National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Faculty of Computing

CS 368: Reinforcement Learning

Lab 02: Data Visualization in Python

Date: 20 September 2024

Time: 2:00 PM – 5:00 PM

Instructor: Dr. Zuhair Zafar

Lab 02: Data Visualization in Python

Objectives

Understand different visualizations techniques using matplotlib library.

Tools/Software Requirement

Google Colab, Python

Introduction

Python has different modules for visualizing data such as matplotlib, seaborn. Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It presents data in 2D graphics. Seaborn is a visualization library that is built on top of Matplotlib. It provides data visualizations that are typically more aesthetic and statistically sophisticated. Matplotlib can be installed using the following command:

```
pip install matplotlib
```

Once the module installed, it must be imported into the program using the following command:

```
import matplotlib as mpl
```

, where mpl is the alias name given to matplotlib library.

matplotlib.pyplot is a state-based interface to matplotlib. matplotlib.pyplot is a collection of functionsthat make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decoratesthe plot with labels etc. pyplot can be imported into the program using following command:

```
import matplotlib.pyplot as plt
```

Following are some of the basic data visualization plots:

1. Line plots
2. Area plots
3. Histograms
4. Bar charts
5. Pie charts
6. Box plots
7. Scatter plots

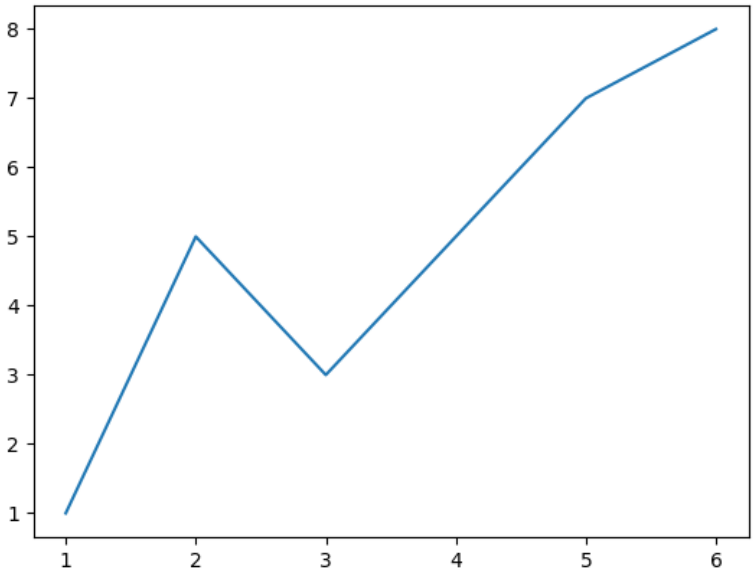
Line Plots:

A line plot is used to represent quantitative values over a continuous interval or time period. It is generally used to depict trends on how the data has changed over time.

Program:

```
x = [1, 2, 3, 4, 5, 6]
y = [1, 5, 3, 5, 7, 8]
plt.plot(x, y)
plt.show()
```

Output:



Area Plots:

An Area Plot is also called as Area Chart which is used to display magnitude and proportion of multiple variables.

Program:

```
days = [1,2,3,4,5]
sleeping =[7,8,6,11,7]
eating = [2,3,4,3,2]
working =[7,8,7,2,2]
playing = [8,5,7,8,13]
plt.plot([],[],color='m', label='Sleeping', linewidth=5)
plt.plot([],[],color='c', label='Eating', linewidth=5)
plt.plot([],[],color='r', label='Working', linewidth=5)
plt.plot([],[],color='k', label='Playing', linewidth=5)
plt.stackplot(days, sleeping,eating,working,playing, colors=['m','c','r','k'])
plt.xlabel('x')
plt.ylabel('y')
```

```
plt.title('Stack Plot')
plt.legend()
plt.show()
```

Output:

Histograms:

Histogram represents the frequency distribution of a dataset. It is a graph showing the number of observations within each given interval.

Program:

```
population_age=[22,55,62,45,21,22,34,42,42,4,2,102,95,85,55,110,120,70,65,55,111,115,80]
bins = [0,10,20,30,40,50,60,70,80,90,100]
plt.hist(population_age, bins, histtype='bar', rwidth=0.8)
plt.xlabel('age groups')
plt.ylabel('Number of people')
plt.title('Histogram')
plt.show()
```

Output:

Bar Charts:

A Bar chart or bar graph is a chart or graph that presents categorical data with rectangular bars with heights or lengths proportional to the values that they represent. A bar plot is a way of representing data where the length of the bars represents the magnitude/size of the feature/variable.

