UNIT – 1

TEN MARK QUESTIONS

1. DISCUSS IN DETAIL ABOUT AESTHETICS AND TYPES OF DATA.

All data visualizations map data values into quantifiable features of the resulting graphic. We refer to these features as *aesthetics*.

Aesthetics and Types of Data

Aesthetics describe every aspect of a given graphical element. A few examples are provided in Figure 2-1.

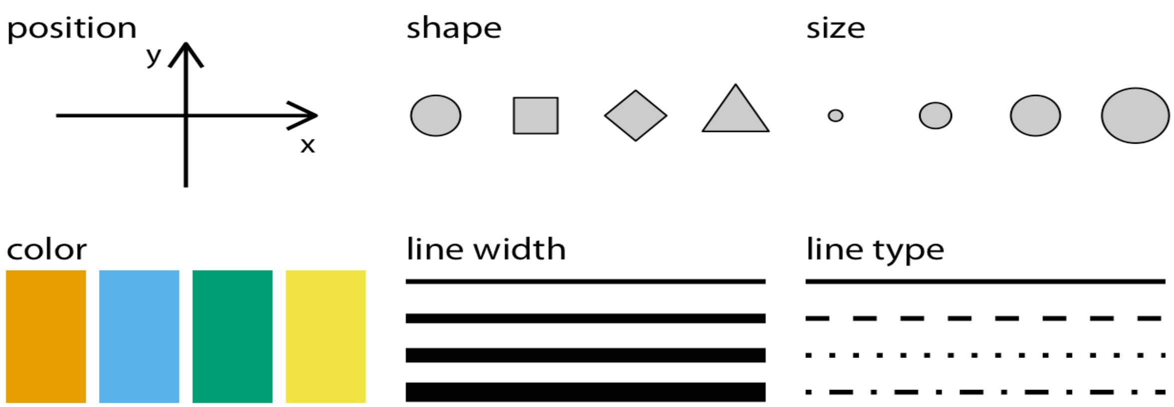


Figure 2-1. Commonly used aesthetics in data visualization: position, shape, size, color, line width, line type. Some of these aesthetics can represent both continuous and discrete data (position, size, line width, color), while others can usually only represent discrete data (shape, line type).

A critical component of every graphical element is of course its *position*, which describes where the element is located. In standard 2D graphics, we describe positions by an *x* and *y* value, but other coordinate systems and one- or three-dimensional visualizations are possible.

Next, all graphical elements have a *shape*, a *size*, and a *color*. While preparing a black-and-white drawing, graphical elements need to have a color to be visible: for example, black if the background is white or white if the background is black. Finally, to the extent we are using lines to visualize data, these lines may have different widths or dash–dot patterns.

Beyond the examples shown in Figure 2-1, there are many other aesthetics we may encounter in a data visualization. For example, if we want to display text, we may have to specify font family, font face, and font size, and if graphical objects overlap, we may have to specify whether they are partially transparent.

All aesthetics fall into one of two groups:

those that can represent continuous data

and those that cannot.

Continuous data values are values for which arbitrarily fine intermediates exist. For example, time duration is a continuous value. Between any two durations, say 50 seconds and 51 seconds, there are arbitrarily many intermediates, such as 50.5 seconds, 50.51 seconds, 50.50001 seconds, and so on.

By contrast, number of persons in a room is a discrete value. A room can hold 5 persons or 6, but not 5.5. For the examples in Figure 2-1, position, size, color, and line width can represent continuous data, but shape and line type can usually only represent discrete data.

Types of Data

In addition to continuous and discrete numerical values, data can come in the form of discrete categories, in the form of dates or times, and as text (Table 2-1).

When data is numerical we also call it *quantitative* and when it is categorical we call it *qualitative*. Variables holding qualitative data are *factors*, and the different categories are called *levels*. The levels of a factor are most commonly without order (as in the example of *dog, cat, fish* in Table 2-1), but factors can also be ordered, when there is an intrinsic order among the levels of the factor (as in the example of *good, fair, poor* in Table 2-1).

Table 2-1. Types of variables encountered in typical data visualization scenarios.

Type of variable	Examples	Appropriate scale	Description
Quantitative/ numerical continuous	1.3, 5.7, 83, 1.5×10^{-2}	Continuous	Arbitrary numerical values. These can be integers, rational numbers, or real numbers.
Quantitative/ numerical discrete	1, 2, 3, 4	Discrete	Numbers in discrete units. These are most commonly but not necessarily integers. For example, the numbers 0.5, 1.0, 1.5 could also be treated as discrete if intermediate values cannot exist in the given dataset.
Qualitative/ categorical unordered	dog, cat, fish	Discrete	Categories without order. These are discrete and unique categories that have no inherent order. These variables are also called *factors*.
Qualitative/ categorical ordered	good, fair, poor	Discrete	Categories with order. These are discrete and unique categories with an order. For example, "fair" always lies between "good" and "poor." These variables are also called *ordered factors*.
Date or time	Jan. 5 2018, 8:03am	Continuous or discrete	Specific days and/or times. Also generic dates, such as July 4 or Dec. 25 (without year).
Text	The quick brown fox jumps over the lazy dog.	None, or discrete	Free-form text. Can be treated as categorical if needed.

To examine a concrete example of these various types of data, take a look at Table 2-2.

Table 2-2. First 8 rows of a dataset listing daily temperature normals for four weather stations. Data source: National Oceanic and Atmospheric Administration (NOAA).

Month	Day	Location	Station ID	Temperature (°F)
Jan	1	Chicago	USW00014819	25.6
Jan	1	San Diego	USW00093107	55.2
Jan	1	Houston	USW00012918	53.9
Jan	1	Death Valley	USC00042319	51.0
Jan	2	Chicago	USW00014819	25.5
Jan	2	San Diego	USW00093107	55.3
Jan	2	Houston	USW00012918	53.8
Jan	2	Death Valley	USC00042319	51.2

It shows the first few rows of a dataset providing the daily temperature normals (average daily temperatures over a 30-year window) for four US locations. This table contains five variables: month, day, location, station ID, and temperature (in degrees Fahrenheit).

Month is an ordered factor,
day is a discrete numerical value,
location is an unordered factor,
station ID is similarly an unordered factor, and
temperature is a continuous numerical value.

2. HOW DO SCALES MAP DATA VALUES ON TO AESTHETICS? ILLUSTRATE WITH AN EXAMPLE DATASET.

To map data values onto aesthetics, we need to specify which data values correspond to which specific aesthetics values.

For **example**, if our graphic has an *x* axis, then we need to specify which data values fall onto particular positions along this axis. Similarly, we may need to specify which data values are represented by particular shapes or colors. This mapping between data values and aesthetics values is created via *scales*.

A scale defines a unique mapping between data and aesthetics (Figure 2-2). Importantly, a scale must be one-to-one, such that for each specific data value there is exactly one aesthetics value and vice versa. If a scale isn't one-to-one, then the data visualization becomes ambiguous.

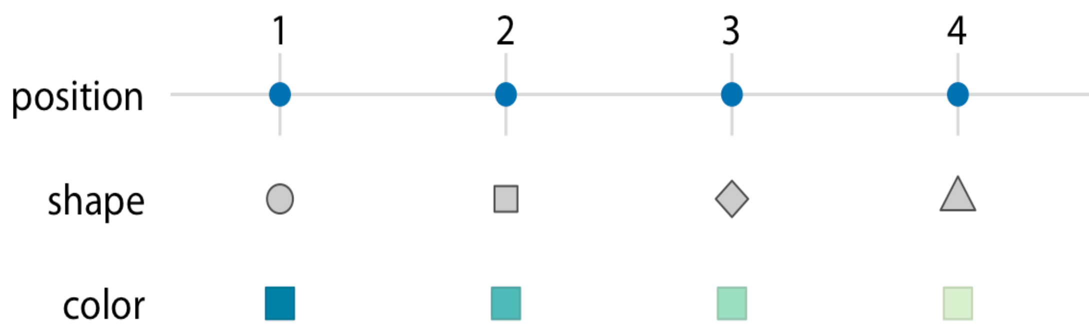


Figure 2-2. Scales link data values to aesthetics. Here, the numbers 1 through 4 have been mapped onto a position scale, a shape scale, and a color scale. For each scale, each number corresponds to a unique position, shape, or color, and vice versa.

<u>Example: Scales illustrated using Temperature dataset</u>

We can take the dataset shown in Table 2-2, map temperature onto the *y* axis, day of the year onto the *x* axis, and location onto color, and visualize these aesthetics with solid lines. The result is a standard line plot showing the temperature normals at the four locations as they change during the year (Figure 2-3).

Month	Day	Location	Station ID	Temperature (°F)
Jan	1	Chicago	USW00014819	25.6
Jan	1	San Diego	USW00093107	55.2
Jan	1	Houston	USW00012918	53.9
Jan	1	Death Valley	USC00042319	51.0
Jan	2	Chicago	USW00014819	25.5
Jan	2	San Diego	USW00093107	55.3
Jan	2	Houston	USW00012918	53.8
Jan	2	Death Valley	USC00042319	51.2

Table 2-2. First 8 rows of a dataset listing daily temperature normals for four weather stations. Data source: National Oceanic and Atmospheric Administration (NOAA).

