```
{
 "cells": [
  {
   "cell_type": "markdown",
   "id": "c6269683",
   "metadata": {},
   "source": [
    "# Global Sales Data Analytics"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "5209ddf6",
   "metadata": {},
   "source": [
    "<h4> IMPORTING REQUIRED LIBRARIES\n",
    "  "
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 2,
   "id": "0b03c99b",
   "metadata": {},
   "outputs": [],
   "source": [
    "#Data Manipulation\n",
    "import pandas as pd\n",
    "#Data Visualization \n",
    "import matplotlib.pyplot as plt\n",
    "%matplotlib inline\n",
    "import seaborn as sns\n",
    "import numpy as np"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "83b5347c",
   "metadata": {},
   "source": [
    "#### IMPORTING THE DATASET"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 3,
   "id": "96bd42b9",
   "metadata": {},
   "outputs": [],
   "source": [
    "df = pd.read_excel('Global_Superstore.xlsx')"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 4,
   "id": "c2ce2bd6",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/html": [
       "<div>\n",
       "<style scoped>\n",
"    .dataframe tbody tr th:only-of-type {\n",
"        vertical-align: middle;\n",
"    }\n",
"\n",
"    .dataframe tbody tr th {\n",
"        vertical-align: top;\n",
"    }\n",
"\n",
"    .dataframe thead th {\n",
"        text-align: right;\n",
"    }\n",
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",
"      <th></th>\n",
"      <th>Row ID</th>\n",
"      <th>Order ID</th>\n",
"      <th>Order Date</th>\n",
"      <th>Ship Date</th>\n",
"      <th>Ship Mode</th>\n",
"      <th>Customer ID</th>\n",
"      <th>Customer Name</th>\n",
"      <th>Segment</th>\n",
"      <th>City</th>\n",
"      <th>State</th>\n",
"      <th>...</th>\n",
"      <th>Product ID</th>\n",
"      <th>Category</th>\n",
"      <th>Sub-Category</th>\n",
"      <th>Product Name</th>\n",
"      <th>Sales</th>\n",
"      <th>Quantity</th>\n",
"      <th>Discount</th>\n",
"      <th>Profit</th>\n",
"      <th>Shipping Cost</th>\n",
"      <th>Order Priority</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>32298</td>\n",
"      <td>CA-2012-124891</td>\n",
"      <td>31-07-2012</td>\n",
"      <td>31-07-2012</td>\n",
"      <td>Same Day</td>\n",
"      <td>RH-19495</td>\n",
"      <td>Rick Hansen</td>\n",
"      <td>Consumer</td>\n",
"      <td>New York City</td>\n",
"      <td>New York</td>\n",
"      <td>...</td>\n",
"      <td>TEC-AC-10003033</td>\n",
"      <td>Technology</td>\n",
"      <td>Accessories</td>\n",
"      <td>Plantronics CS510 - Over-the-Head monaural Wir...</td>\n",
"      <td>2309.650</td>\n",
"      <td>7</td>\n",
"      <td>0.0</td>\n",
"      <td>762.1845</td>\n",
"      <td>933.57</td>\n",
"      <td>Critical</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>26341</td>\n",
"      <td>IN-2013-77878</td>\n",
"      <td>05-02-2013</td>\n",
"      <td>07-02-2013</td>\n",
"      <td>Second Class</td>\n",
"      <td>JR-16210</td>\n",
"      <td>Justin Ritter</td>\n",
"      <td>Corporate</td>\n",
"      <td>Wollongong</td>\n",
"      <td>New South Wales</td>\n",
"      <td>...</td>\n",
"      <td>FUR-CH-10003950</td>\n",
"      <td>Furniture</td>\n",
"      <td>Chairs</td>\n",
"      <td>Novimex Executive Leather Armchair, Black</td>\n",
"      <td>3709.395</td>\n",
"      <td>9</td>\n",
"      <td>0.1</td>\n",
"      <td>-288.7650</td>\n",
"      <td>923.63</td>\n",
"      <td>Critical</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>25330</td>\n",
"      <td>IN-2013-71249</td>\n",
"      <td>17-10-2013</td>\n",
"      <td>18-10-2013</td>\n",
"      <td>First Class</td>\n",
"      <td>CR-12730</td>\n",
"      <td>Craig Reiter</td>\n",
"      <td>Consumer</td>\n",
"      <td>Brisbane</td>\n",
"      <td>Queensland</td>\n",
"      <td>...</td>\n",
"      <td>TEC-PH-10004664</td>\n",
"      <td>Technology</td>\n",
"      <td>Phones</td>\n",
"      <td>Nokia Smart Phone, with Caller ID</td>\n",
"      <td>5175.171</td>\n",
"      <td>9</td>\n",
"      <td>0.1</td>\n",
"      <td>919.9710</td>\n",
"      <td>915.49</td>\n",
"      <td>Medium</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>13524</td>\n",
"      <td>ES-2013-1579342</td>\n",
"      <td>28-01-2013</td>\n",
"      <td>30-01-2013</td>\n",
"      <td>First Class</td>\n",
"      <td>KM-16375</td>\n",
