MDS651
Unit 3 – Attribute Data Visualization

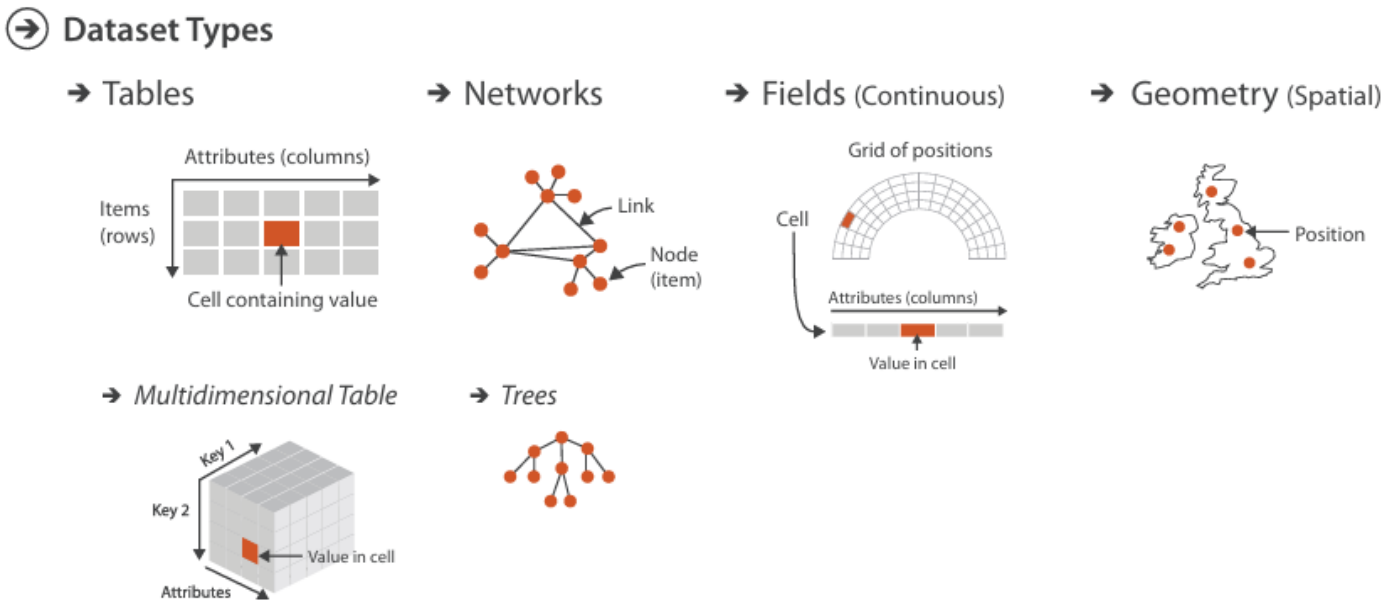
Dipesh Koirala

Outline

- Visualization of one, two and multi-dimensional data, Tabular data, quantitative values (scatter plot)
- Separate, Order and align (Bar, staked bar, dots and line charts),
- Tree data, Displaying Hierarchical structures,
- Graph data, Rules for graph drawing and labeling
- Time series data, Characteristics of time data, Visualization time series data, Mapping of time

Attribute Data Visualization

- A dataset is any collection of information that is the target of analysis.



- The four basic dataset types are tables, networks, fields and geometry.

Attribute Data Visualization

- Many datasets come in the form of tables that are made up of rows and columns, a familiar form to anybody.
- Each row represents an item of data, and each column is an attribute of the dataset.
- Each cell in the table is fully specified by the combination of a row and a column—an item and an attribute.

A	B	C	S	T	U
Order ID	Order Date	Order Priority	Product Container	Product Base Margin	Ship Date
3	10/14/06	5-Low	Large Box	0.8	10/21/06
6	2/21/08	4-Not Specified	Small Pack	0.55	2/22/08
32	7/16/07	2-High	Small Pack	0.79	7/17/07
32	7/16/07	2-High	Jumbo Box		7/17/07
32	7/16/07	2-High	Medium Box		7/18/07
32	7/16/07	2-High	Medium Box		7/18/07
35	10/23/07	4-Not Specified	Wrap Bag	0.52	10/24/07
35	10/23/07	4-Not Specified	Small Box	0.58	10/25/07
36	11/3/07	1-Urgent	Small Box	0.55	11/3/07
65	3/18/07	1-Urgent	Small Pack	0.49	3/19/07
66		5-Low	Wrap Bag	0.56	1/20/05
69		4-Not Specified	Small Pack	0.44	6/6/05
69		4-Not Specified	Wrap Bag	0.6	6/6/05
70	12/18/06	5-Low	Small Box	0.59	12/23/06
70	12/18/06	5-Low	Wrap Bag	0.82	12/23/06
96	4/17/05	2-High	Small Box	0.55	4/19/05
97	1/29/06	3-Medium	Small Box	0.38	1/30/06
129	11/19/08	5-Low	Small Box	0.37	11/28/08
130	5/8/08	2-High	Small Box	0.37	5/9/08
130	5/8/08	2-High	Medium Box	0.38	5/10/08
130	5/8/08	2-High	Small Box	0.6	5/11/08
132	6/11/06	3-Medium	Medium Box	0.6	6/12/06
132	6/11/06	3-Medium	Jumbo Box	0.69	6/14/06
134	5/1/08	4-Not Specified	Large Box	0.82	5/3/08
135	10/21/07	4-Not Specified	Small Pack	0.64	10/23/07
166	9/12/07	2-High	Small Box	0.55	9/14/07
193	8/8/06	1-Urgent	Medium Box	0.57	8/10/06
194	4/5/08	3-Medium	Wrap Bag	0.42	4/7/08

Attribute Data Visualization

Attribute

- An attribute is some specific property that can be measured, observed, or logged.
- **E.g.,** attributes could be salary, price, number of sales, protein expression levels, or temperature.

- An item is an individual entity that is discrete, such as a row in a simple table or a node in a network.

- Attribute types are categorical, ordinal or quantitative.

Visualization of One Dimensional Data

- One-dimensional data refers to a dataset where each individual item or data point has only **one single attribute, or value.**
- There's only one variable being measured or observed.
- Given a one-dimensional sequence of univariate data or only one value per data item.
- **E.g.,** temperature, age, sales of category etc.

	Tax
0	3500
1	4200
2	2700
3	5600
4	3100
5	7500
6	6600
7	2900
8	8100
9	4700

Visualization of One Dimensional Data

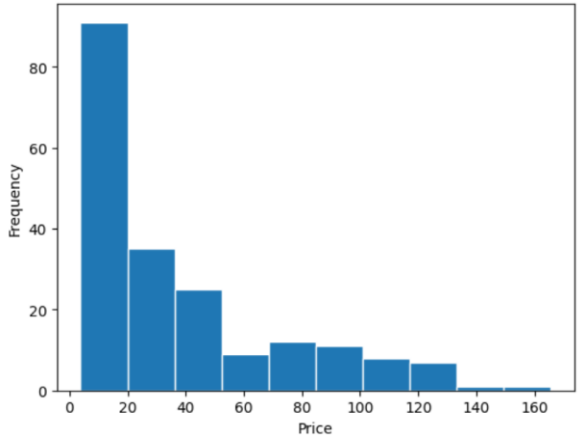
- The primary goal of visualizing one-dimensional data is to understand its **distribution, spread or frequency:**
  - Where are the values concentrated?
  - Is the data symmetrical or skewed?
  - What is the count of the values?

