DATA SCIENCE

UNIT – IV

DATA VISUALIZATION

DATA VISUALIZATION & TYPES:

Data Visualization is defined as the pictorial representation of the data to provide the fact-based analysis to decision-makers. Typically text data might not be able to reveal the pattern or trends needed to recognize data. Data visualization is broadly classified into 6 different types. They are:

I. **Temporal**: Data for these types of visualization should satisfy both conditions: data represented should be linear and should be one dimensional. These types of visualization are represented through lines that might overlap and also have a common start and finish data point.
   - Scatter Plots – uses dots to represent a data point.
   - Pie chart - includes circular graphics where the arc length signifies the magnitude.
   - Line graphs - Like the scatter plot, the data is represented by points, except joined by lines to maintain continuity

II. **Hierarchical**: These types of visualizations portray ordered groups within a larger group. In simple language, the main intuition behind these visualizations is the clusters can be displayed if the flow of the clusters starts from a single point.
   - Tree diagram - In a tree diagram, the hierarchical flow is represented in the form of a tree as the name suggests
   - Tree Map - The tree is represented in the form of rectangles closely packed. The area signifies the quantity contained.

III. **Network**: The visualization of these type connects datasets to datasets. These visualizations portray how these datasets relate to one another within a network.

- Matrix chart - This type of visualization is widely used to find the connection between different variables within themselves. For example, correlation plot
- Word cloud - This is typically used for representing text data. The words are closely packed, and the size of the text signifies the frequency of the word.

IV. **Multidimensional**: In contrast to the temporal type of visualization, these types can have multiple dimensions. In this, we can use 2 or more features to create a 3-D visualization through concurrent layers.

- Scatter plots - In multi-dimensional data, we select any 2 features and then plot them in a 2-D scatter plot.
- Stacked bar graphs - The representation shows segmented bars on top of each other.

V. **Geospatial**: These visualizations relates to present real-life physical location by crossing it over with maps and it may be a geospatial or spatial map.

- Flow map - Movement of information or objects from one location to another is presented where the size of the arrow signifies the amount.
- Choropleth Map - The geospatial map is colored on the basis of a particular data variable
- Heat Map - These are very similar to Choropleth in the geospatial genre but can be used in areas apart from geospatial as well.

TYPES OF DATA

i. **CATEGORICAL DATA**: Categorical data represents characteristics. Therefore it can represent things like a person's gender, language etc. Categorical data can also take on numerical values (Example: 1 for female and 0 for male).

ii. **NOMINAL DATA**: Nominal values represent discrete units and are used to label variables, that have no quantitative value. Nominal data that has no order. Therefore if you would change the order of its values, the meaning would not change. Example: Gender – Male/Female. To visualize nominal data you can use a pie chart or a bar chart.

iii. **ORDINAL DATA**: Ordinal values represent discrete and ordered units. It is therefore nearly the same as nominal data, except that it's ordering matters. This is the main

limitation of ordinal data, the differences between the values is not really known. Because of that, ordinal scales are usually used to measure non-numeric features like happiness, customer satisfaction and so on. Example: Educational Background – 1. Elementary 2. High School 3. Undergraduate 4. Postgraduate. Ordinal data can be visualized with pie and bar charts.

iv. **NUMERICAL DATA**:

- **DISCRETE DATA** - Discrete data means its values are distinct and separate. In other words, discrete data can only take on certain values. This type of data can't be measured but it can be counted. Example: Number of heads in 100 coin flips.
- **CONTINUOUS DATA** - Continuous Data represents measurements and therefore their values can't be counted but they can be measured. Example: Height of a person, which can be described by using intervals on the real number line. To visualize continuous data, you can use a histogram or a box-plot.
  - **INTERVAL DATA** - Interval values represent ordered units that have the same difference. Therefore Interval data is used when we have a variable that contains numeric values that are ordered and where we know the exact differences between the values. Example: Temperature of a given place.
  - **RATIO DATA** - Ratio values are also ordered units that have the same difference. Ratio values are the same as interval values, with the difference that they do have an absolute zero. Examples: height, weight, length.

ENCODING:

Encoding in data visualization means translating the data into a visual element on a chart or map through position, shape, size, symbols and color. Encoding must be done correctly because only then users can understand what is shown in the map or chart. Most visual relationships can be visually expressed with points, lines and bars.

Example: Consider the standard column chart shown In Fig.4.1.

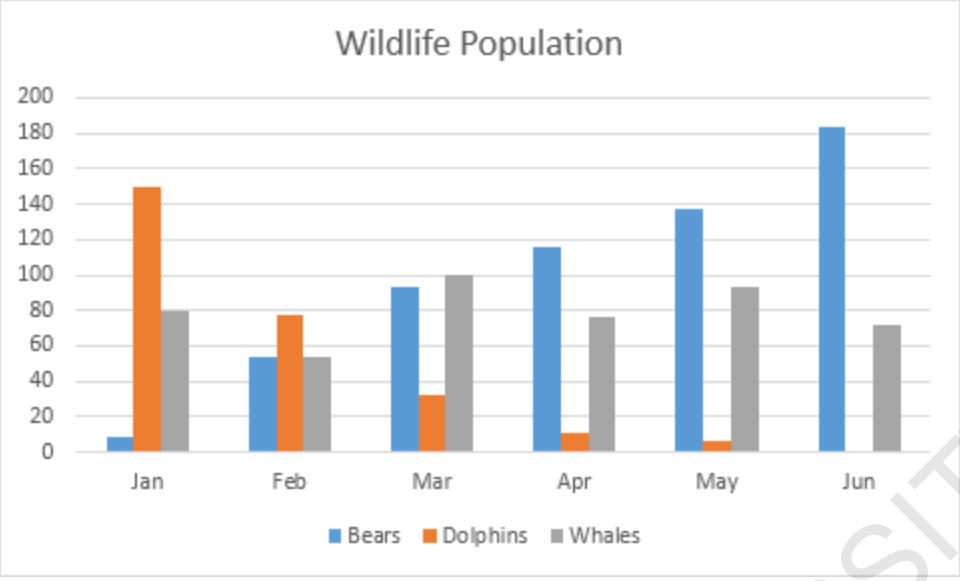

Fig.4.1. Column Chart

The chart in Fig.4.1 is encoded in the following ways:

- Colour: colour of bears, dolphins, whales
- Size: every time animal count increases, the column height increases.
- Grouping: Every time month changes, new cluster of columns must be created.

