**Seaborn Distribution Plots**

1. **Histogram with KDE:** Generate a Pandas Series of 100 random data points from a normal distribution (np.random.normal(0,1,100)). Create a **histogram** of this data using Seaborn, including a **Kernel Density Estimate (KDE)** overlay.
2. **KDE Plot:** Using the same random data from Question 1, create a standalone **KDE plot** (sns.kdeplot()) and fill the area underneath the curve.
3. **Rug Plot:** Create a Series of 20 random data points. Use a **rug plot** (sns.rugplot()) to show the location of each individual data point along the x-axis.
4. **DistPlot (Deprecated replacement):** Explain in one sentence which Seaborn function is now recommended to replace the older, more general distplot().

**Seaborn Relational Plots**

1. **Scatter Plot (scatterplot):** Use the sample Pandas DataFrame df = sns.load\_dataset('tips'). Create a **scatter plot** of the 'total\_bill' versus the 'tip'.
2. **Hue for Categorical Data:** Enhance the scatter plot from Question 5 by setting the color of the points (hue) based on the categorical column 'smoker'.
3. **Relational Line Plot:** Create a DataFrame suitable for a time-series (e.g., x= days 1 to 10, y= random values). Use **sns.relplot** to generate a line plot, demonstrating how it can use relational data.
4. **Size of Points:** In a scatter plot of 'total\_bill' vs. 'tip', map the **size** of the points to the 'size' column (party size) using the size parameter.

**Seaborn Categorical Plots**

1. **Box Plot:** Using the tips DataFrame, create a **box plot** showing the distribution of 'total\_bill' across the different categories in the 'day' column.
2. **Violin Plot:** Create a **violin plot** showing the distribution of 'tip' based on the 'time' of day (Lunch/Dinner).
3. **Swarm Plot:** Overlay a **swarm plot** on the box plot from Question 9 to visualize the raw data points alongside the summary statistics.
4. **Count Plot:** Using the tips DataFrame, create a **count plot** showing the total number of observations for each category in the 'day' column.
5. **Bar Plot (Mean Estimation):** Create a **bar plot** that shows the **mean** 'tip' for each 'day'. (Note: Seaborn bar plots calculate a mean by default).

**Figure Structure and Customization**

1. **Subplots with FacetGrid (or relplot):** Use the tips DataFrame to create two separate scatter plots (side-by-side) showing 'total\_bill' vs. 'tip', separated by the 'sex' column, using **sns.relplot** and the col parameter.
2. **Setting Style:** Use a Seaborn function to set the overall plot style to the aesthetically dark background style **darkgrid**.
3. **Removing Spines:** After creating any plot, use the Matplotlib methods often returned or accessed via Seaborn to remove the **top and right axes lines (spines)**.

**Matrix and Regression Plots**

1. **Heatmap:** Create a 4×4 NumPy array of random integers. Use sns.heatmap() to visualize this array, adding numerical annotations to each cell.
2. **Pair Plot:** Load the iris dataset. Use **sns.pairplot()** to create a matrix of scatter plots showing the relationship between all numerical variables, coloring the points by the 'species' column.
3. **Joint Plot (Scatter and Histograms):** Create a **joint plot** of 'total\_bill' vs. 'tip' from the tips DataFrame. Use the default scatter plot with histograms on the margins.
4. **Regression Plot:** Create a **regression plot** (sns.regplot()) of 'total\_bill' vs. 'tip', showing the linear regression line and the 95% confidence interval.