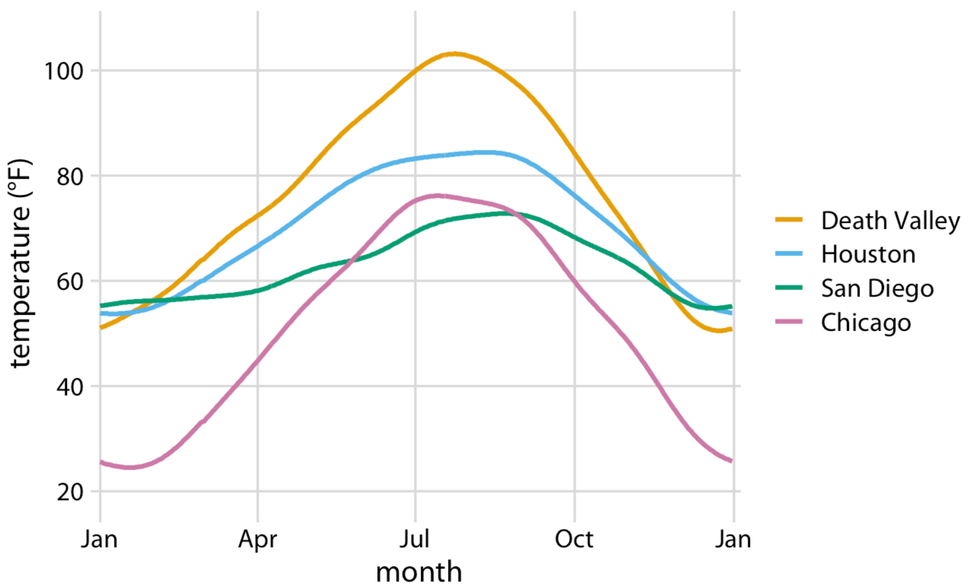


Figure 2-3. Daily temperature normals for four selected locations in the US. Temperature is mapped to the y *axis, day of the year to the* x *axis, and location to line color. Data source: NOAA.*

Figure 2-3 is a fairly standard visualization for a temperature curve and likely the visualization most data scientists would intuitively choose first. However, it is up to us which variables we map onto which scales. For example, instead of mapping temperature onto the *y* axis and location onto color, we can do the opposite. Because now the key variable of interest (temperature) is shown as color, we need to show sufficiently large colored areas for the colors to convey useful information [Stone, Albers Szafir, and Setlur 2014]. Therefore, for this visualization I have chosen squares instead of lines, one for each month and location, and I have colored them by the average temperature normal for each month (Figure 2-4).

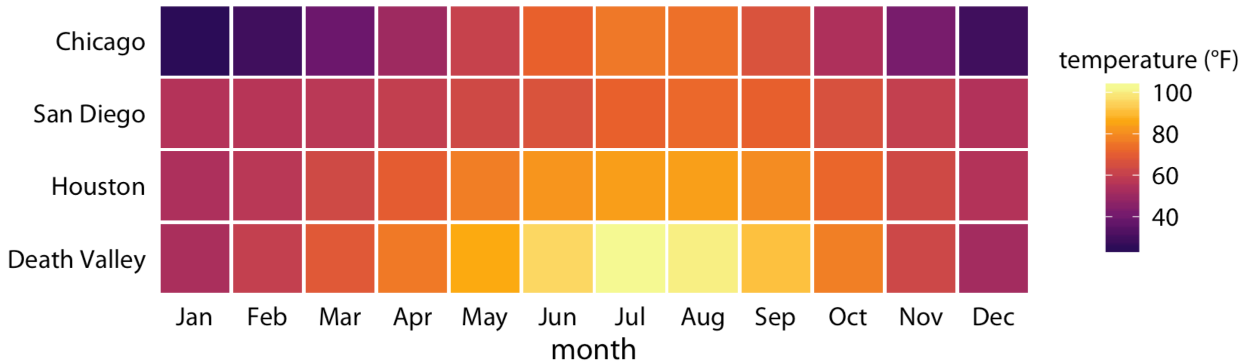


Figure 2-4. Monthly normal mean temperatures for four locations in the US. Data source: NOAA.

Figure 2-4 uses two position scales
(month along the *x* axis and
location along the *y* axis),
but neither is a continuous scale. Month is an ordered factor with 12 levels and location is an unordered factor with 4 levels. Therefore, the two position scales are both discrete.

For discrete position scales, we generally place the different levels of the factor at an equal spacing along the axis. If the factor is ordered (as is here the case for month), then the levels need to be placed in the appropriate order. If the factor is unordered (as is here the case for location), then the order is arbitrary, and we can choose any order we want. Here, the locations are ordered from overall coldest (Chicago) to overall hottest (Death Valley) to generate a pleasant staggering of colors.

3. WRITE IN DETAIL ABOUT CARTESIAN COORDINATES.

The combination of a set of position scales and their relative geometric rrangement is called a *coordinate system.*

The most widely used coordinate system for data visualization is the 2D *Cartesian coordinate system*, where each location is uniquely specified by an *x* and a *y* value. The *x* and *y* axes run orthogonally to each other, and data values are placed in an even spacing along both axes (Figure 3-1). The two axes are continuous position scales, and they can represent both positive and negative real numbers. To fully specify the coordinate system, we need to specify the range of numbers each axis covers. In Figure 3-1, the *x* axis runs from –2.2 to 3.2 and the *y* axis runs from –2.2 to 2.2. Any data values between these axis limits are placed at the appropriate respective location in the plot. Any data values outside the axis limits are discarded.

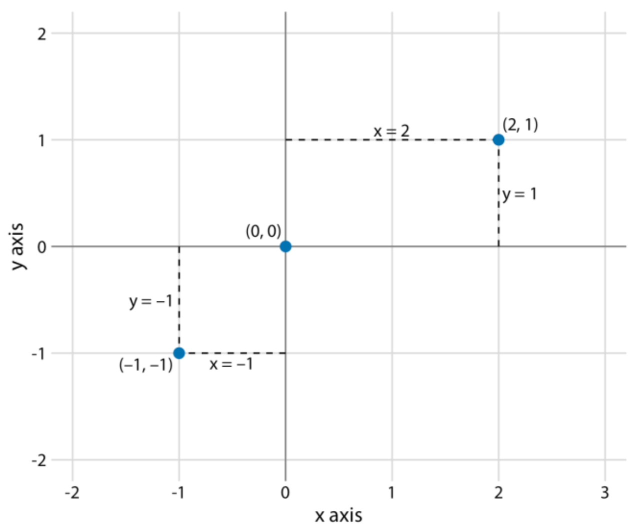


Figure 3-1. Standard Cartesian coordinate system. The horizontal axis is conventionally called x *and the vertical axis* y*. The two axes form a grid with equidistant spacing. Here, both the* x *and the* y *grid lines are separated by units of one. The point (2, 1) is located two* x *units to the right and one* y *unit above the origin (0, 0). The point (–1, –1) is located one* x *unit to the left and one* y *unit below the origin.*

Data values usually aren't just numbers, however. They come with units.

For example,

If we're measuring temperature, the values may be measured in degrees Celsius or Fahrenheit.
If we're measuring distance, the values may be measured in kilometers or miles, and
If we're measuring duration, the values may be measured in minutes, hours, or days.

In a Cartesian coordinate system, the spacing between grid lines along an axis corresponds to discrete steps in these data units. In a temperature scale, for example, we may have a grid line every 10 degrees Fahrenheit, and in a distance scale, we may have a grid line every 5 kilometers.

A Cartesian coordinate system can have two axes representing two different units. This situation arises quite commonly whenever we're mapping two different types of variables to *x* and *y*.
For example, in Figure 2-3, we plotted temperature versus days of the year. The *y* axis of Figure 2-3 is measured in degrees Fahrenheit, with a grid line every at 20 degrees, and the *x* axis is measured in months, with a grid line at the first of every third month. Whenever the two axes are measured in different units, we can stretch or compress one relative to the other and maintain a valid visualization of the data (Figure 3-2). Which version is preferable may depend on the story we want to convey.

