**2a. Graphical Representation**

**Instructions:**

Please share your answers filled in-line in the word document. Submit code separately wherever applicable.

Please ensure you update all the details:

**Name: Vishvash C Batch ID: 23012024**

**Topic: Data Visualization**

**Guidelines:**

**1. An assignment submission is considered complete only when the correct and executable code(s) is submitted along with the documentation explaining the method and results. Failing to submit either of those will be considered an invalid submission and will not be considered a correct submission.**

**2. Ensure that you submit your assignments correctly. Resubmission is not allowed.**

**3. Post the submission you can evaluate your work by referring to the keys provided. (will be available only post the submission).**

**Hints: Follow CRISP-ML(Q) methodology steps, where were appropriate.**

1. **Data Understanding: work on each feature of the dataset to create a data dictionary as displayed in the image below:**

Table

Description automatically generated

**Make a table as shown above and provide information about the features such as its data type and its relevance to the model building. And if not relevant, provide reasons and a description of the feature.**

**Problem Statements:**

1. Univariate plots for UNIV data (Plot must have Title, X & Y label)

A) Plot numerical column with 3 different plots ?

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

df = pd.read\_excel(r"C:\Users\Lenovo\Downloads\University\_Clustering.xlsx")

df

df.describe()

df.dtypes

df1 = df.drop(columns = df.columns[:3]) #dropping categorical columns

df2 = df1.fillna(df1.mean()) #handling null values with mean

# Plot 1: Box plot

for i in range(3):

plt.subplot(2, 2, i + 1)

plt.boxplot(df2[df2.columns[i]])

plt.title(f"Box plot of {df2.columns[i]}")

plt.xlabel(df2.columns[i])

plt.ylabel('Frequency')

plt.tight\_layout()

plt.show()

# Plot 2: Histogram

for i in range(3):

plt.subplot(2, 2, i + 1)

plt.hist(df2[df2.columns[i]], bins='auto')

# sns.histplot(df2[df2.columns[i]], kde=True)

plt.title(f"Histogram of {df2.columns[i]}")

plt.xlabel(df2.columns[i])

plt.ylabel('Frequency')

plt.tight\_layout()

plt.show()

# Plot 3: Density plot over histogram

for i in range(3):

plt.subplot(2, 2, i + 1)

plt.hist(df2[df2.columns[i]], bins='auto', density=True, alpha=0.5, color='blue')

df2[df2.columns[i]].plot(kind='density', color='red')

plt.title(f"Density plot of {df2.columns[i]}")

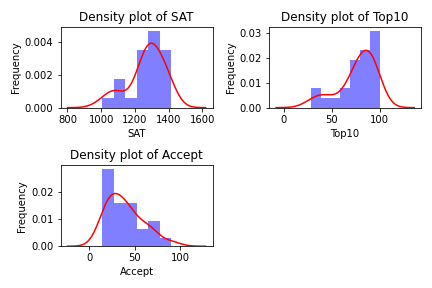
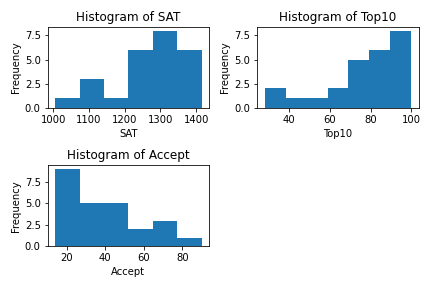
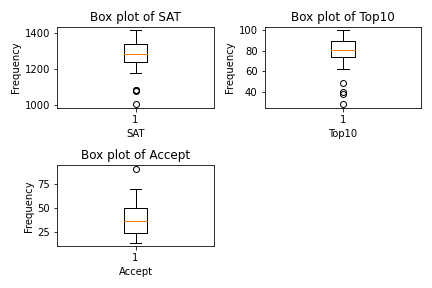
plt.xlabel(df2.columns[i])

plt.ylabel('Frequency')

plt.tight\_layout()

plt.show()

**Output:**



B) What are bin parameters? What are the methods to define the number of bins and bin sizes ?

* Bins are intervals into which data points are grouped or divided.
* The number of bins and their size can impact the interpretation and visualization of the data. Some of the methods are
* Equal-width bins: Dividing the data range into a fixed number of equally spaced intervals.
* Equal-frequency bins: Dividing the data into intervals with an equal number of data points in each interval.

C) Why do density plots exceed the range values of the column ?

