Tree Visualization Techniques

Group 6

Neeraj Kumar Kondaparthi

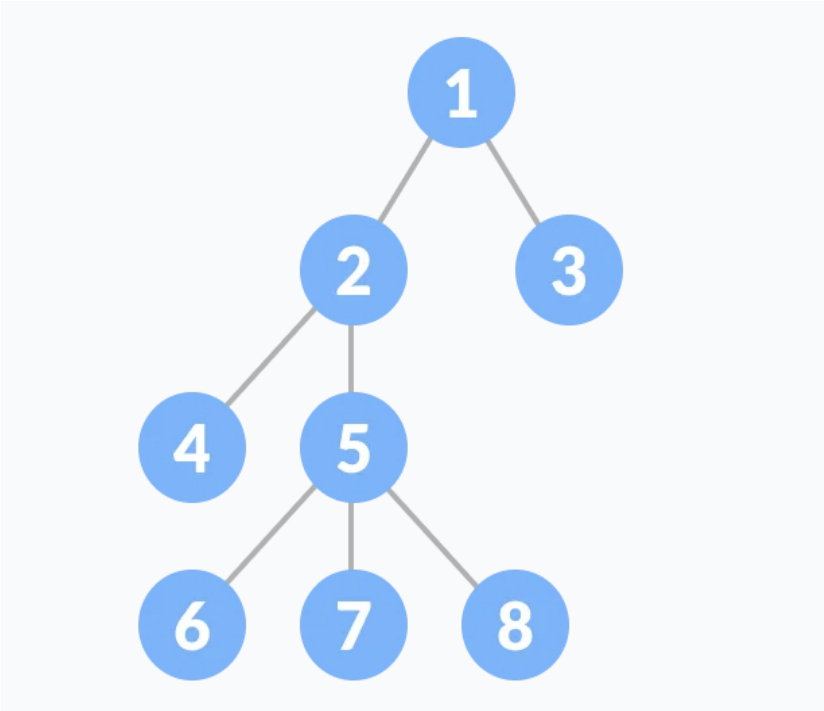
Tushar Nayan

Yash Mundra

Srimant Kumar Mohanty

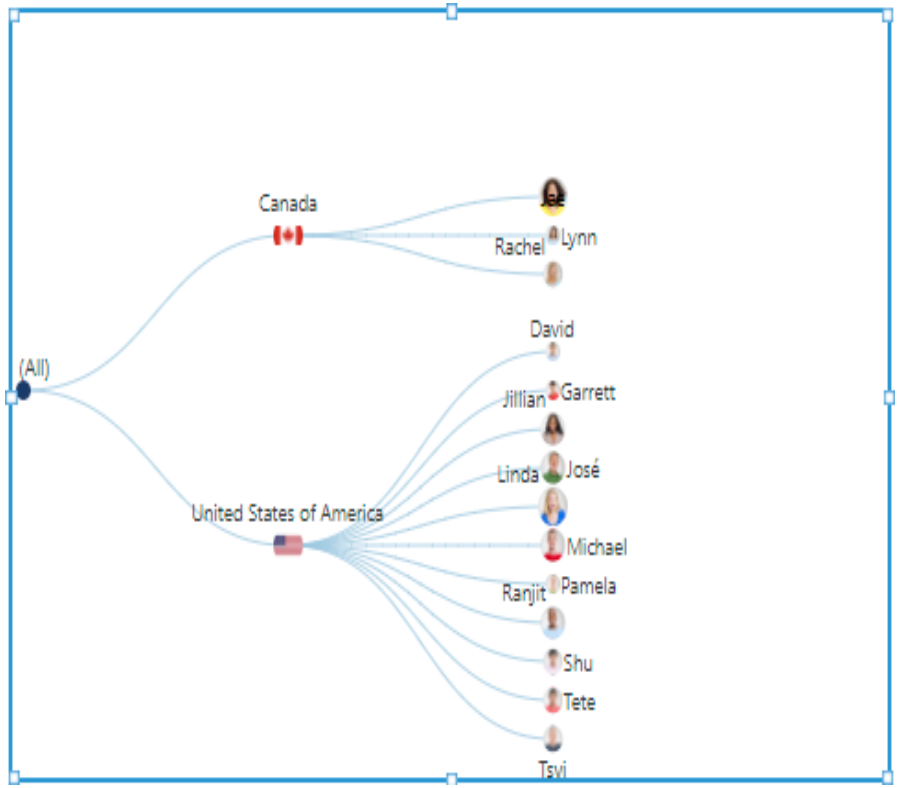
Tree

- A tree is a nonlinear hierarchical data structure that consists of nodes connected by edges.
- Some of the Applications of Tree data structure are File system , Database Indexing ,Compiler Design



Tree Visualization

- Tree visualization (sometimes called hierarchy visualization) is a branch of information visualization dedicated to the graphical representation of Trees.



Tree visualizations - Problem

- The plethora of tree visualization techniques poses challenges to researchers and developers.
- Researchers, especially those new to the field, have no way of knowing every tree visualization that has been published, even over just the last two decades.
- Results in the reinvention of existing visualization techniques .

Treevis.net- Solution

- To address the challenges and exploit the opportunities, existing tree visualization techniques are collected and formed into a reference for them that’s as complete as possible.

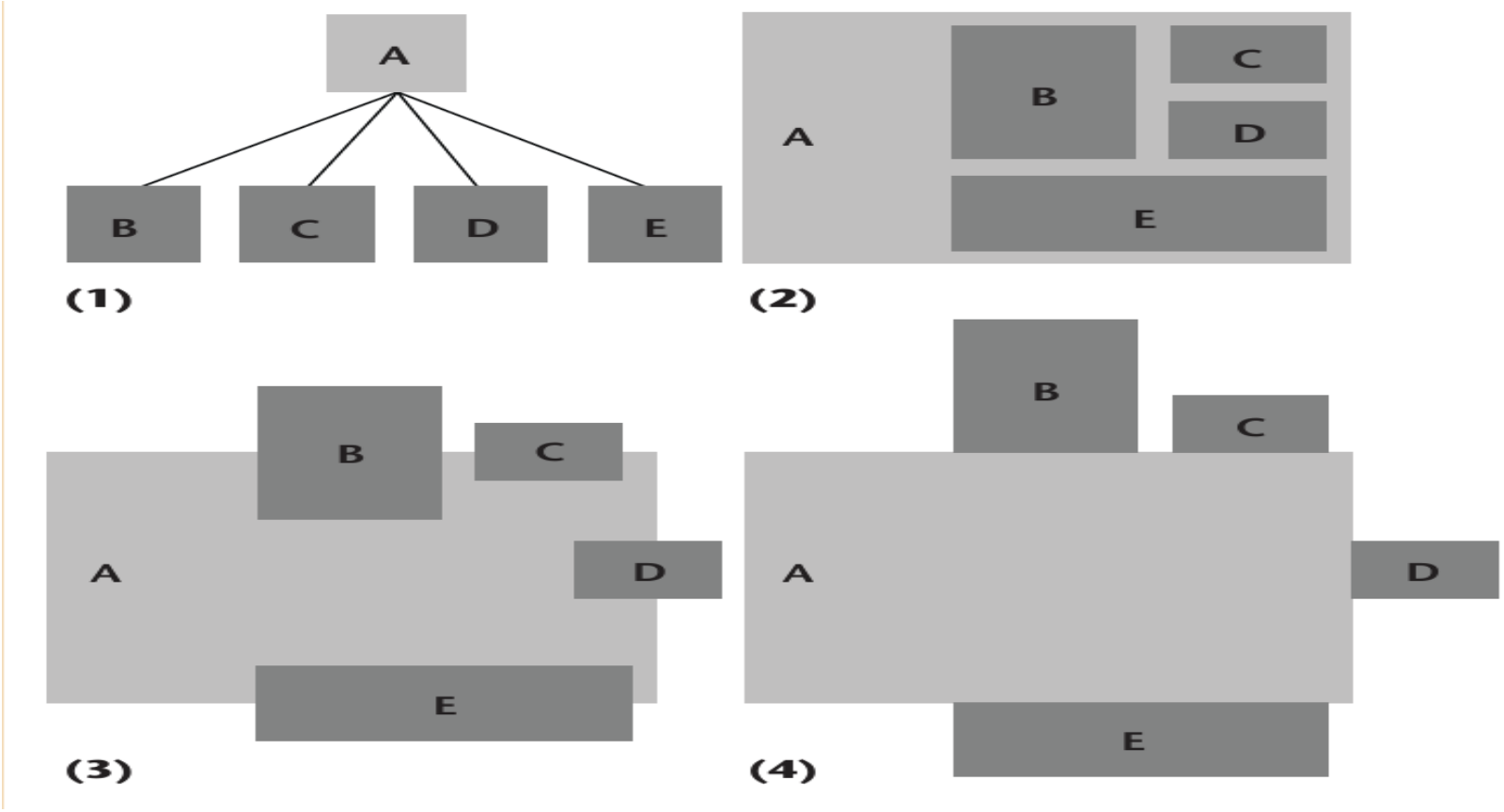
- This is where the treevis.net project comes into the picture.

Treevis.net

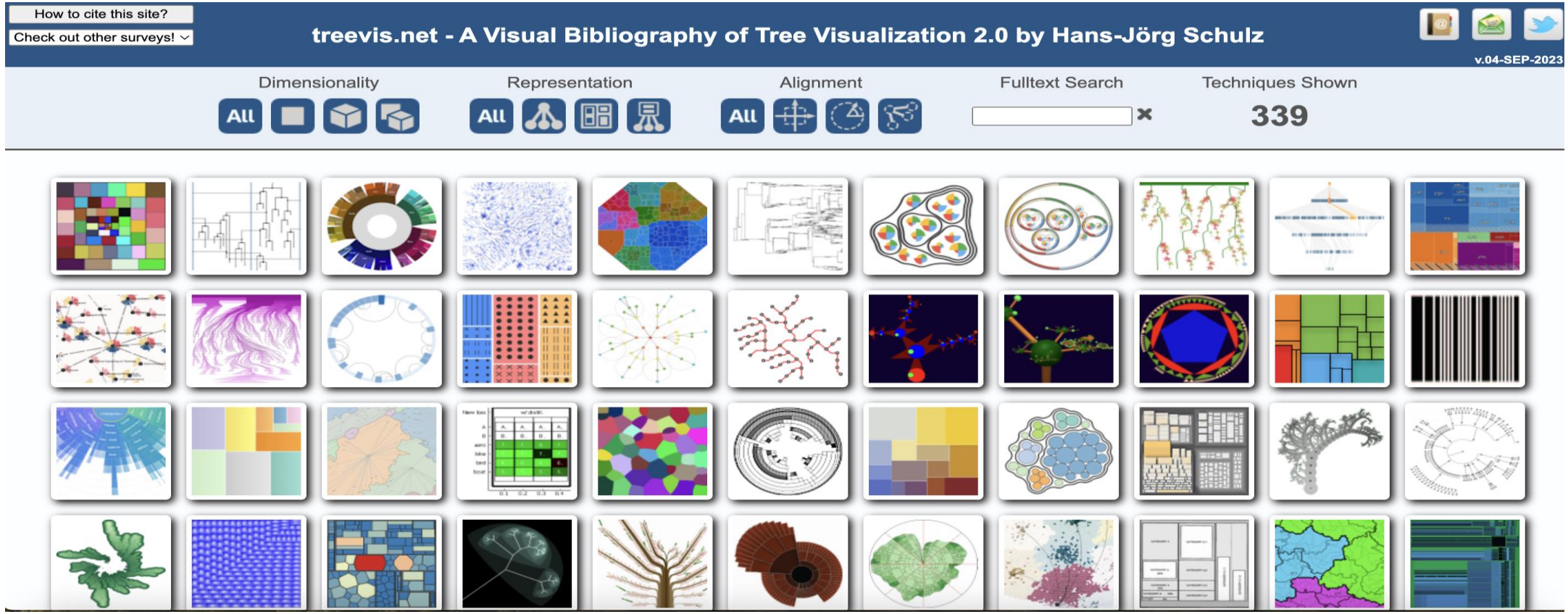
To organize the visualization techniques in treevis.net , three design axes are considered

1. Dimensionality (2D, 3D, or hybrid),
2. Edge representation (explicit, implicit, or hybrid)
3. Node alignment (radial, axis-parallel, or free).

Edge Representation



Treevis.net - https://treevis.net/



Treevis.net - Usage

- The first type of searching is for a known technique to find out who the authors are or when it was first published. Researchers perform such searches .

- The second type of searching is for a technique with a look and feel similar to a given one. Both researchers and developers perform this kind of search.

- The third type of searching is for a suitable technique for a given dataset or application, as developers often do, possibly with their clients or customers.

Treevis.net - Future Work

- Improvements to the search capabilities
- Add more intuitive full-text searching with wildcards
- layout of visualization thumbnails

Tree Visualization

- Tree Visualization is a graphical representation of hierarchical data.
- Data points - parent-child relationships
- Data can be broken down into multiple levels
- Give ability to analyze, explore, and communicate complex hierarchical relationships within datasets.
- Used in fields including, computer science, decision analysis, biology and organizational management.

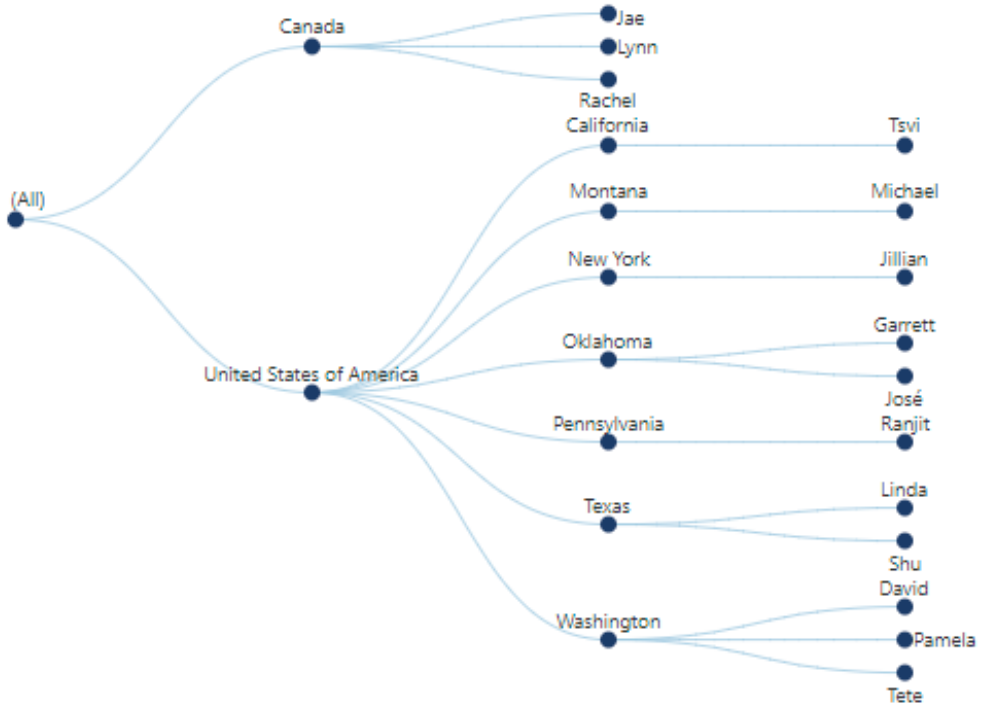


Figure T1: Visualization of hierarchical data

Source: https://towardsdatascience.com/6-hierarchical-datavisualizations-98318851c7c5

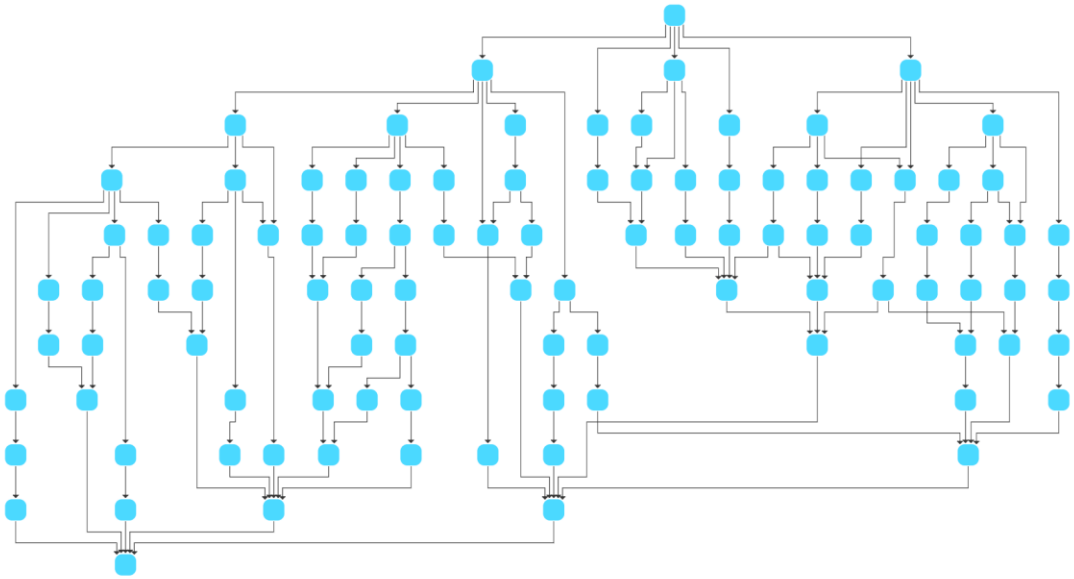
We will cover

- Traditional Trees
- Radial Trees
- Treemap

Tree Visualization Cont...

- Layering structure provides a clear visual indication of hierarchy depth
- Preferred to represent spatial relationships
  - - a map, photographs, historical information

Don’t use - if the size ratios are large the leaf nodes become hard to interpret especially in large trees.



Source: https://www.yworks.com/pages/layered-graph-layout

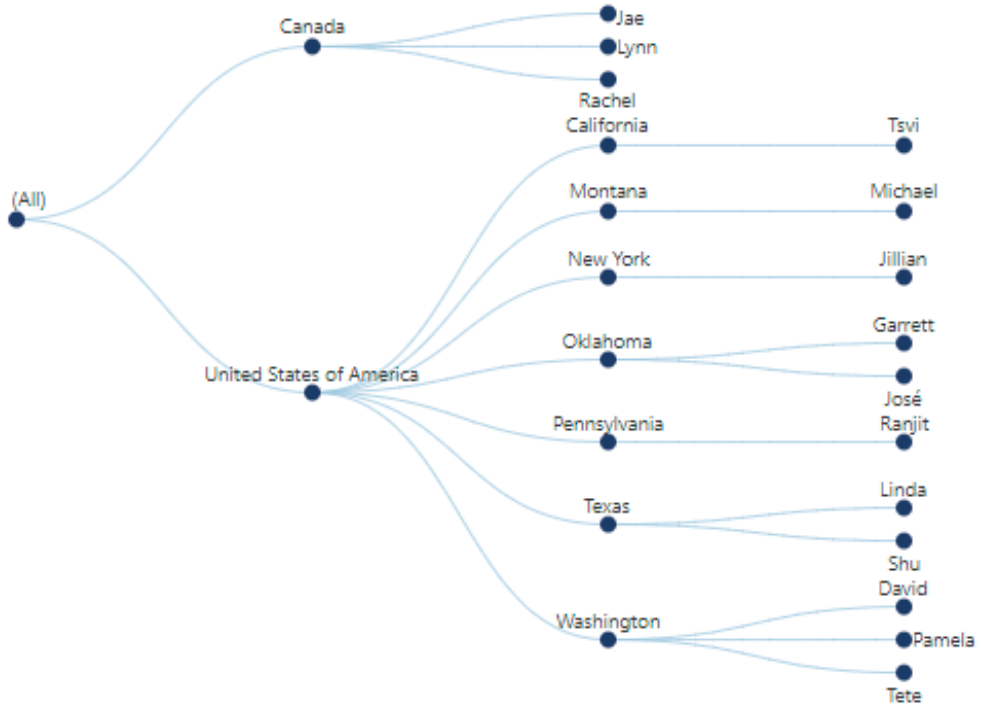
Figure T3: Layered Tree Visualization

Tree Cont...

