

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.

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.

WEEK	SALES
1	1
2	3
3	4
4	8

```
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)
```

Implot() 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 false
```

```
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')
```

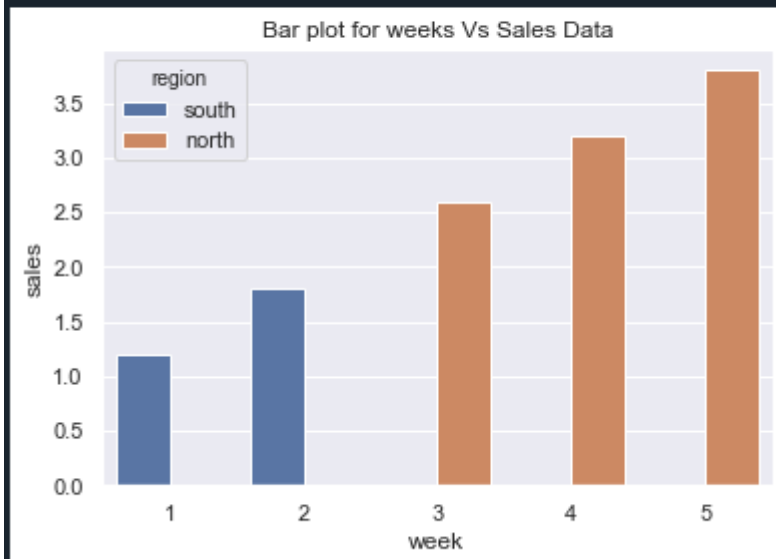
```