In general, the different ways to encode data are: Size, shape, colour, grouping, area, position, saturation, line pattern, line weight, angle, connections.

VISUAL ENCODING:

The various dimensions of data can be encoded with appropriate visual properties. Many visual properties may be used to encode multiple data types. The data types are: quantitative (relates to quantities and possible to do arithmetic on the data), ordinal (can be ordered), categorical, relational data. There are two factors that will indicate whether a visual property is best suited to a data type. They are:

i. **Natural Ordering** – is well suited to representing quantitative differences (27, 33, 41), or ordinal differences (small, medium, large, enormous).
   Examples:
   a. Naturally ordered - position. Length, line thickness, weight, brightness (luminance), intensity (saturation)

b. Not naturally ordered – shape, texture, line style (dotted, dashed, solid), color (hue)

ii. **Number of distinct values** – The number of distinct values the user will be able to perceive, differentiate, or remember.
Examples: shape, position, numbers

The common visual properties to select an appropriate encoding for a data type are given in Fig.4.2 and the grouping of Visual properties by the types of data they can be used to encode is shown in Fig.4.3.

PLANAR ENCODING

Planar encoding is as simple as the laying of axis, like the x & y axis in a simple line chart.

RETINAL ENCODING

To represent data in 3 or more variables, retinal encoding comes into the picture. Size, texture, shape, orientation, color gradient and color hue are some examples.

VISUAL STRUCTURES:

Mapping Data to Visual form includes the following steps:

raw data -> data tables ->visual structures ->views.

The basic building blocks of visual structures are:

i. **Position** - The 4 possible axes are: unstructured, nominal, ordinal, quantitative.
ii. **Marks** - The 4 types of marks are: points, lines, areas, volumes.
iii. **Connections** – show a relationship between objects.
iv. **Enclosure** – indicates related objects.
v. **Retinal properties** - include colour, size, texture, shape, orientation.
vi. **Temporal encoding** – changes in mark position and their retinal properties.

Example	Encoding	Ordered	Useful values	Quantitative	Ordinal	Categorical	Relational
	position, placement	yes	infinite	Good	Good	Good	Good
1, 2, 3; A, B, C	text labels	optional alpha or num	infinite	Good	Good	Good	Good
	length	yes	many	Good	Good		
	size, area	yes	many	Good	Good		
	angle	yes	medium	Good	Good		
	pattern density	yes	few	Good	Good		
	weight, boldness	yes	few		Good		
	saturation, brightness	yes	few		Good		
	color	no	few (<20)			Good	
	shape, icon	no	medium			Good	
	pattern texture	no	medium			Good	
	enclosure, connection	no	infinite			Good	Good
	line pattern	no	few				Good
	line endings	no	few				Good
	line weight	yes	few		Good		

Fig. 4.2. Common visual properties to select an appropriate encoding for a data type

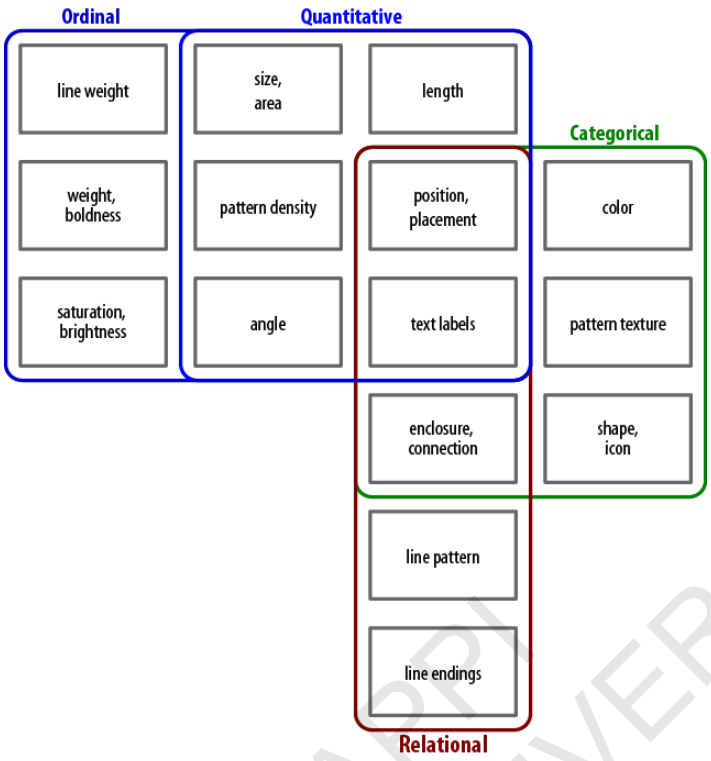

Fig.4.3 Grouping of Visual properties by the types of data they can be used to encode.

The encodings of common chart elements are:

Points

- *Position:* since points generally have no meaningful height and width, they convey position with more precision than lines and bars
- *Size:* points can use size as an encoding (see bubble charts), however, since it gives substance to the points' height and width, the precision of their positions is reduced
- *Orientation:* if the shapes of your points are not symmetrical, then orientation could be an encoding, but it's not often used, and for good reason
- *Color Saturation/Hue:* scatter plots often use these two encodings, but there are wrong ways to do so (which we will see in an upcoming post exclusively about color)
- *Shape:* while it's possible to encode a categorical variable with a point's shape, it quickly leads to a cluttered graph and should be avoided
- *Texture:* since the small size of points make their position so precise, textures are too hard to distinguish and should be avoided

Lines

- *Position:* lines are most effectively used to connect two categorical data points, so the positions of the endpoints encode some quantitative property of the data points

- *Size:* lines have no width, so size here means length, which represents the amount of time between sampling (which is commonly constant throughout the graph, but not necessarily so)
- *Orientation:* the main reason to use lines is to compare changes from data point to data point, and the orientation of the line (slope) depicts this change, so orientation is a very salient encoding for line charts
- *Color Saturation/Hue:* it's possible to have multiple lines representing different categories of data, in which case color hue can encode these different categories, but color saturation is not appropriate for categorical variables and so should be avoided
- *Shape:* if a line were a different shape, then it wouldn't be a line, therefore shape is not applicable
- *Texture:* since lines have no width, texture would translate to dotted lines, which are too hard to distinguish, so texture as an encoding should be avoided