Advantages of tree visualization techniques

- **Clear Hierarchical Representation** - within data
- **Facilitates Exploration** - explore different levels of hierarchy by expanding or collapsing branches
- **Identifying Patterns** - branching patterns and node characteristics
- **Comparative Analysis** - compare hierarchical structures or variations

But - certain issues and limitations



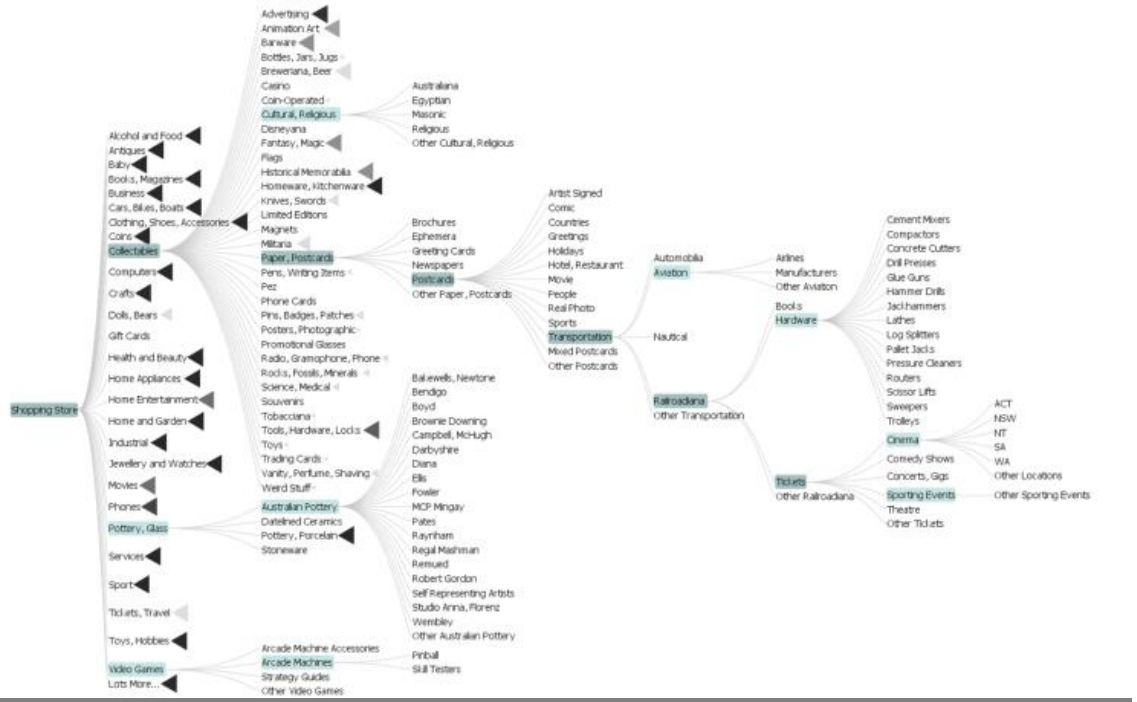
Source: https://towardsdatascience.com/6-hierarchical-datavisualizations-98318851c7c5

Figure T4: Normal Tree

Tree Cont...

Traditional Tree Visualization - certain issues and limitations

- Complexity and Comprehension (with larger dataset)
- Distortions and Precision
- Other visualizations/variation

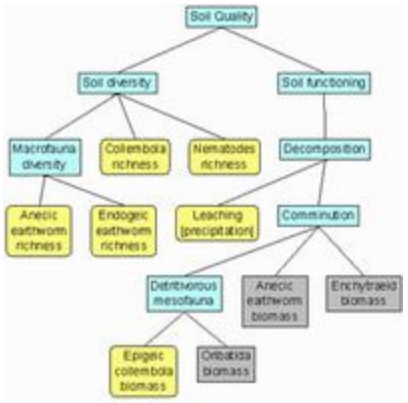


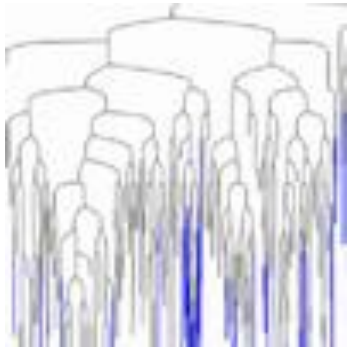
Source: https://staff.cdms.westernsydney.edu.au/~vinh/projects_TreeVis.php

Figure T5: Complex Tree Visualization

Advance Tree Visualization Techniques

- Icicle Tree
- Pansy Tree
- Control Tree
- 3D Cone Tree
- Layered Tree





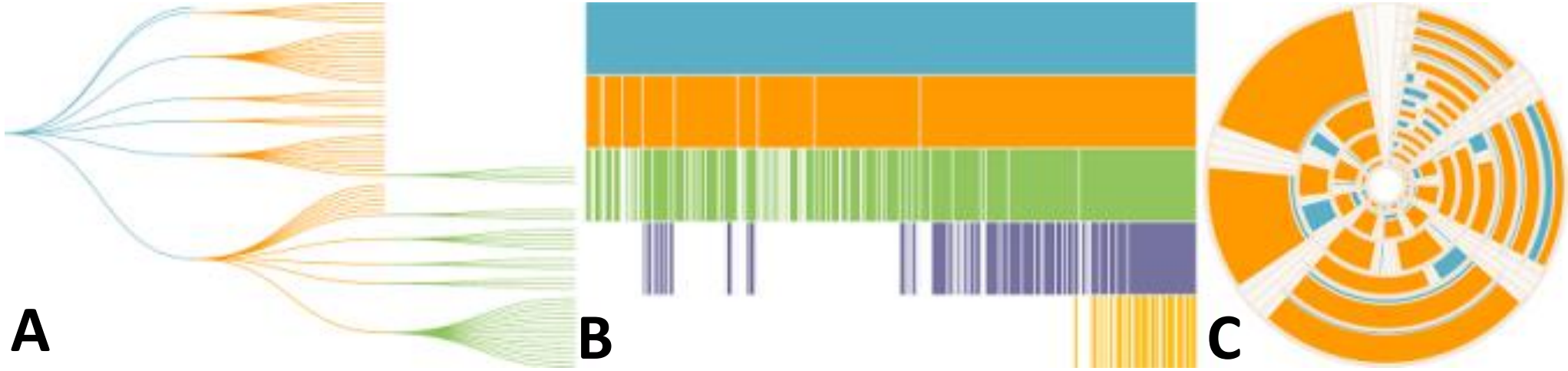
Source: https://treevis.net/#Rafelsberger2007

Figure T6: Complex Tree Visualizations Example

Icicle Tree plots

- Represents hierarchical structures in a layered or tiered format.
- Nodes are organized into either horizontal or vertical layers,
- With parent nodes positioned above their child nodes.

Which one is Icicle tree visualization?



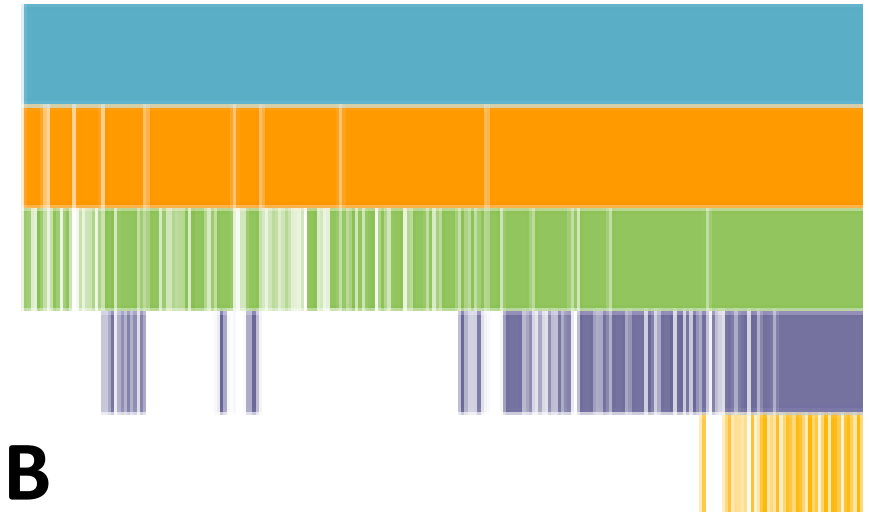
Source: https://livebook.manning.com/book/d3js-in-action-second-edition/chapter-6/v-6/8

Figure T7: Various visualizations adopted so far in industry

Icicle tree plots

- Represents hierarchical structures in a layered or tiered format.
- Nodes are organized into either horizontal or verticall layers,
- With parent nodes positioned above their child nodes.

Yah!!



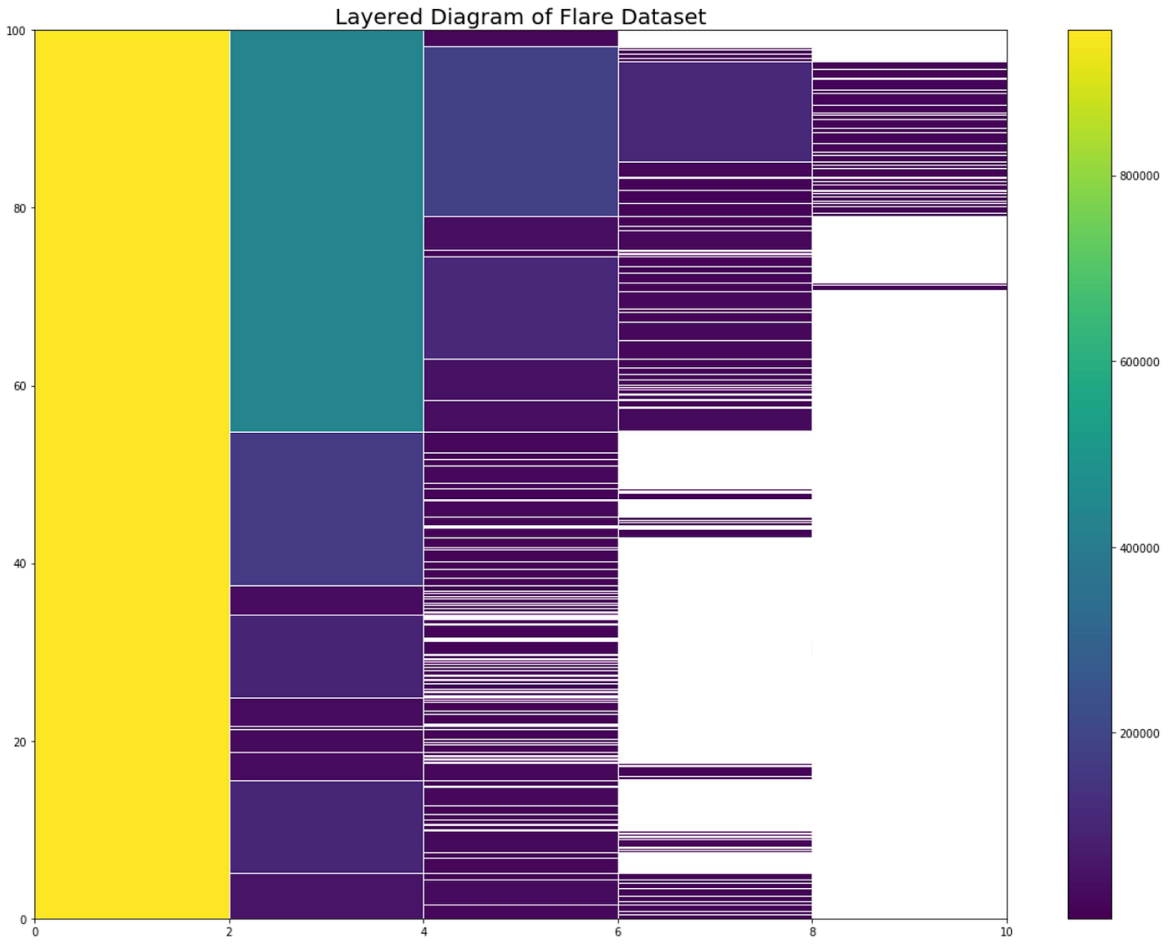
B

Source: https://livebook.manning.com/book/d3js-in-action-second-edition/chapter-6/v-6/8

Figure T8: Icicle tree Visualization

Icicle tree Visualization Example

- Visualization on Flare Dataset
  - The first column is the root which represents the size of the flare class overall,
  - And the second column is the sizes of classes within the flare case, and so on



Source: https://towardsdatascience.com/6-hierarchical-datavisualizations-98318851c7c5

Figure T9: Flare Dataset Explanation

Pansy Tree

- Uses a tree metaphor to visualize merged hierarchies.
- Uses unique icon named pansy to represent each merged node in the structure.
- Each Pansy is encoded by three colors mapping data items from three different datasets in the same hierarchical position (or tree node).
- The petals and sepal on Pansy are designed for showing each attribute's values and hierarchical information.
- Merge makes it much easier to explore and compare the structures, data items and data attributes with visual tools.

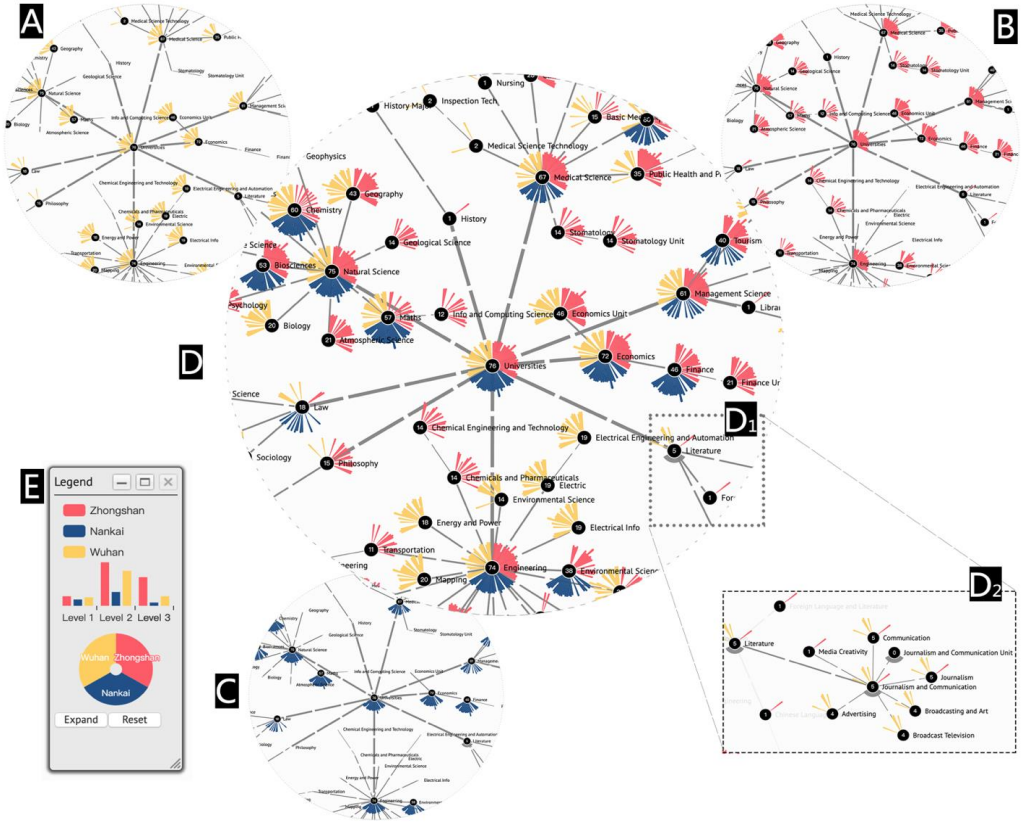
Names	Elements	Descriptions
Flower center		Node in Pansy, named "node quick-view"
Petal/Petal set		Colored petals in Pansy, represents attribute value by their height
Sepal		Sepal in Pansy, represents no attribute values but occupies hierarchies
Pansy		Pansy represents node element and its attribute values by colors

Source: https://staff.cdms.westernsydney.edu.au/~vinh/projects_TreeVis.php

Figure T10: Pansy Tree Visualization Elements

Pansy Tree Visualization

- Uses a tree metaphor to visualize merged hierarchies.
- Uses unique icon named pansy to represent each merged node in the structure.
- Each Pansy is encoded by three colors mapping data items from three different datasets in the same hierarchical position (or tree node).
- The petals and sepal on Pansy are designed for showing each attribute's values and hierarchical information.
- Merge makes it much easier to explore and compare the structures, data items and data attributes with visual tools.

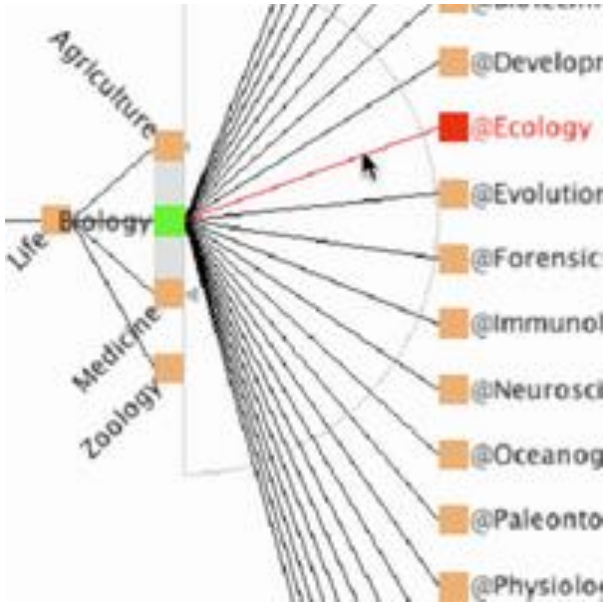


Source: https://staff.cdms.westernsydney.edu.au/~vinh/projects_TreeVis.php

Figure T11: Pansy Tree Visualization

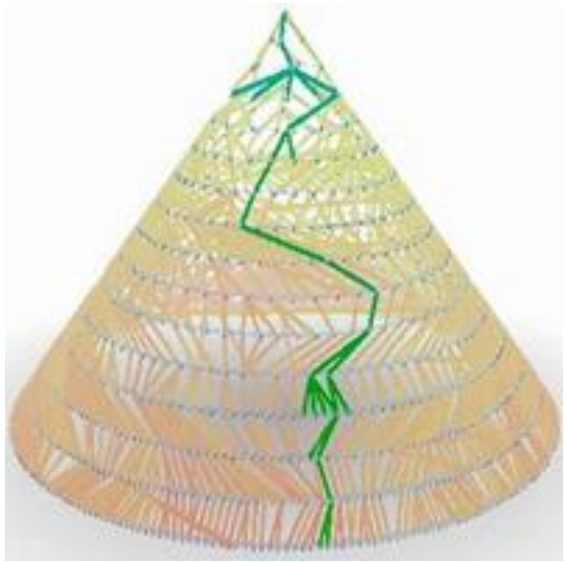
Other Adv Tree Visualization

Source: https://treevis.net/



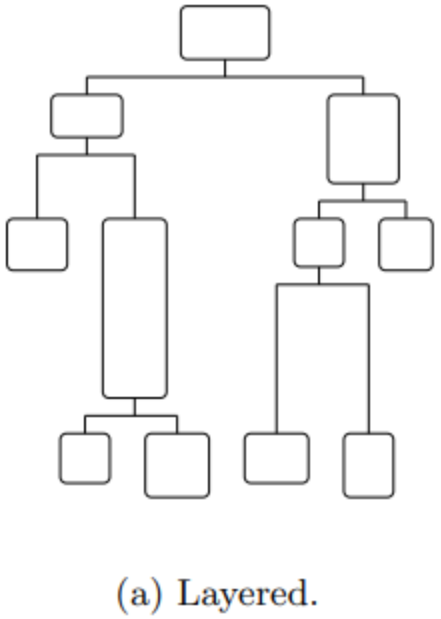
Source: https://treevis.net/

Figure T12: Control Tree



Source: https://treevis.net/

Figure T13: 3D Cone Tree Layout



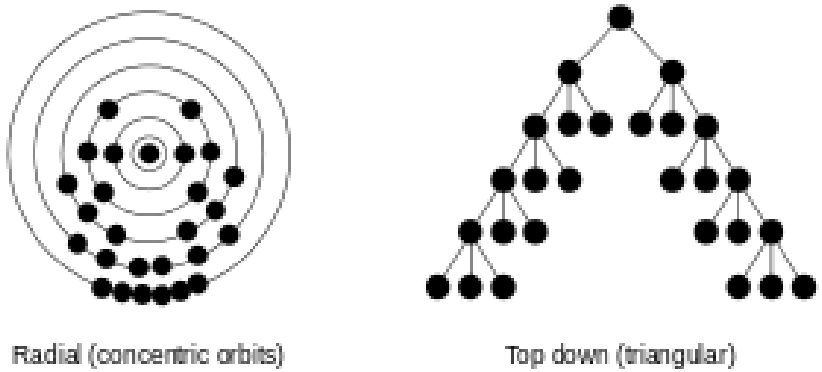
(a) Layered.

