Unit 3

Non-spatial data visualization

Non-spatial data is such kind of data that does not have any geographical location associated with it i.e. it can be number, text or image and can be used to describe people, events and things. This type of dat provides context and additional information about the spatial features and can be in form of alphanumeric values, numbers dates and so on. Some example of non-spatial data are names, phone numbers, email address, genders, prices etc. non-spatial data can be used in a variety of ways such as to build machine learning model, to make prediction, to improve decision making, to conduct statistical analysis etc. the non-spatial data can be used with spatial data to provide more meaning to the spatial data. For eg map of landslide affected area and earthquake affected area to understand the relationship between the two.

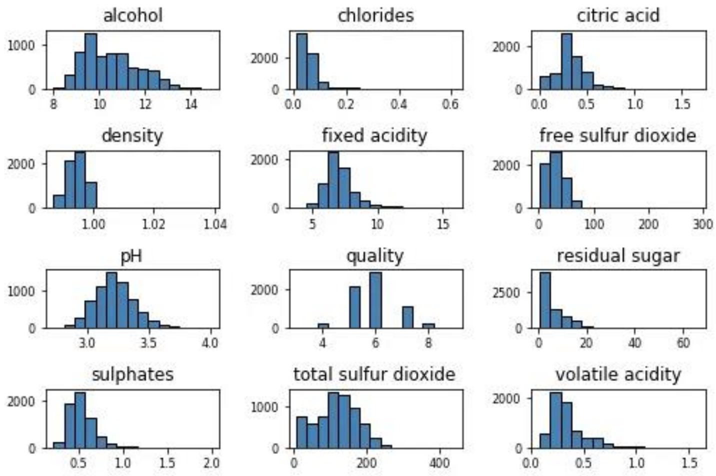
Non-spatial data visualization is the process of representing non-spatial data in a visual way to help user to understand in easy way. Visualization of non-spatial data can be done using charts and graphs (used to show trend, patterns and relationship of data), tables (used to show relationship between different variables in the data), heatmaps (uses color to show the distribution of data or highlight low or high values), tree maps (used to show the structure of a organization and relationship between different categories of data), scatterplots (uses points to represent the values of two variables in the data), word clouds (uses size of words to represent the frequency in the data and used to show the most important words in a document), bar charts (used to display categorical or discrete data using a bar and height of bar corresponds to the value of category), pie chart (use slice of pie to show composition of whole), line chart (used to display trend over time), box and whisker plot (used to show summary of statistics of a datasets including median, quartiles and potential outliers).

For effective non-spatial data visualization following terms should be considered:

- Use color and shapes that are easy to distinguish from each other
- Use a legend to explain the meaning of each color and shapes
- Use labels to identify the different parts of the visualization
- Use consistent style throughout the visualization.

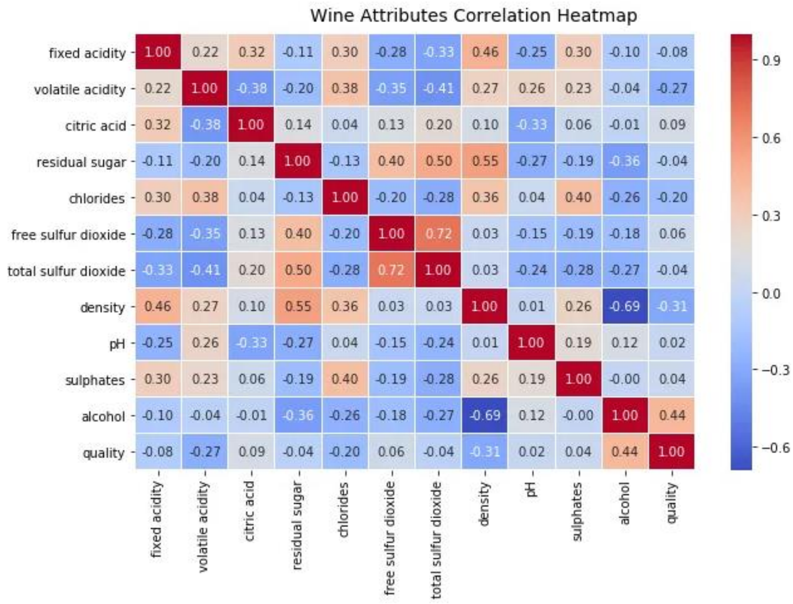
In data visualization, dimension refers to the number of variables that are being represented in data**.**

One dimensional data also known as univariate data that consists of a single variable or attribute for each data point and can be represented along single axis. For example, a list of numbers or a time series, daily temperatures for a specific location over a year etc. Visualization technique used for one dimensional data are histogram, bar charts, line charts, box plot.



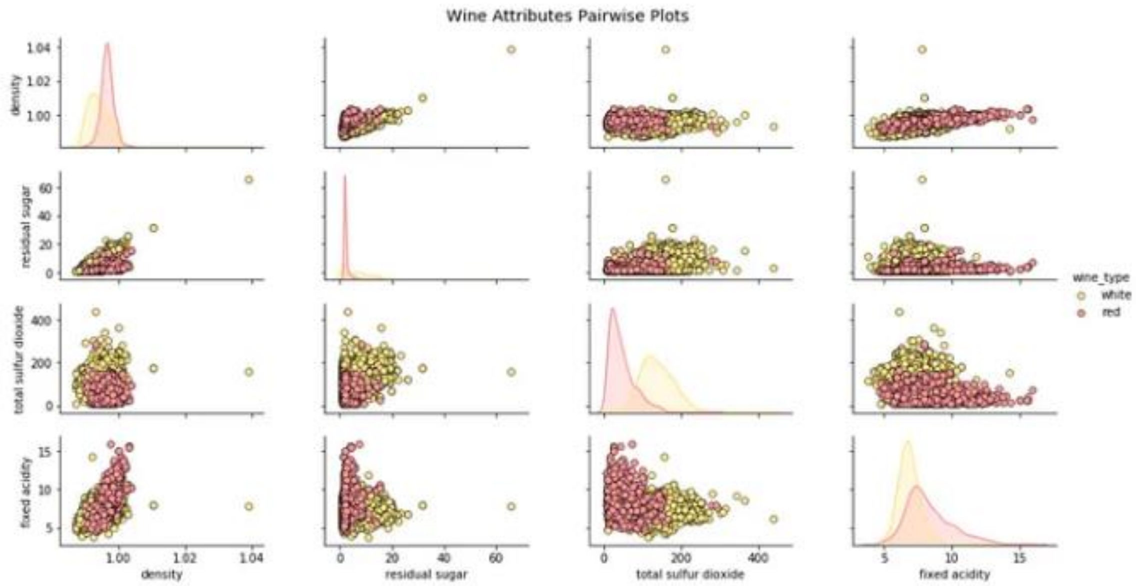
Visualizing attributes as one-dimensional data

Two-Dimensional data is such kind of data that involves two variable or attributes for each data points and each data points is represented by a pair of values. Two-dimensional data can be represented along two axes. For example: scatterplot showing height vs weight, heatmap showing correlation between data etc. some visualization technique used for two-dimensional data are scatter plots, bubble charts, heat map etc.



Visualizing two-dimensional data with a correlation heatmap

Multi-dimensional data also known as multivariate data are such kind of data that have more than two variables or attributes for each data point i.e. can be thought of as data with three or more dimensions and represented along three or more axes. For example, ecommerce datasets with variables like product category, price, customer rating and sales volume, climate data with variables such as temperature, humidity, wind, speed recorded at different locations and times. Some visualization technique used in multi-dimensional data are 3D scatter plots, ternary plots etc.



Visualizing three-dimensional data with scatter plots and **hue** (color)

Tabular data is such kind of data that is organized in form of row and column which are used in data analysis, reporting and decision making. Each row typically represents a single data entry or observation while each column represents a specific attribute or variables associated with that data. Some visualization technique used for tabular data are line charts, bar charts, heat map scatter plots and tree maps

For visualization of different dimension of data following factor should be consider:

- The number of dimensions of the data
- The goal of visualization
- The user for the visualization

Why you need data tables

Human eyes are drawn to patterns and blocks with specific padding. So data tables engage people with the cells, rows, and columns separation.

We can list lots of benefits of data visualization in tables.

- Organizing dynamic data types
- Data correlation is clear
- Rows and columns
- Merged table cells
- Sticky header or column
- Smart info distribution
- Multiple sets of related data
- Compare data
- Condense info

A data table, or a spreadsheet, is an efficient format for comparative data analysis on categorical objects. Usually, the items being compared are placed in a column, while the categorical objects are in the rows. The quantitative value is then placed at the intersection of the row and column, called the cell. The following examples demonstrate data tables.

	Total defects	A	B	C	D	E
A4636	131	37	21	28		45
A2524	86	20	24	21	1	20
A3713	75	17	13	18		27
A4452	73	5	33	17		18
A4088	72	14	16	12	2	28
A2103	68	14	13	14	1	26
A2156	68	16	13	19	2	18
A3681	66	12	16	9	1	28
A1366	50	11	15	12		12
A2610	39	5	7	12		15
Total	728	151	171	162	7	237

	Total defects	A	B	C	D	E
A4636	131	37	21	28		45
A2524	86	20	24	21	1	20
A3713	75	17	13	18		27
A4452	73	5	33	17		18
A4088	72	14	16	12	2	28
A2103	68	14	13	14	1	26
A2156	68	16	13	19	2	18
A3681	66	12	16	9	1	28
A1366	50	11	15	12		12
A2610	39	5	7	12		15
Total	728	151	171	162	7	237

Scatter plot:

It is a data visualization technique used to show the relationship between two continuous variables and are useful for identifying patterns, trends and outliers in the data. The data for each point is represented by its horizontal (x) and vertical (y) position on the visualization. It shows linear relationship (such relationship between two variable that can be represented by a straight line for example: height and weight of a group of people), non-linear relationship (such relationship between two variables that cannot be represented by a straight line. For example, price of stock over a time), correlation (used to measure strength of relationship between two variables by looking at how closely the points are clustered together) and outliers (data points that are far from each other. This can be shown by scatterplot by looking for a point that are much farther away from the rest of the data points). To create effective scatter plot flowing factor should be consider:

- Choose right variables
- Use appropriate labels and title to describe variables
- Use legend to describe each points, shape of points and colors
- Add annotation to highlight important features of the visualization

Creating scatter plot using python matplotlib

Step 1: import dataset: here we are using flower data set

Step 2: import matplotlib. Pyplot library by:
Import matplotlib.pyplot as plt

Step 3: use plt.scatter(x-axis, y-axis, hue, size)

Here, x-axis refers to data points placed in x-coordinate, y-axis refers to data points placed in y-coordinates, hue refers to the another data with which comparison is to be made based of x and y axis, size refers to size of figure

#importing libraries

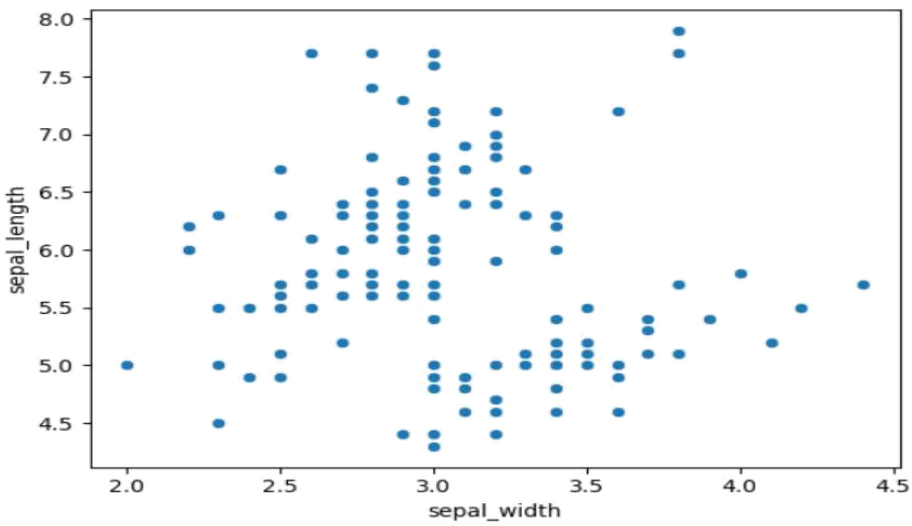
import matplotlib.pyplot as plt

import seaborn as sns

#importing default dataset

flowerData = sns.load_dataset("iris")

sns.scatterplot(x=flowerData.sepal_width,y=flowerData.sepal_length)

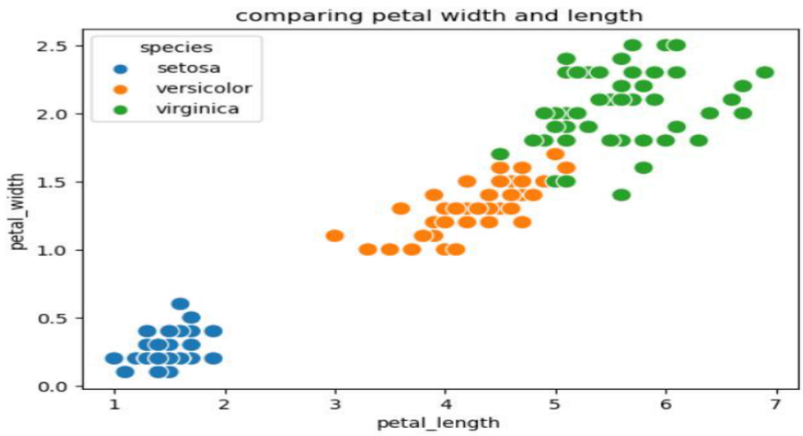


Here, comparison is done between column speal_widht and sepal_lenght to find out relationship between two variables. It is find out that flower with sepal length (6.0 to 6.5) and width(3.0 to 3.5) are more in numbers. Most of the points are clustered near to each other so there are no much outliers between these two variables.

