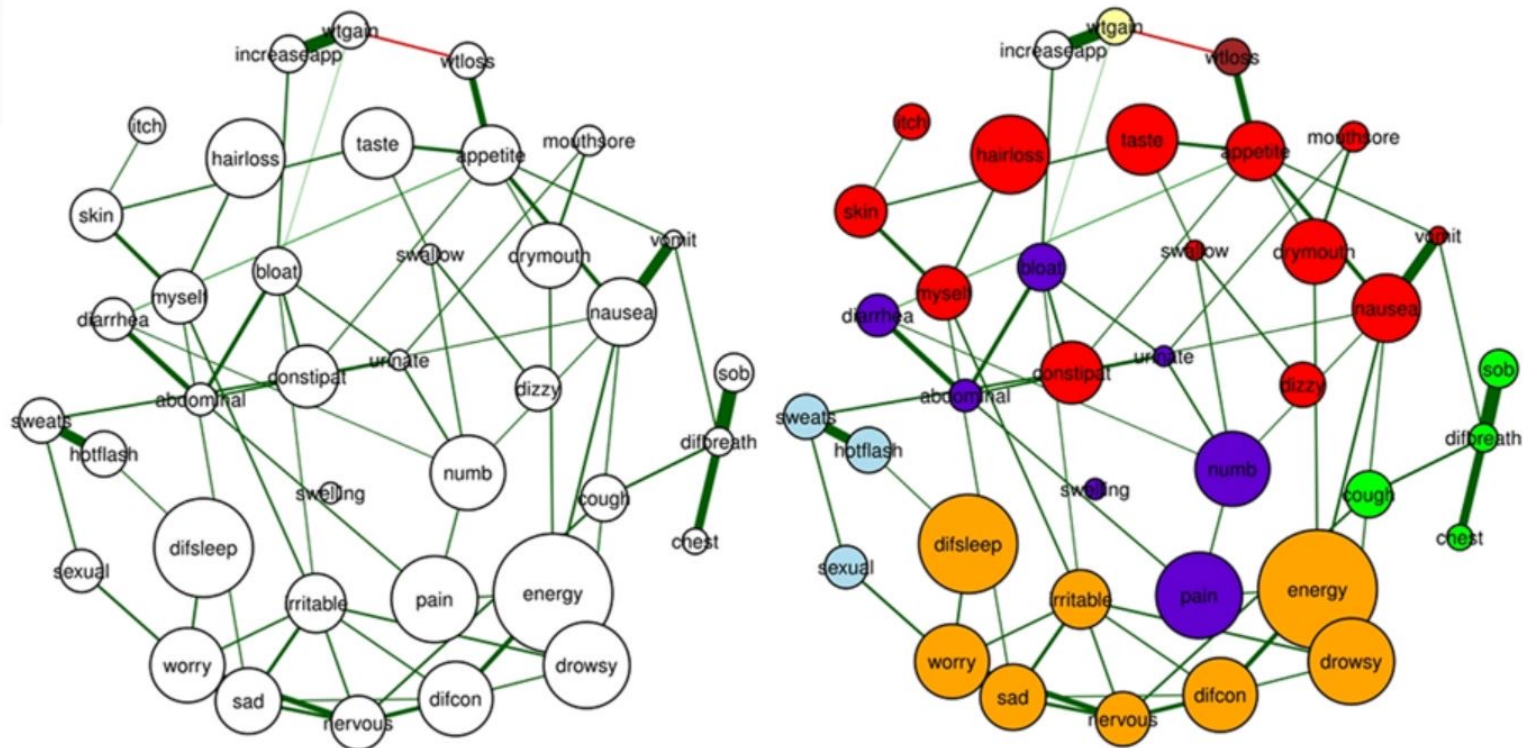
Node-link Layouts

- The direction and length of the edges can be used to provide information



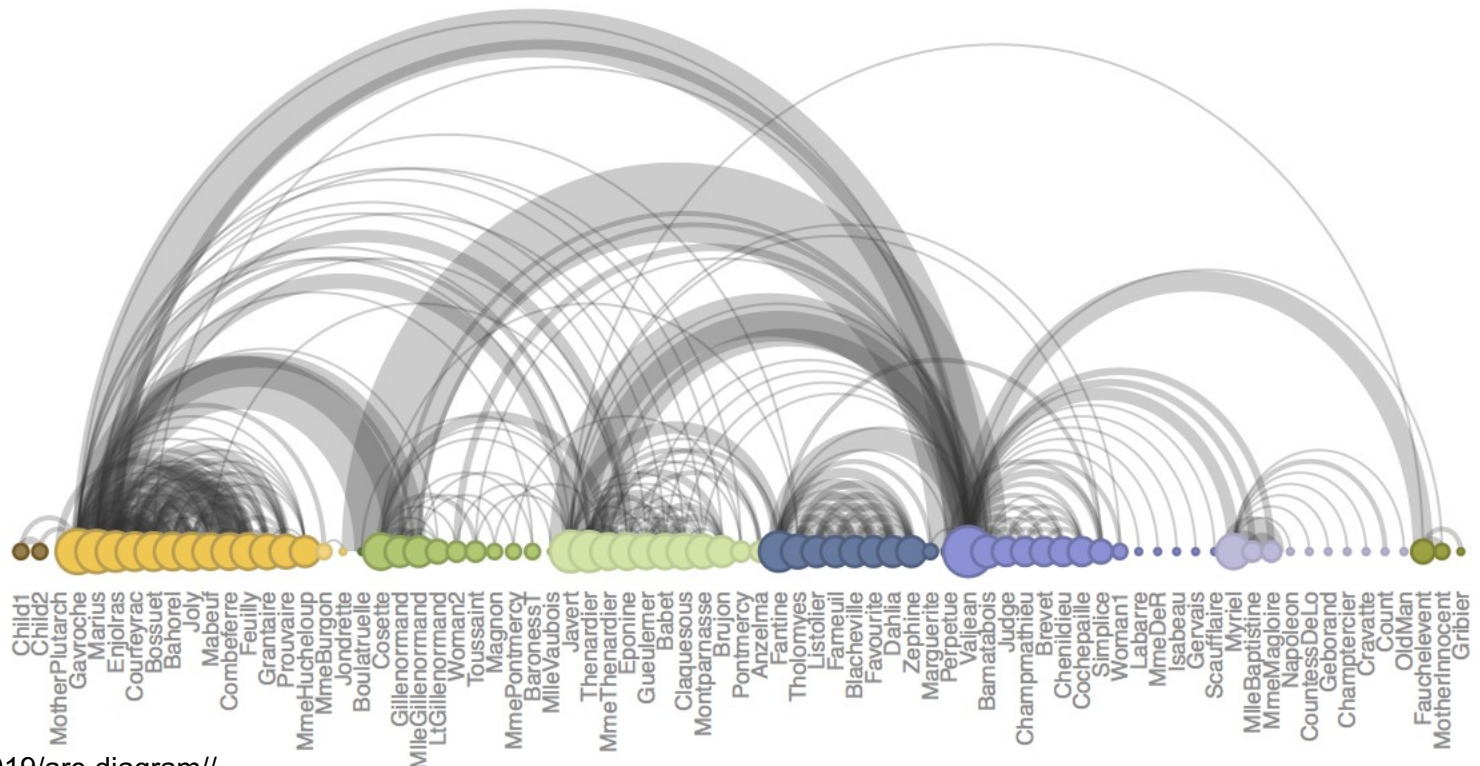
Node-link Enhancements

- Can use color, size to provide information at nodes and/or edges



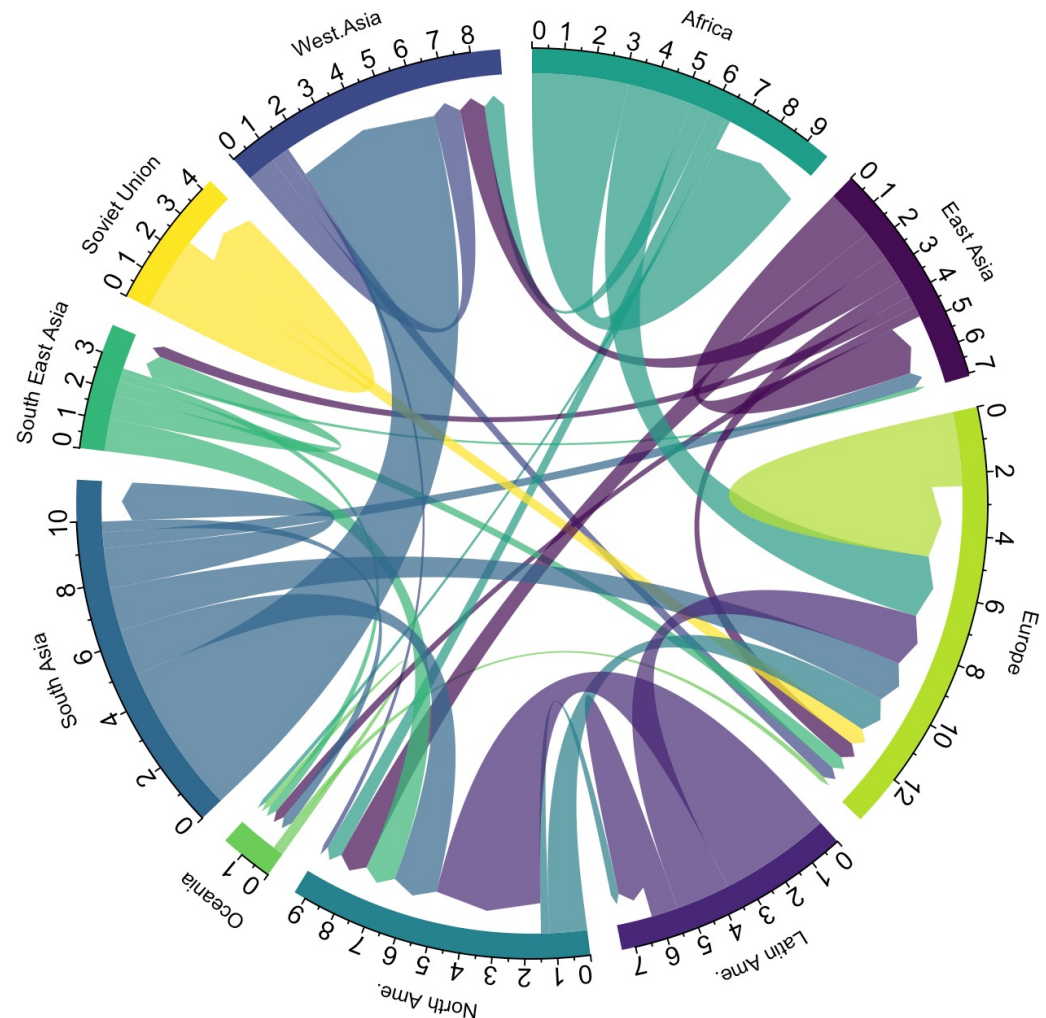
Arc Diagram

- Vertices are along a line
- Edges are semicircle or elliptical arcs



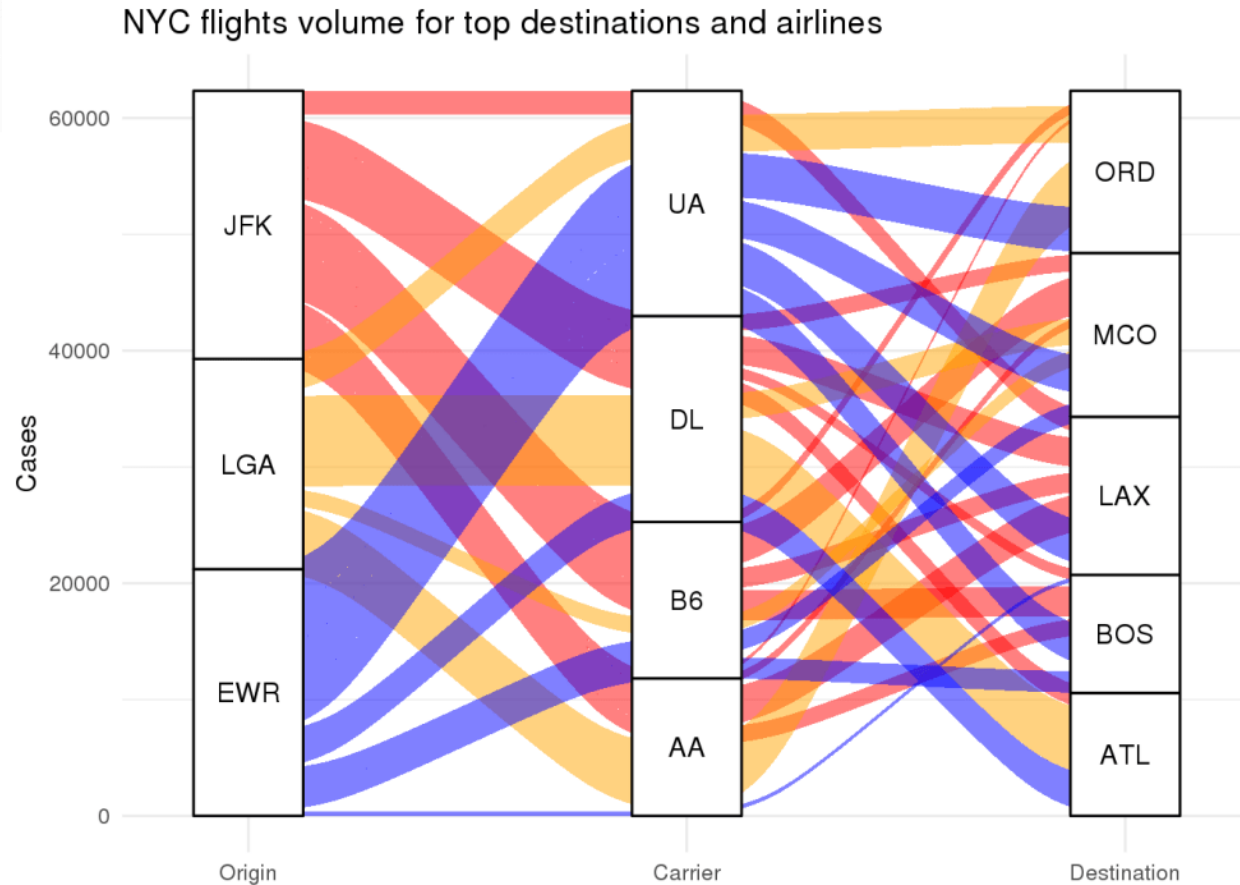
Chord Diagram

- Entities arranged around circle
- Arcs drawn between entities
 - Shows flow or connection between pairs
 - One asymmetric arc
 - Two directed arcs
 - Larger is more important