Source: https://github.com/stetrevor/non-layered-tidy-tree-layout

Figure T14: Layered Tree Visualization

Radial Tree

- Same as a normal tree visualization but in a circular format.
- The root node is positioned at the center,
- and its children radiate outward from in a radial pattern.
- level of the hierarchy is represented by concentric circles.
- <u>Any benefit?</u>
  - it's more compact than a normal tree so it is better for larger trees (much better in case of no labels in the dataset).
  - multivariate data - multiple scalar values.



Source: https://upload.wikimedia.org/wikipedia/commons/thumb/1/1a/Radial-vs-tri.svg/300px-Radial-vs-tri.svg.png

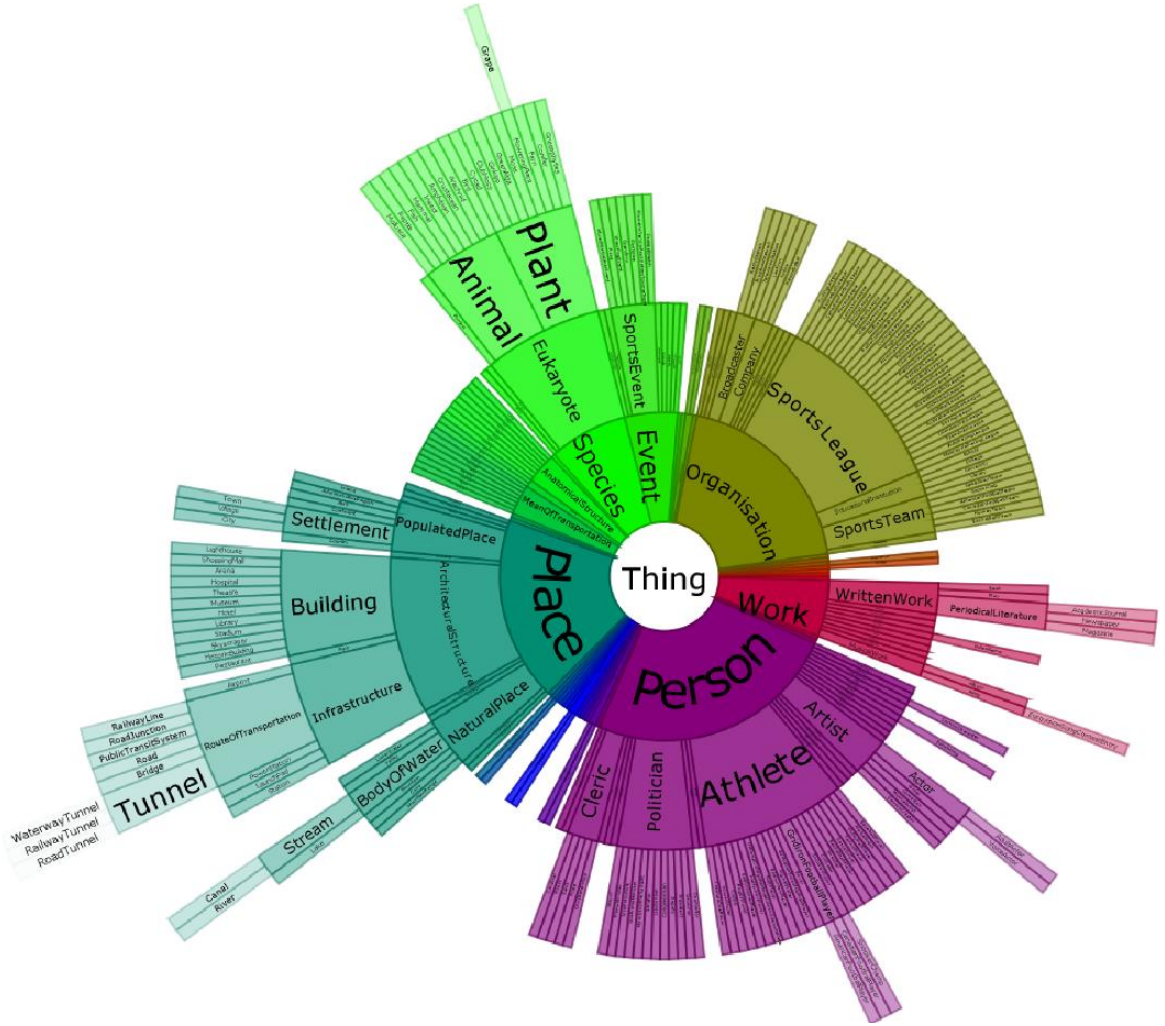
Figure T15: Radial Tree and Normal Tree Comparision

Radial Tree Cont

- Same concept as the normal layered diagram except in a radial format

Additional Benefits - Such node placements, branch curvature, and color coding, can enhance the **Aesthetics and Readability.**

Note: The choice between radial tree or normal tree visualization depends on the specific requirements of the data and the goals of the visualization



Source: hhttps://d3i71xaburhd42.cloudfront.net/7c2fe425e6a7105bd64808f7343b602a79a16c49/3-Figure2-1.png

Figure T16: Radial Tree Visualization

Radial Tree Applications!

Time series data, and that is when a design goal is to convey cyclical or seasonal trends.

- Weather and climate data, for example WeatherWheel, Weather Radials, Climate Spirals
- Human activity that relates to time (Figure T17: Radial Tree Visualization - The Baby Spike)
- Refer this page for more interesting radial visualization - https://observablehq.com/@observablehq/why-use-a-radial-data-visualization

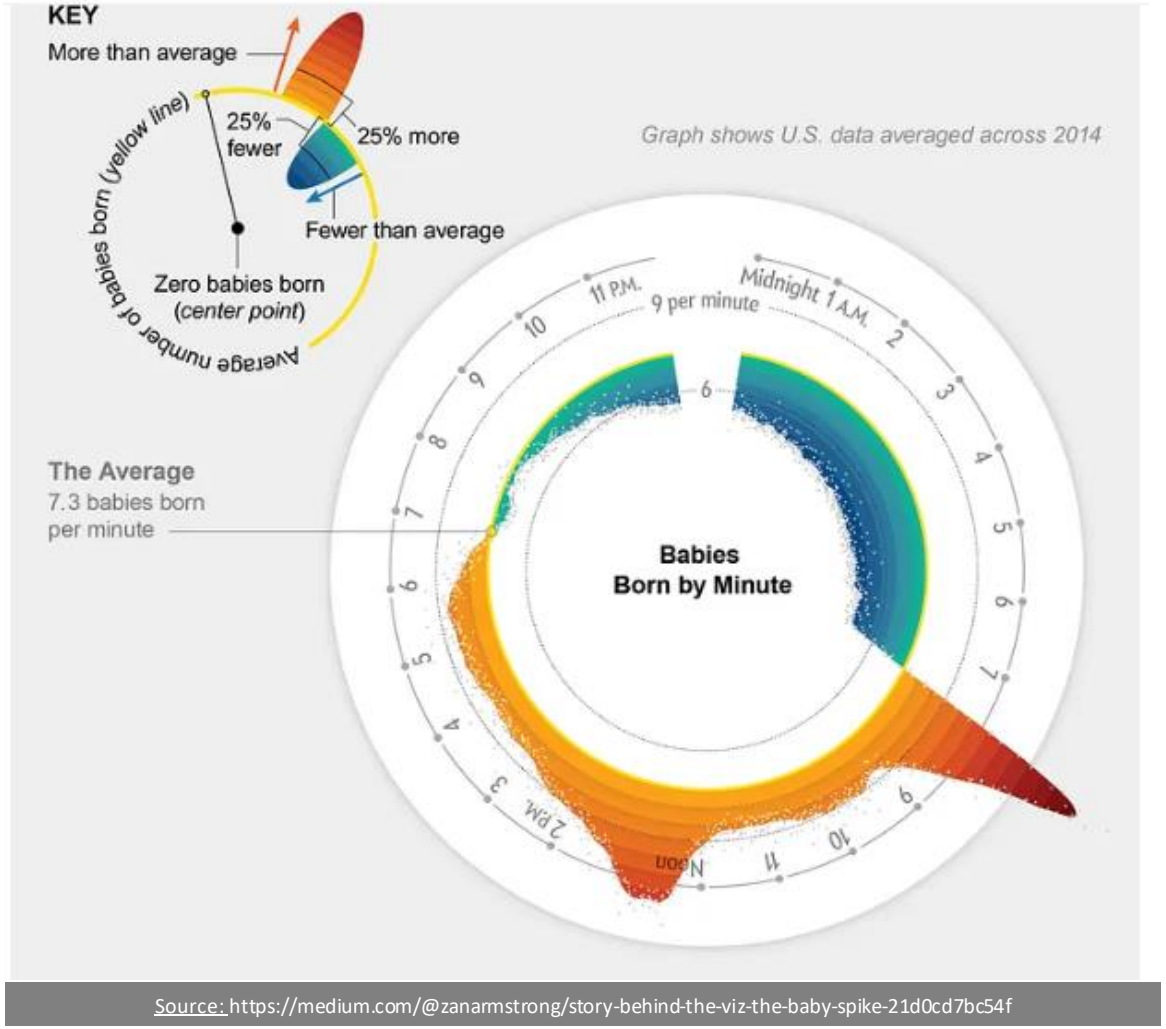
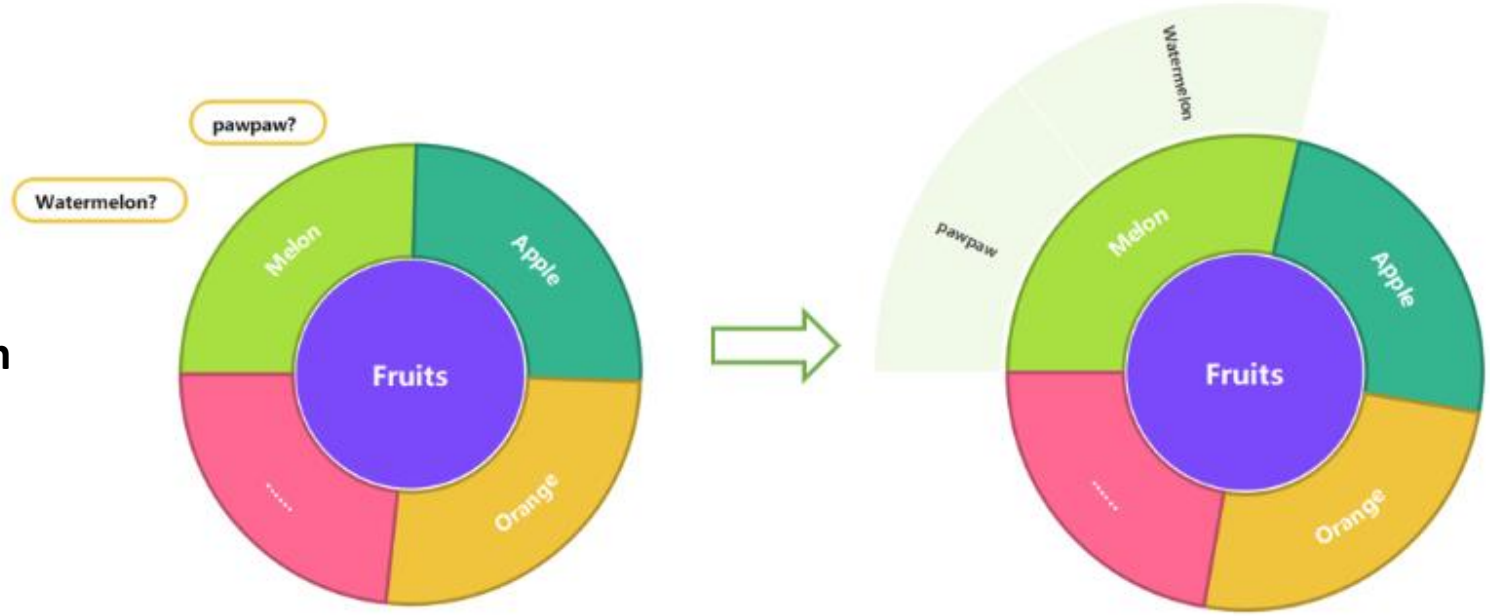


Figure T17: Radial Tree Visualization - The Baby Spike

Radial Example - Sunburst charts

Pie chart? - No

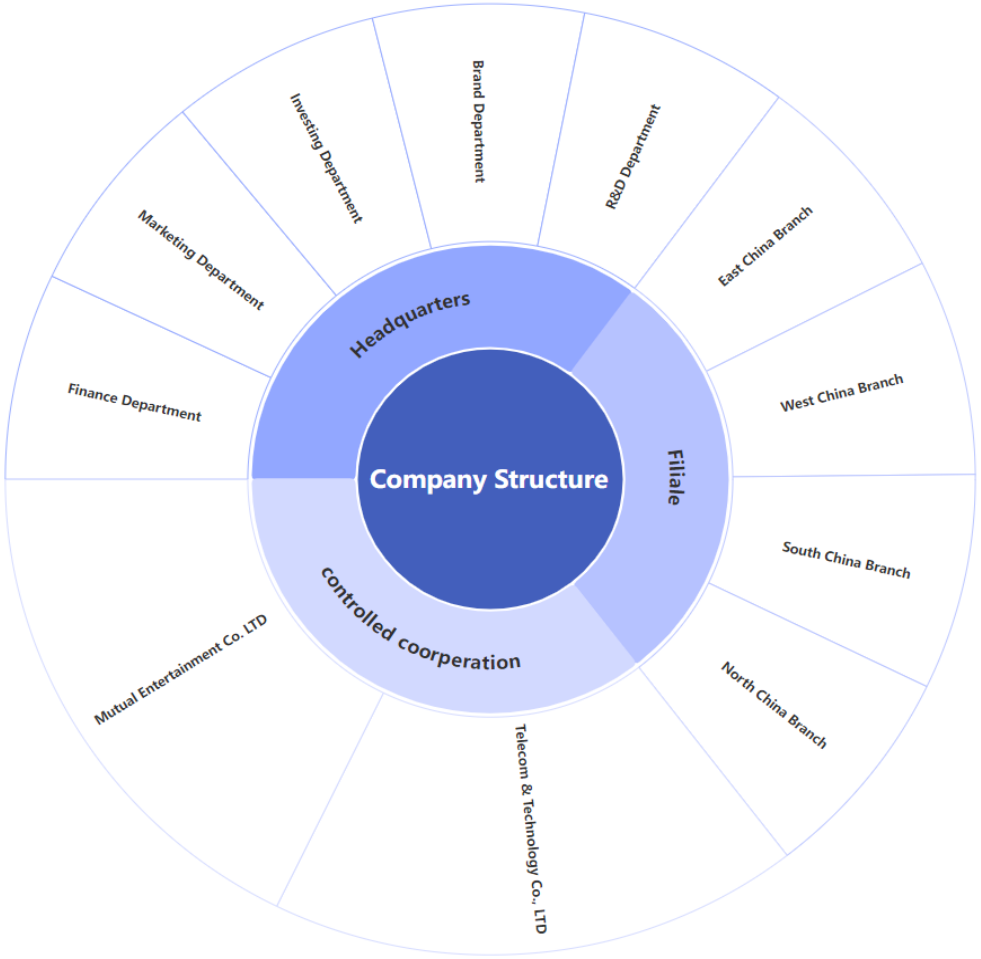
When dealing with multi-levels data
- **pie charts are not enough.**
- **the sunburst chart solved this problem**



Source: https://excel-tutorial.com/sunburst-chart/

Figure T18: Sunburst charts Visualization

Radial Example#1 - Sunburst charts

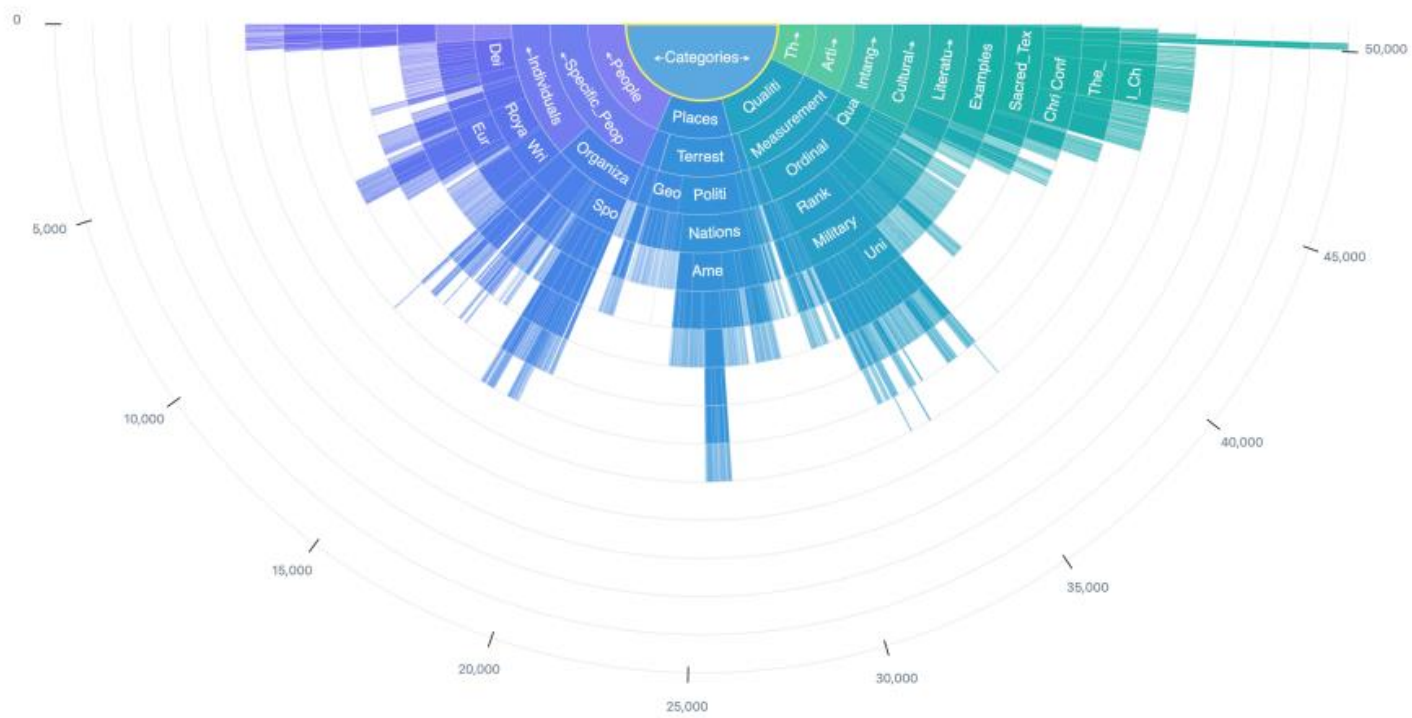


Source: https://www.edrawmind.com/article/sunburst-chart-template.html

Figure T19: Sunburst charts Visualization

Radial Example#2 - Sundown Chart

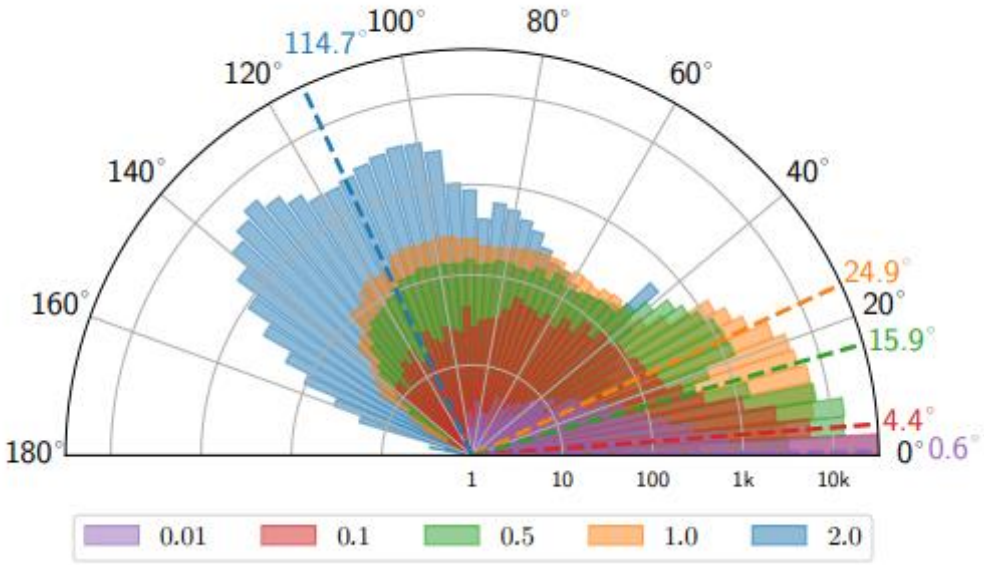
Histograms? display the distribution of numerical data,
Sundown charts are used to visualize the timing and duration of tasks



