UNIT 4

UNIT IV: VISUALIZATION OF GROUPS:
Visualization of groups, trees, graphs, clusters, networks, software, Metaphorical visualization. Various visualization techniques, data structures used in data visualization.

Visualization of groups:

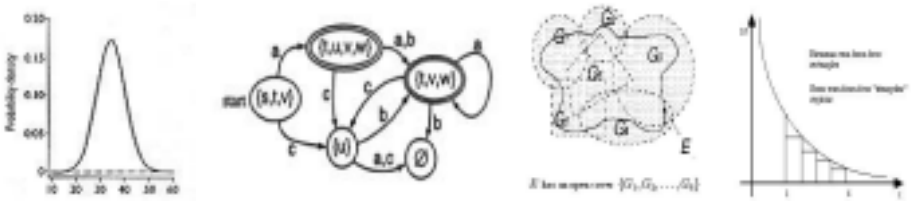
Clustering and visualization are key issues in exploratory data analysis and are fundamental principles of many unsupervised learning schemes. For a given data set, the aim of any clustering approach is to extract a description of the inherent group structure. The object space is partitioned into groups where each partition is as homogeneous as possible and two partitions are maximally heterogeneous. For several reasons it is useful to deal with probabilistic partitioning approaches:

1. The data generation process itself might be stochastic, resulting in overlapping partitions. Thus, a probabilistic group description is adequate and provides additional information about the inter-cluster relations.

2. The number of clusters might be chosen too large. Forcing the algorithm to a hard clustering solution creates artificial structure not supported by the data. On the other hand, superfluous clusters can be identified by a probabilistic group description .

3. There exists theoretical and empirical evidence that probabilistic assignments avoid over-fitting phenomena

Most branches of mathematics involve some sort of pictures; calculus has continuous curves, statistics has probability density functions, topology has surfaces, set theory has Venn diagrams.



Group theory tends to have very few pictures, and when it does (e.g. wallpaper groups or polyhedra) they serve only to exemplify a few select groups, and indirectly. Thus it is possible for a student to have a year or more of abstract algebra without ever picturing the subject in his or her head! This is a loss for any group theory student, and is a particular roadblock to visual learners.

Group Explorer provides advanced, interactive visualization techniques for group theory. It is particularly designed to be an aid for building intuition and understanding for students as they learn, but those seasoned in algebra may find themselves seeing the subject in a new light as well. I will briefly explain what Group Explorer can do.

- Multiplication tables (or Cayley tables) are a common introduction point for new students into binary operations and groups. Group Explorer allows exploration with these not before possible--not just viewing them, but highlighting them, comparing them, drawing homomorphisms between them, taking quotients of them, and more.
- Cayley diagrams, a little used but highly potent group visualization tool, are the flagship intuition-building device used in Group Explorer. Better than any other technique, Cayley diagrams truly expose the structure of a group and the relationships between its elements and

generators. These, too, can be used like multiplication tables: highlighting, homomorphisms, etc.

- Other visualization techniques (symmetry objects--very common, and cycle graphs--very rare) are also integrated throughout Group Explorer. See an example object of symmetry or see an example cycle graph.
- Group Explorer's elaborate help and ideas allow the student to be guided by the software on their explorations and investigations in group theory. Group Explorer's group library is the perfect place to start for building conjectures or finding counter examples.

Visualization of Graphs and Trees:

Graph Drawing The primary concern of graph drawing is the spatial arrangement of nodes and links Often (but not always) the goal is to effectively depictthe graph structure:
• Connectivity patterns
• Partitions / Clusters
• Outliers

Graph visualization
- node-link diagrams
- matrices

Graphs
• Describe relations among data items
• Using nodes and edges

a tree is a connected graph with no cycles
a directed tree is a digraph
(directed graph) whose
underlying graph is a tree

– a directed tree consists of a
number of nodes and parent
child relationships

– every node has just one parent
and any number of children
– directed trees are the most
common form in computer science
Directed acyclic graph / connected graph with n-1 edges • Nodes have one parent & 0-N children

Trees:

A tree is defined as a set of nodes and edges. It can otherwise be defined as a network of connected nodes where there are no loops. Every edge has a pair of nodes called the
parent node and child node.

1.A child node has only one parent node.
2.The root node which is a single node has no parents and the leaf nodes have no children.
3.Between any two nodes there exists a unique path.
4.The depth of a tree is the number of nodes from the root to the leaf.TRADITIONAL TREE DATA REPRESENTATIONS:

There are three types of tree data representations they are

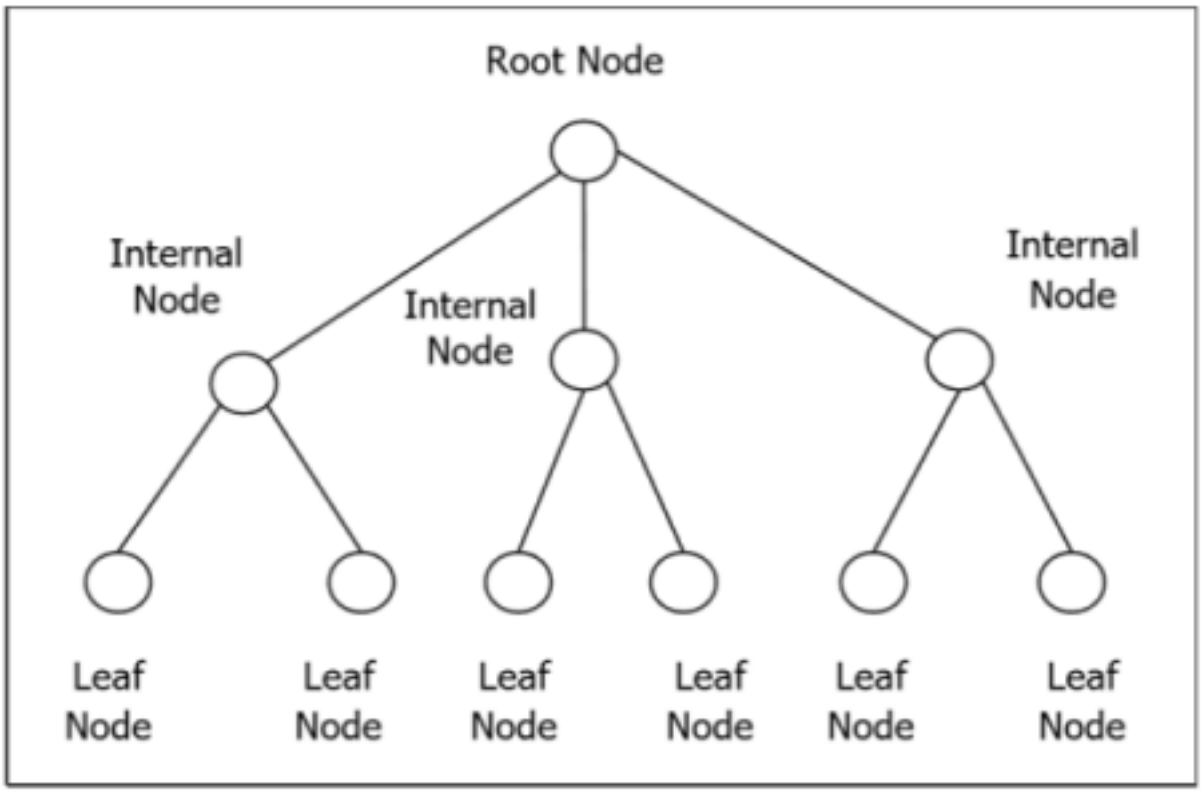
1.Classical Tree layouts
2.Balloon view: related to 3-d cone tree

3.H-tree layout: best for balanced trees

The classical tree layout shows the hierarchy of nodes clearly
but compromises with the screen space.
The balloon, H-tree and radial view layouts have difficulty in finding the root but they efficiently use the screen space.

Visualizing Trees

The rooted trees, it has a root of a tree, have many techniques to visualize the data. In this post, I will introduce three techniques. Maybe, you already know the some of techniques.



Rooted Tree

Node-Link Diagrams

This technique just uses a rooted tree itself with design goals. Common design goals:

- Nodes at the same depth share the same vertical position
- Horizontal whitespace communicates hierarchy ●

Minimize the required area

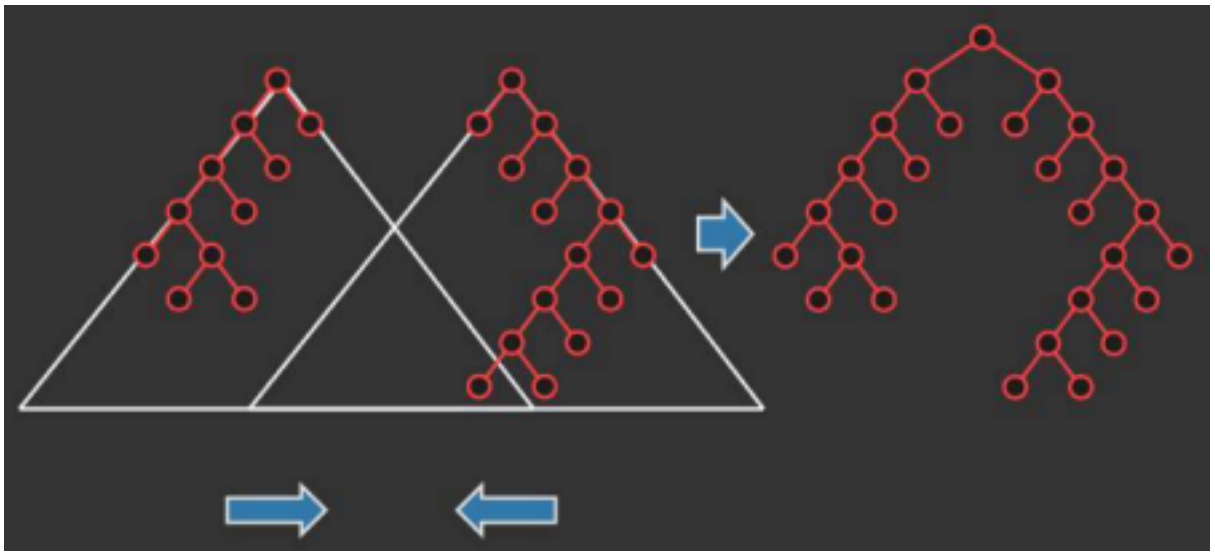
- Minimize the total length of edges
- Achieve a good aspect ratio

Reingold-Tilford

Reingold and Tilford,[Reingold/Tilford 1981], formulate how to draw the binary trees for visualization. Its rule can be generalized.