* Density plots estimate the probability density function (PDF) of the data using a kernel density estimation which estimates that extend beyond the observed range of the data.
* This behavior is a result of the smoothing process used in KDE, which can lead to extrapolation beyond the observed data range.

D) Plot categorical columns by taking unique values ?

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Plot categorical column

plt.figure(figsize=(8, 6))

sns.countplot(x=df[df.columns[2]])

plt.title('Count plot of {}'.format(df.columns[2]))

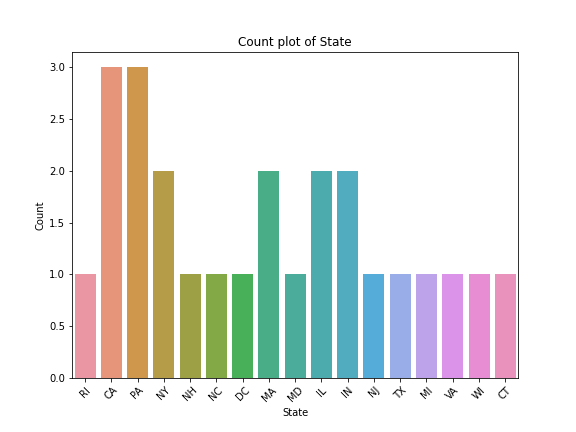
plt.xlabel('{}'.format(df.columns[2]))

plt.ylabel('Count')

plt.xticks(rotation=45)

plt.show()

**Output:**



2. Bivariate graphs for UNIV data (Plot must be readable [use rotation], have all labels)

A) Plot 2 numerical columns with scatter plot [use grid] ?

plt.figure(figsize=(8, 6))

plt.scatter(df[df.columns[3]], df[df.columns[4]])

plt.title('Scatter plot of {} vs {}'.format(df.columns[3], df.columns[4]))

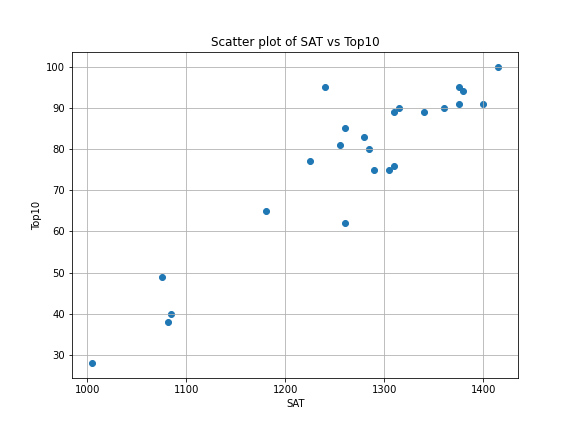
plt.xlabel('{}'.format(df.columns[3]))

plt.ylabel('{}'.format(df.columns[4]))

plt.grid(True) # Add grid

plt.show()

**Output:**



B) 2 Different plots for plotting a numerical column with a categorical column (bar, line) ?

# Bar plot

plt.figure(figsize=(8, 6))

sns.barplot(x=df3[df3.columns[2]], y=df3[df3.columns[3]], data=df3, )

plt.title('Bar plot of {} by {}'.format(df3.columns[3], df3.columns[2]))

plt.xlabel('{}'.format(df3.columns[2]))

plt.ylabel('{}'.format(df3.columns[3]))

plt.xticks(rotation=45) # Rotate x-axis labels for better readability

plt.show()

# Line plot

plt.figure(figsize=(8, 6))

sns.lineplot(x=df3[df3.columns[2]], y=df3.columns[3], data = df3)

plt.title('Line plot of {} by {}'.format(df3.columns[3], df3.columns[2]))

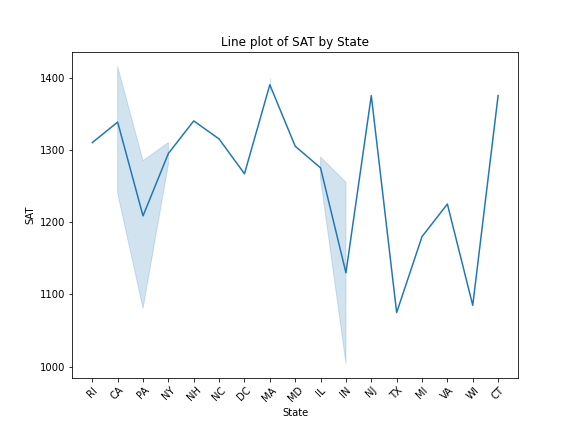
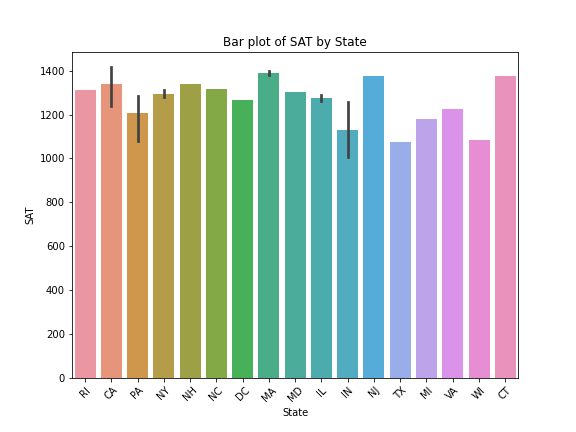
plt.xlabel('{}'.format(df3.columns[2]))