Scatter plot for each species

plt.title("comparing petal width and length")

sns.scatterplot(data=flowerData,x="petal_length",y="petal_width",hue="species",s=100)



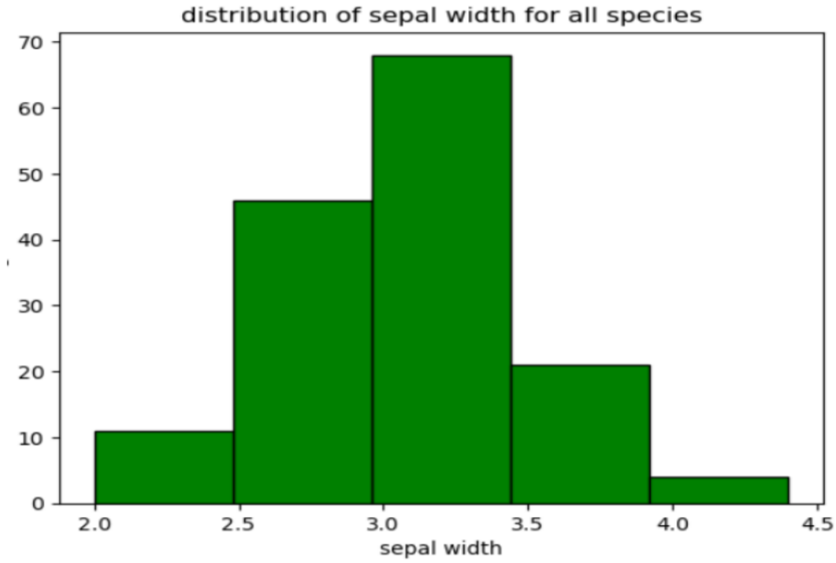
Here, comparison is being made between petal length and petal width with respect to species of flower. Hue shows the different color marks in points for each species. It shows that flower species setosa are more in number with petal length in range 1 to 2 and petal width in range 0 to 0.5 that means it is a shortest species than other two. The species versicolor represented using orange color are more in number with petal length in range 4 to 5 and petal width in range 1 to 1.5 and last species virginica represented using green color are more in number with petal length in rang 5 to 7 and petal width in range 2.0 to 2.5. it shows that virginca is taller species that other two.

Histogram:

It shows the distribution of a datasets i.e. display the frequency or count of data points falling into specific intervals known as bins. It is useful for identifying patterns, understanding the spread and shape of data. It is like a bar graph in which the bar represent the number of data points that fall within the certain range of values and the height of each bar represent the frequency of the data points in that range. To create histogram, number of bins should be determined first (bins are the range based on which count to be made like within rang of 10, 20, 5 etc.), appropriate labels and titles, consistent style, add legends and proper annotation. Histogram works with only one variables.

Plotting histogram to show distribution of sepal width of all species:

```
plt.hist(flowerData.sepal_width,edgecolor='black',color='green',bins=5);

plt.xlabel("sepal width");

plt.ylabel("range");

plt.title("distribution of sepal width for all species")
```



Here, we are trying to find out frequency (count or number) of species having sepal width with in range of 5 units. Bins indicate the range of sepal width in which the count is to be made. It shows that species having sepal width in range 3.0 to 3.5 is more in number than species with other sepal width.

Finding out distribution of each species

setosa = flowerData[flowerData.species=="setosa"]

versicolor = flowerData[flowerData.species=="versicolor"]

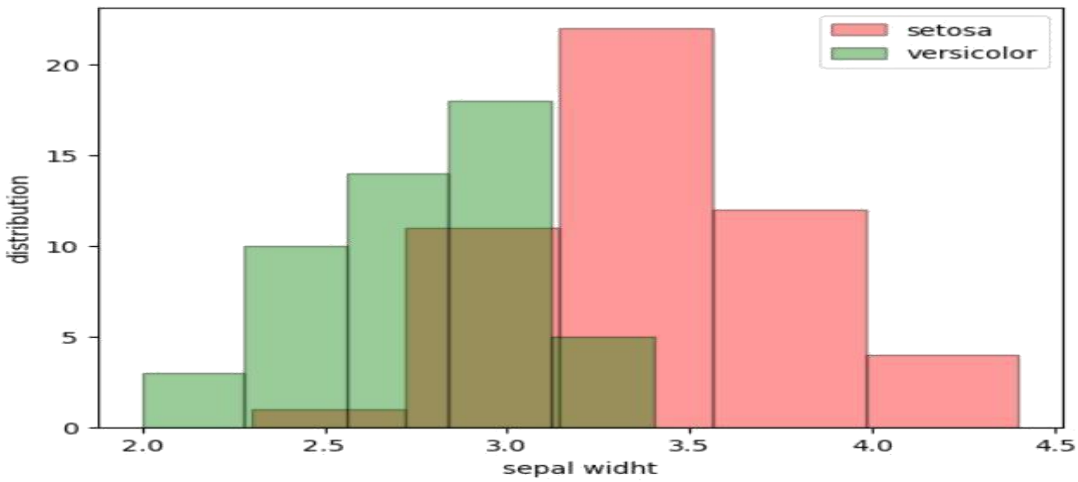
plt.hist(setosa.sepal_width,edgecolor='black',color='red',alpha=0.4,bins=5)

plt.hist(versicolor.sepal_width,edgecolor='black',color='green',alpha=0.4,bins=5)

plt.legend(['setosa','versicolor'])

plt.xlabel("sepal widht")

plt.ylabel("distribution")



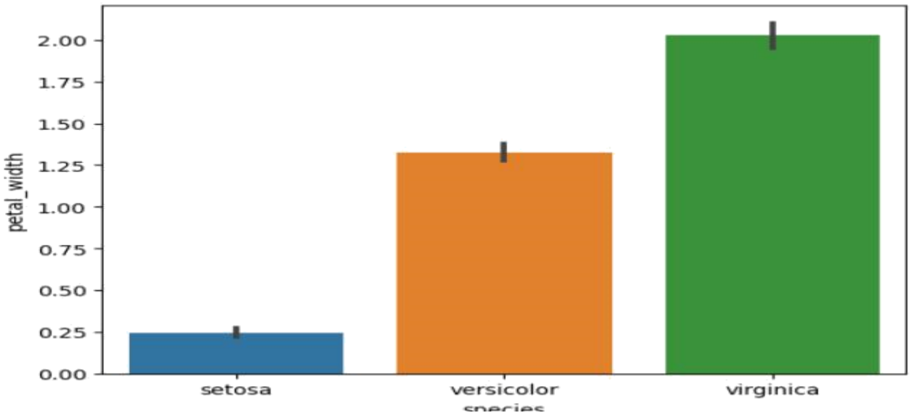
Here, at first data for each species is extracted and initialize in different variable. Different color of bar indicates different species. Red bar is for setosa and green bar is for versicolor. Comparison of frequency of species with respect to sepal width is done in range of 5 units and found that frequency of setosa is more in number with sepal width in range 3.0 to 3.5 and frequency of versicolor are more in number with sepal width in range 2.5 to 3.0.

Bar Chart:

It is used to represent categorical or discrete data and display data using rectangular bars or column. The height of each bar represents the frequency associated with a specific category. Bar charts are specially used for comparing the values of different categories over time, comparing values of a category across different groups. To create effective bar chars first categories of data should be choosen, appropriate label and title should be used to describe each coordinate, proper legend should be used to convey meaning of color, shapes and sizes of the bars.

Creating bar chart using seaborn library

sns.barplot(data=flowerData,x="species",y="petal_width")



In above figure, comparison of petal width for different species is compared and result shows that virginica have highest petal width than setosa and versicolor

Heat Map:

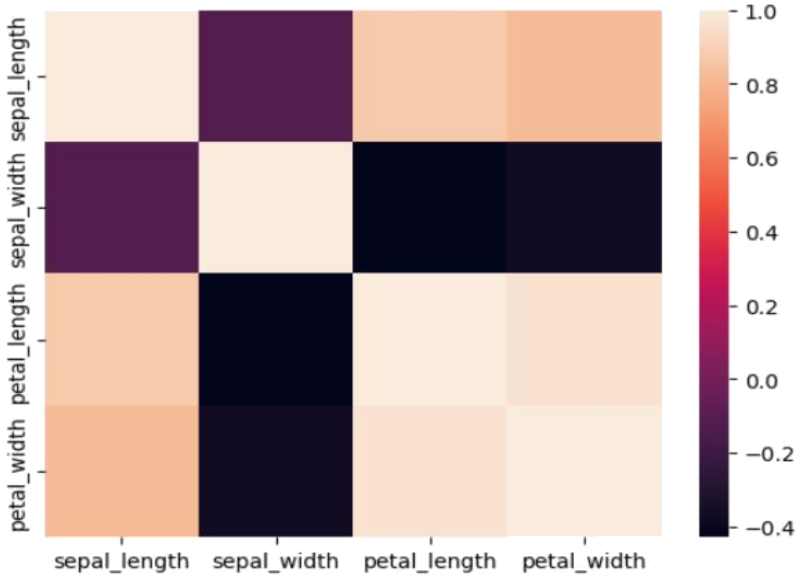
Heat map uses a color to represent the values of data. It is used to visualize two-dimensional data. It contains matrix where cells are colored according to the values of the data. The warmer colors represent higher values and the cooler colors represents lower values. Heat map represent data in a tabular format where individual values are depicted as colors which is useful for finding out patterns, trends and relationship in two-dimensional data. It shows correlation between variables. To create effective heat maps, first proper colors should be choses, add proper labels and titles, add proper legend and annotation

sns.heatmap(flowerData.corr(), cmap='coolwarm')

plt.xlabel('x-axis labels')

plt.ylabel('y-axis labels')

plt.title('heatmap example')

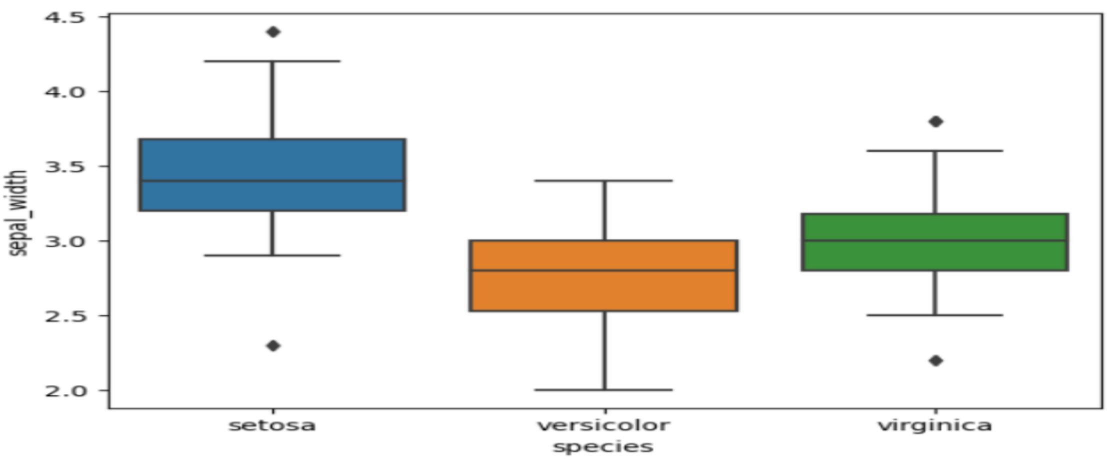


In above figure, heat map shows the correlation between all the column in data set. The darker color shows negative corelated relation between columns where as light color shows the positive corelated relation between column. That is darker color indicates that the values between column lies farther from each other and light color indicates that the values between column lies near to each other.