Program:

```
plt.bar([0.25,1.25,2.25,3.25,4.25],[50,40,70,80,20],label="BMW",width=.5)
plt.bar([.75,1.75,2.75,3.75,4.75],[80,20,20,50,60],label="Audi", color='r',width=.5)
plt.legend()
plt.xlabel('Days')
plt.ylabel('Distance (kms)')
plt.title('Information')
plt.show()
```

Output:

Pie Charts:

A Pie chart is a circular statistical chart, which is divided into sectors to illustrate numerical proportion.

Program:

```
days = [1,2,3,4,5]
sleeping =[7,8,6,11,7]
eating = [2,3,4,3,2]
working =[7,8,7,2,2]
playing = [8,5,7,8,13]
slices = [7,2,2,13]
activities = ['sleeping','eating','working','playing']
cols = ['c','m','r','b']
plt.pie(slices, labels=activities, colors=cols, startangle=90, shadow= True,
explode=(0,0.1,0,0), autopct='%1.1f%%')
plt.title('Pie Plot')
plt.show()
```

Output:

Box Plots:

A Box plot (or box-and-whisker plot) shows the distribution of quantitative data in a way that facilitates comparisons between variables or across levels of a categorical variable. Box plot shows the quartiles of the dataset while the whiskers extend encompass the rest of the distribution but leave out the points that are the outliers.

Program:

```
x=[1,2,3,4,5,6,7]
y=[1,2,4,5,3,6,9]
z=[x,y]
plt.boxplot(z,labels=["A","B"],showmeans=True)
plt.show()
```

Output:

Scatter Plots:
A Scatter chart, also called a scatter plot, is a chart that shows the relationship between two variables.

Program:

```
x=[1,1.5,2,2.5,3,3.5,3.6]
y=[7.5,8,8.5,9,9.5,10,10.5]
x1=[8,8.5,9,9.5,10,10.5,11]
y1=[3,3.5,3.7,4,4.5,5,5.2]
plt.scatter(x,y, label='high income low saving',color='r')
plt.scatter(x1,y1,label='low income high savings',color='b')
plt.xlabel('saving*100')
plt.ylabel('income*1000')
plt.title('Scatter Plot')
plt.legend()
plt.show()
```

Output:

Lab Tasks

Begin by importing pandas and seaborn, a data analysis tookit and graphing library, respectively.

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

To ignore warnings, use the following code to make the display more attractive.

```
import warnings
warnings.filterwarnings("ignore")
```

Import the Iris dataset

```
iris = pd.read_csv("iris.csv")
```

1. **Create a scatter plot using Pandas to visualize the relationship between sepal_length and sepal_width from the Iris dataset. Use plt.show() from Matplotlib to display the plot.**

2. **Create a bivariate scatterplot with univariate histograms using Seaborn's jointplot() to visualize sepal_length and sepal_width from the Iris dataset. Use plt.show() to display the plot.**

3. **Create a box plot using Seaborn's boxplot() to visualize the distribution of sepal_length across different species in the Iris dataset. Use plt.show() to display the plot.**

4. **Create a box plot using Seaborn's boxplot() to visualize sepal_length across species, then overlay data points using striplot() with jitter=True to scatter the points. Assign ax to stack both plots and use plt.show() to display them.**

5. **Use Seaborn's pairplot() to analyze relationships between species across all feature combinations in the Iris dataset. Set hue="species" to color the data by species and display the plot with plt.show().**

Deliverable:

Please submit your notebook on LMS before the deadline **(23 September 2024, 11:59pm).**

Lab Rubrics

Assessment	Does not meet expectation (1/2 marks)	Meets expectation (3/4 marks)	Exceeds expectation (5 marks)
Software Problem Realization **(CLO1, PLO1)**	The student struggles to apply the given visualization problem and does not identify an appropriate technique to solve it. There is a lack of demonstration of the problem's data requirements and no attempt to preprocess or explore the data effectively.	The student applies the given visualization problem with some guidance, identifies a suitable technique with hints, and shows basic visualizations. However, the approach might not be fully optimized or lacks a thorough justification.	The student independently applies the given visualization problem, selects the most appropriate technique without guidance, and effectively visualizes and explores the data. The choice of algorithm and data handling demonstrates a deep understanding of the problem's context and data characteristics.
Software Tool Usage **(CLO4, PLO5)**	Code has syntax errors, and the implementation of the visualization is incorrect or incomplete. The code is not modular and lacks comments for readability and reuse. The student shows no ability to use relevant visualization libraries where required.	The student has implemented the visualization correctly with minor mistakes. The code is mostly correct in terms of syntax and functionality but might not be optimized or well-structured for reuse. Some documentation is provided. The student also shows working knowledge of relevant visualization libraries where required.	The student has implemented the visualization algorithm efficiently and correctly. The code is clean, modular, well-documented, and follows best practices for reproducibility and reuse. The student demonstrates full command of visualization libraries and tools.