"      <td>Katherine Murray</td>\n",
"      <td>Home Office</td>\n",
"      <td>Berlin</td>\n",
"      <td>Berlin</td>\n",
"      <td>...</td>\n",
"      <td>TEC-PH-10004583</td>\n",
"      <td>Technology</td>\n",
"      <td>Phones</td>\n",
"      <td>Motorola Smart Phone, Cordless</td>\n",
"      <td>2892.510</td>\n",
     "      <td>5</td>\n",
     "      <td>0.1</td>\n",
     "      <td>-96.5400</td>\n",
     "      <td>910.16</td>\n",
     "      <td>Medium</td>\n",
     "    </tr>\n",
     "    <tr>\n",
     "      <th>4</th>\n",
     "      <td>47221</td>\n",
     "      <td>SG-2013-4320</td>\n",
     "      <td>05-11-2013</td>\n",
     "      <td>06-11-2013</td>\n",
     "      <td>Same Day</td>\n",
     "      <td>RH-9495</td>\n",
     "      <td>Rick Hansen</td>\n",
     "      <td>Consumer</td>\n",
     "      <td>Dakar</td>\n",
     "      <td>Dakar</td>\n",
     "      <td>...</td>\n",
     "      <td>TEC-SHA-10000501</td>\n",
     "      <td>Technology</td>\n",
     "      <td>Copiers</td>\n",
     "      <td>Sharp Wireless Fax, High-Speed</td>\n",
     "      <td>2832.960</td>\n",
     "      <td>8</td>\n",
     "      <td>0.0</td>\n",
     "      <td>311.5200</td>\n",
     "      <td>903.04</td>\n",
     "      <td>Critical</td>\n",
     "    </tr>\n",
     "  </tbody>\n",
     "</table>\n",
     "<p>5 rows × 23 columns</p>\n",
     "</div>"
    ],
    "text/plain": [
     "   Row ID         Order ID  Order Date   Ship Date     Ship Mode Customer ID  \\\n",
     "0   32298   CA-2012-124891  31-07-2012  31-07-2012      Same Day    RH-19495   \n",
     "1   26341    IN-2013-77878  05-02-2013  07-02-2013  Second Class    JR-16210   \n",
     "2   25330    IN-2013-71249  17-10-2013  18-10-2013   First Class    CR-12730   \n",
     "3   13524  ES-2013-1579342  28-01-2013  30-01-2013   First Class    KM-16375   \n",
     "4   47221     SG-2013-4320  05-11-2013  06-11-2013      Same Day     RH-9495   \n",
     "\n",
     "      Customer Name      Segment           City            State  ...  \\\n",
     "0      Rick Hansen     Consumer  New York City         New York  ...   \n",
     "1    Justin Ritter    Corporate     Wollongong  New South Wales  ...   \n",
     "2     Craig Reiter     Consumer       Brisbane       Queensland  ...   \n",
     "3  Katherine Murray  Home Office         Berlin           Berlin  ...   \n",
     "4      Rick Hansen     Consumer          Dakar            Dakar  ...   \n",
     "\n",
     "         Product ID    Category Sub-Category  \\\n",
     "0   TEC-AC-10003033  Technology  Accessories   \n",
     "1   FUR-CH-10003950   Furniture       Chairs   \n",
     "2   TEC-PH-10004664  Technology       Phones   \n",
     "3   TEC-PH-10004583  Technology       Phones   \n",
     "4  TEC-SHA-10000501  Technology      Copiers   \n",
     "\n",
     "                                        Product Name     Sales Quantity  \\\n",
     "0  Plantronics CS510 - Over-the-Head monaural Wir...  2309.650        7   \n",
     "1          Novimex Executive Leather Armchair, Black  3709.395        9   \n",
     "2                  Nokia Smart Phone, with Caller ID  5175.171        9   \n",
     "3                     Motorola Smart Phone, Cordless  2892.510        5   \n",
     "4                     Sharp Wireless Fax, High-Speed  2832.960        8   \n",
     "\n",
     "  Discount    Profit  Shipping Cost  Order Priority  \n",
       "0      0.0  762.1845         933.57        Critical  \n",
       "1      0.1 -288.7650         923.63        Critical  \n",
       "2      0.1  919.9710         915.49          Medium  \n",
       "3      0.1  -96.5400         910.16          Medium  \n",
       "4      0.0  311.5200         903.04        Critical  \n",
       "\n",
       "[5 rows x 23 columns]"
      ]
     },
     "execution_count": 4,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "df.head()"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "677e84e0",
   "metadata": {},
   "source": [
    "#### CUSTOMER ANALYSIS"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "c9a7b9f9",
   "metadata": {},
   "source": [
    "Profile the customers based on their frequency of purchase - calculate frequency of
purchase for each customer and plot a histogram to get the threshold for Low/Mid/High
frequency customers\n",
    "\n",
    "Does the high frequent customers are contributing more revenue\n",
