Sinhgad Institutes

Name of the Student: _____ **Roll No:** ___

CLASS: - T. E. [COMP] **Division: B** **Course: DSBDA**

Group A : Data Science

Assignment No. 10

Data Visualization III

Marks:

Date of Performance: / /2023

Sign with Date:

Title: Perform Data Visualization

Objectives:

- To perform EDA using Seaborn library

Outcomes:

CO5: Implement data visualization techniques

PEOs, POs, PSOs and COs satisfied

PEOs: I, III **POs: 1, 2, 3, 5** **PSOs: 1** **COs: 5**

Problem Statement:

Data Analysis on the IRIS Flower Dataset

Download the Iris flower dataset or any other dataset into a DataFrame.

(eg https://archive.ics.uci.edu/ml/datasets/Iris)

Use Python/R and perform following –

1. How many features are there and what are their types (e.g., numeric, nominal)?
2. Compute and display summary statistics for each feature available in the dataset. (eg. minimum value, maximum value, mean, range, standard deviation, variance and percentiles)
3. Data Visualization-Create a histogram for each feature in the dataset to illustrate the feature distributions. Plot each histogram.
4. Create a boxplot for each feature in the dataset. All of the boxplots should be combined into a single plot. Compare distributions and identify outliers.

Theory:

Data Visualization using Seaborn:

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

Keys Features

- Seaborn is a statistical plotting library

- It has beautiful default styles
- It also is designed to work very well with Pandas dataframe objects.

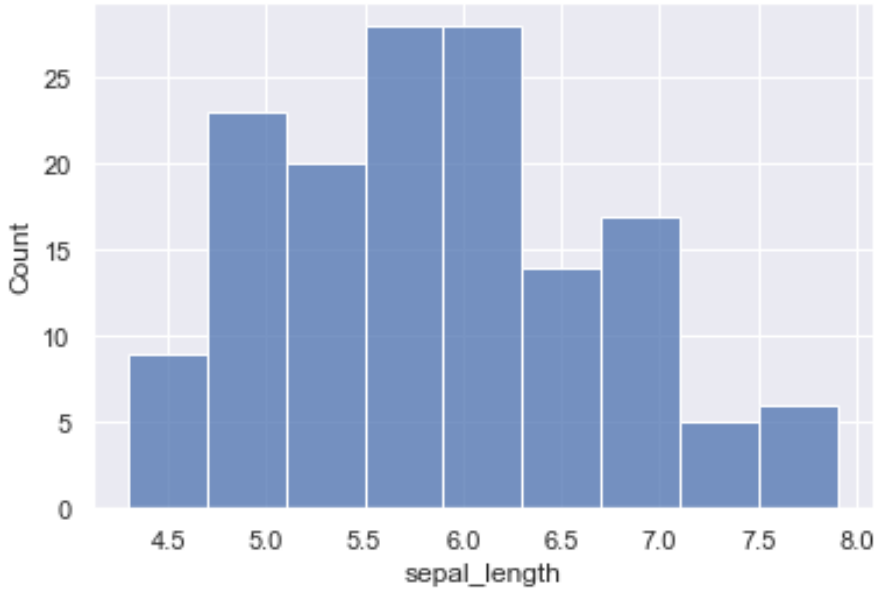
Histogram:

In Seaborn, you can create a histogram using the histplot() function. This function takes a dataset and a few optional parameters, such as the number of bins, and creates a histogram of the data.

Here's an example of how to create a histogram using Seaborn:

```
# libraries & dataset

import seaborn as sns

import matplotlib.pyplot as plt

# set a grey background (use sns.set_theme() if seaborn version 0.11.0 or above)

sns.set(style="darkgrid")

df = sns.load_dataset("iris")

sns.histplot(data=df, x="sepal_length")

plt.show()
```

Output:



Use of Histogram:

Histograms are commonly used in data science for exploratory data analysis (EDA) to understand the distribution of a dataset. Some of the ways histograms can be useful in data science include:

- Understanding the shape of the distribution: Histograms can help you quickly visualize whether the data is normally distributed, skewed, bimodal, or has other characteristics that may impact how you analyze the data.

- Identifying outliers: Histograms can also help identify any extreme values or outliers in the data that may skew your analysis.

- Comparing distributions: Histograms can be used to compare the distribution of two or more variables or subgroups within a dataset, which can be useful in identifying patterns or trends.

- Choosing appropriate statistical methods: Understanding the shape of the distribution can help you choose appropriate statistical methods for analyzing the data, such as whether to use parametric or non-parametric tests.

- Feature engineering: Histograms can be used to create new features or variables by binning or discretizing continuous variables into categorical variables based on the distribution of the data.