plt.ylabel('{}'.format(df3.columns[3]))

plt.xticks(rotation=45) # Rotate x-axis labels for better readability

plt.show()

**Output:**



C) How are bar plots different from histogram?

* Bar plots are used to visualize the distribution of categorical data, where each bar represents the frequency or count of a category.
* Histograms are used to visualize the distribution of continuous numerical data, where each bar represents the frequency or count of data points falling within a specific range or bin.

3. Plot multivariate graphs (correlation heatmap, pairplot)

A) Plot for only numerical data ?

import seaborn as sns

import matplotlib.pyplot as plt

# Correlation heatmap for numerical data

plt.figure(figsize=(10, 8))

sns.heatmap(df2.corr(), annot=True, cmap='coolwarm', fmt=".2f")

plt.title('Correlation Heatmap for Numerical Data')

plt.show()

# Pairplot for numerical data

sns.pairplot(df2)

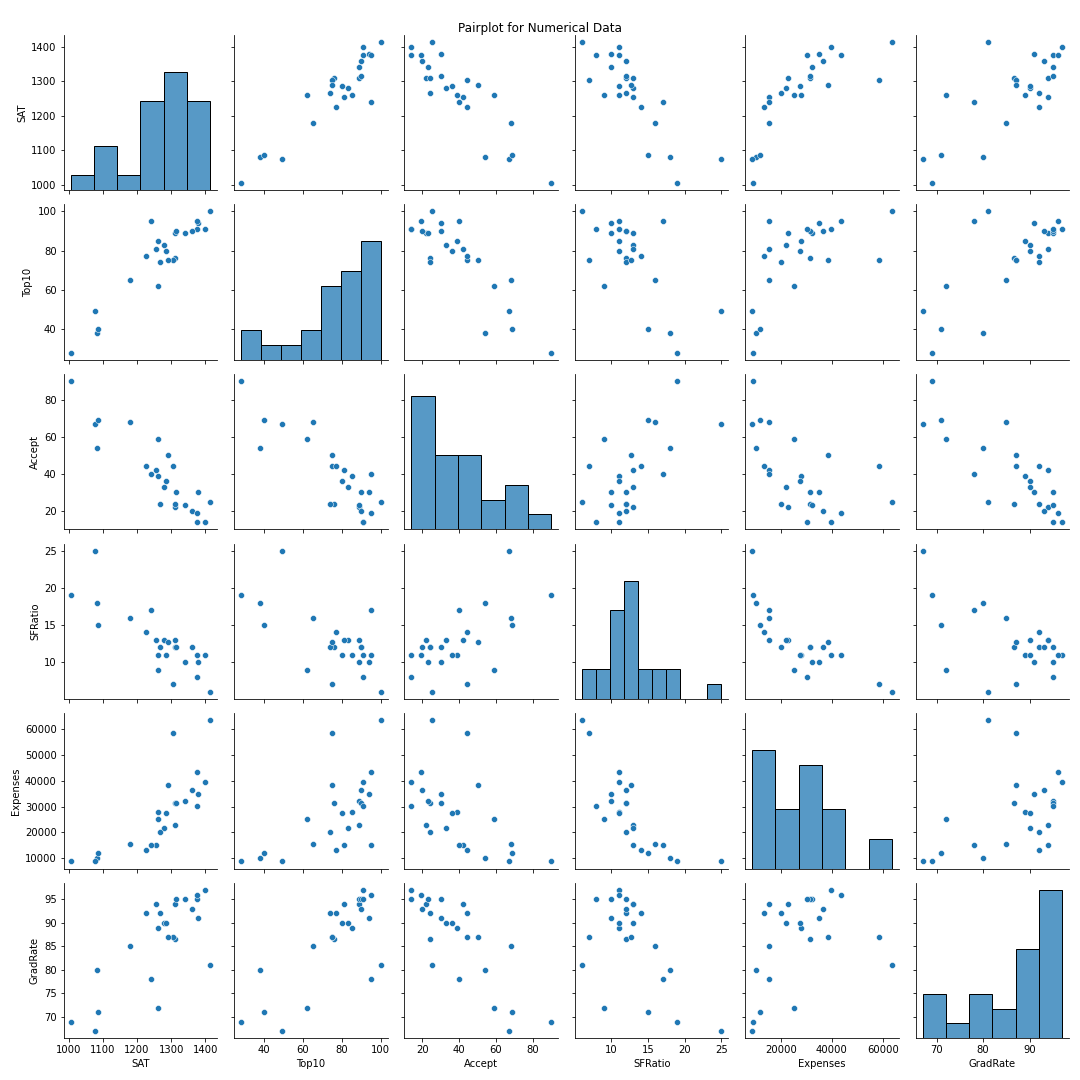
plt.suptitle('Pairplot for Numerical Data')