    "\n",
    "Are they also profitable - what is the profit margin across the buckets\n",
    "\n",
    "Which customer segment is most profitable in each year ( there is a column called
customer segment)\n",
    "\n",
    "How the customers are distributed across the countries - pie chart\n",
    "\n",
    "Write a function to split the global store data into different unique data frames based
on the unique values in country column (Means, we should have one data frame for one country
as function output )"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 5,
   "id": "10f708c6",
   "metadata": {},
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "<class 'pandas.core.frame.DataFrame'>\n",
      "RangeIndex: 51290 entries, 0 to 51289\n",
      "Data columns (total 23 columns):\n",
      " #   Column          Non-Null Count  Dtype  \n",
      "---  ------          --------------  -----  \n",
      " 0   Row ID          51290 non-null  int64  \n",
      " 1   Order ID        51290 non-null  object \n",
      " 2   Order Date      51290 non-null  object \n",
      " 3   Ship Date       51290 non-null  object \n",
      " 4   Ship Mode       51290 non-null  object \n",
      " 5   Customer ID     51290 non-null  object \n",
      " 6   Customer Name   51290 non-null  object \n",
      " 7   Segment         51290 non-null  object \n",
      " 8   City            51290 non-null  object \n",
      " 9   State           51290 non-null  object \n",
      " 10  Country         51290 non-null  object \n",
      " 11  Market          51290 non-null  object \n",
      " 12  Region          51290 non-null  object \n",
      " 13  Product ID      51290 non-null  object \n",
      " 14  Category        51290 non-null  object \n",
      " 15  Sub-Category    51290 non-null  object \n",
      " 16  Product Name    51290 non-null  object \n",
      " 17  Sales           51290 non-null  float64\n",
      " 18  Quantity        51290 non-null  int64  \n",
      " 19  Discount        51290 non-null  float64\n",
      " 20  Profit          51290 non-null  float64\n",
      " 21  Shipping Cost   51290 non-null  float64\n",
      " 22  Order Priority  51290 non-null  object \n",
      "dtypes: float64(4), int64(2), object(17)\n",
      "memory usage: 9.0+ MB\n"
     ]
    }
   ],
   "source": [
    "df.info()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 6,
   "id": "ffb3ec19",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "(51290, 23)"
      ]
     },
     "execution_count": 6,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "df.shape"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 7,
   "id": "9d0bdd3b",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/html": [
       "<div>\n",
       "<style scoped>\n",
       "    .dataframe tbody tr th:only-of-type {\n",
       "        vertical-align: middle;\n",
"    }\n",
"\n",
"    .dataframe tbody tr th {\n",
"        vertical-align: top;\n",
"    }\n",
"\n",
"    .dataframe thead th {\n",
"        text-align: right;\n",
"    }\n",
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",
"      <th></th>\n",
"      <th>Row ID</th>\n",
"      <th>Sales</th>\n",
"      <th>Quantity</th>\n",
"      <th>Discount</th>\n",
"      <th>Profit</th>\n",
"      <th>Shipping Cost</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>count</th>\n",
"      <td>51290.00000</td>\n",
"      <td>51290.000000</td>\n",
"      <td>51290.000000</td>\n",
"      <td>51290.000000</td>\n",
"      <td>51290.000000</td>\n",
"      <td>51290.000000</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>mean</th>\n",
"      <td>25645.50000</td>\n",
"      <td>246.490581</td>\n",
"      <td>3.476545</td>\n",
"      <td>0.142908</td>\n",
"      <td>28.610982</td>\n",
"      <td>26.375915</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>std</th>\n",
"      <td>14806.29199</td>\n",
"      <td>487.565361</td>\n",
"      <td>2.278766</td>\n",
"      <td>0.212280</td>\n",
"      <td>174.340972</td>\n",
"      <td>57.296804</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>min</th>\n",
"      <td>1.00000</td>\n",
"      <td>0.444000</td>\n",
"      <td>1.000000</td>\n",
"      <td>0.000000</td>\n",
"      <td>-6599.978000</td>\n",
"      <td>0.000000</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>25%</th>\n",