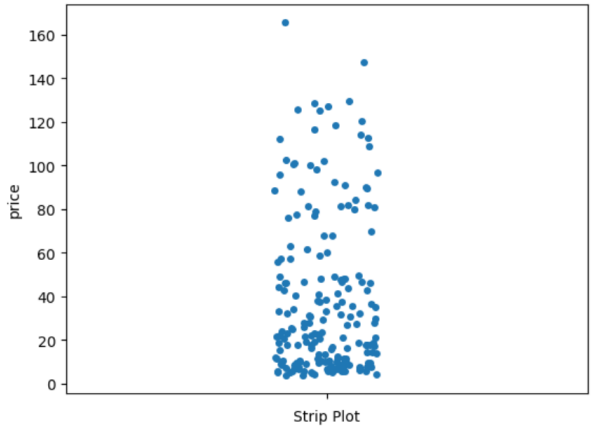
Visualization of One Dimensional Data

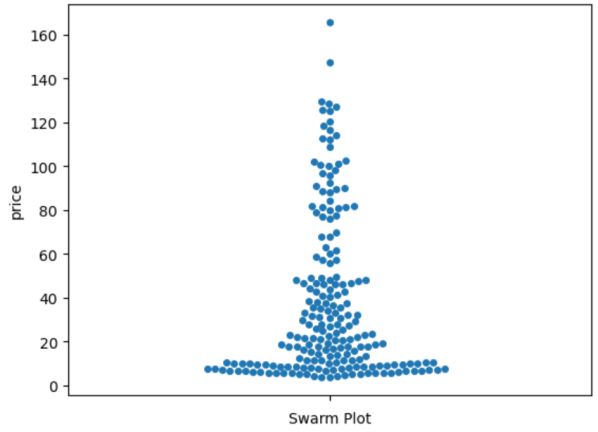
- Histogram
- Boxplots
- Density Plots
- Strip Plots
- Violin Plots
- Count Plots

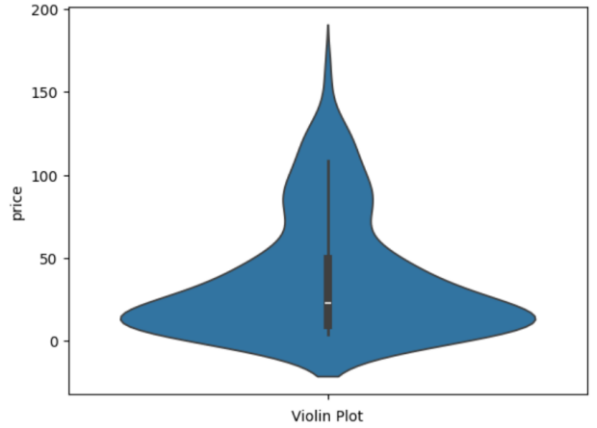
Visualization of One Dimensional Data

- Plots









Visualization of Two Dimensional Data

- Two-dimensional data refers to a dataset where each individual item or data point is characterized by **exactly two distinct attributes or values.**
- There are two variable being measured or observed.
- observing and recording values for two different characteristics.
- These variables can be both numerical, both categorical, or a mix of one numerical and one categorical.

	carat	price
0	0.23	326
1	0.21	326
2	0.23	327
3	0.29	334
4	0.31	335
5	0.24	336
6	0.24	336
7	0.26	337
8	0.22	337
9	0.23	338

Visualization of Two Dimensional Data

Focus on Relationship and Comparison

- **Relationships:** How do the two variables interact or influence each other? Is there a correlation?
- **Comparisons:** How do different groups in one variable differ in terms of the other variable?
- **Trends:** If one variable represents a sequence or time, what patterns or changes can be observed in the other variable over that sequence?

Visualization of Two Dimensional Data

Both Numerical Variables:

1. Scatter Plots:

- is a fundamental type of data visualization that displays the relationship between two numerical variables.
- Each item or observation is plotted as a marker i.e., a dot, a circle, a cross.
- Shows:
  - **Correlation:** Positive, negative, or no correlation.
  - **Patterns:** Linear, non-linear, clusters.
  - **Outliers:** Points far removed from the general trend.

Visualization of Two Dimensional Data

Both Numerical Variables:

2. Line Plots:

- is a type of data visualization that displays information as a series of data points called 'markers' connected by straight line segments.

- used when one of the numerical variables represents a sequence, most commonly **time**.
- The points are connected by lines to emphasize the progression or trend.
- To show how a variable changes over time or another continuous ordered variable.

Visualization of Two Dimensional Data

One Variable is Categorical, one is Numerical

1. Bar charts:

- If the categorical variable defines distinct groups, and the numerical variable is a measure for each group (e.g., sum, average, count), bar charts are ideal for comparison.

2. Box Plots (Grouped):

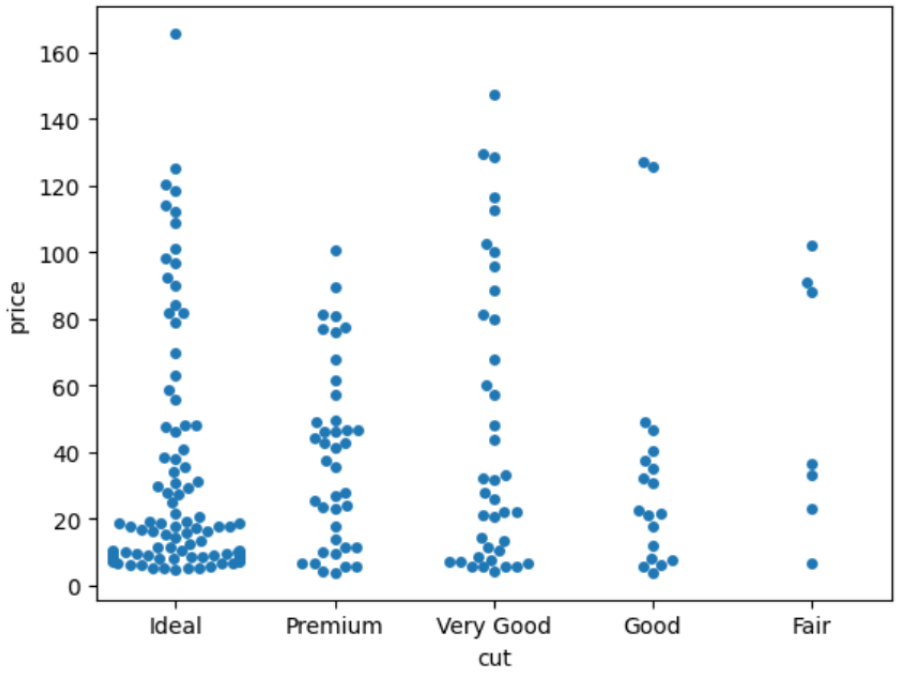
- used for comparing the distribution of a numerical variable across different categories.

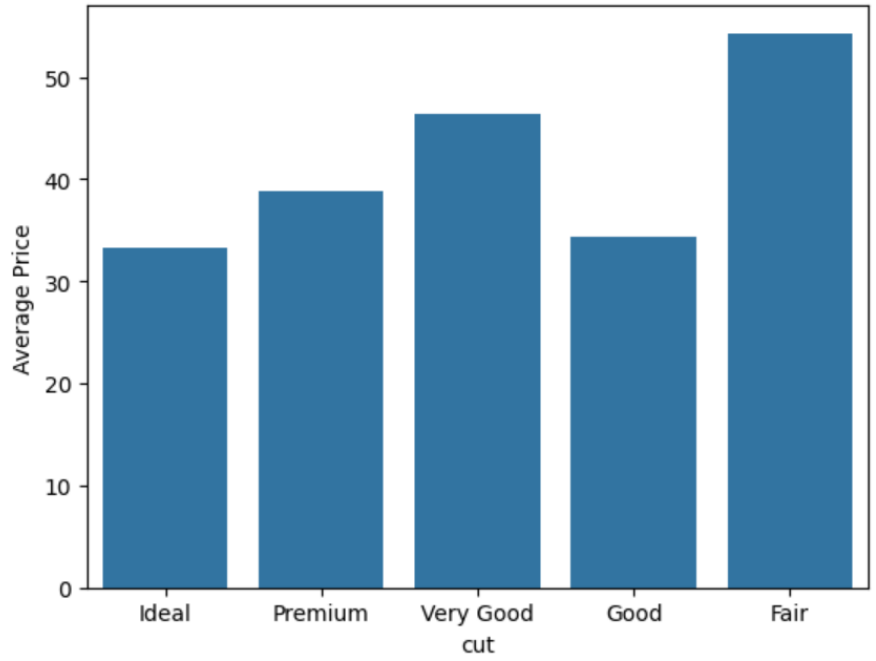
3. Violin Plots:

- similar to box plots, but they also show the underlying density distribution for each category, which provides richer view of the data's shape within each group.