 - Can be directed or show net flow
- Usually needs explanation for understanding



Sankey Diagram

- Shows flow or connections for different stages, times, or categories
- Width of band shows proportion moving from one level to next



HIERARCHICAL VISUALIZATIONS

Hierarchies

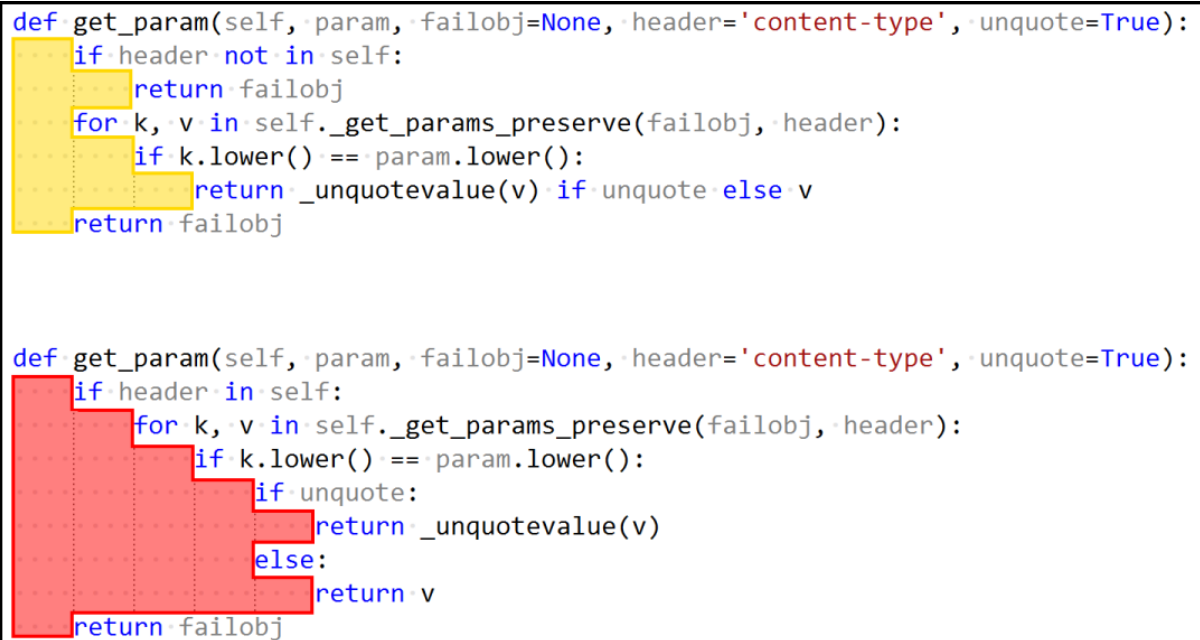
- Hierarchical classifications are common in data representations
 - Trees
 - Organizational charts
 - Time (Year/Month/Day/Hour/Minute/Second)
 - Directory structure
 - Presentations/writing
 - Etc.
- Key aspect: each item can be represented as a subcomponent of a single “parent” item