"      <td>12823.25000</td>\n",
"      <td>30.758625</td>\n",
"      <td>2.000000</td>\n",
"      <td>0.000000</td>\n",
       "      <td>0.000000</td>\n",
       "      <td>2.610000</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>50%</th>\n",
       "      <td>25645.50000</td>\n",
       "      <td>85.053000</td>\n",
       "      <td>3.000000</td>\n",
       "      <td>0.000000</td>\n",
       "      <td>9.240000</td>\n",
       "      <td>7.790000</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>75%</th>\n",
       "      <td>38467.75000</td>\n",
       "      <td>251.053200</td>\n",
       "      <td>5.000000</td>\n",
       "      <td>0.200000</td>\n",
       "      <td>36.810000</td>\n",
       "      <td>24.450000</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>max</th>\n",
       "      <td>51290.00000</td>\n",
       "      <td>22638.480000</td>\n",
       "      <td>14.000000</td>\n",
       "      <td>0.850000</td>\n",
       "      <td>8399.976000</td>\n",
       "      <td>933.570000</td>\n",
       "    </tr>\n",
       "  </tbody>\n",
       "</table>\n",
       "</div>"
      ],
      "text/plain": [
       "            Row ID         Sales      Quantity      Discount        Profit  \\\n",
       "count  51290.00000  51290.000000  51290.000000  51290.000000  51290.000000   \n",
       "mean   25645.50000    246.490581      3.476545      0.142908     28.610982   \n",
       "std    14806.29199    487.565361      2.278766      0.212280    174.340972   \n",
       "min        1.00000      0.444000      1.000000      0.000000  -6599.978000   \n",
       "25%    12823.25000     30.758625      2.000000      0.000000      0.000000   \n",
       "50%    25645.50000     85.053000      3.000000      0.000000      9.240000   \n",
       "75%    38467.75000    251.053200      5.000000      0.200000     36.810000   \n",
       "max    51290.00000  22638.480000     14.000000      0.850000   8399.976000   \n",
       "\n",
       "       Shipping Cost  \n",
       "count   51290.000000  \n",
       "mean       26.375915  \n",
       "std        57.296804  \n",
       "min         0.000000  \n",
       "25%         2.610000  \n",
       "50%         7.790000  \n",
       "75%        24.450000  \n",
       "max       933.570000  "
      ]
     },
     "execution_count": 7,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "df.describe()\n",
    "# checks the negative trend in profit and the decrease in discount of range"
   ]
```

```
  },
  {
   "cell_type": "code",
   "execution_count": 8,
   "id": "3b8181d4",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "Row ID            0\n",
       "Order ID          0\n",
       "Order Date        0\n",
       "Ship Date         0\n",
       "Ship Mode         0\n",
       "Customer ID       0\n",
       "Customer Name     0\n",
       "Segment           0\n",
       "City              0\n",
       "State             0\n",
       "Country           0\n",
       "Market            0\n",
       "Region            0\n",
       "Product ID        0\n",
       "Category          0\n",
       "Sub-Category      0\n",
       "Product Name      0\n",
       "Sales             0\n",
       "Quantity          0\n",
       "Discount          0\n",
       "Profit            0\n",
       "Shipping Cost     0\n",
       "Order Priority    0\n",
       "dtype: int64"
      ]
     },
     "execution_count": 8,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "df.isnull().sum()\n",
    "# checks if there are null entries"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "a4e10014",
   "metadata": {},
   "source": [
    "Checking how many of them are categorical and how many unique values each categorical columns have,"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 9,
   "id": "e58955ed",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "Index(['Order ID', 'Order Date', 'Ship Date', 'Ship Mode', 'Customer ID',\n",
       "       'Customer Name', 'Segment', 'City', 'State', 'Country', 'Market',\n",
       "       'Region', 'Product ID', 'Category', 'Sub-Category', 'Product Name',\n",
       "       'Order Priority'],\n",
       "      dtype='object')"
      ]
     },
     "execution_count": 9,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "cat_cols = df.select_dtypes(exclude= ['int64', 'float64']).columns\n",