Bars

- *Position:* often bars are fixed to a meaningful zero value on one axis, so the position of its free endpoint can represent a quantitative value
- *Size:* since bars often have one endpoint fixed at zero, the size is a double encoding coupled with position, which is why bars are a great tool for emphasizing individual values; the main exception to this idea is a stacked bar chart since only the bottom bar is fixed to zero in that case, and size is the only encoding of the magnitude of the quantitative value in that case
- *Orientation:* Bars should be perpendicular to the axis of the categorical variable they're representing, so orientation is fixed and therefore not applicable
- *Color Saturation/Hue:* color hue is useful to distinguish bars from each other when multiple bars are used per category to represent different values, but saturation would be a much less effective encoding for bars
- *Shape:* just like with lines, a bar is defined by its shape, so shape is not an applicable encoding
- *Texture:* bars are weighty enough to support texturing, but often it's more effective to use color to this end

REDUNDANT ENCODING:

Redundant encoding means, after encoding the main dimensions of data, the unused visual properties can be used to redundantly encode some existing, already-encoded data dimensions. The advantage of redundant encoding is that using more channels to get the same

information into the human brain can make acquisition of that information faster, easier, and more accurate.

Examples:

- If lines are differentiated by ending (arrows, dots, etc.), consider also changing the line style (dotted, dashed, etc.) or color.
- If values are encoded by placement, consider redundantly encoding the value with brightness, or grouping regions with color.

SUGGESTIONS FOR TYPE OF CHARTS TO BE USED FOR VISULIZATION

Figure 4.4 shows the suggestions for the type of chart to use in data visualization.

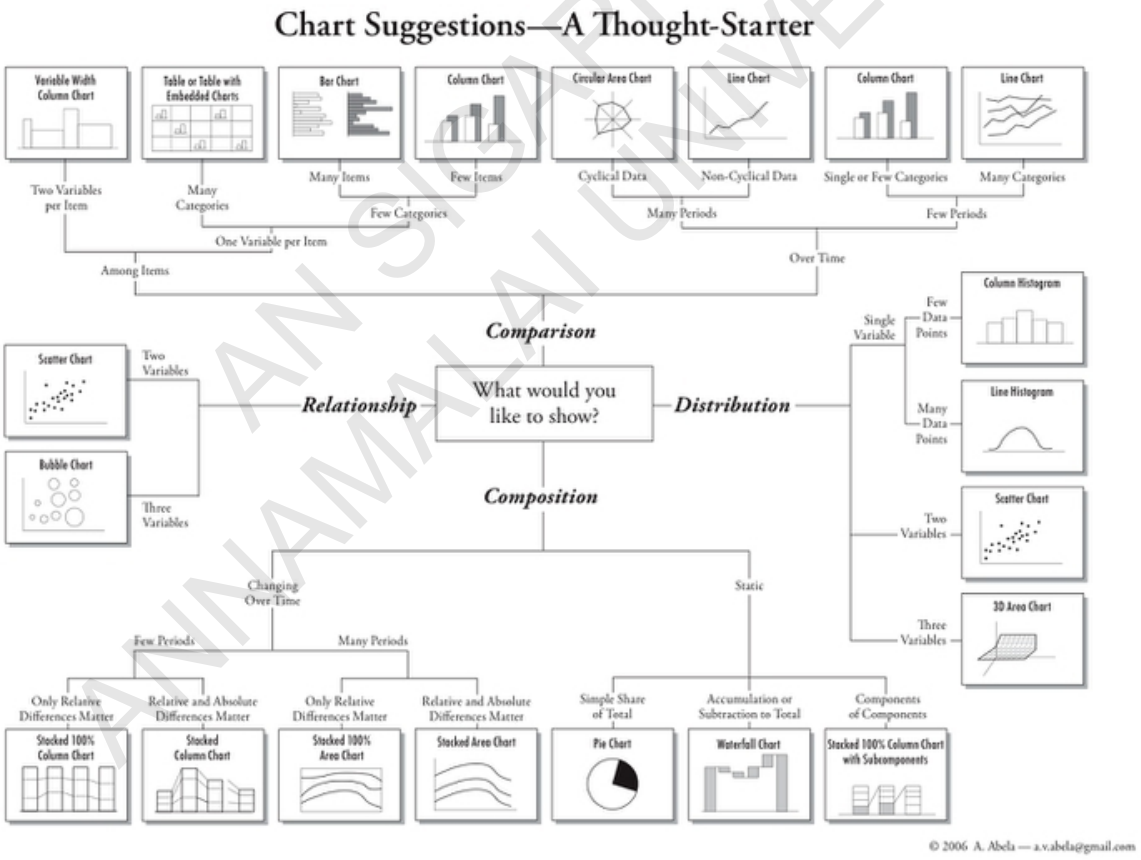

Fig. 4.4. Suggestions for Choice of Charts

Bar and Column Charts: Applying the perceptual hierarchy of visual cues, bar and column charts are usually the best options for simple comparisons. Vertical columns often work well when few items are being compared, while horizontal bars may be a better option when there are many items to compare. Also, Bar or column charts can also be used to illustrate change over time.

Pie Chart: To depict composition: parts of the whole, pie chart can be used.

Square chart: Square chart can be used where an encoding of area seems to beat length for accuracy.

Tree map - uses area to encode the size of parts of the whole, and can be effective to display "nested" variables — where each part of the whole is broken down into further parts.

Stacked column charts - to simultaneously show change in composition over time and change in the size of the whole.

COLOR ENCODING:

Color falls low on the perceptual hierarchy of visual cues, but it is often deployed to highlight particular elements of a chart, and sometimes to encode data values. Color is often used to encode the values of categorical data. There are three colour schemes often used in encoding.

i. **Qualitative color schemes:** used where the aim is to pick colors that will be maximally distinctive, as widely spread around the color wheel as possible.
ii. **Sequential color schemes:** When using color to encode continuous data, it usually makes sense to use increasing intensity, or saturation of color to indicate larger values. These are called "sequential" color schemes.
iii. **Diverging color schemes:** In some circumstances, you may have data that has positive and negative values, or which highlights deviation from a central value. Here, you should use a "diverging" color scheme, which will usually have two colors reasonably well

separated on the color wheel as its end points, and cycle through a neutral color in the middle.

RETINAL VARIABLES:

The retina in the human eye is sensitive to graphical properties independent of their position. These properties were called “retinal variables”. A designer can relate to component of a visualization with one of these variables:

- Size
- Value
- Texture
- Colour hue
- Orientation
- Shape

Any of these variables can be used in the representation of any component - or, the other way around – data attributes can be mapped to these properties in a visualization. But not each variable is suited to every component. At the level of information visualization, the two planar dimensions are able to represent two components of the information, in (geographic) maps they these components are the space. To introduce a third component of information, the usage of the retinal variables is necessary.