Visualization of Two Dimensional Data

	cut	price
0	Ideal	8.73
1	Ideal	9.83
2	Very Good	7.77
3	Very Good	28.02
4	Ideal	9.00
5	Ideal	5.44
6	Very Good	13.48
7	Ideal	98.17
8	Ideal	10.06
9	Ideal	34.17





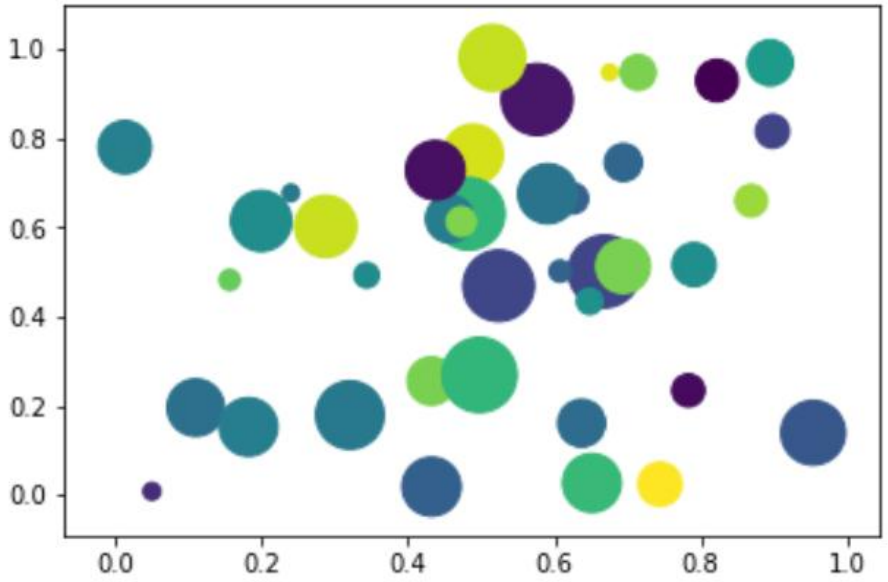
Visualization of Multi Dimensional Data

- Multidimensional data, also known as high-dimensional data, refers to datasets where each individual observation or data point is characterized **by three or more distinct attributes, features, or variables.**

Technique:

- Adding Dimensions through Visual Attributes (Encoding):
- Color, Size, Shape/Marker

E.g., Bubble Charts



Visualization of Multi Dimensional Data

Technique:

- Stacking
- Grouping
- Faceting: Create a grid of multiple 2D plots, where each plot represents a subset of the data based on one or more additional categorical dimensions.
- Parallel Coordinate plots
- Radar charts

Visualization of Multi Dimensional Data

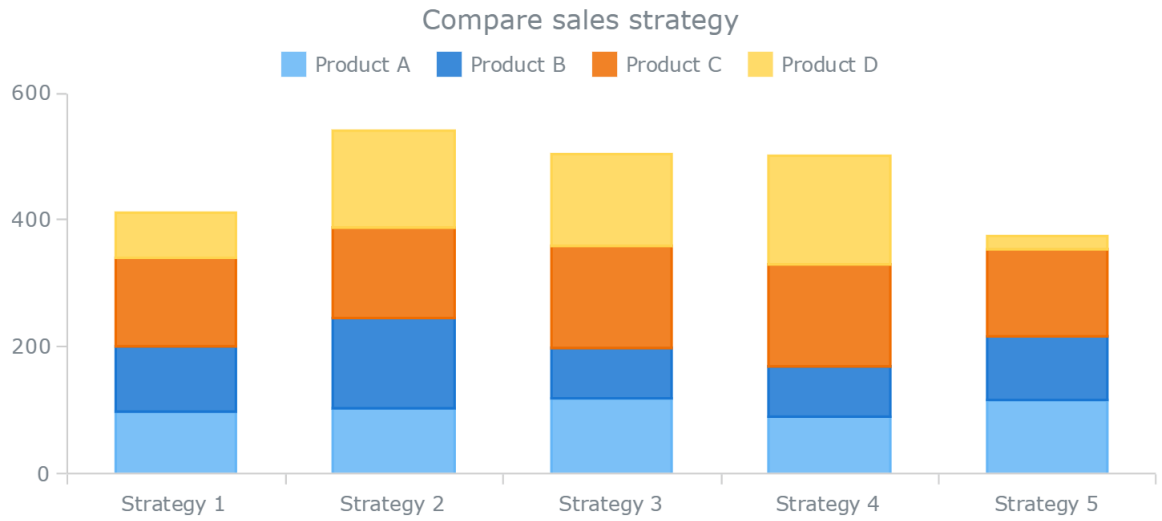
- Example:

Cookie	Raw	Burnt	Chewy	Round	Intact
Chocolate Chip	7	15	20	4	12
Sugar	28	18	17	8	14
Oatmeal	19	13	12	12	19
Peanut Butter	23	14	9	17	18
Gingersnap	10	19	2	24	31

Visualization of Multi Dimensional Data

Stacked Bar charts:

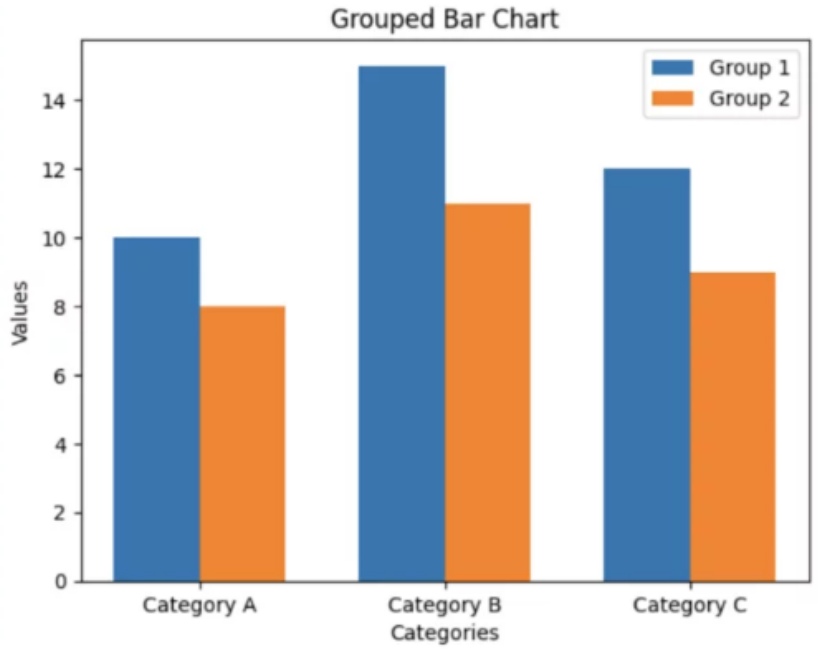
- Is a charts which displays parts of a whole for different categories.
- Each bar represents a total , and that bar is divided into segments, with each segment representing the count or proportion of a sub-category.
- It is used to show the composition, i.e., how the parts contribute to the whole within each primary category.



Visualization of Multi Dimensional Data

Grouped Bar Charts

- Is a chart which displays bars for categories of a secondary variable side-by-side within each category of a primary variable. Each group of bars corresponds to a single category of the primary variable.
- is to compare the numerical values of the sub-categories *across* the primary categories.