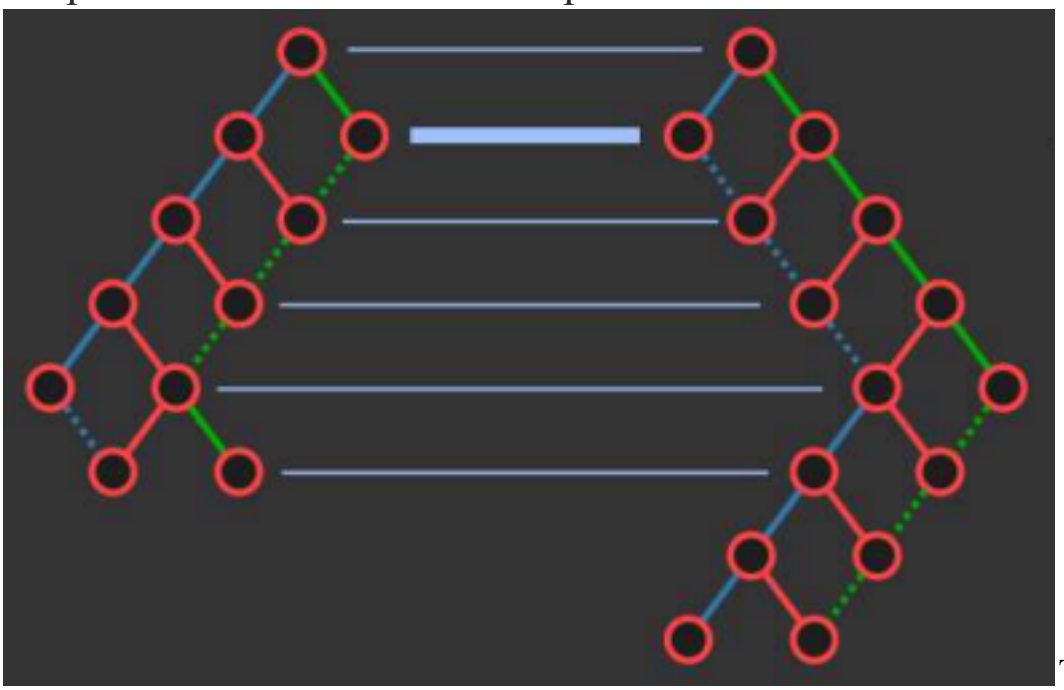
Aesthetic Goals:

- Nodes at the same level should be aligned
- Maintain the relative ordering of left and right subtrees
- The parent should be centered over the children
- A tree and its mirror image should be drawn as reflections of each other
- A subtree should be drawn the same way regardless of where it occurs in the tree



Recursive Construction example[Image from Pat Hanrahan]

The process recursively constructs the subtree and finds the rightmost point in the left subtree and the leftmost point in the right subtree. It defines the distance, users can choose, between the points and make the center point of it to connect the subtrees.

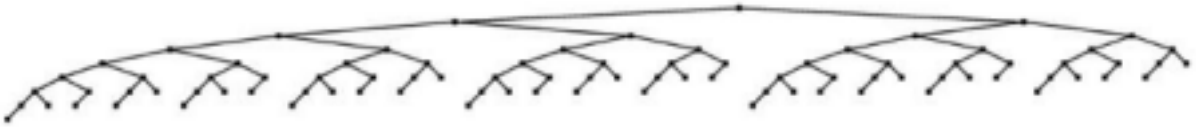


Thread definition [Image from Pat Hanrahan]

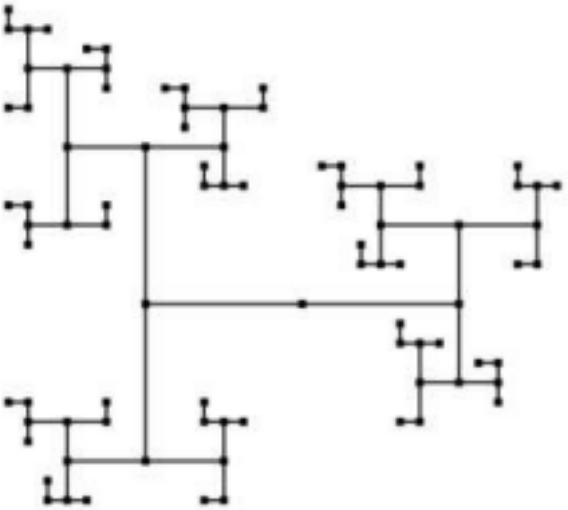
If we are trying to find the rightmost point and the leftmost point, we need to traverse the whole graph every time. Thus, we define the thread, the connection of the contour for left and right. You

can check the dotted line in the above picture. Thread is defined when there are no children even if it is right or left contour, it just hops to the next depth.

Pros and Cons



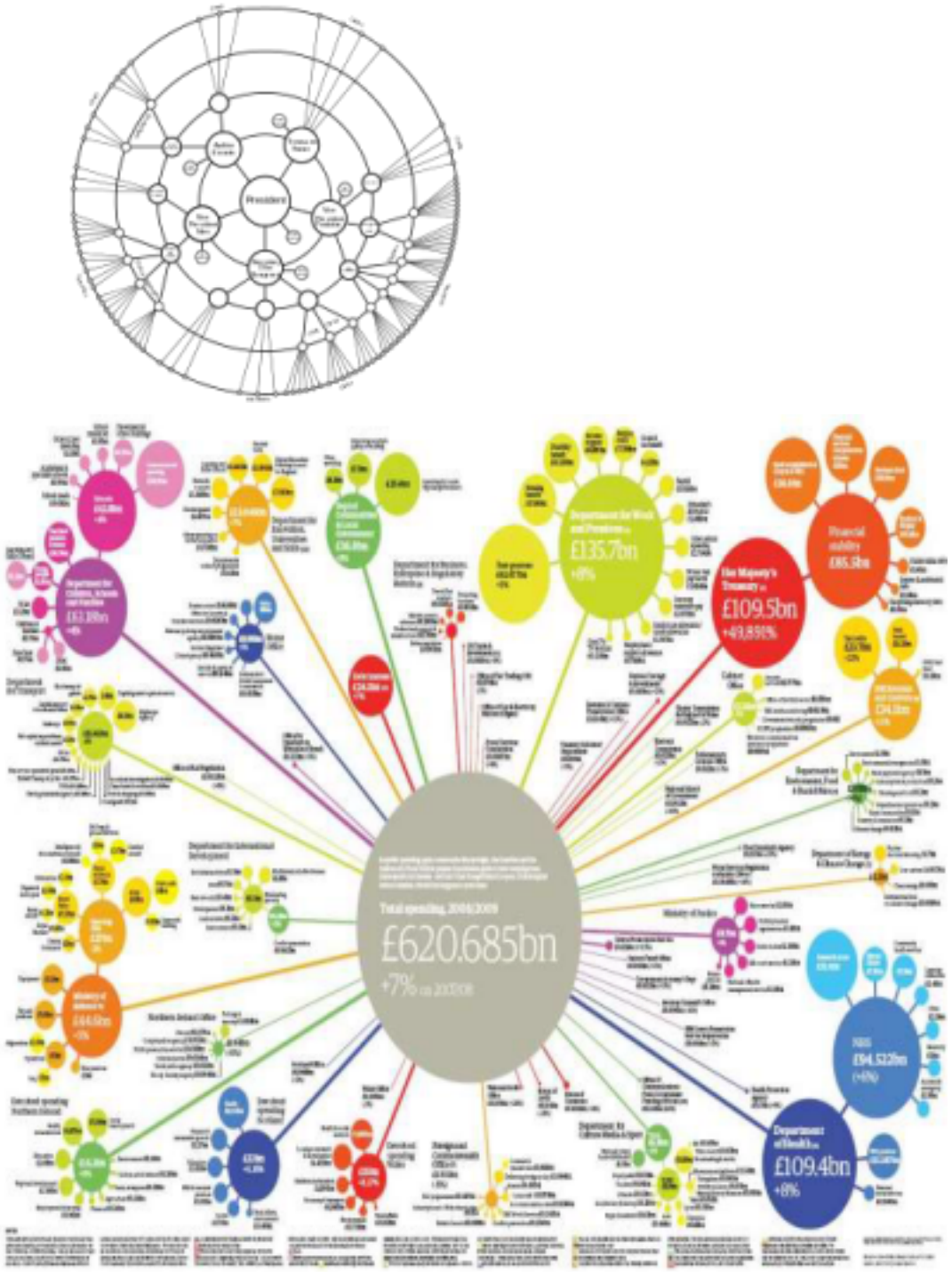
Images from Adrian Rusu



The non-level layout of the same tree Images from Adrian Rusu

It is easy to understand and implement but there is a significant drawback, it can lead to poor aspect ratios. To solve this problem, there are many alternatives. The above picture shows you one of the alternatives, it gives up width and hierarchy.