    "cat_cols"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 10,
   "id": "629b5013",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/html": [
       "<div>\n",
       "<style scoped>\n",
       "    .dataframe tbody tr th:only-of-type {\n",
       "        vertical-align: middle;\n",
       "    }\n",
       "\n",
       "    .dataframe tbody tr th {\n",
       "        vertical-align: top;\n",
       "    }\n",
       "\n",
       "    .dataframe thead th {\n",
       "        text-align: right;\n",
       "    }\n",
       "</style>\n",
       "<table border=\"1\" class=\"dataframe\">\n",
       "  <thead>\n",
       "    <tr style=\"text-align: right;\">\n",
       "      <th></th>\n",
       "      <th>Row ID</th>\n",
       "      <th>Sales</th>\n",
       "      <th>Quantity</th>\n",
       "      <th>Discount</th>\n",
       "      <th>Profit</th>\n",
       "      <th>Shipping Cost</th>\n",
       "    </tr>\n",
       "  </thead>\n",
       "  <tbody>\n",
       "    <tr>\n",
       "      <th>0</th>\n",
       "      <td>32298</td>\n",
       "      <td>2309.650</td>\n",
       "      <td>7</td>\n",
       "      <td>0.0</td>\n",
       "      <td>762.1845</td>\n",
       "      <td>933.57</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>1</th>\n",
       "      <td>26341</td>\n",
       "      <td>3709.395</td>\n",
       "      <td>9</td>\n",
       "      <td>0.1</td>\n",
       "      <td>-288.7650</td>\n",
       "      <td>923.63</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>2</th>\n",
       "      <td>25330</td>\n",
       "      <td>5175.171</td>\n",
       "      <td>9</td>\n",
       "      <td>0.1</td>\n",
       "      <td>919.9710</td>\n",
       "      <td>915.49</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>3</th>\n",
       "      <td>13524</td>\n",
       "      <td>2892.510</td>\n",
       "      <td>5</td>\n",
       "      <td>0.1</td>\n",
       "      <td>-96.5400</td>\n",
       "      <td>910.16</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>4</th>\n",
       "      <td>47221</td>\n",
       "      <td>2832.960</td>\n",
       "      <td>8</td>\n",
       "      <td>0.0</td>\n",
       "      <td>311.5200</td>\n",
       "      <td>903.04</td>\n",
       "    </tr>\n",
       "  </tbody>\n",
       "</table>\n",
       "</div>"
      ],
      "text/plain": [
       "   Row ID     Sales  Quantity  Discount    Profit  Shipping Cost\n",
       "0   32298  2309.650         7       0.0  762.1845         933.57\n",
       "1   26341  3709.395         9       0.1 -288.7650         923.63\n",
       "2   25330  5175.171         9       0.1  919.9710         915.49\n",
       "3   13524  2892.510         5       0.1  -96.5400         910.16\n",
       "4   47221  2832.960         8       0.0  311.5200         903.04"
      ]
     },
     "execution_count": 10,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "num_colms = df._get_numeric_data()\n",
    "num_colms.head()"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "9b67c396",
   "metadata": {},
   "source": [
    "###### 1. Profile the customers based on their frequency of purchase - calculate
frequency of purchase for each customer and plot a histogram to get the threshold for
Low/Mid/High frequency customers"
   ]
  },
{
 "cell_type": "code",
 "execution_count": 11,
 "id": "1a91c5b0",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "Row ID            51290\n",
     "Order ID          25035\n",
     "Order Date         1430\n",
     "Ship Date          1464\n",
     "Ship Mode             4\n",
     "Customer ID        1590\n",
     "Customer Name       795\n",
     "Segment               3\n",
     "City               3636\n",
     "State              1094\n",
     "Country             147\n",
     "Market                7\n",
     "Region               13\n",
     "Product ID        10292\n",
     "Category              3\n",
     "Sub-Category         17\n",
     "Product Name       3788\n",
     "Sales             22995\n",
     "Quantity             14\n",
     "Discount             27\n",
     "Profit            24575\n",
     "Shipping Cost     10037\n",
     "Order Priority        4\n",
     "dtype: int64"
    ]
   },
   "execution_count": 11,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "df.nunique()"
 ]
},
{
 "cell_type": "code",