Source: https://www.edrawmind.com/article/sunburst-chart-template.html

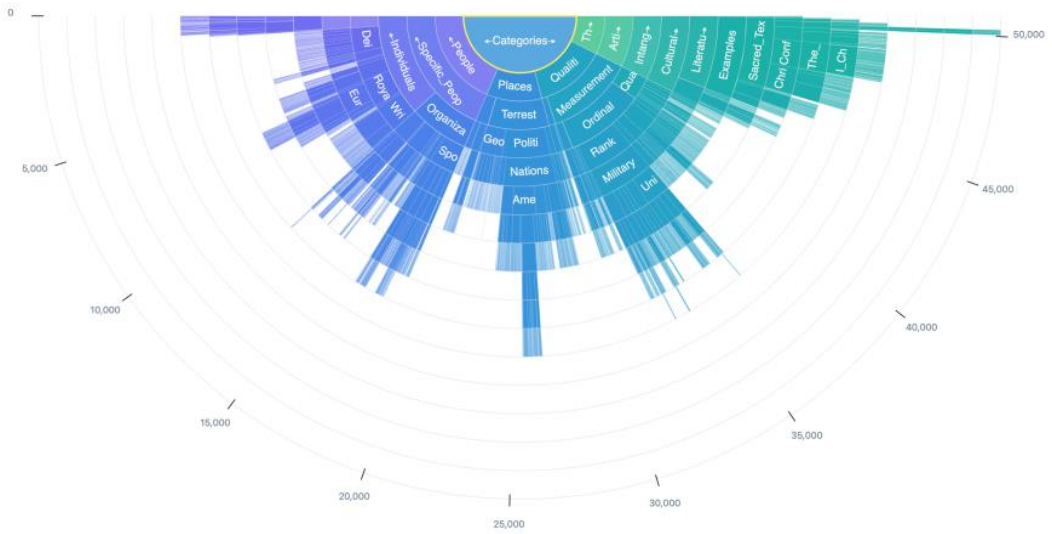
Figure T20: Sundown charts Visualization

Radial Example#2.1 - Sundown Chart



Source:https://arxiv.org/pdf/1906.10908.pdf

Figure T21: Present for ML Model Stealing attack on CIFAR10 with
various choices of threshold



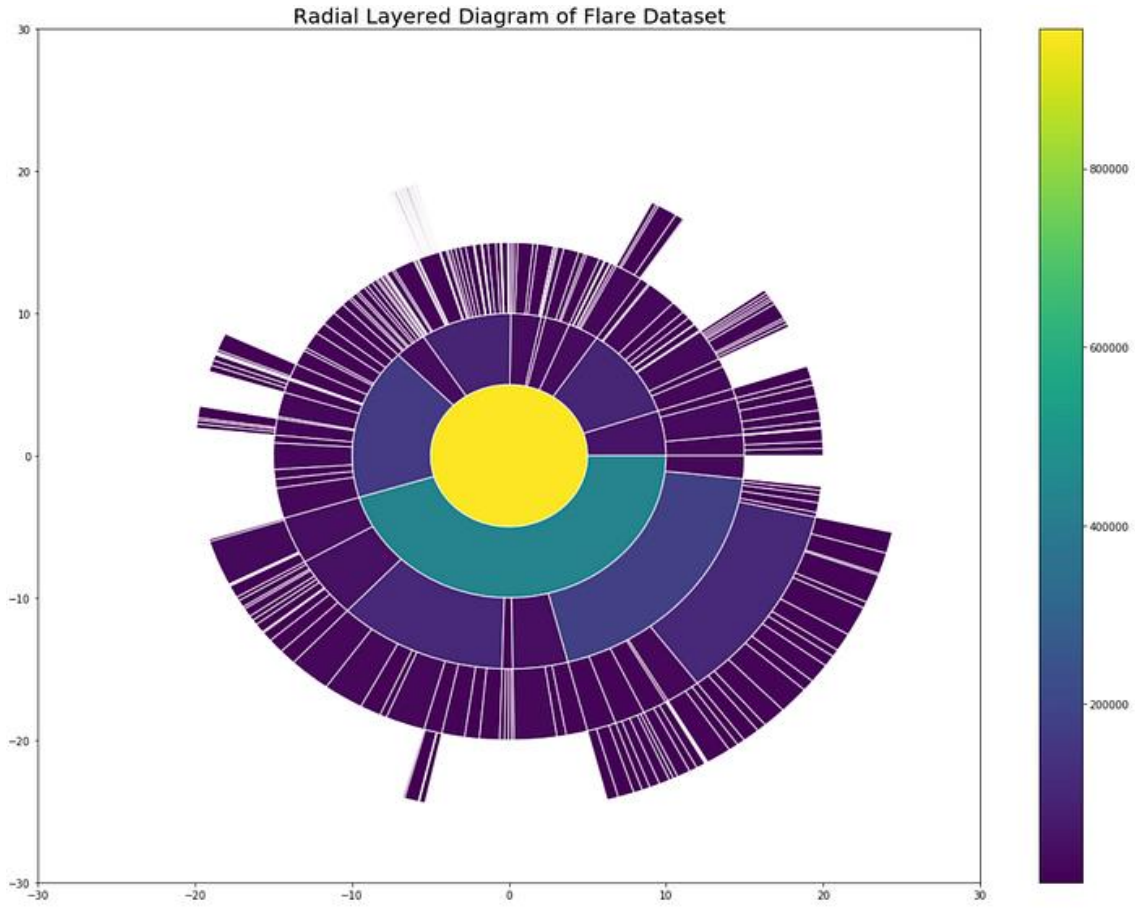
Source: https://www.edrawmind.com/article/sunburst-chart-template.html

Figure T22: Sundown charts Visualization

Radial Diagrams

The code for both layered tree visualization can be found here

- https://github.com/kruthik109/Data-Visualization/blob/main/Advanced-Visualizations/Layered-Diagram.ipynb
    - Can be easily tested on Kaggle/Google Colab/VS Code

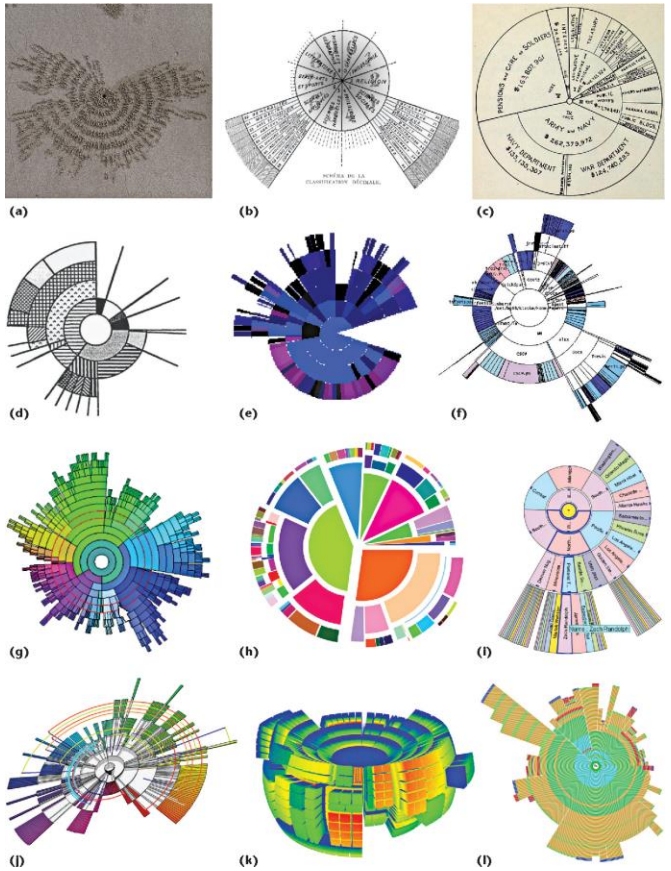


Source: https://towardsdatascience.com/6-hierarchical-datavisualizations-98318851c7c5

Figure T23: Radial Tree Visualization

Advance Radial Tree Visualizations

- **Radial Icicle Tree (RIT)** - Next Slide
- **Enhanced Radial Space-Filling Layout** - is a variation of the traditional radial space-filling layout.
- **Reversed Radial Tree** - Root node is positioned at the outermost circle or layer, and child nodes are placed closer to the center.
- **Radial Tree in Bunches** - tree nodes into clusters or bunches within concentric circles, rather than spreading them out evenly around the central point.



Source: https://www.semanticscholar.org/

Figure T24: Tree Visualization Reference

Radial Icicle Tree (RIT)

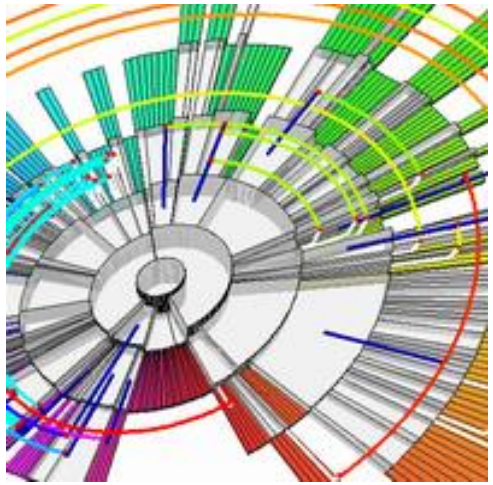
- Combines elements of **radial layouts** and **icicle plots**
- Icicle plots -
  - parent nodes represented by wider rectangles and child nodes nested within them as narrower rectangles.
- The branches are represented using rectangular "icicles" - radiate outward from the center.
- Effective way to explore - Large hierarchies where traditional tree layouts may become cluttered or difficult to interpret

Source:https://arxiv.org/pdf/2307.10481.pdf

Figure T25: Radial Icicle Tree

Other Adv Radia Tree Visualization

Source: https://treevis.net/

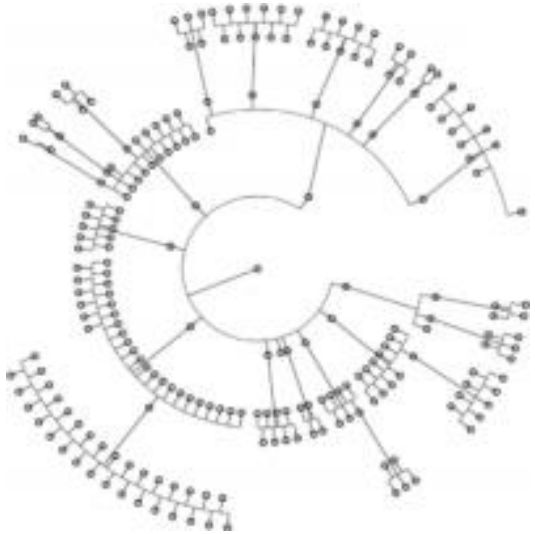


Source: https://treevis.net/

Figure T26: Enhanced Radial Space-Filling Layout

Source: https://treevis.net/

Figure T27: Reversed Radial Tree



Source: https://github.com/stetrevor/non-layered-tidy-tree-layout

Figure T28: Radial Tree in Bunches

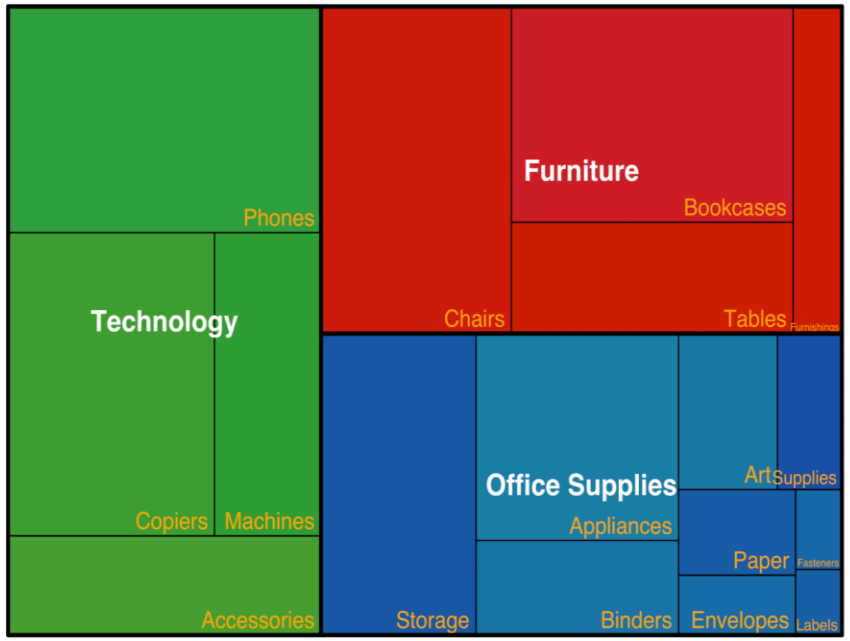
TreeMap

- Represented as a set of nested rectangles,
where each rectangle represents a node in the hierarchy.

- The size of each rectangle corresponds to a quantitative measure,
such as the size or proportion of the data it represents.

- Allows the viewer to easily see the node sizes in proportion to their parents
which in turn makes it very useful for comparisons

- Can make the static visual confusing



Source: https://towardsdatascience.com/6-hierarchical-datavisualizations-98318851c7c5

Figure T29: Treemap Example

TreeMap Cont...

- Size of each rectangle in a TreeMap corresponds to a quantitative measure.
- TreeMap visualizations often use color encoding to convey additional information about the data.
- Compactness and space-efficient layout, makes them ideal for visualizing large hierarchical datasets
- Applications?
  - Financial Analysis, Website Analytics, File System Visualization

- Can make the static visual confusing

Source: https://www.inetsoft.com/info/treemap-charts-definition-how-to-create/

Figure T30: Treemap Example

TreeMap Example...

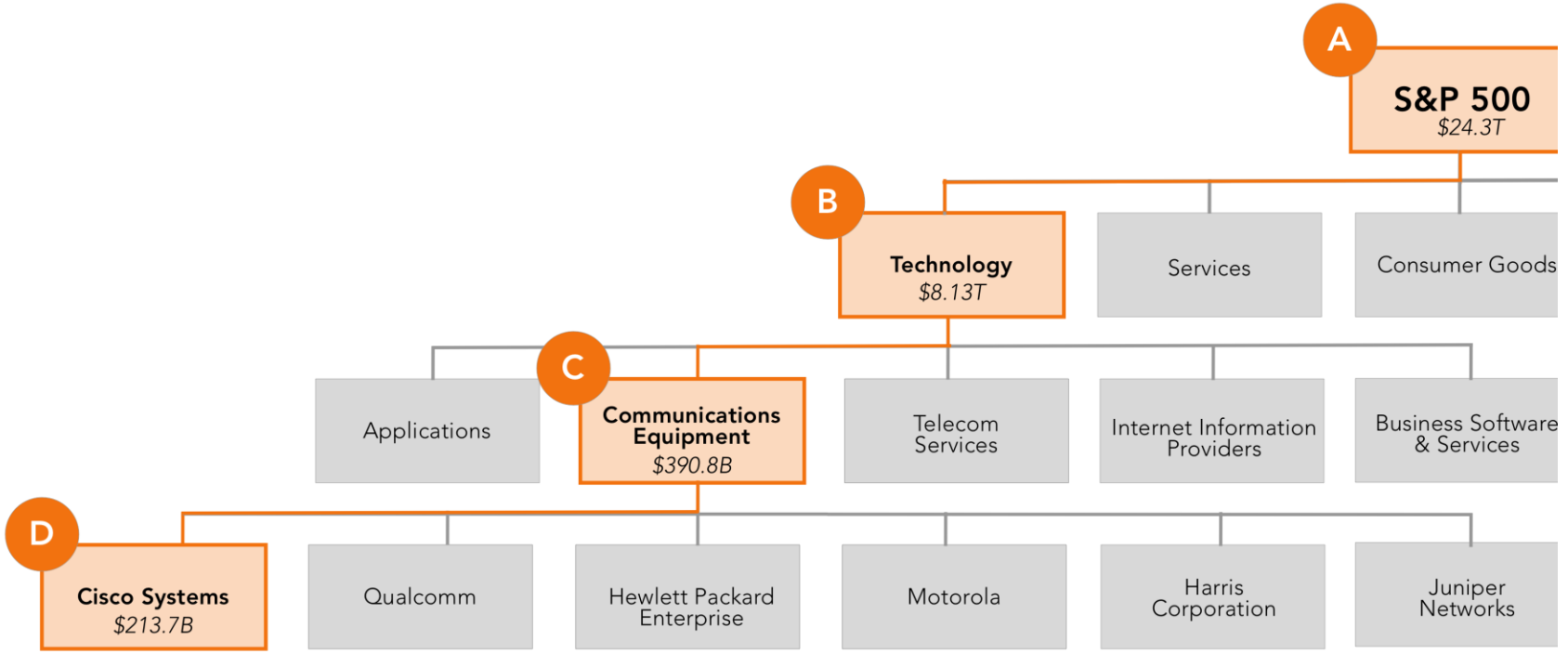
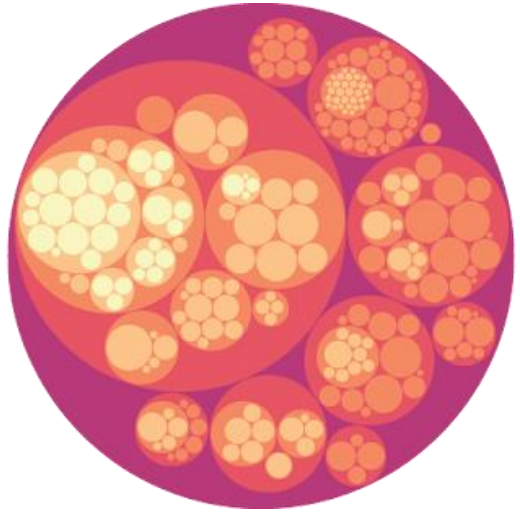


Figure: T31 A hierarchical tree diagram, showing the structure of the S&P 500. This structure is the basis of the treemap.