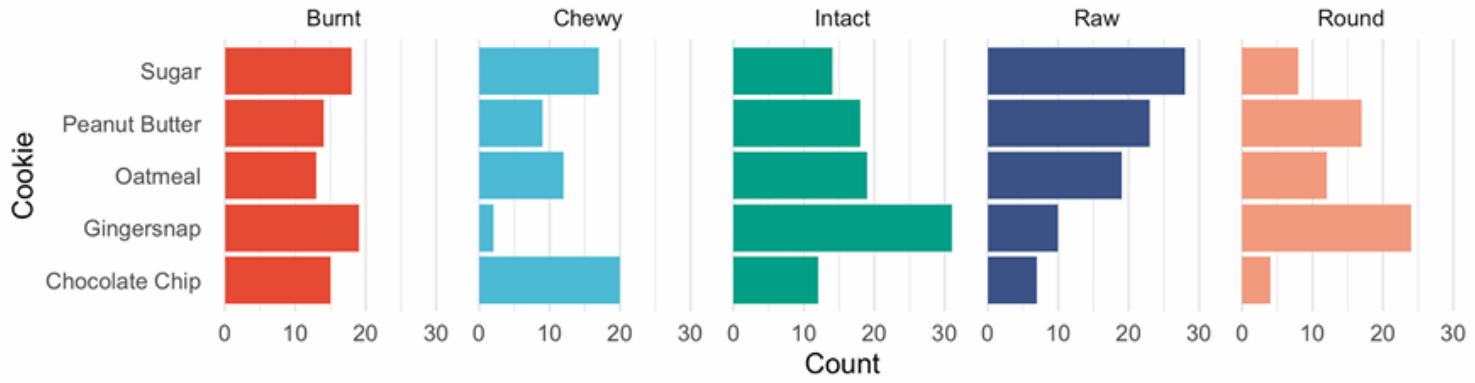


Visualization of Multi Dimensional Data

Faceting

- is a technique in where a **separate, identical plot is created for each sub category**. Each of these individual plots is called a "facet" or a "panel."

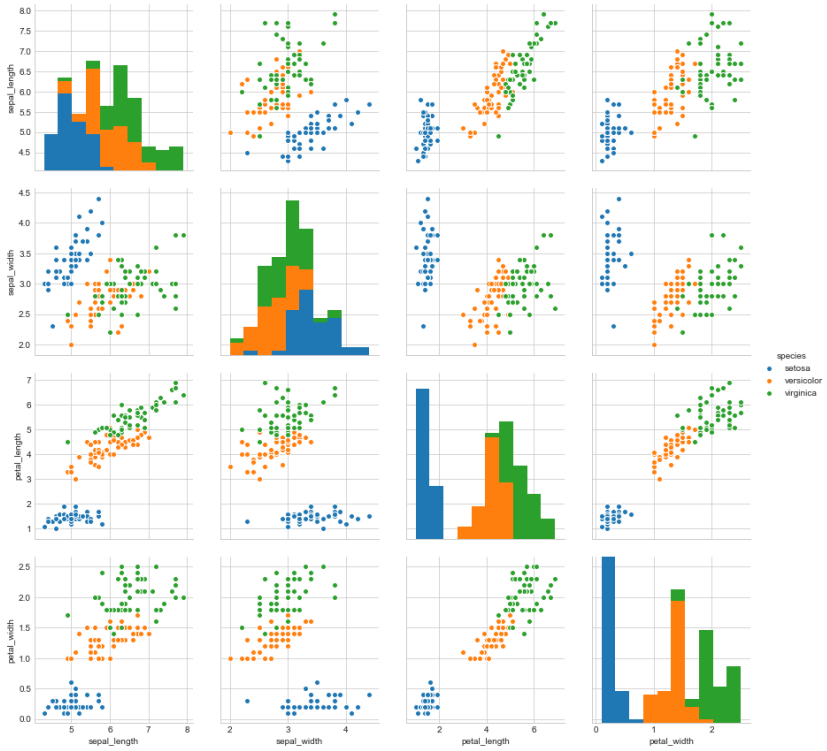
- Faceting creates 'small multiples' – multiple, related visualizations displayed together.



Visualization of Multi Dimensional Data

PairGrid

- PairGrid is another type of grid for plotting pairwise relationships in a dataset.
- By drawing the same plot types across an entire dataframe, it enables detailed analysis of how every variable relates to all others.



- https://seaborn.pydata.org/generated/seaborn.PairGrid.html

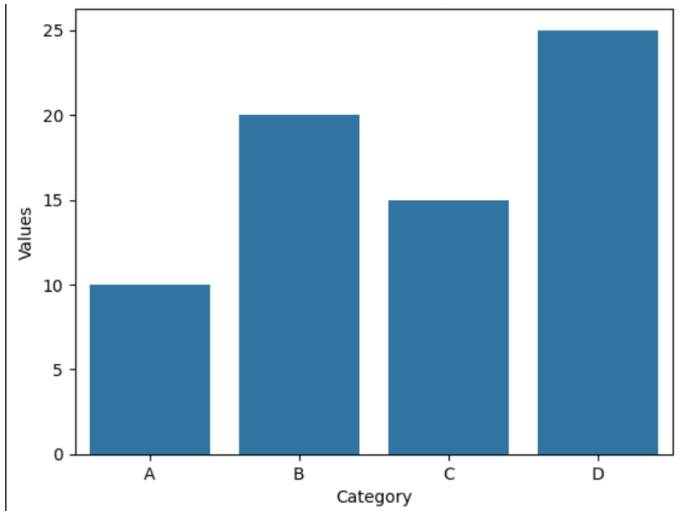
Separate, Order and align

- 'Separate', 'order' and 'align' are design choices for how to arrange and display attribute data to effectively communicate insights.

Separate:

- refers to the act of creating distinct visual regions or groups for different categories within the attribute data.
- Used to distinguish between different categories or groups of data.
- It is achieved with visual encodings:

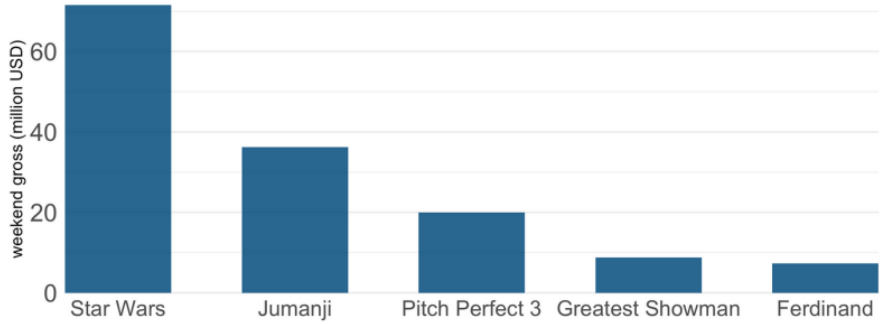
  Spatial separation, color hue, shape

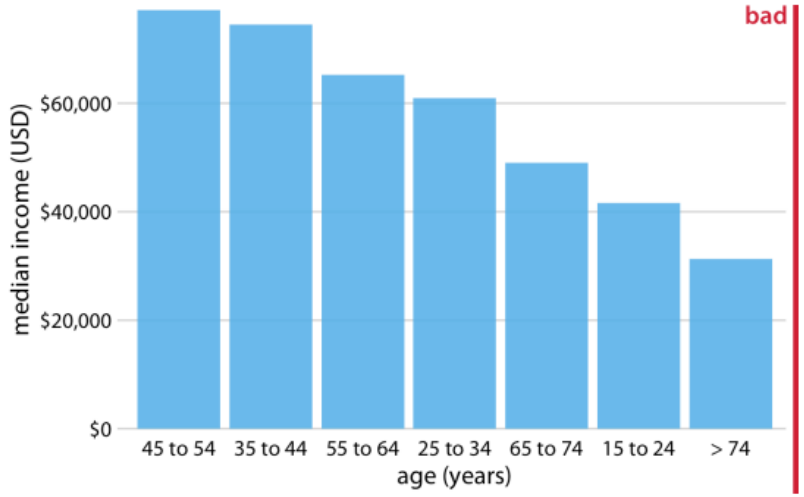


Separate, Order and align

Order:

- refers to arranging data elements based on a specific attribute's value or a predefined sequence.
- To reveal trends, patterns, and hierarchies.
- To make comparisons easier by presenting data in a logical flow.

