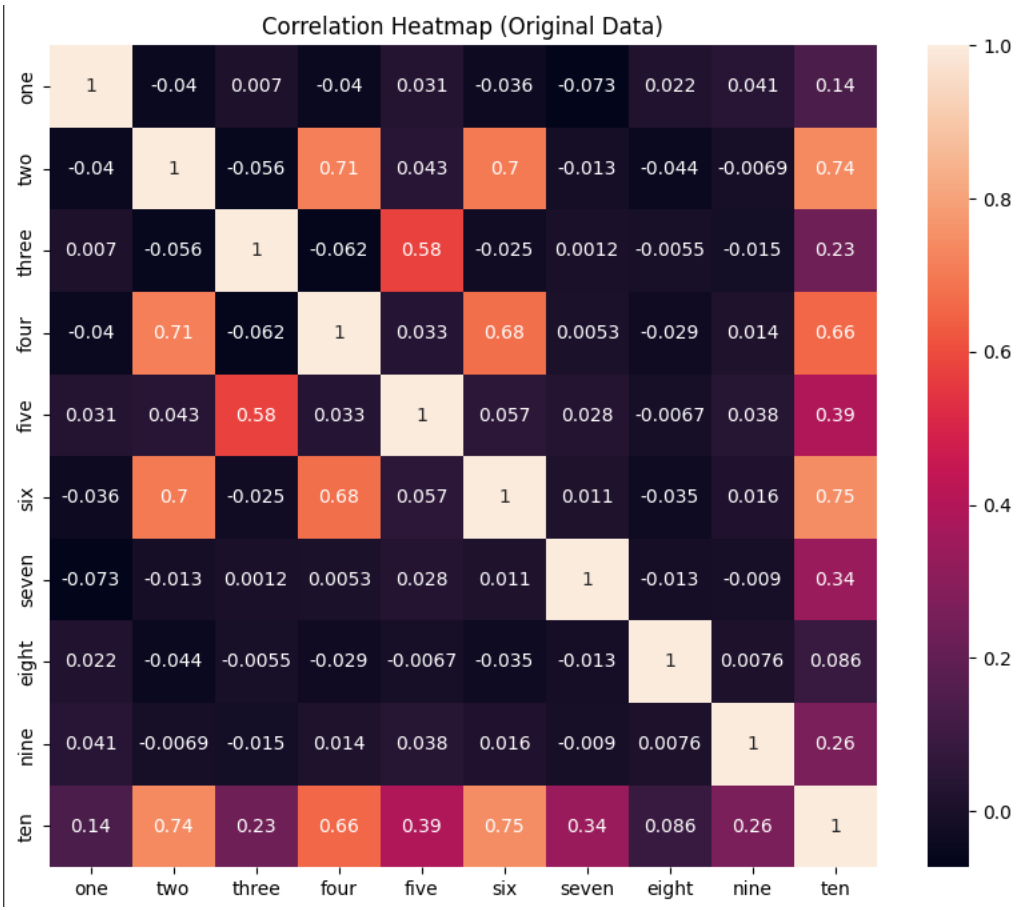
REPORT FOR DATASET ANALYSIS

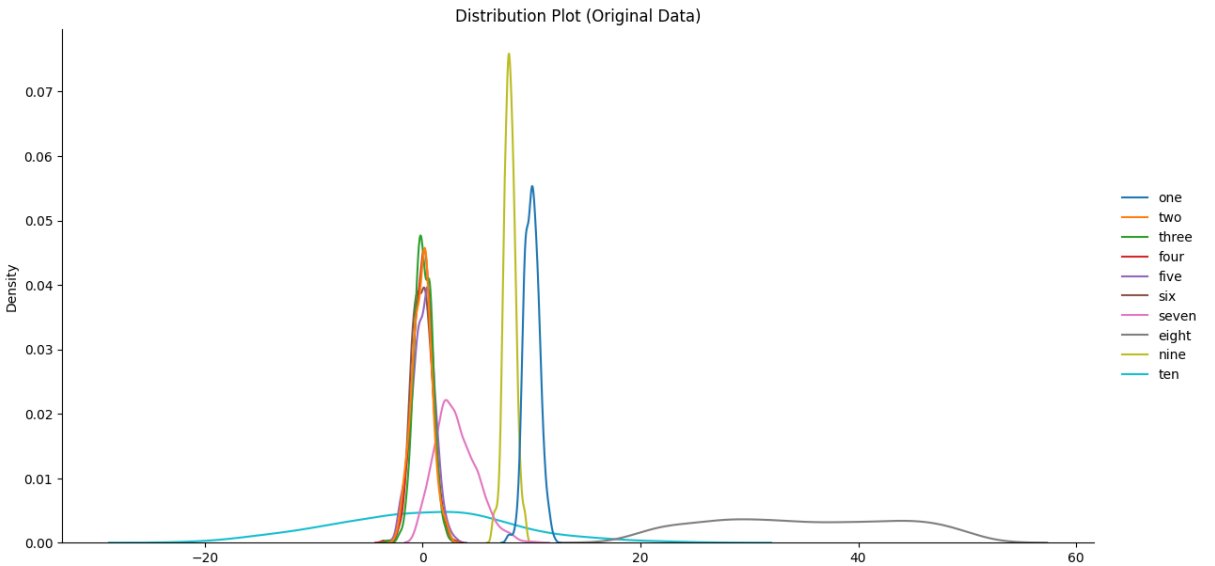
Introduction:

In this code, we analyze a dataset using pandas, numpy, seaborn, and matplotlib libraries in Python. The dataset is loaded from a CSV file named "dataset.csv". The data is first analyzed in its original form and then normalized to analyze the normalized data.

Data Analysis:

The code first loads the dataset into a pandas dataframe and drops the index column. The summary statistics of the original data are then printed using the describe() function. Next, a correlation heatmap and a distribution plot are created using seaborn and matplotlib libraries.





Data Normalization:

The data is then normalized using the min-max normalization technique. The normalized data is obtained by subtracting the minimum value of each column from each value of that column and then dividing the result by the difference between the maximum and minimum values of that column. The summary statistics of the normalized data are then printed using the describe() function. Next, a correlation heatmap and a distribution plot are created for the normalized data.

```
Normalized Data Analysis
------------------------
Summary Statistics:
              one         two       three        four        five         six  \
count  500.000000  500.000000  500.000000  500.000000  500.000000  500.000000
mean     0.517396    0.484046    0.554979    0.548118    0.506372    0.463504
std      0.157671    0.180392    0.135454    0.135640    0.160713    0.180400
min      0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
25%      0.395695    0.361878    0.465990    0.457533    0.404060    0.326834
50%      0.516503    0.491729    0.549293    0.550378    0.520566    0.464730
75%      0.624384    0.594390    0.652032    0.631446    0.610235    0.592495
max      1.000000    1.000000    1.000000    1.000000    1.000000    1.000000

            seven       eight        nine         ten
count  500.000000  500.000000  500.000000  500.000000
mean     0.305000    0.504232    0.507584    0.464023
std      0.184453    0.289065    0.169793    0.174080
min      0.000000    0.000000    0.000000    0.000000
25%      0.200000    0.260258    0.393374    0.346392
50%      0.300000    0.491356    0.500840    0.471626
75%      0.400000    0.763384    0.615521    0.572044
max      1.000000    1.000000    1.000000    1.000000
```

Heatmap:

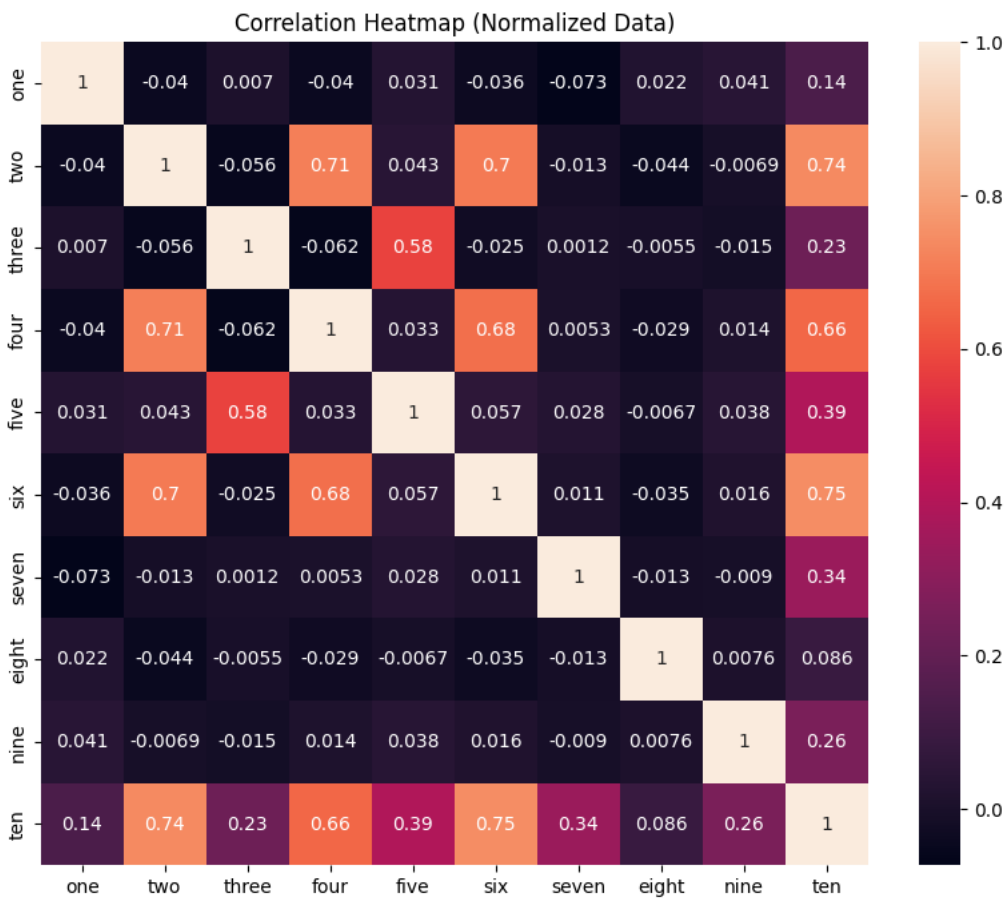
In data visualization, a heatmap is a graphical representation of data where the individual values in a matrix are represented as colors. The heatmap is a 2D plot, where the values are

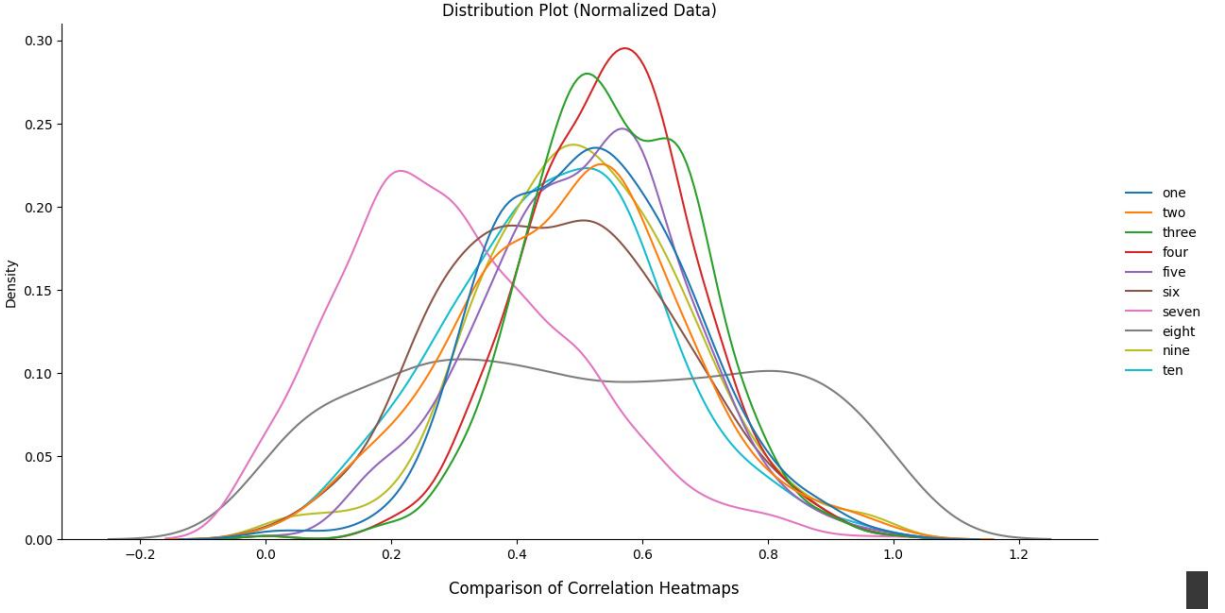
represented by colors, and the rows and columns of the matrix are displayed on the x and y-axis of the plot, respectively.

The heatmap is a useful tool for analyzing the correlation between different variables in a dataset. A correlation heatmap is a type of heatmap that is used to visualize the correlation between different features of a dataset. It provides a color-coded representation of the strength of the correlation between different pairs of variables.

The colors used in a heatmap are typically arranged in a spectrum from cool to warm. Cool colors such as blue and green represent low values, while warm colors such as red and yellow represent high values. The middle color is typically white, which represents a neutral value.