Using Hierarchies in Visualizations

- Viewer should be able to understand data, and how it relates
- Hierarchy should be used to organize and display data so that the groupings are clear
 - The display of the hierarchy itself can also be used to capture a variable
 - Or, the hierarchy can just be used to clarify relationships

Indentation

- One of the simplest ways to demonstrate hierarchical relationship
 - Indentation level shows generation, most recent item of lesser level is parent
 - e.g. code in Python

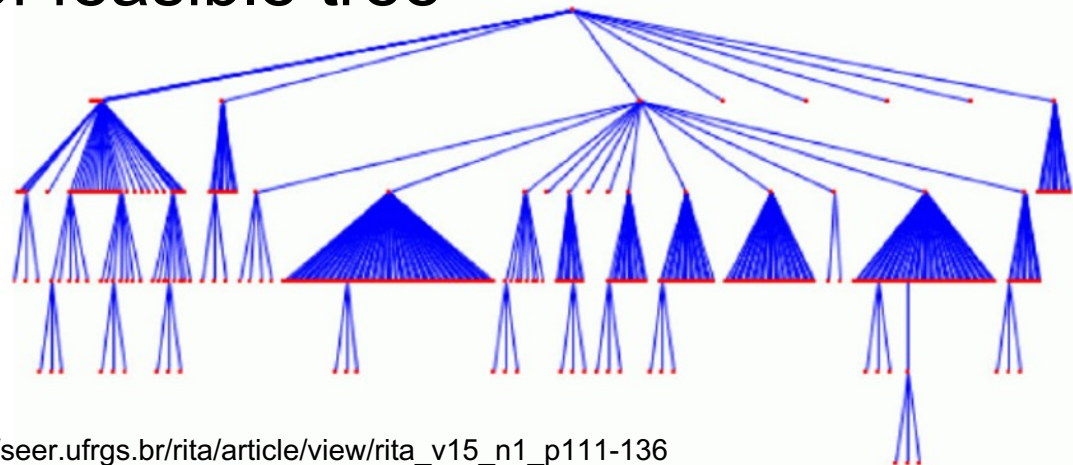
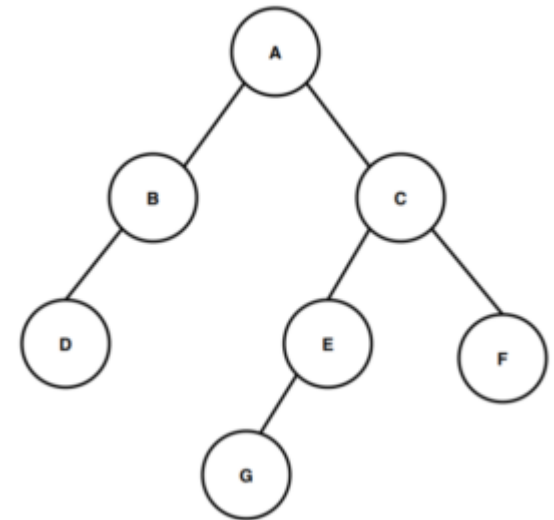


```
def get_param(self, param, failobj=None, header='content-type', unquote=True):
    if header not in self:
        return failobj
    for k, v in self._get_params_preserve(failobj, header):
        if k.lower() == param.lower():
            return _unquotevalue(v) if unquote else v
    return failobj

def get_param(self, param, failobj=None, header='content-type', unquote=True):
    if header in self:
        for k, v in self._get_params_preserve(failobj, header):
            if k.lower() == param.lower():
                if unquote:
                    return _unquotevalue(v)
                else:
                    return v
    return failobj
```


Node-link layout

- The most basic approach – like traditional tree diagrams in Computer Science
- Root at top (or left, bottom)
- Each generation at same height
- No crossing lines
- Limitations to size of feasible tree



Node-link Layouts

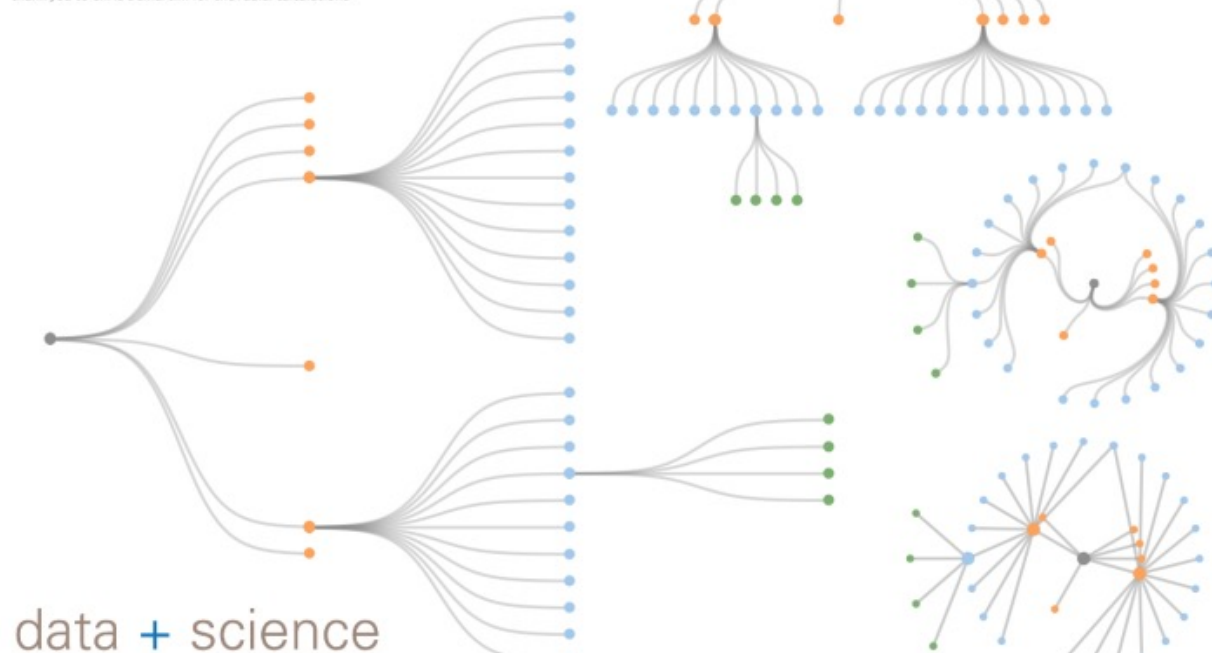
- Tree does not have to be laid out in hierarchy
- Placement of root, direction of nodes, edge shape, etc. can change

Node-Link Tree Diagram in Tableau 2.0

created by: Jeffrey A. Shaffer | DataPlusScience.com

created 1/1/2015 and updated 1/23/2019

thank you to Chris DeMartini for the radial calculations



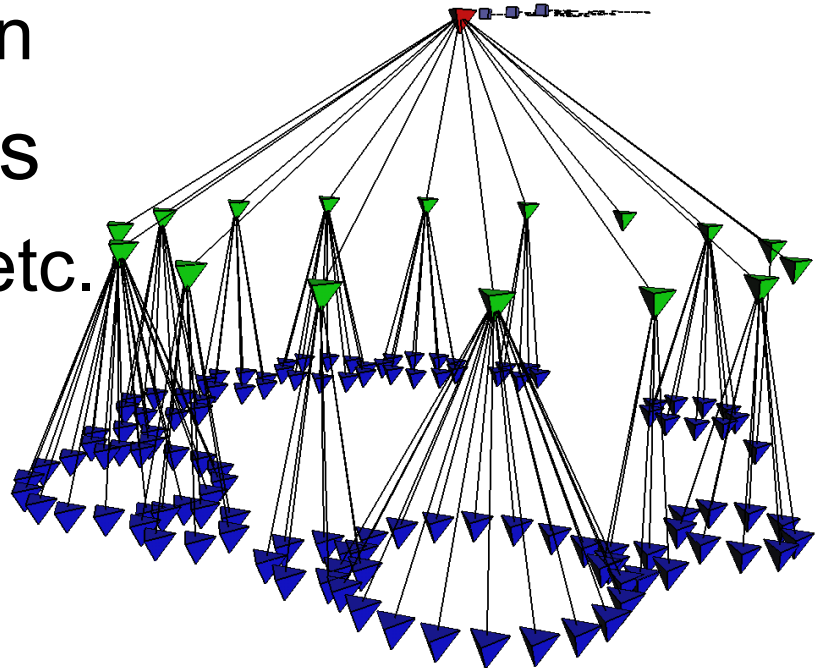
Reingold-Tilford “Tidy” Trees

- An algorithm to produce “nice” layouts of trees in traditional sense
 - Compact layout
 - Isomorphic subtrees are drawn the same way
 - Preserves ordering, symmetry
 - Idea has been adapted