Box Plot:

A box plot or box and whisker plot is used to show the distribution of data in a five-number summary: the minimum, first quartile (Q1), median, third quartile (Q3) and a maximum. It represents central tendency, spread and the presence of outliers. Box represent the interquartile range which spans the 50% of the data. It contains a horizontal line that shows the median of data and the top and bottom of edges of box represent the first quartile and third quartile of the data. Whisker extends from the box to minimum and maximum values within a defined range. Data points beyond whiskers are considered as outliers. Outliers are data points that falls outside of whiskers and are represented as individual point or dots.

sns.boxplot(data=flowerData,x='species',y='sepal_width')



Above figure shows the box and whisker plot for species and sepal width, the middle line inside box shows median and lower line shows first quartile and upper line shows third quartile, the straight horizontal line in lower bound represents minimum data and upper bound represents maximum value and the point indicates outliers. For species setosa, median is 3.4, first quartile is 3.2, third quartile is 3.7, minimum value is 2.8, maximum value is 4.2 and outlier is 4.4. For species versicolor, median is 2.8, first quartile is 2.4, third quartile is 3.0, minimum value is 2.0, maximum value is 3.4 and outlier is 0. For species verginica, median is 3.0, first quartile is 2.8, third quartile is 3.2, minimum value is 2.4, maximum value is 3.6 and outlier is 3.8.

Tree data Visualization:

Tree data is a type of data that is organized hierarchically with each data point having one or more parent data points. Tree data can be used to represent a variety of real-world structure such as organizations, product hierarchies and family trees. The data points are organized in a tree like structure with each data point having one or more child data points. The root is the topmost data point in the tree and it has no parent data points. The child data points are its immediate descendants. Visualizing tree data can help reveal patterns, relationship and structures within the data. Some of the visualizing technique used for displaying hierarchical data are tree maps, sunburst chart etc. key components of a tree data structure are:

- Root nodes: the topmost node which serves as the starting point for traversing the tree and it has no parent
- Child node: a node that is directly connected to another node and is one level below it in the hierarchy. A node can have multiple child nodes
- Parent node: a node that has one or more child nodes. It is one level above its child nodes
- Sibling: nodes that share same parent node are siblings
- Leaf node: a node that has no child nodes i.e. it is a terminal node at the end of branch

The tree data structure are used to represent hierarchical relationship and organizational structure.

Visualizing hierarchical data:

1. **<u>Treemaps:</u>** it uses nested rectangles to represent the hierarchical structure of tree data. The size of each rectangle represents the value of the data point it represents. It is done in python my using plotly library. Following program shows an example on treemaps.

   Step1: import library
   Import plotly.express as px
   Step 2: create data set or import dataset
   Step3: use plotly.treemap(dataset, path, values,color)

   Here dataset is the actual data for which tree map is to be created, path indicates list of columns name or column of rectangular data frame defining hierarchy of sector from root to leaves. Values is either a name of a column or pandas' series and such field is used to set values associated to sector. Color are used to assign appropriate color pallet.

   <u>Treemap by creating own series of data</u>

   let's create two series (array) using pandas: first one is name of people and second one is name of parent for corresponding name of people.

```
Import plotly.express as pl
res=pl.treemap(names=["Ram","Sam", "Hari", "Gita", "Sita", "Ramila", "Pawan", "Riya", "Sarmila"], parents=["", "Ram", "Ram", "Hari", "Hari", "Ram", "Ram", "Pawan", "Ram"])
res.update_traces(root_color='lightgrey')
res.update_layout(margin = dict(t=50, l=25, r=25,
b=25)) res.show()
```

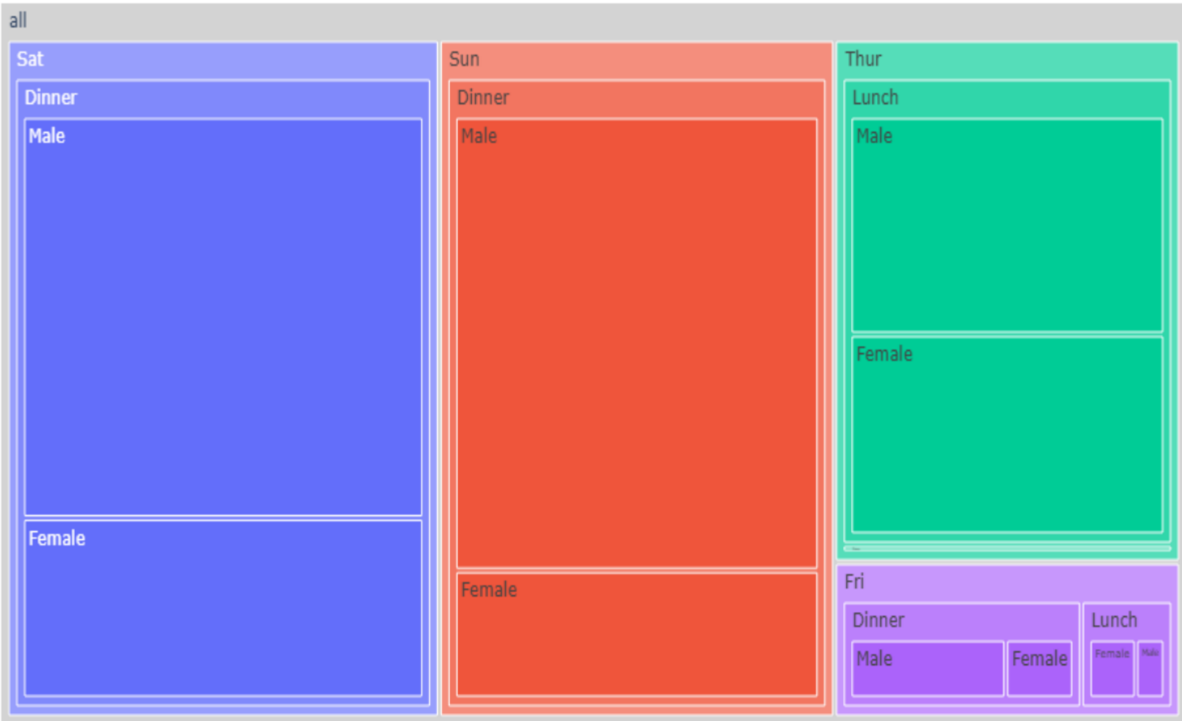
In above code:

```
res=pl.treemap(names=["Ram","Sam", "Hari", "Gita", "Sita", "Ramila", "Pawan", "Riya", "Sarmila"], parents=["", "Ram", "Ram", "Hari", "Hari", "Ram", "Ram", "Pawan", "Ram"])
```

names represent series of name of people and parents represent parent of each people represented in names series. In parents: data in first position is empty meaning that first element of name series (Ram) does not have any parent therefore it is root element. The data in second position of parent is Ram meaning that it is the parent of Sam which is in second position of name's series and so on.

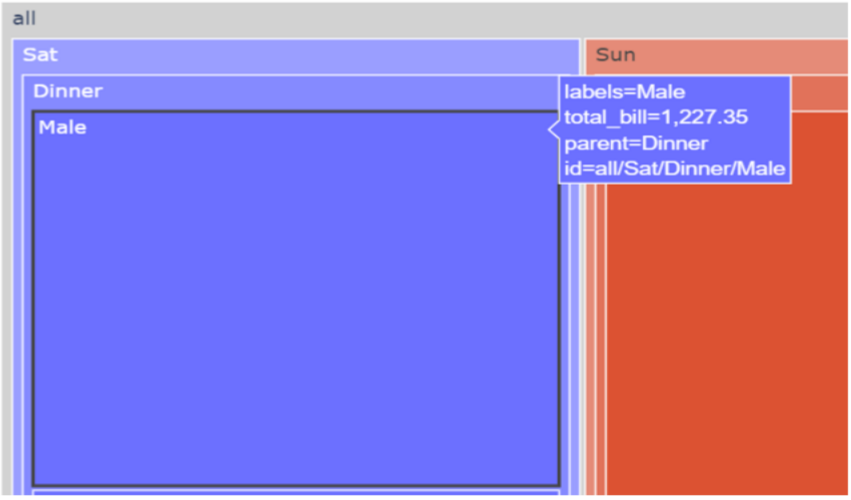
Treemap for tip data sets

```
import plotly.express as px
import seaborn as sns
#load dataset
df=sns.load_dataset('tips')
#plotting treemap
fig=px.treemap(df,path=[px.Constant("all"),'day','time','sex'],values='total_bill')
fig.update_traces(root_color="lightgrey") # to put color in root
fig.update_layout(margin = dict(t=50, l=25, r=25, b=25))
fig.show()
```



In above code, px.treemap(df,path=[px.Constant("all"),'day','time','sex'],values='total_bill')

Px.Constant("all") it refers to root node which doesnot have any parent, day comes under all (parent), time comes under day and sex comes under time. If mouse is hover in each box then it shows the detail of such box. For example when hover on male of first box coded with blue color following output is generated:



Above figure shows the result of box male which shows that total_bill is 1227.35 for sex male who order dinner on day Saturday.

Graph data:

Graph data refers to data structure that consist of nodes (vertices) and edge (connections) that define relationships between the nodes. In data visualization, graph data is known as network data that refers to datasets that contain information about relationship or connection between entities. These entities can be represented as node (vertices) and the relationships between them are represented as edge (links or connections). Graph data represents relationships between entities which can be anything like people, place or things. Graph data are used to solve problem like finding shortest path, finding connected components and finding communities. Following are the visualization eccentrics used for visualizing graph data

- Tree map
- Node-link diagram

Rules of graph drawing and labeling in data visualization:

- Use labels to identify the nodes and edges
- Choose an appropriate graph layout algorithm that organize nodes and edges in a visually logical and meaningful way. Common layouts include force-directed, hierarchical and circular layouts. The layout of graph should be easy to read and understand and the node should be spaced out so that they are not close together and the edge should be routed in a way that does not clutter the graph
- Hierarchy: for hierarchy data, use hierarchical layouts to emphasize the structure and ensure that the parent child relationships are clearly represented.
- Alignment: align node and edge as needed to create visual structure and aid in understanding
- Node placement: place lables or data values close to their corresponding nodes and use readable fonts and font sizes

- Spacing: use appropriate spacing between node and edge to prevent overlapping and improve readability.
- Use legend to explain meaning of the colors, shapes and sizes. The legend should be placed in a location where it is easy to find and read
- Use consistent style. The color, shape and sizes of the nodes and edges should be consistent throughout the graph.

Rules of graph labeling:

- Node labels: labels should be in short for and not too long and labels should be meaningful names or identifier.
- Edge labels: if label for edge is necessary then ensure that they convey important information. Edge label should be placed closed to the edge they describe
- Node and edge color: use color for node and edge strategically to convey information. Different color should be used for node and edge
- Legend: legend should be provided if different symbols are used to represent information
- Annotation: text annotation should be used to add explanatory notes or highlight specific features or data points in the graph
- Axes labels: if the graph includes axes then it should be label with units of measurement and description
- Consistency: the label and annotation should be consistent with the data and visual elements in the graph
- Language: use clear and concise language in labels and annotations

Visualizing graph data:

In python, NetworkX library is used for creating and analyzing graphs and matplotlib for visualization. NetworkX is used to create and manipulate graphs.

Step1: import necessary library i.e. matplotlib and network

import matplotlib.pyplot as plt

import networkx as nx

Step2: create a empty graph using Graph () method of network

Step 3: add node (vertex) by using add_nodes() function

Step 4: add edges by using add_edges() function

Step 5: draw the graph by using draw() function of network.