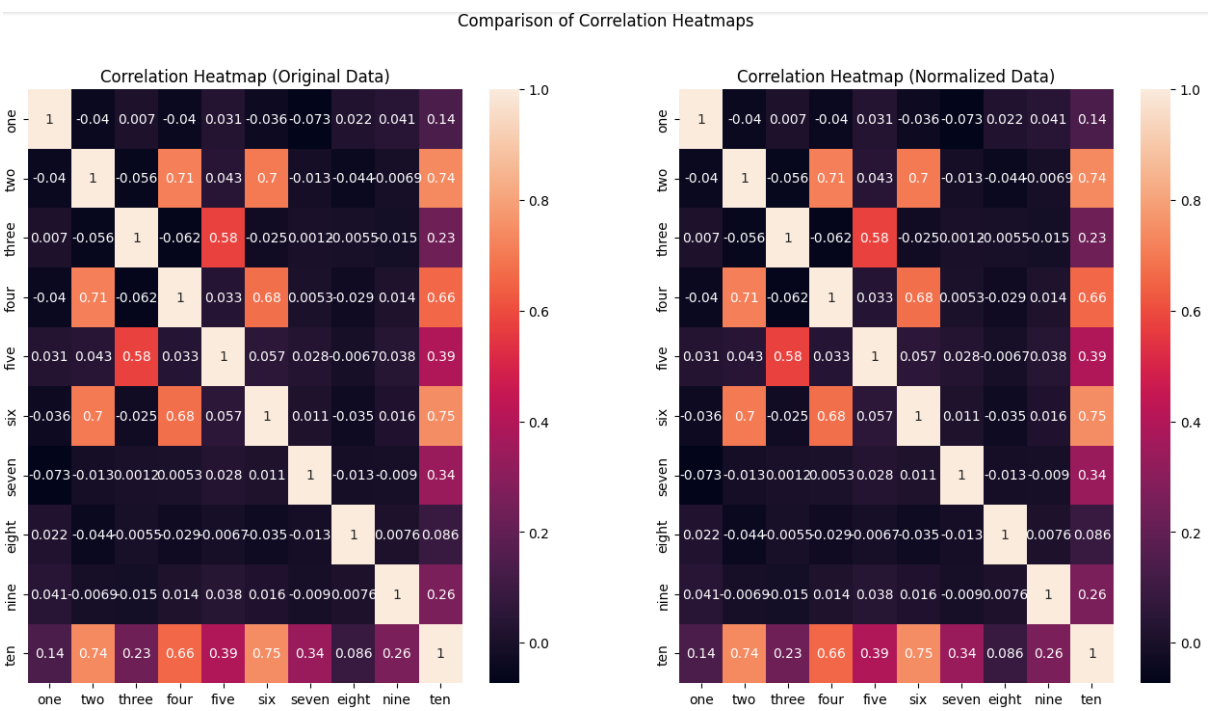
In a correlation heatmap, warm colors such as red and yellow are used to represent high correlation values, indicating a positive correlation between the variables. Cool colors such as blue and green are used to represent low correlation values, indicating a negative correlation between the variables. The intensity of the colors represents the strength of the correlation, with darker colors indicating a stronger correlation.





