A picture containing company name

Description automatically generated

**SCHOOL OF COMPUTING**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**UNIT – III**

**Exploratory Data Analysis and the Data Science Process – SCSA3016**

**UNIT 3 EXPLORATORY DATA ANALYSIS AND THE DATA SCIENCE PROCESS**

Exploratory Data Analysis and the Data Science Process - Basic tools (plots, graphs and summary statistics) of EDA -Philosophy of EDA - The Data Science Process – Data Visualization - Basic principles, ideas and tools for data visualization - Examples of exciting projects- Data Visualization using Tableau.

**3.1 EXPLORATORY DATA ANALYSIS**

Exploratory Data Analysis, or EDA, is an important step in any Data Analysis or Data Science project. EDA is the process of investigating the dataset to discover patterns, and anomalies (outliers), and form hypotheses based on our understanding of the dataset. EDA involves generating summary statistics for numerical data in the dataset and creating various graphical representations to understand the data better.

To explore data in a systematic way, a task that statisticians call exploratory data analysis, or EDA.

EDA is an iterative cycle.

1. Generate questions about your data.
2. Search for answers by visualising, transforming, and modelling your data.
3. Use what you learn to refine your questions and/or generate new questions.

**WHAT IS EDA?**

Exploratory Data Analysis (EDA) is an approach to analyzing datasets to summarize their main characteristics, often with visual methods. EDA is used for seeing what the data can tell us before the modeling task. It is not easy to look at a column of numbers or a whole spreadsheet and determine important characteristics of the data. It may be tedious, boring, and/or overwhelming to derive insights by looking at plain numbers. Exploratory data analysis techniques have been devised as an aid in this situation. EDA assists Data science professionals in various ways: -

1. Getting a better understanding of data
2. Identifying various data patterns
3. Getting a better understanding of the problem statement.

The EDA is important to,

* Detect outliers and anomalies
* Determine the quality of data
* Determine what statistical models can fit the data
* Find out if the assumptions about the data, that you or your team started out with is correct or way off.
* Extract variables or dimensions on which the data can be pivoted.
* Determine whether to apply univariate or multivariate analytical techniques.
* EDA is typically used for these four goals:
* Exploring a single variable and looking at trends over time.
* Checking data for errors.
* Checking assumptions.
* Looking at relationships between variables

## **3.1.1. Why is exploratory data analysis important in data science?**

The main purpose of EDA is to help look at data before making any assumptions. It can help identify obvious errors, as well as better understand patterns within the data, detect outliers or anomalous events, find interesting relations among the variables.

Data scientists can use exploratory analysis to ensure the results they produce are valid and applicable to any desired business outcomes and goals. EDA also helps stakeholders by confirming they are asking the right questions. EDA can help answer questions about standard deviations, categorical variables, and confidence intervals. Once EDA is complete and insights are drawn, its features can then be used for more sophisticated data analysis or modelling, including machine learning.

**3.1.2. Various exploratory data analysis methods like:**

* **Descriptive Statistics**, which is a way of giving a brief overview of the dataset we are dealing with, including some measures and features of the sample.
* **Grouping data** (Basic grouping with *group by*)
* **ANOVA**, Analysis Of Variance, which is a computational method to divide variations in an observations set into different components.
* **Correlation and correlation** methods.

**Descriptive Statistics**: It is a helpful way to understand characteristics of your data and to get a quick summary of it. Pandas in python provide an interesting method **describe()**. The describe function applies basic statistical computations on the dataset like extreme values, count of data points standard deviation etc. Any missing value or NaN value is automatically skipped. describe() function gives a good picture of distribution of data.

**Grouping data**: Group by is an interesting measure available in pandas which can help us figure out effect of different categorical attributes on other data variables.

**ANOVA**

* ANOVA stands for Analysis of Variance. It is performed to figure out the relation between the different group of categorical data.
* Under ANOVA we have two measures as result:   
  – F-testscore : which shows the variation of groups mean over variation   
  – p-value: it shows the importance of the result
* This can be performed using python module scipy method name *f\_oneway()*

**Correlation and Correlation computation**: Correlation is a simple relationship between two variables in a context such that one variable affects the other. Correlation is different from act of *causing*.

**3.1.3. Types of EDA**

Exploratory data analysis is generally cross-classified in two ways. First, each method is either non-graphical or graphical. And second, each method is either univariate or multivariate (usually just bivariate)

There are broadly two categories of EDA, graphical and non-graphical.

* Univariate Non-graphical
* Multivariate Non-graphical
* Univariate graphical
* Multivariate graphical

**Univariate non-graphical**: This is the simplest form of data analysis as during this we use just one variable to research the info. The standard goal of univariate non-graphical EDA is to know the underlying sample distribution/ data and make observations about the population. Outlier detection is additionally part of the analysis.

The characteristics of population distribution include:

* **Central tendency:**  The central tendency or location of distribution has got to do with typical or middle values. The commonly useful measures of central tendency are statistics called mean, median, and sometimes mode during which the foremost common is mean. For skewed distribution or when there’s concern about outliers, the median may be preferred.
* **Spread:**Spread is an indicator of what proportion distant from the middle we are to seek out the find the info values. the quality deviation and variance are two useful measures of spread. The variance is that the mean of the square of the individual deviations and therefore the variance is the root of the variance
* **Skewness and kurtosis:**Two more useful univariates descriptors are the skewness and kurtosis of the distribution. Skewness is that the measure of asymmetry and kurtosis may be a more subtle measure of peakedness compared to a normal distribution

**Multivariate non-graphical**: Multivariate non-graphical EDA technique is usually wont to show the connection between two or more variables within the sort of either cross-tabulation or statistics.

* + For categorical data, an extension of tabulation called cross-tabulation is extremely useful. For 2 variables, cross-tabulation is preferred by making a two-way table with column headings that match the amount of one-variable and row headings that match the amount of the opposite two variables, then filling the counts with all subjects that share an equivalent pair of levels.
  + For each categorical variable and one quantitative variable, we create statistics for quantitative variables separately for every level of the specific variable then compare the statistics across the amount of categorical variable.
  + Comparing the means is an off-the-cuff version of ANOVA and comparing medians may be a robust version of one-way ANOVA.

**Univariate graphical:** Non-graphical methods are quantitative and objective, they are doing not give the complete picture of the data; therefore, graphical methods are more involve a degree of subjective analysis also are required.

**Common sorts of univariate graphics are:**

* **Histogram:**The foremost basic graph is a histogram, which may be a bar plot during which each bar represents the frequency (count) or proportion (count/total count) of cases for a variety of values. Histograms are one of the simplest ways to quickly learn a lot about your data, including central tendency, spread, modality, shape and outliers.
* **Stem-and-leaf plots:** An easy substitute for a histogram may be stem-and-leaf plots. It shows all data values and therefore the shape of the distribution.
* **Boxplots:**Another very useful univariate graphical technique is that the boxplot. Boxplots are excellent at presenting information about central tendency and show robust measures of location and spread also as providing information about symmetry and outliers, although they will be misleading about aspects like multimodality. One among the simplest uses of boxplots is within the sort of side-by-side boxplots.
* **Quantile-normal plots:**The ultimate univariate graphical EDA technique is that the most intricate. it’s called the quantile-normal or QN plot or more generally the quantile-quantile or QQ plot. it’s wont to see how well a specific sample follows a specific theoretical distribution. It allows detection of non-normality and diagnosis of skewness and kurtosis

**Multivariate graphical:** A graphical representation always gives you a better understanding of the relationship, especially among multiple variables.

**Other common sorts of multivariate graphics are:**

* **Scatterplot:** For 2 quantitative variables, the essential graphical EDA technique is that the scatterplot, sohas one variable on the x-axis and one on the y-axis and therefore the point for every case in your dataset.
* **Run chart:**  It’s a line graph of data plotted over time.
* **Heat map:**  It’s a graphical representation of data where values are depicted by color.
* **Multivariate chart:** It’s a graphical representation of the relationships between factors and response.
* **Bubble chart:** It’s a data visualization that displays multiple circles (bubbles) in two-dimensional plot.

**3.2. TOOLS in EDA**

**Non-graphical exploratory data analysis** involves data collection and reporting in nonvisual or non-pictorial formats. Some of the most common data science tools used to create an EDA include:

* **Python:** An interpreted, object-oriented programming language with dynamic semantics. Its high-level, built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for rapid application development, as well as for use as a scripting or glue language to connect existing components together. Python and EDA can be used together to identify missing values in a data set, which is important so you can decide how to handle missing values for machine learning.
* **R:** An open-source programming language and free software environment for statistical computing and graphics supported by the R Foundation for Statistical Computing.

The R language is widely used among statisticians in data science in developing statistical observations and data analysis.

**Graphical exploratory data analysis** employs visual tools to display data, such as:

**Box plots**

* Box plots are used where there is a need to summarize data on an interval scale like the ones on the stock market, where ticks observed in one whole day may be represented in a single box, highlighting the lowest, highest, median and outliers.