sns.stripplot(x = "species", y = "petal_length", data = df1).set_title('Strip plot for petal length and class')
```

Output

Histogram of week Vs sales

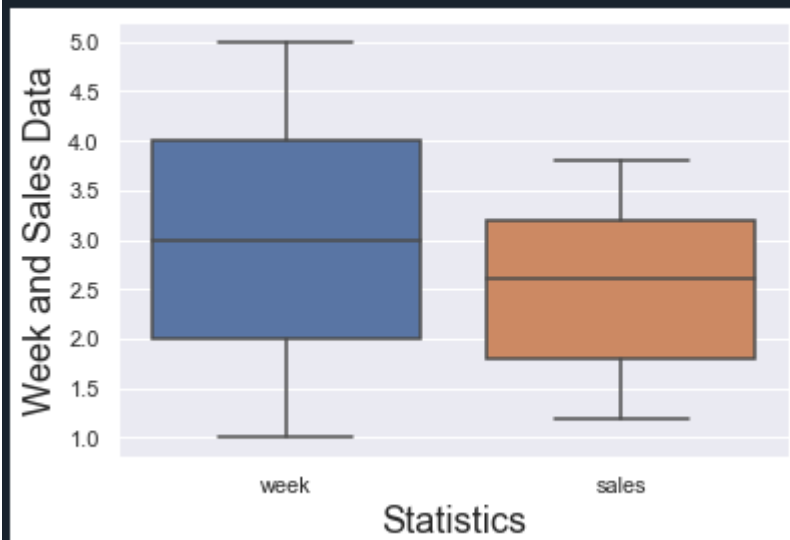
```
In [29]: runfile('D:/Test/sns-example-uni.py', wdir='D:/Test')
```

Histogram of week vs Sales

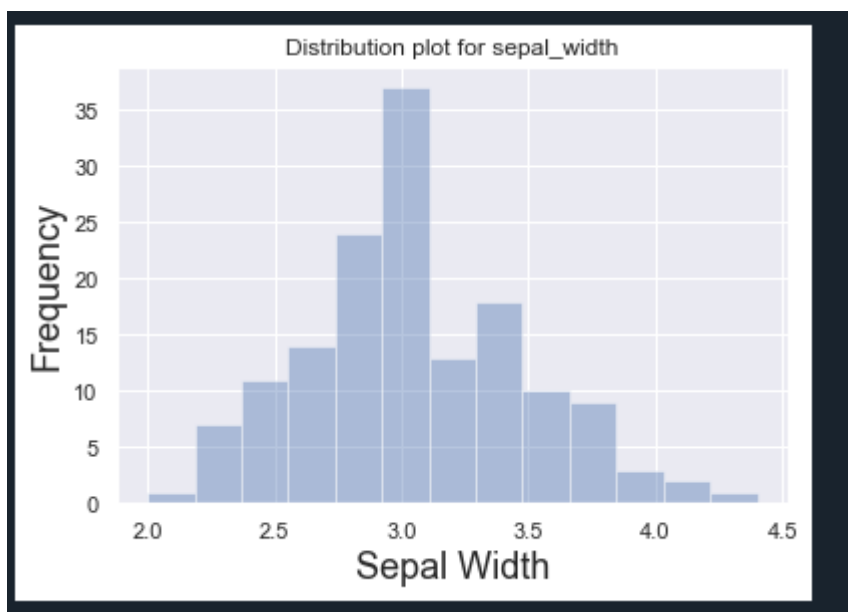


Box Plot for week and sales

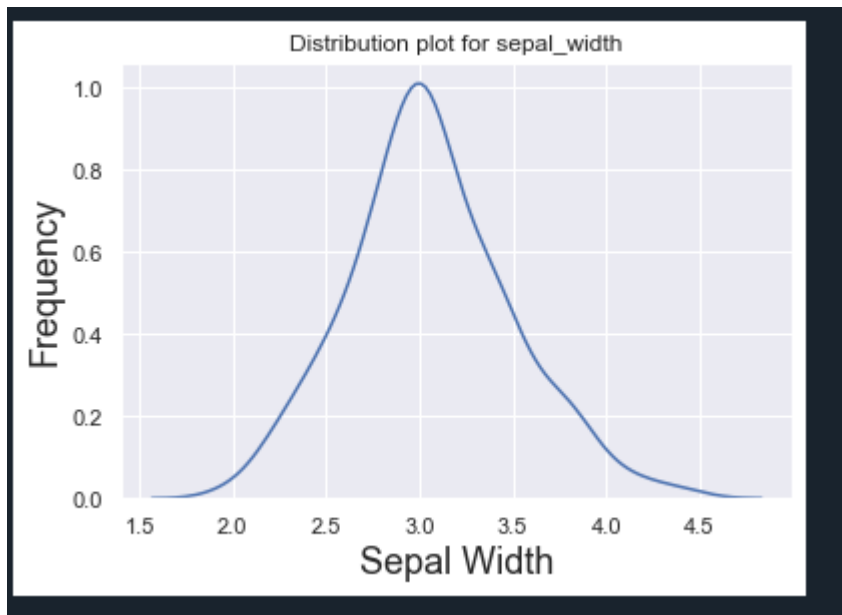