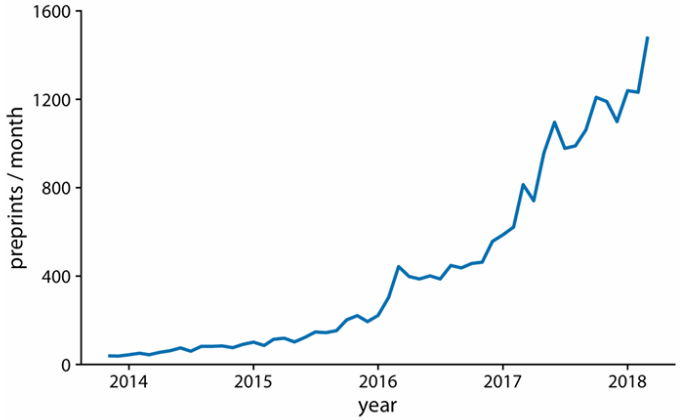


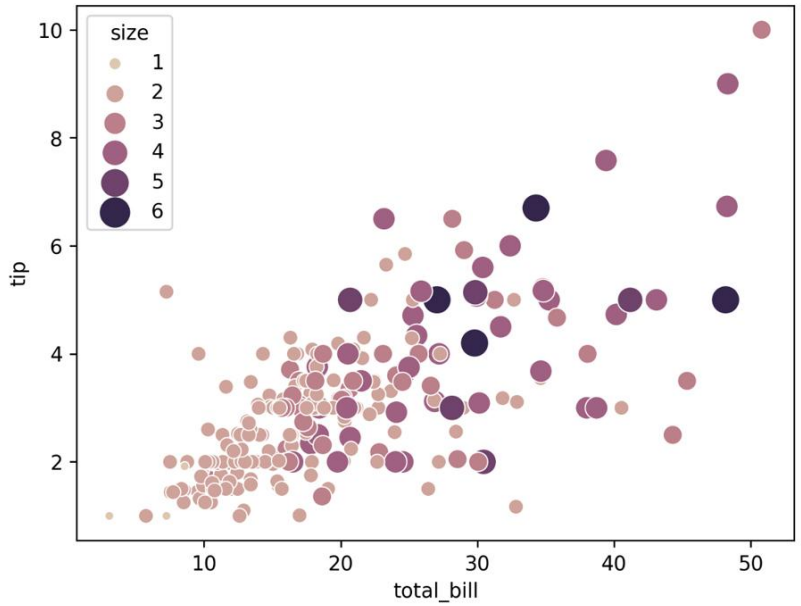


Separate, Order and align

Order:

- **Position on an axis:** Values are placed along a continuous axis. E.g., a time series on an x-axis, or a numerical value on a y-axis
- **Size:** Larger values are represented by larger visual elements. E.g., larger bubbles in a bubble chart, longer bars in a bar chart
- Darker or more intense colors represent higher values.

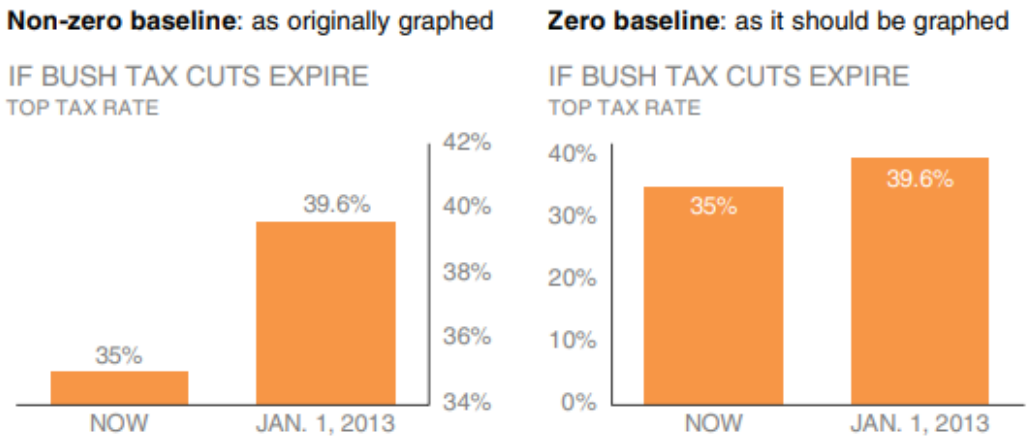




Separate, Order and align

Align:

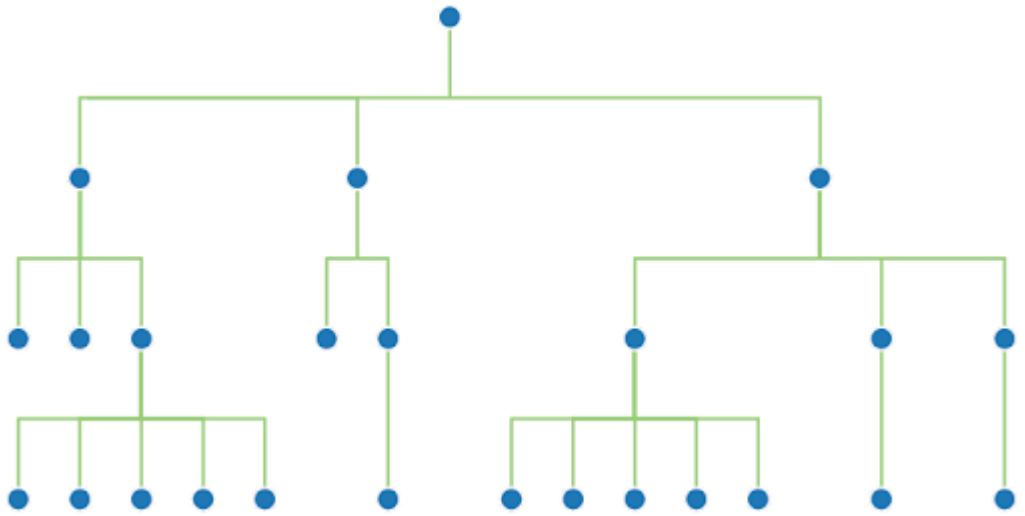
- refers to positioning visual elements so that their baselines, edges, or other key features are consistent across the visualization. This often involves using a common scale or reference point.
- To enable precise comparisons between different data points or categories.



Tree data, Displaying Hierarchical structures

- Tree data refers to datasets that inherently represent a hierarchical structure.

- convey relational information, e.g., how data items are related to each other. These interrelationships can take many forms:
  - part/subpart, parent/child or other hierarchical relation



