DSP505: Programming Lab for Data Science and Artificial Intelligence

TPL616: Advanced Programming for DSAI

(Seaborn Tutorial)

Vishwesh Jatala

Assistant Professor

Department of CSE

Indian Institute of Technology Bhilai

vishwesh@iitbhilai.ac.in

What is Seaborn and Why?

- Seaborn is a popular Python data visualization library built on top of Matplotlib.
- It provides a high-level interface for drawing attractive and informative statistical graphics.
- Features:
  - Simpler Syntax
  - Built-in Themes
  - Better Statistical Plots
  - Automatic Handling of DataFrames
  - Integration with Matplotlib
  - Beautiful Default Visualizations

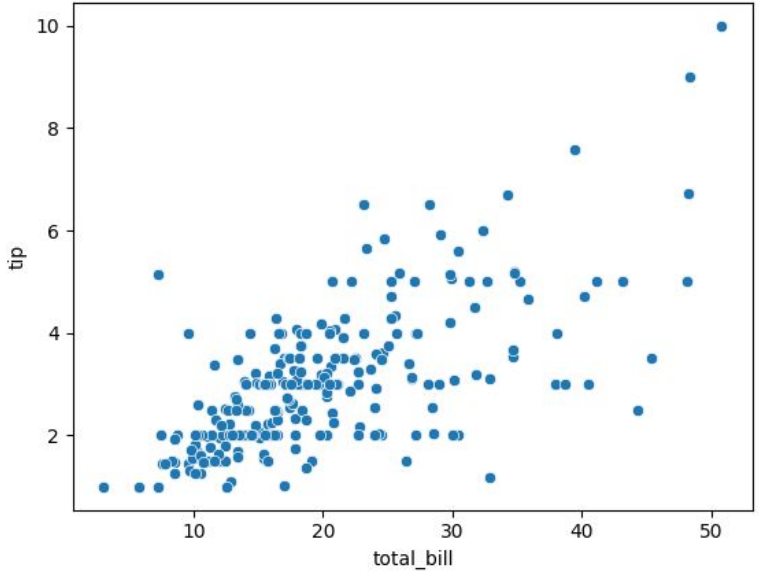
Tips Dataset

```
import pandas as pd
import seaborn as sns

tips = sns.load_dataset("tips")

tips.head()
```