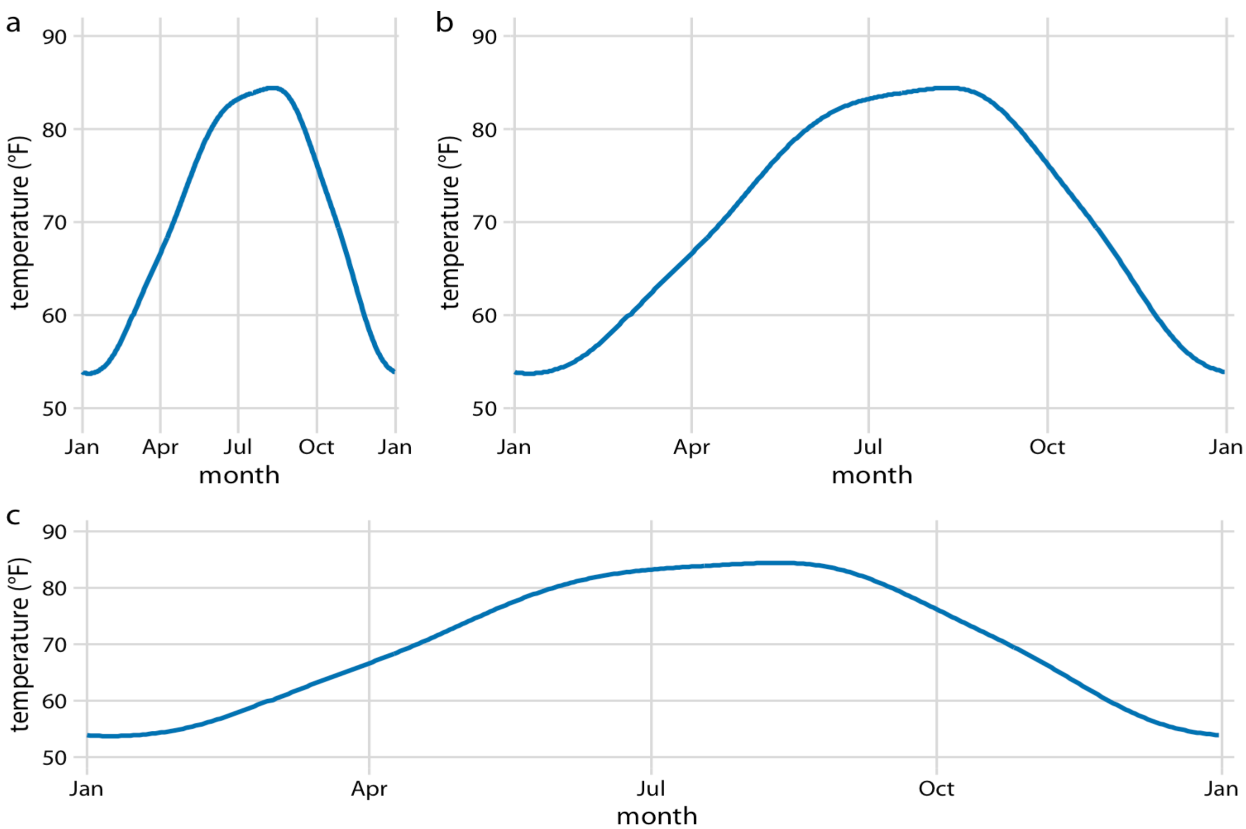


Figure 3-2. Daily temperature normals for Houston, TX. Temperature is mapped to the y *axis and day of the year to the* x *axis. Parts (a), (b), and (c) show the same*

figure in different aspect ratios. All three parts are valid visualizations of the temperature data. Data source: NOAA.

A tall and narrow figure emphasizes change along the *y* axis and a short and wide figure does the opposite. Ideally, we want to choose an aspect ratio that ensures that any important differences in position are noticeable.

On the other hand, if the *x* and *y* axes are measured in the same units, then the grid spacings for the two axes should be equal, such that the same distance along the *x* or *y* axis corresponds to the same number of data units.

4. COMPARE AND CONTRAST CARTESIAN AND POLAR COORDINATES.

Cartesian Coordinates

The most widely used coordinate system for data visualization is the 2D *Cartesian coordinate system*, where each location is uniquely specified by an *x* and a *y* value. The *x* and *y* axes run orthogonally to each other, and data values are placed in an even spacing along both axes (Figure 3-1). The two axes are continuous position scales, and they can represent both positive and negative real numbers. To fully specify the coordinate system, we need to specify the range of numbers each axis covers. In Figure 3-1, the *x* axis runs from –2.2 to 3.2 and the *y* axis runs from –2.2 to 2.2. Any data values between these axis limits are placed at the appropriate respective location in the plot. Any data values outside the axis limits are discarded.

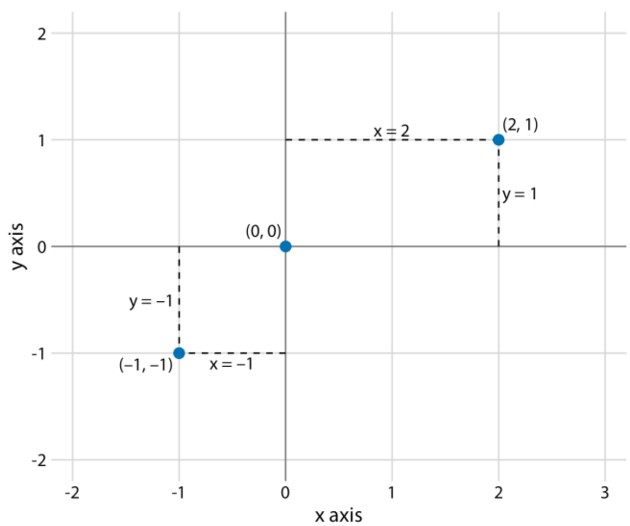


Figure 3-1. Standard Cartesian coordinate system. The horizontal axis is conventionally called x *and the vertical axis* y*. The two axes form a grid with equidistant spacing. Here, both the* x *and the* y *grid lines are separated by units of one. The point (2, 1) is located two* x *units to the right and one* y *unit above the origin (0, 0). The point (–1, –1) is located one* x *unit to the left and one* y *unit below the origin.*

Data values usually aren't just numbers, however. They come with units.

For example,

If we're measuring temperature, the values may be measured in degrees Celsius or Fahrenheit.

If we're measuring distance, the values may be measured in kilometers or miles, and If we're measuring duration, the values may be measured in minutes, hours, or days.

In a Cartesian coordinate system, the spacing between grid lines along an axis corresponds to discrete steps in these data units. In a temperature scale, for example, we may have a grid line every 10 degrees Fahrenheit, and in a distance scale, we may have a grid line every 5 kilometers.

A Cartesian coordinate system can have two axes representing two different units. This situation arises quite commonly whenever we're mapping two different types of variables to *x* and *y*.
For example, in Figure 2-3, we plotted temperature versus days of the year. The *y* axis of Figure 2-3 is measured in degrees Fahrenheit, with a grid line every at 20 degrees, and the *x* axis is measured in months, with a grid line at the first of every third month. Whenever the two axes are measured in different units, we can stretch or compress one relative to the other and maintain a valid visualization of the data (Figure 3-2). Which version is preferable may depend on the story we want to convey.

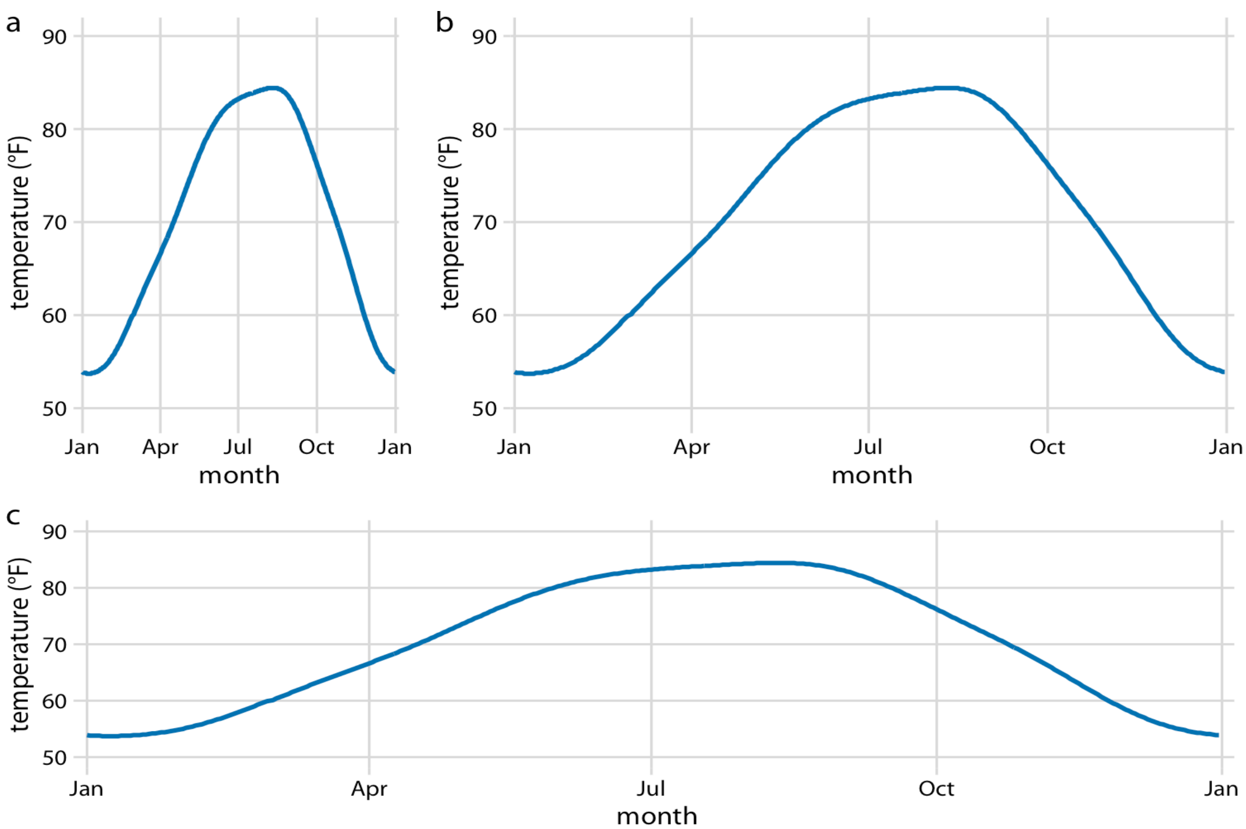


Figure 3-2. Daily temperature normals for Houston, TX. Temperature is mapped to the y *axis and day of the year to the* x *axis. Parts (a), (b), and (c) show the same*

figure in different aspect ratios. All three parts are valid visualizations of the temperature data. Data source: NOAA.

A tall and narrow figure emphasizes change along the *y* axis and a short and wide figure does the opposite. Ideally, we want to choose an aspect ratio that ensures that any important differences in position are noticeable.

On the other hand, if the *x* and *y* axes are measured in the same units, then the grid spacings for the two axes should be equal, such that the same distance along the *x* or *y* axis corresponds to the same number of data units.

Polar Coordinates

The *polar* coordinate system is a coordinate system in which we specify positions via an angle and a radial distance from the origin. It is shown in (Figure 3-9).

