**1. Data Wrangling I**

• *Steps involved in data wrangling:*  
Data wrangling includes data collection, cleaning, transformation, and enrichment. It begins with importing raw data, followed by handling missing values and correcting data types. Then, the data is transformed through filtering, sorting, and aggregating to prepare it for analysis. The final step often involves merging or joining datasets for consistency and usability.

• *Handling missing data using pandas:*  
Pandas offers methods like isnull() and dropna() to detect and remove missing values. Alternatively, fillna() can be used to replace missing values with a specific value or statistic (mean, median, etc.). These methods help maintain dataset integrity and prevent errors during analysis or modeling.

• *Significance of type conversion in data preprocessing:*  
Type conversion ensures that each column has the correct data type for accurate computations and analysis. For example, converting strings to dates or numbers allows for proper sorting and mathematical operations. It also helps in optimizing memory usage and compatibility with machine learning algorithms.

**2. Data Wrangling II**

• *Detecting and handling outliers:*  
Outliers can be detected using statistical methods like IQR (Interquartile Range), Z-score, or visual tools like boxplots. Once identified, they can be removed, capped, or transformed based on context. Handling outliers is important as they can skew results and reduce model performance.

• *Transformations to reduce skewness:*  
Common transformations include log, square root, and Box-Cox transformations. These methods help in normalizing the distribution of data and improving the performance of statistical models. Reducing skewness is essential for models that assume normally distributed input.

• *Importance of normalizing/scaling data:*  
Normalization or scaling ensures that all features contribute equally to a model's performance. Techniques like MinMax scaling and Standardization help prevent features with large values from dominating those with smaller scales. This is crucial for distance-based models like KNN and algorithms like SVM.

**3. Descriptive Statistics**

• *Mean, median, and mode:*  
Mean is the average of values, median is the middle value when data is sorted, and mode is the most frequently occurring value. Mean is sensitive to outliers, whereas median provides a better central tendency for skewed data. Mode is useful for categorical data.

• *Measures of central tendency vs variability:*  
Central tendency measures (mean, median, mode) describe where data is centered, while variability measures (range, variance, standard deviation) show how spread out the data is. Both are important to understand the shape and dispersion of the data. Variability helps assess consistency and predictability.

• *Importance of standard deviation:*  
Standard deviation measures how much data varies from the mean. A low standard deviation indicates that data points are close to the mean, while a high value suggests greater spread. It helps in understanding the risk and volatility in data, which is essential in fields like finance and quality control.

**4. Data Analytics I: Linear Regression**

• *Assumptions of linear regression:*  
Linear regression assumes linearity between features and target, independence of errors, homoscedasticity (constant variance), and normal distribution of residuals. It also assumes no multicollinearity among independent variables. These assumptions ensure the validity and interpretability of the model.

• *Evaluating linear regression accuracy:*  
Model accuracy can be evaluated using metrics like Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R² score. Residual plots also help assess whether assumptions hold. These tools help in understanding how well the model predicts outcomes.

• *R² value indication:*  
The R² value, or coefficient of determination, indicates the proportion of variance in the dependent variable explained by the model. A value close to 1 means the model fits the data well, while a value near 0 indicates poor fit. It helps assess model strength and explanatory power.

**5. Data Analytics II: Logistic Regression**

• *Difference between linear and logistic regression:*  
Linear regression predicts continuous outcomes, while logistic regression predicts binary or categorical outcomes. Logistic regression uses the sigmoid function to convert outputs into probabilities. It’s commonly used for classification tasks like spam detection or medical diagnosis.

• *Confusion matrix representation:*  
A confusion matrix is a table that shows the performance of a classification model using actual vs. predicted values. It includes counts of True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). It helps in calculating various performance metrics.

• *Definitions of TP, FP, TN, FN, precision, recall, and accuracy:*  
TP: correctly predicted positives; FP: incorrectly predicted positives; TN: correctly predicted negatives; FN: incorrectly predicted negatives. Precision = TP / (TP + FP), Recall = TP / (TP + FN), and Accuracy = (TP + TN) / Total. These metrics evaluate the effectiveness of a classification model.