Cone Trees

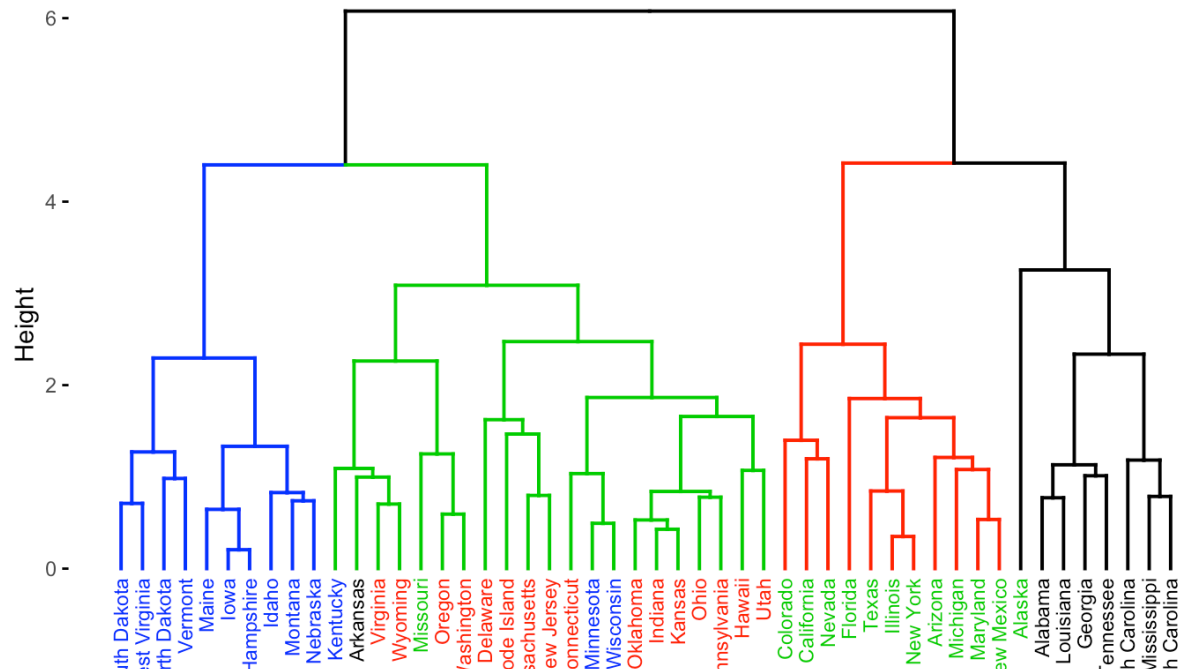
- Can put trees in 3D rather than 2D
 - Allows greater spread, less clutter
 - Also common to place in hyperbolic space, and draw projection
- 3D can have issues
 - Depth, occlusion, etc.



Dendrogram

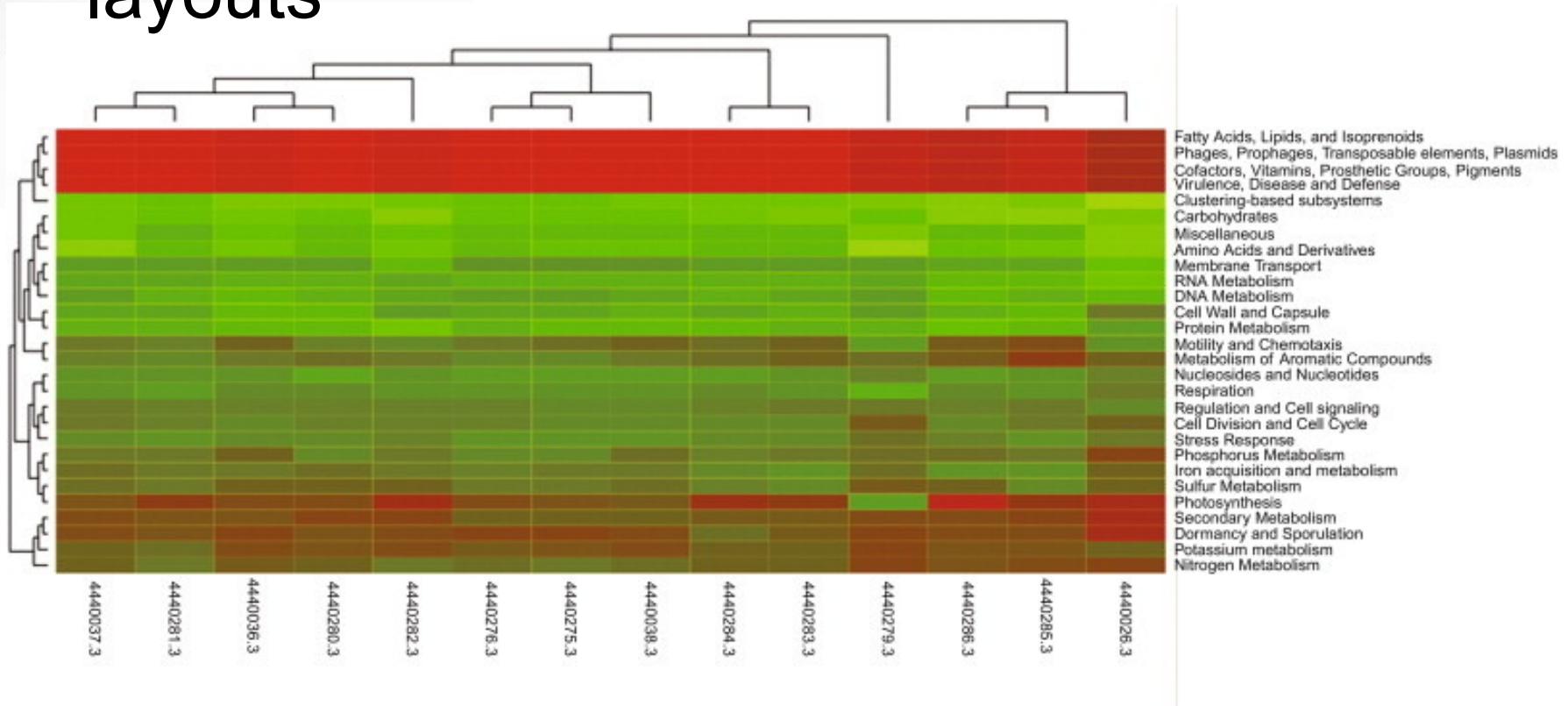
- Dendrograms use orthogonal lines
- All leaves are at the same level

Cluster Dendrogram



Dendrograms With Heat Maps

- Dendrograms combine easily with grid layouts



Icicle Trees

- Each generation of tree is a new level
- All levels are same width
- Children implied by lining up underneath parent
- Size can encode a variable (e.g. amount)

