No	Category	Function Name	IMP Arguments	Function Description	Example
1	Scatter Plot	*scatterplot()*	x: Data for the x-axis y: Data for the y-axis hue: Grouping variable that maps to color style: Grouping variable that maps to markers.	The scatterplot() function is used to create scatter plots with additional features like grouping by color and marker style.	`sns.scatterplot(x='sepal_length', y='sepal_width', hue='species', data=df)`
2	Line Plot	*lineplot()*	x: Data for the x-axis y: Data for the y-axis hue: Grouping variable that maps to color style: Grouping variable that maps to line style.	The lineplot() function is used to create line plots for time series or other continuous data.	`sns.lineplot(x='year', y='pop', hue='continent', data=df)`
3	Bar Plot	*barplot()*	x: Categorical variable for the x-axis y: Numeric variable for the y-axis hue: Grouping variable that maps to color.	The barplot() function is used to create bar plots for categorical data.	`sns.barplot(x='day', y='total_bill', hue='sex', data=df)`
4	Histogram	*histplot()*	x: Numeric variable to plot bins: Number of bins for the histogram hue: Grouping variable that maps to color.	The histplot() function is used to create histograms with options for grouping and styling.	`sns.histplot(x='age', hue='gender', bins=30, data=df)`
5	KDE Plot	*kdeplot()*	x: Data for the x-axis y: Optional data for the y-axis (2D KDE) fill: Whether to fill the area under the curve hue: Grouping variable that maps to color.	The kdeplot() function is used to plot kernel density estimates for univariate or bivariate data.	`sns.kdeplot(x='age', hue='gender', fill=True, data=df)`
6	Pair Plot	*pairplot()*	data: DataFrame to use hue: Grouping variable that maps to color diag_kind: Type of plot for the diagonal ('auto', 'hist', 'kde').	The pairplot() function is used to create pairwise plots for all numerical columns in a DataFrame.	`sns.pairplot(df, hue='species', diag_kind='kde')`
7	Heatmap	*heatmap()*	data: 2D dataset (e.g., DataFrame) annot: If True, write the data value in each cell cmap: Colormap for the heatmap.	The heatmap() function is used to create a heatmap to visualize 2D data with color-coded values.	`sns.heatmap(data=corr_matrix, annot=True, cmap='coolwarm')`
8	Box Plot	*boxplot()*	x: Categorical variable for the x-axis y: Numeric variable for the y-axis hue: Grouping variable that maps to color.	The boxplot() function is used to create box-and-whisker plots for visualizing distributions.	`sns.boxplot(x='day', y='total_bill', hue='sex', data=df)`
9	Violin Plot	*violinplot()*	x: Categorical variable for the x-axis y: Numeric variable for the y-axis hue: Grouping variable that maps to color split: If True, split the violin into halves for each hue.	The violinplot() function is used to visualize the distribution of numeric data with kernel density estimation.	`sns.violinplot(x='day', y='total_bill', hue='sex', split=True, data=df)`
10	Swarm Plot	*swarmplot()*	x: Categorical variable for the x-axis y: Numeric variable for the y-axis hue: Grouping variable that maps to color.	The swarmplot() function is used to create a scatter plot with points adjusted to avoid overlap.	`sns.swarmplot(x='day', y='total_bill', hue='sex', data=df)`
11	Count Plot	*countplot()*	x: Categorical variable to count hue: Grouping variable that maps to color.	The countplot() function is used to visualize the count of observations in each category.	`sns.countplot(x='day', hue='sex', data=df)`

12	Facet Grid	*FacetGrid.map()*	row: Variable to create row facets col: Variable to create column facets hue: Grouping variable for color func: Function to map onto the grid.	The FacetGrid.map() function is used to create a grid of subplots for visualizing conditional relationships.	`g = sns.FacetGrid(df, row='species', col='island'); g.map(sns.scatterplot, 'bill_length_mm', 'bill_depth_mm')`