**LAB EXERCISE – 4**

**Data Visualization using Seaborn**

**Aim of the Experiment**

To write python program using Seaborn for data visualization. The data visualization is done for both synthetic data as well as for preloaded Iris dataset.

**Reference to Textbook and Explanation**

Chapter 2 and Appendix 2 for details about Data visualization.

The Bar plot for week Vs sales can be done as follows: Here hue is used to specify the colour. Set\_title() is used to display the heading as part of the plot.

sns.barplot(x = "week", y = "sales", hue = "region", data = df) \

.set\_title('Bar plot for weeks Vs Sales Data')

The box plot can be done as follows:

sns.boxplot(data = df).set\_title('Box Plot for week and sales data')

plt.show()

distplots() are used to display distribution plots.

sns.distplot(df1['sepal\_width'],kde = False).set\_title('Distribution plot for sepal\_width')

plt.show()

Violin plot combines both box and distplot ability.

sns.violinplot(data = df).set\_title('Violin Plot for week and sales data')

Count plots can be used to count the attributes as follows:

sns.countplot(x = "region", data = df, palette = "Blues").set\_title('Count plot Based on regions')

Strip plots can be used to display the data based on certain attributes. Here, it is done as class vs petal length of Iris dataset.

sns.stripplot(x = "species", y = "petal\_length", data = df1).set\_title('Strip plot for petal length and class')

Joint plot can be used for bivariate distribution plot as shown below:

sns.jointplot(x = 'week',y = 'sales',data = df)

lmplot() can be used for creating regression plot of two attributes based on another attribute.

g1 = sns.lmplot(x="week", y="sales", hue="region", data=df)

Multivariate analysis can be done using pairplots for all the attributes of the dataset.

sns.pairplot(df1,hue = 'species',kind = "scatter")

**Program Listing – 1**

import seaborn as sns

import pandas as pd

from matplotlib import pyplot as plt

# Create a synthetic dataset

data = {'week': [1,2,3,4,5],

'sales':[1.2,1.8,2.6,3.2,3.8],

'region':['south','south','north','north','north']

}

df = pd.DataFrame(data)

# Read the preloaded dataset Iris using data frame 1 - df1

df1 = sns.load\_dataset('iris')

sns.set(color\_codes=True)

### Univariate Plots

# Histogram plot using barplot

print("\nHistogram of week vs Sales\n")

sns.barplot(x = "week", y = "sales", hue = "region", data = df) \

.set\_title('Bar plot for weeks Vs Sales Data')

plt.show()

# Box Plot

print("\nBox Plot of week and sales data\n")

sns.boxplot(data=df)

plt.xlabel("Statistics", size=18)

plt.ylabel("Week and Sales Data", size=18)

plt.show()

# Distplot - Distribution plots

# Display the histogram by making kde flag as falsse

sns.distplot(df1['sepal\_width'],kde = False).set\_title('Distribution plot for sepal\_width')

plt.xlabel("Sepal Width", size=18)

plt.ylabel("Frequency", size=18)

plt.show()

# Display the Distribution plot by making hist flag as false

sns.distplot(df1['sepal\_width'],hist = False).set\_title('Distribution plot for sepal\_width')

plt.xlabel("Sepal Width", size=18)

plt.ylabel("Frequency", size=18)

plt.show()

# Violin Plot

print("\n Violin Plot of week and sales data\n")

sns.violinplot(data = df).set\_title('Violin Plot for week and sales data')

plt.xlabel("Statistics", size=18)

plt.ylabel("Region", size=18)

plt.show()

# Count Plot for Iris Data

sns.countplot(x = "region", data = df, palette = "Blues").set\_title('Count plot Based on regions')

plt.xlabel("Statistics", size=18)

plt.ylabel("Region", size=18)

plt.show()

# Strip plot for Iris Data

print('\n Strip plot\n')

sns.stripplot(x = "species", y = "petal\_length", data = df1).set\_title('Strip plot for petal length and class')

**Output**

**Histogram of week Vs sales**

**Chart, bar chart

Description automatically generated**

**Box Plot for week and sales**

**Chart, box and whisker chart

Description automatically generated**

**Distribution plot in histograms**

**Chart, histogram

Description automatically generated**

**Distribution plot in kernel Distribution function**

**Chart, line chart

Description automatically generated**

**Violin Plot**

**A picture containing chart

Description automatically generated**

**Count plot of week and sales based on Region**

**Chart, treemap chart

Description automatically generated**

**Strip Plot**

**Chart, scatter chart

Description automatically generated**

**Program Listing 2**

import seaborn as sns

import pandas as pd

from matplotlib import pyplot as plt

# Create a synthetic dataset

data = {'week': [1,2,3,4,5],

'sales':[1.2,1.8,2.6,3.2,3.8],

'region':['south','south','north','north','north']

}

df = pd.DataFrame(data)

# Read the preloaded dataset Iris using data frame 1 - df1

df1 = sns.load\_dataset('iris')

sns.set(color\_codes=True)

### Bivariate Plots

# Joint Plot

print('\n Joint Plot of week and sales\n')

sns.jointplot(x = 'week',y = 'sales',data = df)

plt.show()

#lmplot

print('\n lmplot of week and sales Based on Region\n\n')

g1 = sns.lmplot(x="week", y="sales", hue="region", data=df)

### Multivariate Plots

# pairplot

sns.pairplot(df1,hue = 'species',kind = "scatter")

plt.show()

**Output**

A picture containing sitting, water, photo, side

Description automatically generated

Lmplot of week vs Sales based on region

A close up of text on a white surface

Description automatically generated

Pairplot of week and sales