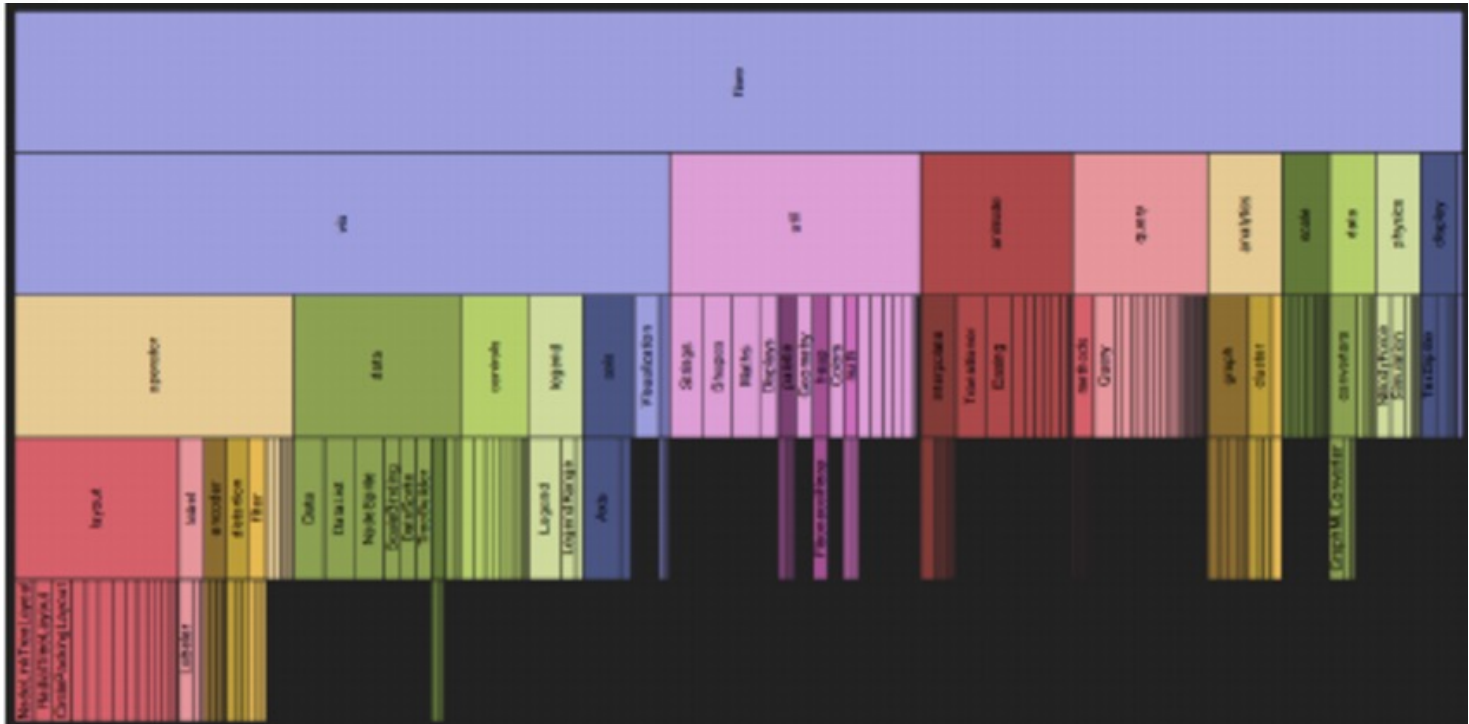
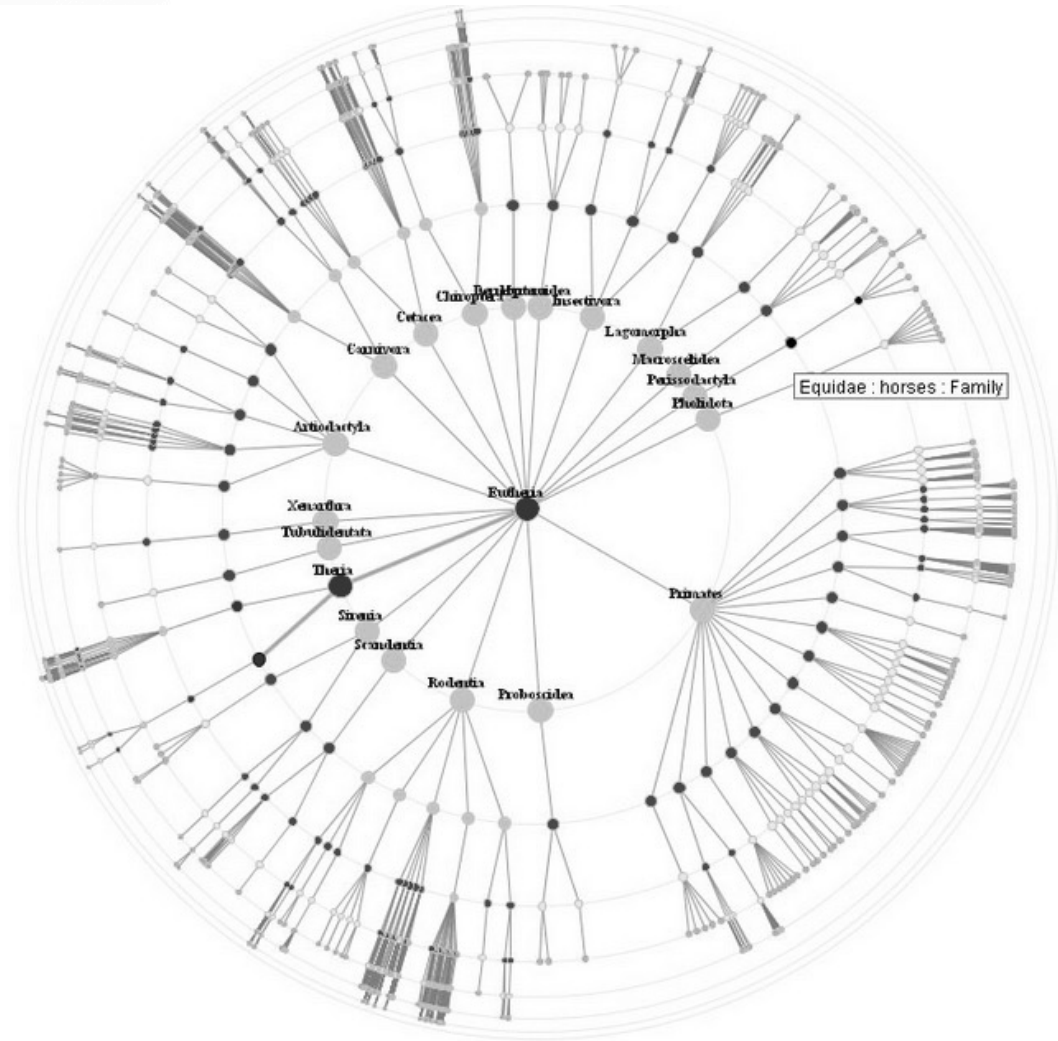


Image from: 
<https://www.cs.umd.edu/~ben/papers/Wongsuphasawat2011LifeFlow.pdf>

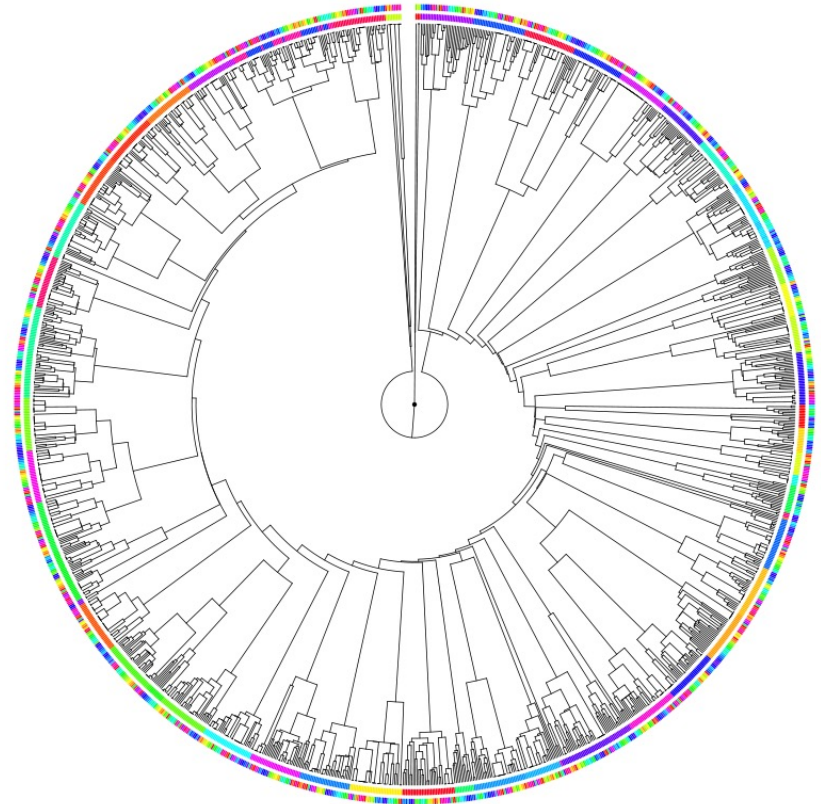
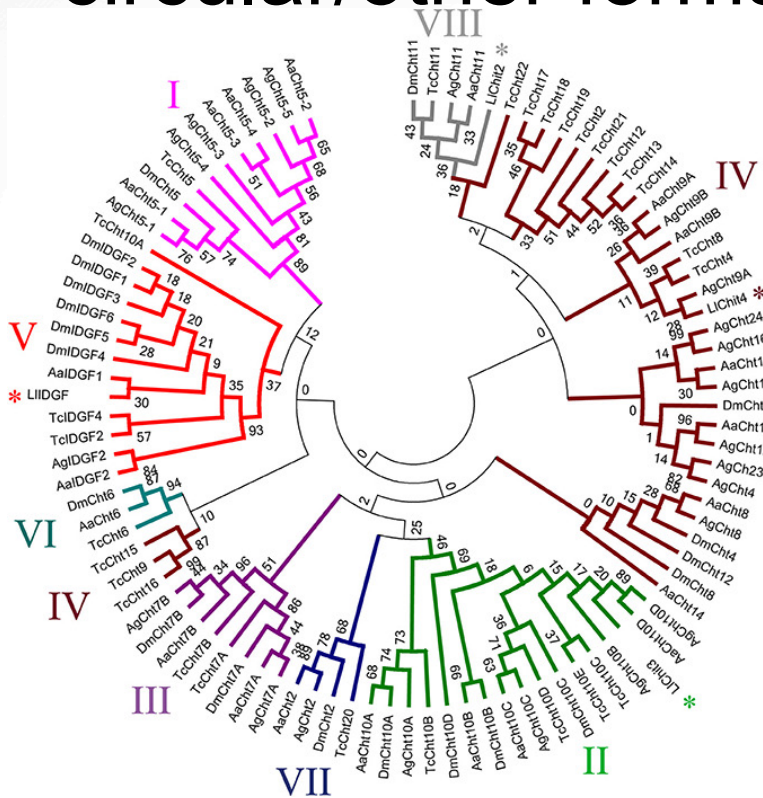
Radial Subdivision