**Heatmap**

* Heatmaps are most often used for the representation of the correlation between variables. Here is an example of a heatmap.
* As you can see from the chart, there is a strong correlation between density and residual sugar and absolutely no correlation between alcohol and residual sugar.

**Histograms**

* The histogram is the graphical representation of numerical data that splits the data into ranges. The taller the bar, the greater the number of data points falling in that range. A good example here is the height data of a class of students. You would notice that the height data looks like a bell curves for a particular class with most the data lying within a certain range and a few of outside these ranges. There will be outliers too, either very short or very small.

**Line graphs**: one of the most basic types of charts that plots data points on a graph; has a wealth of uses in almost every field of study.

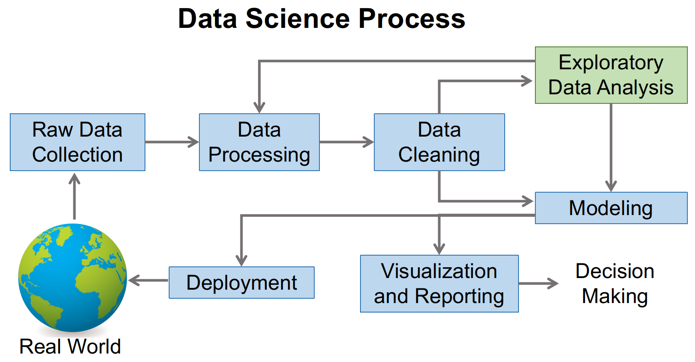
**Pictograms**: replace numbers with images to visually explain data. They’re common in the design of infographics, as well as visuals that data scientists can use to explain complex findings to non-data-scientist professionals and the public.

**Scattergrams or scatterplots**: typically used to display two variables in a set of data and then look for correlations among the data. For example, scientists might use it to evaluate the presence of two particular chemicals or gases in marine life in an effort to look for a relationship between the two variables.

**3.3 PHILOSOPHY OF EDA**

* The father of EDA is John Tukey who officially coined the term in his [1977 masterpiece](https://books.google.com.au/books/about/Exploratory_Data_Analysis.html?id=UT9dAAAAIAAJ). Lyle Jones, the editor of the multi-volume “The collected works of John W. Tukey: Philosophy and principles of data analysis” describes EDA as “an attitude towards flexibility that is absent of prejudice”.
* The key frame of mind when engaging with EDA and thus VDA is to approach the dataset with little to no expectation, and not be influenced by rigid parameterisations. EDA commands to let the data speak for itself. To use the words of Tukey (1977, preface):
* “It is important to understand what you CAN DO before you learn to measure how WELL you seem to have DONE it… Exploratory data analysis can never be the whole story, but nothing else can serve as the foundation stone –as the first step.”
* Since the inception of EDA as unifying class of methods, it has influenced the development of several other major statistical developments including in non-parametric statistics, robust analysis, data mining, and visual data analytics. These classes of methods are motivated by the need to stop relying on rigid assumption-driven mathematical formulations that often fail to be confirmed by observables.
* EDA is not identical to statistical graphics although the two terms are used almost interchangeably. Statistical graphics is a collection of techniques--all graphically based and all focusing on one data characterization aspect. EDA encompasses a larger venue; EDA is an approach to data analysis that postpones the usual assumptions about what kind of model the data follow with the more direct approach of allowing the data itself to reveal its underlying structure and model. EDA is not a mere collection of techniques; EDA is a philosophy as to how we dissect a data set; what we look for; how we look; and how we interpret. It is true that EDA heavily uses the collection of techniques that we call "statistical graphics", but it is not identical to statistical graphics.

**3.4. DATA SCIENCE PROCESS**

****

**Figure 3.1: Data Science process**

The key steps involved in Data Science Modelling are:

Step 1: Understanding the Problem

Step 2: Data Extraction

Step 3: Data Cleaning

Step 4: Exploratory Data Analysis

Step 5: Feature Selection

Step 6: Incorporating Machine Learning Algorithms

Step 7: Testing the Models

Step 8: Deploying the Model

**Step 1: Understanding the Problem**

The first step involved in Data Science Modelling is understanding the problem. A Data Scientist listens for keywords and phrases when interviewing a line-of-business expert about a business challenge. The Data Scientist breaks down the problem into a procedural flow that always involves a holistic understanding of the business challenge, the Data that must be collected, and various Artificial Intelligence and Data Science approach that can be used to address the problem.

**Step 2: Data Extraction**

* The next step in Data Science Modelling is Data Extraction. Not just any Data, but the Unstructured Data pieces you collect, relevant to the business problem you’re trying to address. The Data Extraction is done from various sources online, surveys, and existing Databases.

**Step 3: Data Cleaning**

Data Cleaning is useful as you need to sanitize Data while gathering it. Data cleaningis the process of detecting, correcting and ensuring that your given data set is free from error, consistent and usable by identifying any errors or corruptions in the data, correcting or deleting them, or manually processing them as needed to prevent the error from corrupting our final analysis.

The following are some of the most typical causes of Data Inconsistencies and Errors:

* Duplicate items are reduced from a variety of Databases.
* The error with the input Data in terms of Precision.
* Changes, Updates, and Deletions are made to the Data entries.
* Variables with missing values across multiple Databases.

**Steps In Data Preprocessing:**

* Gathering the data
* Import the dataset & Libraries
* Dealing with Missing Values
* Divide the dataset into Dependent & Independent variable
* dealing with Categorical values
* Split the dataset into training and test set
* Feature Scaling

**Gathering the data**

* Data is raw information, its the representation of both human and machine observation of the world. Dataset entirely depends on what type of problem you want to solve. Each problem in machine learning has its own unique approach.

Some website to get the dataset :

* Kaggle:  
  <https://www.kaggle.com/datasets>
* UCI Machine Learning Repository: One of the oldest sources on the web to get the dataset.  
  <http://mlr.cs.umass.edu/ml/>
* This awesome GitHub repository has high-quality datasets.  
  <https://github.com/awesomedata/awesome-public-datasets>

**Import the dataset & Libraries**

* First step is usually importing the libraries that will be needed in the program. A library is essentially a collection of modules that can be called and used.
* [Pandas](https://www.geeksforgeeks.org/python-pandas-dataframe/) offer tools for cleaning and process your data. It is the most popular Python library that is used for data analysis. In pandas, a data table is called a dataframe.

**Dealing with Missing Values**

* Sometimes we may find some data are missing in the dataset. if we found then we will remove those rows or we can calculate either **mean, mode or median**of the feature and replace it with missing values. This is an approximation which can add variance to the dataset.

#**Check for null values-**dataset.isna() or dataset.isnull() to see the null values in dataset.

#**Drop Null values-** Pandas provide a **dropna()** function that can be used to drop either row or columns with missing data.

**#Replacing Null values with Strategy:** For replacing null values we use the strategy that can be applied on a feature which has numeric data. We can calculate the *Mean, Median or Mode* of the feature and replace it with the missing values.

* De-Duplicate means remove all duplicate values. There is no need for duplicate values in data analysis. These values only affect the accuracy and efficiency of the analysis result. To find duplicate values in the dataset we will use a simple dataframe function i.e. duplicated(). Let’s see the example:

dataset.duplicated()

**Feature Scaling**

* The final step of data preprocessing is to apply the very important feature scaling.
* Feature Scaling is a technique to standardize the independent features present in the data in a fixed range. It is performed during the data pre-processing.
* **Why Scaling** :- Most of the times, your dataset will contain features highly varying in magnitudes, units and range. But since, most of the machine learning algorithms use Euclidean distance between two data points in their computations, this is a problem.

**Standardization and Normalization**

* Data Standardization and Normalization is a common practice in machine learning.
* Standardization is another scaling technique where the values are centered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation.
* Normalization is a scaling technique in which values are shifted and rescaled so that they end up ranging between 0 and 1. It is also known as Min-Max scaling.

**Step 4: Exploratory Data Analysis**

* Exploratory Data Analysis (EDA) is a robust technique for familiarising yourself with Data and extracting useful insights. Data Scientists sift through Unstructured Data to find patterns and infer relationships between Data elements. Data Scientists use Statistics and Visualisation tools to summarise Central Measurements and variability to perform EDA.

**Step 5: Feature Selection**

* Feature Selection is the process of identifying and selecting the features that contribute the most to the prediction variable or output that you are interested in, either automatically or manually.
* The presence of irrelevant characteristics in your Data can reduce the Model accuracy and cause your Model to train based on irrelevant features. In other words, if the features are strong enough, the Machine Learning Algorithm will give fantastic outcomes.
* Two types of characteristics must be addressed:
  + Consistent characteristics that are unlikely to change.

