ASSIGNMENT 1

BASIC STATISTICS LEVEL 1

1. **Libraries imported:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sb

sb.set()

Matplotlib.pyplot is the library we use to get the visualizations required

1. **Loading the dataset & calling it:**

a=pd.read\_csv("/content/drive/MyDrive/Assignments/basic stats 1/sales\_data\_with\_discounts.csv")

a.head()

A screenshot of a computer screen

Description automatically generated

Here a is the dataframe and a.head() gives the first 5 entries of the df and a.tail() gives you the last 5 entries of the df

1. **Numerical columns of the dataset:**

b=a.select\_dtypes(include=np.number)

b.head()

A screenshot of a graph

Description automatically generated

1. **Calculating Mean, Median and Mode for these numerical columns :**

A screenshot of a computer screen

Description automatically generated

a.describe() is the function we call in here to find out the statistical values for the numerical features in the given datasets that also include finding the count, mean,standard deviation , min value, max value, and the quartile values

but this doesn’t give the mode so we need to find the mode separately

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

**5) Plotting histogram for all these numerical values:**

a.hist(figsize=(20,10))

plt.show()

A graph of different sizes and shapes

Description automatically generated with medium confidence

**6) Create boxplots for numerical variables to identify outliers and the interquartile range:**

plt.subplot(2,3,1 )

plt.boxplot(a["Volume"])

plt.subplot(2,3,2)

plt.boxplot(a["Avg Price"])

plt.subplot(2,3,3)

plt.boxplot(a["Total Sales Value"])

plt.subplot(2,3,4)

plt.boxplot(a["Discount Rate (%)"])

plt.subplot(2,3,5)

plt.boxplot(a["Discount Amount"])

plt.subplot(2,3,6)

plt.boxplot(a["Net Sales Value"])

plt.show()

here using subplot method, plotted all the box plots for the numerical values at once

A graph of numbers and a number of objects

Description automatically generated with medium confidence

**7) Finding out the categorical data in the given dataset:**

a.select\_dtypes(["object"])

A screenshot of a calendar

Description automatically generated

**8) Plotting Bar-Plot for all the categorical data:**

plt.figure(figsize=(20,20))

plt.subplot(2,3,1)

a['Day'].value\_counts().plot(kind='bar')

plt.subplot(2,3,2)

a['SKU'].value\_counts().plot(kind='bar')

plt.subplot(2,3,3)

a['City'].value\_counts().plot(kind='bar')

plt.subplot(2,3,4)

a['BU'].value\_counts().plot(kind='bar')

plt.subplot(2,3,5)

a['Brand'].value\_counts().plot(kind='bar')

plt.subplot(2,3,6)

a['Model'].value\_counts().plot(kind='bar')

A group of blue and white bars

Description automatically generated

9)

**Standardization of Numerical Variables:**

Standardization transforms/scaling data into a standard format, making it easier for computers to use and understand.

**z-score normalisation:**

Z score standardization uses mean and standard deviation from given data to standardize.

If z-score is 0 than data point is at mean.

If z-score is 1. Tells data point lies 1standard deviation more than mean

If z-score is -2. Tells data point lies 2standard deviation less than mean

Z = x -µ /σ

X -data point, µ - mean , σ – Standard deviation

A diagram of a function

Description automatically generated

A screenshot of a computer screen

Description automatically generated

so we saved the numerical values into b so the we can check the difference between the original and standardized values

the above values are the ones before standardization

#STANDARDIZATION

def get\_stand(x):

  return ((x-x.mean())/(x.std()))

here we defined a function to standardize the values and it is the implementation of the formula that is discussed above

A screenshot of a computer

Description automatically generated

These are the values after standardization, and to remember we use standardization to make the data scale free

**10) Creating Dummy Variables:**

Dummy variables are used to convert the categorical data into numerical data, or we can also say the dummy variables are the numerical representation of the categorical data

We use dummy variables so we can apply the machine learning algorithms to the converted numerical data

Let’s say the brand feature has certain different inputs in it but none of them are in numerical form and we cannot apply any function to it

So while converting them to dummy variables, we have different types of method and we precisely use “ONE HOT ENCODING” because we are making dummy variables in the input features side and “OHE” is mostly used for the input features transformation

While doing this encoding, the active entry will represented as 1 and the remaining or the inactive ones will be kept 0

A screenshot of a computer

Description automatically generated

So this is how it looks when we create the dummy variables for the categorical data