- Many of the linear forms we've seen can be wrapped around a circle to create a radial version.

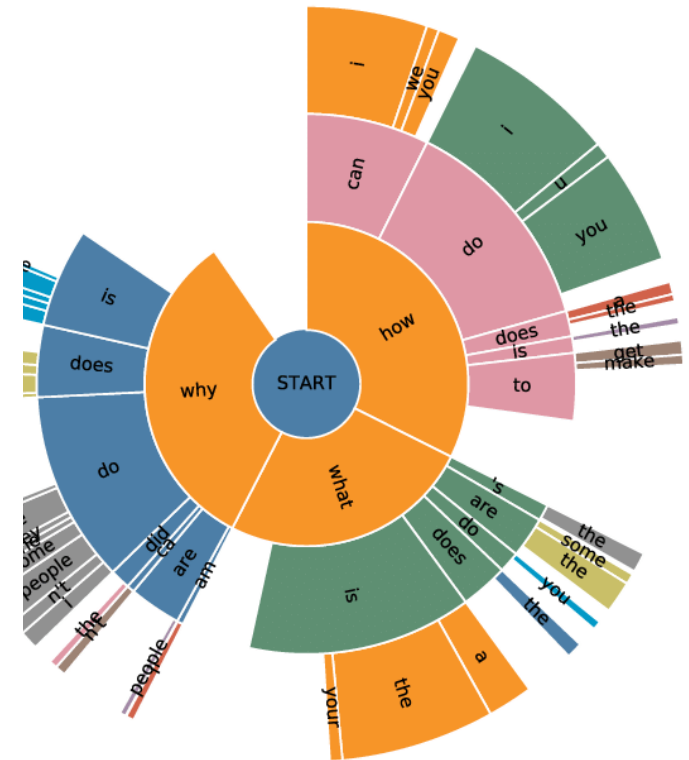


Radial Dendrogram Layouts

- Dendrograms can be laid out in linear or in circular/other forms



“Sunburst” – Radial Icicle Tree



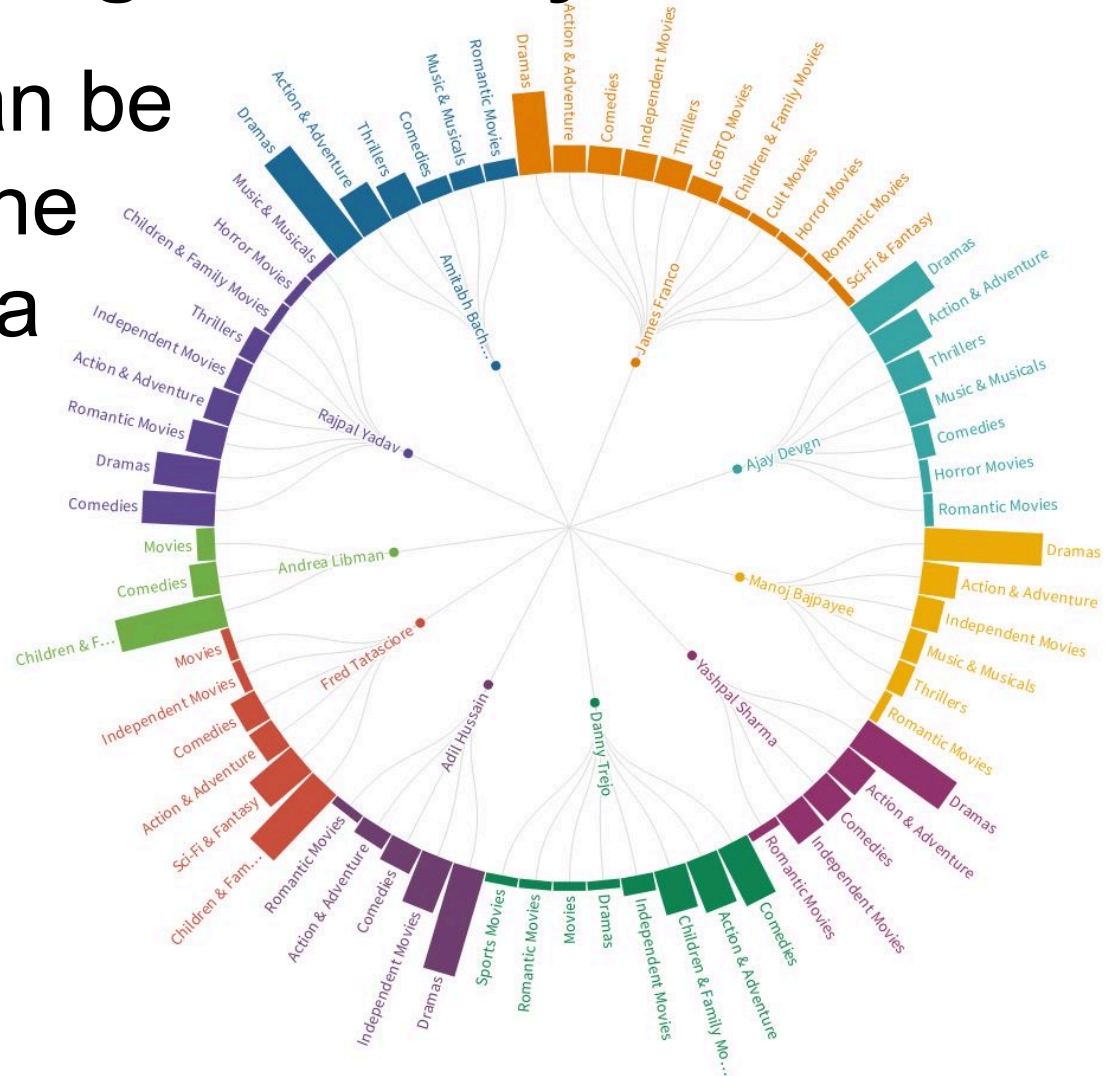
Images from:

<https://www.syncfusion.com/jquery/aspnet-web-forms-ui-controls/sunburst-chart> and

https://link.springer.com/chapter/10.1007/978-3-030-45442-5_21

Enhancing Radial Layouts

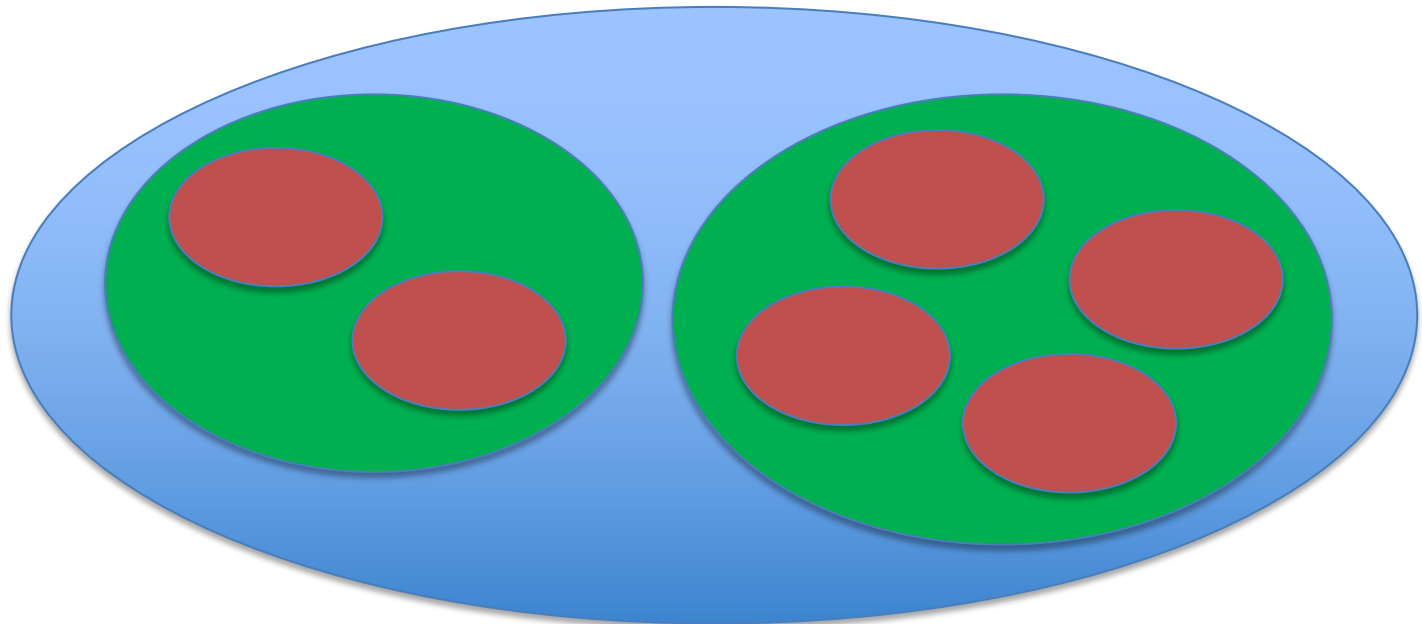
- Information can be added along the outer edge of a radial layout



CONTAINMENT

Hierarchy By Containment

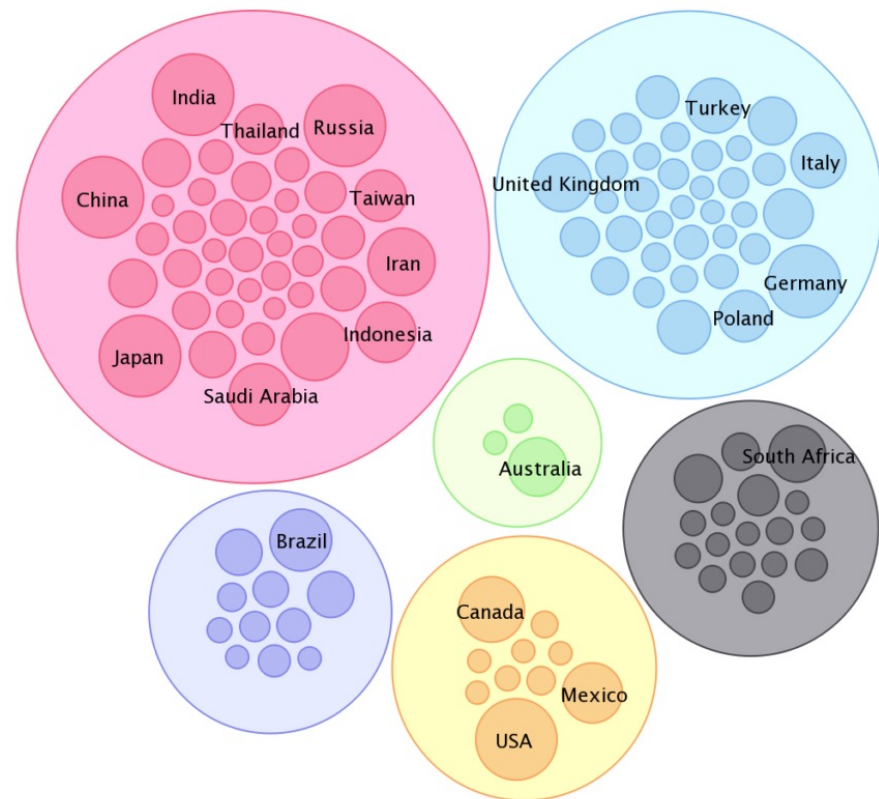
- Basic Idea: we will show hierarchy by containment
 - Sort of like a Venn Diagram



Circle Packing

Carbon emissions around the world (2014)

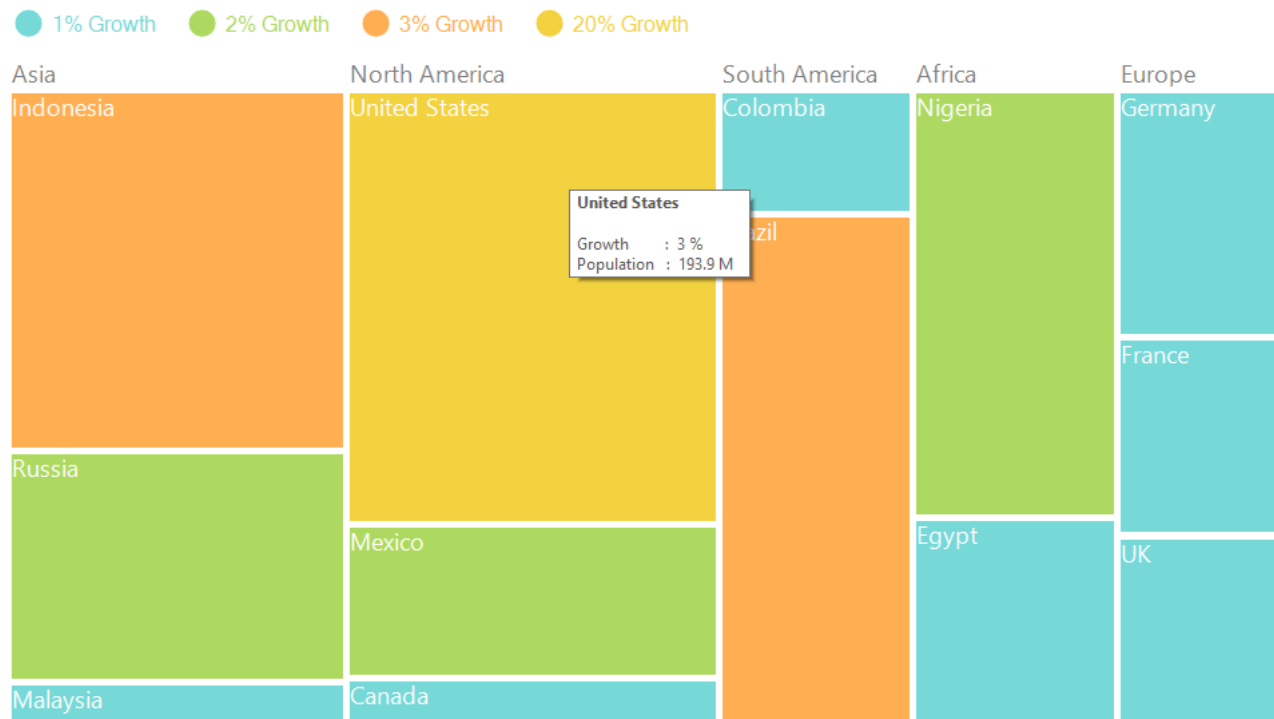
- Idea is to use a Venn-like approach to show hierarchy
- Circle size can show values
 - Only comparable at same generation



Europe Africa Oceania North America
South America Asia

Tree Maps

- Divide a rectangular region by hierarchy
 - Alternate horizontal/vertical divisions by generation
- Area is used to show a value
- Color often used to show second value (dependent variables)



Tree Maps and Hierarchy

- The hierarchy is sometimes difficult to determine
 - The dividing lines are often not obvious
 - Usually not the best choice if hierarchy is critical to understand
- Several algorithms for optimizing treemap layout when there are many entities at one level
 - Clustering, squarification, etc.