A perceptual classification known as the levels of organization of retinal variables, is based on the following four attributes:

• **Associative perception** - Associative perception is useful when one is seeking to equalize a variation, and to group correspondences with all categories of this variation combined. Examples: Shape, orientation, colour and texture as associative, whereas value and size are dissociative

• **Selective perception** - Selective perception is used to give an answer to the question: “Where is a given category”. The eye must be able to isolate all the elements of this category immediately.

Example: Shape is not selective at all, nor orientation when represented by area.

• **Ordered perception** - When comparing two or several orders, ordered perception must be used. Example: Shapes, orientations and colors are not ordered, whereas value, size and texture should be ordered.

• **Quantitative perception** - quantitative perception is given when it is required to define numerically the ration between two signs and group homogenous signs.

In data visualization, addition to the data, encoded through the visual cues, various items of charts that must be kept in mind are:

- Title and subtitle - These provide context for the chart.
- Coordinate system - For most charts, this is provided by the horizontal and vertical axes, giving a cartesian system defined by X and Y coordinates; for a pie chart it is provided by angles around a circle, called a polar coordinate system.
- Scale - Labeled tick marks and grid lines can help the audience read data values.
- Labels - Each axis must be labelled. Also other labels that may be necessary to explain the message may be included.
- Legend – To explain the color or shape used to encode data.
- Source information - Usually given as a footnote.

DATA VISULAIZATION PRINCIPLES

- Éncoding data using visual cues - When displaying quantities, position and length are preferred over angles and/or area. Brightness and colour are harder to quantify than angles.
- Know when to include 0 - When using bar plots, it is misinformative not to start the bars at 0. This is because, barplot typically implies the length is proportional to the quantities being displayed. By avoiding 0, relatively small differences can be made to look much bigger than they actually are.
- Do not distort quantities - Instead of using area, position and length can be used to avoid distorting quantities.

- Order categories by a meaningful value -When one of the axes is used to show categories, order the categories alphabetically when they are defined by character strings. If they are defined by factors, they are ordered by the factor levels.
- Show the data
- Ease comparisons
  - Use common axes - Since there are so many points, it is more effective to show distributions rather than individual points. An important principle is to keep the axes the same when comparing data across two plots.
  - Align plots vertically to see horizontal changes and horizontally to see vertical changes - In histograms, the visual cue related to decreases or increases in height are shifts to the left or right, respectively: horizontal changes. Aligning the plots vertically helps to see this change when the axes are fixed.
  - Consider transformations - the use of the log transformation in cases where the changes are multiplicative. Population size was an example in which we found a log transformation to yield a more informative transformation.
  - Visual cues to be compared should be adjacent –
  - Use color - The comparison becomes even easier to make if we use color to denote the two things we want to compare.
  - Plots for two variables - n general, you should use scatterplots to visualize the relationship between two variables.
  - Encoding a third variable - Categorical variables can be encoded with color and shape. For continuous variables, color, intensity, or size can be used. When selecting colors to quantify a numeric variable, choose between two options: sequential and diverging. Sequential colors are suited for data that goes from high to low. High values are clearly distinguished from low values. Diverging colors are used to represent values that diverge from a center.
- Avoid pseudo-three-dimensional plots.
- Avoid too many significant digits

DATA SCIENCE

Unit V

APPLICATIONS OF DATA SCIENCE

Some of the applications of data science include:

Fraud and Risk Detection - Over the years, banking companies learned to divide and conquer data via customer profiling, past expenditures and other essential variables to analyze the probabilities of risk and default. Moreover, it also helped them to push their banking products based on customer's purchasing power.

Healthcare – The health care sector includes several applications such as medical image analysis, genetics and genomics, drug development, virtual assistance for patients and customer support.

Internet Search - All search engines make use of data science algorithms to deliver the best result for our searched query in fraction of seconds.

Targeted Advertising - Starting from the display banners on various websites to the digital bill boards at the airports – almost all of them are decided by using data science algorithms. This is the reason why digital ads have been able to get a lot higher CTR than traditional advertisements. They can be targeted based on user's past behaviour.

Website Recommendations –

Price comparison websites - At a basic level, these websites are being driven by lots and lots of data which is fetched using APIs and RSS Feeds. If you have ever used these websites, you would know, the convenience of comparing the price of a product from multiple vendors at one place. PriceGrabber, PriceRunner, Junglee, Shopzilla, DealTime are some examples of price comparison websites. Now a days, price comparison website can be found in almost every domain such as technology, hospitality, automobiles, durables, apparels etc.

Recommender systems - They not only help you find relevant products from billions of products available with them, but also adds a lot to the user experience. A lot of companies have fervidly used this engine / system to promote their products / suggestions in accordance with user's

interest and relevance of information. Internet giants like Amazon, Twitter, Google Play, Netflix, Linkedin, imdb and many more uses this system to improve user experience. The recommendations are made based on previous search results for a user.

Image, Speech Recognition – Image and speech recognition have immense applications in data science. Speech Recognition - Some of the best example of speech recognition products are Google Voice, Siri, Cortana.

Airline Route Planning - Now using data science, the airline companies can: Predict flight delay, Decide which class of airplanes to buy, whether to directly land at the destination, or take a halt in between (For example: A flight can have a direct route from New Delhi to New York. Alternatively, it can also choose to halt in any country.), Effectively drive customer loyalty programs.

Gaming - Games are now designed using machine learning algorithms which improve / upgrade themselves as the player moves up to a higher level. In motion gaming also, your opponent (computer) analyzes your previous moves and accordingly shapes up its game.

Augmented Reality - Data Science and Virtual Reality do have a relationship, considering a VR headset contains computing knowledge, algorithms and data to provide you with the best viewing experience. A very small step towards this is the high trending game of Pokemon GO. The ability to walk around things and look at Pokemon on walls, streets, things that aren't really there. The creators of this game used the data from Ingress, the last app from the same company, to choose the locations of the Pokemon and gyms.

TECHNOLOGIES FOR VISUALIZATION

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. In the early days of visualization, the most common visualization technique was using a Microsoft Excel spreadsheet to transform the information into a table, bar graph or pie chart. While these visualization methods are still commonly used, more intricate techniques are now available, including, scatter plots (2D and 3D), Tree maps, Heat maps, time series charts, word cloud, Cartogram, Network etc.

