Basic Plot Creation

Commands	Description	Example
`sns.lineplot()`	Draws a line plot (good for time-series, regression). Handles multiple groups automatically.	`sns.lineplot(x="time", y="value", hue="category", data=df)`
`sns.scatterplot()`	Scatter plot. Supports size, hue, style and alpha.	`sns.scatterplot(x="x", y="y", size="size_col", hue="group", data=df)`
`sns.relplot()`	*Relational* plot that creates a figure with multiple axes (grid). `kind='scatter'` or `'line'`.	`sns.relplot(x="x", y="y", hue="group", kind="scatter", data=df)`
`sns.histplot()`	Histogram (1-D) or KDE. Supports bins, multiple datasets, rug plots.	`sns.histplot(data=df["col"], kde=True, bins=30)`
`sns.kdeplot()`	Kernel density estimate plot (1-D or 2-D).	`sns.kdeplot(x="x", y="y", data=df, fill=True)`
`sns.ecdfplot()`	Empirical CDF plot.	`sns.ecdfplot(data=df["col"])`
`sns.boxplot()`	Box-and-whisker plot. Handles multiple groups, whiskers can be set to 1.5*IQR or percentiles.	`sns.boxplot(x="group", y="value", data=df)`
`sns.violinplot()`	Violin plot (kernel density + box).	`sns.violinplot(x="group", y="value", data=df)`
`sns.stripplot()`	Jittered scatter plot (useful for categorical data).	`sns.stripplot(x="group", y="value", jitter=True, data=df)`
`sns.swarmplot()`	Non-overlapping scatter plot for categorical data.	`sns.swarmplot(x="group", y="value", data=df)`
`sns.countplot()`	Bar plot of counts for categorical variables.	`sns.countplot(x="category", data=df)`
`sns.barplot()`	Estimate of mean (or other estimator) with CI bars.	`sns.barplot(x="group", y="value", data=df, ci=95)`
`sns.pointplot()`	Similar to barplot but with a point and error bars.	`sns.pointplot(x="group", y="value", data=df)`
`sns.heatmap()`	2-D heat map of a matrix or DataFrame.	`sns.heatmap(df.corr(), annot=True, cmap="coolwarm")`
`sns.pairplot()`	Grid of pairwise relationships with scatter / histogram / KDE on the diagonal.	`sns.pairplot(df, hue="class")`
`sns.lmplot()`	Linear model fit with optional seaborn regression diagnostics.	`sns.lmplot(x="x", y="y", hue="group", data=df)`
`sns.jointplot()`	Joint distribution plot with marginal histograms / KDE.	`sns.jointplot(x="x", y="y", kind="hex")`
`sns.distplot()` (deprecated)	1-D histogram + KDE. Use `histplot` and/or `kdeplot`.	–
`sns.rugplot()`	Small tick marks along an axis to show data density.	`sns.rugplot(data=df["col"], height=0.05)`
`sns.ecdfplot()`	Empirical cumulative distribution function plot.	–

Multi-Plot - Facet Functions

Command	Description	Example
`sns.relplot()`	*Relational* grid of plots (scatter / line). Handles `col`, `row`, `hue`.	`sns.relplot(x="x", y="y", col="group", hue="species", data=df)`
`sns.catplot()`	General categorical grid (bar, box, violin, swarm, strip).	`sns.catplot(x="group", y="value", col="category", kind="box", data=df)`
`sns.pairplot()`	Grid of pairwise relationships.	–
`sns.FacetGrid()`	Lower-level API to build custom grids; can map any function.	`g = sns.FacetGrid(df, col="group"); g.map(sns.scatterplot, "x", "y")`

Statistical Functions

Command	Description	Example
`sns.regplot()`	Fit a linear regression line with optional lowess smoothing.	`sns.regplot(x="x", y="y", data=df)`
`sns.rugplot()`	Small tick marks along an axis.	–
`sns.residplot()`	Plot residuals of a regression fit.	`sns.residplot(x="x", y="y", data=df)`
`sns.lmplot()`	Wrapper around `regplot` for faceted plots.	–
`sns.rugplot()`	–	–

Heat Maps - Matrix Plots

Command	Description	Example
`sns.heatmap()`	Heat map of a matrix or DataFrame, supports annotations and clustering.	`sns.clustermap(df.corr(), cmap="coolwarm")`
`sns.clustermap()`	Hierarchical clustering + heatmap.	–

Styling - Themes

Command	Description	Example
`sns.set()`	Set default context, style, palette, font scale.	`sns.set(context="talk", style="whitegrid", palette="deep")`
`sns.set_style()`	Change the aesthetics of the plot (e.g., "white", "darkgrid").	`sns.set_style("ticks")`
`sns.set_context()`	Adjust context (paper, notebook, talk, poster).	`sns.set_context("poster")`
`sns.set_palette()`	Set the color palette (list, seaborn name).	`sns.set_palette("Set2")`
`sns.color_palette()`	Return a palette object.	`palette = sns.color_palette("viridis", n_colors=5)`
`sns.cubehelix_palette()`	Generate a cube-helix palette.	–
`sns.light_palette()` / `dark_palette()`	Generate light/dark palettes from a base color.	–
`sns.despine()`	Remove the top and right spines of a plot.	`sns.despine(offset=10, trim=True)`
`sns.move_legend()`	Move the legend to a specified location.	–