plt.tight\_layout()

plt.show()

**Output:**





B) Plot multivariate graphs for both numerical and categorical columns ?

# Boxplot for numerical data grouped by category

sns.boxplot(x='State', y='SAT', data=df3)

plt.title('Boxplot of Numeric1 by Category')

plt.show()

# Scatter plot comparing two numerical columns with categorical differentiation

sns.scatterplot(x='SAT', y='Top10', hue='State', data=df3, palette='Set1')

plt.legend(bbox\_to\_anchor=(1.05, 1.1), loc='upper left')

plt.title('Scatter plot of SAT vs Top10 with State differentiation')

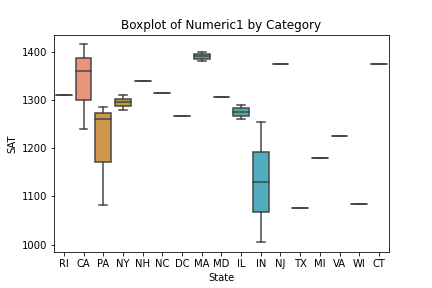
plt.xlabel('SAT')

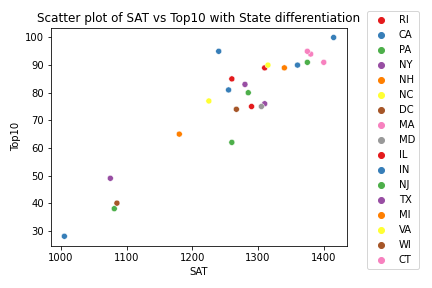
plt.ylabel('Top10')

plt.tight\_layout()

plt.show()

**Output:**





C) What does it mean when a correlation value says 1? When it is negative? When it is zero?

* When a correlation value is 1, it indicates a perfect positive correlation between two variables. This means that as one variable increases, the other variable also increases in a linear manner.
* When a correlation value is negative, it indicates a perfect negative correlation between two variables. This means that as one variable increases, the other variable decreases in a linear manner.
* When a correlation value is zero, it indicates no linear relationship between the two variables. However, it's important to note that zero correlation does not necessarily mean there is no relationship between the variables; there could still be a nonlinear relationship present.

4. Plot Skewness & Probability distribution for each column of **marks data.** (Hist, box, density)

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

from scipy.stats import skew, norm

df\_marks = pd.read\_csv(r"C:\Users\Lenovo\Downloads\Marks Data.csv")

# Plotting Skewness & Probability distribution for each column

for column in df\_marks.columns:

plt.figure(figsize=(12, 6))

# Histogram

plt.subplot(2, 2, 1)

sns.histplot(df\_marks[column], kde=True, bins=10, color='skyblue')

plt.title(f'Histogram of {column}')

# Box plot

plt.subplot(2, 2, 2)

sns.boxplot(x=df\_marks[column], color='lightgreen')

plt.title(f'Box plot of {column}')

# Density plot

plt.subplot(2, 1, 2)

sns.kdeplot(df\_marks[column], shade=True, color='salmon')

plt.title(f'Density plot of {column}')

