**Project Title - Company XYZ growth Analysis**

Company XYZ owns a supermarket chain across the country. Each major branch located in 3 cities across the country recorded sales information for 3 months, analysis was done using the 3 months information to help the company understand sales trends and determine its growth, as the rise of supermarket competition is seen to increase.

**PROJECT STEPS**

**Step 1 - Loading Datasets**

I was able to combine the dataset from each branch into one dataset for easy analysis. I wrote syntaxes that read multiple files from my current working directory and exported a CSV file after combining.

Codes used are shown below,

--Import Libraries using the below

import os

import glob

import pandas as pd

os.chdir(r"C:\Users\ACER\Documents\Data-Analysis-Project-main")

--Use glob to match the pattern "csv"

extension = 'csv'

path = (r"C:\Users\ACER\Documents\Data-Analysis-Project-main")

files = glob.glob(path + '/\*.csv')

files

All = []

for f in files:

temp\_df = pd.read\_csv(f)

All.append(temp\_df)

print(f'Successfully created dataframe for {f} with shape {temp\_df.shape}')

--Combine all the files in the generated list above and export to a CSV

df = pd.concat(All, axis=0)

--Export to csv

df.to\_csv(r'C:\Users\ACER\Documents\Data-Analysis-Project-main\All\_Branches.csv')

Read the CSV file

pd.read\_csv(r'C:\Users\ACER\Documents\Data-Analysis-Project-main\All\_Branches.csv')

**Step 2 - Data Exploration**

Exploration of the loaded dataset was done using some built-in Pandas function.

Libraries used are below-

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

%matplotlib inline

plt.style.use('fivethirtyeight')

import warnings

warnings.filterwarnings('ignore')

-- Use the head() method to view first few rows of the dataset

df.head()

-- Check the number of rows and columns present in the data using the shape attribute.

df.shape

-- Generate the names of the columns using the columns attribute.

df.columns

-- Use describe function to generate the statistical summary of the dataframe

df.describe

-- Use meaningful sentences to describe findings from the data statistical summary

The mean, median, standard deviation and sum were calculated for total column which provided insights in purchases for the 3 months.

average = df['Total'].mean()

print(average)

med = df['Total'].median()

print(med)

standard\_deviation = df['Total'].std()

print(standard\_deviation)

Sum = df['Total'].sum()

print(Sum)

-- Use of correct method to check for Missing values

df.isnull()

df.notna()

-- Check the information of the DataFrame using the info method.

df.info()

**Step 3 - Dealing with DateTime Features**

The below codes were written to achieve desired results.

-- Use to\_datetime() to convert the date column to datetime

df['Date']= pd.to\_datetime(df['Date'])

-- Check the datatype to confirm if it's in datetime

df['Date'].dtype

-- Accurate conversion of the time column & prints appropriate data type

df['Time']= pd.to\_datetime(df['Time'])

df['Time'].dtype

-- Accurate extraction of the Day, Month, Year & Hour features

df['day'] = df.Date.dt.day

df.head()

df['month'] = df.Date.dt.month

df.head()

df['year'] = df.Date.dt.year

df.head()

df['hour'] = df.Time.dt.hour

df.head()

-- The numbers of unique hours of sales in the supermarket are accurately determined.

-- Result that shows an array that contains the unique sales hours.

df['hour'].unique()

**Step 4 - Unique Values in Columns**

The below codes were written to achieve desired results.

-- Appropriate method to generate the unique values in the categorical columns (apart from the example - Branch column).

df['City'].unique().tolist()

df['Customer type'].unique().tolist()

df['Gender'].unique().tolist()

df['Payment'].unique().tolist()

df['Product line'].unique().tolist()

-- Generated the count figure of the categorical values using the value\_counts() method.

categorical\_columns = [col for col in df.columns if df[col].dtype == "object" ]

categorical\_columns

--For the categorical columns above, generate the count figure of the values

df['City'].value\_counts()

df['Payment'].value\_counts()

df['Product line'].value\_counts()

df['Gender'].value\_counts()

**Step 5 - Aggregation with GroupBy**

The below codes were written to achieve desired results.

-- A groupby object with the "City Column", and aggregation function of sum and mean.

df.groupby('City')['gross income'].sum()

df.groupby('City')['gross income'].mean()

-- A table that shows the gross income of each city, and determines the city with the highest total gross income.

df.groupby('City')['gross income'].max()

df.groupby('City')['gross income'].min()

-- Optional - Use of appropriate methods & descriptions to explore other columns such as Unit Price, Quantity.

df.groupby('City')['Unit price'].max()

df.groupby('City')['Quantity'].max()

df.groupby('City')['Quantity'].min()

df.groupby('Gender')['Quantity'].max()

df.groupby('Gender')['Quantity'].min()

df.groupby('Gender')['Unit price'].max()

df.groupby('Gender')['Unit price'].min()

**Step 6 - Data Visualization**

The below codes were written to achieve desired results.