 "execution_count": 12,
 "id": "48bcf2fa",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "Customer ID    51290\n",
     "Order ID       51290\n",
     "Order Date     51290\n",
     "Ship Date      51290\n",
     "Ship Mode      51290\n",
     "Country        51290\n",
     "dtype: int64"
    ]
   },
   "execution_count": 12,
   "metadata": {},
   "output_type": "execute_result"
```

```
    }
   ],
   "source": [
    "df_customer = df[['Customer ID','Order ID','Order Date', 'Ship Date', 'Ship
Mode','Country']]\n",
    "df_customer.count()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 13,
   "id": "5ce03486",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "Customer ID    25754\n",
       "Order ID       25754\n",
       "Order Date     25754\n",
       "Ship Date      25754\n",
       "Ship Mode      25754\n",
       "Country        25754\n",
       "dtype: int64"
      ]
     },
     "execution_count": 13,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "df_customer1 = df[['Customer ID','Order ID','Order Date', 'Ship Date', 'Ship
Mode','Country']].drop_duplicates()\n",
    "df_customer1.count()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 14,
   "id": "55552d57",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/html": [
       "<div>\n",
       "<style scoped>\n",
       "    .dataframe tbody tr th:only-of-type {\n",
       "        vertical-align: middle;\n",
       "    }\n",
       "\n",
       "    .dataframe tbody tr th {\n",
       "        vertical-align: top;\n",
       "    }\n",
       "\n",
       "    .dataframe thead th {\n",
       "        text-align: right;\n",
       "    }\n",
       "</style>\n",
       "<table border=\"1\" class=\"dataframe\">\n",
       "  <thead>\n",
       "    <tr style=\"text-align: right;\">\n",
       "      <th></th>\n",
       "      <th>Customer ID</th>\n",
       "      <th>Order ID</th>\n",
       "      <th>Order Date</th>\n",
       "      <th>Ship Date</th>\n",
       "      <th>Ship Mode</th>\n",
       "      <th>Country</th>\n",
       "    </tr>\n",
       "  </thead>\n",
       "  <tbody>\n",
       "    <tr>\n",
       "      <th>0</th>\n",
       "      <td>RH-19495</td>\n",
       "      <td>CA-2012-124891</td>\n",
       "      <td>31-07-2012</td>\n",
       "      <td>31-07-2012</td>\n",
       "      <td>Same Day</td>\n",
       "      <td>United States</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>1</th>\n",
       "      <td>JR-16210</td>\n",
       "      <td>IN-2013-77878</td>\n",
       "      <td>05-02-2013</td>\n",
       "      <td>07-02-2013</td>\n",
       "      <td>Second Class</td>\n",
       "      <td>Australia</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>2</th>\n",
       "      <td>CR-12730</td>\n",
       "      <td>IN-2013-71249</td>\n",
       "      <td>17-10-2013</td>\n",
       "      <td>18-10-2013</td>\n",
       "      <td>First Class</td>\n",
       "      <td>Australia</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>3</th>\n",
       "      <td>KM-16375</td>\n",
       "      <td>ES-2013-1579342</td>\n",
       "      <td>28-01-2013</td>\n",
       "      <td>30-01-2013</td>\n",
       "      <td>First Class</td>\n",
       "      <td>Germany</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>4</th>\n",
       "      <td>RH-9495</td>\n",
       "      <td>SG-2013-4320</td>\n",
       "      <td>05-11-2013</td>\n",
       "      <td>06-11-2013</td>\n",
       "      <td>Same Day</td>\n",
       "      <td>Senegal</td>\n",
       "    </tr>\n",
       "  </tbody>\n",
       "</table>\n",
       "</div>"
      ],
      "text/plain": [
       "  Customer ID         Order ID  Order Date   Ship Date     Ship Mode  \\\n",
       "0    RH-19495   CA-2012-124891  31-07-2012  31-07-2012      Same Day   \n",
       "1    JR-16210    IN-2013-77878  05-02-2013  07-02-2013  Second Class   \n",
       "2    CR-12730    IN-2013-71249  17-10-2013  18-10-2013   First Class   \n",
       "3    KM-16375  ES-2013-1579342  28-01-2013  30-01-2013   First Class   \n",
       "4     RH-9495     SG-2013-4320  05-11-2013  06-11-2013      Same Day   \n",
       "\n",
       "         Country  \n",
       "0  United States  \n",
       "1      Australia  \n",
       "2      Australia  \n",
       "3        Germany  \n",
       "4        Senegal  "
      ]
     },