Generally speaking, Excel, R language, ggplot2 and Python are the most popular tools used for data visualization. However there are numerous other software tools that are used for data visualization. Some of them are:

- **Tableau** is a business intelligence tool for visually analyzing data. Users can create and distribute interactive and shareable dashboards, depicting trends, changes and densities of data in graphs and charts. Tableau can connect to files, relational data sources and big data sources to get and process data.
- **FineReport** is an enterprise-level web reporting tool written in pure Java, combining data visualization and data entry. It is designed based on "no-code development" concept. With FineReport, users can make complex reports and cool dashboards and build a decision-making platform with simple drag-and-drop operations.
- **Power BI** is a set of business analysis tools that provide insights in the organization. It can connect hundreds of data sources, simplify data preparation and provide instant analysis. Organizations can view reports generated by Power BI on web and mobile devices.
- **D3.js** is a JavaScript library based on data manipulation documentation. D3 combines powerful visualization components with data-driven DOM manipulation methods.
- **HighCharts** is a chart library written in pure JavaScript that makes it easy and convenient for users to add interactive charts to web applications. This is the most widely used chart tool on the Web, and business use requires the purchase of a commercial license.
- **Echarts** is an enterprise-level chart tool from the data visualization team of Baidu. It is a pure Javascript chart library that runs smoothly on PCs and mobile devices, and it is compatible with most current browsers.
- **Leaflet** is a JavaScript library of interactive maps for mobile devices. It has all the mapping features that most developers need.
- **Vega** is a set of interactive graphical grammars that define the mapping rules from data to graphic, common interaction grammars, and common graphical elements. Users can freely combine Vega grammars to build a variety of charts.
- **eck.gl** is a visual class library based on WebGL for big data analytics. It is developed by the visualization team of Uber.

BOKEH (PYTHON)

Bokeh is a Python library for interactive visualization that targets web browsers for representation. Bokeh has multiple language bindings (Python, R, lua and Julia). These bindings produce a JSON file, which works as an input for BokehJS (a Javascript library), which in turn presents data to the modern web browsers. Some of the benefits of Bokeh are:

- Bokeh allows you to build complex statistical plots quickly and through simple commands.
- Bokeh provides you output in various medium like html, notebook and server.
- Bokeh visualization can be embedded to Flask and Django app.
- Bokeh can transform visualization written in other libraries like matplotlib, seaborn, ggplot.
- Bokeh has flexibility for applying interaction, layouts and different styling option to visualization.
- Bokeh can produce elegant and interactive visualization like D3.js with high-performance interactivity over very large or streaming datasets.
- Bokeh can help anyone who would like to quickly and easily create interactive plots, dashboards, and data applications.

Bokeh provides multiple visualization interfaces to the user. Bokeh interface has three components:

i. **Charts**: a high-level interface that is used to build complex statistical plots as quickly and in a simplistic manner. Charts are used to present information in standard visualization form. These forms include box plot, bar chart, area plot, heat map, donut chart and many others. These plots can be generated just by passing data frames, numpy arrays and dictionaries.

ii. **Plotting**: an intermediate-level interface that is centered around composing visual glyphs.

iii. **Models**: a low-level interface that provides the maximum flexibility to application developers.

Building a visualization with Bokeh involves the following steps:

i. Prepare the data
ii. Determine where the visualization will be rendered
iii. Set up the figure(s)
iv. Connect to and draw your data
v. Organize the layout
vi. Preview and save your beautiful data creation
vii. Prepare the data

Prepare the data: This step commonly involves data handling libraries like Pandas and Numpy and is all about taking the required steps to transform it into a form that is best suited for the intended visualization.

Determine where the Visualization Will Be Rendered: Determine how you want to generate and ultimately view the visualization (such as generating a static HTML file and rendering your visualization inline in a Jupyter Notebook).

Set up the Figure(s): Assemble the figure, preparing the canvas for visualization. In this step, everything from the titles to the tick marks can be customized. A suite of tools that can enable various user interactions with the visualization can also be set up.

Connect to and Draw Data: Use Bokeh's multitude of renderers to give shape to the data. There is enough flexibility to draw the data from scratch using the many available marker and shape options, all of which are easily customizable. Additionally, Bokeh has some built-in functionality for building things like stacked bar charts and plenty of examples for creating more advanced visualizations like network graphs and maps

Organize the Layout: Not only does Bokeh offer the standard grid-like layout options, but it also allows to easily organize visualizations into a tabbed layout in just a few lines of code. In addition, the plots can be quickly linked together, so a selection on one will be reflected on any combination of the others.

Preview and Save the Visualization: Visualization can be saved to an image file or viewed in a browser or notebook.

The following examples illustrate the use of Bokeh:

Example 1: Create a bar chart and visualize it on web browser using Bokeh.

Methodology: The common methodology followed to create a chart:

i. Import the library and functions/ methods
ii. Prepare the data
iii. Set the output mode (Notebook, Web Browser or Server)
iv. Create chart with styling option (if required)
v. Visualize the chart

```
#Import library
from bokeh.charts import Bar, output_file, show #use output_notebook to visualize it in notebook
# prepare data (dummy data)
data = {"y": [1, 2, 3, 4, 5]}
# Output to Line.HTML
output_file("lines.html", title="line plot example") #put output_notebook() for notebook
# create a new line chat with a title and axis labels
p = Bar(data, title="Line Chart Example", xlabel='x', ylabel='values', width=400, height=400)
# show the results
show(p)
```

Example-2: Create a line plot to Bokeh server

Methodology: To start plotting on Bokeh server, the command bokeh-server must be executed to initialize it followed by the commands used for visualization.

```
from bokeh.plotting import figure, output_server, show
```

```
output_server("line")

p = figure(plot_width=400, plot_height=400)

# add a line renderer

p.line([5, 2, 3, 4, 5], [5, 7, 2, 4, 5], line_width=2)

show(p)
```

Example-3: Create a scatter square mark on XY frame of notebook

Methodology: Bokeh plots created using the bokeh.plotting interface comes with a default set of tools and visual styles. For plotting, follow the below steps:

i. Import library, methods or functions
ii. Select the output mode (notebook, web browser, server)
iii. Activate a figure (similar like matplotlib)
iv. Perform subsequent plotting operations, it will affect the generated figure.
v. Visualize it

The major concept of Bokeh is that graphs are built up one layer at a time. Create a figure, and then add elements, called glyphs, to the figure by calling the appropriate method and passing in data. Glyphs can take on many shapes depending on the desired use: circles, lines, patches, bars, arcs, and so on. A few tools come with any Bokeh plot which are on the right side and include panning, zooming, selection, and plot saving abilities. These tools are configurable and will come in handy when we want to investigate our data.

```
from bokeh.plotting import figure, output_notebook, show
# output to notebook
output_notebook()
p = figure(plot_width=400, plot_height=400)
# add square with a size, color, and alpha
p.square([2, 5, 6, 4], [2, 3, 2, 1, 2], size=20, color="navy")
# show the results
```

```
show(p)
```

Example-4: Add a hover tool and axis labels to above plot.