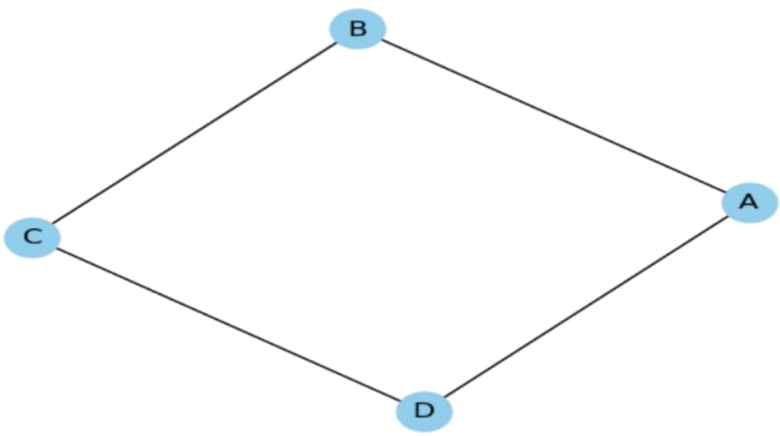
import matplotlib.pyplot as plt

import networkx as nx

#creating an empty graph

gs = nx.Graph()

```
#add Nodes
gs.add_node("A")
gs.add_node("B")
gs.add_node("C")
gs.add_node("D")
#add edges
gs.add_edge("A","B")
gs.add_edge("B","C")
gs.add_edge("C","D")
gs.add_edge("D","A")
#drawing a graph
fig,ax=plt.subplots()
nx.draw(gs,ax=ax,with_labels=True, node_size=500,node_color='skyblue')
plt.show()
```



Drawing different layouts of graph

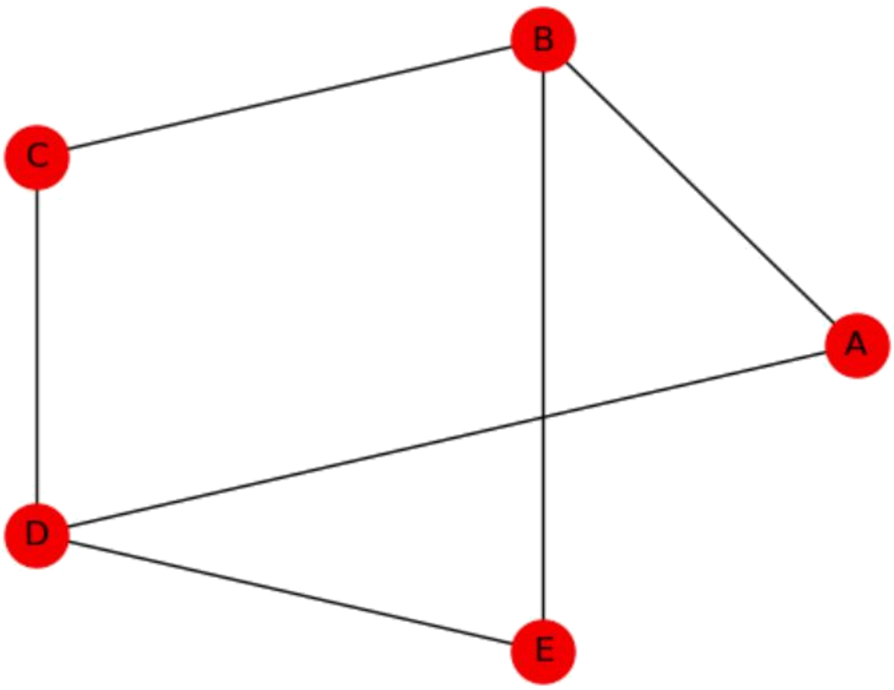
Datasets:

```
#creating an empty graph
gs1 = nx.Graph()
#add Nodes
gs1.add_node("A")
gs1.add_node("B")
```

```
gs1.add_node("C")

gs1.add_node("D")

gs1.add_node("E")

#add edges

gs1.add_edge("A","B")

gs1.add_edge("B","C")

gs1.add_edge("C","D")

gs1.add_edge("D","A")

gs1.add_edge("D","E")

gs1.add_edge("E","B")
```

Circular Layout:

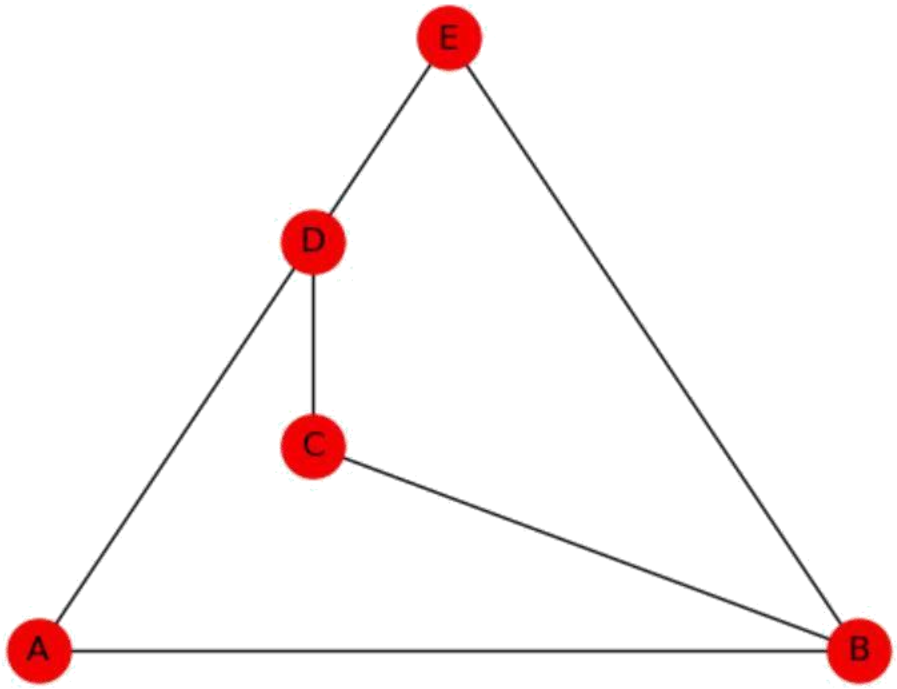
```
#drawing a circular layout

fig,ax=plt.subplots()

nx.draw_circular(gs1,ax=ax,with_labels=True, node_size=500,node_color='red')

plt.show()
```



Planner Layout:

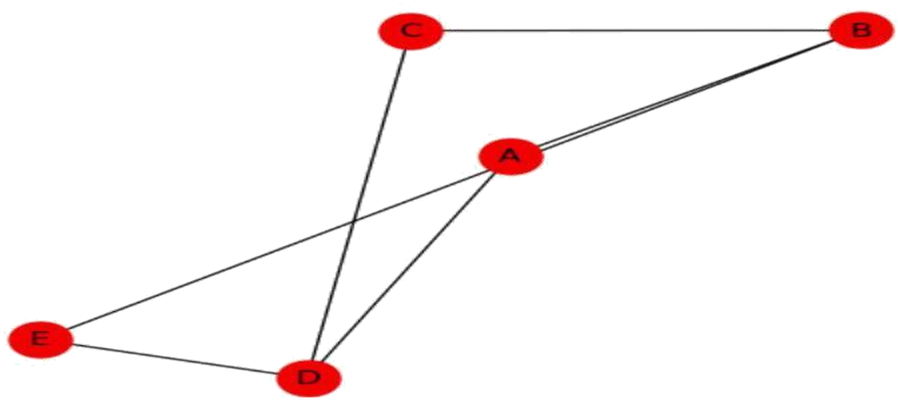
```
#planner layout

fig,ax=plt.subplots()
```

```
nx.draw_planar(gs1,ax=ax,with_labels=True, node_size=500,node_color='red')
plt.show()
```



Random Layout

```
#random layout
fig,ax=plt.subplots()
nx.draw_random(gs1,ax=ax,with_labels=True, node_size=500,node_color='red')
plt.show()
```



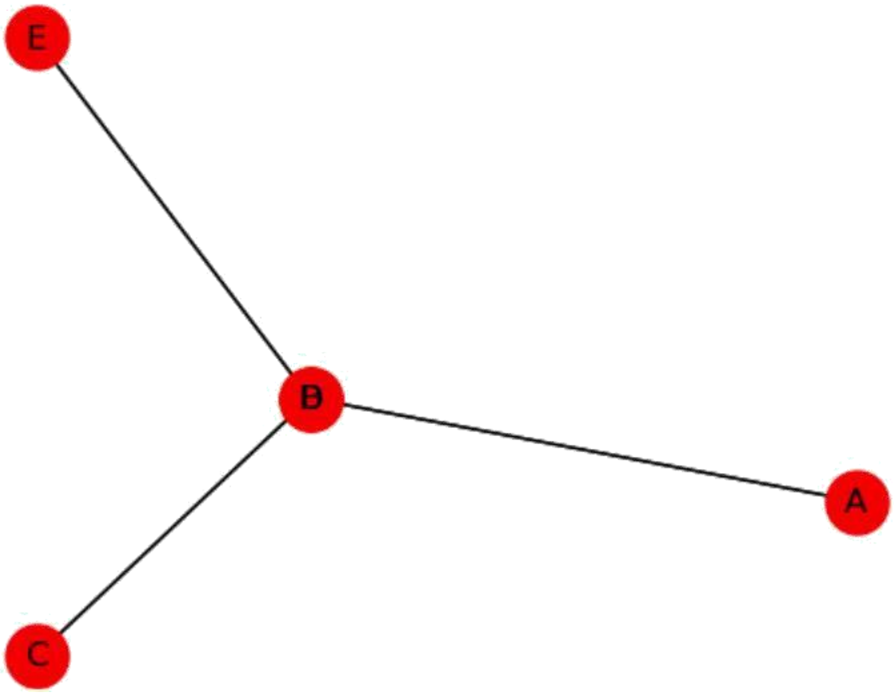
Spectral Layout

#spectral layout

fig,ax=plt.subplots()

nx.draw_spectral(gs1,ax=ax,with_labels=True, node_size=500,node_color='red')

plt.show()



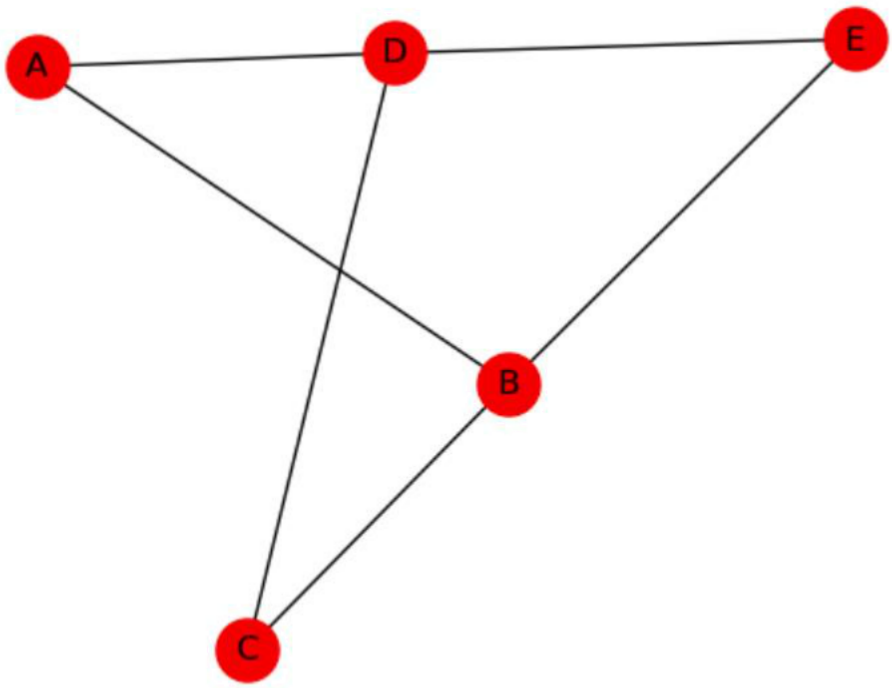
Spring Layout

#spring layout

fig,ax=plt.subplots()

nx.draw_spring(gs1,ax=ax,with_labels=True, node_size=500,node_color='red')

plt.show()



Text and document data in data visualization:

Text and document data are a type of data that consists of text and other unstructured data which are difficult to visualize. Text data refer to textual information and documents that can he in various formats such as plain text, pdf, word document etc. Text visualization involves techniques for representing and visualizing textual information and document in a way that helps users to understand and gain insight from data. Text data can be from various sources such as articles, books, customer review etc. some visualization technique used for text visualization are word clouds (display the most frequently occurring words in a text. Used to identify frequently mentioned terms in documents), bar chart and histogram, network diagram, heatmaps, time series plot (shows specific text segment by highlighting and annotating document or articles). Some characteristics of text data are unstructured, diversity, volume, variety etc.
Here we consider visualizing the text within a document, and collections of documents which are likely related (corpus).

Difficulty in analysis includes the loose structure, varied vocabulary, and optional metadata such as author(s), date, modification dates, comments, keywords, catalog codes, citations.