Variable characteristics whose values change over time

**Step 6: Incorporating Machine Learning Algorithms**

* This is one of the most crucial processes in Data Science Modelling as the Machine Learning Algorithm aids in creating a usable Data Model. There are a lot of algorithms to pick from, the Model is selected based on the problem. There are three types of Machine Learning methods that are incorporated:

**1) Supervised Learning**

* It is based on the results of a previous operation that is related to the existing business operation. Based on previous patterns, Supervised Learning aids in the prediction of an outcome. Some of the Supervised Learning Algorithms are:
  + Linear Regression
  + Random Forest
  + Support Vector Machines

**2) Unsupervised Learning**

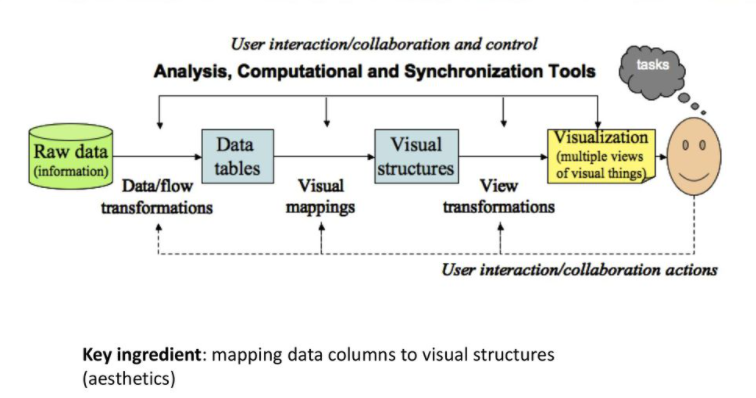
* This form of learning has no pre-existing consequence or pattern. Instead, it concentrates on examining the interactions and connections between the presently available Data points. Some of the Unsupervised Learning Algorithms are:
  + KNN (k-Nearest Neighbors)
  + K-means Clustering
  + Hierarchical Clustering
  + Anomaly Detection

**3.5 DATA VISUALIZATION**

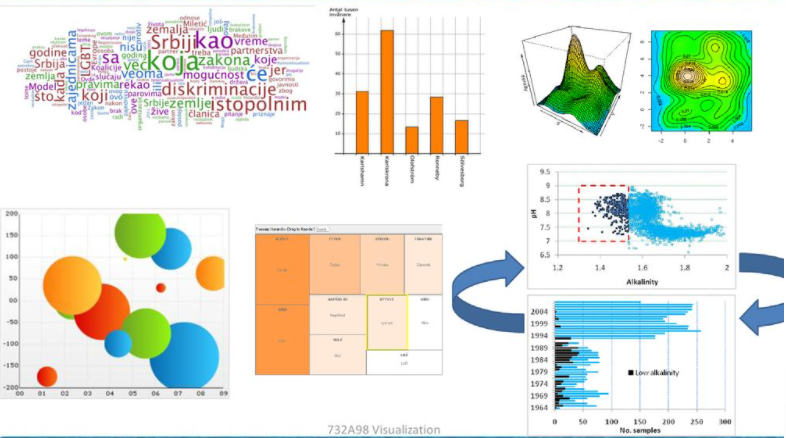
* Data visualization is the process of translating large data sets and metrics into charts, graphs and other visuals.
* The resulting visual representation of data makes it easier to identify and share real-time trends, outliers, and new insights about the information represented in the data.
* Data visualization is one of the steps of the [data science](https://www.techtarget.com/searchenterpriseai/definition/data-science) process, which states that after data has been collected, processed and modeled, it must be visualized for conclusions to be made.
* Data visualization is also an element of the broader data presentation architecture (DPA) discipline, which aims to identify, locate, manipulate, format and deliver data in the most efficient way possible.

**Why Data Visualization is important?**

* It’s hard to think of a professional industry that doesn’t benefit from making data more understandable. Every STEM field benefits from understanding data—and so do fields in government, finance, marketing, history, consumer goods, service industries, education, sports, and so on. And, since visualization is so prolific, it’s also one of the most useful professional skills to develop. The better we can convey the points visually, whether in a dashboard or a slide deck, the better we can leverage that information. The concept of the citizen data scientist is on the rise. Skill sets are changing to accommodate a data-driven world. It is increasingly valuable for professionals to be able to use data to make decisions and use visuals to tell stories of when data informs the who, what, when, where, and how. While traditional education typically draws a distinct line between creative storytelling and technical analysis, the modern professional world also values those who can cross between the two: data visualization sits right in the middle of analysis and visual storytelling.



* **Some examples of Data Visualization**



**Common general types of data visualization:**

* Charts
* Tables
* Graphs
* Maps
* Infographics
* Dashboards
* **More specific examples of methods to visualize data:**
* Area Chart
* Bar Chart
* Box-and-whisker Plots
* Bubble Cloud
* [Bullet Graph](https://www.tableau.com/data-insights/reference-library/visual-analytics/charts/bullet-graphs)
* Cartogram
* Circle View
* Dot Distribution Map
* Gantt Chart
* Heat Map
* Highlight Table
* Histogram
* Matrix
* Network
* Polar Area
* Radial Tree
* Scatter Plot (2D or 3D)
* Streamgraph
* Text Tables
* Timeline
* Treemap
* Wedge Stack Graph
* Word Cloud
* And any mix-and-match combination in a dashboard

**Challenges in Data Visualization**

* Which graphs can be used for analysis of my data?
* How to create these graphs
* How should these graphs be analysed?
* How to make these graphs looking good for publication or presentation?

**3.6. Data Visualization Tools**

**1. Tableau**

* It is a business intelligence service that aids people in visualizing as well as understanding their data it’s also one of those very widely used services in the field of business intelligence. It allows you to design an interactive reports dashboard and worksheets to obtain business visions it has outstanding visualization capabilities and has a great performance.

Pros:

* Outstanding visual library
* User friendly
* Great performance
* Connectivity to data
* Powerful computation
* Quick insights

Cons:

* Inflexible pricing
* No option for auto-refresh
* Restrictive imports
* Manual updates for static features

**2. Power BI**

* Power BI, Microsoft's easy-to-use data visualization tool, is available for both on-premise installation and deployment on the cloud infrastructure. Power BI is one of the most complete data visualization tools that supports a myriad of backend databases, including Teradata, Salesforce, PostgreSQL, Oracle, Google Analytics, Github, Adobe Analytics, Azure, SQL Server, and Excel. The enterprise-level tool creates stunning visualizations and delivers real-time insights for fast decision-making.

The Pros of Power BI:

* No requirement for specialized tech support
* Easily integrates with existing applications
* Personalized, rich dashboard
* High-grade security
* No speed or memory constraints
* Compatible with Microsoft products

The Cons of Power BI:

* Cannot work with varied, multiple datasets

**3. Dundas BI**

* Dundas BI offers highly-customizable data visualizations with interactive scorecards, maps, gauges, and charts, optimizing the creation of ad-hoc, multi-page reports. By providing users full control over visual elements, Dundas BI simplifies the complex operation of cleansing, inspecting, transforming, and modeling big datasets.

The Pros of Dundas BI:

* Exceptional flexibility
* A large variety of data sources and charts
* Wide range of in-built features for extracting, displaying, and modifying data

The Cons of Dundas BI:

* No option for predictive analytics
* 3D charts not supported

**4. JupyteR**

* A web-based application, JupyteR, is one of the top-rated data visualization tools that enable users to create and share documents containing visualizations, equations, narrative text, and live code. JupyteR is ideal for data cleansing and transformation, statistical modeling, numerical simulation, interactive computing, and [machine learning](https://www.simplilearn.com/tutorials/machine-learning-tutorial/what-is-machine-learning).

The Pros of JupyteR:

* Rapid prototyping
* Visually appealing results
* Facilitates easy sharing of data insights

The Cons of JupyteR:

* Tough to collaborate
* At times code reviewing becomes complicated

**5. Zoho Reports**

* Zoho Reports, also known as Zoho Analytics, is a comprehensive data visualization tool that integrates Business Intelligence and online reporting services, which allow quick creation and sharing of extensive reports in minutes. The high-grade visualization tool also supports the import of Big Data from major databases and applications.

**The Pros of Zoho Reports:**

* Effortless report creation and modification
* Includes useful functionalities such as email scheduling and report sharing
* Plenty of room for data
* Prompt customer support.

**The Cons of Zoho Reports:**

* User training needs to be improved
* The dashboard becomes confusing when there are large volumes of data

**6. GoogleCharts**

* One of the major players in the data visualization market space, Google Charts, coded with SVG and [HTML5](https://www.simplilearn.com/tutorials/html-tutorial/html-vs-html5), is famed for its capability to produce graphical and pictorial data visualizations. Google Charts offers zoom functionality, and it provides users with unmatched cross-platform compatibility with iOS, Android, and even the earlier versions of the Internet Explorer browser.

**The Pros of Google Charts:**

* User-friendly platform
* Easy to integrate data
* Visually attractive data graphs
* Compatibility with Google products.

**The Cons of Google Charts:**

* The export feature needs fine-tuning
* Inadequate demos on tools
* Lacks customization abilities
* Network connectivity required for visualization

**7. Sisense**

Regarded as one of the most agile data visualization tools, Sisense gives users access to instant data analytics anywhere, at any time. The best-in-class visualization tool can identify key data patterns and summarize statistics to help decision-makers make data-driven decisions.