Passive interactions are actions the viewer can take which do not alter the data displayed. These are referred to as inspectors because they allow viewers to “investigate” the data in more detail. A useful inspector is the tooltip which appears when a user mouses over data points and is called the HoverTool in Bokeh.

The HoverToolinstance is passed a list of tooltips as Python tuples where the first element is the label for the data and the second references the specific data to highlight. We can reference either attributes of the graph, such as x or y position using ‘$’ or specific fields in our source using ‘@’. An example of a HoverTool usage to do both:

```
# Hover tool referring to our own data field using @ and a position on the graph using $

h = HoverTool(tooltips = [('Delay Interval Left ', '@left'), ('(x,y)', '($x, $y)')])
```

The left data field is referenced in the ColumnDataSource (which corresponds to the ‘left’ column of the original dataframe) using ‘@’ and the (x,y) position of the cursor is referenced using ‘$’.

```
from bokeh.plotting import figure, output_notebook, show
from bokeh.models import HoverTool, BoxSelectTool #For enabling tools
# output to notebook
output_notebook()
#Add tools
TOOLS = [BoxSelectTool(), HoverTool()]
p = figure(plot_width=400, plot_height=400, tools=TOOLS)
# add a square with a size, color, and alpha
p.square([2, 5, 6, 4], [2, 3, 2, 1, 2], size=20, color="navy", alpha=0.5)
#Visual Elements
p.xaxis.axis_label = "X-axis"
p.yaxis.axis_label = "Y-axis"
```

```
# show the results
show(p)
```

TRENDS IN VARIOUS DATA COLELCTION AND ANALYSIS TECHNIQUES

Data collection is defined as the procedure of collecting, measuring and analyzing accurate insights for research using standard validated techniques. The approach of data collection is different for different fields of study, depending on the required information. The most critical objective of data collection is ensuring that information-rich and reliable data is collected for statistical analysis so that data-driven decisions can be made for research. Essentially there are four choices for data collection – in-person interviews, mail, phone and online.

Analyzing information involves examining it in ways that reveal the relationships, patterns, trends, etc. that can be found within it. Quantitative data refer to the information that is collected as, or can be translated into, numbers, which can then be displayed and analyzed mathematically. Qualitative data are collected as descriptions, anecdotes, opinions, quotes, interpretations, etc., and are generally either not able to be reduced to numbers, or are considered more valuable or informative if left as narratives. Quantitative and qualitative information needs to be analyzed differently.

Quantitative data are typically collected directly as numbers. Some examples include:

- The frequency (rate, duration) of specific behaviors or conditions
- Test scores (e.g., scores/levels of knowledge, skill, etc.)
- Survey results (e.g., reported behavior, or outcomes to environmental conditions; ratings of satisfaction, stress, etc.)
- Numbers or percentages of people with certain characteristics in a population (diagnosed with diabetes, unemployed, Spanish-speaking, under age 14, grade of school completed, etc.).

Data can also be collected in forms other than numbers, and turned into quantitative data for analysis. Quantitative data is usually subjected to statistical procedures such as calculating the

mean or average number of times an event or behavior occurs (per day, month, year). Qualitative data can sometimes be changed into numbers, usually by counting the number of times specific things occur in the course of observations or interviews, or by assigning numbers or ratings to dimensions (e.g., importance, satisfaction, ease of use). It may reveal why certain methods are working or not working, whether part of what you're doing conflicts with participants' culture, what participants see as important, etc. It may also show patterns – in behavior, physical or social environment, or other factors – that the numbers in your quantitative data don't, and occasionally even identify variables that researchers weren't aware of.

METHODOLOGY TO COLLECT AND ANALYZE DATA:

- Implement the measurement system
- Organize the data collected
- Conduct data graphing, visual inspection, statistical analysis, or other operations on the data as appropriate
- Take note of any significant or interesting results
- Interpret the results

Some of the most common types of data collection used today are: Interviews, Surveys and Questionnaires, Focus Group Discussions, Case Studies, Online Tracking, Transactional data tracking, Online marketing analytics, Social media monitoring, Collecting subscription and registration data, In-store traffic monitoring.

VARIOUS VISULAIZATION TECHNIQUES

There are many different techniques to visualize data, such as Bar Chart, Line Chart which are already discussed in previous Units. The other techniques are explained below:

2 DIMENSIONAL AREA TECHNIQUES

- **Area or distance cartograms** are the copies of some parts of maps, depicting some additional parameters like demography, population size, traveltimes and any other variables.

- **Choropleth** is a map colored with different colors depending on the level of the examined variable, like the sales level per state or the biggest inventory stocks per state.
- **Dot distribution map** is the data visualization method relying on using the dots to highlight the level of presence of the examined variable within the area.

MULTI DIEMNSIONAL DATA VISUALIZATIONS

- The **pie chart** is among the most popular tools for data representation. It is split into sectors illustrating some numerical values, with the angle and the arc length in each sector being proportional to the value represented.
- The **histogram** is a series of rectangles, representing both the time periods (width) and the parameter values (height), which helps clearly grasp the dynamics of the parameter adjustments.
- The **scatter plot** is the model of data visualization depicting 2 sets of unconnected dots as parameter values.

HEIRARCHICAL DATA VISUALIZATION

- A **dendrogram** is an illustration of a hierarchical clustering of various data sets, helping to understand their relations in an instant.
- A **sunburst chart** (or a ring chart) is a pie chart with concentric circles, describing the hierarchy of data values.
- The **tree diagram** allows to describe the tree-like relations within the data structure, usually from the upside down or from the left to the right.