-- Appropriate use of countplot to determine the branch with the highest sales record.

sns.set\_theme(style="whitegrid")

ax = sns.countplot(x="Branch", data=df).set(title='Barnch with Highest sale')

-- Optional - Appropriate use of countplot to determine the most used payment method & city with the most sales.

sns.set\_theme(style="darkgrid")

ax = sns.countplot(x="Payment", data=df,

facecolor=(0, 0, 0, 0),

linewidth=5,

edgecolor=sns.color\_palette("dark", 3)).set(title='Payment Method')

sns.set\_theme(style="white")

ax = sns.countplot(x="City", data=df,

facecolor=(0, 0, 0, 0),

linewidth=4,

edgecolor=sns.color\_palette("dark", 3)).set(title='City with Highest sales')

-- Appropriate use of countplot to determine the highest & lowest sold product line.

sns.set\_theme(style="whitegrid")

ax = sns.countplot(y="Product line", data=df,

facecolor=(0, 0, 0, 0),

linewidth=3,

edgecolor=sns.color\_palette("dark", 6)).set(title='Highest Sold Product line')

-- Payment channel used by most customers to pay for each product line.

sns.set\_theme(style="whitegrid")

ax = sns.countplot(y="Product line",hue='Payment', data=df,

linewidth=3,

edgecolor=sns.color\_palette("dark",6)).set(title='most used payment channel')

--Determine the Payment channel for each branch.

sns.set\_theme(style="ticks")

ax = sns.countplot(x="Payment",hue='Branch', data=df,

linewidth=3,

edgecolor=sns.color\_palette("dark",6)).set(title='branch payment channel')

--Determine the branch with the lowest rating.

sns.set\_theme(style="darkgrid")

ax = sns.boxplot(x="Branch", y="Rating", data=df,linewidth=1).set(title='Branch Rating')

--Determine purchase by gender.

g = sns.catplot(x="Product line", y="Quantity", hue="Gender",

data=df, kind="bar",

height=4, aspect=4).set(title='Product Purchase By Gender A')

g = sns.catplot(x="Product line", y="Total", hue="Gender",

data=df, kind="bar",

height=4, aspect=4).set(title='Product Purchase By Gender B')

--Interaction of Unit price on the Quantity of goods purchased.

g = sns.catplot(x="Product line", y="Unit price",

data=df, kind="point",

height=4,aspect=4).set(title='Product line Vs Unit price')

g = sns.catplot(x="Product line", y="Quantity",

data=df, kind="point",

height=4,aspect=4).set(title='Product line Vs Quantity')

Step 7 - StandOut Section

The below codes were written to get more insights from the dataset.

--Determination of Branch purchases for all product line.

g = sns.catplot(x="Product line", y="Quantity", hue='Branch',

data=df, kind="bar",

height=4,aspect=4).set(title='Branch Purchase')

--Determination of gender purchase for all product line.

g = sns.catplot(x="Product line", y="Quantity", hue='Gender',

data=df, kind="bar",

height=4,aspect=4).set(title='Gender Purchase')

**INSIGHTS**

The following are insights from the dataset;

1) The data sheet for each city is clean and does not require cleaning up.

2) Total amount sold by the company for the past 3 months is N116,268,029.64000002.

3) Lagos has the highest sales at 340, while Port Harcourt has the lowest sales at 328.

4) Epay payment was the most used at 345, while Card payment was the least at 311.

5) Fashion accessories sold more at 178, while Health and beauty had the least sales at 152.

6) Port Harcourt city has the highest gross income at N1,895,463.54, while Abuja has the least at N1,820,531.52.

7) Branch A recorded the highest sales, while Branch C recorded the lowest.

8) Cash payment was used mostly for electronic accessories purchase.

9) Dominant mode of payment in Branch A is Epay while for Branch B its card payment and Branch C its cash.

10) Branch A and C have the highest rating while Branch B has the lowest.

11) Females purchase more of home and lifestyle products, while males purchase more of health and beauty products.

12) Product with the highest purchase is home & lifestyle done by females, followed by food & beverages.

13) Electronic accessories and Home/Lifestyle products sold the highest quantity

**FUTURE WORK**

The below can be included in the project

1) Task involving treatment of null values.

2) More usage of visualisation techniques.

**STANDOUT SECTION**

I was able to determine the below

1) The branch with the highest sales for each product line using a catplot.

2) The gender with the highest purchase for each product line using a catplot.

(See jupyter file for more details and insights)