Circular Treemap

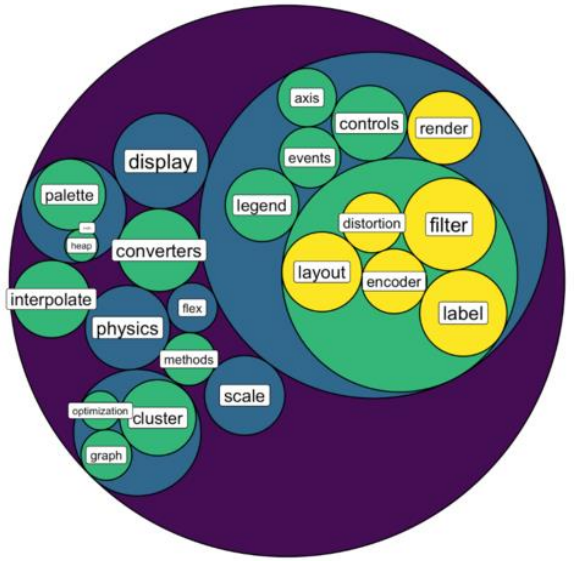
- Circular treemap is similar to a normal treemap except it nested in circle.
- Arrangement of the nodes is circular rather than rectangular
- Doesn't use space as efficiently as a normal treemap (Figure

- Example: Figure T33
  - circle area used to represent the number of employees
  - and colors to represent different departments



Source: http://visualizingrights.org/kit/charts/circle-packing.html

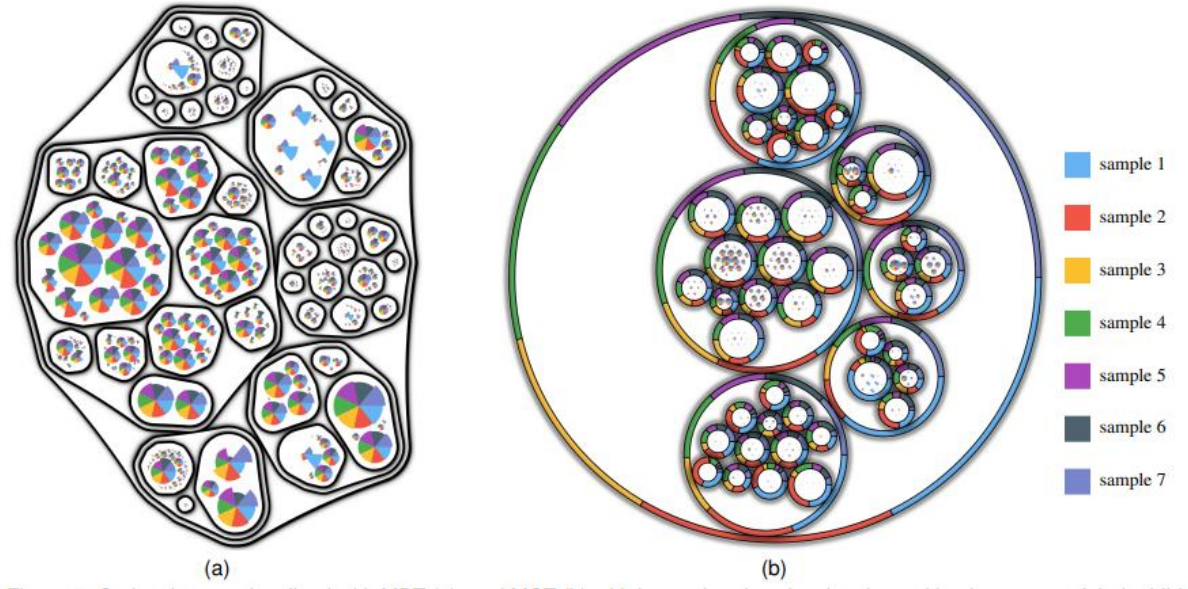
Figure T32: Treemap Example



Source: http://visualizingrights.org/kit/charts/circle-packing.html

Figure T33: Treemap Example

Circular Treemap Example



Source:https://www.researchgate.net/publication/350840794_On_the_Visualization_of_Hierarchical_Multivariate_Data

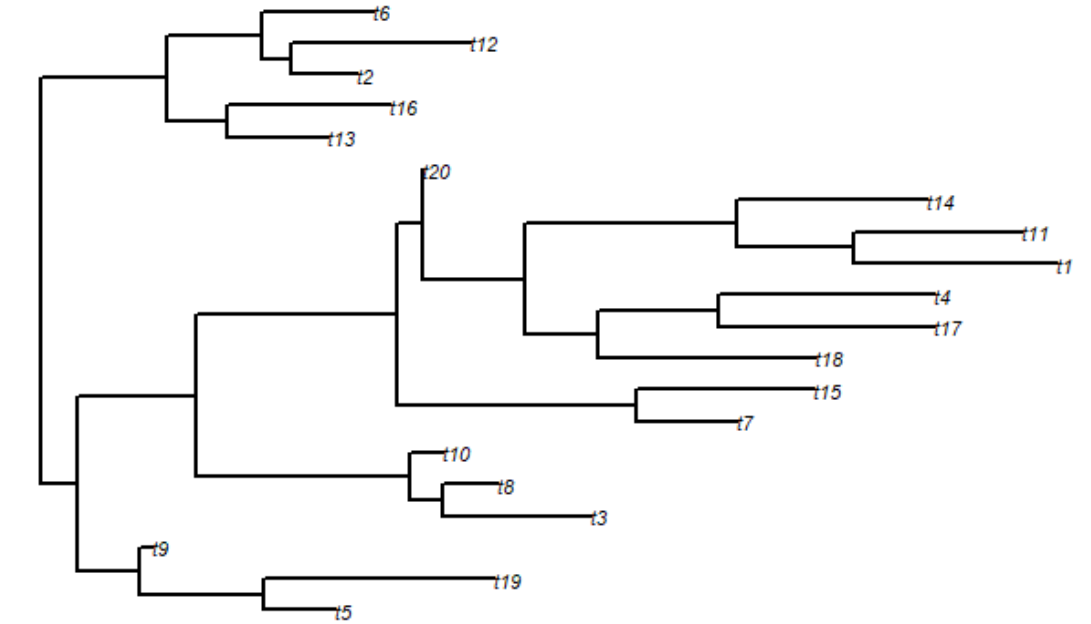
Figure T34: . Omics dataset visualized with MBT (a), and MCT (b) with internal nodes showing the multivariate mean of their children

#1 Practice Challenge

Traditional tree visualization, we can use the *ape* package, which provides functions for phylogenetic analysis and plotting

```r
# Install and load required packages
install.packages("ape")
library(ape)

# Sample data: Creating a random tree
set.seed(123)
tree <- rtree(20)

# Plotting the tree
plot(tree, cex=0.7, edge.width=2)
```

Code



Output

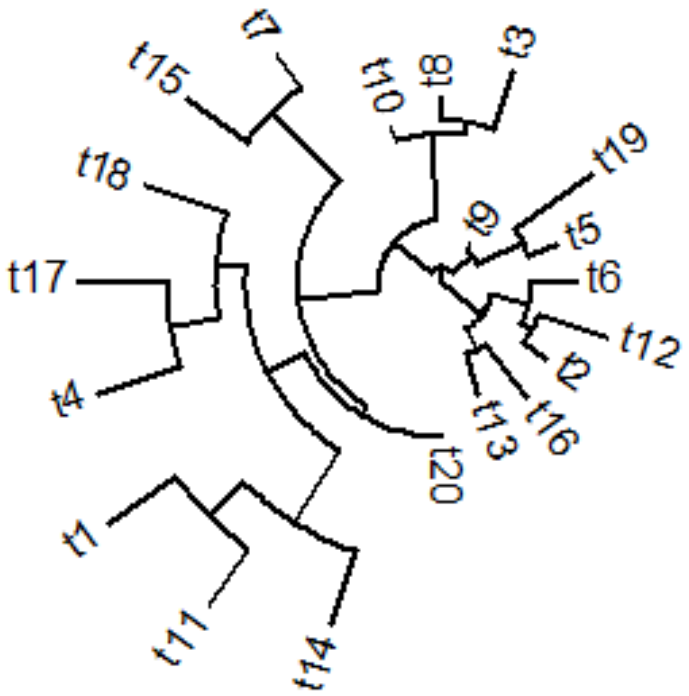
Figure: T35

#2 Practice Challange

Radial tree visualization, we can use the phytools package, which provides functions for phylogenetic analysis and plotting

```r
# - Radial Tree visualisation

# Install and load required packages
install.packages("phytools")
library(phytools)

# Sample data: Creating a random tree
set.seed(123)
tree <- rtree(20)

# Plotting the radial tree
plotTree(tree, type="fan", cex=0.7, edge.width=2)
```

Code



Output

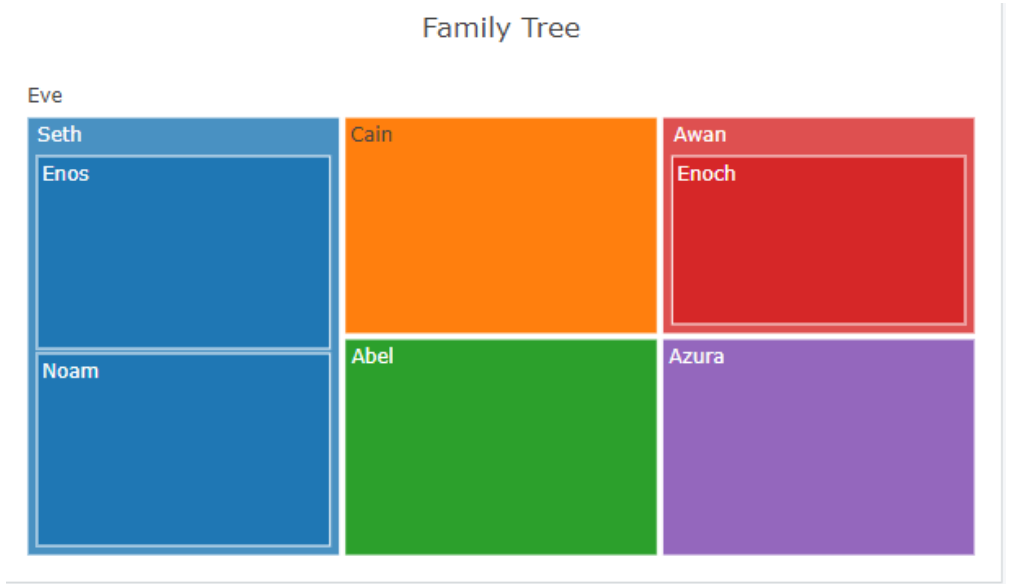
Figure: T36

#3 Practice Challenge – TreeMap visualization

```r
# - TreeMAP visualisation
# Install and load required packages
install.packages("igraph")
library(igraph)

# Sample data: Creating a tree structure
tree <- make_tree(10, children = 2, mode = "undirected")

# Plotting the layered tree
plot(tree, layout=layout_as_tree, vertex.size=15, vertex.label.cex=0.8)


library(plotly)

# Define hierarchical data
labels <- c("Eve", "Cain", "Seth", "Enos", "Noam", "Abel", "Awan", "Enoch", "Azura")
parents <- c("", "Eve", "Eve", "Seth", "Seth", "Eve", "Eve", "Awan", "Eve")

# Create TreeMap
fig <- plot_ly(
  type = "treemap",
  labels = labels,
  parents = parents
)

# Customize layout
fig <- fig %>% layout(
  title = "Family Tree",
  margin = list(l = 10, r = 10, t = 50, b = 10)  # Adjust margin for better visualization
)

# Show TreeMap
fig
```

Code

Source: https://plotly.com/r/treemaps/



Output

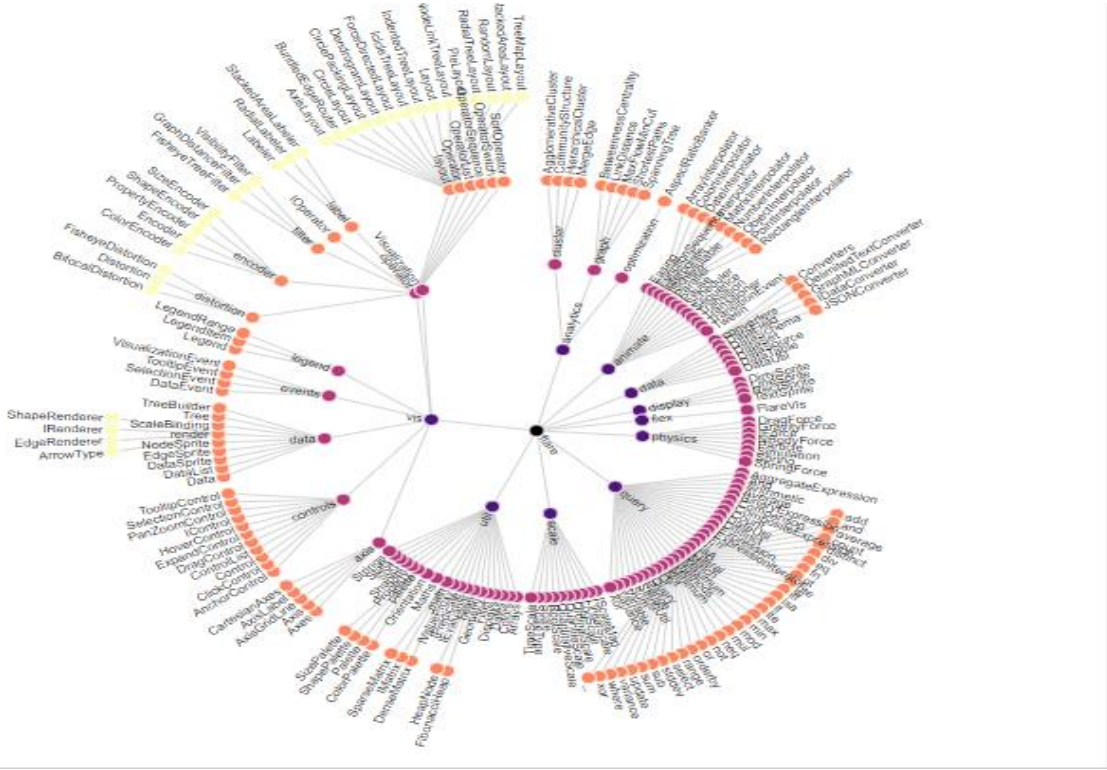
Figure: T37

#4 Practice Challange

Radial tree visualization (For complex data visualization)

Visit:
https://vega.github.io/editor/#/examples/vega/radial-tree-layout

```
{
  "$schema": "https://vega.github.io/schema/vega/v5.json",
  "description": "An example of a radial layout for a
  node-link diagram of hierarchical data.",
  "width": 720,
  "height": 720,
  "padding": 5,
  "autosize": "none",

  "signals": [
    {
      "name": "labels", "value": true,
      "bind": {"input": "checkbox"}
    },
    {
      "name": "radius", "value": 280,
      "bind": {"input": "range", "min": 20, "max": 600}
    },
    {
      "name": "extent", "value": 360,
      "bind": {"input": "range", "min": 0, "max": 360,
      "step": 1}
    },
    {
      "name": "rotate", "value": 0,
      "bind": {"input": "range", "min": 0, "max": 360,
      "step": 1}
    },
    {
      "name": "layout", "value": "tidy",
      "bind": {"input": "radio", "options": ["tidy",
      "cluster"]}
    },
    {
```



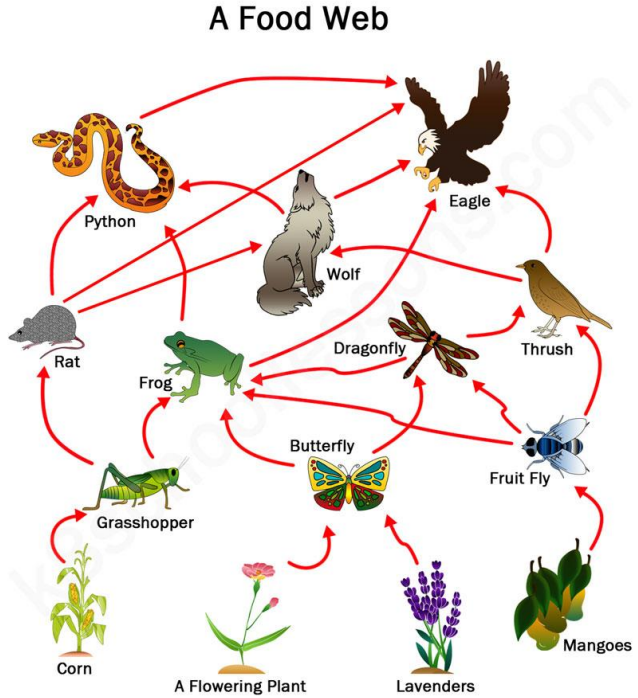
Check Out

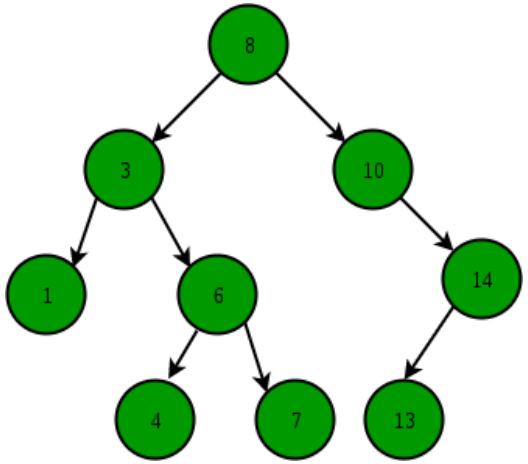
Tree Visualization:
https://treevis.net/

GitHub Code Source:
https://github.com/tusharnayan10/tree-visualization.git

Tree Drawing Algorithm

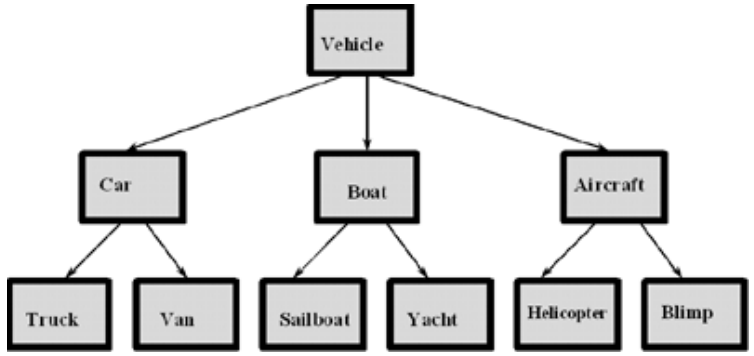
- Types of Trees?
- What is Tree Drawing?
- Why it is important?