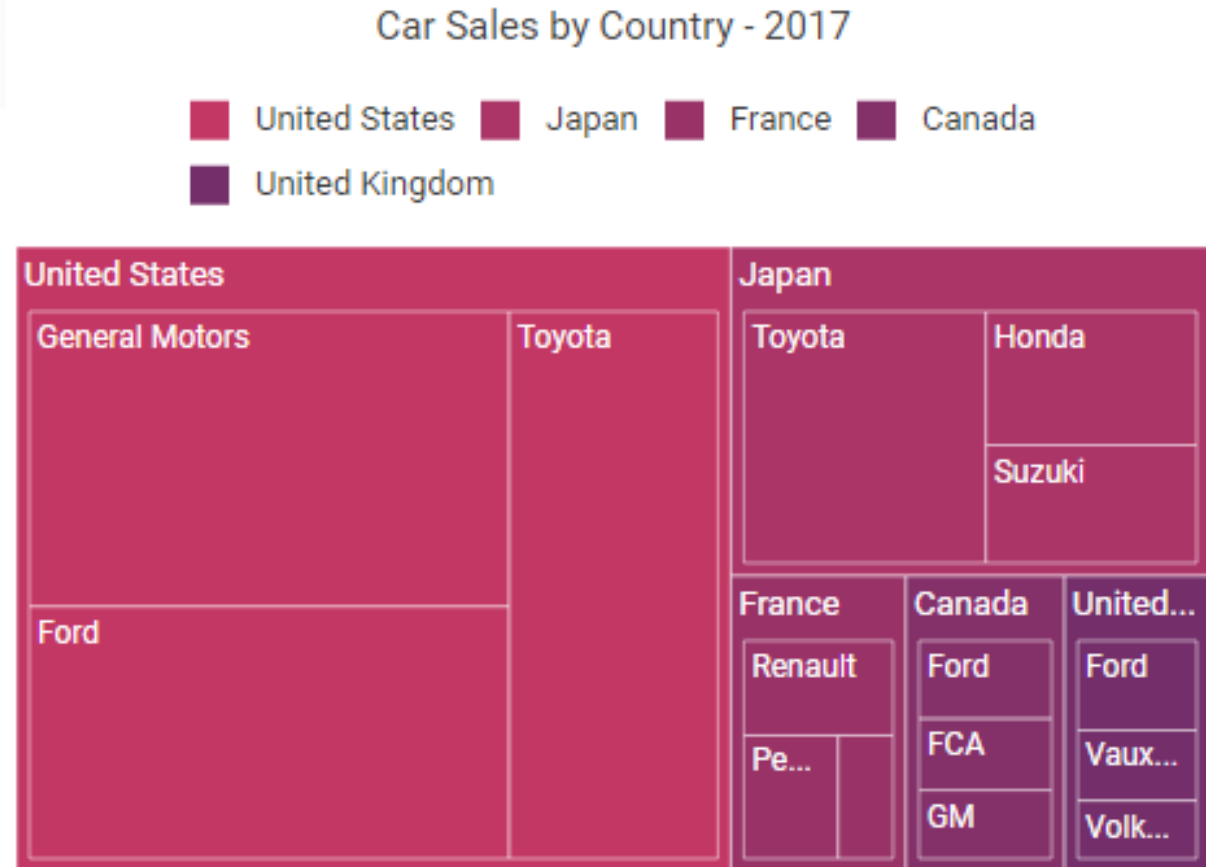


Image from:

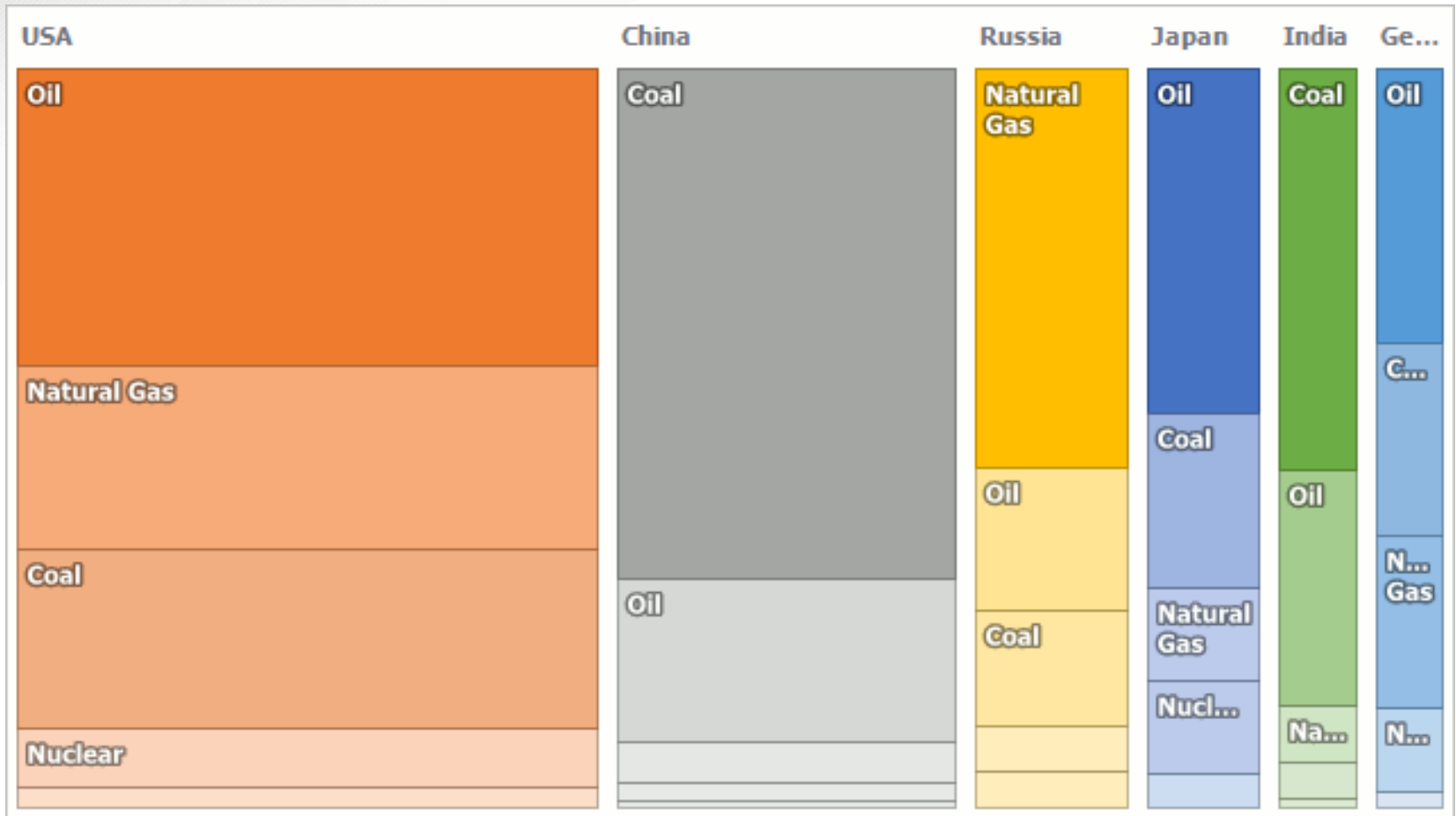
<https://insightwhale.medium.com/how-to-show-hierarchy-with-data-visualization-526fb45ee4c2>

Tree Maps - Containment

- Hierarchy can be made explicit by labeling containment
- But, area is no longer comparable at all levels



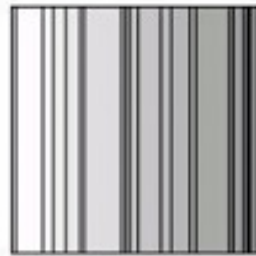
Tree Maps - Containment



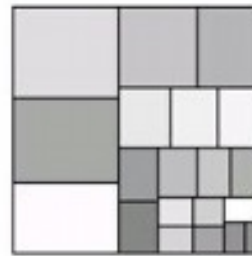
Tree Maps and Ordering

- Tree maps can be used without hierarchy or containment
 - One gets assigned to provide “nice” layout
 - Techniques can be used to modify size to make layout more appealing
 - The ordering may be adjusted for layout
 - Useful when there are many items at one level

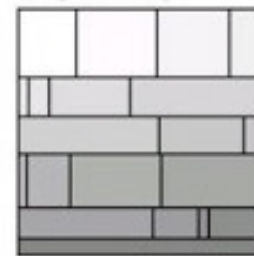
SliceAndDice



Squarified



Strip



order	ordered	unordered	ordered
aspect ratios	very high	lowest	medium
stability	stable	medium	medium

Tree Maps Advantages

- Can display large amounts of data, multiple variables, with little waste



Still to Come

- We will discuss maps separately next time
 - As well as taxonomies of visualizations
- Class reminders:
 - No class/office hours Monday & Tuesday (Fall break)
 - Due today: HW 2 and Survey 5
 - Paper selection: 1 week from today
 - Step 2: 2 weeks from today
 - Coming up: Survey 6