Polar coordinates can be useful for data of a periodic nature, such that data values at one end of the scale can be logically joined to data values at the other end.
For example, consider the days in a year. December 31st is the last day of the year, but it is also one day before the first day of the year. If we want to show how some quantity varies over the year, it can be appropriate to use polar coordinates with the angle coordinate specifying each day.

Let's apply this concept to the temperature normals of Figure 2-3.

By plotting the temperature normals in a polar coordinate system, we emphasize this cyclical property they have (Figure 3-10). In comparison to Figure 2-3, the polar version highlights how similar the temperatures are in Death Valley, Houston, and San Diego from late fall to early spring.

In the Cartesian coordinate system, this fact is obscured because the temperature values in late December and in early January are shown in opposite parts of the figure and therefore don't form a single visual unit.

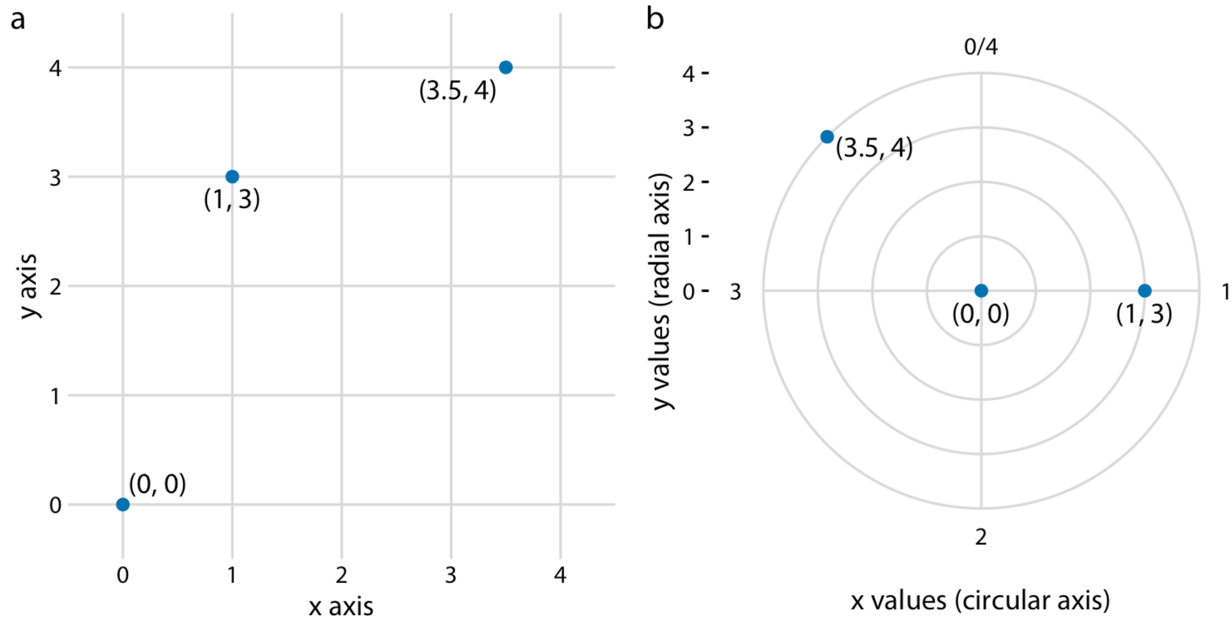


Figure 3-9. Relationship between Cartesian and polar coordinates. (a) Three data points shown in a Cartesian coordinate system. (b) The same three data points shown in a polar coordinate system. We have taken the x *coordinates from part (a) and used them as angular coordinates and the* y *coordinates from part (a) and used them as radial coordinates. The circular axis runs from 0 to 4 in this example, and therefore* x *= 0 and* x *= 4 are the same locations in this coordinate system.*

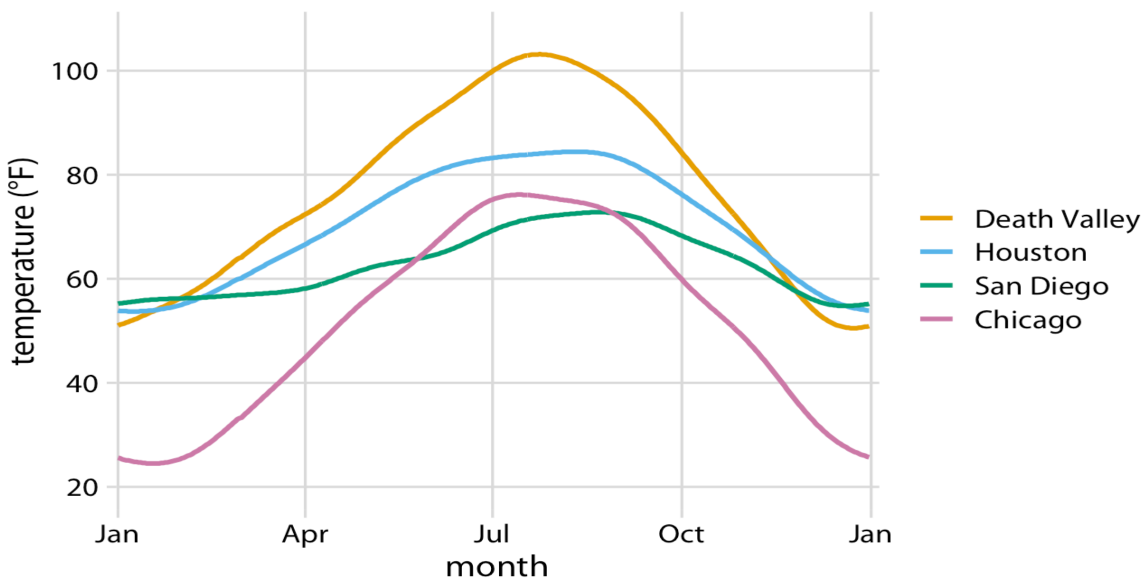


Figure 2-3. Daily temperature normals for four selected locations in the US. Temperature is mapped to the y *axis, day of the year to the* x *axis, and location to line color. Data source: NOAA.*

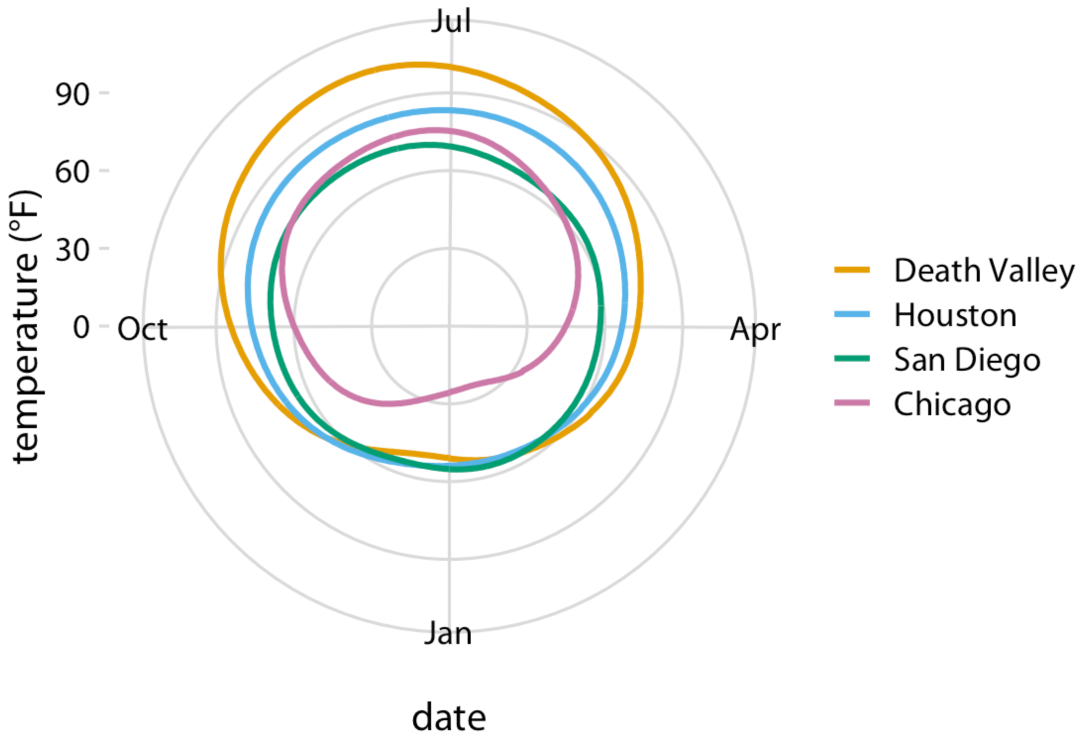
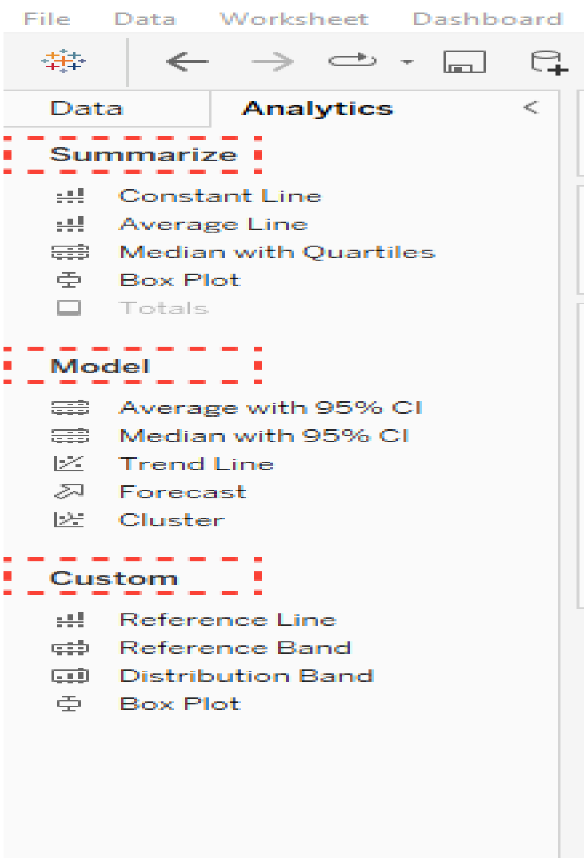


Figure 3-10. Daily temperature normals for four selected locations in the US, shown in polar coordinates. The radial distance from the center point indicates the daily temperature in Fahrenheit, and the days of the year are arranged counterclockwise starting with Jan. 1st. Data source: NOAA.