Tree data, Displaying Hierarchical structures

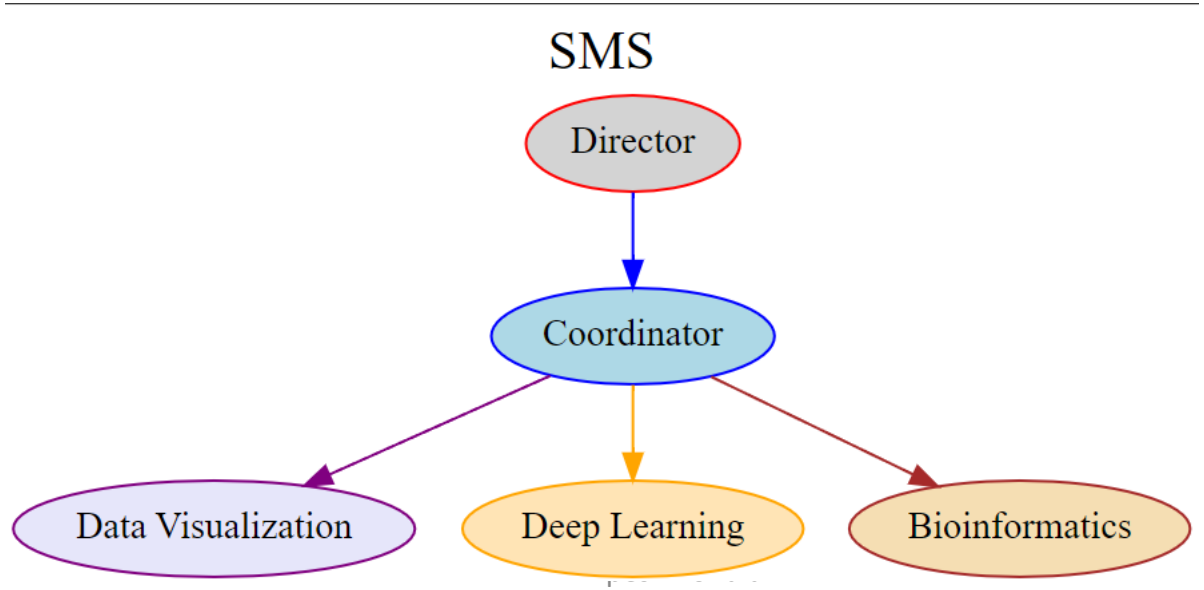
- the data elements are organized in levels, with hierarchical relationships between them.

	Category	Item	Type	Sales
0	Electronics	Laptop	Gaming Laptop	60000
1	Electronics	Laptop	Business Laptop	40000
2	Electronics	Laptop	Ultrabook	20000
3	Electronics	Smartphone	Android Phone	50000
4	Electronics	Smartphone	iPhone	35000
5	Electronics	Tablet	None	40000
6	Appliances	Washing Machine	None	30000
7	Appliances	Refrigerator	None	50000
8	Furniture	Office Chair	None	20000
9	Furniture	Desk	None	25000

Tree data, Displaying Hierarchical structures

Node-link Diagrams

- The most common visual encoding for tree and network data is with *node-link diagrams.*
- Nodes are drawn *as point marks* and the links connecting them are drawn *as line marks.*
- This idiom uses connection marks to indicate the relationships between items.



Tree data, Displaying Hierarchical structures

Node-link Diagrams

Used library:

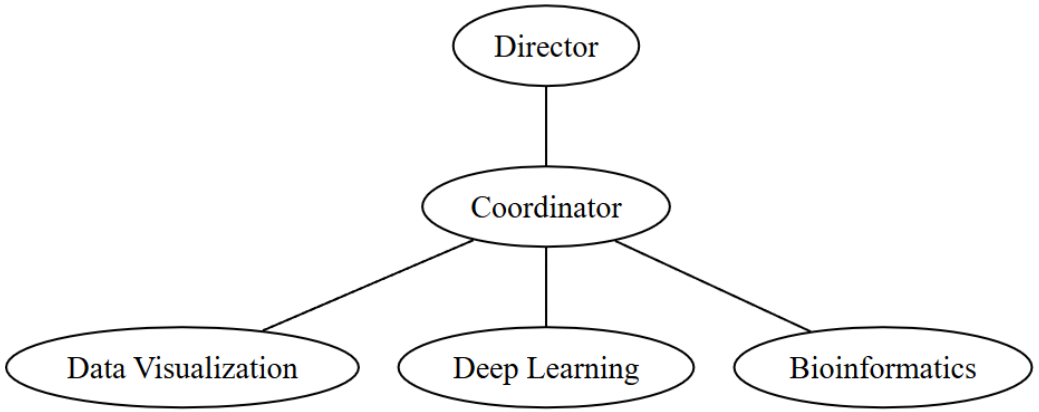
```
pip install graphviz
```

```
from graphviz import Graph

dot = Graph()
```

Tree data, Displaying Hierarchical structures

Node-link Diagrams

```
dot.node('1','Director')
dot.node('2','Coordinator')
dot.node('3','Data Visualization')
dot.node('4','Deep Learning')
dot.node('5','Bioinformatics')

dot.edge('1','2')
dot.edge('2','3')
dot.edge('2','4')
dot.edge('2','5')

dot
```

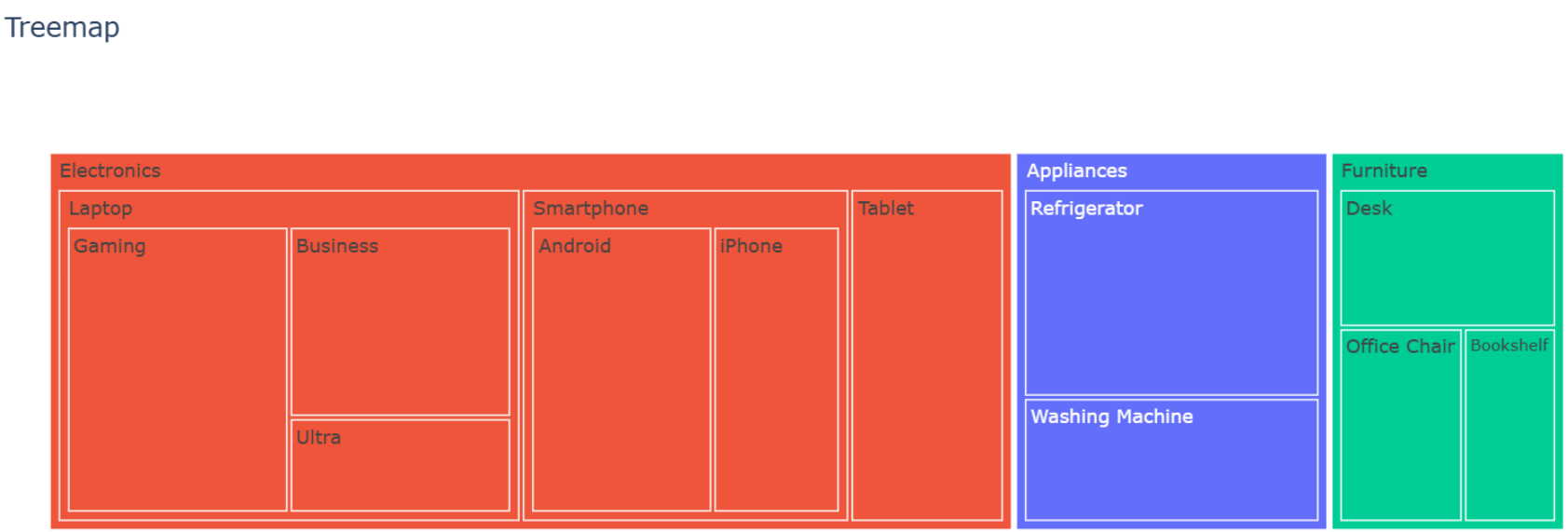



```
dot.node('1','Director',color='red', style='filled', fillcolor='lightgrey')
dot.node('2','Coordinator', color='blue', style='filled', fillcolor='lightblue')
dot.node('3','Data Visualization',color='purple', style='filled', fillcolor='lavender')
dot.node('4','Deep Learning',color='orange', style='filled', fillcolor='moccasin')
dot.node('5','Bioinformatics',color='brown', style='filled', fillcolor='wheat')

dot.edge('1','2',color='blue')
dot.edge('2','3',color='purple')
dot.edge('2','4',color='orange')
dot.edge('2','5',color='brown')

dot.attr(label='SMS',labelloc='t',fontsize='20')
```

Tree data, Displaying Hierarchical structures

Treemap

- is a space-filling visualization that represents hierarchical data using nested rectangles.
- The entire area of the chart represents the root of the hierarchy, and it is recursively subdivided into smaller rectangles for each child node.
- The nesting of rectangles visually represents the parent-child relationships in the data.