# Calculate skewness

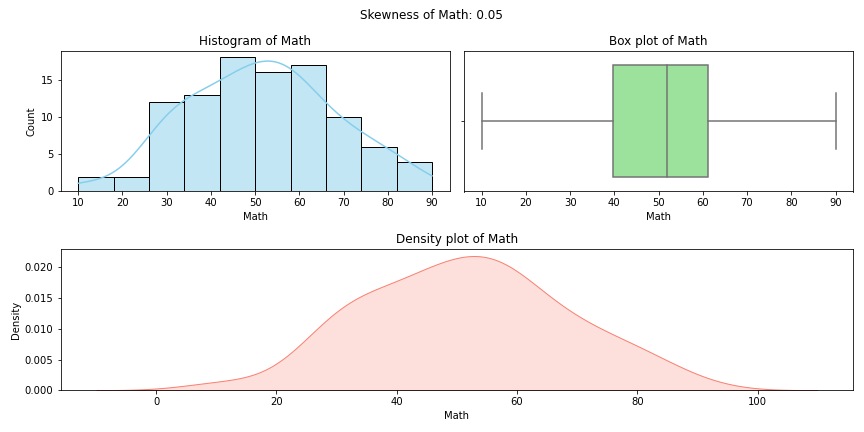
skewness = skew(df\_marks[column])

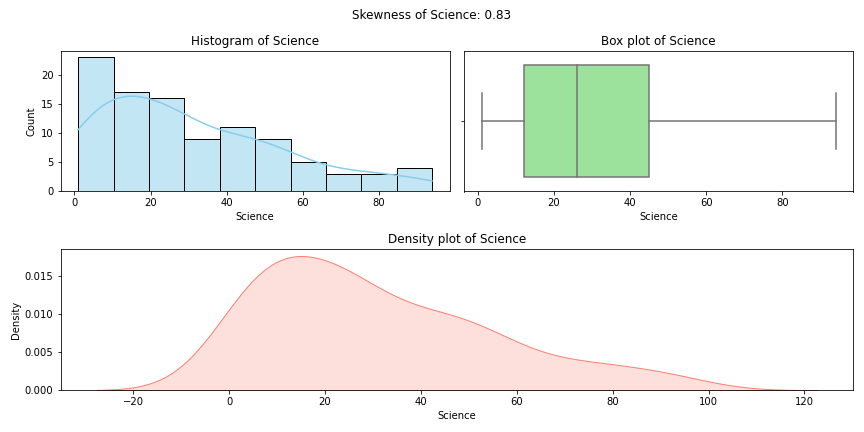
plt.suptitle(f'Skewness of {column}: {skewness:.2f}')

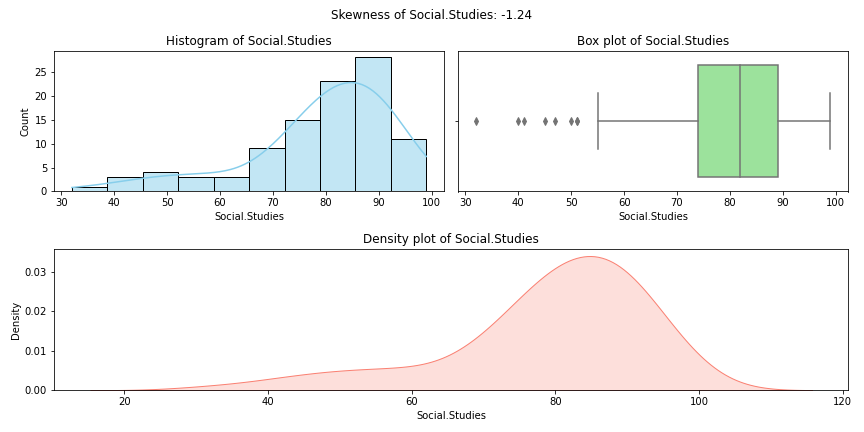
plt.tight\_layout()

plt.show()

**Output:**







1. What is normally distributed and What will be the relationship between mean, median & mode ?

Normally distributed data has a symmetric bell-shaped curve. In a normal distribution, mean = median = mode.

1. Which data variables are positively skewed and What will be the relationship between mean, median & mode

Positively skewed data has a tail extending to the right. In positively skewed distributions, mean > median > mode.

1. What are negatively skewed/distributed and What will be the relationship between mean, median & mode

Negatively skewed data has a tail extending to the left. In negatively skewed distributions, mean < median < mode.

1. What are the distinctive differences between skewness and distribution?

Skewness measures the asymmetry of the distribution, indicating whether it is positively or negatively skewed.

Distribution refers to the shape of the data, such as normal, uniform, skewed, kurtosis etc.