Levels of text to be represented:

Lexical level -- Simple grouping of characters into "tokens" which are typically words, but word stems, phrases, word n-grams and character n-grams may be beneficial
Syntactic level --Parsing purpose of token, grammatical category, tense, plurality, in the context of the phrase, sentence and paragraph
Semantic level -- Extract meaning of the syntactic structure with the tokens using fuller analysis of the context.

Text visualization is a visual way of presenting information—word clouds, graphs, maps, timelines, networks and more, can all be used to visualize text data. Doing so provides a brief understanding of the most important keywords, and sums up and communicates trends and frameworks within a specific text.

Text visualization is the technique of using graphs, charts, or word clouds to showcase written data in a visual manner. This provides quick insight into the most relevant keywords in a text, summarizes content, and reveals trends and patterns across documents.

Companies use text visualization to:

- **Summarize large amounts of text.** Automatically highlight key terms in a series of texts, and categorize text by topic, sentiment, and more, saving hours of reading time. How long would it take you to read 500 online reviews? With a word cloud or data visualization dashboard, you can understand text data at a glance.
- **Make text data easy to understand.** The human brain loves visual data. In fact, we are able to process images much faster than text. Text visualization is an effective way of simplifying complex data and communicating ideas and concepts to team managers.
- **Find insights in qualitative data.** Customer feedback holds a trove of insights. Through text visualization, you can get an overview of the features, products, and topics that are most important to your customers. Learn what their pain points are and what you're doing right.
- **Discover hidden trends and patterns.** Analyze and visualize insights over time to detect fluctuations, and quickly find the root cause.

Text is maybe the most underrated element in any data visualization. There's a lot of text in any chart or map — titles, descriptions, notes, sources, bylines, logos, annotations, labels, color keys, tooltips, axis labels — but often, it's an afterthought in the design process. This article explains how to use text to make your visualizations easier to read and nicer to look at.

Show information where readers need it

01 Label directly

02 Repeat the units your data is measured in

03 Remind people what they're looking at in tooltips

04 Move the axis ticks where they're needed

05 Emphasize and explain with annotations

Design for readability

06 Use a font that's easy to read

07 Lead the eye with font sizes, styles, and colors

08 Limit the number of font sizes in your visualization

09 Don't center-align your text

10 Don't make your readers turn their heads

11 Use a text outline

Phrase for readability

12 Use straightforward phrasings

13 Be conversational first and precise later

14 Choose a suitable number format

Why Do We Need Text Visualization?

Text Visualization can help reveal your audience's thoughts

You can use the chart to understand your audience's feelings about a topic/situation. Besides, you can leverage the chart to summarize data-driven views. The chart can help you summarize the market feedback using first-hand data.

Quick and informative

You can easily get live feedback from your audience in real-time

Exciting and emotional

The chart can help audiences feel part of your data story.

Engaging

The Word cloud is incredibly engaging and visually appealing to many audiences. The chart can be an icebreaker or an entry point for a topic of discussion.

Word Clouds are visual

Our brains process visual content 60,000 times faster than texts and numbers. This provides a logical rationale for using the Word Cloud generator to analyze your textual data for actionable insights.

Creating a text visualization is straightforward

Generating text visualization examples is easy to follow. Yes, you read that right. Besides, the chart can provide you with insights into large data sets.

Text Data Visualization Examples

Word Cloud

Word Clouds are charts that display insights into qualitative data frequency.

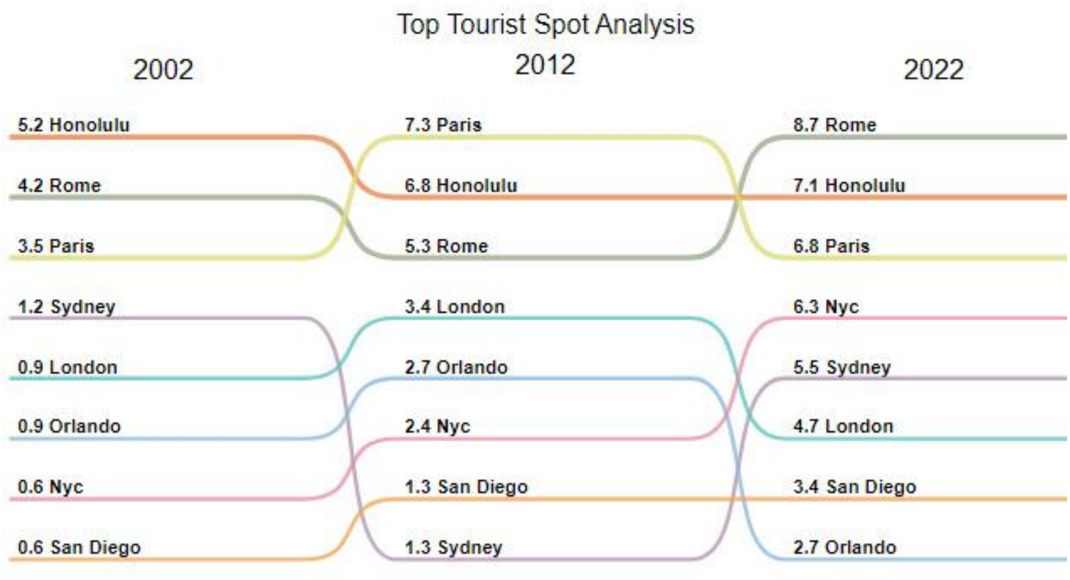
The visualization design gives greater prominence to words that appear more frequently in a source text. The larger the word, the higher its frequency. You can use the chart (one of the text visualization examples) to perform exploratory textual analysis by identifying words that frequently appear in a set of interviews, documents, or other text.

Tag Clouds

Tag clouds or text clouds are ideal if your goal is to pull out the most pertinent parts of textual data, from blog posts to databases. You can use the tag cloud as a text visualization tool to compare and contrast two different pieces of text for similarities and differences.

Slope Chart

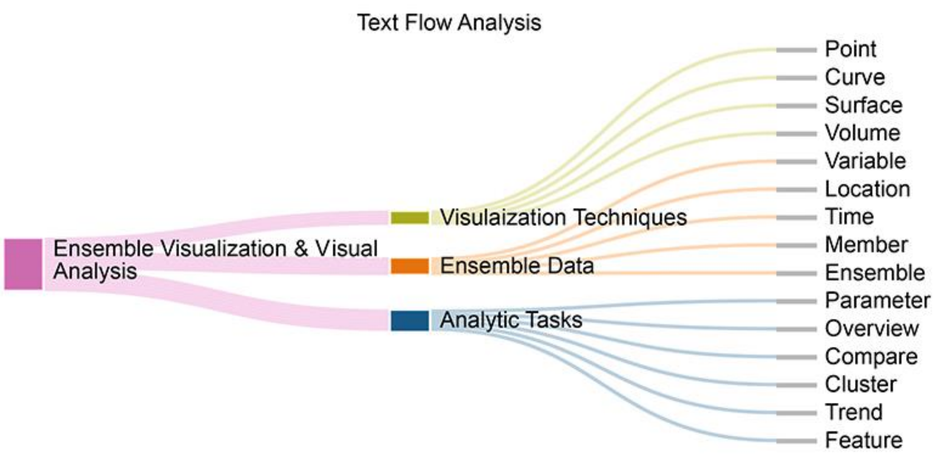
Slope Charts show transitions, changes over time, absolute values, and even rankings. Besides, they’re also called Slope Graphs.



Slope Graphs (one of the text visualization examples) can be useful when you have two time periods or points of comparison and want to show relative increases and decreases quickly across various categories between two data points.

Sankey Chart

A Sankey Diagram visualizes "a flow" from one set of values to the next. The two items being connected are referred to as "nodes." The connections are labeled as "links".



Text Visualization is Useful for:

Condensing a lot of content. Cut down on time spent reading by emphasizing central phrases across multiple texts, grouping content by topic, sentiment and more. Could you imagine having to get through hundreds of client reviews? With a word cloud or bar chart, you can visualize data and instantly make sense of things.

Simplifying text data. Our brains are wired to enjoy and make sense of visual data and it's proven that we sort through images quicker than we do with the written word. If you're looking to simplify complex data and transmit those concepts to team managers, then text visualization is the way to go.

Determining insights in qualitative data. Customer feedback is jam-packed with practical insights. You'll get an effective outline of the products, features and subjects that matter most to your clientele and the opportunity to figure out not only their pain points but where you're succeeding with them.

Discover hidden trends. Use text analysis and gradually visualize insights in order to spot easily any inconsistencies and figure out the leading causes.

Text Mining

The fast growth spurt of social media platforms and availability of the internet means that year after year, a massive quantity of unstructured text data is produced. And that's what text analysis is all about—acquiring insights or assembling this raw data with a view to propelling research, projects, business and other such activities.

A fresh area of research has emerged in the use of machinery to investigate texts—text mining. This is in contrast to the process of data mining used in computer science.

Text mining aims to uncover statistical patterns as it uses machines to analyze data points in a body of content with a large volume of text. Through this procedure, various patterns within a big data system begin to emerge.

Text mining benefits from text visualization tools as it's so easy to read for both machine and human alike. The most vital bits of information are communicated through easy-to-read visual representations such as a bar chart, word cloud, graph, map, timeline or network.

Why Text Visualizations are Necessary

Makes Text Data Easy to Grasp

Did you know that your brain sorts through visual data 60 000 times faster than words or numbers? Text visualizations make complex data clearer and powerfully transmit ideas to team managers.

Communicates What's on Your Audience's Mind

A chart can help you figure out how your audience feels about a certain subject or issue. This chart can also be leveraged to condense data-driven views. First-hand data can be used to summarize any market feedback.

Condenses Big Volumes of Text

Reduce the time you'd spend reading big volumes of text. Instantly emphasize the main terms in a string of texts, categorize content by subject, sentiment or other themes.

A quick scan of a text data visualization or dashboard will update you on all the vital info you want and need to know.

It Captivates

If you take a look at a word cloud, you'll see that it is both eye-catching and informative. A well-designed chart can be used to start a conversation on an array of interesting topics.

It is Simple and Direct

Creating and reading text visualizations are actually pretty straightforward. Whether it's a bar chart or a graph, you'll gain some actionable insights into sizable data sets.

Word Cloud

A word cloud is a text visualization technique that focuses on the frequency of words and correlates the size and opacity of a word to its frequency within a body of text. The output is usually an image that depicts different words in different sizes and opacities relative to the word frequency.

An application of this form of visualization is document summarization, where you can process a body of text within a document and, based on the most prominent words, get a general summary of what the document is all about. This can also be applied in job applications where if the job description is analyzed, the largest words to appear are most likely the most important skills for the job.

Visualizing text data using word cloud:

Use to visualize the frequency of words in a text groups with word size proportional to frequency. To use wordcloud, first download the package using pip install wordcloud. After this import the library using from wordcloud import WordCloud. Further process are shown below

Importing library

from wordcloud import WordCloud

import matplotlib.pyplot as plt

#text to be visualized

text = "hello BCA 8th semester. you are studying data visualization and best of luck for your board exam. you exam is on sunday and center is in padmakanya. if you fail in exam you get astrick in marksheet. Best of Luck "

plotting word cloud

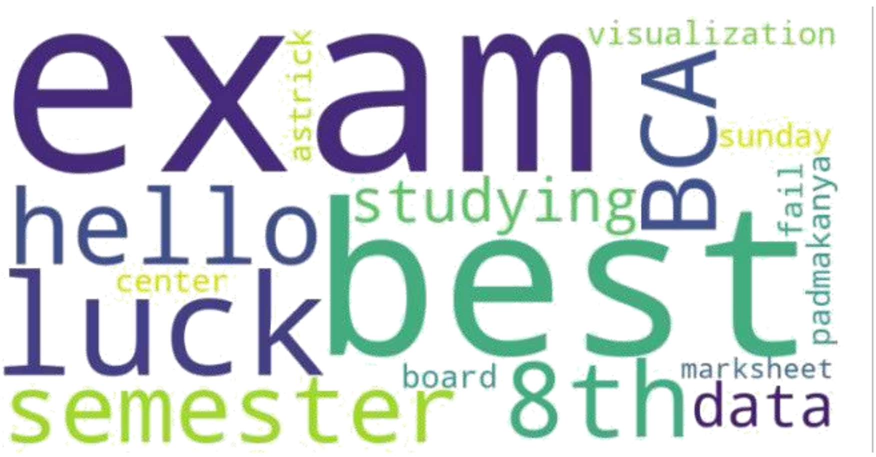
wc = WordCloud(width=800, height=400, background_color="white").generate(text)

plt.figure(figsize=(10, 5))

plt.imshow(wc, interpolation="bilinear")

plt.axis("off")

plt.show()



In above code: wc = WordCloud(width=800, height=400, background_color="white").generate(text)

WordCloud function is used to generate wordcloud and text to be represented is calculated by generate() method.