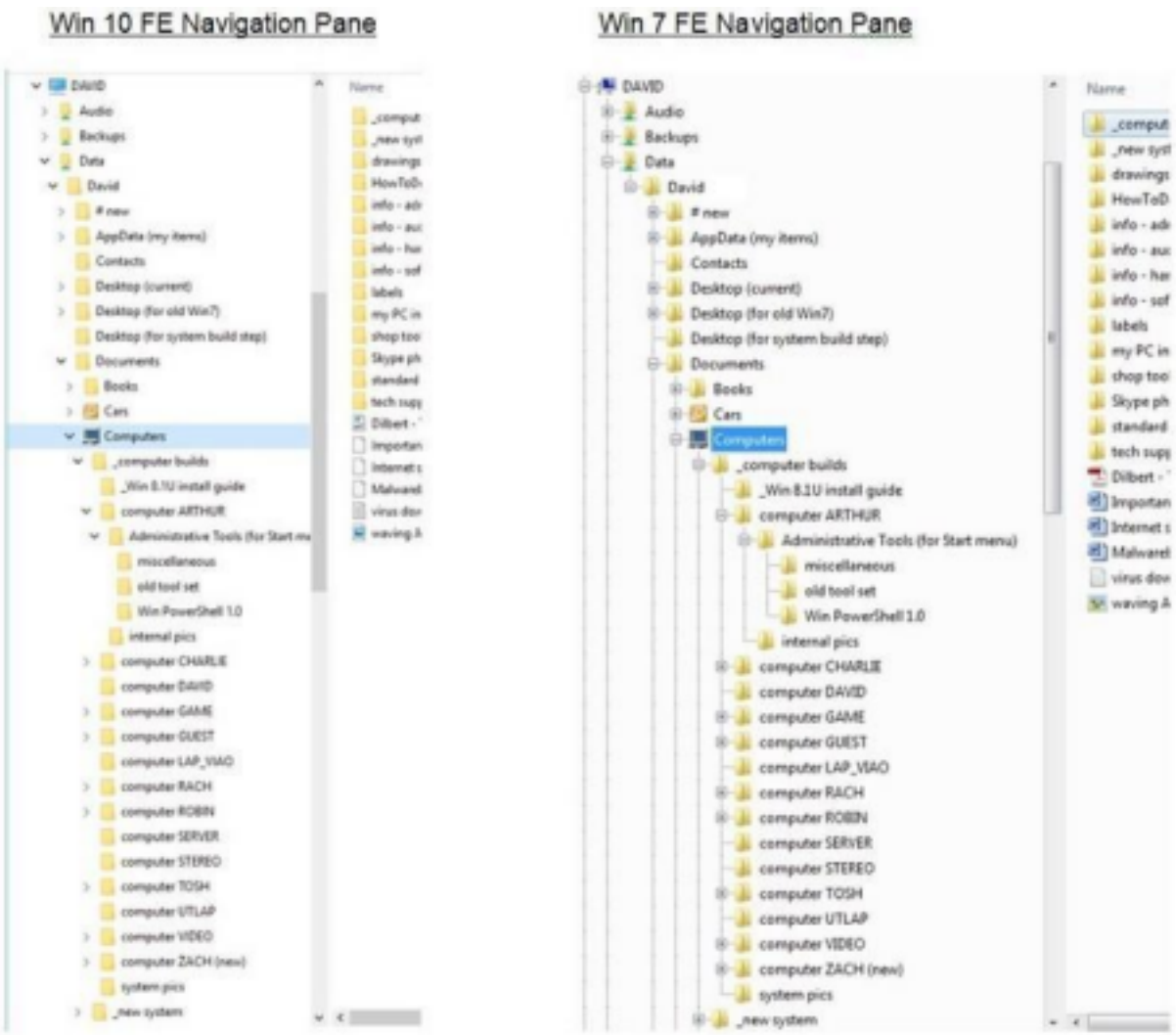
Radial Layout & Bubble Tree Layout



Radial Layout and Bubble Tree Layout

You can check other methods like these.

Indentation



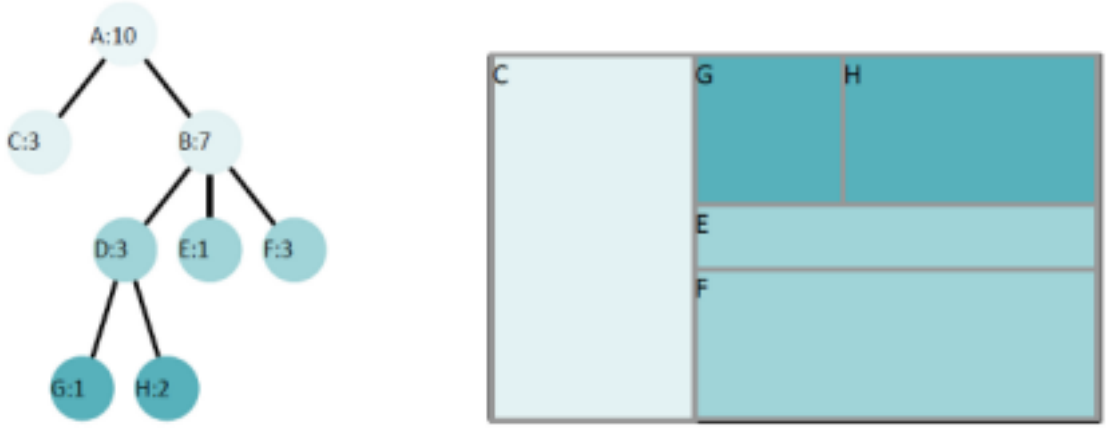
Window visualization of the data

Indentation is a well-known visualization method. You can open your file manager and check it. It places all items along vertically spaced rows. Indentation used to show parent/child relationships. The drawback is that it needs a great deal of scrolling to represent the files.

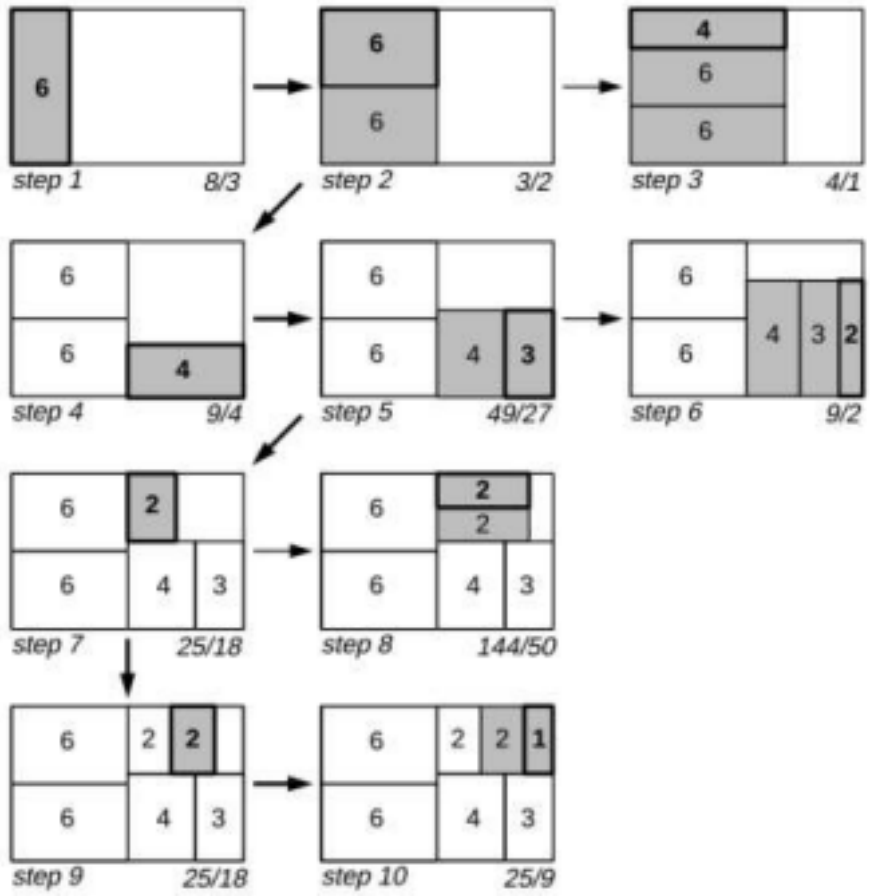
Treemaps

Treemaps

To solve the problem of indentation, treemaps encode structure using a spatial enclosure. It provides a single view of the entire tree and it is easy to tell the size of the node, encode additional attributes. However, it is difficult to accurately read depth.