Tree data, Displaying Hierarchical structures

Treemap

Used library:

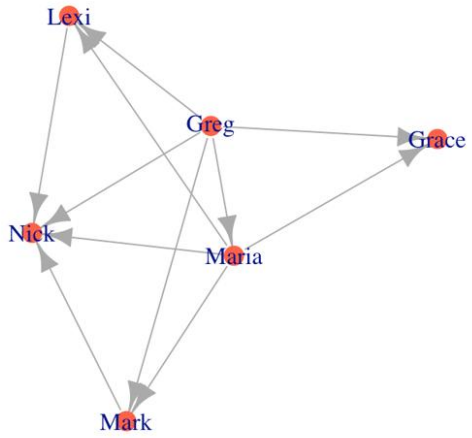
```
import plotly.express as px
```

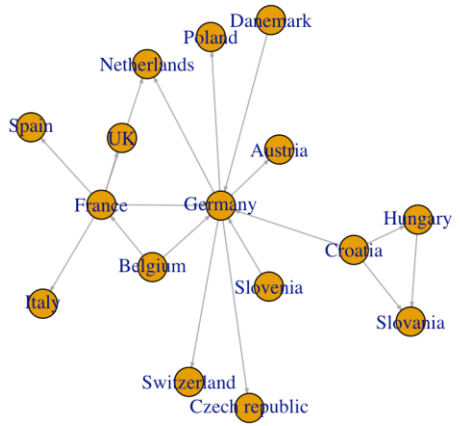
```
fig = px.treemap(
    df,
    path=['Category', 'Item', 'Type'],
    values='Sales',
    color='Category',
    title='Tree map'
)

fig.show()
```

Graph data

- Graph/network visualization is the process of visually representing relationships between data points.
- the goal is not just to show individual data points, but to represent the relationships and structure of a network.
- Trees are just one type of a more general representation of relations called a graph.

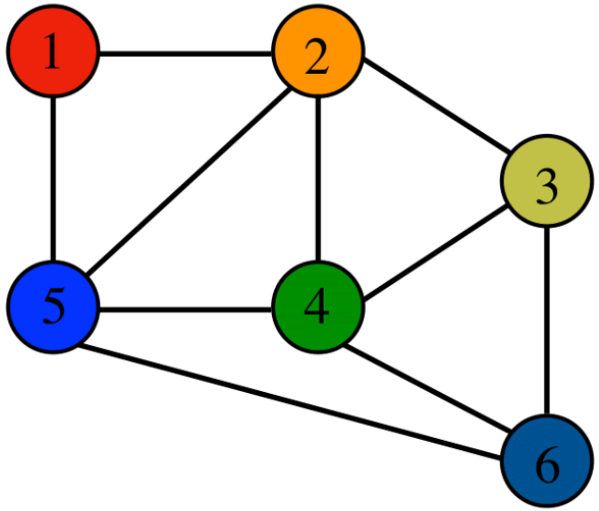
- **E.g.,** connectedness, such as cities connected by roads or computers connected by networks

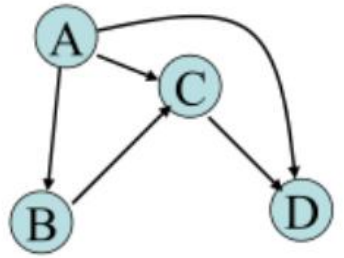




Graph data

1. Node-Link Diagrams (or Network Diagrams)
2. Matrix Representations





↗	A	B	C	D
A		X	X	X
B			X	
C				X
D				

Graph data

Node Link Diagrams

- **Nodes:** Represent individual entities, objects, or data points within the network. These can have associated attributes (e.g., size, color) to convey additional information.

- **Links/Edges:** Represent the relationships or interactions between nodes. Links can be directed or undirected, and can also have attributes (e.g., thickness, color) to represent the strength or type of relationship.

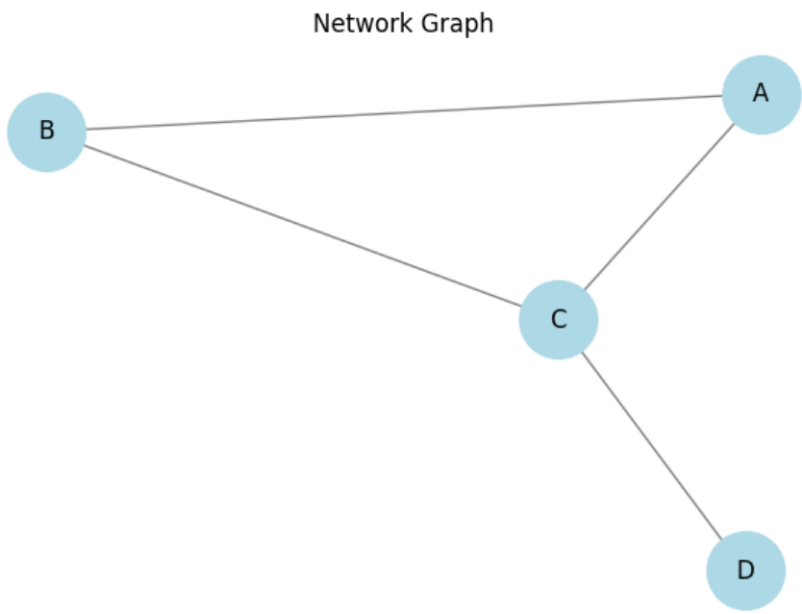
- **Layout Algorithms:**
- layout algorithms are employed to arrange the nodes and links in a visually comprehensible manner, aiming to minimize clutter, reveal clusters, and highlight important structural features. Common layouts include force-directed layouts, hierarchical layouts, and circular layouts.

Graph data

Used library:

```
import networkx as nx
import matplotlib.pyplot as plt
```

```
G = nx.Graph()

G.add_nodes_from(['A', 'B', 'C', 'D'])

G.add_edges_from([
    ('A', 'B'),
    ('A', 'C'),
    ('B', 'C'),
    ('C', 'D')
])

plt.figure(figsize=(6, 4))
nx.draw(G, with_labels=True, node_color='lightblue', node_size=1500, edge_color='gray')

plt.title('Network Graph')
plt.show()
```




```
# layout: spring, circular, shell, random, spectral
pos = nx.spring_layout(G)

nx.draw(G, pos, with_labels=True, node_color='lightblue', node_size=1500, edge_color='gray')
plt.title('Spring Layout')
plt.show()
```

Rules for graph drawing

- minimize line crossings
- maintain a pleasing aspect ratio
- minimize the total area of the drawing
- minimize the total length of the edges
- minimize the number of bends in the edges
- minimize the number of distinct angles or curvatures used
- strive for a symmetric structure.

Graph data

Applications:

- Social Networks, Computer Networks, Biological Networks, Organizational Structures

Limitations:

- can become challenging to interpret with very large and dense networks, as visual clutter can obscure meaningful patterns.
- Techniques like filtering, grouping, or hierarchical layouts are used to address this.

Time series data

- is a sequence of data points indexed or listed in time order.
- each data point has a corresponding timestamp, and the order of these timestamps is meaningful and important.
- the data points are collected at successive, equally spaced points in time (e.g., daily, monthly, annually) or sometimes at irregular intervals.