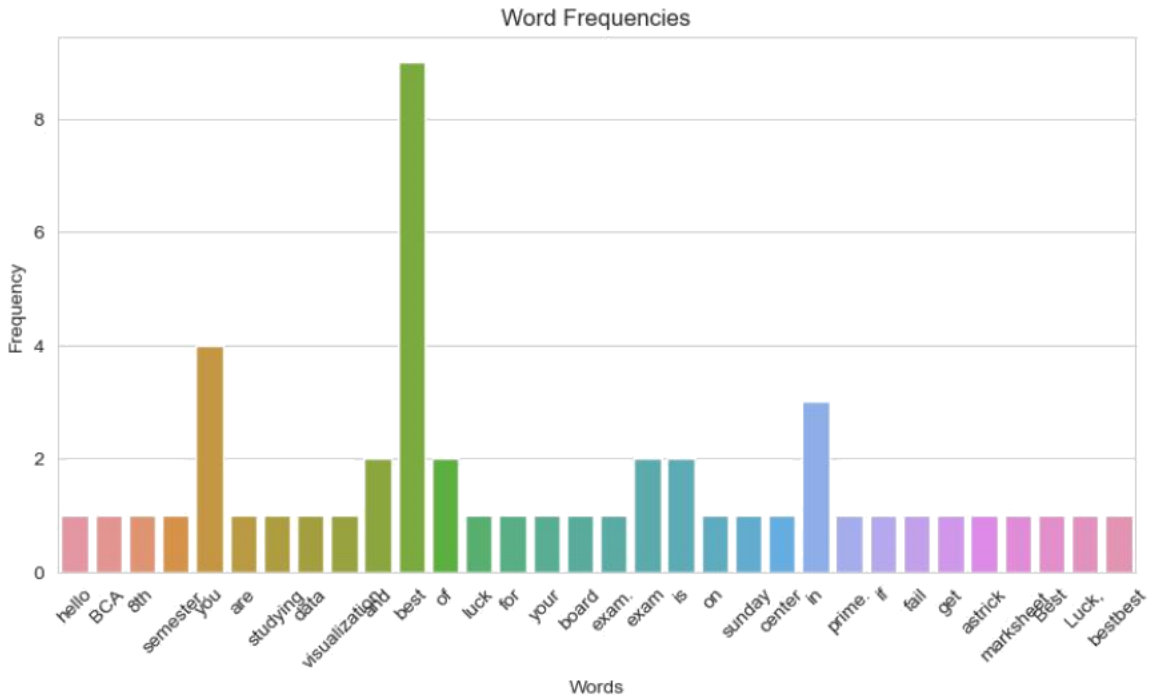
Visualizing text data using bar graph:

import matplotlib.pyplot as plt

import seaborn as sns

```
from collections import Counter

text = "hello BCA 8th semester. you are studying data visualization and best of luck for your board exam. you exam is on sunday and center is in prime. if you fail in exam you get astrick in marksheet. Best of Luck, best best best best bestbest best best best best "


word_list = text.split()

word_freq = Counter(word_list)
```

```
sns.set_style("whitegrid")

plt.figure(figsize=(10, 5))

sns.barplot(x=list(word_freq.keys()), y=list(word_freq.values()))

plt.xticks(rotation=45)

plt.xlabel("Words")

plt.ylabel("Frequency")

plt.title("Word Frequencies")

plt.show()
```



Separate, Order and Align
In information visualization the most effective channel for encoding information is position. The usage of this channel depends of the type of attribute encoded. Quantitative variables should *express* its value across a continues scale, but for categorical ones you should **separate, order and align**.

Separate:

- To separate means to divide or set apart different elements or entities.
- It involves creating distinct boundaries or divisions between items or groups.
- The purpose of separation is to distinguish or isolate individual components, ensuring that they are distinct and independent from one another.

Separate (Data Visualization):

- In data visualization, separation involves visually distinguishing or isolating different data elements or categories.
- It entails creating clear boundaries or visual cues that set apart various data points, groups, or variables.
- By separating data, you can enhance the viewer's ability to differentiate and understand the distinct components within a visual representation, such as charts, graphs, or diagrams.

Example

- Suppose you have a bar chart representing sales data for different product categories. To separate the data, you can use distinctive colors for each category. For instance, you could assign the color blue to electronics, red to clothing, and green to furniture. This visual separation helps viewers easily identify and differentiate the sales figures for each product category.

Order:

- Order refers to arranging or organizing things in a particular sequence or pattern.
- It involves placing items or elements in a structured manner according to a specific criterion or system.
- Ordering can be done based on various factors, such as numerical, alphabetical, chronological, or hierarchical order.
- It helps to bring clarity, efficiency, and logic to a set of elements.

Order (Data Visualization):

- Ordering data in the context of data visualization means arranging the data points or categories in a structured manner to convey meaning or facilitate comprehension.
- Depending on the nature of the data, you can order it based on numerical values, alphabetical sequences, time periods, or hierarchical relationships.
- By ordering the data, you bring a logical and coherent structure to the visualization, allowing viewers to identify patterns, trends, or comparisons more easily.

Example

- Consider a line graph depicting stock market prices over time.
- To order the data, you would typically arrange the time series on the x-axis in chronological order, with the oldest date on the left and the most recent date on the right.
- This allows viewers to observe the progression of stock prices over time and identify any patterns or trends, such as upward or downward movements.

Align

- Alignment means adjusting or positioning things in such a way that they are in proper or accurate coordination or arrangement with one another.
- It involves making sure that different elements are correctly placed or matched relative to a reference point or a common set of guidelines.

- Alignment ensures that components are in harmony, agreement, or conformity with each other, facilitating coordination and coherence.

Align (Data Visualization):

- Alignment in data visualization refers to positioning and coordinating visual elements to create a visually harmonious and informative display.
- It involves aligning data points, labels, axes, or other graphical components with precision.
- Proper alignment ensures that the visual elements are visually cohesive, making it easier for viewers to interpret and make accurate comparisons or connections within the visualization.
- Alignment can also apply to text, spacing, and grid lines, promoting readability and clarity in the visualization.

Example

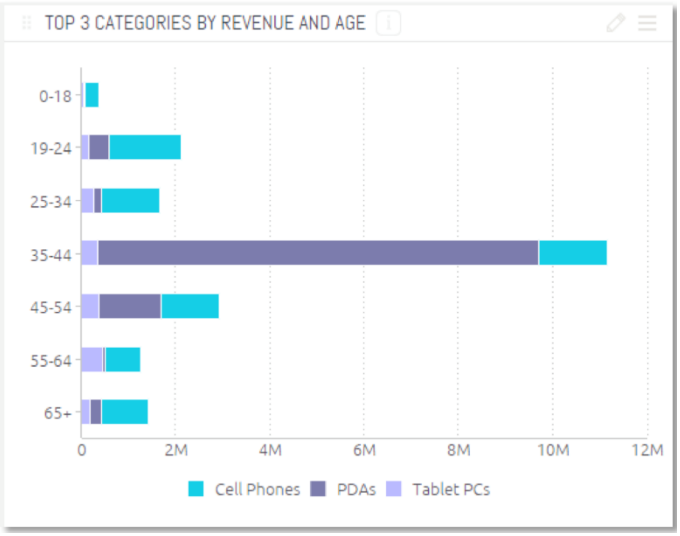
- Imagine a scatter plot representing the relationship between a person's age and their income.
- To align the data points, you would ensure that the x-axis represents age, starting from the lowest age value on the left and increasing towards the right.
- The y-axis would represent income, starting from the lowest income value at the bottom and increasing as you move upward.
- Aligning the axes in this way ensures that the data points align with the appropriate age and income values, enabling viewers to accurately interpret the relationship between the two variables.

Summary:

- Separate: Visually distinguish or isolate different data elements or categories, creating clear boundaries or cues to enhance differentiation.
- Order: Arrange data points or categories in a structured manner based on a specific criterion, facilitating comprehension and pattern identification.
- Align: Position and coordinate visual elements with precision, creating a visually cohesive and informative display that promotes readability and interpretation.

Bar Chart

- Use the bar chart to compare many items. The bar chart typically presents categories or items displayed along the Y axis, with their values displayed on the X axis. You can also break up the values by another category or group.



What can I use bar and column charts for?

Use bar and column charts to summarize and compare values in a data category, and provide a snapshot of your data at specified points in time (or other dimensions).

Although bar and column charts are similar in their appearance and functions, their respective orientations mean they are better suited to different types of analysis:

- Bar charts use a horizontal display, which provides more room for long, complex or numerous labels on the Y-axis. The labels also go from left to right, making labels on bar charts easy to read. In bar charts, categories are usually displayed along the Y-axis, so they're commonly used for analyses where time is not a factor.
- In column charts, time is usually displayed along the X-axis, with the unit of measurement on the Y-axis. As this left to right direction is associated with a chronological sequence, column charts are ideal for highlighting changes over time.

You can also use bar and column charts to rank items in a data series. To do this, you must sort your data (by saving a view in your module, or designing a custom view) before creating a chart, so that the ranking is reflected in it.

Negative values are represented clearly in bar and column charts, as any negative values are plotted in the opposite direction to positive values. However, relatively large negative or positive values reduce the amount of space available on a chart, which can make it hard to differentiate between similar values.

Use a bar or column chart to answer:

- How does A differ from B?
- Which salesperson sold the most product?
- What was the average growth over the last X years?
- What is the composition of our website traffic?

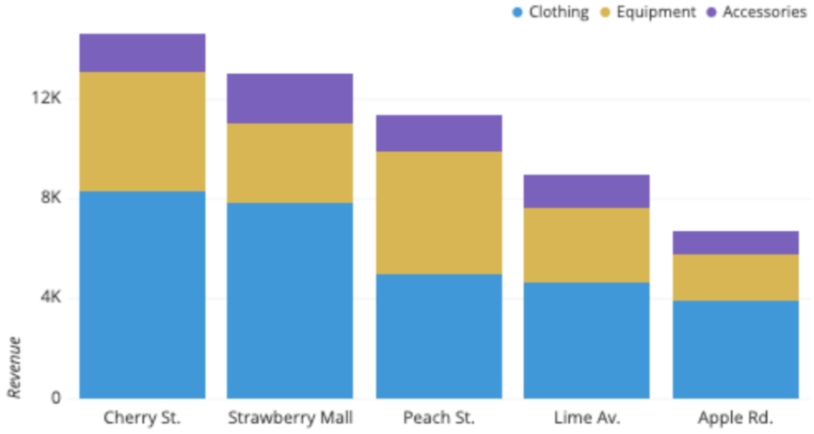
Considerations

A bar or column chart may not be the best option when:

- your data categories have long labels.
- the height or width size of a chart card has been altered. This can change the scale of your unit of measurement, which can be misleading as this diminishes or emphasizes differences in value in your bar or column chart.

Staked Bar Graph

- The stacked bar chart (aka stacked bar graph) extends the standard bar chart from looking at numeric values across one categorical variable to two. Each bar in a standard bar chart is divided into a number of sub-bars stacked end to end, each one corresponding to a level of the second categorical variable.



The stacked bar chart above depicts revenue from a fictional fitness retailer for a particular period of time, across two categorical variables: store location and department. The primary categorical variable is store location: we can see from the sorted overall bar heights that the Cherry St. location has the highest revenue and Apple Rd. lowest. Each bar is subdivided based on levels of the second categorical variable, department. We can see that for most locations, clothing is quite a bit larger in sales than equipment, which in turn is larger than accessories. The Strawberry Mall location appears to have a lower proportion of revenue attributed to equipment, while equipment has a larger share for Peach St.

When you should use a stacked bar chart

The main objective of a standard bar chart is to compare numeric values between levels of a categorical variable. One bar is plotted for each level of the categorical variable, each bar's length indicating numeric value. A stacked bar chart also achieves this objective, but also targets a second goal.

We want to move to a stacked bar chart when we care about the relative decomposition of each primary bar based on the levels of a second categorical variable. Each bar is now comprised of a number of sub-bars, each one corresponding with a level of a secondary categorical variable. The total length of each stacked bar is the same as before, but now we can see how the secondary groups contributed to that total.

Order of categorical variables

One important consideration in building a stacked bar chart is to decide which of the two categorical variables will be the primary variable (dictating major axis positions and overall bar lengths) and which will be the secondary (dictating how each primary bar will be subdivided). The most 'important' variable should be the primary; use domain knowledge and the specific type of categorical variables to make a decision on how to assign your categorical variables.