NETWORK MODELS

- An **alluvial diagram** is the example of a flow diagram that represents changes in the data structure over time or under certain conditions.
- A **node-link diagram** is usually a circular image with dots representing the data nodes and lines representing the links between said nodes. This helps visualize the relations between the data sources and understand what results are based on what data.
- **Matrix diagram** or chart is used when we have multiple data sets connected to each other via some relations. Matrix helps show both the data set positions against each other and the relations between these sets.

TEMPORAL VISUALIZATION

- **Connected Scatter Plot** is the plot of values for two variables taken from a data set. These values are scattered throughout the picture and connected with a line.
- **Polar area diagram** might look like a standard pie chart, yet the size of the sector is evaluated by the distance from the center in addition to the arc length and angle. Thus said, a sharp sector stretched far away from the center might be more important than a blunt sector that does not reach far.
- **Time series** is the most often used example of continuous data evaluations over a period of time. The graph of CPU usage, the number of website visits over a month and a plethora of other historical data are best described using this data visualization technique.

APPLICATION DEVELOPMENT METHODS IN DATA SCIENCE

Applications of data science include Internet search, recommendation systems, Image/Speech recognition, Gaming, customer services, network operation in telecommunications industry, Enterprise management, precision marketing, health care, public security and surveillance, environmental protection, smart cities and Stock market analysis. The data science application development pipeline has the following elements: Obtain the data, wrangle the data, explore the data, model the data and generate the report. Each element requires lot of skills and expertise in several domains such as statistics, machine learning, and programming.

M.E.(CSE) II SEMESTER

19CSCSOE25. DATA SCIENCE

QUESTION BANK

1. What is the Central Limit Theorem and why is it important?

Given a dataset with unknown distribution (it could be uniform, binomial or completely random), the sample means will approximate the normal distribution.

> Example: Suppose that we are interested in estimating the average height among all people. Collecting data for every person in the world is impossible. While we can't obtain a height measurement from everyone in the population, we can still sample some people. The question now becomes, what can we say about the average height of the entire population given a single sample. The Central Limit Theorem addresses this question exactly.

2. What is data sampling?

Data sampling is a statistical analysis technique used to select, manipulate and analyze a representative subset of data points to identify patterns and trends in the larger data set being examined.

3. What is linear regression?

Linear regression is a statistical technique where the score of a variable Y is predicted from the score of a second variable X. X is referred to as the predictor variable and Y as the criterion variable. A linear regression is a good tool for quick predictive analysis: for example, the price of a house depends on a myriad of factors, such as its size or its location. In order to see the relationship between these variables, we need to build a linear regression, which predicts the line of best fit between them and can help conclude whether or not these two factors have a positive or negative relationship.

The linear regression equation is a one-degree equation with the most basic form being Y = mX + C where m is the slope of the line and C is the standard error. It is used when the response variable is continuous in nature for example height, weight, and the number of hours. It can be a simple linear regression if it involves continuous dependent variable with one independent variable and a multiple linear regression if it has multiple independent variables.

Linear regression is a standard statistical practice to calculate the best fit line passing through the data points when plotted. The best fit line is chosen in such a way so that the distance of each data point is minimum from the line which reduces the overall error of the system. Linear regression assumes that the various features in the data are linearly related to the target. It is often used in predictive analytics for calculating estimates in the foreseeable future.

4. What are the assumptions required for linear regression?

There are four major assumptions: 1. There is a linear relationship between the dependent variables and the regressors, meaning the model you are creating actually fits the data, 2. The errors or residuals of the data are normally distributed and independent from each other, 3. There is minimal multicollinearity between explanatory variables, and 4. Homoscedasticity. This means the variance around the regression line is the same for all values of the predictor variable.

5. What is a statistical interaction?

Basically, an interaction is when the effect of one factor (input variable) on the dependent variable (output variable) differs among levels of another factor.

6. Why data cleaning plays a vital role in analysis?

Cleaning data from multiple sources to transform it into a format that data analysts or data scientists can work with is a cumbersome process because - as the number of data sources increases, the time take to clean the data increases exponentially due to the number of sources and the volume of data generated in these sources. It might take up to 80% of the time for just cleaning data making it a critical part of analysis task.

7. Differentiate between univariate, bivariate and multivariate analysis.

These are descriptive statistical analysis techniques which can be differentiated based on the number of variables involved at a given point of time. For example, the pie charts of sales based on territory involve only one variable and can be referred to as univariate analysis. If the analysis attempts to understand the difference between 2 variables at time as in a scatterplot, then it is referred to as bivariate analysis. For example, analysing the volume of sale and a spending can be considered as an example of bivariate analysis. Analysis that deals with the study of more than two variables to understand the effect of variables on the responses is referred to as multivariate analysis.

8. What do you understand by the term Normal Distribution?

Data is usually distributed in different ways with a bias to the left or to the right or it can all be jumbled up. However, there are chances that data is distributed around a central value without any bias to the left or right and reaches normal distribution in the form of a bell shaped curve. The random variables are distributed in the form of an symmetrical bell shaped curve.

9. What is the difference between Supervised Learning an Unsupervised Learning?

If an algorithm learns something from the training data so that the knowledge can be applied to the test data, then it is referred to as Supervised Learning. Classification is an example for

Supervised Learning. If the algorithm does not learn anything beforehand because there is no response variable or any training data, then it is referred to as unsupervised learning. Clustering is an example for unsupervised learning.

10. How can outlier values be treated?

Outlier values can be identified by using univariate or any other graphical analysis method. If the number of outlier values is few then they can be assessed individually but for large number of outliers the values can be substituted with either the 99th or the 1st percentile values. All extreme values are not outlier values. The most common ways to treat outlier values –

- To change the value and bring in within a range
- To just remove the value.

11. What are various steps involved in an analytics project?

- Understand the business problem
- Explore the data and become familiar with it.
- Prepare the data for modelling by detecting outliers, treating missing values, transforming variables, etc.
- After data preparation, start running the model, analyse the result and tweak the approach. This is an iterative step till the best possible outcome is achieved.
- Validate the model using a new data set.
- Start implementing the model and track the result to analyse the performance of the model over the period of time.

12. During data analysis, how do you treat missing values?

The extent of the missing values is identified after identifying the variables with missing values. If any patterns are identified the analyst has to concentrate on them as it could lead to interesting and meaningful business insights. If there are no patterns identified, then the missing values can be substituted with mean or median values (imputation) or they can simply be ignored. There are various factors to be considered when answering this question-

- Understand the problem statement, understand the data and then give the answer. Assigning a default value which can be mean, minimum or maximum value.
- If it is a categorical variable, the default value is assigned. The missing value is assigned a default value.
- If you have a distribution of data coming, for normal distribution give the mean value.
- Should we even treat missing values is another important point to consider? If 80% of the values for a variable are missing then you can answer that you would be dropping the variable instead of treating the missing values.