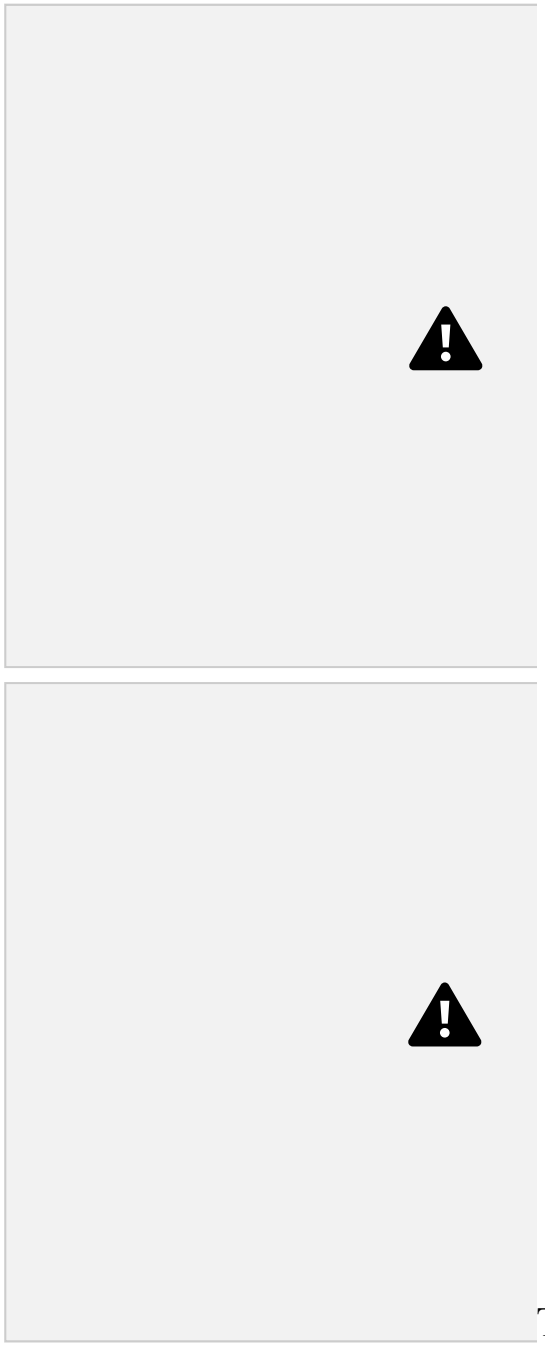


The basic algorithm of treemap recursively fills available space based on a size metric for nodes. The naive application of this algorithm leads to poor aspect ratios.



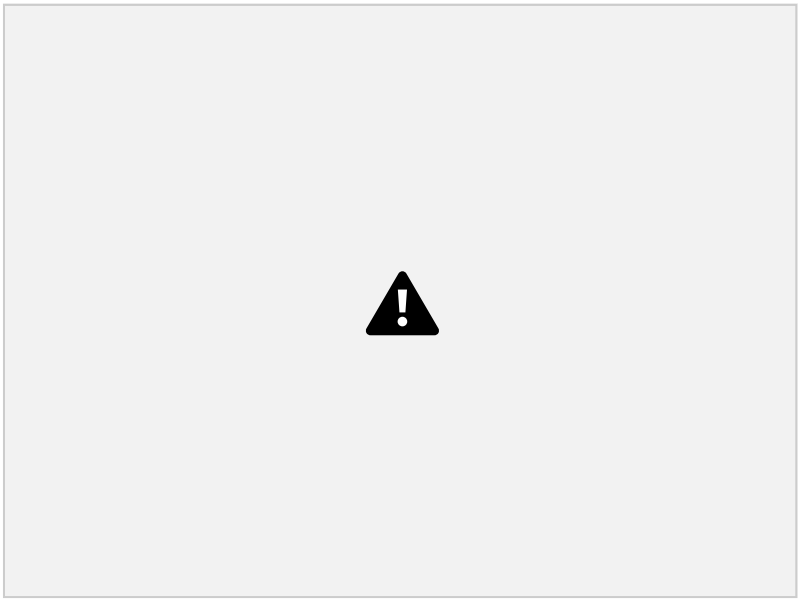
Squarified treemaps

When we divide the space, we check the aspect ration every step and pick the best one by the greedy search, calculating every possible case is np problem.



Traditional Treemap, Squarified Treemap

You can check the difference but it becomes harder to perceive hierarchical structure.



Images from [van Wijk et al. 1999], [Bruls et al. 2000]

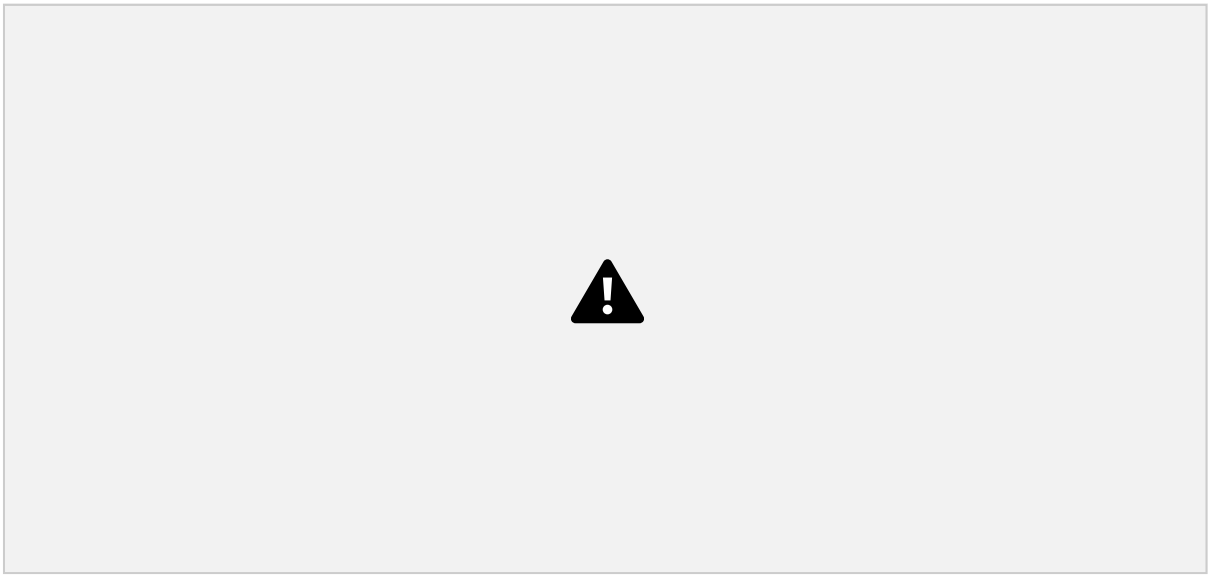
Cushion shading can solve this problem.

Various Visualization Techniques:

There are many data visualization types. The following are the commonly used data visualization charts.

1. Distribution plot

A distribution plot is used to visualize data distribution. Example: Probability distribution plot or density curve.



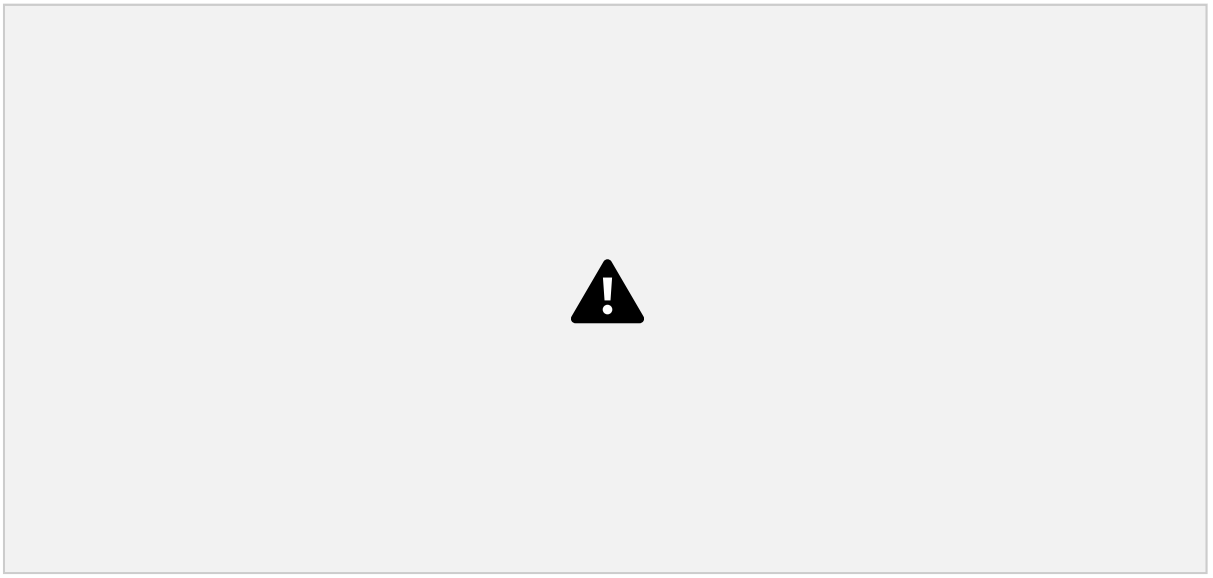
Source: seaborn.pydata.org

2. Box and whisker plot

This plot is used to plot the variation of the values of a numerical feature. You can get the values' minimum, maximum, median, lower and upper quartiles.