**The Pros of Sisense:**

* Ideal for mission-critical projects involving massive datasets
* Reliable interface
* High-class customer support
* Quick upgrades
* Flexibility of seamless customization

**The Cons of Sisense:**

* Developing and maintaining analytic cubes can be challenging
* Does not support time formats
* Limited visualization versions

**8. Plotly**

* An open-source data visualization tool, Plotly offers full integration with analytics-centric programming languages like Matlab, Python, and R, which enables complex visualizations. Widely used for collaborative work, disseminating, modifying, creating, and sharing interactive, graphical data, Plotly supports both on-premise installation and cloud deployment.

**The Pros of Plotly:**

* Allows online editing of charts
* High-quality image export
* Highly interactive interface
* Server hosting facilitates easy sharing

**The Cons of Plotly:**

* Speed is a concern at times
* Free version has multiple limitations
* Various screen-flashings create confusion and distraction

**9. Data Wrapper**

* Data Wrapper is one of the very few data visualization tools on the market that is available for free. It is popular among media enterprises because of its inherent ability to quickly create charts and present graphical statistics on Big Data. Featuring a simple and intuitive interface, Data Wrapper allows users to create maps and charts that they can easily embed into reports.

**The Pros of Data Wrapper:**

* Does not require installation for chart creation
* Ideal for beginners
* Free to use

**The Cons of Data Wrapper:**

* Building complex charts like Sankey is a problem
* Security is an issue as it is an open-source tool

**10. QlikView**

A major player in the data visualization market, Qlikview provides solutions to over 40,000 clients in 100 countries. Qlikview's data visualization tool, besides enabling accelerated, customized visualizations, also incorporates a range of solid features, including analytics, enterprise reporting, and Business Intelligence capabilities.

**The Pros of QlikView:**

* User-friendly interface
* Appealing, colorful visualizations
* Trouble-free maintenance
* A cost-effective solution

**The Cons of QlikView:**

* RAM limitations
* Poor customer support
* Does not include the 'drag and drop' feature

**3.7. DATA VISUALIZATION WITH PYTHON**

Python offers multiple great graphing libraries that come packed with lots of different features.

Here are a few popular plotting libraries:

* [Matplotlib:](https://matplotlib.org/) low level, provides lots of freedom
* [Pandas Visualization:](https://pandas.pydata.org/pandas-docs/stable/visualization.html) easy to use interface, built on Matplotlib
* [Seaborn:](https://seaborn.pydata.org/) high-level interface, great default styles
* [ggplot:](http://ggplot.yhathq.com/) based on R’s ggplot2, uses [Grammar of Graphics](https://www.amazon.com/Grammar-Graphics-Statistics-Computing/dp/0387245448)
* [Plotly:](https://plot.ly/python/) can create interactive plots

**Matplotlib**

* Matplotlib is a visualization library in Python for 2D plots of arrays. Matplotlib is written in Python and makes use of the NumPy library. It can be used in Python and IPython shells, Jupyter notebook, and web application servers. Matplotlib comes with a wide variety of plots like line, bar, scatter, histogram, etc. which can help us, deep-dive, into understanding trends, patterns, correlations. It was introduced by John Hunter in 2002.

**Seaborn**

* Conceptualized and built originally at the Stanford University, this library sits on top of *matplotlib*. In a sense, it has some flavors of *matplotlib* while from the visualization point, its is much better than *matplotlib* and has added features as well. Below are its advantages
  + Built-in themes aid better visualization
  + Statistical functions aiding better data insights
  + Better aesthetics and built-in plots
  + Helpful documentation with effective examples

**Bokeh**

* Bokeh is an interactive visualization library for modern web browsers. It is suitable for large or streaming data assets and can be used to develop interactive plots and dashboards. There is a wide array of intuitive graphs in the library which can be leveraged to develop solutions. It works closely with PyData tools. The library is well-suited for creating customized visuals according to required use-cases. The visuals can also be made interactive to serve a what-if scenario model. All the codes are open source and available on GitHub.

**plotly**

* plotly.py is an interactive, open-source, high-level, declarative, and browser-based visualization library for Python. It holds an array of useful visualization which includes scientific charts, 3D graphs, statistical charts, financial charts among others. Plotly graphs can be viewed in Jupyter notebooks, standalone HTML files, or hosted online. Plotly library provides options for interaction and editing. The robust API works perfectly in both local and web browser mode.

**plotly**

* plotly.py is an interactive, open-source, high-level, declarative, and browser-based visualization library for Python. It holds an array of useful visualization which includes scientific charts, 3D graphs, statistical charts, financial charts among others. Plotly graphs can be viewed in Jupyter notebooks, standalone HTML files, or hosted online. Plotly library provides options for interaction and editing. The robust API works perfectly in both local and web browser mode.

**ggplot**

* ggplot is a Python implementation of the grammar of graphics. The Grammar of Graphics refers to the mapping of data to aesthetic attributes (colour, shape, size) and geometric objects (points, lines, bars). The basic building blocks according to the grammar of graphics are data, geom (geometric objects), stats (statistical transformations), scale, coordinate system, and facet.
* Using ggplot in Python allows you to develop informative visualizations incrementally, understanding the nuances of the data first, and then tuning the components to improve the visual representations.

**3.9. Examples Of Exciting Projects- Exploratory Data Analysis : Iris Dataset**

Importing relevant libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn import metrics

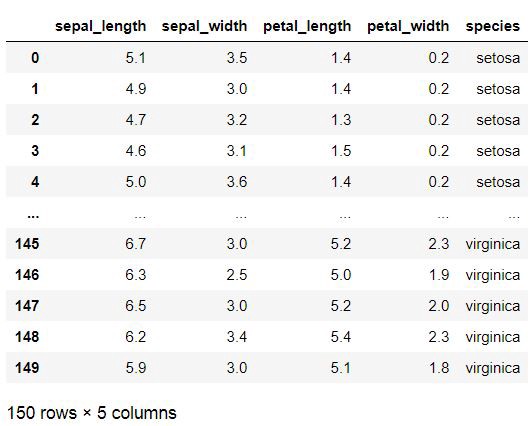
sns.set()

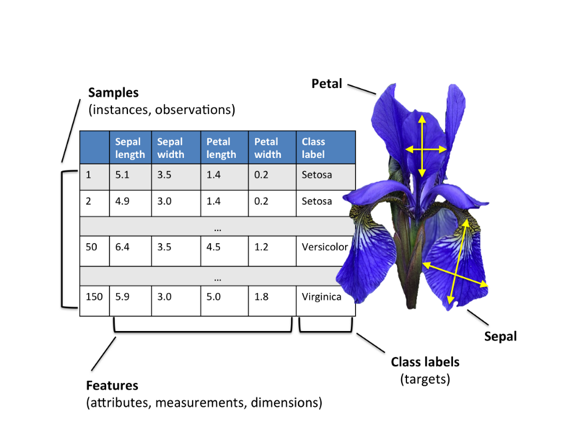
**Source Of Data**

* Data has been stored inside a csv file namely ‘iris.csv’

**Loading data**

* *iris\_data = pd.read\_csv(‘iris.csv’)*
* *iris\_data*





**Getting Information about the Dataset**

* We will use the shape parameter to get the shape of the dataset.

*iris\_data.shape*

**Output:**

* (150, 6)We can see that the dataframe contains 6 columns and 150 rows.

**Gaining information from data**

*iris\_data.info()*

<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 150 entries, 0 to 149  
Data columns (total 5 columns):  
# Column Non-Null Count Dtype   
--- ------ -------------- -----   
0 sepal\_length 150 non-null float64  
1 sepal\_width 150 non-null float64  
2 petal\_length 150 non-null float64  
3 petal\_width 150 non-null float64  
4 species 150 non-null object   
dtypes: float64(4), object(1)  
memory usage: 6.0+ KB

We can see that only one column has categorical data and all the other columns are of the numeric type with non-Null entries.

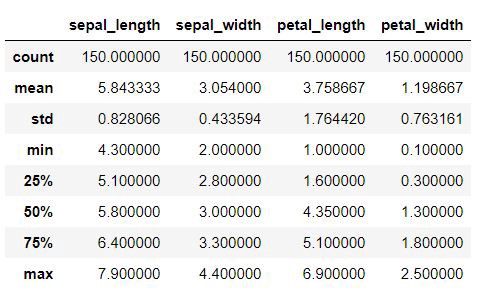
* We can see that only one column has categorical data and all the other columns are of the numeric type with non-Null entries.

**Data Insights:**

* *1 All columns are not having any Null Entries*
* *2 Four columns are numerical type*
* *3 Only Single column categorical type*

**Statistical Insight**

* *iris\_data.describe()*

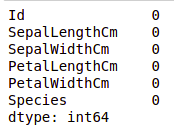


**Data Insights:**

* Mean values
* Standard Deviation ,
* Minimum Values
* Maximum Values

**Checking Missing Values**

* We will check if our data contains any missing values or not. Missing values can occur when no information is provided for one or more items or for a whole unit. We will use the [isnull()](https://www.geeksforgeeks.org/python-pandas-isnull-and-notnull/) method.
* *iris\_data*.isnull().sum()



We can see that no column as any missing value.