Dot Charts

- Dot charts display the values of your data as a series of colored dots on two axes. Dimensions, such as time, display on the horizontal x-axis, and the dependent data displays on the vertical y-axis.
- Dot charts show the exact values for your data points. Dot charts can be very effective as part of a combination chart.

What can I use dot charts for?

Dot charts show pinpoint values for your data against a dimension, such as time. Use them when you want to place emphasis on the exact values of your data rather than its comparative size.

Dot charts enable you to compare a large number of data series, where other chart types are liable to become visually overcrowded.

Dot charts work well as part of a combination chart. You can overlay a dot chart over another type of chart to highlight a target or benchmark, without the need to hide data.

Use a dot chart to answer:

- What are the exact values of my data for different time periods?
- How have sales differed across different locations and demographics?
- Which product performed the best each month?

Considerations

A dot chart may not be the best option when:

- data points are close in value, as dots can overlap and be hard to read, unlike other charts.

- visualizing total quantities or proportions.

Line Chart

- A line chart is a way of plotting data points on a line. Often, it is used to show trend data, or the comparison of two data sets.
- A **line chart** provides the clearest graphical representation of time-dependent variables.
- It is also the preferred mode of representing trends or variables over a period of time.
- People are familiar with this simple chart, which is made up of data values plotted as points along the X and Y axes and are connected using line segments.
- Usually, time is plotted along the X-axis, and the Y-axis represents some metric of interest in the context of the period being tracked.
- Line charts represent everyday items like weekly weather trends, the price of stocks, what topics are trending on social media, and health information.
- Example
    - Publicists, brand managers, or public relations specialists track their client's social media ranking over time to plan campaigns.

What can I use line charts for?

Line charts display one or more dependent variables against one independent variable, such as time. This enables you to easily see spikes and troughs in a continuous data set.

Line charts are particularly useful for emphasizing trends, as they clearly show the rate of change over a fixed period of time or other dimension. When included in a combination chart, line charts can show the effect that a trend has on other values.

Use a line chart to answer:

- What are the fluctuations in X?
- How does A differ from Y?
- Is X related to A?
- How is X affecting Y?
- Have sales increased over the last financial year?

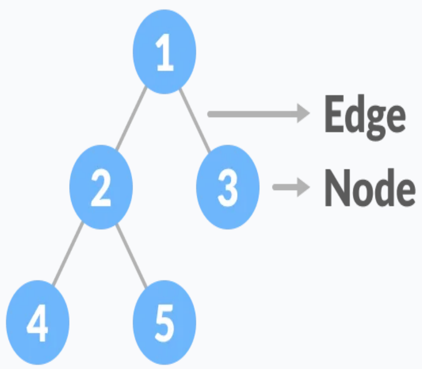
Considerations

A line chart may not be the best option when:

- conveying proportions or quantitative data.
- there are a large number of categories, as they can become hard to read.
- your X-axis dimension doesn't have data over a long time period, as this could give a false impression of ongoing trends.

Tree Data

- Tree data is a common type of data that represents hierarchical relationships between objects.
- Tree data is used to represent a wide range of information, from organizational structures to computer file systems.
- The nodes of a tree data structure represent the objects in the hierarchy, while the edges represent the relationships between them.
- In data visualization, displaying hierarchical structures is commonly done using a tree diagram.



Nodes and edges of a tree

Tree Terminologies

Node

A node is an entity that contains a key or value and pointers to its child nodes.

The last nodes of each path are called **leaf nodes or external nodes** that do not contain a link/pointer to child nodes.

The node having at least a child node is called an **internal node**.

Edge

It is the link between any two nodes.

Root

It is the topmost node of a tree.

Height of a Node

The height of a node is the number of edges from the node to the deepest leaf (ie. the longest path from the node to a leaf node).

Depth of a Node

The depth of a node is the number of edges from the root to the node.

Height of a Tree

The height of a Tree is the height of the root node or the depth of the deepest node.

Degree of a Node

The degree of a node is the total number of branches of that node.

Forest

A collection of disjoint trees is called a forest.

Tree Applications

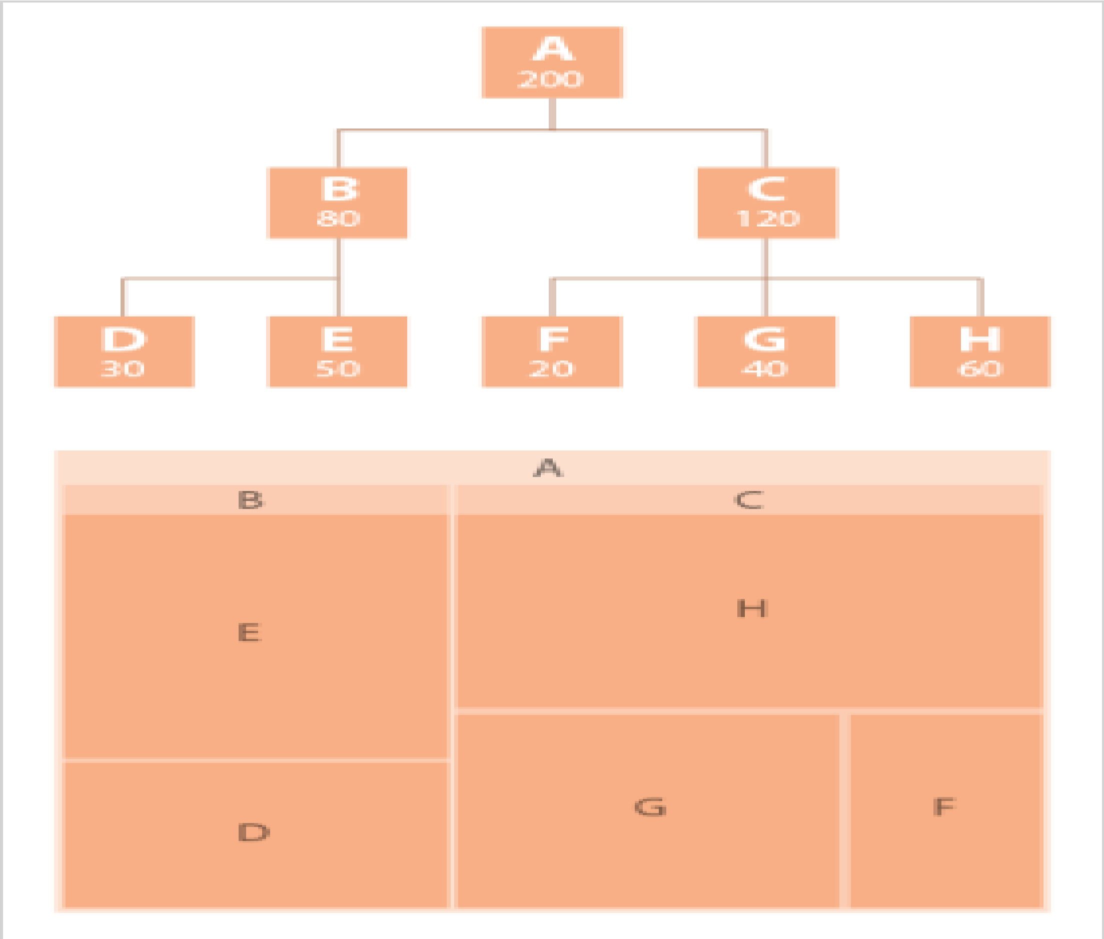
- Binary Search Trees(BSTs) are used to quickly check whether an element is present in a set or not.
- Heap is a kind of tree that is used for heap sort.
- A modified version of a tree called Tries is used in modern routers to store routing information.
- Most popular databases use B-Trees and T-Trees, which are variants of the tree structure we learned above to store their data

Tree Map

- The tree map functions as a visualization composed of nested rectangles.
- These rectangles represent certain categories within a selected dimension and are ordered in a hierarchy, or "tree."
- Quantities and patterns can be compared and displayed in a limited chart space.
- Tree maps represent part to whole relationships.

- Treemaps are an alternative way of visualising the hierarchical structure of a Tree Diagram while also displaying quantities for each category via area size. Each category is assigned a rectangle area with the subcategory rectangles nested inside.
- When a quantity is assigned to a category, its area size is in proportion to that quantity and any other quantities within the same parent category in a part-to-whole relationship. Also, the area size of the parent category is the total of its subcategories. If no quantity has been assigned to a subcategory, then its area is divided equally amongst the other subcategories within the parent category.
- The way rectangles are divided and ordered into sub-rectangles depends on the tiling algorithm used. Many tiling algorithms have been developed, but the "squarified algorithm", which keeps each rectangle as square-like as possible is the one commonly used.
- Ben Shneiderman originally developed Treemaps as a way of visualising a vast file directory on a computer, without taking up too much space on the screen. This makes Treemaps a more compact and space-efficient option for displaying hierarchies, that can give a quick overview of the hierarcal structure. Treemaps are also great at comparing the proportions between categories via their area size.

- The downside to Treemaps is that they doesn't show the hierarchal levels as clearly as other charts that visualise hierarchal data (such as a Tree Diagram or Sunburst Diagram).



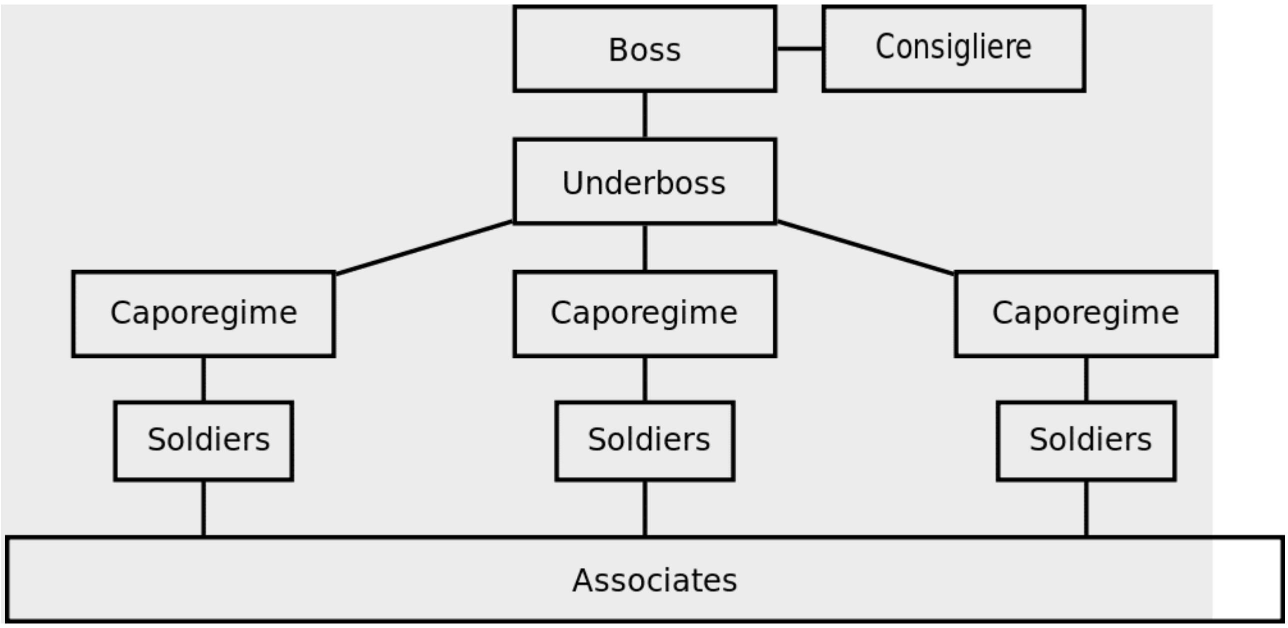
How to Show Hierarchical Data with Information Visualization (Displaying Hierarchical Structures)

Hierarchical data is essentially a specialized form of network data – in that while entities within the dataset do not have dependent relationships; they are all related to each other by the principle of containment. They, unlike standard data networks, do not use the principle of connection.

A hierarchy begins with a root entity. This might be the CEO of a company, the name of a book, the title of a folder, etc. and then the root entity has at least one “child node” and every further child node has zero or more children.

An entity which comes below another is a child node to the entity above. Similarly, an entity which comes above another is a parent node to the node below.

Hierarchical data is shown in tree graphs; so called because of their similarity to a tree’s structure (though a tree which has been turned upside down so that the root is at the top and the branches form below it).