3. Violin plot

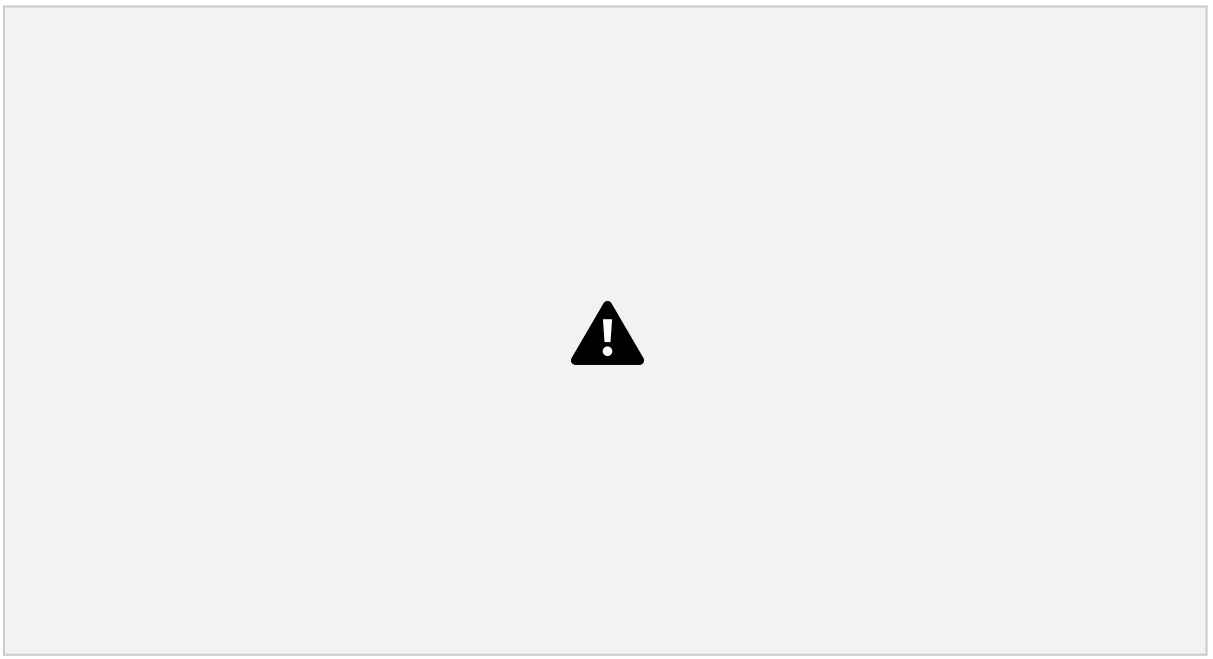
Similar to the box and whisker plot, the violin plot is used to plot the variation of a numerical feature. But it contains a kernel density curve in addition to the box plot. The kernel density curve estimates the underlying distribution of data.



Source: seaborn.pydata

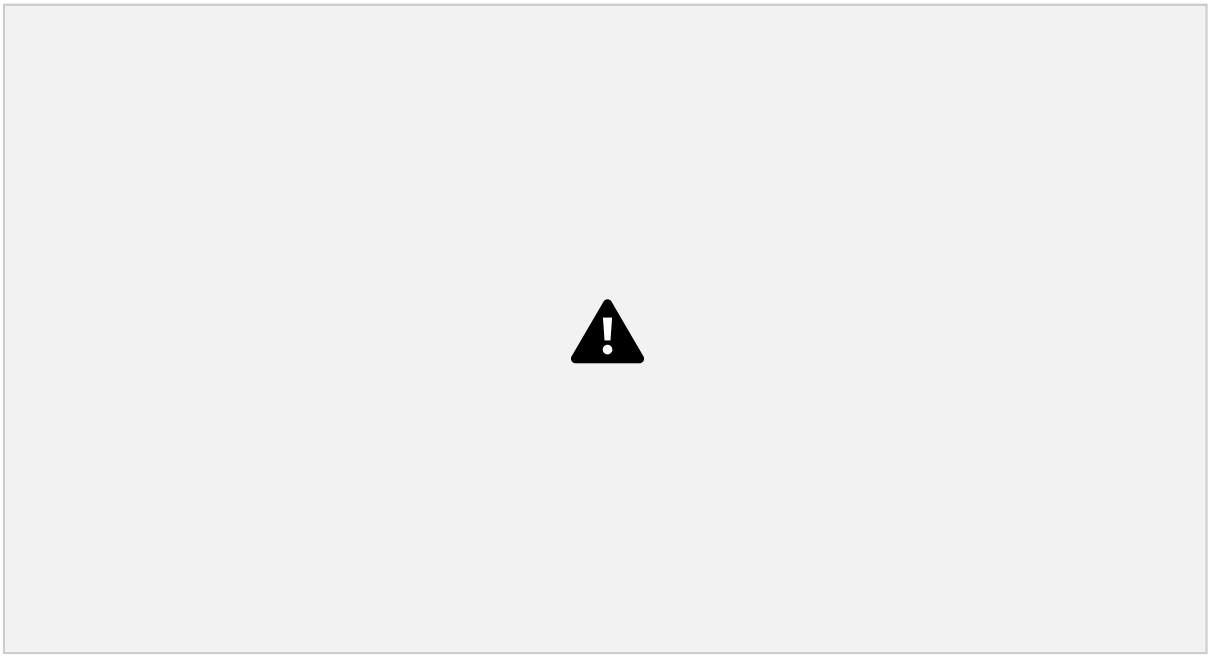
4. Line plot

A line plot is created by connecting a series of data points with straight lines. The number of periods is on the x-axis.



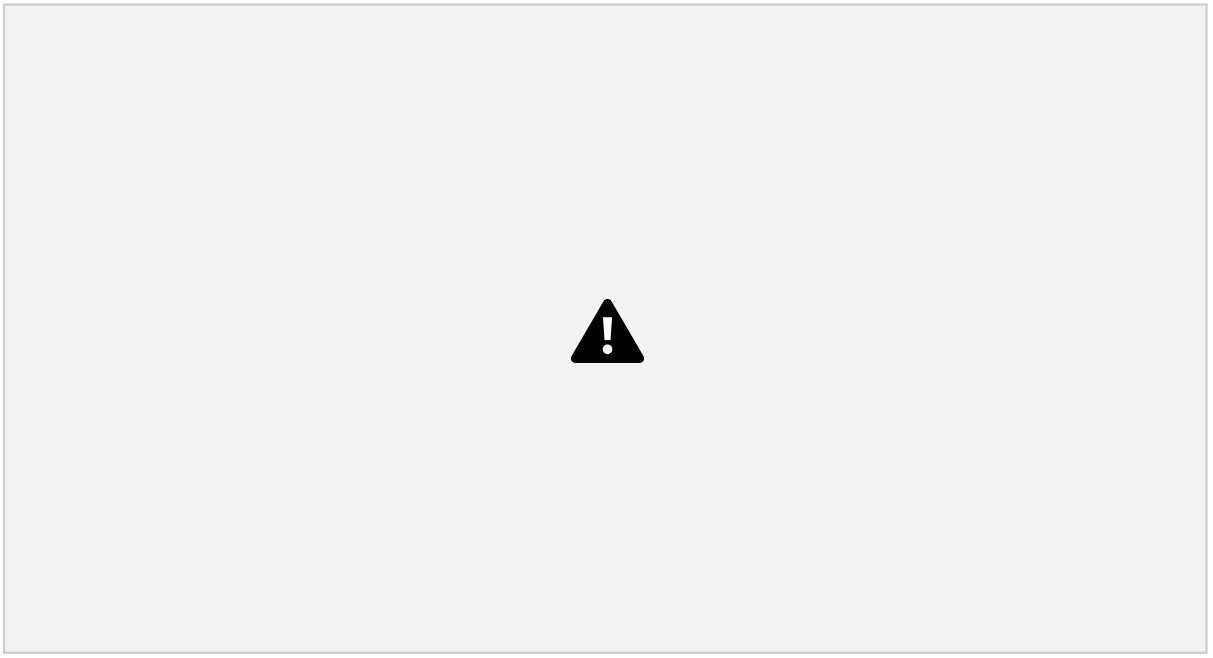
5. Bar plot

A bar plot is used to plot the frequency of occurring categorical data. Each category is represented by a bar. The bars can be created vertically or horizontally. Their heights or lengths are proportional to the values they represent.