5. DISCUSS IN DETAIL ABOUT THE DATA ANALYTICS IN TABLEAU.

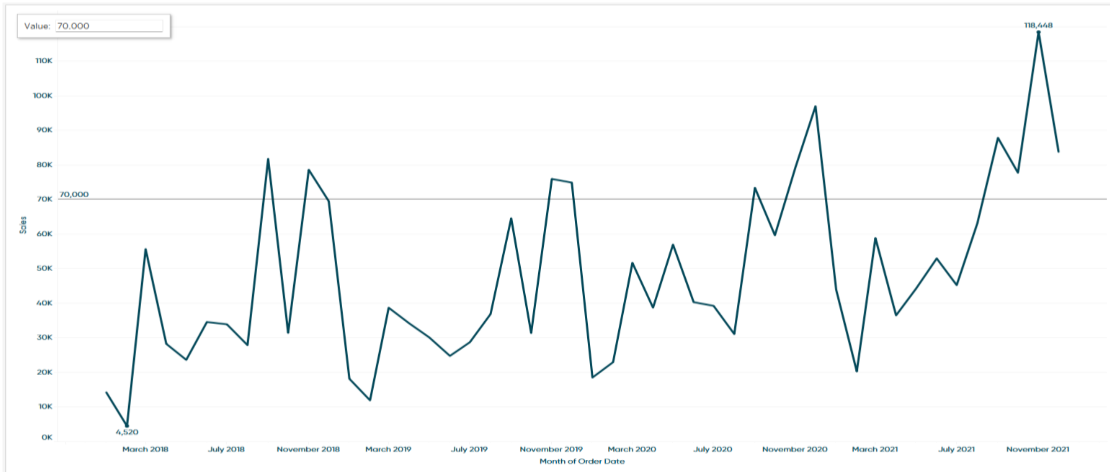
- Tableau provides a set of pre-defined functions in analytics tab for few data analytics functions to be performed.
- It helps to add statistics to the visual analytics.
- It gives audience more insights about the data and helps to make better decisions.



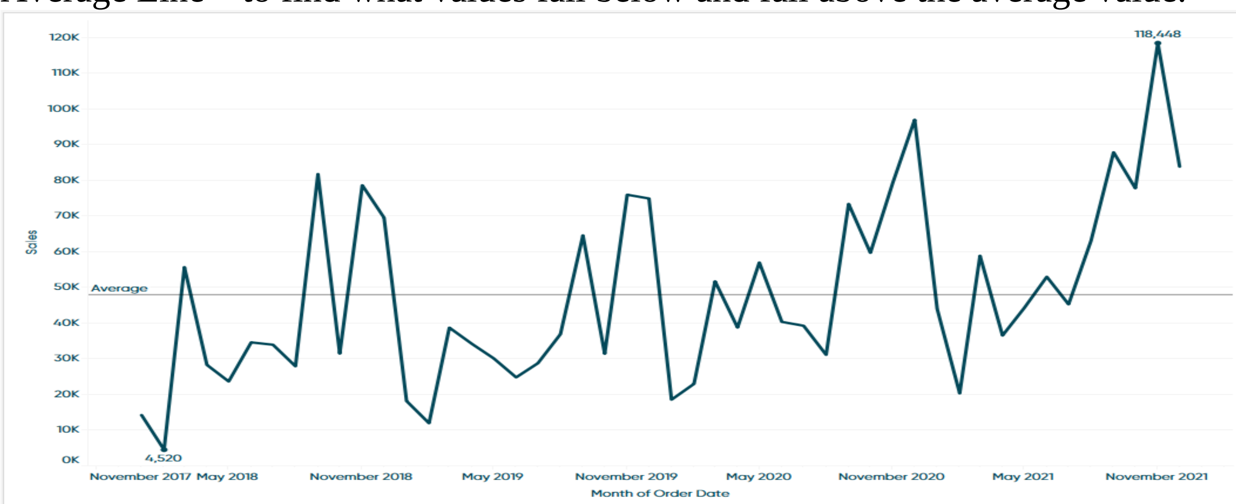
Example: Line chart of sales by month of order of date is drawn.

Summarize Section

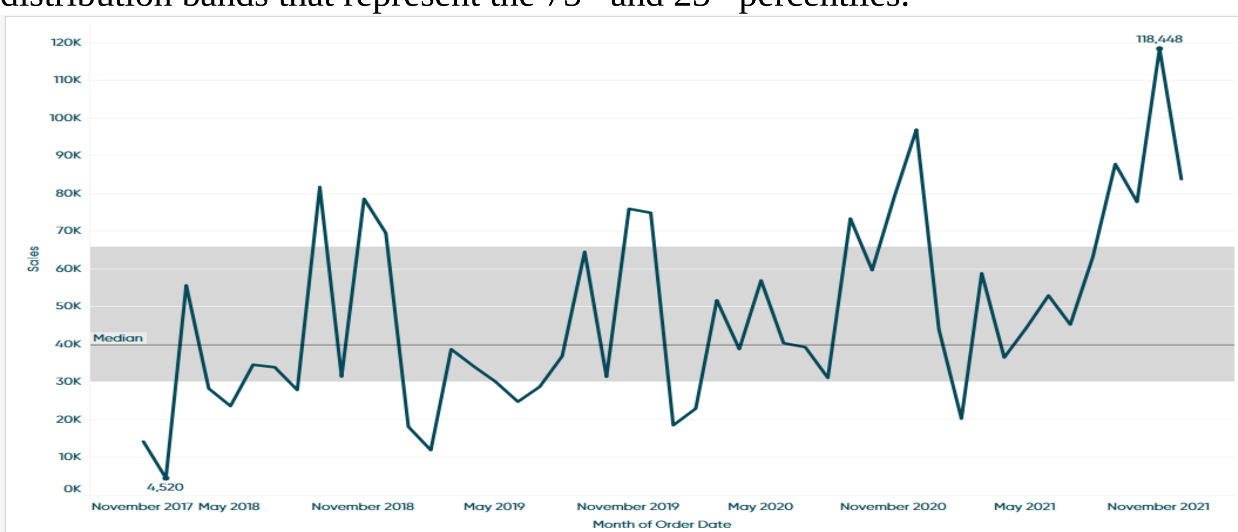
Constant Line - Used to show a specific benchmark on the view as a reference. In the figure below, the value taken for benchmark is 70,000. So, A constant line is drawn at the value of 70,000.



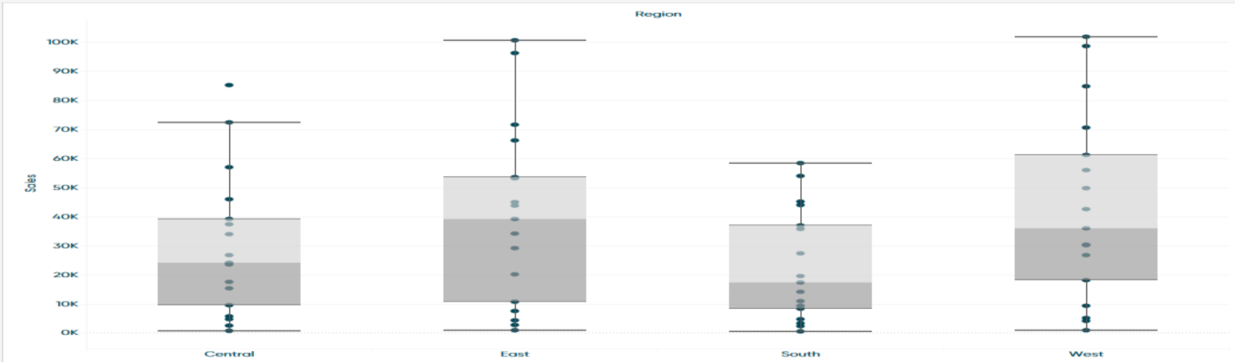
Average Line – to find what values fall below and fall above the average value.



Median with Quartiles – adds a reference line that represents the median and some distribution bands that represent the 75th and 25th percentiles.