13. How is Data Science different from Big Data and Data Analytics?

Data Science utilizes algorithms and tools to draw meaningful and commercially useful insights from raw data. It involves tasks like data modelling, data cleansing, analysis, pre-processing etc.

14. What is the use of Statistics in Data Science?

Statistics provides tools and methods to identify patterns and structures in data to provide a deeper insight into it. Serves a great role in data acquisition, exploration, analysis, and validation. It plays a really powerful role in Data Science. Data Science is a derived field which is formed from the overlap of statistics probability and computer science. Whenever one needs to do estimations, statistics is involved. Many algorithms in data science are built on top of statistical formulae and processes. Hence statistics is an important part of data science.

15. What is the importance of Data Cleansing?

As the name suggests, data cleansing is a process of removing or updating the information that is incorrect, incomplete, duplicated, irrelevant, or formatted improperly. It is very important to improve the quality of data and hence the accuracy and productivity of the processes and organization as a whole. Real-world data is often captured in formats which have hygiene issues. There are sometimes errors due to various reasons which make the data inconsistent and sometimes only some features of the data. Hence data cleansing is done to filter the usable data from the raw data, otherwise many systems consuming the data will produce erroneous results.

16. Explain Normal Distribution.

Normal Distribution is also called the Gaussian Distribution. It is a type of probability distribution such that most of the values lie near the mean. It has the following characteristics:

- The mean, median, and mode of the distribution coincide
- The distribution has a bell-shaped curve
- The total area under the curve is 1
- Exactly half of the values are to the right of the centre, and the other half to the left of the centre.

17. What is correlation and covariance in statistics?

Correlation is defined as the measure of the relationship between two variables. If two variables are directly proportional to each other, then its positive correlation. If the variables are indirectly proportional to each other, it is known as a negative correlation. Covariance is the measure of how much two random variables vary together.

18. What is ‘Naive’ in a Naive Bayes?

A naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature, given the class variable. The Naive Bayes Algorithm is based on the Bayes Theorem. Bayes’ theorem describes the probability of an

event, based on prior knowledge of conditions that might be related to the event. Basically, it's "naive" because it makes assumptions that may or may not turn out to be correct.

19. Explain the SVM machine learning algorithm.

SVM stands for support vector machine, it is a supervised machine learning algorithm which can be used for both Regression and Classification. If you have n features in your training data set, SVM tries to plot it in n-dimensional space with the value of each feature being the value of a particular coordinate. SVM uses hyper planes to separate out different classes based on the provided kernel function. SVM is an ML algorithm which is used for classification and regression. For classification, it finds out a multi dimensional hyperplane to distinguish between classes. SVM uses kernels which are namely linear, polynomial, and RBF. There are few parameters which need to be passed to SVM in order to specify the points to consider while the calculation of the hyperplane.

20. What are the ways (methods) to collect data?

Surveys, Online tracking, transactional data tracking, online marketing analytics, Social media monitoring, Collecting subscription and registration data, In-store traffic monitoring.

21. What are the different kernels functions in SVM ?

There are four types of kernels in SVM. They are: Linear Kernel, Polynomial kernel, Radial basis kernel, Sigmoid kernel.

22. What is the difference between Regression and classification ML techniques?

Both Regression and classification machine learning techniques come under Supervised machine learning algorithms. In Supervised machine learning algorithm, we have to train the model using labelled data set, While training we have to explicitly provide the correct labels and algorithm tries to learn the pattern from input to output. If our labels are discrete values then it will a classification problem, e.g A,B etc. but if our labels are continuous values then it will be a regression problem, e.g 1.23, 1.333 etc.

23. What is data visualization?

Data visualization is a process of representing data in a graphical format by using different visual elements such as charts, tables, graphs, maps, infographics, etc. There are various data visualization tools available in the market to represent the overview of the data in a user/customer understandable format. Visualization tools depict the trends, outliers, and patterns in data.

24. How is standard deviation affected by the outliers?

The variation in the input value of x, that is, a variation in its value between high and low would adversely affect the standard deviation and its value would be farther away from the mean. Therefore, we conclude that outliers will have an effect on the standard deviation.

25. What is skewed Distribution & uniform distribution?

Skewed distribution occurs when if data is distributed on any one side of the plot whereas uniform distribution is identified when the data is spread is equal in the range.

26. Treating a categorical variable as a continuous variable would result in a better predictive model?

Yes, the categorical value should be considered as a continuous variable only when the variable is ordinal in nature. So it is a better predictive model.

27. Is it possible to capture the correlation between continuous and categorical variable?

Yes, we can use analysis of covariance technique to capture the association between continuous and categorical variables.

28. What is Prior probability and likelihood?

Prior probability is the proportion of the dependent variable in the data set while the likelihood is the probability of classifying a given observant in the presence of some other variable.

29. Five numbers are given: (5, 10, 15, 5, 15). Now, what would be the sum of deviations of individual data points from their mean?

    The sum of deviations of the individual will always be 0.

ADDITIONAL QUESTIONS: (require detailed answers)

1. Explain the data science process.
2. Explain the stages in the life cycle of a data science project.
3. Describe the various types of data.
4. Enumerate the elements of data science toolkit.
5. What are the sources of data? Explain.
6. Narrate the APIs useful in data collection.
7. Discuss the methods for fixing the issues in data collection and analysis.
8. Explain the mechanisms for data storage and management.
9. Describe the role of multiple data sources in data science projects.
10. Explain the central limit theorem and the properties of distributions.
11. Define mean, variance, standard deviation, covariance, correlation.
12. Explain the following machine learning algorithms: Linear Regression, Naïve Bayes, SVM.
13. Explain the types of data visualization

14. Discuss the types of data with examples.
15. What are data encodings?
16. Explain visual encodings.
17. Describe the approach for mapping variables to encodings.
18. Describe the technologies for data visualization.
19. Explain the features of Bokeh useful for data visualization.
20. Explain the various visualization techniques and discuss their applications.
21. Describe the techniques for data collection and analysis.
22. Discuss the steps in the development of data science applications and explain the visualization techniques suitable for the applications.