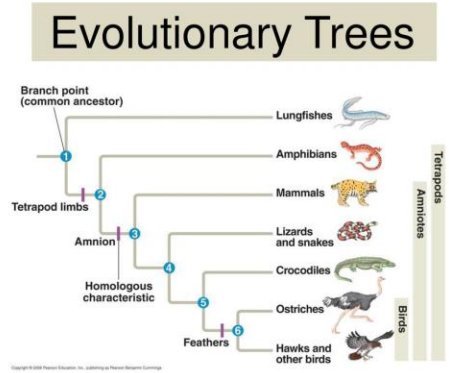


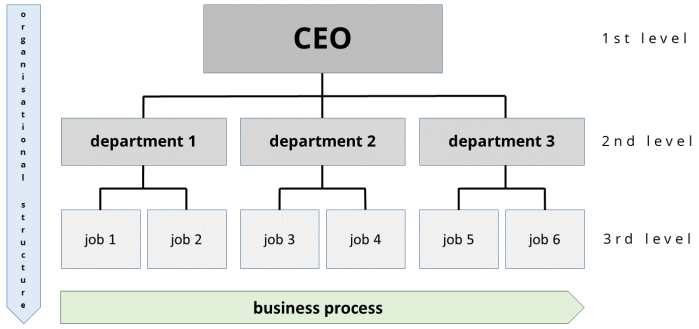


Applications

- Object Oriented class hierarchies
- Organization Charts
- Structure of a website
- Evolutionary Trees
- Molecular Drawings

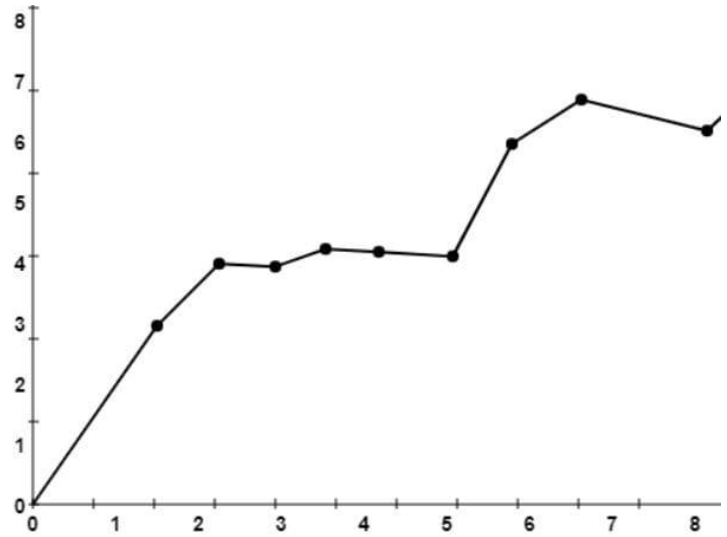


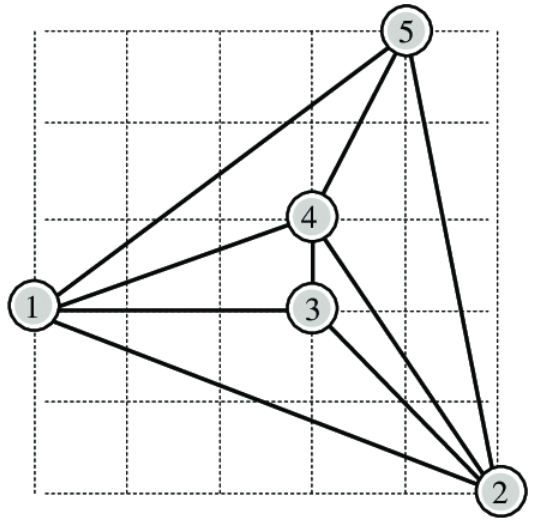


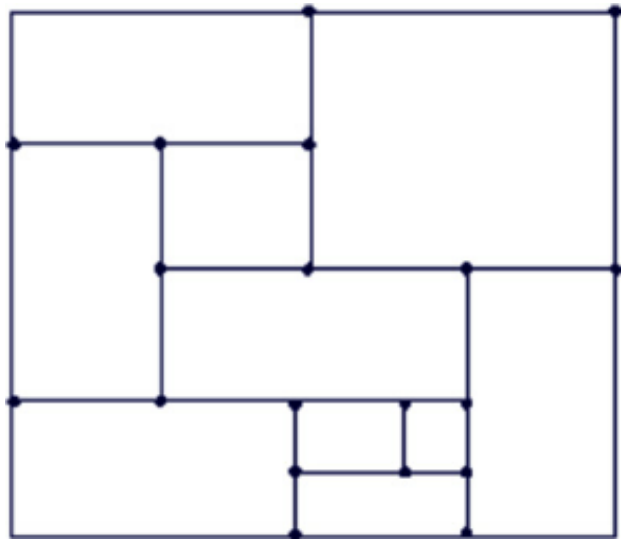


Drawing Conventions

- Polyline Drawings
- Orthogonal Drawings
- Upward and Non-Upward Drawings
- Grid Drawings
- Planar Drawings
- Straight Line Drawings







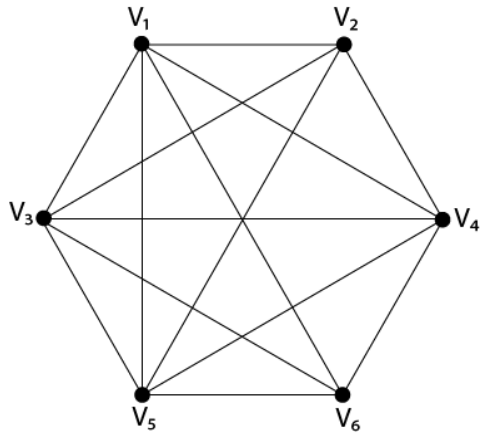


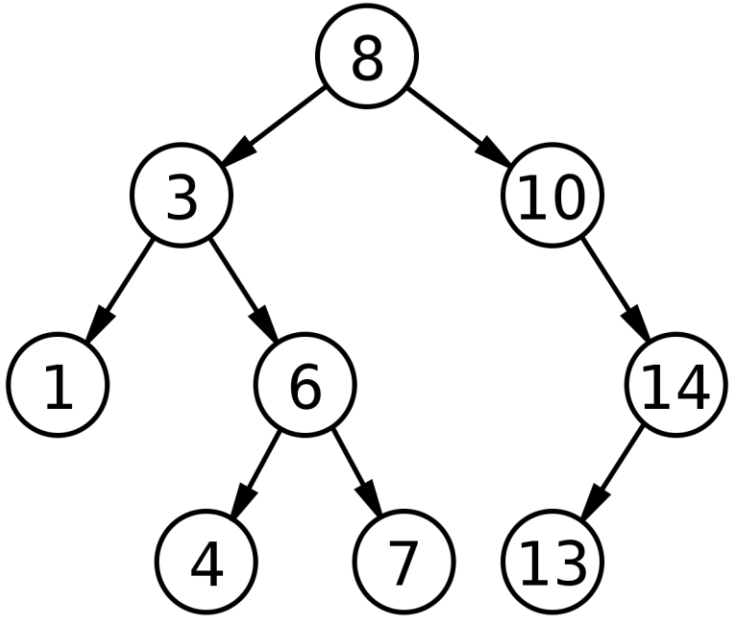
Fig: G_1

Aesthetics

- What is the aesthetics in terms of Graph Drawing

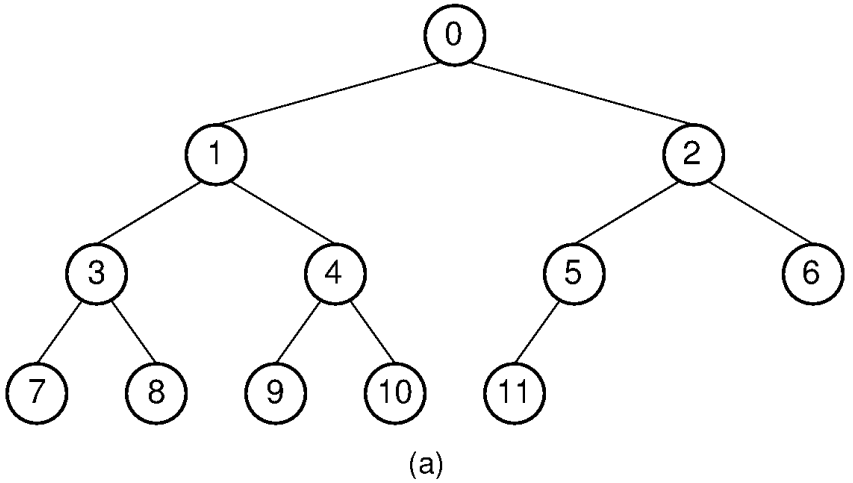
Important Aesthetics of Drawing of Trees

- Area
- Aspect Ratio
- Subtree Separation
- Closest Leaf
- Farthest Leaf



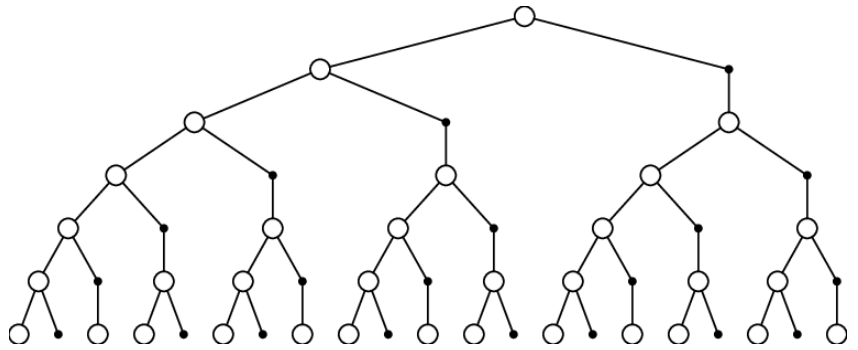
Level Based Approach

1. Nodes that are same distance from root are horizontally aligned.



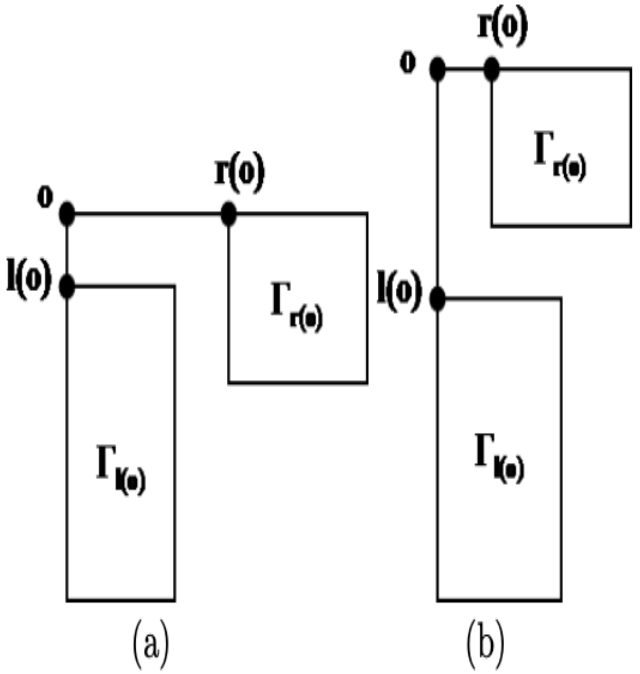
Disadvantages

1. Area- n^2
2. Width will much larger than height if the tree is balanced



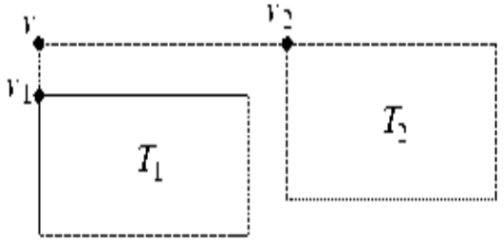
H-V Approach

- Divide and conquer constructs tree drawings with root at top-left, placing left and right subtrees next to or below each other.
- It is an upward tree and the configuration can be either horizontal or vertical

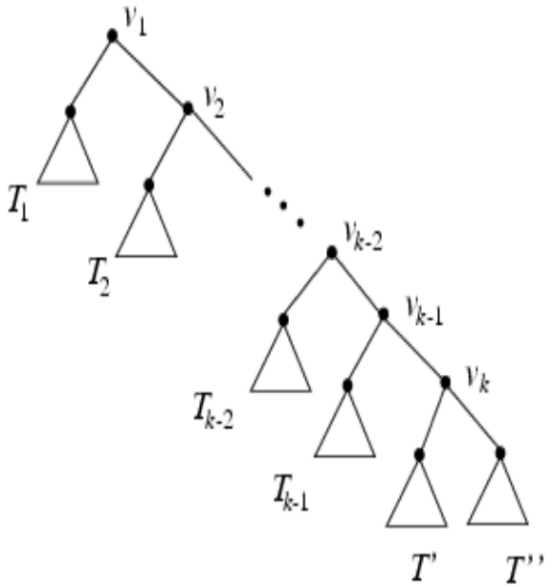


Path Based Approach

- The path-based approach draws a binary tree by recursively laying out nodes towards a distinguished node, then placing subtrees opposite the distinguished node's children.



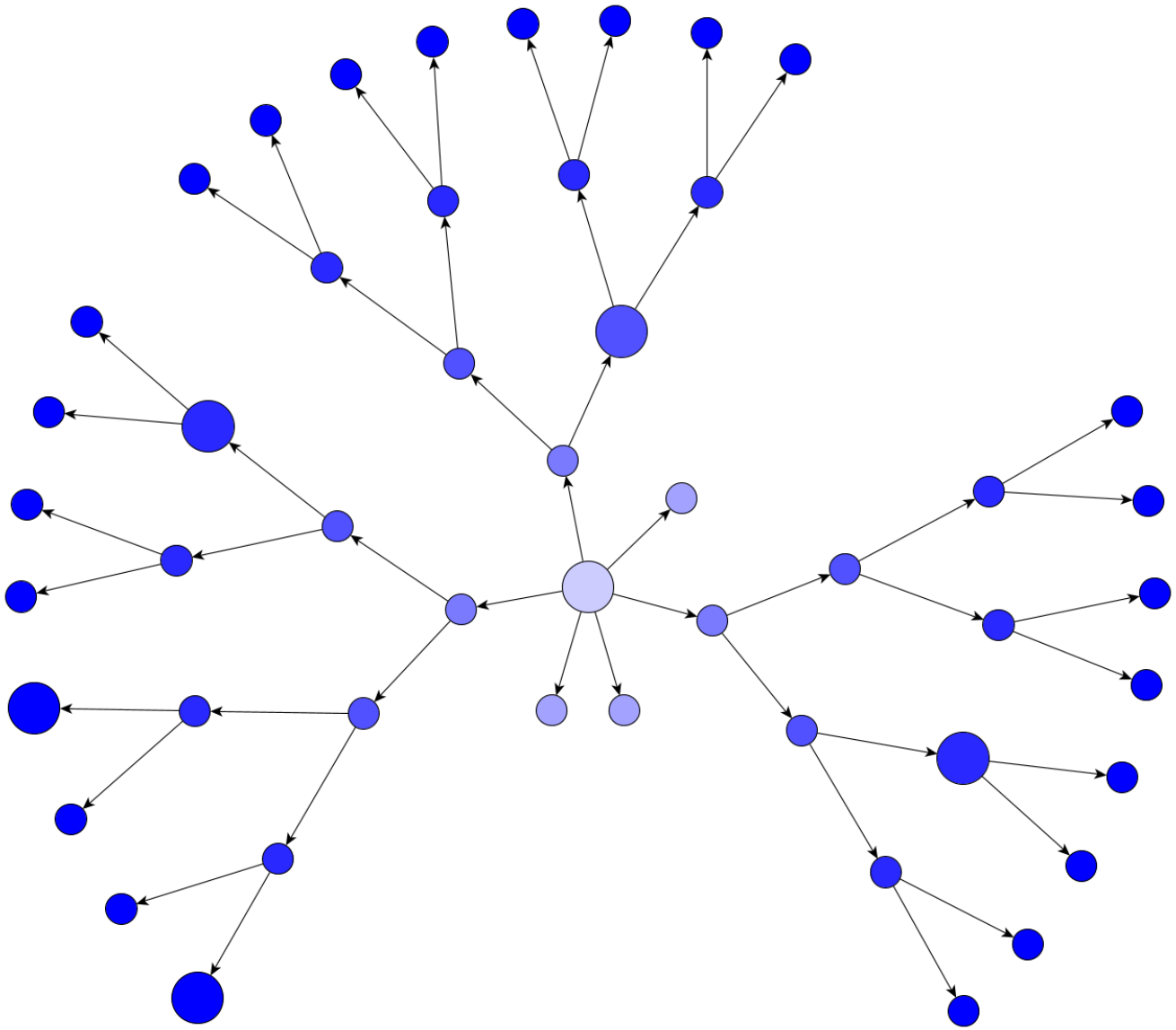
(a)



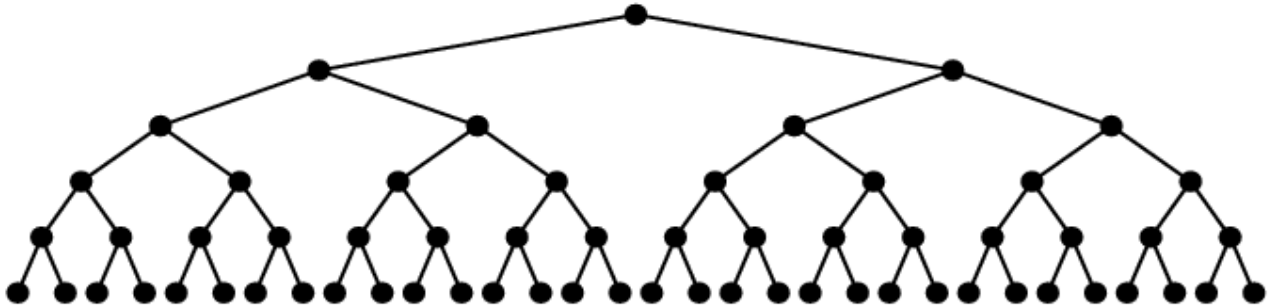
(b)

Ring Circular Layout Approach

- In these algorithms, children are placed on the circumference or the interior of a circle centered at their parents

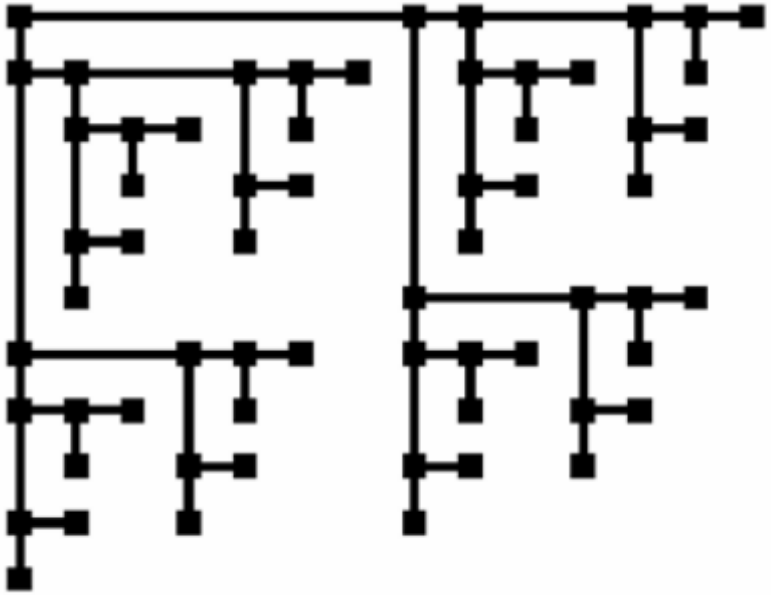


Algorithm for Drawing Binary Trees



Unordered Trees

- Find a Separator Edge or a Separator Node
- Split Tree
- Assign Aspect Ratios
- Draw Partial Trees
- Compose Drawings



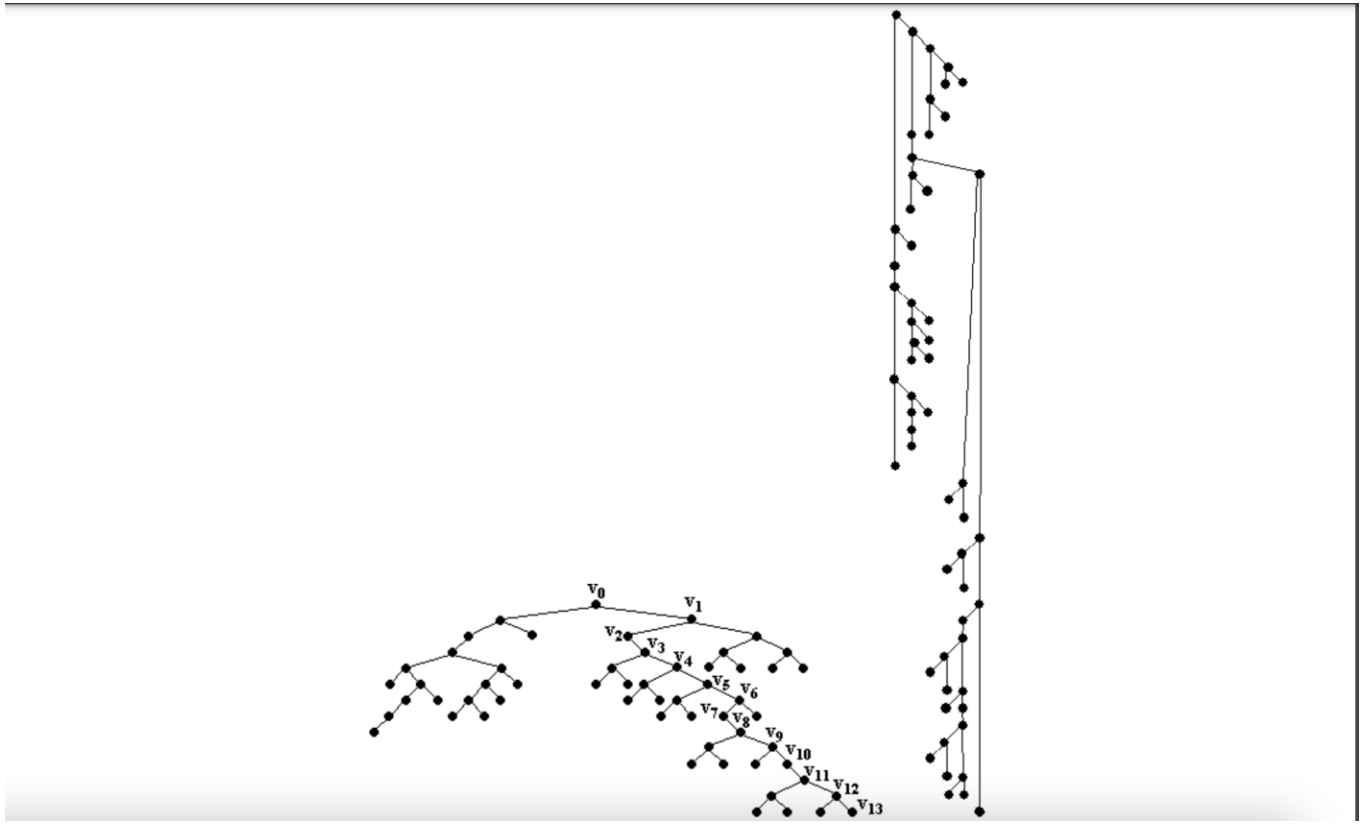
Drawing of a complete Binary Tree with 63 nodes