Box Plot of week and sales data



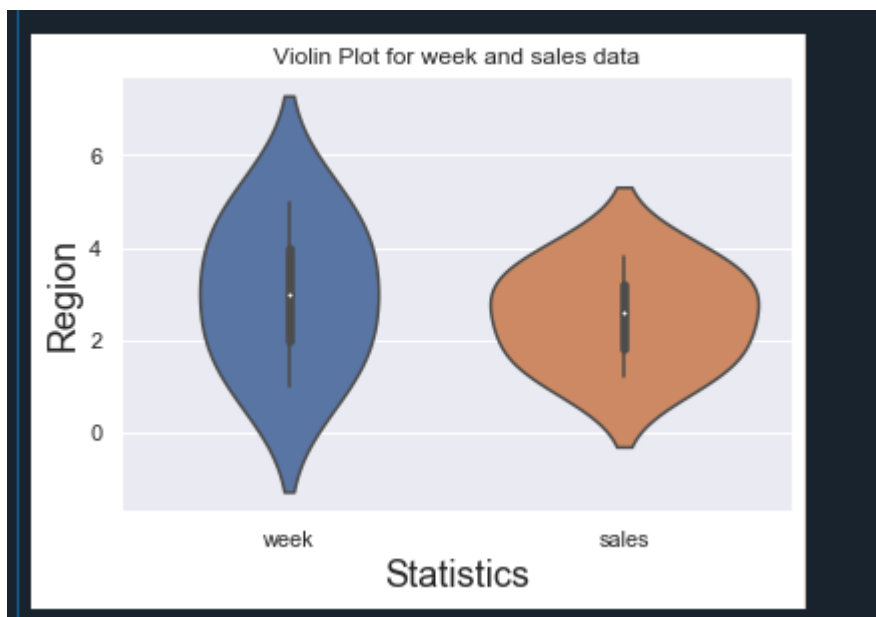
Distribution plot in histograms



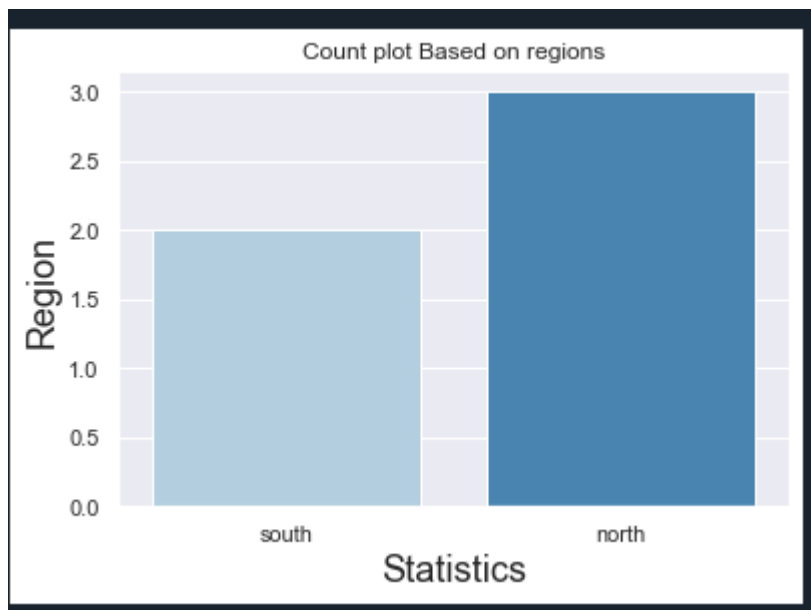
Distribution plot in kernel Distribution function



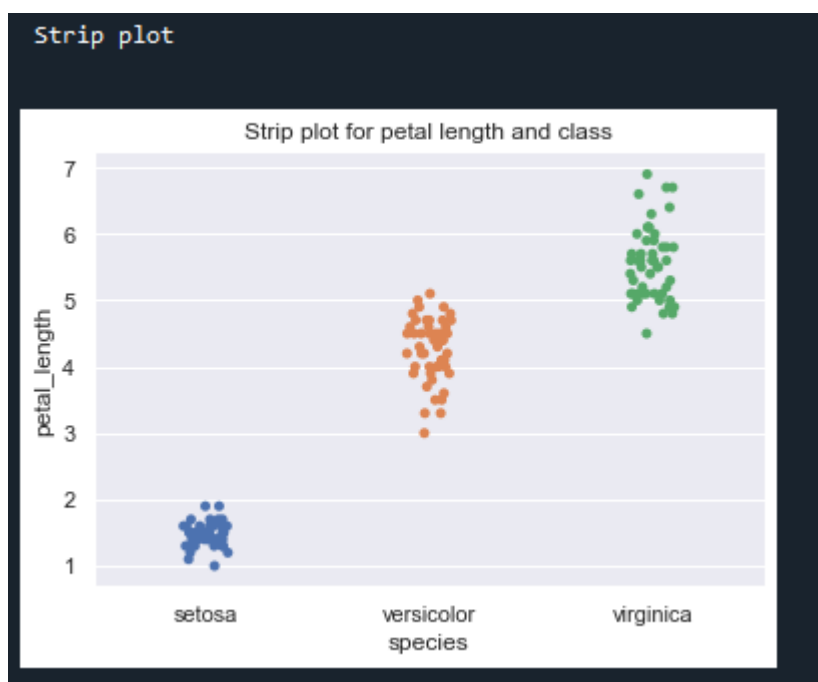
Violin Plot



Count plot of week and sales based on Region



Strip Plot



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()

