BUSINESS PROCESS OPTIMISATION LABORATORY

PROJECT REPORT

XYZ ANALYSIS

TEAM MEMBERS:

HANSA A S

GIRIDHARA VIJAY R A

ABSTRACT

XYZ analysis is a method commonly employed in inventory management and supply chain optimization to categorize items based on their consumption patterns. The project aims to develop a user-friendly graphical user interface tool for conducting XYZ analysis using Python programming language.

The Python code will then perform the XYZ analysis calculations, which involve categorizing items into three main groups: X (items with high consumption value and high demand frequency), Y (items with moderate consumption value and demand frequency), and Z (items with low consumption value and demand frequency).

The GUI will display the categorized items in a visually appealing format, such as tables or charts, allowing users to easily identify and prioritize items in their inventory.

By developing a Python-based GUI tool for XYZ analysis, this project aims to streamline and automate the inventory management process, enabling businesses to optimize their stocking decisions, reduce costs, and improve overall operational efficiency. The combination of Python's analytical capabilities and a user-friendly GUI interface will empower users to make informed decisions based on data-driven insights.

Overall, this project will contribute to the field of inventory management by providing a practical and accessible solution for conducting XYZ analysis, enabling businesses to better manage their inventory and improve supply chain performance.

**ALGORITHM:**

1. Import the required libraries: openpyxl, streamlit, pandas, and matplotlib.pyplot.
2. Define a function named findRow that takes the sheet object as input and iterates through the rows to find the first non-empty row. Return the row number incremented by 1.
3. Define a function named annualSum that takes the sheet object and row number as input. Iterate through the columns to find the first numeric value. Calculate the sum of values in the row until an empty cell is encountered or until column 13 is reached. Return the sum.
4. Define a function named stdev that takes the sheet object and row number as input. Use the annualSum function to calculate the sum of values in the row. Iterate through the columns to find the first numeric value. Calculate the deviation by subtracting each value from the sum and squaring the result. Sum up all the squared deviations until an empty cell is encountered or until column 13 is reached. Finally, calculate the standard deviation by dividing the deviation by the number of columns minus 2 and taking the square root. Return the standard deviation.
5. Define the main function XYZanal. Set the path to the Excel file containing the dataset. Load the workbook and select the active sheet.
6. Find the starting row for the analysis by calling the findRow function
7. Initialize variables: xyzanalysis as an empty dictionary, total as 0.
8. Iterate through the rows starting from the starting row. For each row:

* Retrieve the item code.
* Calculate the standard deviation using the stdev function.
* Update the total by adding the current standard deviation.
* Store the standard deviation in the xyzanalysis dictionary with the item code as the key.

1. Normalize the standard deviations and sort the xyzanalysis dictionary in descending order.
2. Calculate the cumulative percentage and assign categories ('X', 'Y', or 'Z') based on the cumulative percentage for each item in the xyzanalysis dictionary.
3. Use streamlit to create a container with two columns.
4. In the first column:

* Create a pie chart using matplotlib.pyplot to visualize the distribution of categories ('X', 'Y', 'Z') and their respective sizes.
* Set the labels and sizes for the pie chart.
* Display the pie chart using st.pyplot.

1. In the second column:

* Create a bar chart using pandas to visualize the counts of each category ('X', 'Y', 'Z').
* Create a dataframe with the category labels and counts.
* Display the bar chart using st.bar\_chart.

1. Create dataframes for the count of each category and the xyzanalysis dictionary.
2. Display the count dataframe and the first 10 rows of the xyzanalysis dataframe using st.write.
3. Call the XYZanal function to execute the XYZ analysis.