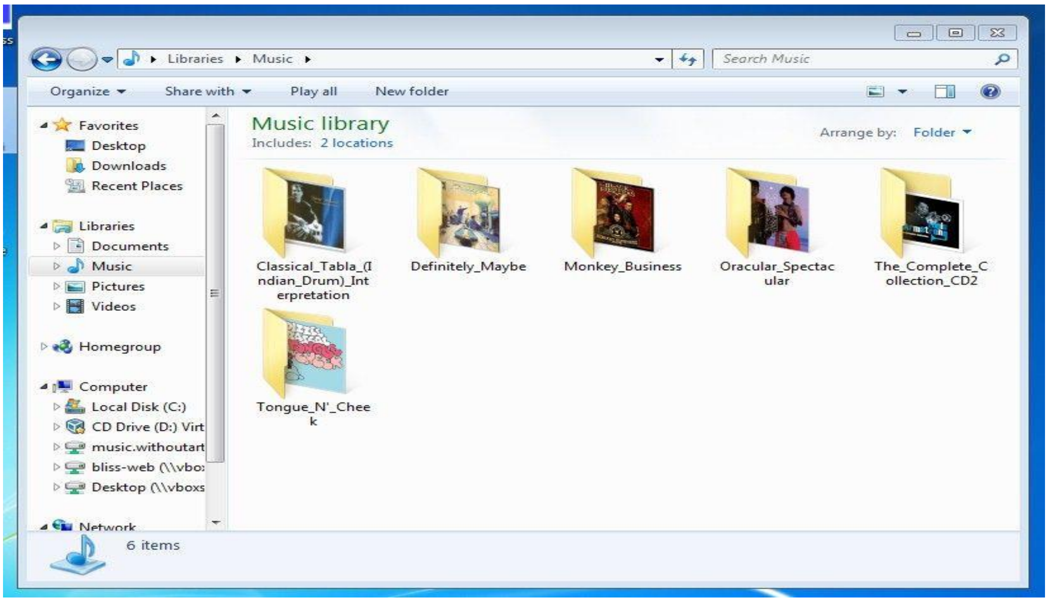
Above we see a simple tree diagram for the structure of a mafia family. The root entity is the boss of the family and the underboss is the first child entity. This is a very basic hierarchical relationship and it is possible to map much more complex hierarchies using information visualization techniques.

This diagram also includes a “sibling node” in the form of a consigliere who is not the boss of the organization but whose authority is equal to that of the boss.

The term “tree diagram” was coined by Noam Chomsky in his 1965 work; Aspects of the Theory of Syntax.

1. The File and Folder System
One of the most common hierarchies, which many of us deal with daily, is the computer file system. There is a root directory which then has a selection of child folders, which in turn have child folders, and in some or all of these folders there are files to be found.

The file tree is normally rendered in a visual format by the operating system. To provide a certain level of familiarity it uses images of the classic paper folder to connect the viewer with the property of the abstract file system used on the disc drive. This tree is interactive in nature and clicking on any given folder enables the user to determine what is inside that folder. However, it is also possible to use the command shell of an operating system to present this information textually too.



The Windows file system above is essentially a hierarchical tree and one with which many of us are already incredibly familiar with.

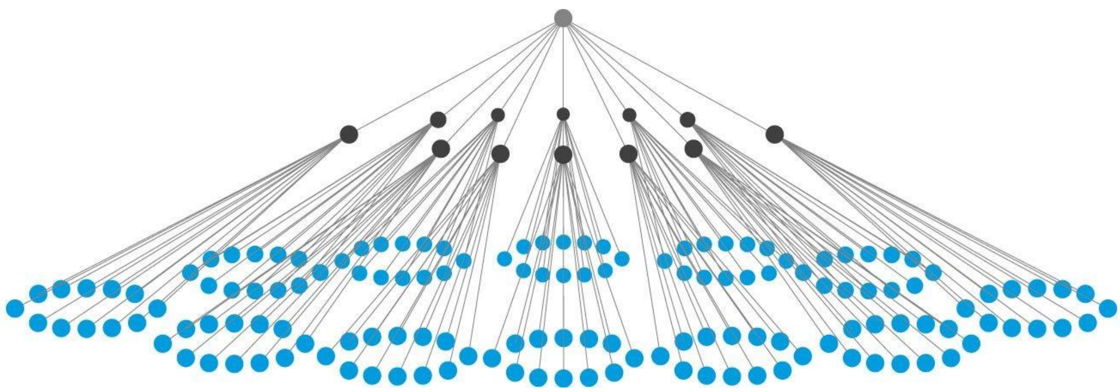
2. Cone Tree Diagram

A cone tree is a 3-D hierarchy model which was developed at Xerox PARC in the 1990s. It was designed to enable the representation of hierarchies with large multiples of nodes. The 3D means that the physical limitations of displaying complexity on a flat screen can, to some extent, be overcome.

It works by beginning with a root node and then arranging all the child nodes of that root equidistant from the parent. This forms a cone with some transparency. The process is then performed again and again for each set of child nodes and the diameter of the cone is reduced at each level of the hierarchy.

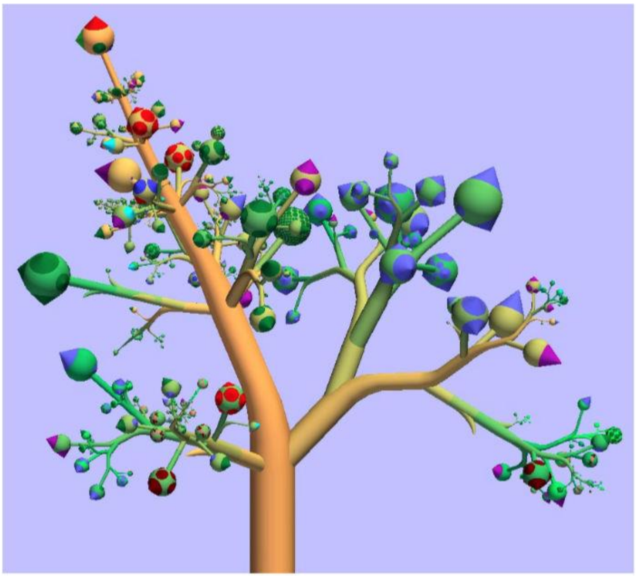
In general cone tree diagrams are generated using software which enables interactivity and a useful property of these diagrams is the ability to rotate them so that a particular child is occluded (hidden) by another child.

You can also get an idea of the numbers of child entities on any parent simply by observing the density of shading on any given cone. (The edges used to define the cone are properties of the children).



3. The Botanical Tree Diagram

The botanical tree diagram was invented by researchers at Eindhoven University of Technology in the Netherlands. They had noticed that the limitation of tree diagrams was that they can quickly become too complex to be functional and then they noticed that real trees had leaves. They also saw that it didn't matter how many leaves and branches there were on a tree – these were always distinct entities visually. So they extended the concept of tree diagrams by adding leaves and branches.



4. The Treemap Diagram

The treemap was invented by Ben Shneiderman of the University of Maryland in 1990. It represents hierarchies by using all the available space and in the form of nested rectangles. The rectangles can be defined in proportion to the "space" that they take up within the data set. These information visualizations can be very useful for comparing nodes and see patterns within them.

The math involved to create a treemap is quite complex but the good news is that you don't have to do that math; there are plenty of software packages available that can do this for you.

Above is a treemap of market share for different soft drinks. Comparing Coke, Coke Light, Ice Tea, Fanta and Dr. Pepper.

When Al Shalloway, the founder and CEO of Net Objectives, said; "Visualizations act as a campfire around which we gather to tell stories."

When to use?

Scatter Plot

Scatter plots instantly report a large volume of data. It is beneficial in the following situations:

- For a large set of data points given
- Each set comprises a pair of values
- The given data is in numeric form

When to Use a Scatterplot?

1. When you need to discover if your data expresses a trend
2. When you need to invest in inferential statistics and predict a future trend
3. When you need to determine the degree of correlation which exists in your dataset

When Not to Use a Scatterplot?

1. When you do not have paired numerical data but labels
2. When you need to understand the rate of change between individual data points.

Difference between Bar Chart and Histogram

The main difference between bar diagram and histogram is that the bars in a histogram are contiguous, whilst the bars in a bar chart are not and there is spacing between the bars.

Comparing the difference between bar chart and histogram in statistics, a bar chart or a histogram is a good way to show a large amount of data. Both a bar chart and a histogram are visual representations of groups of data, which is what they have in common. The table below will show you the difference between bar diagram and histogram.

Bar Chart	Histogram
Categorical information is represented by the bars in a bar graph.	The bar in a histogram is a graphical depiction of quantitative data sets, grouped under a certain parameter.
The bars in a bar graph do not touch each other.	The bars in a histogram touch each other.
Variables represented on a bar graph are discrete.	Variables represented on a histogram can be non-discrete.
The width of bars in a bar graph has to be the same.	The width of bars in a histogram need not be the same.
A bar chart is a graphical representation of data that compares distinct variables.	The histogram depicts the frequency distribution of a continuous variable.

Advantages

Advantages of bar chart are as follows

- A bar chart with digital or categorical data may be used.
- In a frequency distribution, the bar graph shows each type of data.
- It includes relative numbers or multi-category proportions
- Visually, you can sum up a broad collection of data.
- At a glance, estimate key values.
- Show close numbers or outline proportions.
- Wide visual data helps to explain patterns than tables better.

Advantages of the histogram are as follows

- It allows you to view a vast volume of data in a tabular form that is difficult to understand.
- They explain what time of occurrences values.
- Useful in deciding a process's power.
- This allows you to foresee the potential success of the system.
- You could display the frequency of the data incident along with an interval in the histogram.

Drawbacks of the bar chart are here

- This is the inverse of a bar chart.
- Only the frequencies for a data set are shown in a bar diagram.
- With the Bar Graph, you need more details.
- It does not disclose essential conclusions, triggers, implications or trends.

Drawbacks of the histogram are here

- You can't read correct values when data is classified.
- It just uses continuous knowledge. Two data sets cannot be contrasted conveniently in Histogram.
- The use of intervals in the histogram avoids any precision measurement of central propensity being to be computed.

Heat Maps

Who Uses Heat Maps?
There are four primary users for heat maps:

- User experience (UX) designers are those responsible for making sure people enjoy using the website or application or can find the information they're looking for. Heat maps help point out areas where users get frustrated or cannot find the resources they need.

- Marketing professionals want to turn website visitors into paying customers. Heat maps help them make sure visitors click on featured products, banner ads and calls to action.

- Data scientists review and analyze information from across the company, often stored in data warehouses. And heat maps can turn some of that information into simple-to-understand visualizations that present critical and actionable intelligence in ways that can be quickly absorbed.

- Product managers use heat maps when their product is the web application itself. For example, in a SaaS application, heat maps show the product manager important usage patterns in the app for which users are paying. Optimizing the application based on these patterns can help increase revenue and decrease customer churn.

When to Use Heat Maps

Heat maps can tell you a significant amount of helpful information, but they aren't a cure-all. They work best in the specific use-cases described below. Generally speaking, heat maps are best used when something has changed either in your customer base or your website and you want to understand how that affects useability.

- **A/B testing**
  For example, if you move a website element above the fold or change its color, does that increase clicks and conversions? Use a click map, and the difference becomes immediately apparent.

- **Website redesigns or refreshes**
  Your website might have a slick new design, but is it genuinely engaging customers? Use eye tracking or hover maps to understand what's drawing their attention.

- **Conversion rate optimization (CRO)**
  Web designers must ensure their site's aesthetics combine seamlessly with functionality. In other words, your site may be fancy, but are prospects clicking through? Scroll maps and click maps are valuable tools to gauge the effectiveness of a design.

- **Content marketing**
  Your blogs and white papers need to capture user's attention — at least until they find the CTA. Use scroll maps to understand how much content your users are reading, and then place your CTA buttons accordingly.

- **Usability testing**
  Above all else, your site should be accessible and functional to the average user. Heat maps can highlight areas where users could become confused while navigating your site.

Box Plots

We typically create box plots in one of three scenarios:

Scenario 1: To visualize the distribution of values in a dataset.

A box plot allows us to quickly visualize the distribution of values in a dataset and see where the five number summary values are located.

Scenario 2: To compare two or more distributions.

Side-by-side box plots allow us to visualize the differences between two or more distributions and compare the median values and the spread of values between distributions.

Scenario 3: To identify outliers.

In box plots, outliers are typically represented by tiny circles that extend beyond either whisker. An observation is defined to be an outlier if it meets one of the following criteria:

- An observation is less than Q1 – 1.5*(Interquartile range)
- An observation is greater than Q3 + 1.5*(Interquartile range)

By creating a box plot, we can quickly see whether or not a distribution has any outliers.