

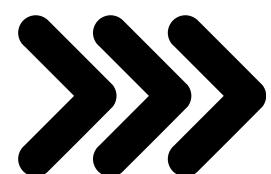


Tajamul Khan

Seaborn Cheat Sheet



@Tajamulkhan



Distribution Plots

- **sns.histplot(data, bins=30):** Visualize the distribution of data with histogram.
- **sns.kdeplot(data, shade=True):** Plot kernel density estimate to analyze distribution.
- **sns.rugplot(data):** marginal tick marks to display data points along the axis.
- **sns.distplot(data):** Combine histogram and KDE (deprecated but still useful).
- **sns.ecdfplot(data):** Visualize cumulative distribution of data.
- **sns.violinplot(x=data):** Combine KDE and boxplot features in one visualization.
- **sns.boxplot(data):** Summarize data distribution and detect outliers.
- **sns.jointplot(x, y, kind='hex'):** Show bivariate distribution using hexagonal bins.



@Tajamulkhan



Categorical Plots

- **sns.barplot(x, y, data)**: Show mean values with confidence intervals for categories.
- **sns.countplot(x, data)**: Count and visualize frequency of categorical data.
- **sns.stripplot(x, y, data, jitter=True)**: Scatterplot for categorical variables.
- **sns.swarmplot(x, y, data)**: Scatterplot avoiding overlap for better visibility.
- **sns.boxplot(x, y, data)**: Boxplots for categories with statistical insights.
- **sns.violinplot(x, y, data)**: Visualize data distribution across categories.
- **sns.pointplot(x, y, data)**: Display mean values with confidence intervals.
- **sns.catplot(x, y, data, kind='strip')**: Flexible plotting for multiple categories.



@Tajamulkhan



Relational Plots

- **sns.scatterplot(x, y, data)**: Visualize relationships between two continuous variables.
- **sns.lineplot(x, y, data)**: Show trends or time-series relationships.
- **sns.relplot(x, y, data, kind='scatter')**: Flexible relational plot (scatter/line).
- **sns.regplot(x, y, data)**: Add a regression line to the scatterplot.
- **sns.lmplot(x, y, data)**: Linear regression with options for subplots.
- **sns.residplot(x, y, data)**: Visualize residuals of regression models.
- **sns.hexbin(x, y, gridsize=30)**: Density plot for large relational datasets.
- **sns.scatterplot(x, y, hue, data)**: Differentiate relationships using color.



@Tajamulkhan



Matrix & Heatmaps

- **sns.heatmap(data, annot=True):**
Display correlation or pivot tables with annotations.
- **sns.clustermap(data):** Visualize hierarchical clustering of data matrices.
- **sns.corrplot(corr_matrix):** Custom heatmaps for correlation matrices.
- **sns.pairplot(data):** Show pairwise relationships in a dataset.
- **sns.heatmap(data, cmap='coolwarm'):**
Use specific colormaps for emphasis.
- **sns.heatmap(data, linewidths=0.5):**
Add spacing lines for better readability.
- **sns.heatmap(data, vmin=0, vmax=1):**
Normalize heatmaps for better clarity.
- **sns.pairplot(data, hue='column'):**
Differentiate data with colors in pair plots.



@Tajamulkhan



Statistical Plots

- **sns.regplot(x, y, data)**: Plot regression line with scatterplot.
- **sns.lmplot(x, y, hue, data)**: Multivariable linear regression.
- **sns.kdeplot(x, y, cmap='Blues', fill=True)**: Bivariate KDE for density visualization.
- **sns.rugplot(a, axis='x', height=0.05)**: Marginal ticks for quick insights.
- **sns.jointplot(x, y, data, kind='kde')**: Overlay bivariate KDE on joint distribution.
- **sns.jointplot(x, y, data, kind='reg')**: Combine regression and joint plots.
- **sns.boxenplot(x, y, data)**: Enhanced boxplot for large datasets.
- **sns.ecdfplot(data, complementary = True)**: Complementary cumulative distribution function.



@Tajamulkhan



Subplots & Styling

- **`sns.set_style('whitegrid')`**: Apply grid-based styling for readability.
- **`sns.set_palette('pastel')`**: Change color palettes for consistency.
- **`sns.despine()`**: Remove chart spines for cleaner visuals.
- **`sns.set_context('talk')`**: Adjust figure scaling for presentations.
- **`sns.catplot(x, y, data, col='category')`**: Create facet grids for subcategories.
- **`sns.relplot(x, y, col='category', data)`**: Subplot relational data by categories.
- **`sns.set_theme(style='dark')`**: Apply Seaborn themes to enhance readability.
- **`sns.color_palette('deep', as_cmap=True)`**: Set and reuse color palettes.



@Tajamulkhann



Exporting Visuals

- **plt.savefig('plot.png')**: Save Seaborn plots in PNG format.
- **plt.savefig('plot.pdf', dpi=300)**: Export plots in high-resolution PDF.
- **plt.savefig('plot.svg', transparent=True)**: Scalable vector graphics for publication.
- **plt.savefig('plot.jpeg', quality=95)**: Save JPEG with high quality.
- **plt.savefig('plot.png', bbox_inches='tight')**: Minimize whitespace in saved plots.
- **sns.despine()**: Adjust plot spines before saving.
- **sns.set_context('notebook')**: Ensure optimal size for notebook visuals before export.
- **plt.show()**: Display the final plot for validation before saving.



@Tajamulkhan



Found Helpful?

Repost



Follow for more!