Assignment

Que 1: Name any five plots that we can plot using the Seaborn library. Also, state the uses of each plot.

Ans:Seaborn is a statistical data visualization library based on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. Here are five types of plots that can be created using Seaborn, along with their common use cases:

Bar Plot:

- Use: Bar plots are useful for displaying the distribution of a categorical variable and comparing the values across different categories.
- Seaborn Function: `seaborn.barplot()`

Histogram:

- Use: Histograms are used to visualize the distribution of a univariate dataset and understand the frequency or density of values in different bins.
- Seaborn Function: `seaborn.histplot()`

Scatter Plot:

- Use: Scatter plots help in visualizing the relationship between two continuous variables, showing how one variable changes with respect to another.
- Seaborn Function: `seaborn.scatterplot()`

Heatmap:

- Use: Heatmaps are effective for displaying the correlation matrix of variables, providing insights into the relationships between multiple variables in a dataset.
- Seaborn Function: `seaborn.heatmap()`

Pair Plot:

- Use: Pair plots are used for exploring pairwise relationships in a dataset, especially when you have multiple variables, by creating scatter plots and histograms for each pair of variables.
- Seaborn Function: `seaborn.pairplot()`

These are just a few examples, and Seaborn provides many other plot types and customization options for creating informative visualizations tailored to different types of data and analysis needs.

Que 2: Load the "fmri" dataset using the load_dataset function of seaborn. Plot a line plot using x =
"timepoint" and y = "signal" for different events and regions.

Ans:can use Seaborn's `load_dataset` function to load the "fmri" dataset and then create a line plot using the "timepoint" as the x-axis and "signal" as the y-axis for different events and regions. Here's the code to achieve this:

python

Copy code

```
import          as
```

```
import                    as

                             "fmri"

               "timepoint"    "signal"      "event"        "region"


          'Line Plot of "timepoint" vs "signal" for different events and regions'
           'Timepoint'
           'Signal'
```

In this code:

- `sns.load_dataset("fmri")` loads the "fmri" dataset.
- `sns.lineplot(x="timepoint", y="signal", hue="event", style="region", data=fmri_data)` creates a line plot with "timepoint" on the x-axis, "signal" on the y-axis, different colors for each "event," and different line styles for each "region."
- `plt.title()`, `plt.xlabel()`, and `plt.ylabel()` are used to add a title and labels to the plot.
- `plt.show()` displays the plot.

Adjust the plot parameters and styling according to your preferences and analysis requirements.

Note: timepoint, signal, event, and region are columns in the fmri dataset.
Que 3: Load the "titanic" dataset using the load_dataset function of seaborn. Plot two box plots using x =
'pclass', y = 'age' and y = 'fare'.
Ans:can use Seaborn's `load_dataset` function to load the "titanic" dataset and then create two box plots: one for the relationship between 'pclass' and 'age', and the other for 'pclass' and 'fare'. Here's the code:
python

Copy code

```
import         as
import                    as
```

```
                    "titanic"


          12  6


  1  2  1
    'pclass'    'age'
'Box Plot of Age by Pclass'
 'Pclass'
 'Age'


  1  2  2
    'pclass'    'fare'
'Box Plot of Fare by Pclass'
 'Pclass'
 'Fare'
```

In this code:

- `sns.load_dataset("titanic")` loads the "titanic" dataset.
- The first box plot (`sns.boxplot(x='pclass', y='age', data=titanic_data)`) shows the relationship between 'pclass' and 'age'.
- The second box plot (`sns.boxplot(x='pclass', y='fare', data=titanic_data)`) shows the relationship between 'pclass' and 'fare'.
- `plt.title()`, `plt.xlabel()`, and `plt.ylabel()` are used to add titles and labels to the plots.
- `plt.tight_layout()` adjusts the layout for better spacing between subplots.
- `plt.show()` displays the plots.

Adjust the plot parameters and styling according to your preferences and analysis requirements.

Note: pclass, age, and fare are columns in the titanic dataset.

Que 4: Use the "diamonds" dataset from seaborn to plot a histogram for the 'price' column. Use the hue
parameter for the 'cut' column of the diamonds dataset.

Ans:can use Seaborn's `load_dataset` function to load the "diamonds" dataset and then create a histogram for the 'price' column with different colors representing different values of the 'cut' column using the `hue` parameter. Here's the code:

python

Copy code

```
import          as
import                    as


                                 "diamonds"



                    10  6
                                          'price'     'cut'           'stack'       30
          'black'



          'Histogram of Diamond Prices by Cut'
           'Price'
           'Frequency'



                 'Cut'
```

In this code:

- `sns.load_dataset("diamonds")` loads the "diamonds" dataset.
- `sns.histplot()` is used to create a histogram with 'price' on the x-axis, 'cut' as the hue parameter, and the 'stack' option to stack the histograms for different 'cut' values.
- `plt.title()`, `plt.xlabel()`, and `plt.ylabel()` are used to add a title and labels to the plot.
- `plt.legend()` shows the legend with the different 'cut' values.
- `plt.show()` displays the plot.

Adjust the number of bins and other plot parameters based on your preferences and analysis requirements.

Que 5: Use the "iris" dataset from seaborn to plot a pair plot. Use the hue parameter for the "species" column
of the iris dataset.
Ans:can use Seaborn's `load_dataset` function to load the "iris" dataset and then create a pair plot with the 'hue' parameter to distinguish different species. Here's the code:
python

Copy code

```
import          as
import                    as


                          "iris"


                        "species"          "Set2"           "o"   "s"   "D"


              "Pair Plot of Iris Dataset by Species"    1.02
```

In this code:

- `sns.load_dataset("iris")` loads the "iris" dataset.
- `sns.pairplot()` creates a pair plot of the dataset with different markers for each species specified by the 'hue' parameter.
- `plt.suptitle()` adds a title to the entire plot.
- `plt.show()` displays the pair plot.

The pair plot provides a matrix of scatterplots for all pairs of features in the dataset, with different colors representing different species. It's a useful visualization for understanding the relationships between variables in the dataset. Adjust the plot parameters and styling based on your preferences and analysis requirements.

Que 6: Use the "flights" dataset from seaborn to plot a heatmap.

Ans:can use Seaborn's `load_dataset` function to load the "flights" dataset and then create a heatmap to visualize the correlation between months and the number of passengers. Here's the code:

python

Copy code

```
import         as
import                   as


                                "flights"


                                           'month'           'year'
       'passengers'


                     10  8
                                'coolwarm'        True      'd'             .5


          'Number of Passengers by Month and Year'
           'Year'
           'Month'
```

In this code:

- `sns.load_dataset("flights")` loads the "flights" dataset.
- `flights_data.pivot_table()` pivots the data to create a matrix suitable for a heatmap.
- `sns.heatmap()` creates the heatmap with the number of passengers on the y-axis, months on the x-axis, and color indicating the number of passengers. The `cmap` parameter sets the color map, and `annot=True` displays the values in each cell.
- `plt.title()`, `plt.xlabel()`, and `plt.ylabel()` are used to add a title and labels to the plot.
- `plt.show()` displays the heatmap.

This heatmap provides a visual representation of the number of passengers for each month and year. Adjust the plot parameters and styling based on your preferences and analysis requirements.