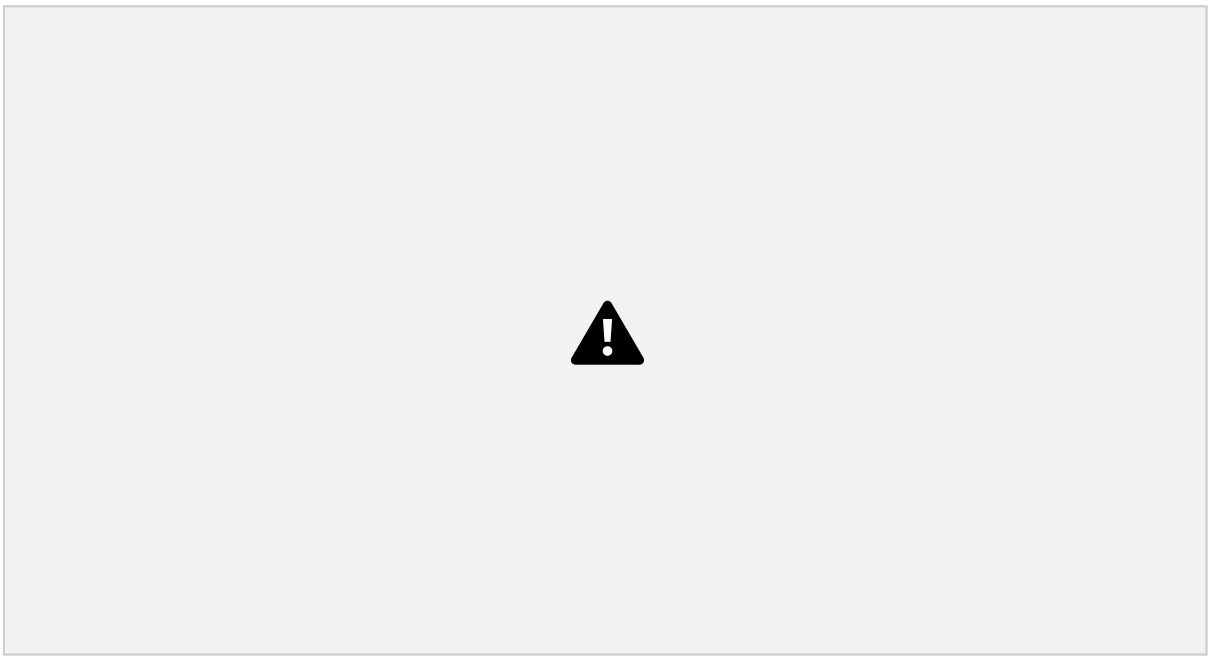
6. Scatter plot

Scatter plots are created to see whether there is a relationship (linear or non-linear and positive or negative) between two numerical variables. They are commonly used in regression analysis.



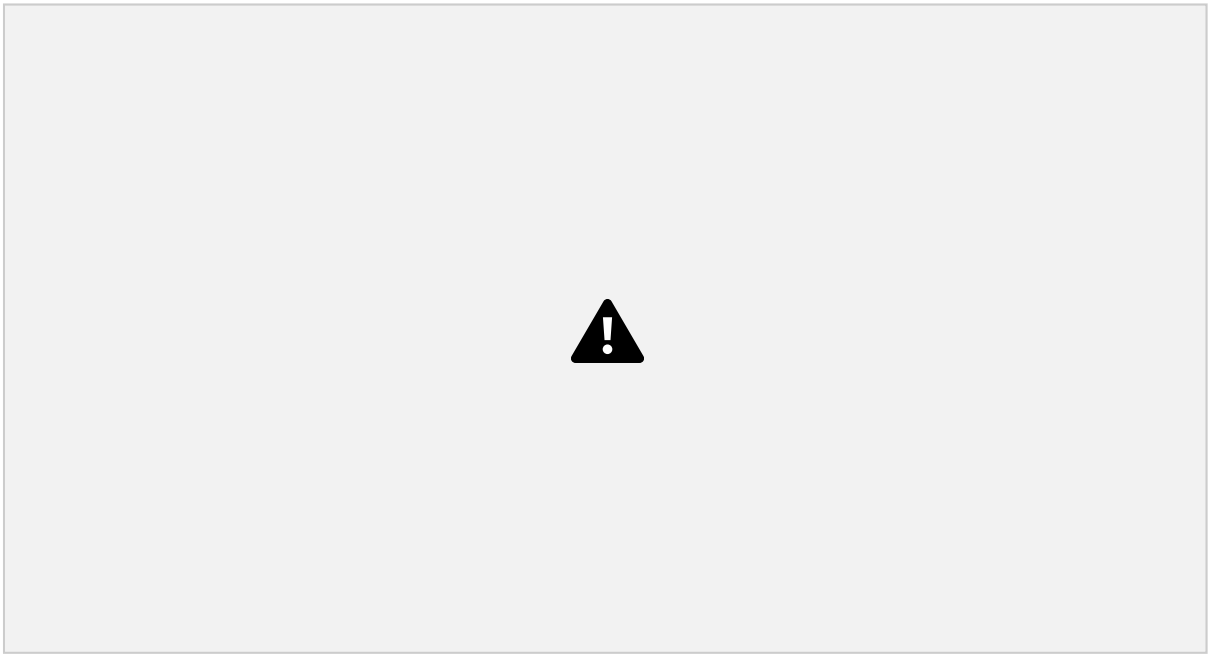
7. Histogram

A histogram represents the distribution of numerical data. Looking at a histogram, we can decide whether the values are normally distributed (a
bell-shaped curve), skewed to the right or skewed left. A histogram of residuals is useful to validate important assumptions in regression analysis.



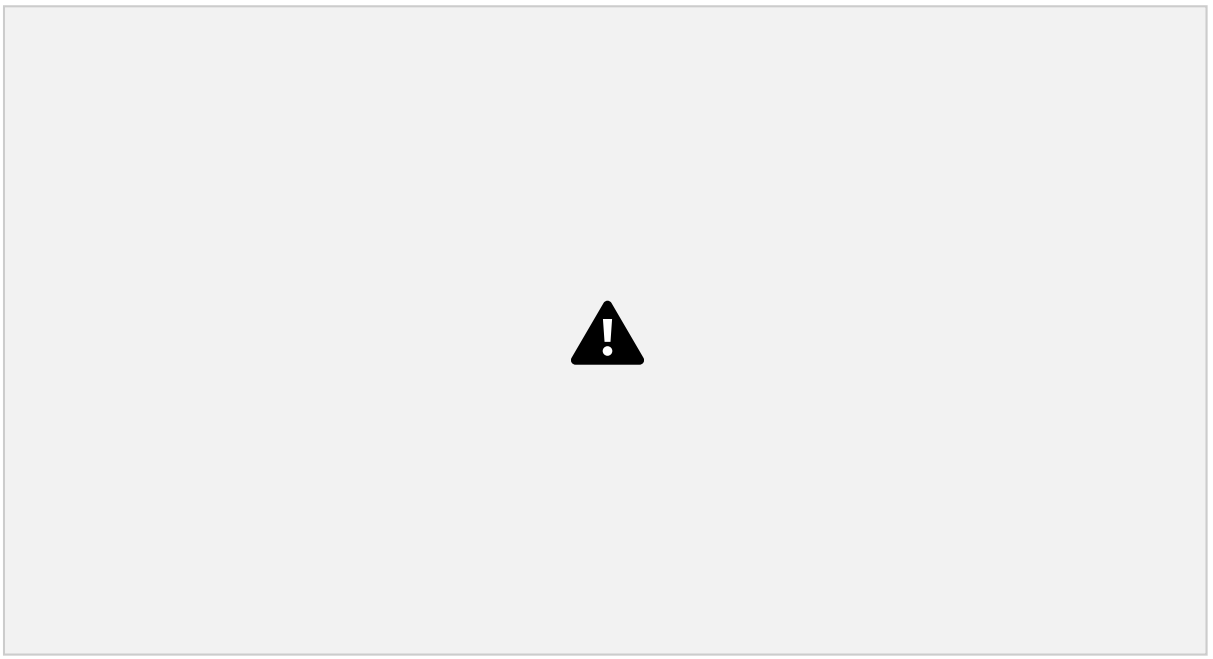
8. Pie chart

A categorical variable pie chart includes each category's values as slices whose sizes are proportional to the quantity they represent. It is a circular graph made with slices equal to the number of categories.



9. Area plot

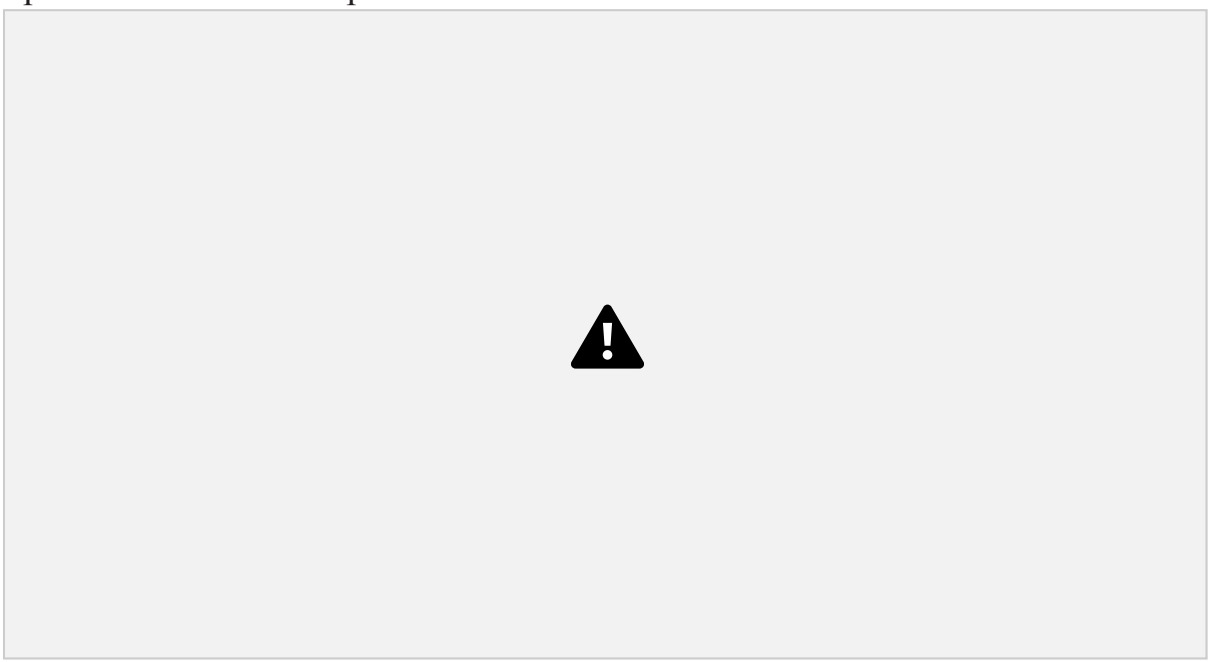
The area plot is based on the line chart. We get the area plot when we cover the area between the line and the x-axis.