     "execution_count": 14,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "df_customer1.head()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 15,
   "id": "74858846",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "1590"
      ]
     },
     "execution_count": 15,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "df_customer1['Customer ID'].nunique()"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "7fbeb640",
   "metadata": {},
   "source": [
    "there are only 1590 unique customer ids out of 25754 rows in the customer data"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 39,
   "id": "12ecf926",
   "metadata": {},
   "outputs": [],
   "source": [
    "#def new_features(x):\n",
    " #   d = []\n",
    "  # d.append(x['Sales'].sum()) \n",
    "   # d.append(x['Shipping Cost'].sum())\n",
    "    #d.append(pd.to_datetime(x['Order Date']).min())\n",
    "   ##d.append(x['City'].nunique())\n",
    "    #return pd.Series(d, index=
['Purchases','Total_Sales','Total_Cost','First_Purchase_Date','Latest_Purchase_Date','Locati
on_Count'])\n",
    "\n",
    "#df_customer_new = df.groupby('Customer ID').apply(new_features)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 21,
   "id": "a780b626",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "Index(['Purchases', 'Total_Sales', 'Total_Cost', 'First_Purchase_Date',\n",
       "       'Latest_Purchase_Date', 'Location_Count'],\n",
       "      dtype='object')"
      ]
     },
     "execution_count": 21,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "df_customer_new.columns"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 22,
   "id": "fc5afbbf",
   "metadata": {},
   "outputs": [],
   "source": [
    "from datetime import datetime\n",
    "df_customer_new['Duration'] = (df_customer_new['Latest_Purchase_Date'] -
df_customer_new['First_Purchase_Date']).dt.days"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 23,
   "id": "a7cda426",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "Customer ID\n",
       "AA-10315     71.736842\n",
       "AA-10375     58.434783\n",
       "AA-10480     66.650000\n",
       "AA-10645     39.527778\n",
       "AA-315      177.285714\n",
       "Name: Frequency, dtype: float64"
      ]
     },
     "execution_count": 23,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "df_customer_new['Frequency'] =
df_customer_new['Duration']/df_customer_new['Purchases']\n",
    "df_customer_new['Frequency'].head()"
   ]
  },
{
 "cell_type": "code",
 "execution_count": 24,
 "id": "44d77510",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/html": [
     "<div>\n",
     "<style scoped>\n",
     "    .dataframe tbody tr th:only-of-type {\n",
     "        vertical-align: middle;\n",
     "    }\n",
     "\n",
     "    .dataframe tbody tr th {\n",
     "        vertical-align: top;\n",
     "    }\n",
     "\n",
     "    .dataframe thead th {\n",
     "        text-align: right;\n",
     "    }\n",
     "</style>\n",
     "<table border=\"1\" class=\"dataframe\">\n",
     "  <thead>\n",
     "    <tr style=\"text-align: right;\">\n",
     "      <th></th>\n",
     "      <th>Purchases</th>\n",
     "      <th>Total_Sales</th>\n",
     "      <th>Total_Cost</th>\n",
     "      <th>First_Purchase_Date</th>\n",
     "      <th>Latest_Purchase_Date</th>\n",
     "      <th>Location_Count</th>\n",
     "      <th>Duration</th>\n",
     "      <th>Frequency</th>\n",
     "    </tr>\n",
     "    <tr>\n",
     "      <th>Customer ID</th>\n",
     "      <th></th>\n",
     "      <th></th>\n",
     "      <th></th>\n",
     "      <th></th>\n",
     "      <th></th>\n",
     "      <th></th>\n",
     "      <th></th>\n",
     "      <th></th>\n",
     "    </tr>\n",
     "  </thead>\n",
     "  <tbody>\n",
     "    <tr>\n",
     "      <th>AA-10315</th>\n",
     "      <td>19</td>\n",
     "      <td>13747.41300</td>\n",
     "      <td>1236.15</td>\n",
     "      <td>2011-03-31</td>\n",
     "      <td>2014-12-23</td>\n",
     "      <td>18</td>\n",
     "      <td>1363</td>\n",
     "      <td>71.736842</td>\n",
     "    </tr>\n",
     "    <tr>\n",
     "      <th>AA-10375</th>\n",
     "      <td>23</td>\n",
     "      <td>5884.19500</td>\n",
     "      <td>903.92</td>\n",
   "      <td>2011-04-21</td>\n",
   "      <td>2014-12-25</td>\n",
   "      <td>23</td>\n",