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

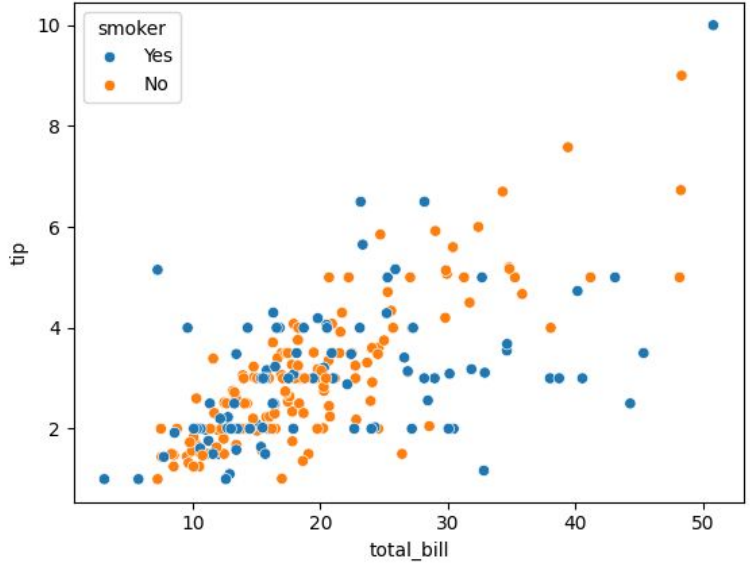
Simple Scatter Plot

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.scatterplot
(x="total_bill", y="tip",
data=tips)
plt.show()
```



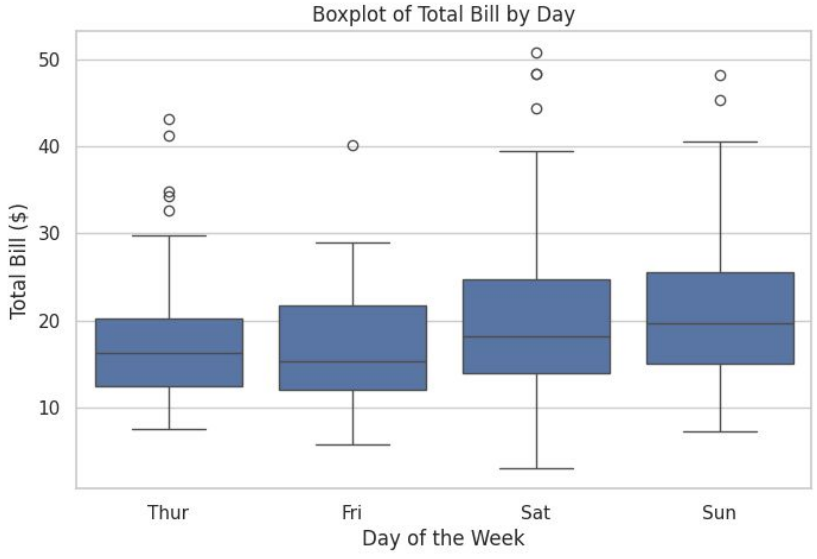
Simple Scatter Plot with Hue

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.scatterplot(x="total_bill",
y="tip",
data=tips,
hue="smoker")
plt.show()
```



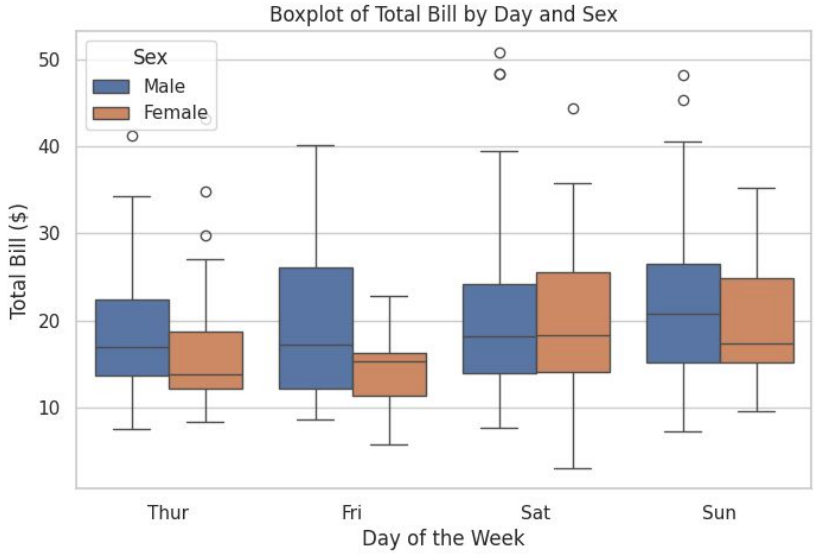
Boxplot

```
import seaborn as sns
import matplotlib.pyplot as plt

tips = sns.load_dataset("tips")
# Create a boxplot
plt.figure(figsize=(8, 5))
sns.boxplot(x="day", y="total_bill",
data=tips)

plt.title("Boxplot of Total Bill by Day and
Sex")
plt.xlabel("Day of the Week")
plt.ylabel("Total Bill ($)")
plt.legend(title="Sex")
plt.show()
```



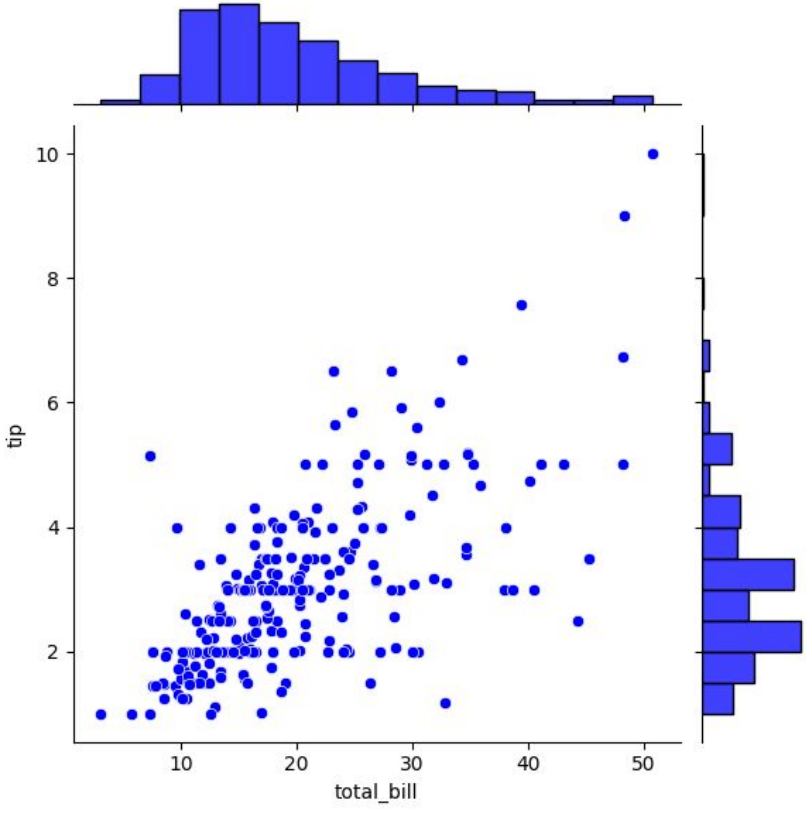
Boxplot

```
import seaborn as sns
import matplotlib.pyplot as plt

# Load the built-in dataset
tips = sns.load_dataset("tips")

# Create a boxplot
plt.figure(figsize=(8, 5))
sns.boxplot(x="day", y="total_bill",
data=tips, hue="sex")

plt.title("Boxplot of Total Bill by Day and
Sex")
plt.xlabel("Day of the Week")
plt.ylabel("Total Bill ($)")
plt.legend(title="Sex")
plt.show()
```