Ordered Trees

- Fixed Spine
- Arbitrary Spine

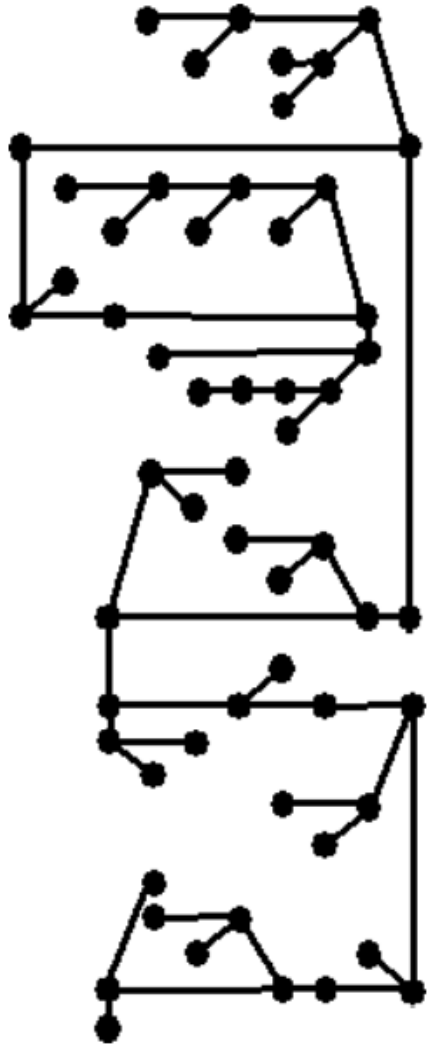
Fixed Spine

- It uses path based approach
- It is an upward planar straight line grid drawing
- Optimal area is nlogn

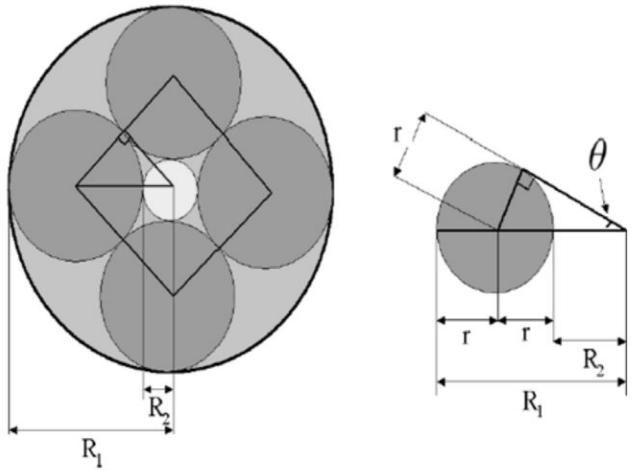


Arbitrary Spine

- It also uses path based approach
- But user can control the aspect ratio
- O((n/A) log A) height and O(A + log n) width.
- A is the aspect ratio



Algorithm for general Trees



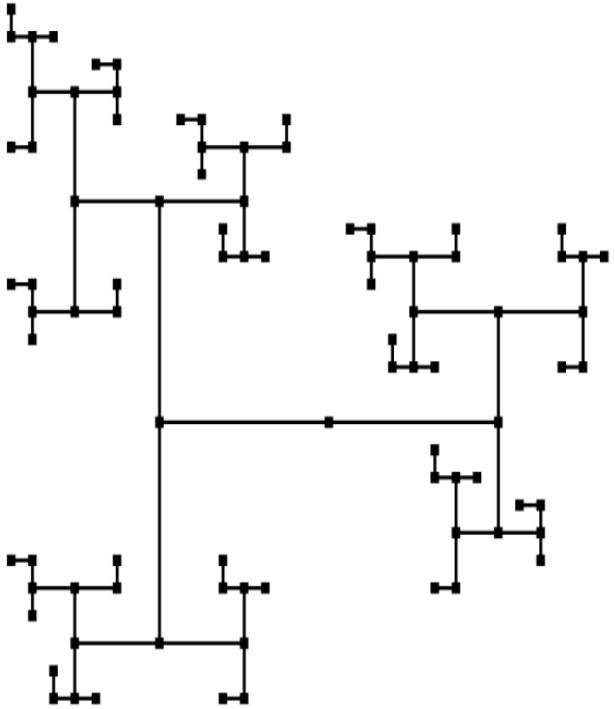
- Unordered Trees

Ring Algorithm

- Sort the children by their number of children;
- Find the smallest k for which the sum of the number of children of the first k children expressed as a fraction of the total number of grandchildren is greater or equal to f(k);
- f(n) is the fraction of the area left after n circles have been placed in the ring
- Place first k children in the outermost ring;
- Place the rest of the children in the same way in the inner rings;

Ordered Trees

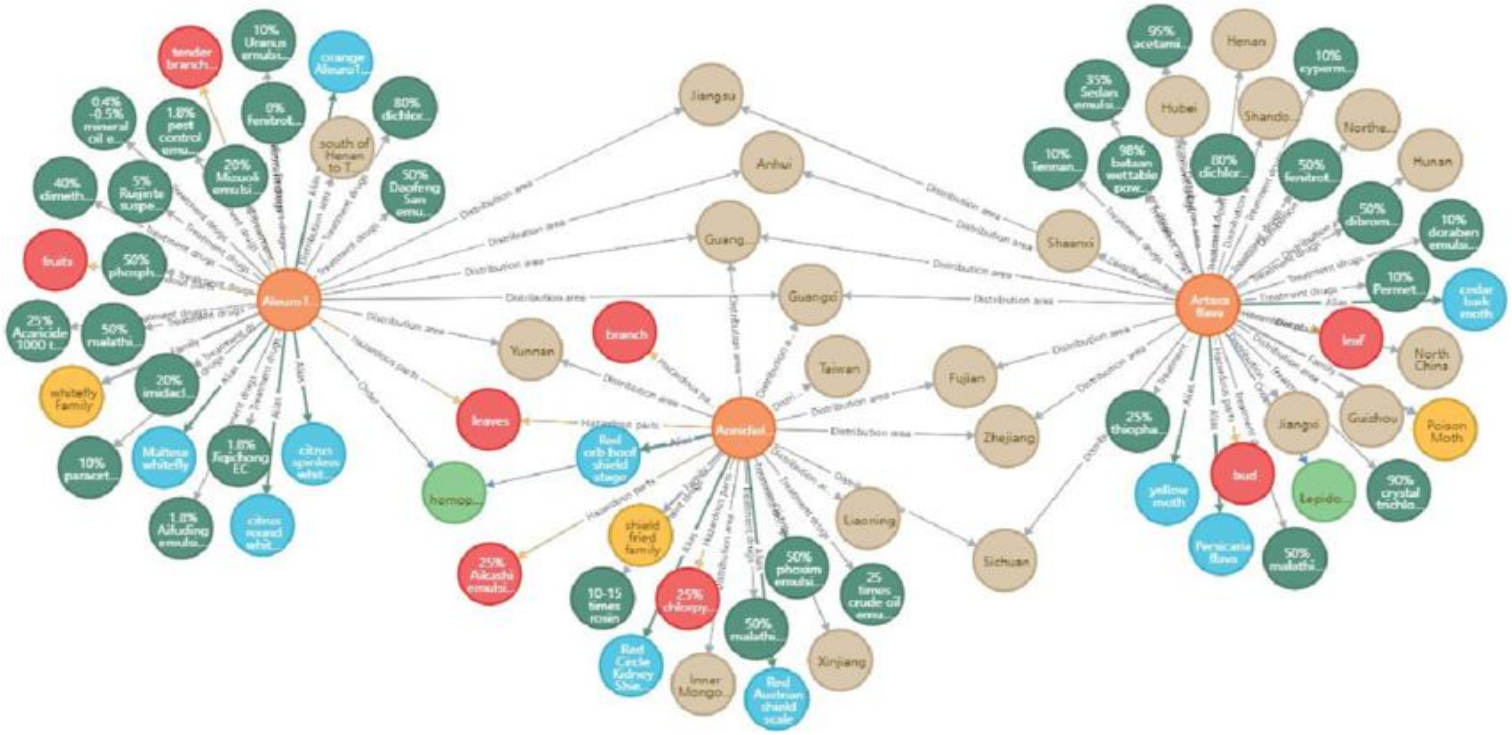
- Let T be an ordered tree with n nodes.
- Break tree recursively into several subtrees.
- Combine them to make the Tree
- The tree should be non-upward planar straight line grid-drawing



Other Tree Drawing Algorithms

- Hyperbolic Tree
- Botanical Tree
- Space Tree
- Adaptive Tree Drawing
- Hexagonal Tree Drawing

A Survey of Multiple Tree Visualization Techniques



An Information Visualization paradigm

–> Martin Graham &. Jessica Kennedy

Paper here.

Introduction to Survey

- Tree visualization has been crucial in Information Visualization, with extensive research on single tree instances, leading to diverse layout styles and interaction techniques.
- Limited Focus in Previous Surveys
- Early Classification Attempts surveys classified graph presentation techniques without considering multiple tree visualizations.
- Through this presentation, I will be covering the survey paper for the multiple tree visualization.

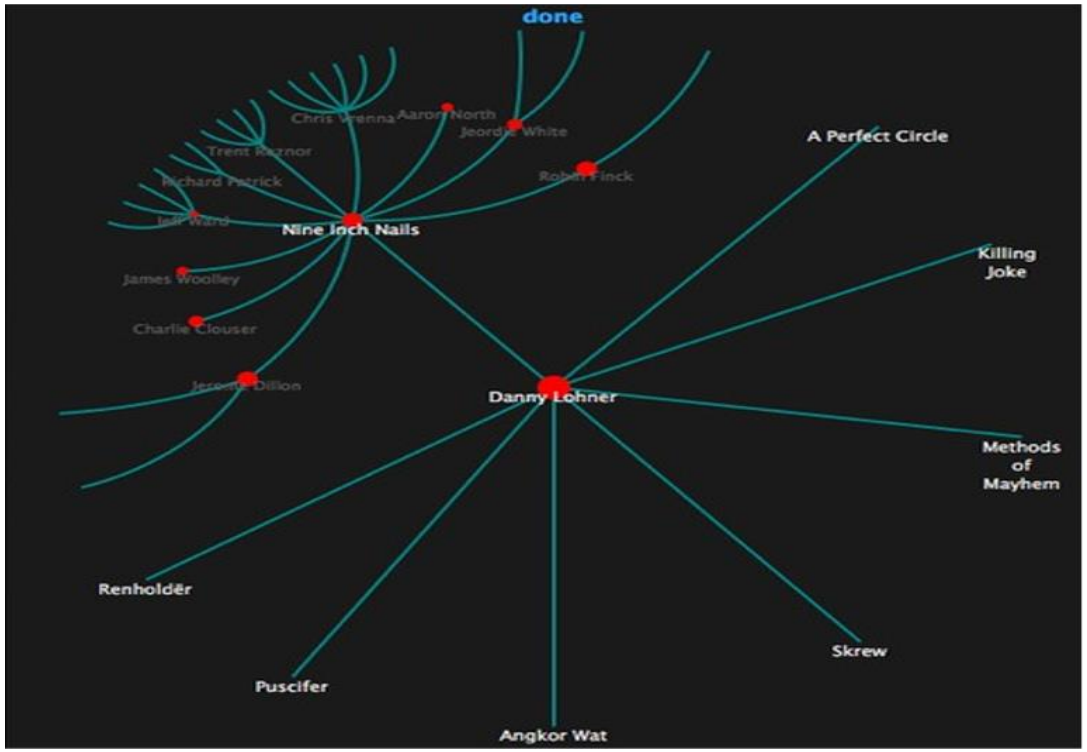


Figure: S1

Application Area

- **Bioinformatics**: Produce multiple conflicting phylogenies and taxonomies over the same or overlapping groups of species.
- **Faceted classifications:** When items can be filtered by multiple *hierarchically-organized* attributes, with the most common example being internet shopping sites where products are classified by price, type, maker, etc.
- **Schema / Ontology mapping:** Ontologies are more complex they do contain major tree-based structures such as concept and relation hierarchies, where the challenge is to find the best match between the various parts of related schemas or ontologies.
- **Software development:** Software developers may wish to compare these snapshots to track package growth and discover where development effort was concentrated at any particular time.

Multiple Tree Structure

- The logical structures formed by merging multiple trees dwell in the messy domain between single-tree structures and general graphs.
- No node overlap occurs between a set of trees. A collection of disjoint trees as shown in Figure a.
- Trees reuse fragments of other classifications instead of building new structures where a previously defined structure already resides, as shown in Figure b.
- A separate parent-node orientation across the trees then constructs its paths between them by defining other internal nodes as shown Figure c.
- If there is no global orientation shared between the structures, then the combined trees form a polyarchy structure as shown in Figure d.

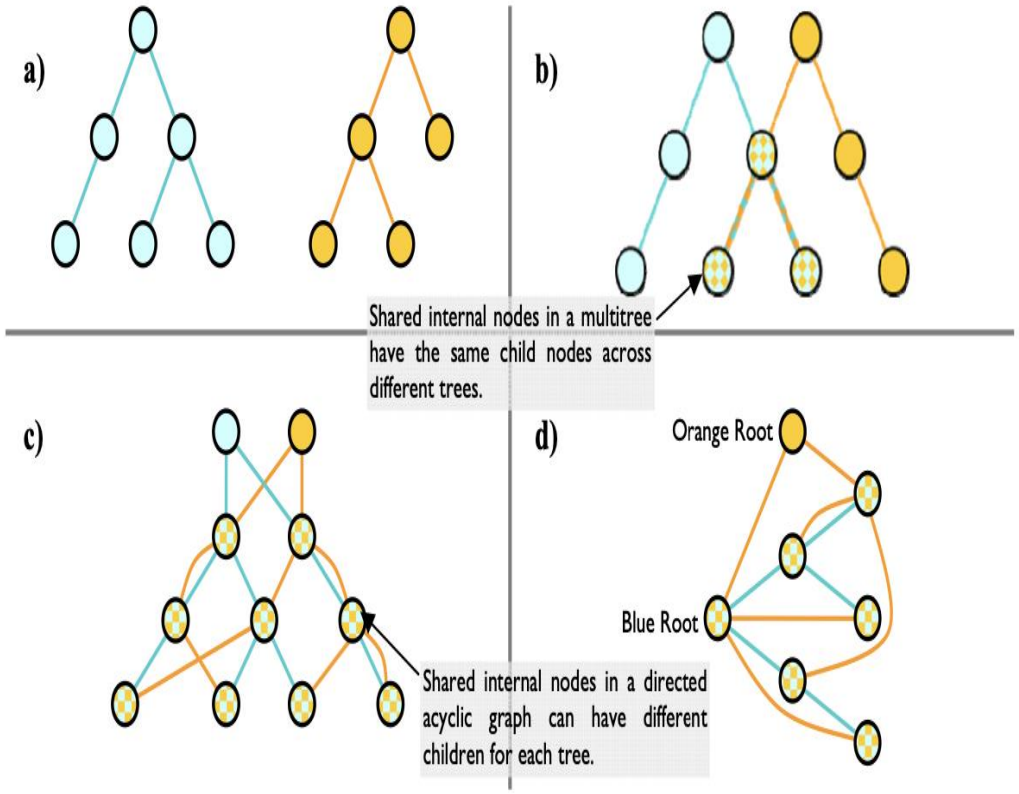


Figure: S2

Two Trees Connected

- For many scenarios, simple node overlapping does not adequately describe the mapping between the trees.
- In that case, mapping is described through inter-tree edges that relate the trees to each other as shown in Figure 3.
- These relations can either be one-to-one or one-to-many.
- These links can indicate a spectrum of different relations based on set notation such as includes, congruence, dissimilarity, etc. and one node may be the source or target of several links involving one or more other trees.

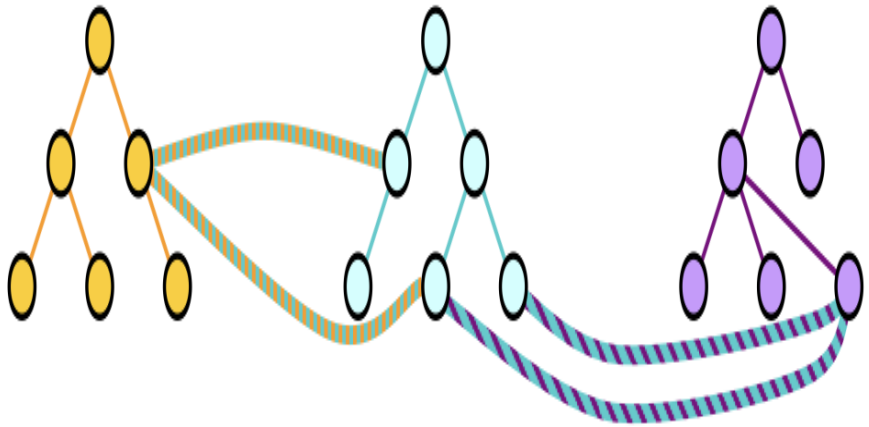


Figure: S3

Representation – Single Tree

- Single tree layout divides into several categories, based on the graphical method used to indicate a parent-child relationship.

  a) Node-link - Most common and wide-spread use
  b) Nested – Space division in OS
  c) Adjacency – Ontology or Biological hierarchy
  d) Indented List – File System in Windows
  e) Matrix Representation – Easy to look up information

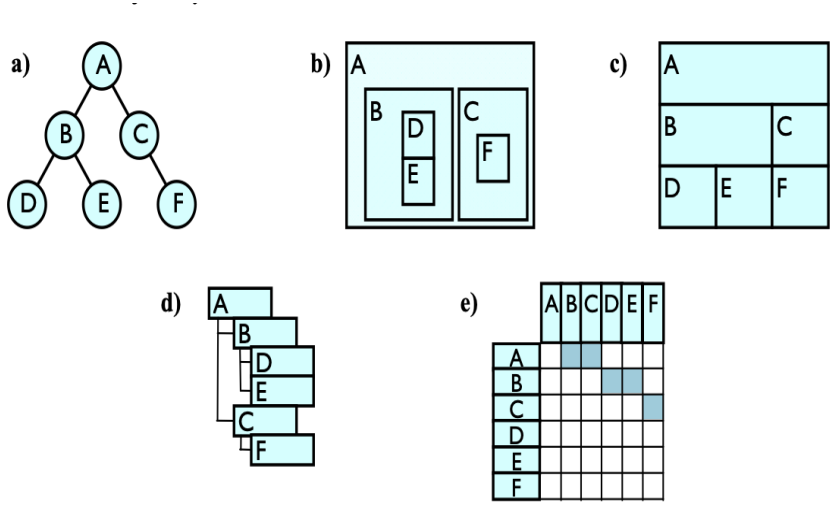


Figure: S4

Disadvantages of Single Tree

- **Node-Link Representations**: More understandable but use up screen space rapidly.
- **Nested Representations**: Display more nodes, but difficulty in perceiving structure due to lacking a global child-parent orientation and emphasis on leaf nodes.
- **Adjacency and Indented List Methods**: Utilize more screen space than node-link displays but maintain relative simplicity in the following structure.
- **Matrix Representations**: Reduce the tree to a look-up table, potentially limiting the understanding of hierarchical structure.
- **Combined Styles (Zhao et al.)**: Adaptability based on screen space and user interaction but may increase complexity.