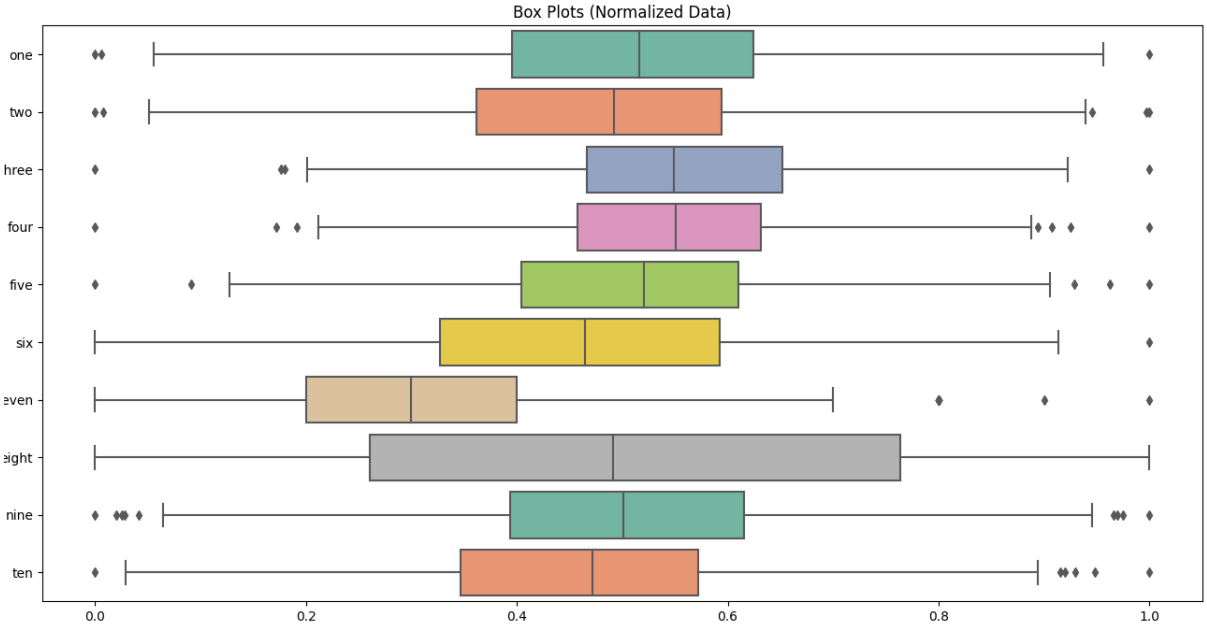
Comparison:

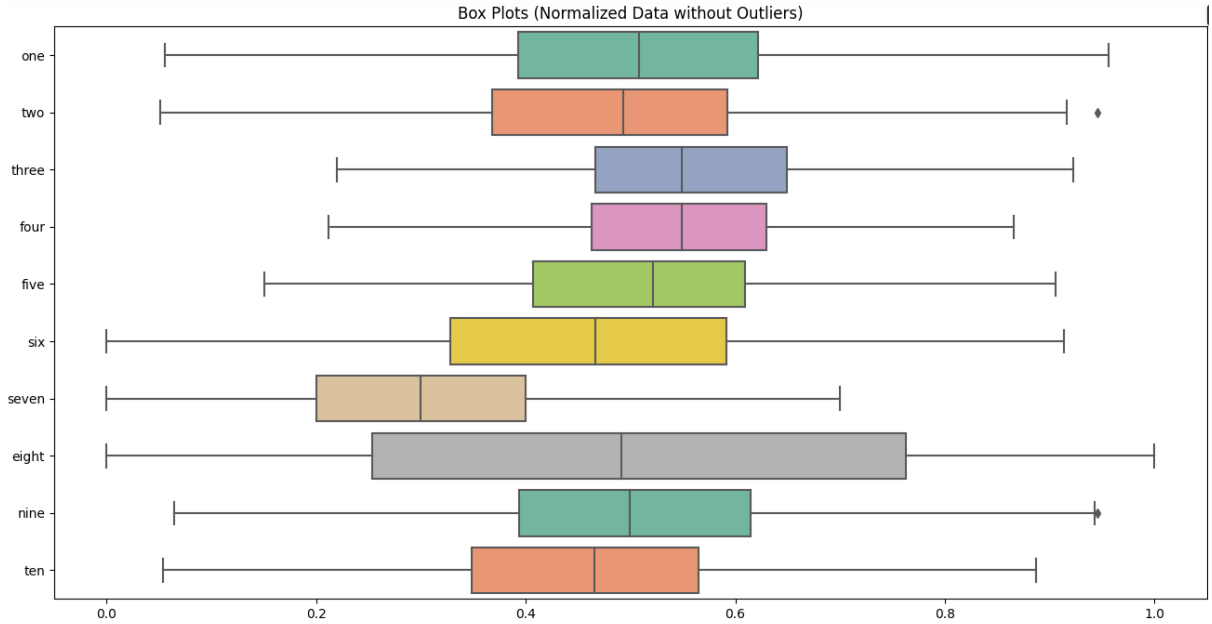
The code then compares the correlation heatmaps of the original and normalized data using a subplot. The subplot shows two heatmaps side by side, one for the original data and the other for the normalized data. The title of the subplot is "Comparison of Correlation Heatmaps".



Box Plots:

Finally, the code creates box plots of the original and normalized data using seaborn library, and removes the outliers from the plots using the "showfliers=False" parameter.





Conclusion:

In this code, we have analyzed a dataset using various statistical techniques and visualizations. We have also normalized the data to improve the correlation analysis. The box plots have provided an insight into the distribution of the data, and the removal of outliers has made the plots more readable. The code provides a good example of how to analyze and visualize a dataset using Python libraries.