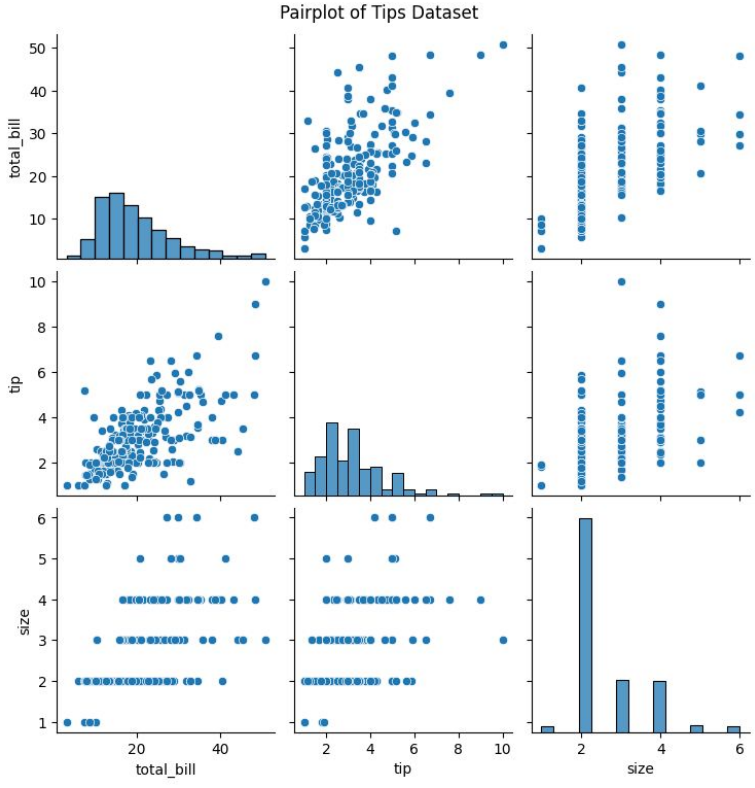
Jointplot

```
import seaborn as sns
import matplotlib.pyplot as plt

# Load the tips dataset
tips = sns.load_dataset('tips')

# Create a jointplot to visualize the
relationship between 'total_bill' and
'tip'
sns.jointplot(x="total_bill", y="tip",
data=tips, kind="scatter", color="blue")

# Show the plot
plt.show()
```