**CODE:**

import openpyxl

import streamlit as st

import pandas as pd

import matplotlib.pyplot as plt

#find the first row

def findRow(sheet\_obj):

r = 1

while(sheet\_obj.cell(row=r,column=1).value==None):

r += 1

return r+1

#To find the sum of each row

def annualSum(sheet\_obj, r):

col = 2

while(str(sheet\_obj.cell(row=r, column=col).value).isdigit() == False):

col += 1

sum = 0

while(sheet\_obj.cell(row=r, column=col).value != None):

sum += sheet\_obj.cell(row=r, column=col).value

col +=1

if (col > 13):

break

return sum

def stdev(sheet\_obj, r):

deviation, col = 0, 2

while(str(sheet\_obj.cell(row=r, column=col).value).isdigit() == False):

col += 1

sum = annualSum(sheet\_obj, r)

while(sheet\_obj.cell(row=r, column=col).value != None):

deviation += (sheet\_obj.cell(row=r, column=col).value - sum) \*\* 2

col +=1

if (col > 13):

break

stdeviation = (deviation / (col - 2)) \*\* (1//2)

return stdeviation

def XYZanal():

path = r"C:\Users\19295\Downloads\project dataset.xlsx"

wb\_obj = openpyxl.load\_workbook(path)

sheet\_obj = wb\_obj.active

r = findRow(sheet\_obj)

xyzanlaysis, total = {}, 0

while(sheet\_obj.cell(row=r, column=1).value != None):

code\_item = sheet\_obj.cell(row=r, column=1).value

stdeviation = stdev(sheet\_obj, r)

total += stdeviation

xyzanlaysis[code\_item] = stdeviation

r += 1

xyzanlaysis = {i[0]:[i[1], i[1]/total\*100] for i in sorted(xyzanlaysis.items(), key=lambda kv:(kv[1],kv[0]), reverse=True)}

cumulative\_percent, xCount, yCount, zCount = 0, 0, 0, 0

for key in xyzanlaysis.keys():

cumulative\_percent = xyzanlaysis[key][1]+cumulative\_percent

xyzanlaysis[key].append(cumulative\_percent)

if cumulative\_percent > 95:

xyzanlaysis[key].append('Z')

xCount += 1

elif cumulative\_percent > 80:

xyzanlaysis[key].append('Y')

yCount += 1

else:

xyzanlaysis[key].append('X')

zCount += 1

with st.container():

col1, col2 = st.columns(2)

with col1:

labels = ['X', 'Y', 'Z']

sizes = [xCount, yCount, zCount]

fig, ax = plt.subplots()

ax.pie(sizes, labels=labels, autopct='%1.1f%%')

ax.axis('equal')

st.pyplot(fig)

with col2:

barchart = pd.DataFrame(

sizes,

labels)

st.bar\_chart(barchart)

count = pd.DataFrame([xCount, yCount, zCount])