**Checking For Duplicate Entries**

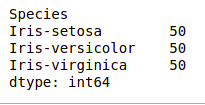
* *iris\_data[iris\_data.duplicated()]*



There are 3 duplicates, therefore we must check whether each species data set is balanced in no's or no

**Checking the balance**

*iris\_data[‘species’].value\_counts()*



Therefore we shouldn’t delete the entries as it might imbalance the data sets and hence will prove to be less useful for valuable insights

**Data Visualization**

**Visualizing the target column**

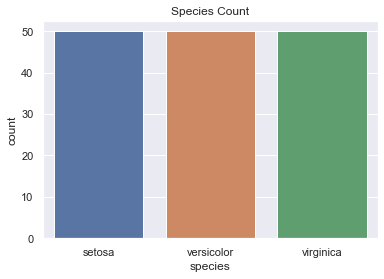
* Our target column will be the Species column because at the end we will need the result according to the species only. **Note:**We will use Matplotlib and Seaborn library for the data visulalization.

# importing packages

import seaborn as sns

import matplotlib.pyplot as plt

*plt.title(‘Species Count’)*  
*sns.countplot(iris\_data[‘species’])*



**Data Insight:**

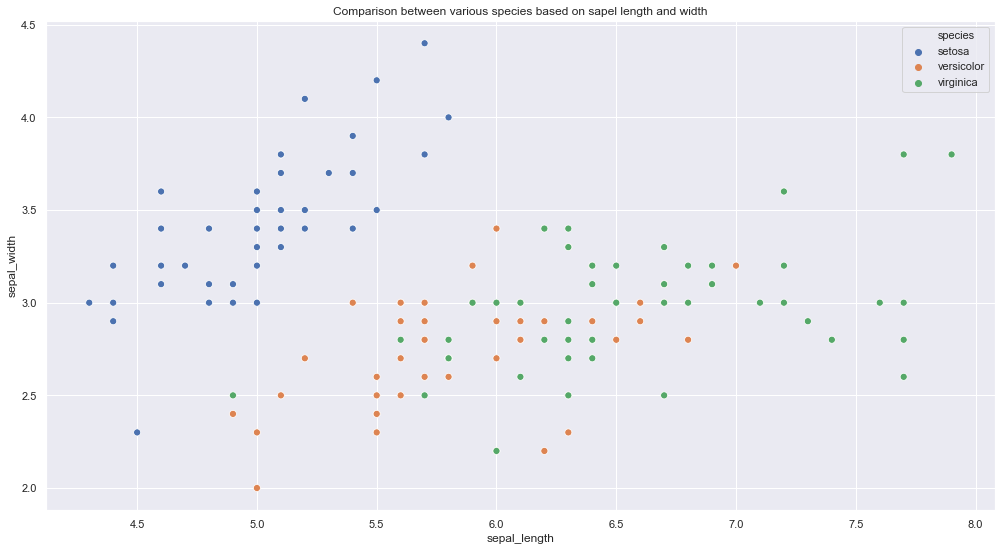
* This further visualizes that species are well balanced
* Each species ( Iris virginica, setosa, versicolor) has 50 as it’s count



**Uni-variate Analysis**

***Comparison between various species based on sepal length and width***

*plt.figure(figsize=(17,9))*  
*plt.title(‘Comparison between various species based on sapel length and width’)*  
*sns.scatterplot(iris\_data[‘sepal\_length’],iris\_data[‘sepal\_width’],hue =iris\_data[‘species’],s=50)*

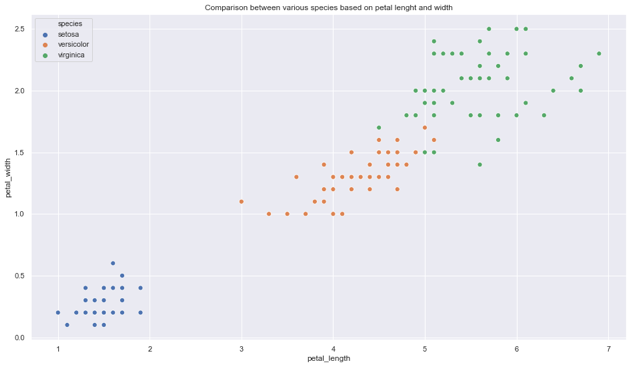


**Data Insights:**

* Iris Setosa species has smaller sepal length but higher width.
* Versicolor lies in almost middle for length as well as width
* Virginica has larger sepal lengths and smaller sepal widths

***Comparison between various species based on petal length and width***

*plt.figure(figsize=(16,9))  
plt.title(‘Comparison between various species based on petal lenght and width’)  
sns.scatterplot(iris\_data[‘petal\_length’], iris\_data[‘petal\_width’], hue = iris\_data[‘species’], s= 50)*

 **Data Insights**

* Setosa species have the smallest petal length as well as petal width
* Versicolor species have average petal length and petal width
* Virginica species have the highest petal length as well as petal width

Let’s plot all the column’s relationships using a pairplot. It can be used for multivariate analysis.

* *sns.pairplot(iris\_data,hue=”species”,height=4)*



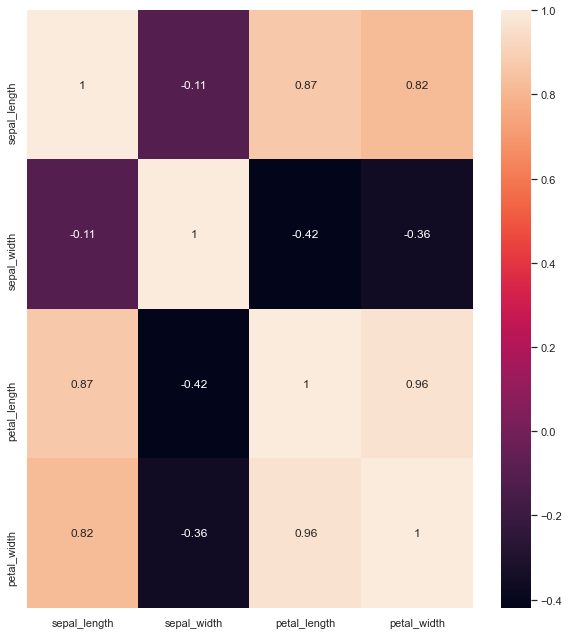
**Data Insights:**

* High co relation between petal length and width columns.
* Setosa has both low petal length and width
* Versicolor has both average petal length and width
* Virginica has both high petal length and width.
* Sepal width for setosa is high and length is low.
* Versicolor have average values for for sepal dimensions.
* Virginica has small width but large sepal length

The **heatmap** is a data visualization technique that is used to analyze the dataset as colors in two dimensions. Basically, it shows a correlation between all numerical variables in the dataset. In simpler terms, we can plot the above-found correlation using the heatmaps.

**Checking Correlation**

* *plt.figure(figsize=(10,11))  
  sns.heatmap(iris\_data.corr(),annot=True)  
  plt.plot()*

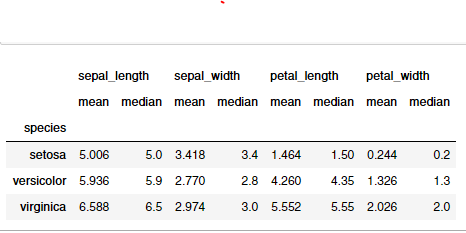


**Data Insights:**

Sepal Length and Sepal Width features are slightly correlated with each other

**Checking Mean & Median Values for each species**

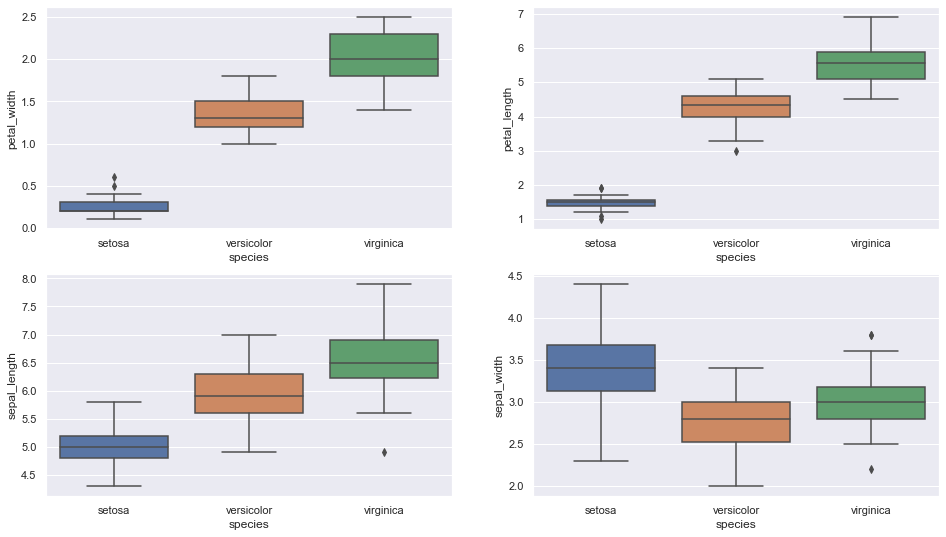
* *iris.groupby(‘species’).agg([‘mean’, ‘median’])*



visualizing the distribution , mean and median using box plots & violin plots

**Box plots to know about distribution**

* boxplot to see how the categorical feature “Species” is distributed with all other four input variables
* *fig, axes = plt.subplots(2, 2, figsize=(16,9))  
  sns.boxplot( y=”petal\_width”, x= “species”, data=iris\_data, orient=’v’ , ax=axes[0, 0])  
  sns.boxplot( y=”petal\_length”, x= “species”, data=iris\_data, orient=’v’ , ax=axes[0, 1])  
  sns.boxplot( y=”sepal\_length”, x= “species”, data=iris\_data, orient=’v’ , ax=axes[1, 0])  
  sns.boxplot( y=”sepal\_width”, x= “species”, data=iris\_data, orient=’v’ , ax=axes[1, 1])  
  plt.show()*