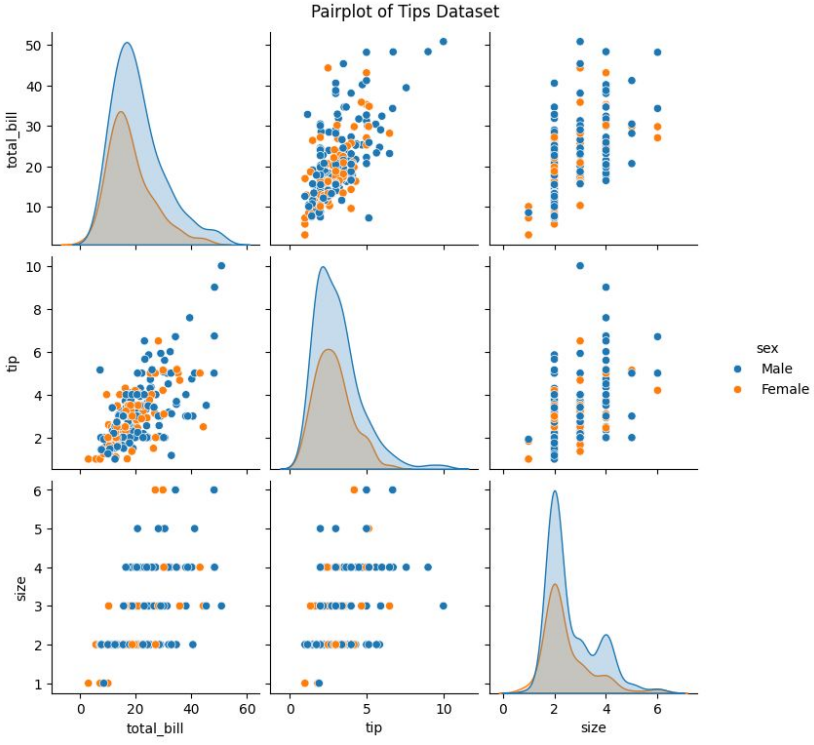
Pairplot

```
import seaborn as sns
import matplotlib.pyplot as plt

# Load the dataset
tips = sns.load_dataset("tips")

# Create a pairplot
sns.pairplot(tips)

plt.suptitle("Pairplot of Tips Dataset",
y=1.02)
plt.show()
```



Pairplot

```
import seaborn as sns
import matplotlib.pyplot as plt

# Load the dataset
tips = sns.load_dataset("tips")

# Create a pairplot
sns.pairplot(tips, hue="sex",
diag_kind="kde")

plt.suptitle("Pairplot of Tips Dataset",
y=1.02)
plt.show()
```



Andrews Curve

- Andrews Curves is a method used for visualizing high-dimensional data by representing each data point (observation) as a curve in a 2D plane.
- Useful when you want to explore multivariate data and detect patterns, clusters, or groupings in datasets with multiple variables.
- The technique is based on projecting high-dimensional data into a 2D curve space where each data point is represented by a continuous curve.
- $x=[x_1,x_2,\ldots,x_N]$ is an N dimensional data point

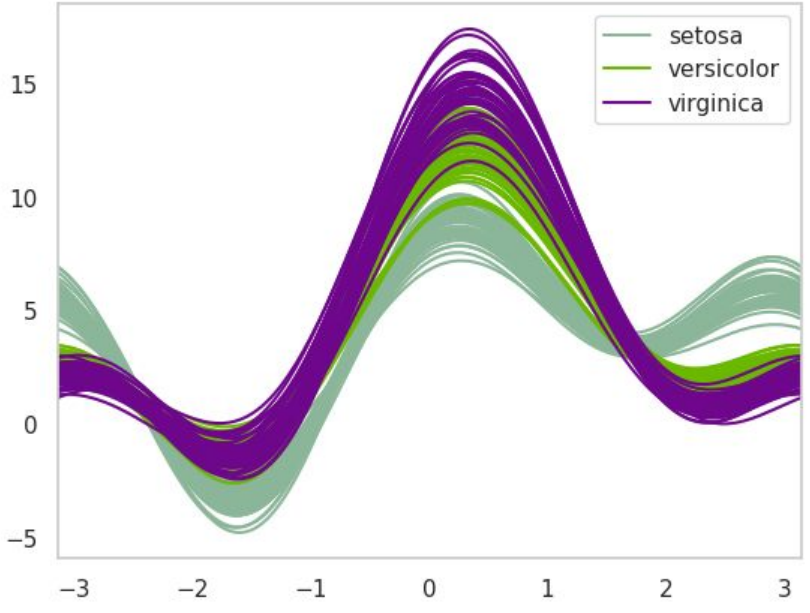
$$y(t) = \frac{1}{\sqrt{N}} \left(x_1 \cdot \cos(t) + x_2 \cdot \sin(t) + x_3 \cdot \cos(2t) + x_4 \cdot \sin(2t) + \cdots \right)$$