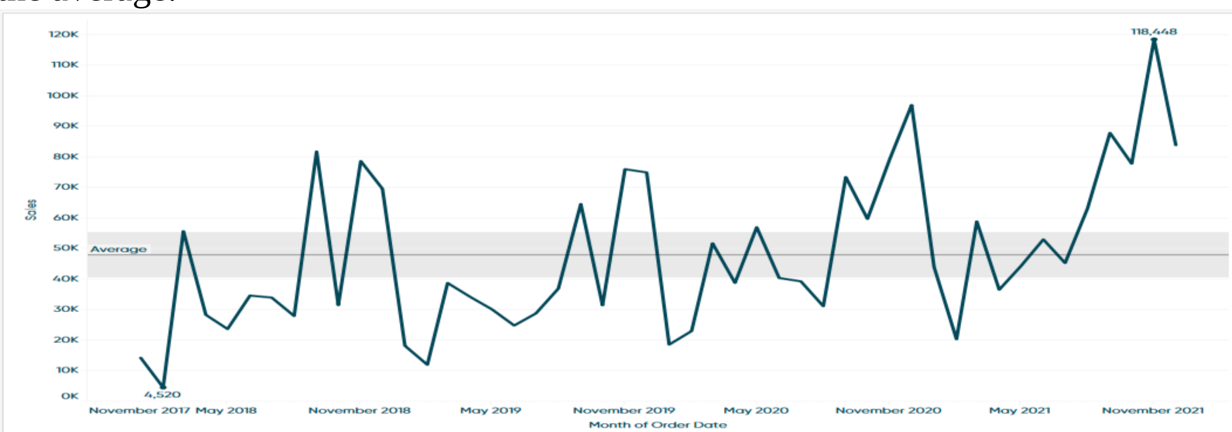
Box Plots – Line charts are not recommended for box plots. So, we go with dot plots. Finding outliers in the data is the great use case for box plots.



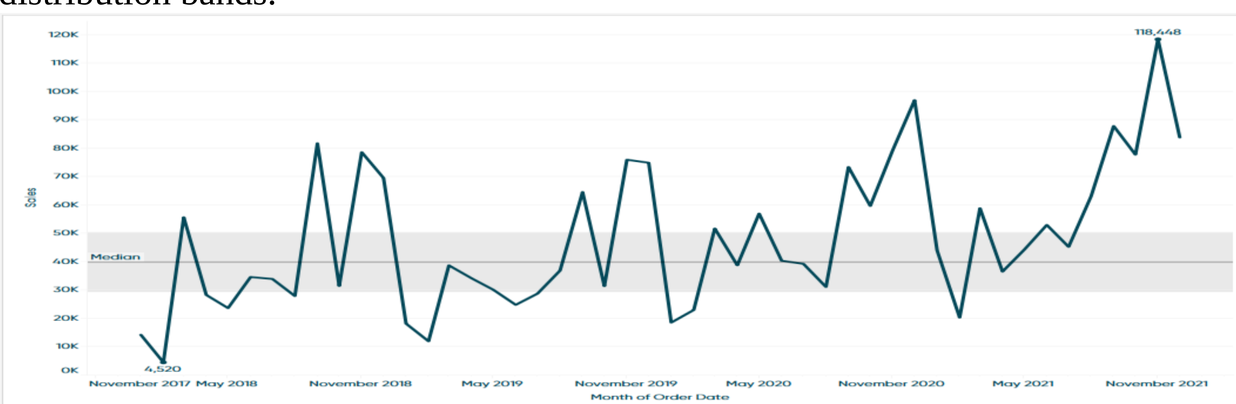
Totals – it is a quick and easy way to view sub-totals, row-grand totals and column-grand totals.

Model Section

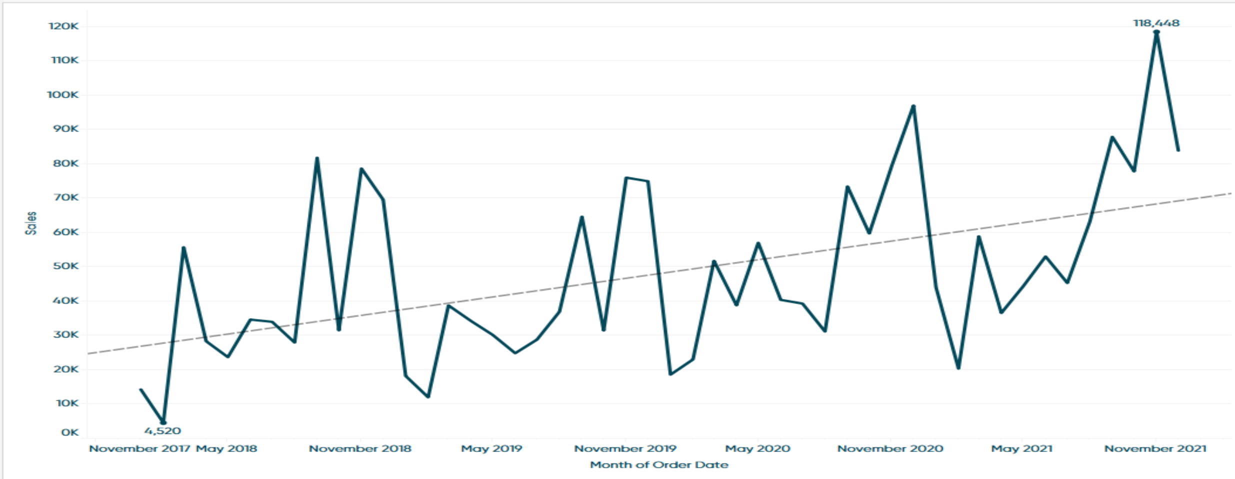
Average with 95% CI – it draws a reference line that represents the average of the data in the view and a distribution band that represents the 95% confidence level of the average.



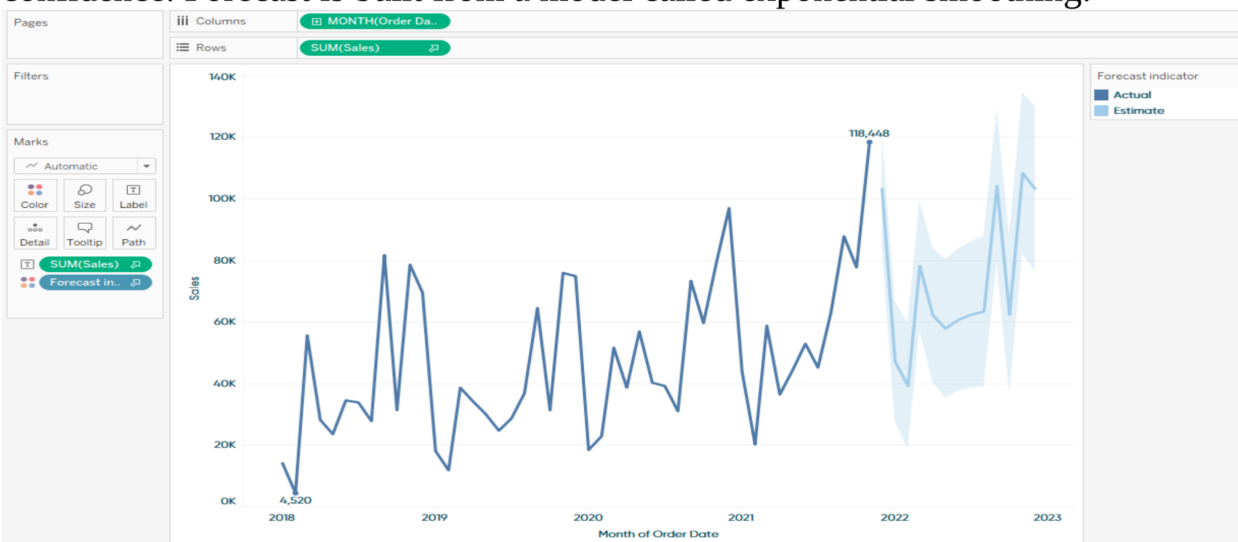
Median with 95% CI – Median is represented by the line in the middle of the distribution bands.



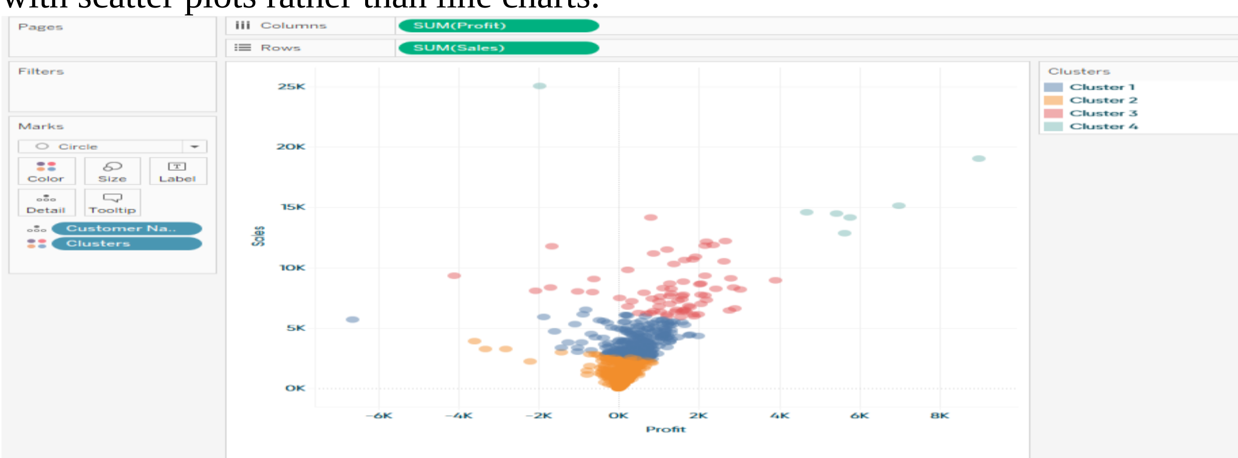
Trend Line – used to draw the trend line for our visualization.



Forecast – It gives us a view on actuals vs. our estimates or predictions with a 95% confidence. Forecast is built from a model called exponential smoothing.



