Statistic: A numerical value calculated from a sample of data. Statistics are used to estimate population parameters. Population parameters are usually unknown and are estimated using statistics.
`df['col_name'].value_counts()`: Counts the occurrences of each unique value in a column (categorical data).
`df['col_name'].count()`: Counts the number of **non-missing** values in a DataFrame. Used with `.groupby()`
`df.groupby("g_col")["value_col"]`: Groups the data by "**`g_col`**" and indexes "**`value_col`**" within each group.
`df.iloc[start_row:end_row, start_col:end_col]`: Selects data by integer-based row and column positions.
`df.loc[row_label, column_label]`: Selects data by row and column labels.
`np.random.choice(list_of_options, size=num_of_samples, replace=True/False): int`
`stats.norm(loc=mean, scale=std_dev).rvs(size=num_of_samples): np.array`
`stats.binom(n=trials, p=prob).rvs(size=num_of_samples): np.array`
`stats.gamma(a=shape, loc=location, scale=scale).rvs(size=num_of_samples): np.array`
Numerical Data: Continuous is a real number (e.g., height, weight). Discrete is a integer (e.g., # of students, dice rolls).
Categorical Data: Nominal has no order (e.g., colors, countries). Ordinal has order (e.g., ratings, education levels).
Binary: Categorical data with only two categories (e.g., True/False, Yes/No).
Histogram: Shows frequency distribution, highlights shape, center, spread, outliers, modality; depends on bin choice, poor for group comparisons. **Box Plot:** Shows mean, median, quartiles, outliers, center, spread, skewness; good for group comparisons, no frequency, shape, or modality detail. **KDE (Kernel Density Estimate) or Violin Plot:** Estimates the probability density function of a continuous variable, highlighting smooth distribution, peaks, and spread; sensitive to bandwidth choice, not great for group comparisons. **Bar Plot:** Displays categorical data with rectangular bars, showing frequencies or values; simple and clear for comparisons, but less effective for continuous data and distributions.
Skewness: Asymmetry in data distribution (left skew (negatively): tail on the left, **mean < median < mode**; right skew (positively): tail on the right, **mode < median < mean**). **Multimodality**: Having multiple peaks in the data distribution.
Bootstrapping: Estimate the sampling distribution of a statistic and create confidence intervals. A resampling technique that estimates the sampling distribution of a statistic. Assume the sample is representative of the population. **P-value:** The probability of getting a test statistic, as or more extreme than, the observed test statistic, assuming the null hypothesis was true. **Bootstrapped 95% CI** constructed from bootstrapped sampling distribution are theoretically going to "work" and construct an interval that does actually capture the actual true parameter value 95% of the time. If the observed test statistic falls **outside** the confidence interval, you can **reject** the null hypothesis (this is superior). Length of **CI** depends on **confidence level** and **sample size**, increasing samples is preferable since it doesn't reduce the confidence level.
Type I Error: Rejecting the null hypothesis when it is actually true (False Positive). **Type II Error**: Failing to reject the null hypothesis when it is actually false (False Negative).
One-Sample Testing: Compares a sample statistic to a hypothesized population parameter (e.g., testing if a coin is fair), $H_0: \mu = 0$ with test stat $\bar{x}$. **Two-Sample Testing:** Compares two independent samples drawn from different populations to assess whether there is a statistically significant difference between their population means/proportions (e.g., before-and-after tests), $H_0: \mu_1 = \mu_2$ with test stat $\overline{x_1} - \overline{x_2}$. Two-sample testing uses permutation tests (shuffling group labels to simulate data under the null hypothesis; non-parametric, robust, and widely applicable but computationally intensive), "double" bootstrapping (bootstrap each sample to form confidence intervals), and indicator variables in regression (encode group membership as binary variables). **Paired-Sample Testing**: Compares two observations from the same individuals to evaluate differences between the paired observations, $H_0: \mu_1 = \mu_2$ with test stat $\overline{x_1 - x_2}$.

Simple Linear Regression: $Y_i = \beta_0 + \beta_1 x_i + \epsilon_i$ where $\epsilon_i \sim N(0, \sigma^2)$. Assume that ϵ_i is normally distributed and homoscedastic.