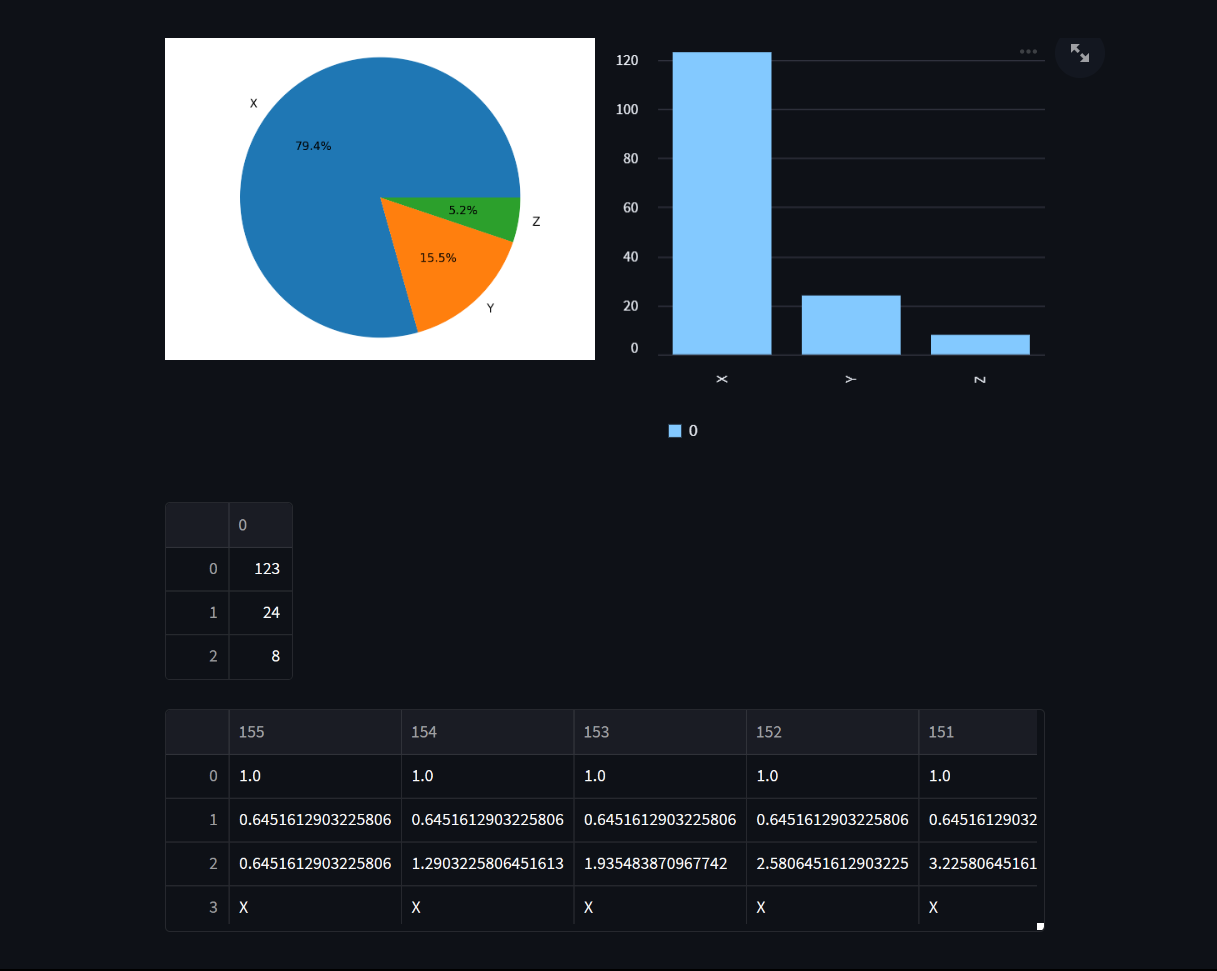
XYZanalysis = pd.DataFrame(xyzanlaysis)

st.write(count)

st.write(XYZanalysis.head(10))

XYZanal()

**OUTPUT:**



**INFERENCE:**

The XYZ analysis with the following distribution: X (79.4%), Y (15.5), and Z (5.2%) provides valuable insights into the categorization of items based on their consumption patterns and demand frequency. Here are some inferences that can be drawn from this distribution:

* X Category (79.4%): Items classified in the X category have a high consumption value and high demand frequency. These items are typically critical for the business and require careful monitoring and management. With the largest percentage among the three categories, it suggests that a significant portion of the inventory consists of high-priority items that contribute significantly to the overall consumption and demand.
* Y Category (15.5%): Items falling in the Y category have a moderate consumption value and demand frequency. These items may not have the same level of importance as those in the X category but are still significant for the business. The moderate percentage suggests that there is a notable portion of the inventory dedicated to these items, although their impact may be somewhat lower compared to the X category.
* Z Category (5.2%): Items categorized in the Z category have a low consumption value and demand frequency. These items are typically low-priority and may have a minimal impact on the overall inventory management. The percentage indicates that a relatively smaller portion of the inventory is allocated to these items.

Based on the distribution, it is crucial for businesses to focus their attention on the X category, which represents the majority of the inventory. Effective management of these high-priority items can help ensure smooth operations, meet customer demand, and prevent stockouts. While the Y category holds a smaller percentage, it should not be overlooked as it still contributes significantly to the overall inventory consumption. Adequate attention should be given to maintaining an appropriate stock level for Y-category items. The Z category, with the smallest percentage, requires lesser emphasis and may warrant periodic review to avoid any obsolete or excessive stock.

In conclusion, the XYZ analysis with the specified distribution highlights the varying importance and contribution of different item categories to the overall inventory. This information enables businesses to allocate resources, prioritize management efforts, and make informed decisions regarding stocking, procurement, and supply chain optimization to maximize efficiency and cost-effectiveness.