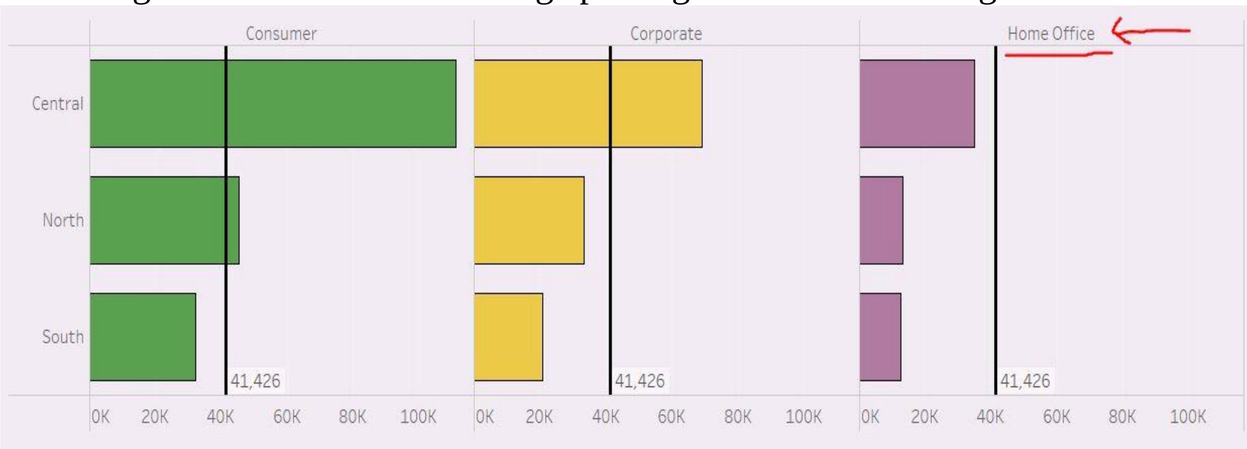
Cluster – it divides the data points into different clusters. It can be viewed efficiently with scatter plots rather than line charts.



The above visualization shows the view with four clusters.

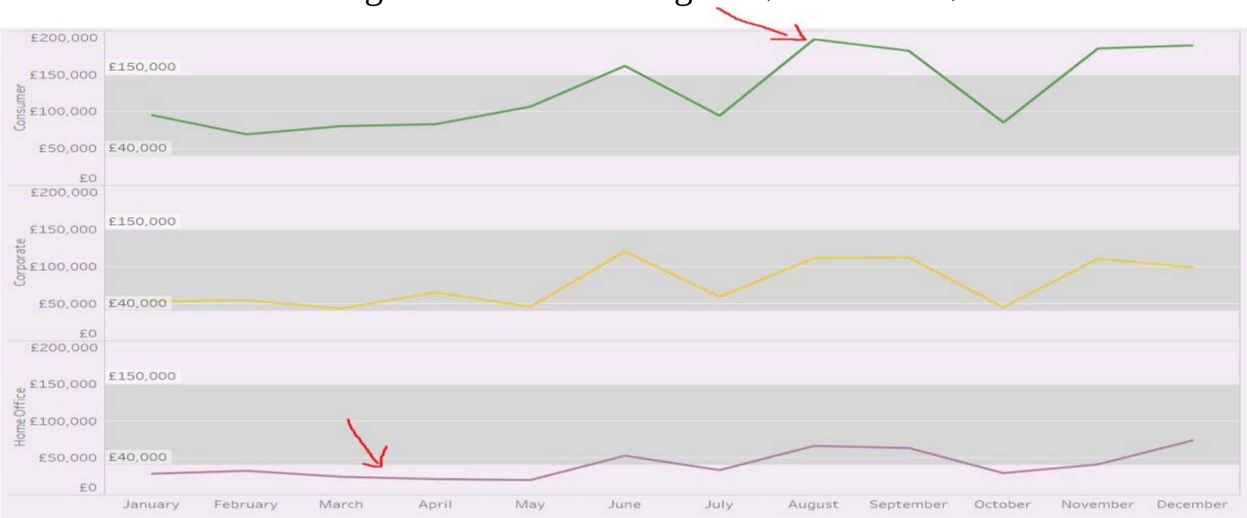
Custom Section

Reference Line – draws a line which can be used as the reference. For example: Find which segment has not hit the average profit goal across all the regions?



In the above view, Home office has not hit the average profit goal.

Reference Band – draws a reference band in the view. For example, Which month have the sales in each segment within the range of $40000 and $ 150000.



6. WHAT IS TABLEAU? MENTION AND EXPLAIN ITS DATA TYPES.

Tableau is a business intelligence tool for visually analyzing the data. It is derived from the French word Tableaux meaning group of models or motionless figure representing a scene from a story.

Data Types in Tableau

Data types characterize the data values into its definite type. Some data values are only text (eg. “DataFlair”) so their data type will be a string as it only contains characters. While some data values have whole numbers (eg. 5550) and so their data type is an integer. If a data value is in the form of a date or time (eg. 22/02/2019 or 03:25:40) then the data type date/time is assigned to it. In a similar way, every data value will fall under a certain category of the data type. Tableau also has a set of data types based on which it characterizes every data value present in it as field values either from a live data connection or as a data extract from the data source.

There are primarily seven data types used in Tableau.

Tableau Data Types

Data Type	Icon
String values (Text)	Abc
Integer values (Numbers)	#
Date values (DD/MM/YYYY or MM/DD/YYYY)	
Date & Time values	
Boolean values (True or False; relational)	T|F
Geographic values (Region, Postal code etc.)	
Cluster group or mixed values	

1. String Data Type

The string data type consists of zero or more characters meaning it is a text data type. A value is considered as a string when the characters are enclosed in a single or double quote (as known as single or double inverted commas). For instance, a string data value can be **“DataFlair”** or **“DataFlair Indore”** etc. You can also incorporate double inverted commas into a string for instance, **‘“Raj says Hello” ’** which will show as **“Raj says Hello”**.
The string data type is further divided into two categories; **Char** and **Varchar**.

- **Char string type:** The values of this data type have a fixed length. Memory corresponding to this fixed length is allocated to char string type values in Tableau. If you try to enter a string value with a length greater than the fixed length, then the system is going to show an error. This data type generally stores alphanumeric data values.
- **Varchar string type:** Varchar string type stands for variable character length where the length of the characters to be entered in a string is not fixed or predefined. You can enter as many characters as you want without having any memory allocation restrictions. This string type can also have alphanumeric data values.

2. Numeric Data Type

The numeric data type values can be either an integer type or floating type. Most fields having numeric data type are of integer type because with float values, users face difficulty in accumulating the decimal point after a certain limit. However, float values can also be used easily in calculations after rounding them up to corresponding integers. A function known as *Round* function is used for rounding up float values.

3. Date and Time Data Types

In Tableau, all formats of date and time are accepted such as dd-mm-yy, dd-mm-yyyy or mm-dd-yyyy, etc. Some fields can only contain a date value or a date can exist along with the time (like a timestamp). The time data values can be decade, year, quarter, month, day, hour, minutes, seconds, etc. Also, if you wish to add a date value in the form of string characters, add a # sign before it (for instance, #22 January 2020). Tableau registers date and time values under two data types which are *Date* data type and *Date & Time* data value.

4. Boolean Data Type

The boolean data type values are a result of relational calculations. Therefore, True and False boolean values are known as logical values. Sometimes, the result of a relational calculation is unknown and obscure. Such values are shown as Null values.

5. Geographic Data Type

The geographic data values are all that are used in a map. Values such as country name, state name, city, region, postal codes, etc. belong to the geographic data type. The fields of this data type are denoted by a globe icon.

6. Cluster or Mixed Data Type

Sometimes a field might not consist of a single data type and rather have a mixture of data types. This is also known as cluster group values or mixed data values. In such scenarios, either you can manually handle such fields by segregating values belonging to different data types into the column or let Tableau handle it.

7. DISCUSS IN DETAIL ABOUT SAVING TABLEAU WORKS. Or DISCUSS IN DETAIL ABOUT DIFFERENT FILE TYPES IN TABLEAU.

Tableau works can be saved in about eight different types of file extensions and folders.

1. Tableau Workbook (.twb)

The Tableau Workbook file type is the one that you will use the most when working in Tableau. This file type has the extension .twb and is set as default for the users.

A workbook in Tableau is a file that contains sheets, dashboards, etc. So, this particular Tableau file type contains information about worksheets and dashboards present within a workbook. All the information regarding the fields, aggregation types, styles, formatting, filters, etc is present in these files.

One important thing to note about this Tableau file type is that we can only create them if we are using live data connections and share them only with the users having access to the same live data connection. The .twb files also contain metadata related to the existing data connection. However, a .twb file does not contain actual data concerning the workbook.

To create .twb file, go to the data source control panel of the active data connection and then go to File option (present on the toolbar) and select Save As.

Then, you can select the file type as Tableau Workbook from the Save As Type drop-down list.

2. Tableau Packaged Workbook (.twbx)

The Tableau Packaged Workbook file type has both information about the constituents of a workbook and the data extracted from the data source.

The data extract taken from the source is present in the form of a .tde file. Tableau Packaged Workbook files have an extension .twbx. You can use a .twbx file type in place of a .twb (Tableau Workbook) file when you wish to share a workbook with a user who does not have access to the live data connection. Thus, in this case, you need to have a file which contains data extracted from the source along with the other information about the workbook.

The Tableau Packaged Workbook files can also contain information about attached images or customized geocoding. To create a .twbx file, go to File then select Save As and then select the .twbx option from the drop-down list given there.

3. Tableau Data Source (.tds)

The ***Tableau Data Source*** files are files that contain all the necessary information regarding a data connection made in Tableau. When we set up a fresh connection to a data source we make a lot of modifications in it as per our requirements such as setting data types, aggregations, custom fields, etc. The Tableau Data Source files contain all the required information on setting up a data connection along with the metadata of other specific modifications made by the users.