IRIS Dataset

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

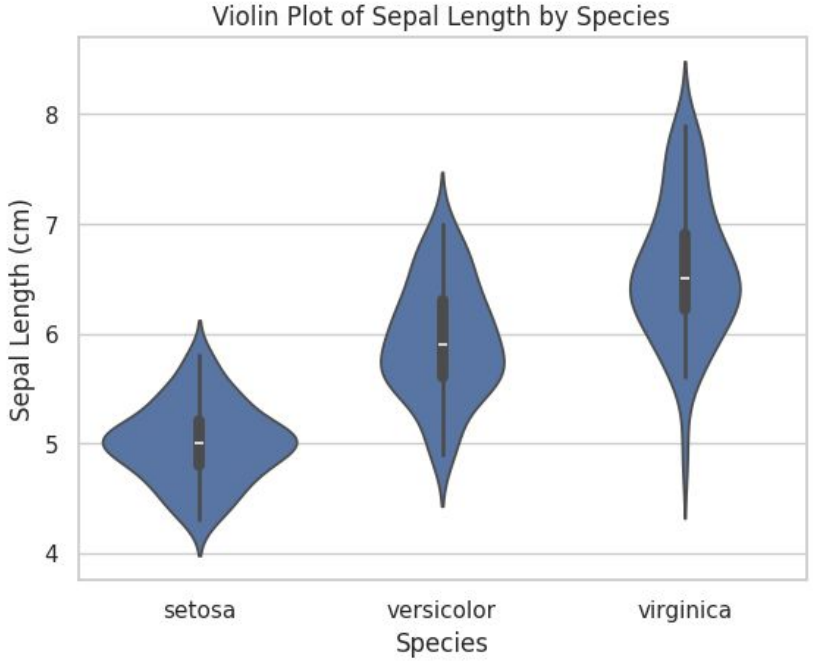
Andrews Curves Plot

```
import seaborn as sns
import pandas.plotting
from pandas.plotting import
andrews_curves

# Load the Iris dataset
iris = sns.load_dataset('iris')
iris


andrews_curves(iris, "species")
```



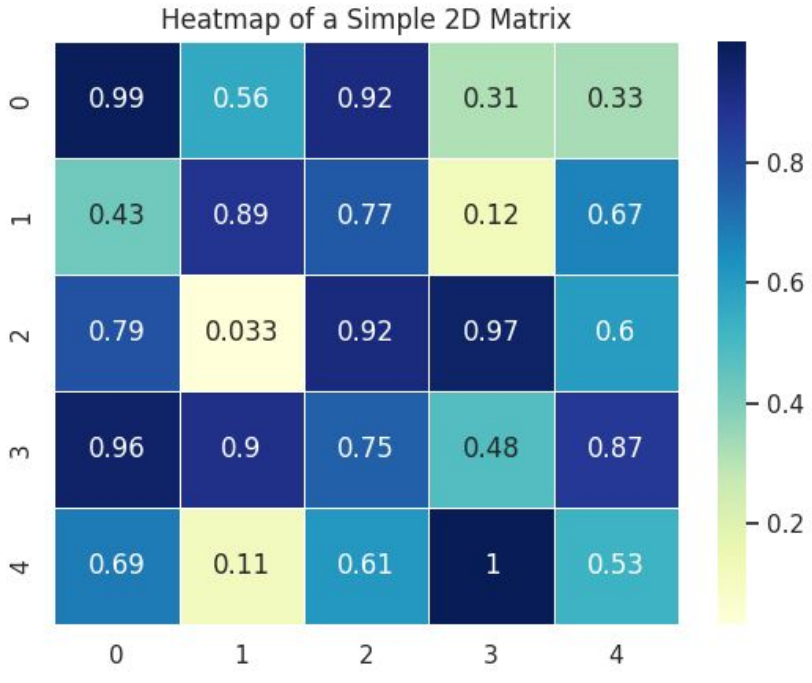
Violin Plots

```
import seaborn as sns
import matplotlib.pyplot as plt

# Load the Iris dataset
iris = sns.load_dataset('iris')
sns.violinplot(x='species', y='sepal_length',
data=iris)

# Add title and labels
plt.title('Violin Plot of Sepal Length by
Species')
plt.xlabel('Species')
plt.ylabel('Sepal Length (cm)')

# Show the plot
plt.show()
```



Heatmaps

```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# Create a simple 5x5 matrix
data = np.random.rand( 5, 5)

# Create a heatmap of the 2D matrix
sns.heatmap(data, annot= True, cmap='YlGnBu',
linewidths=0.5)

# Add a title
plt.title('Heatmap of a Simple 2D Matrix' )

# Show the plot
plt.show()
```



Thank you