Assignment No - 09 (Group A)

Problem statement:

Data Visualization II

1. Use the inbuilt dataset 'titanic' as used in the above problem. Plot a box plot for distribution of age with respect to each gender along with the information about whether they survived or not. (Column names : 'sex' and 'age')

2. Write observations on the inference from the above statistics.

Pre-requisite

1. Basic of Python Programming
2. Seaborn Library, Concept of Data Visualization

Objective

Students should be able to perform the data Visualization operation using Python on any open source dataset

Software and Hardware requirements:-

1. **Operating system:** Linux- Ubuntu 16.04 to 17.10, or Windows 7 to 10,
2. **RAM-** 2GB RAM (4GB preferable)
3. IDE :-Anaconda Jupiter Notebook / pycharm /Visual Studio

Theory-

Advanced Plots:

a. The Strip Plot

The strip plot draws a scatter plot where one of the variables is categorical. We have seen scatter plots in the joint plot and the pair plot sections where we had two numeric variables.

The strip plot is different in a way that one of the variables is categorical in this case, and for each category in the categorical variable, you will see a scatter plot with respect to the numeric column.

The stripplot() function is used to plot the violin plot. Like the box plot, the first parameter is the categorical column, the second parameter is the numeric column while the third parameter is the dataset. Look at the following script:

```
sns.stripplot(x='sex', y='age', data=dataset, jitter=False)
```

You can see the scattered plots of age for both males and females. The data points look like strips. It is difficult to comprehend the distribution of data in this form. To better comprehend the data, pass True for the jitter parameter which adds some random noise to the data. Look at the following script:

```
sns.stripplot(x='sex', y='age', data=dataset, jitter=True)
```

Now you have a better view for the distribution of age across the genders. Like violin and box plots, you can add an additional categorical column to strip plot using hue parameter as shown below:

```
sns.stripplot(x='sex', y='age', data=dataset, jitter=True, hue='survived')
```

b. The Swarm Plot

The swarm plot is a combination of the strip and the violin plots. In the swarm plots, the points are adjusted in such a way that they don't overlap. Let's plot a swarm plot for the distribution of age against gender. The swarmplot() function is used to plot the violin plot. Like the box plot, the first parameter is the categorical column, the second parameter is the numeric column while the third parameter is the dataset. Look at the following script:

```
sns.swarmplot(x='sex', y='age', data=dataset)
```

You can clearly see that the above plot contains scattered data points like the strip plot and the data points are not overlapping. Rather they are arranged to give a view similar to that of a violin plot.

Let's add another categorical column to the swarm plot using the hue parameter.

```
sns.swarmplot(x='sex', y='age', data=dataset, hue='survived')
```

1. Matrix Plots

Matrix plots are the type of plots that show data in the form of rows and columns. Heat maps are the prime examples of matrix plots.

a. Heat Maps

Heat maps are normally used to plot correlation between numeric columns in the form of a matrix. It is important to mention here that to draw matrix plots, you need to have meaningful information on rows as well as columns.

Let's plot the first five rows of the Titanic dataset to see if both the rows and column headers have meaningful information. Execute the following script:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
        dataset = sns.load_dataset('titanic')
        dataset.head()
```

From the output, you can see that the column headers contain useful information such as passengers surviving, their age, fare etc. However the row headers only contain indexes 0, 1, 2, etc.

To plot matrix plots, we need useful information on both columns and row headers. One way to do this is to call the corr() method on the dataset. The corr() function returns the correlation between all the numeric columns of the dataset. Execute the following script:

```
dataset.corr()
```

In the output, you will see that both the columns and the rows have meaningful header information, as shown below:

Now to create a heat map with these correlation values, you need to call the heatmap() function and pass it your correlation dataframe. Look at the following script:

```
corr = dataset.corr()
sns.heatmap(corr)
```

b. Cluster Map:

In addition to the heat map, another commonly used matrix plot is the cluster map. The cluster map basically uses Hierarchical Clustering to cluster the rows and columns of the matrix.

Let's plot a cluster map for the number of passengers who travelled in a specific month of a specific year. Execute the following script:

Checking how the price of the ticket (column name: 'fare') for each passenger is distributed by plotting a histogram.

```
import seaborn as sns
dataset = sns.load_dataset('titanic')
sns.histplot(dataset['fare'], kde=False, bins=10)
```

Conclusion:

Seaborn is an advanced data visualisation library built on top of Matplotlib library.

In this assignment, we looked at how we can draw distributional and categorical plots using the Seaborn library.

We have seen how to plot a box plot in Seaborn. We also saw how to change plot styles and use grid functions to manipulate subplots.