#Implot

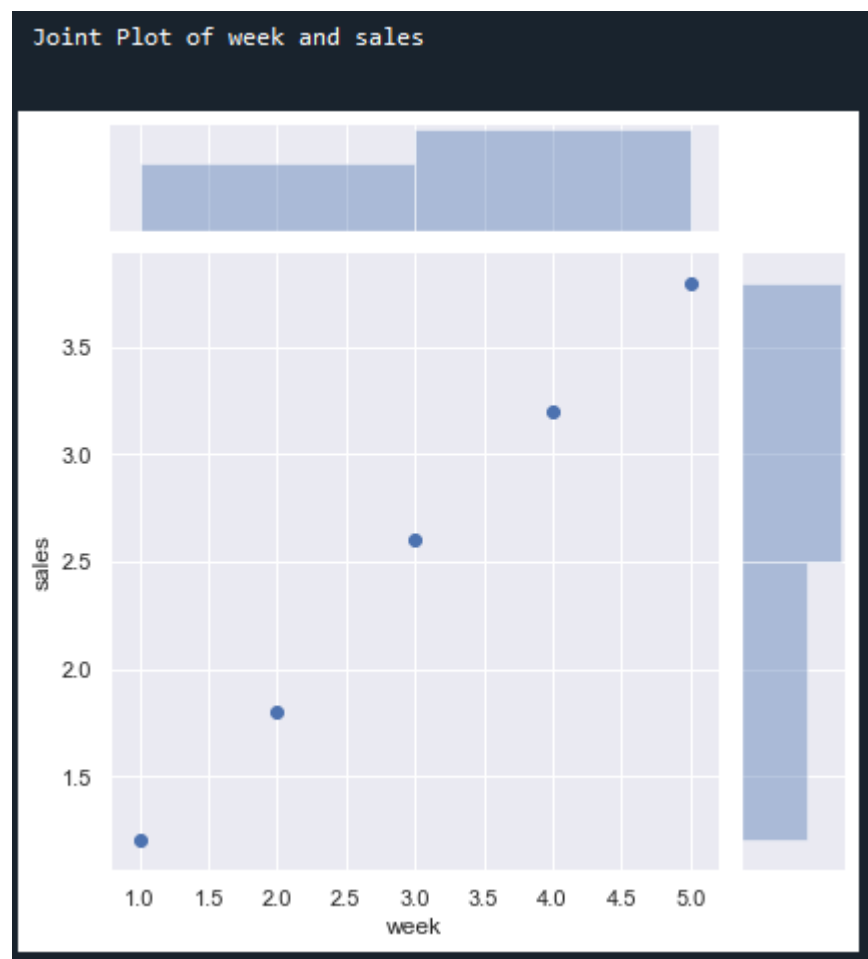
print('\n Implot of week and sales Based on Region\n\n')
g1 = sns.lmplot(x="week", y="sales", hue="region", data=df)

### Multivariate Plots

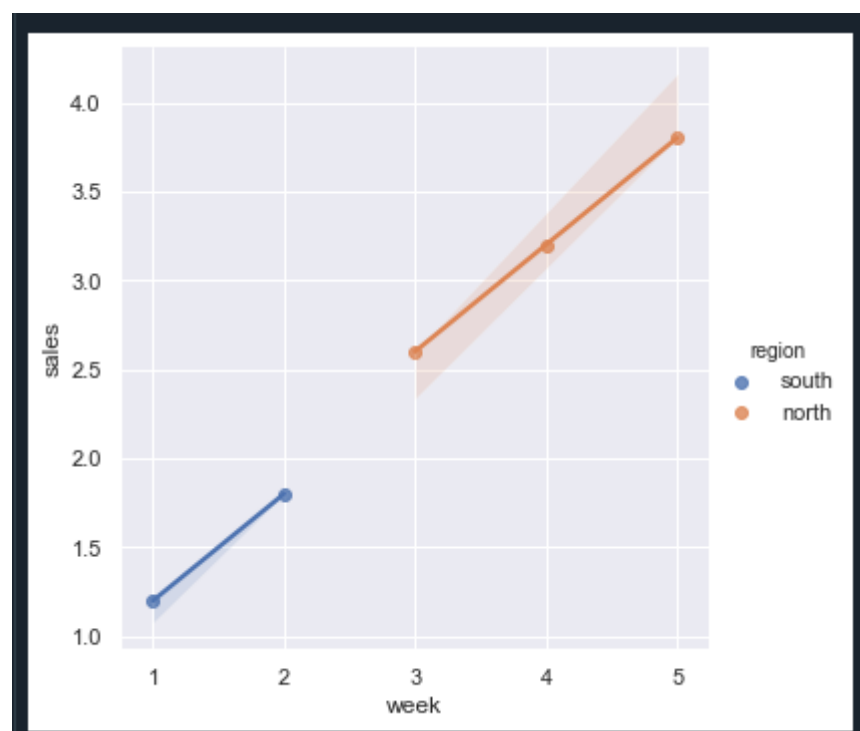
# pairplot
sns.pairplot(df1,hue = 'species',kind = "scatter")
plt.show()

```

Output



lmplot of week vs Sales based on region



Pairplot of week and sales