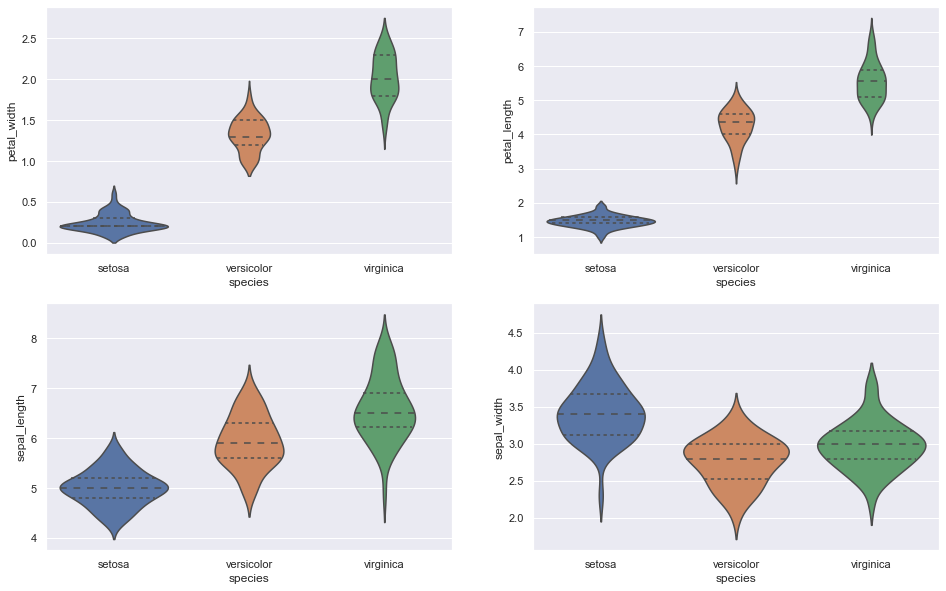
 **Data Insights:**

* Setosa is having smaller feature and less distributed
* Versicolor is distributed in a average manner and average features
* Virginica is highly distributed with large no .of values and features
* Clearly the mean/ median values are being shown by each plots for various features(sepal length & width, petal length & width)

**Violin Plot for checking distribution**

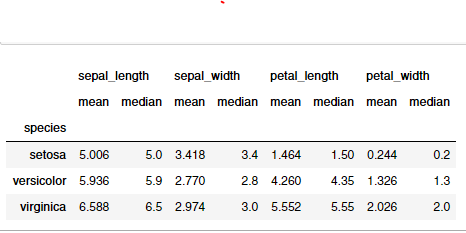
* The violin plot shows density of the length and width in the species. The thinner part denotes that there is less density whereas the fatter part conveys higher density

*fig, axes = plt.subplots(2, 2, figsize=(16,10))*  
*sns.violinplot( y=”petal\_width”, x= “species”, data=iris\_data, orient=’v’ , ax=axes[0, 0],inner=’quartile’)*  
*sns.violinplot( y=”petal\_length”, x= “species”, data=iris\_data, orient=’v’ , ax=axes[0, 1],inner=’quartile’)*  
*sns.violinplot( y=”sepal\_length”, x= “species”, data=iris\_data, orient=’v’ , ax=axes[1, 0],inner=’quartile’)*  
*sns.violinplot( y=”sepal\_width”, x= “species”, data=iris\_data, orient=’v’ , ax=axes[1, 1],inner=’quartile’)*  
*plt.show()*

 **Data Insights:**

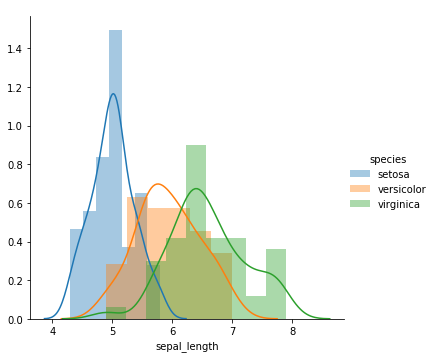
* Setosa is having less distribution and density in case of petal length & width
* Versicolor is distributed in a average manner and average features in case of petal length & width
* Virginica is highly distributed with large no .of values and features in case of sepal length & width
* High density values are depicting the mean/median values, for example: Iris Setosa has highest density at 5.0 cm ( sepal length feature) which is also the median value(5.0) as per the table

**Mean / Median Table for reference**

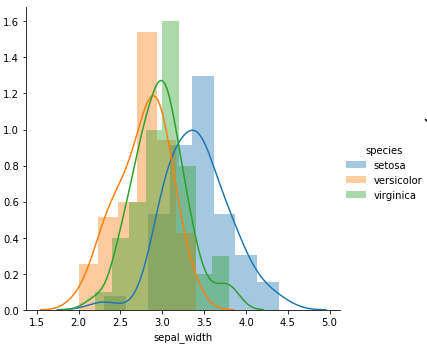


**Plotting the Histogram & Probability Density Function (PDF)**

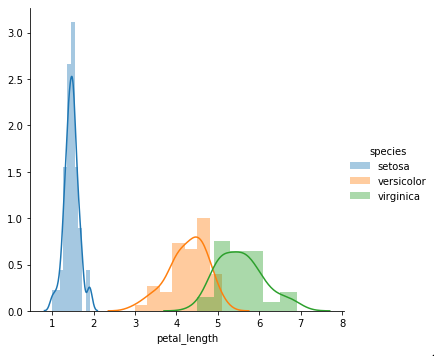
* plotting the probability density function(PDF) with each feature as a variable on X-axis and it’s histogram and corresponding kernel density plot on Y-axis.
* *sns.FacetGrid(iris, hue="species", height=5) \  
  .map(sns.distplot, "sepal\_length") \  
  .add\_legend()*
* *sns.FacetGrid(iris, hue="species", height=5) \  
  .map(sns.distplot, "sepal\_width") \  
  .add\_legend()*
* *sns.FacetGrid(iris, hue="species", height=5) \  
  .map(sns.distplot, "petal\_length") \  
  .add\_legend()*
* *sns.FacetGrid(iris, hue="species", height=5) \  
  .map(sns.distplot, "petal\_width") \  
  .add\_legend()  
  plt.show()*



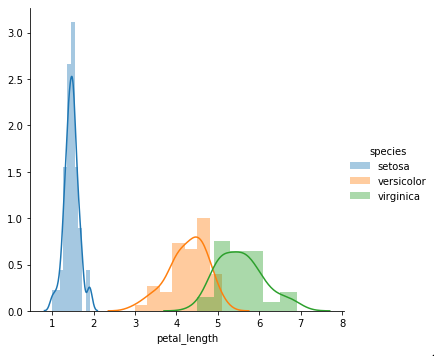
Plot 1 | Classification feature : Sepal Length



Plot 2 | Classification feature : Sepal Width



Plot 3 | Classification feature : Petal Length



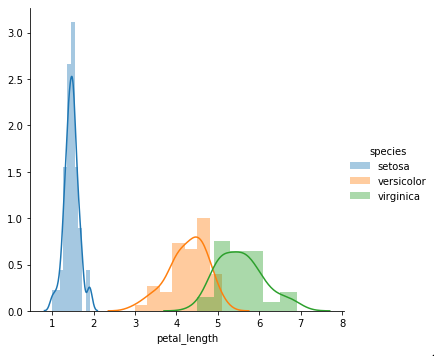
Plot 4 | Classification feature : Petal Width

**Data Insights:**

* Plot 1 shows that there is a significant amount of overlap between the species on sepal length, so it is not an effective Classification feature
* Plot 2 shows that there is even higher overlap between the species on sepal width, so it is not an effective Classification feature
* Plot 3 shows that petal length is a good Classification feature as it clearly separates the species . The overlap is extremely less (between Versicolor and Virginica) , Setosa is well separated from the rest two
* Just like Plot 3, Plot 4 also shows that petal width is a good Classification feature . The overlap is significantly less (between Versicolor and Virginica) , Setosa is well separated from the rest two