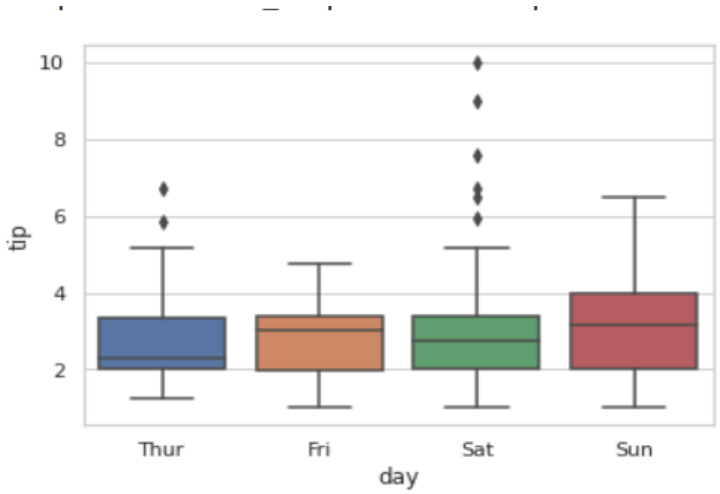
Boxplot:

One of the most commonly used plots in Seaborn is the boxplot, which is a graphical representation of the distribution of a dataset based on five summary statistics: minimum, maximum, median, and first and third quartiles.

To create a boxplot in Seaborn, you can use the boxplot() function, which is part of the Seaborn sns module. Here is an example:

```
import seaborn as sns

import pandas as pd

# Load example data

tips =  sns.load_dataset('tips')

# Create boxplot

sns.boxplot(x="day", y="total_bill", data=tips)
```

Output:



Use of boxplot:

- Visualizing the distribution of a dataset: Boxplots provide a concise visual summary of the distribution of a dataset, showing the median, quartiles, and any outliers. This can help to quickly identify whether a dataset is skewed, symmetric, or has other features.

- Comparing the distribution of multiple datasets: Boxplots can be used to compare the distribution of multiple datasets side-by-side, making it easy to identify differences in the medians, quartiles, or ranges.

- Detecting outliers: Boxplots can also be used to identify potential outliers in a dataset, which may require further investigation or data cleaning.

- Assessing the spread of the data: The width of the box in a boxplot represents the interquartile range (IQR), which is a measure of the spread of the data. Comparing the IQR across different datasets can give insight into differences in the variability of the data.

- Checking for normality: Boxplots can also be used to assess whether a dataset is approximately normally distributed. If the data is normally distributed, the boxplot will show a symmetrical shape with the median in the center.

Outliers:

Outliers are data points that deviate significantly from other observations in a dataset. Outliers can arise due to a variety of reasons, such as measurement error, data entry errors, or genuine extreme values. Outliers can potentially affect the results of data analysis and modeling, and

thus it is important to identify and handle them appropriately. Here are some commonly used techniques for outlier detection and treatment:

Outlier Detection:

- Visual inspection: A simple way to detect outliers is to plot the data and visually inspect for any observations that appear to be significantly different from the others.

- Descriptive statistics: Another way to detect outliers is to calculate descriptive statistics such as the mean, standard deviation, or quartiles, and then look for observations that are far from the central tendency or spread of the data.

- Quantile-based methods: Quantile-based methods involve calculating thresholds based on the quartiles of the data and then identifying observations that fall outside these thresholds.

- Model-based methods: Model-based methods involve fitting a statistical model to the data and then identifying observations that have a large residual or influence on the model.

Outlier treatment methods:

Outliers can potentially affect the results of data analysis and modeling, and thus it is important to identify and handle them appropriately. Here are some commonly used outlier treatment methods in data science:

- Removal: One approach to handling outliers is to simply remove them from the dataset. This can be appropriate if the outliers are due to measurement error or other anomalies that are unlikely to occur again. However, removing outliers can also reduce the size of the dataset and potentially affect the statistical power of subsequent analyses.

- Imputation: If removing outliers is not appropriate, another approach is to impute or replace the outliers with a more appropriate value. This can be done using various methods such as mean imputation, median imputation, or regression imputation. However, imputation methods can potentially introduce bias into the data and may not accurately reflect the true distribution of the data.

- Winsorization: Winsorization is a method of outlier treatment that involves replacing extreme values with the nearest non-extreme value. For example, if an observation is below the 5th percentile or above the 95th percentile, it would be replaced with the value at the 5th or 95th percentile, respectively. Winsorization can help to reduce the impact of outliers while still retaining the original distribution of the data.

- Log-transformation: Log-transformation is a method of outlier treatment that involves applying a logarithmic transformation to the data. This can make the data more symmetric and reduce the impact of outliers on subsequent analyses.

Conclusion:- In this way we have explored the functions of the Seaborn python library for Data Visualization.

A. Write short answer of following questions:

1. **What is a histplot in Seaborn?**
2. **What is the difference between a distplot() and a histplot() in Seaborn?**
3. **How do you create a histplot in Seaborn?**
4. **What is the purpose of the bins parameter in a histplot?**
5. **What is the default estimator used in a histplot?**
6. **What is the purpose of the kde parameter in a histplot?**