**6. Data Analytics III: Naive Bayes**

• *Basic assumption of Naive Bayes:*  
Naive Bayes assumes that all features are independent of each other given the class label. This "naive" assumption simplifies computation and works well in practice despite its simplicity. The algorithm applies Bayes’ theorem to estimate probabilities.

• *Effective dataset types for Naive Bayes:*  
Naive Bayes performs well with large, high-dimensional datasets, especially when features are conditionally independent. It is particularly effective for text classification, spam filtering, and sentiment analysis. It works best when feature distributions align with the assumed probability model.

• *Handling text classification:*  
Naive Bayes converts text into numerical features using techniques like Bag of Words or TF-IDF. It then calculates the probability of each class based on word frequencies. This simplicity and efficiency make it popular for NLP tasks like email spam detection.

**7. Text Analytics**

• *Steps of text preprocessing:*  
Text preprocessing involves tokenization, converting to lowercase, removing stopwords, punctuation, and special characters, followed by stemming or lemmatization. These steps clean and standardize text for analysis. Preprocessing improves model performance and interpretability.

• *Stemming vs lemmatization:*  
Stemming removes suffixes to reduce words to their root form, often resulting in non-standard words. Lemmatization uses vocabulary and grammar rules to convert words to their dictionary form. Lemmatization is more accurate but computationally expensive compared to stemming.

• *Term Frequency (TF) and Inverse Document Frequency (IDF):*  
TF measures how often a term appears in a document, while IDF reflects how unique the term is across all documents. TF-IDF combines both to highlight important and relevant words in a corpus. It helps improve the quality of feature representation in text analysis.

**8. Data Visualization I**

• *Insights from a histogram:*  
A histogram shows the distribution of a variable by dividing it into bins and displaying frequencies. It reveals patterns such as skewness, modality, and the presence of outliers. This helps understand the spread and central tendency of data.

• *Seaborn’s preference for statistical plots:*  
Seaborn provides high-level, aesthetically pleasing plots with built-in themes and color palettes. It simplifies statistical visualization with functions like distplot, pairplot, and boxplot. Its integration with pandas makes it ideal for quick data exploration.

• *Skewed histogram indication:*  
A skewed histogram suggests that data is not symmetrically distributed. Right (positive) skew means the tail is longer on the right, while left (negative) skew means the tail is on the left. Skewness affects statistical analysis and model assumptions.

**9. Data Visualization II**

• *Use of a boxplot:*  
A boxplot visualizes the distribution of a dataset using quartiles and highlights the median, spread, and outliers. It is useful for comparing multiple variables or groups. It provides a compact summary of data variability and central tendency.

• *Boxplots identifying outliers:*  
Boxplots mark outliers as individual points beyond the whiskers, which typically represent 1.5 times the IQR. This visual cue helps quickly identify unusual values. Detecting outliers is important for data cleaning and analysis.

• *Inference from Titanic survival vs. age/gender:*  
Analysis often shows that females and younger passengers had higher survival rates. Visualizations like bar plots and boxplots can highlight these trends. Such insights support decision-making and hypothesis testing.

**10. Data Visualization III**

• *Using histograms and boxplots for feature distribution:*  
Histograms show the frequency distribution of a single variable, while boxplots summarize spread and identify outliers. Together, they provide a detailed understanding of feature behavior. They are useful for identifying skewness, spread, and anomalies.

• *Significance of comparing distributions across features:*  
Comparing distributions helps detect patterns, correlations, or inconsistencies between variables. It can reveal which features contribute most to variance or predictive power. This comparison aids in feature selection and preprocessing.

**Group B: Data Analytics using JAVA/SCALA**

• *Differences between MapReduce and Spark:*  
MapReduce is disk-based and processes data in batches, leading to slower execution. Spark is memory-resident and supports in-memory computation, making it significantly faster. Spark also supports more complex operations like streaming and machine learning.

• *Impala usage for querying data:*  
Impala is a distributed SQL engine for querying large datasets stored in Hadoop. It provides low-latency and real-time querying capabilities using standard SQL syntax. It integrates well with Hive and supports interactive analytics.

• *Scala’s advantages with Spark:*  
Scala is Spark’s native language and provides concise syntax and strong functional programming support. It enables better performance and tighter integration with Spark’s core APIs. Scala’s REPL also helps in rapid prototyping and testing.