**Choosing Plot 3 (Classification feature as Petal Length)to distinguish among the species**

**Choosing Plot 3 (Classification feature as Petal Length)to distinguish among the species**



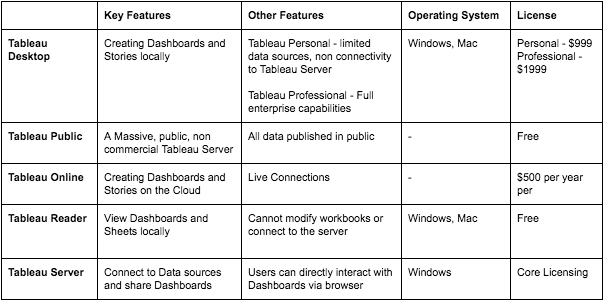
Plot 3 | Classification feature : Petal Length

**Data Insights:**

* The pdf curve of Iris Setosa ends roughly at 2.1
* If petal length < 2.1, then species is Iris Setosa
* The point of intersection between pdf curves of Versicolor and Virginica is roughly at 4.8
* If petal length > 2.1 and petal length < 4.8 then species is Iris Versicolor
* If petal length > 4.8 then species is Iris Virginica

**3.8 DATA VISUALIZATION USING TABLEAU**

* Tableau is a Data Visualisation tool that is widely used for Business Intelligence but is not limited to it. It helps create interactive graphs and charts in the form of dashboards and worksheets to gain business insights. And all of this is made possible with gestures as simple as drag and drop
* Tableau is a powerful and fastest growing data visualization tool used in the Business Intelligence Industry. It helps in simplifying raw data in a very easily understandable format. Tableau helps create the data that can be understood by professionals at any level in an organization. It also allows non-technical users to create customized dashboards.
* Data analysis is very fast with Tableau tool and the visualizations created are in the form of dashboards and worksheets.
* The best features of Tableau software are
  + Data Blending
  + Real time analysis
  + Collaboration of data
* **What Products does Tableau offer?**

****

**Why Tableau?**

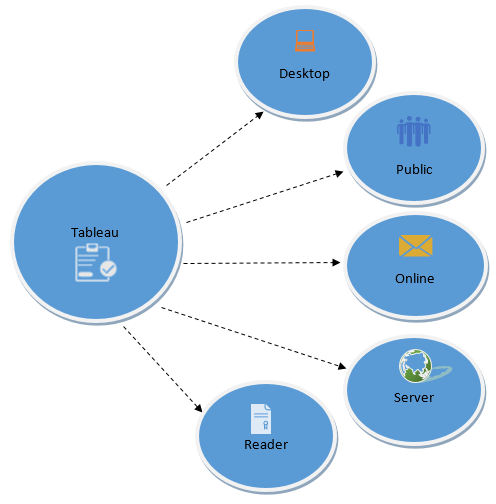
* Tableau is greatly used because data can be analyzed very quickly with it. Also, visualizations are generated as dashboards and worksheets. Tableau allows one to create dashboards that provide actionable insights and drive the business forward. Tableau products always operate in virtualized environments when they are configured with the proper underlying operating system and hardware. Tableau is used by data scientists to explore data with limitless visual analytics.

***Features of Tableau***

* Tableau Dashboard
* Collaboration and Sharing
* Live and In-memory Data
* [Data Sources in Tableau](https://intellipaat.com/blog/tutorial/tableau-tutorial/connect-data-source-tableau/)
* Advanced Visualizations
* Mobile View
* Revision History
* Licensing Views
* Subscribe others
* ETL Refresh and many more make Tableau one of the most famous Data Visualization tools.

**3.8.1. Tableau Product Suite**

* The Tableau Product Suite consists of
  + Tableau Desktop
  + Tableau Public
  + Tableau Online
  + Tableau Server
  + Tableau Reader



**Figure 3.2: Tableau Product Suite**

For a clear understanding, data analytics in Tableau tool can be classified into two section.

* **Developer Tools:** The Tableau tools that are used for development such as the creation of dashboards, charts, report generation, visualization fall into this category. The Tableau products, under this category, are the Tableau Desktop and the Tableau Public.
* **Sharing Tools:** As the name suggests, the purpose of these Tableau products is sharing the visualizations, reports, dashboards that were created using the developer tools. Products that fall into this category are Tableau Online, Server, and Reader.

**Tableau Desktop**

* Tableau Desktop has a rich feature set and allows you to code and customize reports. Right from creating the charts, reports, to blending them all together to form a dashboard, all the necessary work is created in Tableau Desktop.
* For live data analysis, Tableau Desktop provides connectivity to Data Warehouse, as well as other various types of files. The workbooks and the dashboards created here can be either shared locally or publicly.
* Based on the connectivity to the data sources and publishing option, Tableau Desktop is classified into
  + **Tableau Desktop Personal:** The development features are similar to Tableau Desktop. Personal version keeps the workbook private, and the access is limited. The workbooks cannot be published online. Therefore, it should be distributed either Offline or in Tableau Public.
  + **Tableau Desktop Professional:** It is pretty much similar to Tableau Desktop. The difference is that the work created in the Tableau Desktop can be published online or in Tableau Server. Also, in Professional version, there is full access to all sorts of the datatype. It is best suitable for those who wish to publish their work in Tableau Server.

**Tableau Public**

* It is Tableau version specially build for the cost-effective users. By the word “Public,” it means that the workbooks created cannot be saved locally; in turn, it should be saved to the Tableau’s public cloud which can be viewed and accessed by anyone.
* There is no privacy to the files saved to the cloud since anyone can download and access the same. This version is the best for the individuals who want to learn Tableau and for the ones who want to share their data with the general public.

**Tableau Server**

* The software is specifically used to share the workbooks, visualizations that are created in the Tableau Desktop application across the organization. To share dashboards in the Tableau Server, you must first publish your work in the Tableau Desktop. Once the work has been uploaded to the server, it will be accessible only to the licensed users.
* However, It’s not necessary that the licensed users need to have the Tableau Server installed on their machine. They just require the login credentials with which they can check reports via a web browser. The security is high in Tableau server, and it is much suited for quick and effective sharing of data in an organization.
* The admin of the organization will always have full control over the server. The hardware and the software are maintained by the organization.

**Tableau Online**

* As the name suggests, it is an online sharing tool of Tableau. Its functionalities are similar to Tableau Server, but the data is stored on servers hosted in the cloud which are maintained by the Tableau group.
* There is no storage limit on the data that can be published in the Tableau Online. Tableau Online creates a direct link to over 40 data sources that are hosted in the cloud such as the MySQL, Hive, Amazon Aurora, Spark SQL and many more.
* To publish, both Tableau Online and Server require the workbooks created by Tableau Desktop. Data that is streamed from the web applications say Google Analytics, Salesforce.com are also supported by Tableau Server and Tableau Online.

**Tableau Reader**

* Tableau Reader is a free tool which allows you to view the workbooks and visualizations created using Tableau Desktop or Tableau Public. The data can be filtered but editing and modifications are restricted. The security level is zero in Tableau Reader as anyone who gets the workbook can view it using Tableau Reader.
* If you want to share the dashboards that you have created, the receiver should have Tableau Reader to view the document.

**3.8.2. How does Tableau work?**

* Tableau connects and extracts the data stored in various places. It can pull data from any platform imaginable. A simple database such as an excel, pdf, to a complex database like Oracle, a database in the cloud such as Amazon webs services, Microsoft Azure SQL database, Google Cloud SQL and various other data sources can be extracted by Tableau.
* When Tableau is launched, ready data connectors are available which allows you to connect to any database. Depending on the version of Tableau that you have purchased the number of data connectors supported by Tableau will vary.
* The pulled data can be either connected live or extracted to the Tableau’s data engine, Tableau Desktop. This is where the Data analyst, data engineer work with the data that was pulled up and develop visualizations. The created dashboards are shared with the users as a static file. The users who receive the dashboards views the file using Tableau Reader.
* The data from the Tableau Desktop can be published to the Tableau server. This is an enterprise platform where collaboration, distribution, governance, security model, automation features are supported. With the Tableau server, the end users have a better experience in accessing the files from all locations be it a desktop, mobile or email.

**3.8.3. Tableau Uses-** Following are the main uses and applications of Tableau:

* Business Intelligence
* Data Visualization
* Data Collaboration
* Data Blending
* Real-time data analysis
* Query translation into visualization
* To import large size of data
* To create no-code data queries
* To manage large size metadata

**3.8.4. Excel Vs. Tableau**

* Both Excel and Tableau are data analysis tools, but each tool has its unique approach to data exploration. However, the analysis in Tableau is more potent than excel.
* Excel works with rows and columns in spreadsheets whereas Tableau enables in exploring excel data using its drag and drop feature. Tableau formats the data in Graphs, pictures that are easily understandable.