The .tds file helps in saving information on data connections with custom fields and table joins. However, this Tableau file type only saves the information needed to establish a connection with a data source but not the actual data. Thus, such files can be used to share information between users having access to the same data source.

To create a .tds file, go to the Data tab on the toolbar. Then choose a data source that you wish to connect to and select Add To Saved Data Source option. After this, save that file as Tableau Data Source file.

4. Tableau Packaged Data Source (.tdsx)

A Tableau Packaged Data Source file is a file that contains information of a data source connection along with the data extracted from that source. The extracted data is saved as a .tde file and the information on data source as .tds file (like we saw above). The extracted data can be from any local file such as a text file, extract files (.hyper or .tde), Excel files, Access files, etc.

However, the extension of a Tableau Packaged Data Source file has the extension as .tdsx. The Tableau Packaged Data Source files are used when we want to share data and other relevant information about a data source with a user who does not have access to the data source and its data.

To create a .tdsx file, go to the Data tab on the toolbar. Then choose a data source that you wish to connect to and select Add To Saved Data Source option. After this, save that file as Tableau Packaged Data Source file.

5. Tableau Data Extract (.tde)

The Tableau Data Extract files have the extension .tde. These Tableau file types only contain a local copy of the entire or a subset of data from its source. It is important to note here that the .tde files do not contain a file path or information about the data source, workbooks, ***dashboards***, etc. Tableau Data Extract files are important and useful as they are highly compressed and optimized to improve Tableau's performance (especially when you are using a slow data connection). You can use .tde files for offline work as well. One noted drawback of such Tableau file types is that the data in it cannot be refreshed automatically as and when it refreshes at the source. However, Tableau has a few step process to refresh the data present as an extract in your .tde files.

To create a Tableau Data Extract (.tde) file, go to the Data tab present on the top left of the Tableau toolbar. Select a data source and click on the Extract Data option. After this, you can either select fields from the data source that you wish to extract or just click on Extract to create a data extract file (.tde) of the entire data set present at the data source.

6. Tableau Bookmark (.tbm)

Files with the extension .tbm are Tableau Bookmark files. These Tableau file types are most commonly used to save worksheets and share them with others so that they can use it in their workbooks without having to create a new worksheet from scratch.

To create a .tbm file, go to the Windows option present on the toolbar. From there, select Bookmark and then click on Create Bookmark. This will create a .tbm file of the active worksheet.

7. Tableau Map Source (.tms)

A Tableau Map Source file contains information about maps and its elements for use in Tableau. The extension of such files is .tms. As per the default settings, Tableau will fetch map details like background and other layers from a certain map server or provider. In Tableau, you have the option to add map details from a WMS server of your choice or a custom map from Mapbox. Once you create a map file (.tms) of your preference, Tableau will fetch map details from that file instead of the default one and load map images and information accordingly. You can also share these .tms users in your group for others to use.

To create a Tableau Map Source (.tms) file, click on Map from the toolbar. Then go to Background Maps and select WMS Server from Map Services> Add. After adding the map server of your choice, you can export it to your local desktop by selecting an Export option from the WMS Server connections window. To use this map in future, add the .tms file into the Tableau Repository in the Map sources directory.

8. Tableau Preference (.tps)

A Tableau Preference file contains all the information related to a customized color palette. You can create a custom color palette or a theme and save it as a .tps file so that you can use it all over the workbook uniformly at once. The Tableau Preference files have the extension as .tps and exist in XML format. These Tableau Preference files are present in My Tableau Repository.

Location of Tableau Repository

All of the Tableau file types are found in their corresponding folders within the My Tableau Repository directory. The Tableau Repository is created automatically upon ***installation of Tableau*** and is found in My documents folder.

You can change the location of your Tableau Repository by selecting the File option in Tableau desktop. Then select Repository Location. Now, in the Select a Repository dialog box select a folder of your choice that will act as the new repository location. This will set the new folder as the new repository location.

DATA VISUALIZATION

UNIT - 1

ONE MARK QUESTION	
1. What is data visualization? The term data visualization refers to the process of transforming data and information into graphical (visual) representation such as graphs, charts and animations.	
2 What do you mean by Aesthetics? The quantifiable features onto which the data visualizations map data values are referred as aesthetics. The aesthetics describe every aspect of a graphical element.	
3. List the commonly used aesthetics in data visualization? Position, shape, size, color, line width, line type.	
4. What is a coordinate system? The combination of a set of position scales and their relative geometric arrangement is called a coordinate system.	
5. List the types of variables used in typical data visualization. Quantitative, qualitative, numerical continuous, numerical discrete, categorical unordered, categorical ordered, Date or time, text.	
6. What is the need for scales in data visualization? The mapping between data values and aesthetic values is created via scales. A scale defines the unique mapping between data and aesthetic. A scale must always be one-to-one.	
7. What is Tableau? It is a data visualization tool that provides pictorial and graphical representation of data.	
8. List the different datatypes in Tableau. Boolean, Date, Number, String, Date & Time and Geolocation.	
9. What is Cartesian coordinate system? It is a coordinate system in which each location is specified by an x and a y value.	
10. List the various tools of analytics tab in Tableau. Forecasting, clustering, trend line, average line, constant line, reference line, reference band, totals, median with quartiles	

TEN MARKS QUESTIONS	
1. Discuss in detail about aesthetics and types of data. 2. How do scales map data values on to aesthetics? Illustrate with an example dataset. 3. Write in detail about Cartesian coordinates. 4. Compare and contrast Cartesian and polar coordinates. 5. Discuss in detail about the data analytics in Tableau. 6. What is Tableau? Mention its data types and explain the steps involved in creating a bar chart for sales forecasts dataset using Tableau. 7. How will you save tableau works? Explain.	

Prepared by, U. Padmavathi, AP, Dept. of AI & DS, MITS

INTERVIEW QUESTIONS

1. What is TABLEAU?

Tableau is the powerful and fastest visualizing tool that is used in the Business Intelligence(BI) Industry. It simplifies the raw data into an understandable format. Analysis of the data becomes faster with Tableau. The visualizations can be created in the form of dashboards. The visualizations or diagrammatic representation of data can easily be understood by the employees of the organizations who are at different levels.

2. What is data visualization?

Data visualization means the graphical representation of data or information. We can use visual objects like graphs, charts, bars, and a lot more. Data visualization tools provide an accessible way to see and understand the data easily.

3. List out Tableau File Extensions.

The below ones are few extensions in Tableau:

- Tableau Workbook (.twb)
- Tableau Data extract (.tde)
- Tableau Datasource (.tds)
- Tableau Packaged Datasource (.tdsx)
- Tableau Bookmark (.tbm)
- Tableau Map Source (.tms)
- Tableau Packaged Workbook (.twbx) – zip file containing .twb and external files.
- Tableau Preferences (.tps)

4. What is the latest version of Tableau Desktop?

Tableau Desktop's latest version is 2021.3(as of, 7thSep 2021).

5. Define LOD Expression?

LOD Expression stands for Level of Detail Expression, and it is used to run complex queries involving many dimensions at the data sourcing level.

6. Define Heat Map?

A heat map is a graphical representation of data that uses the color-coding technique to represent different values of data. As the marks heat up due to their higher value, dark colors will be shown on the map.

7. Define TreeMap?

TreeMap is a visualization that organizes data hierarchically and shows them as a set of nested rectangles. The size and colors of rectangles are respective to the values of the data points they project. Parent rectangles will be tiled with their child elements.

8. What is a parameter Tableau? And how does it work?

Parameters are dynamic values, we can replace the constant values in calculations.

9. What are the different data types in Tableau?
Boolean, Date, Number, String, Date & Time and Geolocation
10. Give a brief about the tableau dashboard?
Tableau dashboard is a group of various views which allows you to compare different types of data simultaneously. Datasheets and dashboards are connected if any modification happens to the data that directly reflects in dashboards. It is the most efficient approach to visualize the data and analyze it.
11. Define Page Shelf in Tableau?
Page shelf breaks the views into a series of pages. It displays an alternate view on each page. Due to this feature, you can analyze the effect of each field on the rest of the data in the view.
12. Define the story in Tableau?
The story can be defined as a sheet which is a collection of series of worksheets and dashboards used to convey the insights of data. A story can be used to show the connection between facts and outcomes that impacts the decision-making process. A story can be published on the web or can be presented to the audience.
13. Give an overview of the fact and dimensions of the table?
Facts are numeric measures of data. They are stored in fact tables. Fact tables store that type of data that will be analyzed by dimension tables. Fact tables have foreign keys associating with dimension tables.
Dimensions are descriptive attributes of data. Those will be stored in the dimensions table. For example, customer's information like name, number, and email will be stored in the dimension table.
14. State some ways to improve the performance of Tableau

- Use an Extract to make workbooks run faster
- Reduce the scope of data to decrease the volume of data
- Reduce the number of marks on the view to avoid information overload
- Try to use integers or Booleans in calculations as they are much faster than strings
- Hide unused fields
- Use Context filters
- Reduce filter usage and use some alternative way to achieve the same result
- Use indexing in tables and use the same fields for filtering
- Remove unnecessary calculations and sheets

References

1. Fundamentals of Data Visualization, by Claus o. Wilke

2. A step by step guide for data visualization using Tableau by Gourav Singh

3. https://www.guru99.com/download-install-tableau.html
4. https://www.thedataschool.co.uk/marcus-grant/tableau-analytics-pane
5. https://playfairdata.com/definitive-guide-to-the-analytics-pane-in-tableau/
6. https://data-flair.training/blogs/tableau-file-types/