E.g.,

- Daily Stock prices
- Monthly average temperatures
- Annual GDP figures
- Quarterly sales data

	Date	Gold_Price
0	01-10-2000	4538
1	01-11-2000	4483
2	01-12-2000	4541
3	01-01-2001	4466
4	01-02-2001	4370
5	01-03-2001	4269
6	01-04-2001	4267
7	01-05-2001	4441
8	01-06-2001	4400
9	01-07-2001	4380

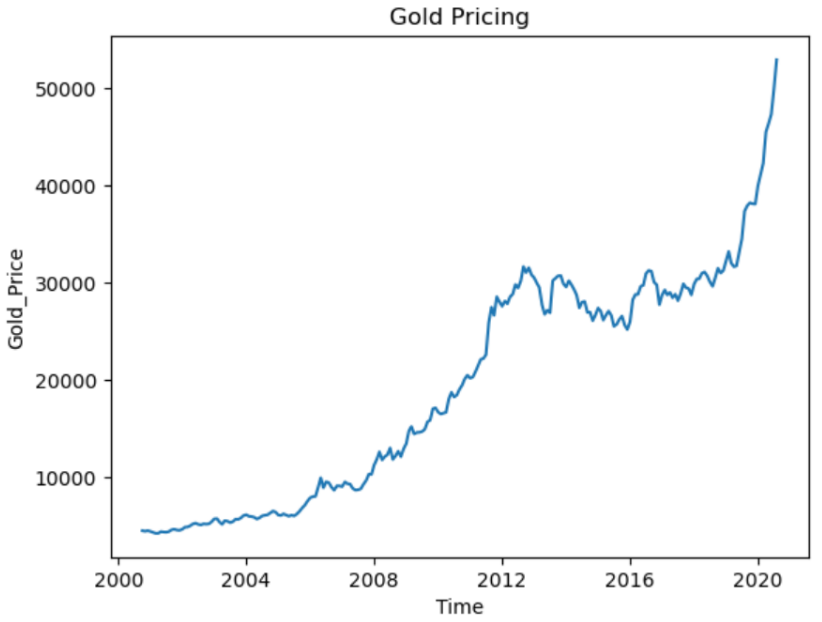
Time series data

Visualization of time series data

Line Plots

- are the **most common and effective** way to visualize time series data.
- Time is plotted on the horizontal (x) axis, and the measured variable is plotted on the vertical (y) axis.
- Data points are connected by lines.

- Used for revealing trends, seasonality, cycles, and highlighting peaks and troughs over time.



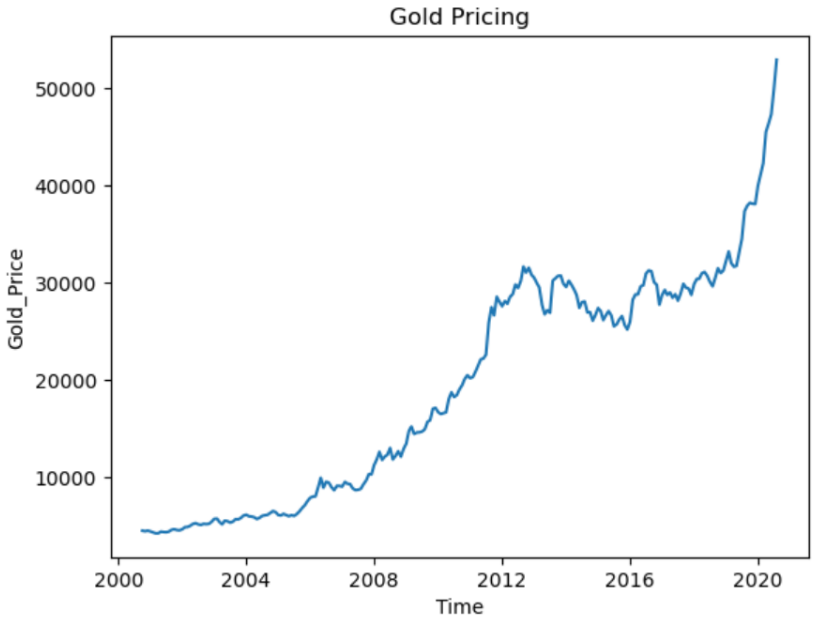
Time series data

Visualization of time series data

Mapping of Time

- Mapping of time refers to how the temporal dimension of the data is translated into visual properties on a chart.

- The primary mapping is usually the horizontal X-axis
- Time is almost universally mapped to the x-axis because of our natural tendency to read time from left to right.
- Color can be used for separating different categories in single line chart.



Time series data

Characteristics of Time Series Data

- Time series data is generally comprised of different characteristics or components that characterize the patterns and behavior of the data over time.

1. Trends
2. Seasonality
3. Cycles
4. Noise

Time series data

Characteristics of Time Series Data

1. Trend

- is the long-term increase or decrease in the data over time. It's the underlying direction of the data.
- **E.g.,** A general upward trend in global temperatures over decades, world population over last five years

2. Seasonality

- This refers to predictable and recurring patterns or cycles in the data that occur at regular intervals (e.g., daily, weekly, monthly, quarterly, yearly). These patterns are often related to calendar events or natural cycles.
- **E.g.,** Retail sales increasing during the holiday season each year, or higher electricity consumption during summer months.

Time series data

Characteristics of Time Series Data

3. Cycles

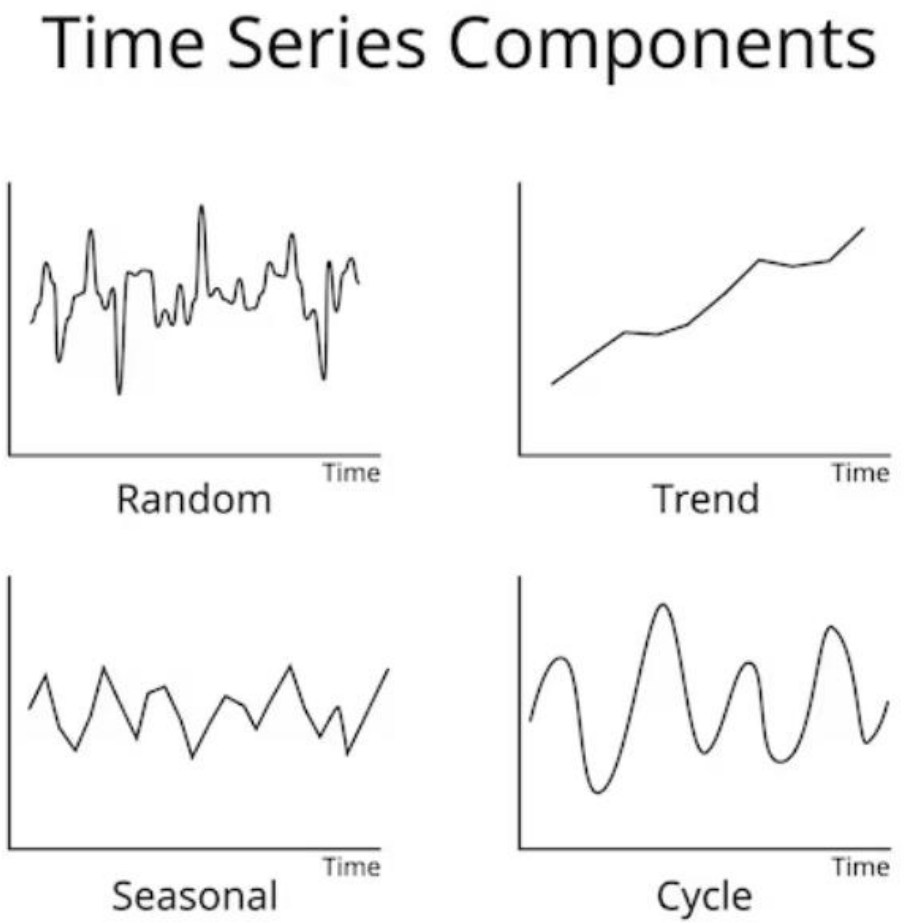
- These are patterns that are not of fixed frequency and usually span longer periods than a season (e.g., business cycles that might last several years).
- They are often less predictable than seasonal patterns.
- **E.g.,** Economic recessions and expansions

4. Noise/Randomness:

- This is the unpredictable, random fluctuations in the data that cannot be explained by trend, seasonality, or cyclical components.
- **E.g.,** A sudden, unexpected spike in website traffic due to a news event.

Time series data

Characteristics of Time Series Data



Attribute Data Visualization

Conclusion

1. Visualize Amounts

   Bar plots, Grouped bars, Stacked bars

2. Visualize Distributions

   Histogram, Density plot, Box plot, Violin plot, Swarm plot

3. Visualize Proportions

   Pie charts, grouped bars, stacked bars

4. x – y Relationships

   Scatterplot, Bubble chart

5. Tree/Graph

   Node link Diagrams, Tree map

6. Time Series / Sequential

   Line charts

End of Unit 3

Thank you