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Excel** | **Tableau** |
| **Purpose** | Spreadsheet application used for manipulating the data. | Perfect visualization tool used for analysis. |
| **Usage** | Most suitable for statistical analysis of structured data. | Most suitable for quick and easy representation of big data which helps in resolving the big data issues. |
| **Performance** | Moderate speed with no option to quicken. | Moderate speed with options to optimize and enhance the progress of an operation. |
| **Security** | The inbuilt security feature is weak when compared to Tableau. The security update needs to be installed on a regular basis. | Extensive options to secure data without scripting. Security features like row level security and permission are inbuilt. |
| **User Interface** | To utilize excel to full potential, macro and visual basic scripting knowledge is required. | The tool can be used without any coding knowledge. |
| **Business need** | Best for preparing on-off reports with small data | Best while working with big data. |
| **Products** | Bundled with MS Office tools | Comes with different versions such as the Tableau server, cloud, and desktop. |
| **Integration** | Excel integrates with around 60 applications | Tableaus integrated with over 250 applications |
| **Real time data exploration** | When you are working in excel, you need have an idea of where your data takes you to get to know the insights | In Tableaus, you are free to explore data without even knowing the answer that you want. With the in-built features like data blending and drill-down, you will be able to determine the variations and data patterns. |
| **Easy Visualizations** | When working in excel, we first manipulate the data that is present and then the visualization such as the different charts, graphs are created manually. To make the visualizations easily understandable, you should understand the features of excel well. | Whereas in Tableau, the data is visualized from the beginning. |

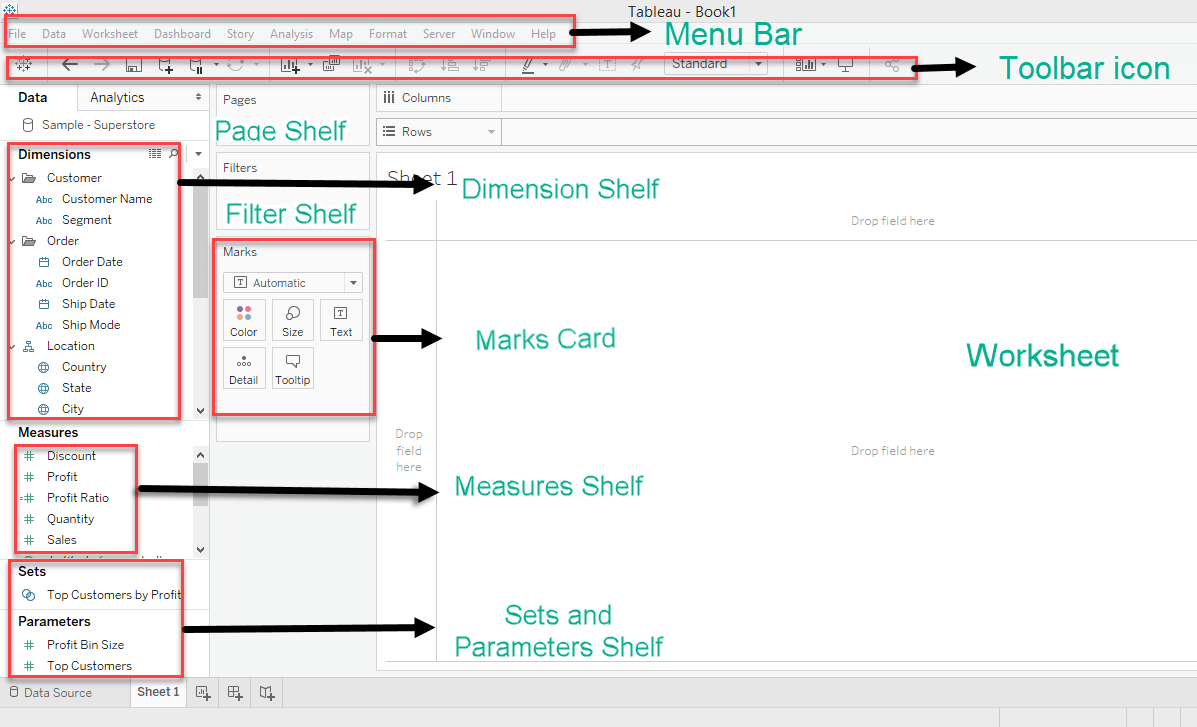
**3.8.5. Creating Visuals in Tableau**

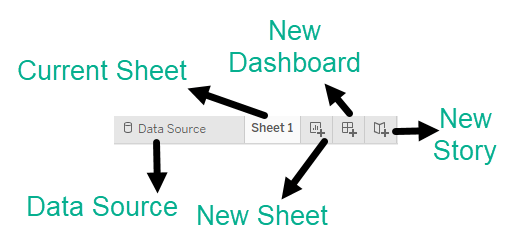
Tableau supports the following data types:

* **Boolean:** True and false can be stored in this data type.
* **Date/Datetime:**  
  This data type can help in leveraging Tableau’s default date hierarchy  
  behavior when applied to valid date or DateTime fields.
* **Number:** These are values that are numeric. Values can be integers or floating-point numbers (numbers with decimals).
* **String:** This is a sequence of characters encased in single or double quotation marks.
* **Geolocation:** These are values that we need to plot maps.

**3.8.6. Understanding different Sections in Tableau**

* Tableau work-page consist of different section.

****



**Figure 3.3: Tableau Work page**

**Source: Local**

* **Menu Bar:** Here you’ll find various commands such as File, Data, and Format.
* **Toolbar Icon**: The toolbar contains a number of buttons that enable you to perform various tasks with a click, such as Save, Undo, and New Worksheet.
* **Dimension Shelf:** This shelf contains all the categorical columns under it. example: categories, segments, gender, name, etc
* **Measure Shelf:** This shelf contains all numerical columns under it like profit, total sales, discount, etc
* **Page Shelf:** This shelf is used for joining pages and create animations. we will come on it later
* **Filter Shelf:** You can choose which data to include and exclude using the Filters shelf, for example, you might want to analyze the profit for each customer segment, but only for certain shipping containers and delivery times. You may make a view like this by putting fields on the Filters tier.
* **Marks Card:** The visualization can be designed using the Marks card. The markings card can be used to change the data components of the visualization, such as color, size, shape, path, label, and tooltip.
* **Worksheet:** In the workbook, the worksheet is where the real visualization may be seen. The worksheet contains information about the visual’s design and functionality.
* **Data Source:**Using Data Source we can add new data, modify, remove data.
* **Current Sheet:**The current sheets are those sheets which we have created and to those, we can give some names.
* **New Sheet:**If we want to create a new worksheet ( blank canvas ) we can do using this tab.
* **New Dashboard:**This button is used to create a dashboard canvas.
* **New Storyboard:**It is used to create a new story

**QUESTION BANK**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Part-A** | | | | | |
| **Q.No** | **Questions** | **Competence** | | **BT Level** | |
|  | Define EDA | Remember | | BTL 1 | |
|  | Why is exploratory data analysis important in data science? | Analysis | | BTL 4 | |
|  | Define Descriptive Statistics | Remember | | BTL 1 | |
|  | List the various methods in EDA | Understand | | BTL 2 | |
|  | List the types of EDA? | Understand | | BTL 2 | |
|  | Enumerate the characteristics of population distribution | Understand | | BTL 2 | |
|  | State the philosophy of EDA? | Understand | | BTL 2 | |
|  | List the steps involved in Data Science Process? | Remember | | BTL 1 | |
|  | List the steps involved in data preprocessing | Understand | | BTL 2 | |
|  | Define feature scaling | Understand | | BTL 2 | |
|  | List some popular plotting libraries of python in data visualization? | Understand | | BTL 2 | |
|  | Define Data Visualization | Remember | | BTL 2 | |
|  | List the data visualization tools | Remember | | BTL 1 | |
|  | Difference between tableau desktop, tableau server and tableau public | Analysis | | BTL 4 | |
|  | How does tableau work? | Understand | | BTL 2 | |
|  | Enumerate the challenges in data visualization? | Analysis | | BTL 4 | |
|  | How to clean the data? | Understand | | BTL 2 | |
|  | Differentiate Excel and Tableau | Analysis | | BTL 4 | |
|  | Why Data Visualization is important**?** | Analysis | | BTL 4 | |
|  | Enumerate the most typical causes of Data Inconsistencies and Errors | Analysis | | BTL 4 | |
| **PART B** | | | | | |
| **Q.No** | **Questions** | | **Competence** | | **BT Level** |
|  | Explain the steps involved in data science process? | | Analysis | | BTL 4 |
|  | Explain various types of EDA? | | Analysis | | BTL 4 |
|  | Explain various univariate and multivariate graphs? | | Analysis | | BTL 4 |
|  | Explain about graphical exploratory data analysis? | | Analysis | | BTL 4 |
|  | Explain about the different stages of preprocessing? | | Analysis | | BTL 4 |
|  | Discuss in detail about preprocessing and data cleaning stages? | | Analysis | | BTL 4 |
|  | Explain the following   1. Univariate Non-graphical 2. Multivariate Non-graphical 3. Univariate graphical 4. Multivariate graphical | | Analysis | | BTL 4 |
| 8. | Discuss about the different data visualization tools | | Analysis | | BTL 4 |
| 9. | Explain about the tableau product suit? | | Analysis | | BTL 4 |
| 10. | How to analyse the data insights for the isis dataset? | | Create | | BTL 5 |