Multiple Tree Model

- The experiment shows how complex the multiple Tree congregation process is.
- This may sound algorithmic but bringing that to the picture is a tough job. The visualization and thinking it needs.

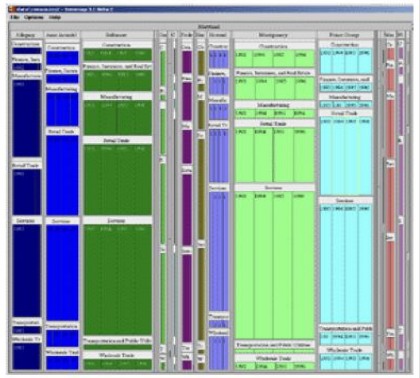
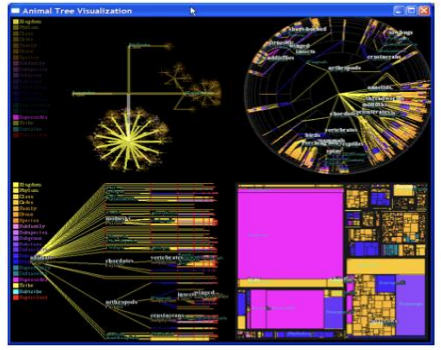
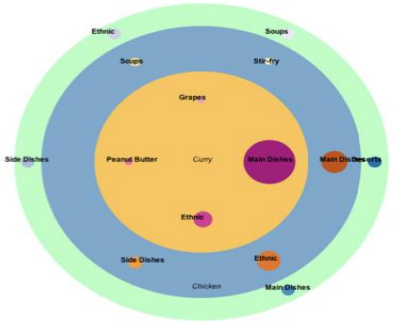
		Number of Tree Representations	
		1 Tree	Multiple Trees
Number of Trees in Structure	1 Tree	Traditional single tree visualisations (e.g. Cone Trees[37], TreeMaps[33])  **Original TreeMap - (screenshot courtesy of Human-Computer Interaction Laboratory, University of Maryland, College Park)**	Multiple/multiform views of a single tree structure (e.g. Wilson & Bergeron[46], Teoh[49])  **Teoh's Multiple Views of a tree data set**
	Multiple Trees	Single hierarchical view of navigation through faceted hierarchies (e.g. MoireTrees[51])  **MoireTrees visualisation**	Full multiple tree visualisation (e.g. Graham & Kennedy[57], Munzner *et al*[58], Sifer[8])  **Graham & Kennedy's multiple taxonomy visualisation**

Figure: S5

Two Tree Approach

a) Linking two spatially separate tree representations.

(Also known as Edge Drawing)

a) Shared coloring between two spatially separate tree representations.

b) Animation between trees (temporally separate).

c) Matrix comparison of two trees.

d) Spatial agglomeration of two tree structures.

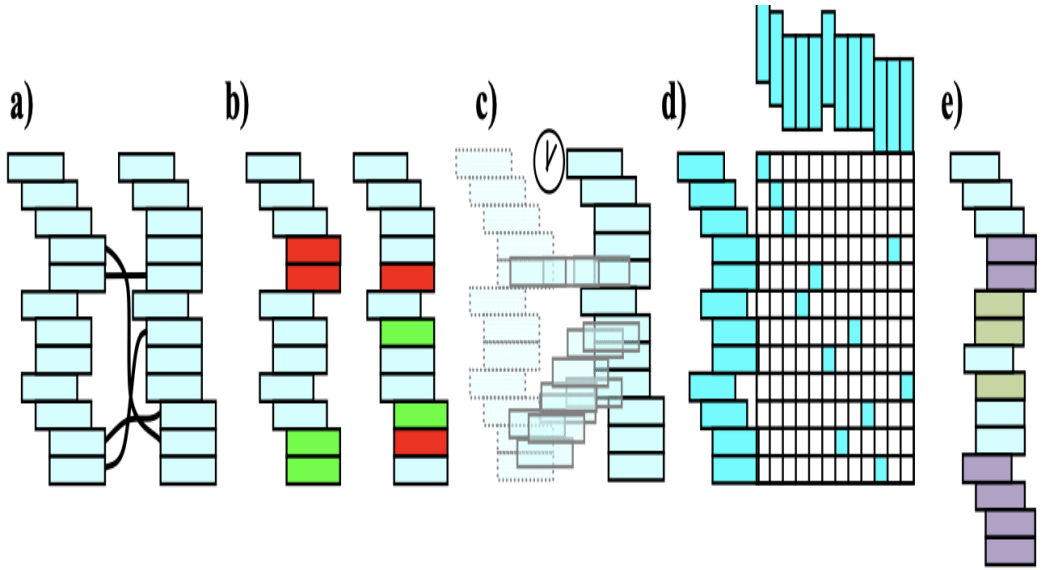
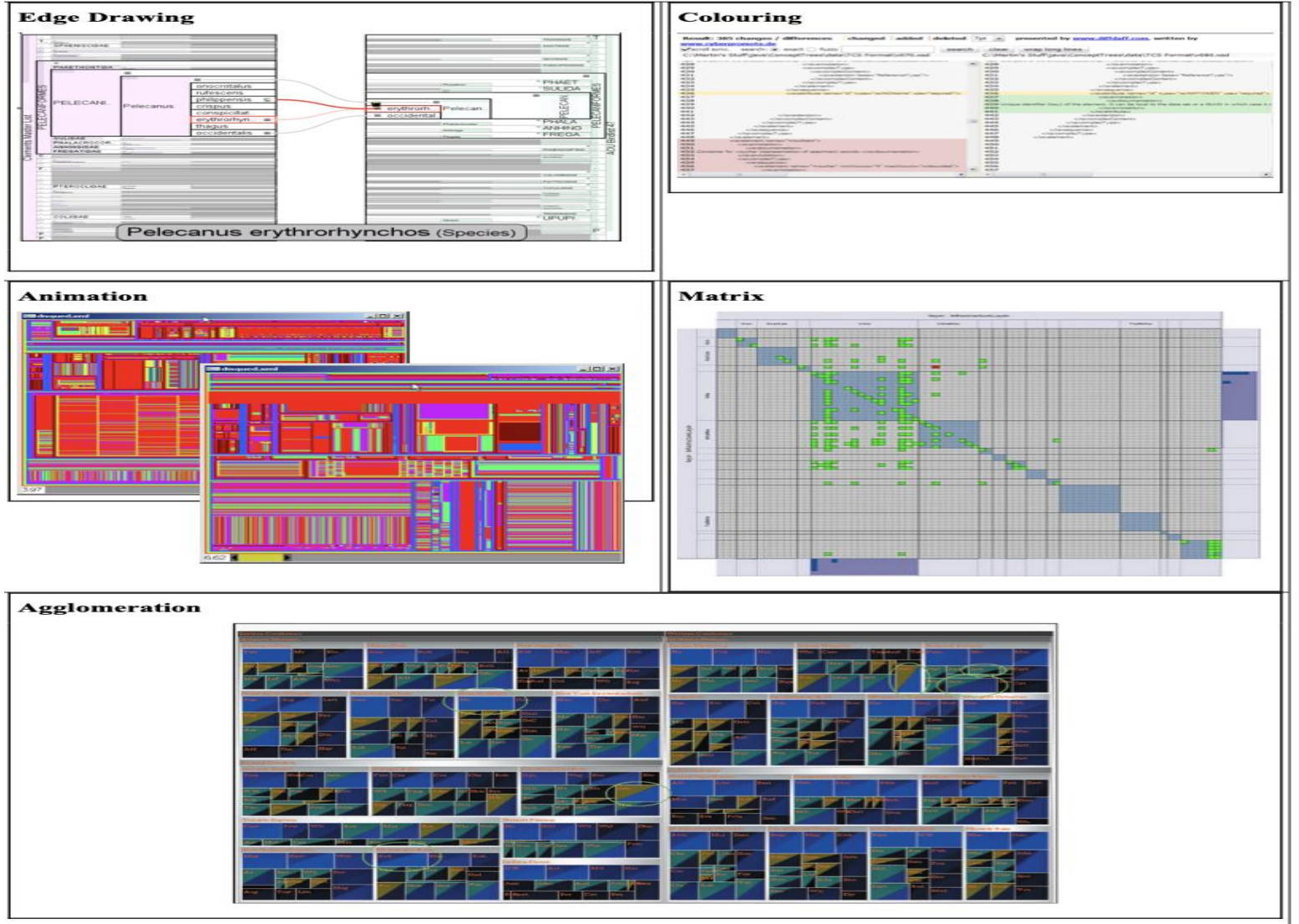


Figure: S6

Visualization of Two Tree



F[redacted]. Screenshots of visualisations that show comparisons of two trees. Linking – Craig and Kennedy's[61] Concept Relationship Editor, Colouring – DiffDaff[83] file utility comparing two XML files, Animation – Ghoniem & Fekete's[78] animated treemaps, Matrix – van Ham's code matrix[82] , Agglomeration - Tu & Shen's Union Tree[84].

Figure: S7

Multiple Tree

- Edge Drawing:
  - Involves displaying multiple individual representations on-screen, subdividing available screen space into areas for individual trees.
  - Often utilizes a small multiples technique to draw edges between multiple tree representations, resembling Parallel Coordinates with hierarchical axes.
- Coloring
  - As the number of trees increases, relationships in small multiples are often shown through color or highlighting techniques.
  - Beyond two trees, the dominant approach for relationship display is through color, addressing issues of edge crossings and the efficient use of screen space in comparison to edge drawing.

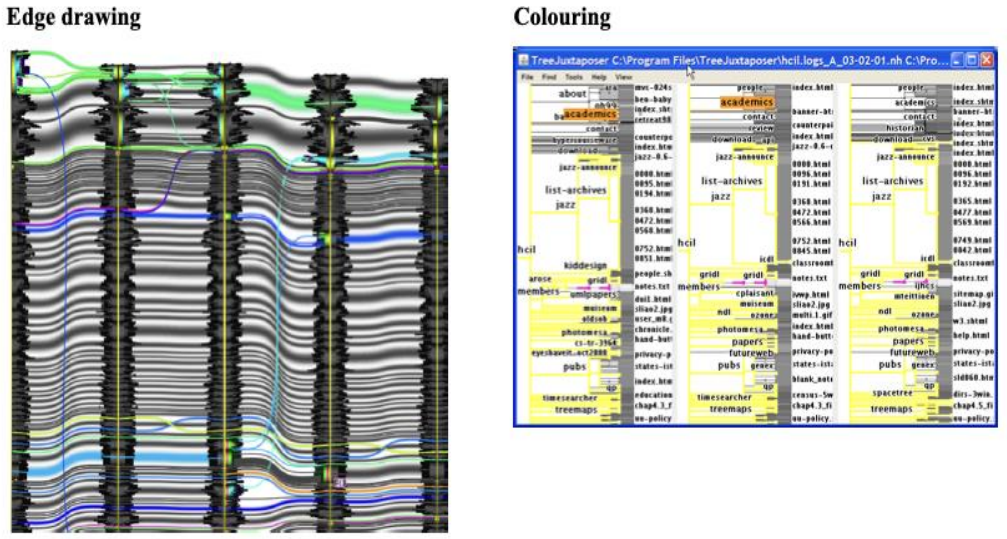


Figure: S8

Visualization of Two Tree

- Animation:
  - Applied to multiple trees, animation has limitations due to human perceptual abilities, as it can only display changes between two trees at a time.
  - TimeTree and Latour tree drawing system utilize animation to show incremental changes and recognize that animation is more suited for displaying structural evolution than complete reorganizations.
- Matrix-Based Visualization:
  - Matrix visualization of multiple trees faces challenges as a matrix has only 2 axes, making it complex to map individual trees.
  - Options include extending the scatterplot technique to use trees for comparison, forming a matrix of matrices, or using the union tree as a hierarchy in one matrix.

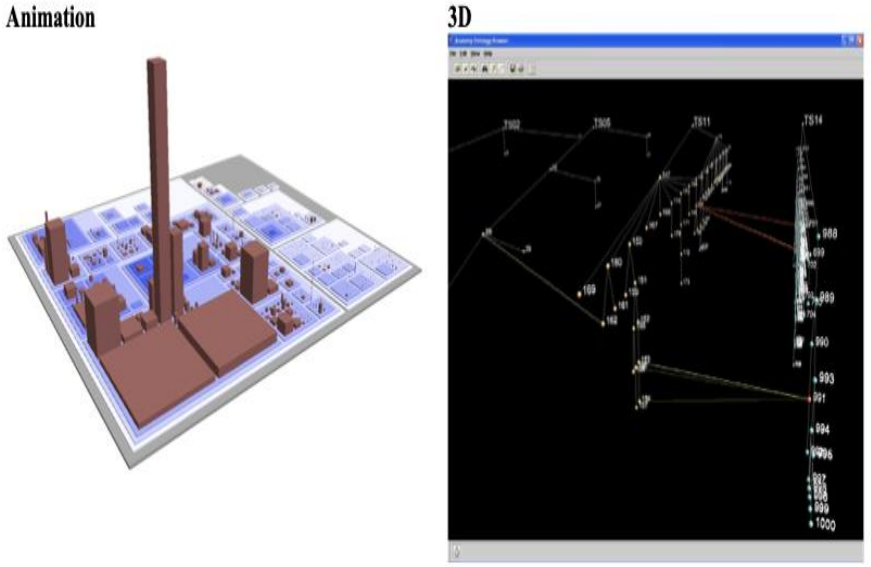


Figure: S9

Visualization of Two Tree

- Agglomeration of Multiple Trees:
  - In agglomeration, nodes can have multiple parents, and options include replicating nodes across trees for tree-like structure or representing a truer cycle-containing graph using node-link representations.
- **Atomic Representations:**
  - Developed for managing extremely large numbers of trees where finite screen space can't display all trees in detail.
  - Parts of the graph and, from there, individual trees or a consensus tree of a tree group can then be interrogated to display further details and to reveal related phylogenies.

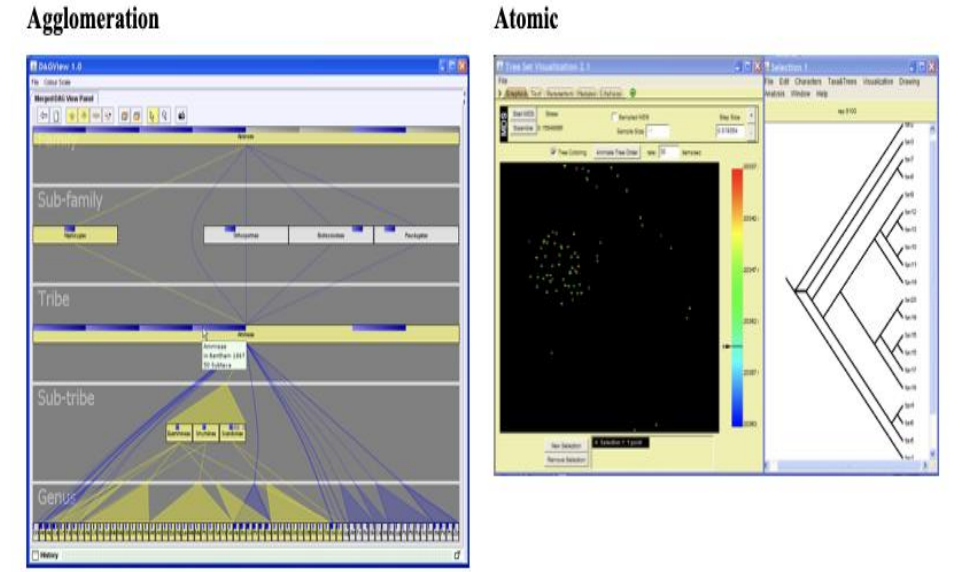
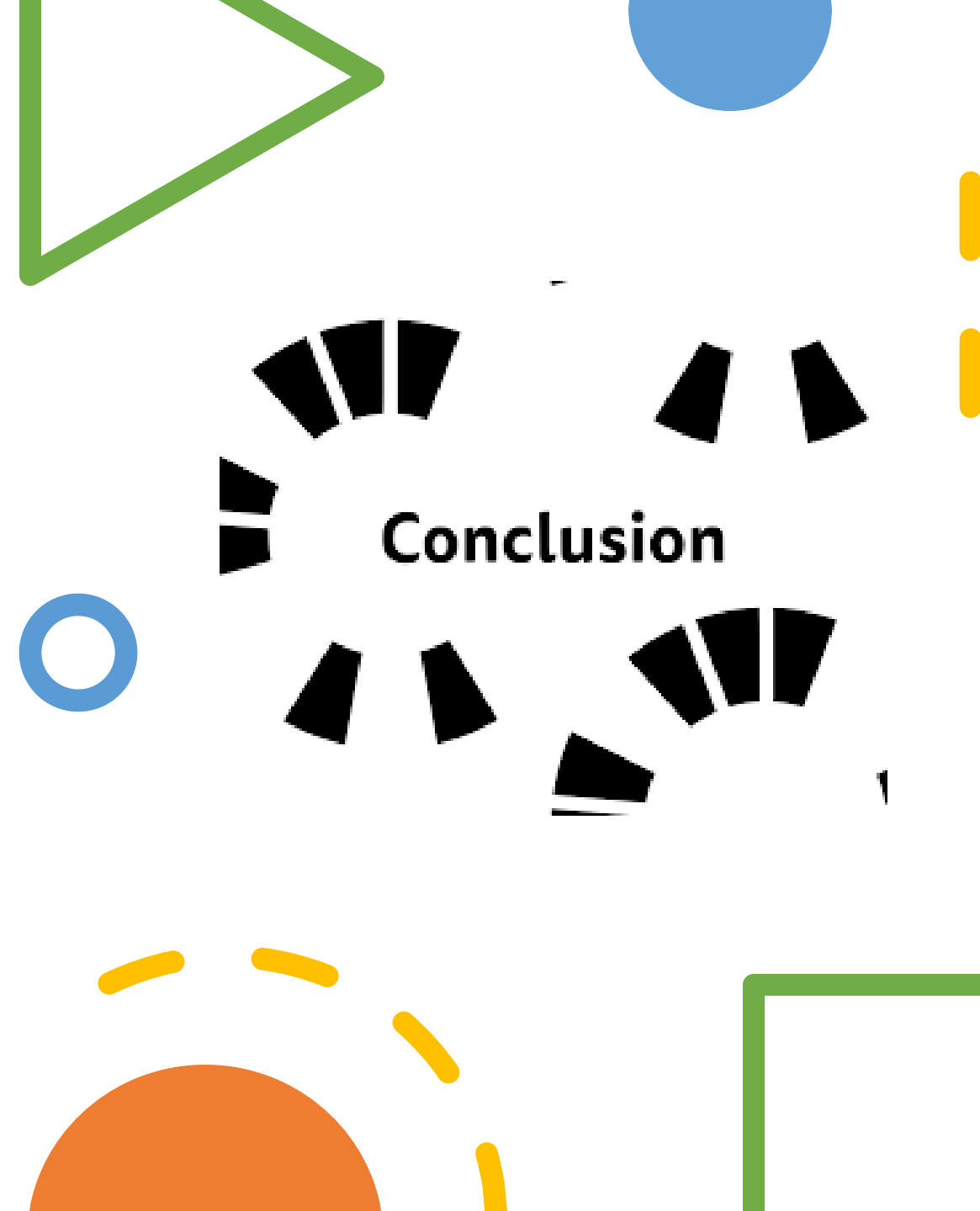


Figure: S10

Conclusion

- Multiple tree visualization is an ongoing research focus, with continuous exploration of novel techniques for single trees and complexities arising when visualizing multiple hierarchies.
- The choice of layout and interaction techniques is influenced by the inter-relationships between hierarchies, ranging from no overlap to complex structures, impacting the visualization design.
- Task considerations guide representation choices, with an emphasis on pair-wise mapping between individual tree representations and minimizing visual complexity during navigation.
- Existing visualizations lack a comprehensive exploration of layout design space, with a notable absence of matrix-style layouts, and conclusive user studies for multiple tree visualization remain scarce.



Check Out

Tree Visualization:
https://treevis.net/

GitHub Code Source:
https://github.com/tusharnayan10/tree-visualization.git

Thank You

Any Questions?