Multiple Linear Regression:

$$Y_i = \beta_0 + \beta_1 x_{1i} + \beta_1 x_{2i} + \beta_1 x_{3i} + \cdots + \epsilon_i \qquad \text{with an } \epsilon_i \sim N(0, \sigma) \text{ assumption}$$

Note that Correlation IS NOT Causation. It just measures the Empirical Strength of a Linear Relationship.

$$E[y_i] = \beta_0 + \beta_{\text{low}} \underbrace{1_{[\text{low}]}(\text{variable}_i)}_{1 \text{ if variable}_i \text{ is "low"; else, } 0} + \beta_{\text{medium}} \underbrace{1_{[\text{medium}]}(\text{variable}_i)}_{1 \text{ if variable}_i \text{ is "medium"; else, } 0}$$

Logistic Regression: Model the relationship between **predictor variables** and a **binary outcome** by predicting the probability of an event using the logit function to convert predictor variables into log-odds and then into probabilities. A one-unit increase in the **predictor variable** multiplies the odds by e^{β}, holding all other variables constant.

R-squared: Measures the proportion of variance in the outcome variable explained by the model.

Adjusted R-squared: Accounts for the number of predictor variables in the model. **Fitted Model Equation:** $\widehat{y}_i = \widehat{\beta}_0 + \widehat{\beta}_1 x_i$.

Residuals (error terms): Differences between observed values and predicted values in your regression model, $e_i = \widehat{\epsilon}_i = y_i - \widehat{y}_i$. Residuals are actually available, while the errors are just a theoretical concept.

Classification Trees: Predict categorical outcomes. **Regression Trees:** Predict continuous outcomes.

Accuracy: Overall correct predictions out of all predictions, $\frac{TP+TN}{TP+TN+FP+FN}$ (e.g., predicting disease presence correctly 90% of the time). High accuracy means the model correctly predicts most instances, while low accuracy indicates many misclassifications overall.

Sensitivity: True positives out of all actual positives, $\frac{TP}{TP+FN}$ (e.g., detecting 85% of patients with a disease). High sensitivity means the model correctly identifies most positive instances, while low sensitivity indicates many false negatives.

Specificity: True negatives out of all actual negatives, $\frac{TN}{TN+FP}$ (e.g., correctly identifying 95% of healthy patients). High specificity means the model correctly identifies most negative instances, while low specificity suggests it falsely classifies negative instances as positive. **False Positive Rate** is 1 minus specificity, or $\frac{FP}{TN+FP}$.

Precision: True positives out of all predicted positives, $\frac{TP}{TP+FP}$ (e.g., 80% of predicted diseased patients truly have the disease). High precision means the model's positive predictions are mostly correct, while low precision suggests many false positives.

Feature Importances: Measure the relative importance of predictor variables in making predictions.

T**hreshold**: Determines the probability cutoff for classifying predictions as positive or negative in binary classification. Adjusting the threshold can balance metrics like sensitivity and specificity, where lower thresholds increase sensitivity and higher thresholds increase specificity.

Overfitting: Occurs when the decision tree is too complex and learns the training data too well, leading to poor generalization to new data. The more complex a model becomes, the greater its flexibility to identify idiosyncratic spurious (false random chance) associations in the data, which may accidentally (in a Type 1 error manner) lead to the model being overfit to relationships arising solely from random sampling variability.

Pruning: Simplifies by removing unnecessary branches. **Setting Limits:** Restricts tree depth, node count, or minimum samples per leaf. **Tuning Parameters:** **`min_samples_split`** (min samples needed to split an internal node), **`min_samples_leaf`** (min samples required at a leaf node), **`max_depth`** (max depth of the tree), **`max_nodes`** (max total number of nodes or leaf nodes in the tree).

Train-Test Splitting: Data is split into training and testing sets to evaluate generalization. Model performance on the test set identifies potential overfitting.

Advantages of Classification Trees: Transparent decision-making and visualizable structure; Captures complex, non-linear relationships; Accommodates numerical and categorical data.

Disadvantages of Classification Trees: Prone to overfitting if not regularized; Sensitive to small data changes, leading to unstable structures.

Applications: Medical Diagnosis (Predict diseases from symptoms and history), Credit Scoring (Evaluate financial creditworthiness), Customer Segmentation (Group customers based on purchasing behavior), Image Recognition (Classify images by visual features).

Decision Boundary Graph: Visualises the **outcomes** in a classification tree **with partitions on a square/graph**, based on the **predictor variable range/rule**. **`"Predictor A"`** on the x-axis and **`"Predictor B"`** on the y-axis.

Partial Dependence Plots (PDP): Visualises the relationship between a **predictor variable** and the **outcome** in classification tree by averaging out other predictors, helping to identify linear or non-linear relationships, feature importance, and interactions. They are useful for understanding complex models but assume predictor independence, which can be misleading with correlated predictors.