Source: python-graph-gallery.com

10. Hexbin plot

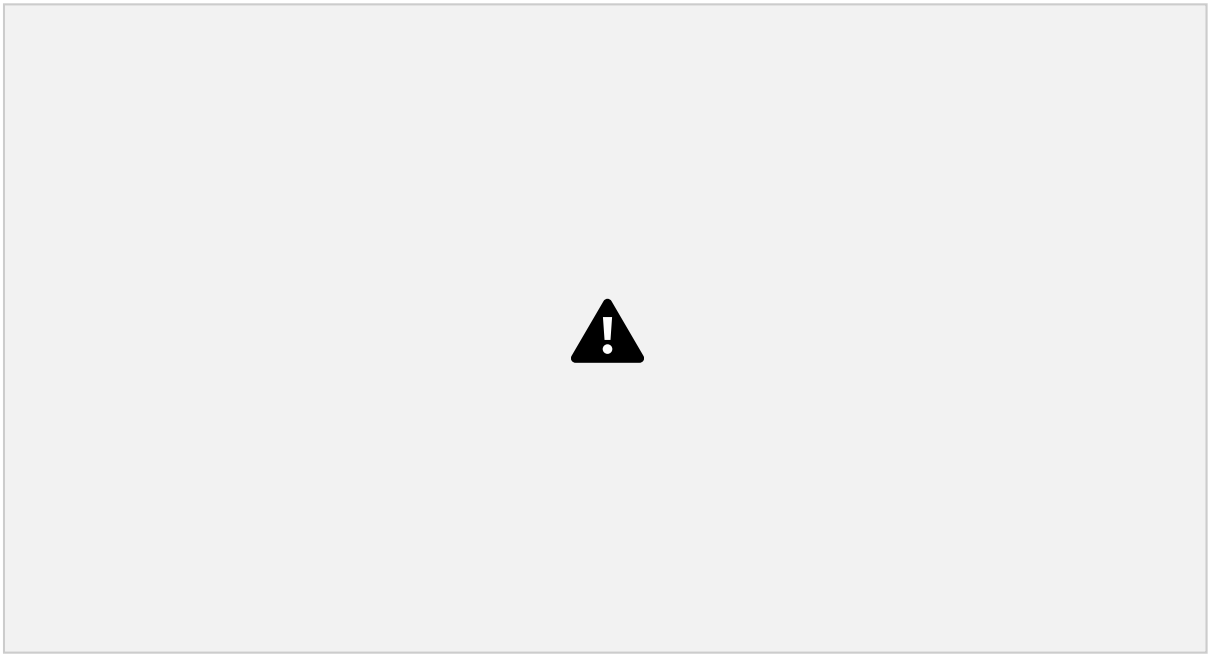
Similar to the scatter plot, a hexbin plot represents the relationship between two numerical variables. It is useful when there are a lot of data points in the two variables. When you have a lot of data points, they will overlap when represented in a scatter plot.



Source: python-graph-gallery.com

11. Heatmap

A heatmap visualizes the correlation coefficients of numerical features with a beautiful color map. Light colors show a high correlation, while dark colors show a low correlation. The heatmap is extremely useful for identifying multicollinearity that occurs when the input features are highly correlated with one or more of the other features in the dataset.



Do you want to be familiar with these plot types and many other things in data science? Enroll in Data Science Online Bootcamp.

Data Visualization Process/Workflow

The data visualization process or workflow includes the fowling key steps.

1. Develop your research question

This may be a business problem or any other related problem that could be solved with a data-driven approach. You should note all the objectives and outcomes plus required resources such as datasets, open-source software libraries, etc.

2. Get or create your data

The next step is collecting data. You can use existing datasets if they're relevant to your research question. Alternatively, you can download open-source datasets from the internet or do web scraping to collect data.

3. Clean your data

Real-world data are messy. So, you need to clean them before using them for visualization. You can identify missing values and outliers and treat them accordingly. You can perform feature selection and remove unnecessary features from the data. You can create a new set of features based on the original features.

4. Choose a chart type

The chart type depends on many factors. For example, it depends on the feature type (numerical or categorical). It also depends on the type of visualization you need. Let's say you have two numerical features. If you want to find their distributions, you can create two histograms for each feature. If you want to plot their variations, you can create box and whisker plots for each feature. You can create a scatterplot if you want to find a relationship (linear or non-linear, positive or negative) between the two features.

5. Choose your tool

You can use open-source data visualization tools such as matplotlib, seaborn, plotty and ggplot. You can also use API-based software such as Matlab, Minitab, SPSS, etc.

6. Prepare data

You can extract relevant features. You can do feature standardization if the values of the features are not on the same scale. You can apply data preprocessing steps such as PCA to reduce the dimensionality of the data. That will allow you to visualize high-dimensional data in 2D and 3D plots!

7. Create a chart

This is the final step. Here. You define the title and names for the axes. You should also choose a proper chart background to ensure the content is easily readable.

Cluster visualization

Cluster visualization renders your cluster data as an interactive map allowing you to see a quick overview of your cluster sets and quickly drill into each cluster set to view subclusters and conceptually-related clusters. This can assist you with the following actions:

1.Prioritising review – Use filters and metadata information to identify, tag, and batch documents that are likely to be relevant.

2.Exploring your data – Perform early assessment to learn about documents in your case and discover useful search terms.

3.Organizing review – Bucket and distribute documents by issue for a more efficient review process.

4.Performing Quality control – Ensure you didn't miss responsive documents by viewing conceptually similar documents.

5.Jump-starting Assisted Review - Locate good training examples for judgmental sampling in Assisted Review.

Visualizing a cluster:

Cluster visualization is integrated with the Documents tab, so you can add a cluster visualization widget directly to your Dashboard.

To view multiple cluster sets, you must create different dashboards. There can be only one cluster visualization widget on a dashboard at a time.

types of cluster visualisations

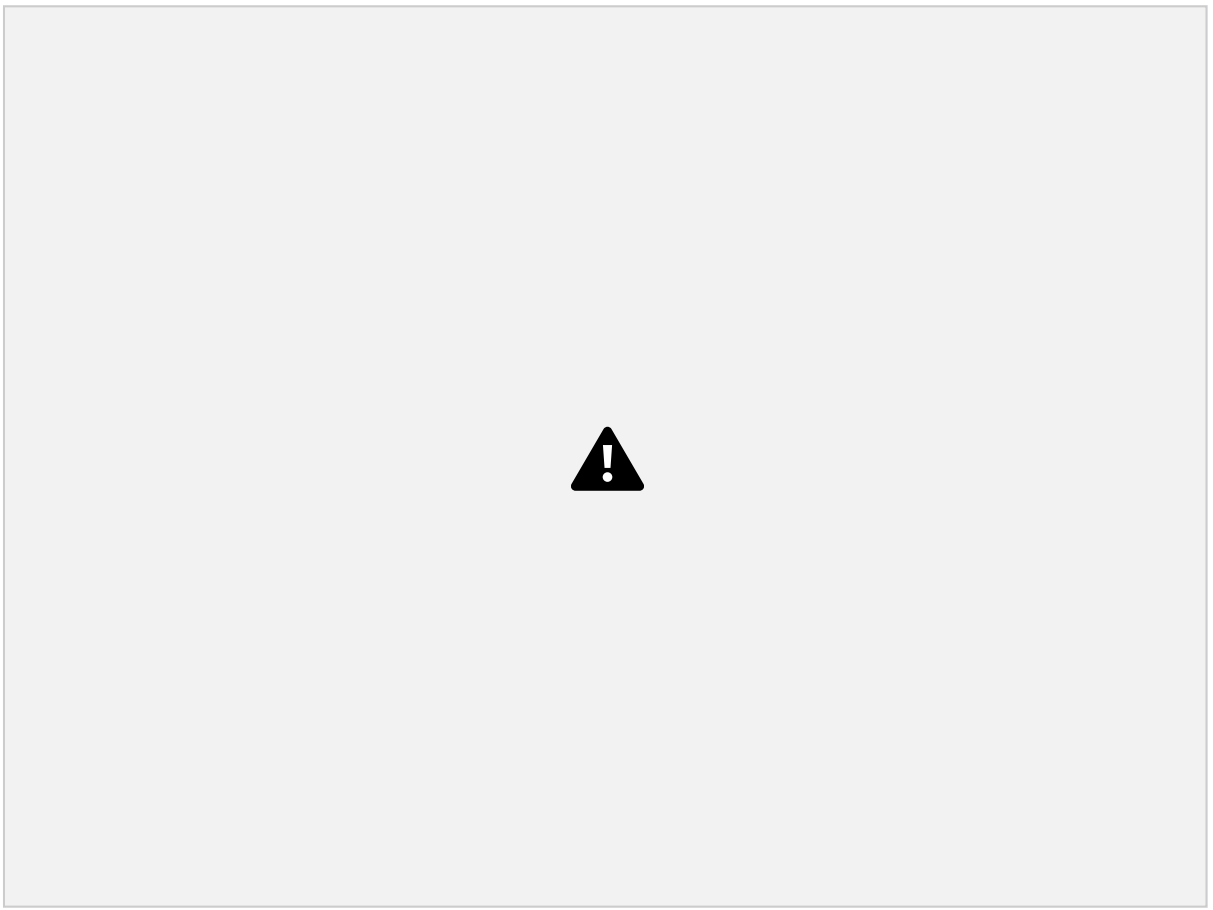
You can click and pan on cluster visualisations to move around the cluster, dial or circle pack visualization at any level of depth using any of the following cluster visualisations:

- **Circle pack**
- **Dial visualization**
- **Nearby clusters**

Circle pack:

The circle pack visualization arranges clusters in a circular pattern by order of the number of documents in each cluster, with the largest cluster representing the one that contains the greatest number of documents. To access

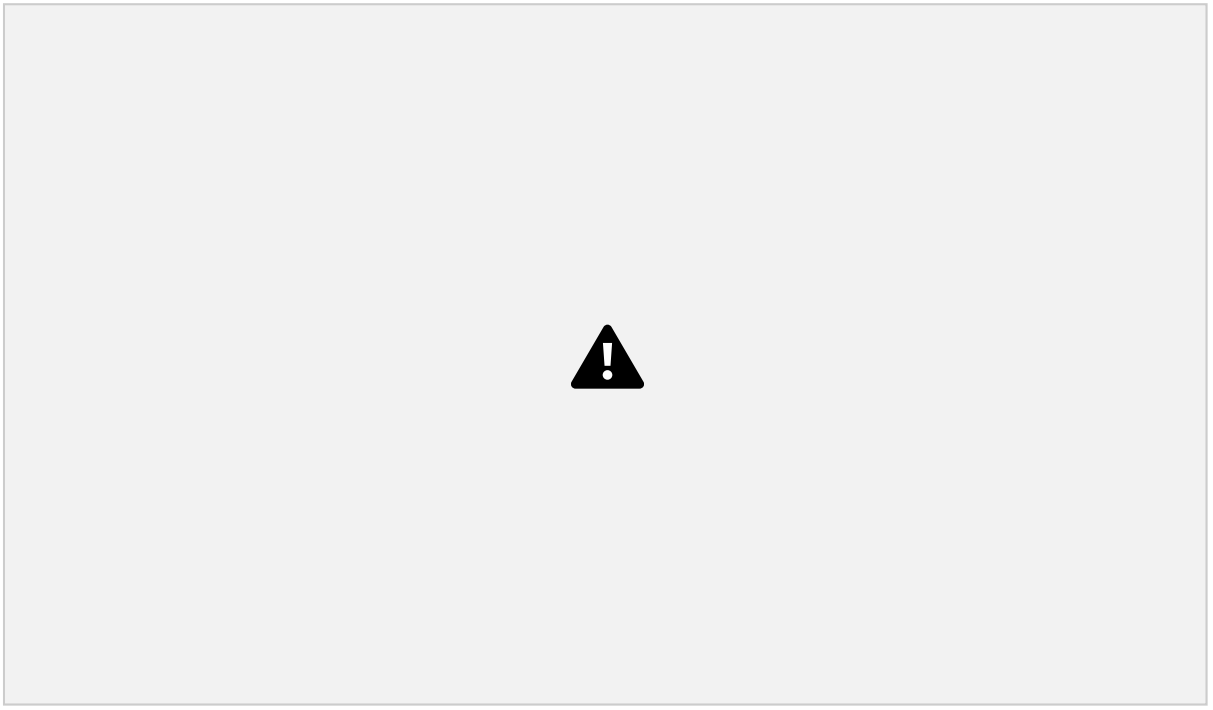
the circle pack, click on the hamburger icon in the top-right corner of the widget. Next, click **Circle Pack**.



Dial visualization

Cluster Visualization defaults to the dial visualization when you click **Visualize Cluster** on the cluster browser.

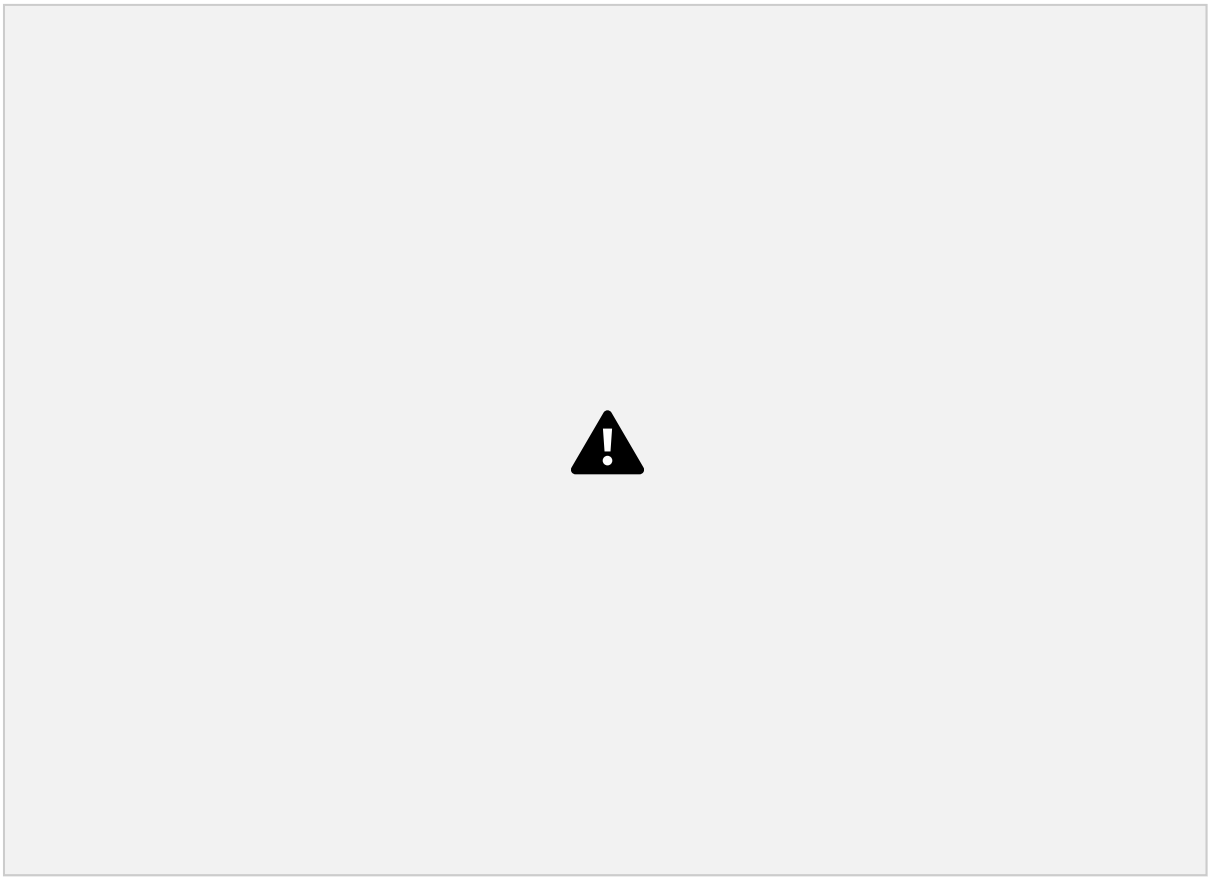
Dial visualization is a different representation of the circle pack. The visualization arranges documents in a circular pattern, with clusters containing the greatest number of documents on the inside. The dial's inner ring, or primary cluster, is equivalent to the top cluster in the cluster browser. The secondary, tertiary, and quaternary rings are child clusters of the primary cluster. Each segment shows up to 10 terms.



Nearby clusters

The nearby clusters visualization reveals clusters conceptually similar to a selected cluster. To open the nearby clusters visualization, right-click a cluster and click **View Nearby Clusters**.

The nearby clusters visualization arranges clusters based on conceptual similarity to a selected cluster. The cluster you selected is positioned in the centre with other clusters positioned according to the degree of similarity. The higher the similarity, the closer a cluster is positioned to the centre. The lower the similarity, the farther the cluster is positioned from the centre.



METAPHORICAL VISUALIZATION

The idea is to map the data to another dataset that is already familiar to the user, and then rely on their existing knowledge to illustrate relationships in the data. We construct the map by preserving pairwise distances or by maintaining relative values of specific data attributes. This metaphorical mapping is very flexible and allows us to adapt the visualization to its application and target audience. We present several examples where we map data to different domains and representations. This includes mapping data to cat images, encoding research interests with neural style transfer and representing movies as stars in the night sky. Overall, we find that although metaphors are not as accurate as the traditional techniques, they can help design engaging and personalised visualisations.