   "      <td>1344</td>\n",
   "      <td>58.434783</td>\n",
   "    </tr>\n",
   "    <tr>\n",
   "      <th>AA-10480</th>\n",
   "      <td>20</td>\n",
   "      <td>17695.58978</td>\n",
   "      <td>1633.67</td>\n",
   "      <td>2011-01-11</td>\n",
   "      <td>2014-09-05</td>\n",
   "      <td>20</td>\n",
   "      <td>1333</td>\n",
   "      <td>66.650000</td>\n",
   "    </tr>\n",
   "    <tr>\n",
   "      <th>AA-10645</th>\n",
   "      <td>36</td>\n",
   "      <td>15343.89070</td>\n",
   "      <td>1752.27</td>\n",
   "      <td>2011-01-12</td>\n",
   "      <td>2014-12-05</td>\n",
   "      <td>35</td>\n",
   "      <td>1423</td>\n",
   "      <td>39.527778</td>\n",
   "    </tr>\n",
   "    <tr>\n",
   "      <th>AA-315</th>\n",
   "      <td>7</td>\n",
   "      <td>2243.25600</td>\n",
   "      <td>215.80</td>\n",
   "      <td>2011-08-06</td>\n",
   "      <td>2014-12-29</td>\n",
   "      <td>7</td>\n",
   "      <td>1241</td>\n",
   "      <td>177.285714</td>\n",
   "    </tr>\n",
   "  </tbody>\n",
   "</table>\n",
   "</div>"
  ],
  "text/plain": [
   "             Purchases  Total_Sales  Total_Cost First_Purchase_Date  \\\n",
   "Customer ID                                                           \n",
   "AA-10315            19  13747.41300     1236.15          2011-03-31   \n",
   "AA-10375            23   5884.19500      903.92          2011-04-21   \n",
   "AA-10480            20  17695.58978     1633.67          2011-01-11   \n",
   "AA-10645            36  15343.89070     1752.27          2011-01-12   \n",
   "AA-315               7   2243.25600      215.80          2011-08-06   \n",
   "\n",
   "            Latest_Purchase_Date  Location_Count  Duration   Frequency  \n",
   "Customer ID                                                             \n",
   "AA-10315              2014-12-23              18      1363   71.736842  \n",
   "AA-10375              2014-12-25              23      1344   58.434783  \n",
   "AA-10480              2014-09-05              20      1333   66.650000  \n",
   "AA-10645              2014-12-05              35      1423   39.527778  \n",
   "AA-315                2014-12-29               7      1241  177.285714  "
  ]
 },
 "execution_count": 24,
 "metadata": {},
 "output_type": "execute_result"
}
```

```
   ],
   "source": [
    "df_customer_new.head()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 26,
   "id": "88b4cff0",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "count    1590.000000\n",
       "mean      111.303339\n",
       "std        80.489358\n",
       "min         0.000000\n",
       "25%        50.669643\n",
       "50%        73.055728\n",
       "75%       155.083333\n",
       "max       655.500000\n",
       "Name: Frequency, dtype: float64"
      ]
     },
     "execution_count": 26,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "df_customer_new['Frequency'].describe()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 27,
   "id": "a9e4e5ad",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "image/png":
"iVBORw0KGgoAAAANSUhEUgAAAX0AAAEGCAYAAACJnEVTAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24
zLjUuMSwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy/YYfK9AAAACXBIWXMAAAsTAAALEwEAmpwYAAAWvklEQVR4nO3de7B
dZ33e8e+DBOJijOXo2FUlUwlGA5U9CReNY0LqONhgcwnytPFUFIISzGG4OOIWKzA
QuwLipIDNsY0vkq1YIMcWFpYIIQGSGmx+/WO9GjbHWzo6Zx+di9b3M7Nnv/td71rrt46Onr3Ou/deO1WFJKkfnjDXBUi
SZo+hL0k9YuhLUo8Y+pLUI4a+JPXI4rkuYDLLli2rVatWzXUZkrSg3Hrrrrd+sqrGJ/fM+9FetWsX4+PhclyFJC0qSvxr
W7/SOJPWIoS9JPWLoS1KPGPqS1COGviT1iKEvST1i6EtSjxj6ktQjhr4k9ci8/0TuKFZd/qm5nLkET3H/lK+e6BKnXPNO
XpB6ZNPSTvC/JwSR3D1l2WZJKsmygb3O+FSe5qy96TJDN3GJKkY3EsZ/ofAC6c2JnkDOClwAMDfWuBDcC
ZbZ2rkixqi68GNgFr2u1x25QkHV+Thn5V3QR8a8ii/wq8DRj8ZvX1wHVV9UhV7QP2AmcnWQ6cXFVfrO6bXFDrO6b2D8IXDRq8ZK
kqZnWnH6SVwNfr6o7JixaATw48Hh/61vR2hP7JUmzaMrv3knzgOmf
6zwZWA3ckuR9YCdyW5B/RncGfMTB2JfBQ6185pH+oqtaVeuqat3Y2OO++EWSNE1TDv2ququqTqquqVVW1ii7QX1BV3wB
2ABuSLEmymu4F21uq6gDwDwnSTntHftvB64fuYOQ5J0LI7lLZsfBr4IPCfJ/iSXHGlsVe0Ctg07gc8Al1bVY23xm4Br6F7
c/Srw6RFrlyRN0aRz+laRz+lX1mkmWr5rweAuwZci4ceCsKdYnSZpBfiJXknrE0JekHjHjHjH0JalHDH1J6hFDX5J6xNCXpB4x9CW
pRwx9SeoRQ1+SesTQl6QeMfQlqUcMfUnqEUNfknrE0JekHjH0JalHDH1J6hFDX5J6xNCXpB4x9CWpRwx9SeqRSUM/yfu
SHExy90Df7yW5N8ST6R5JSBZZuT7E2yJ8kFA/0vTHARsZSsBTY
AZ7Z1rkqyqK1zNbAJWNNuE7cpSTrOJg39qroJ+Naevhuq6tH28EvAytZeD1xxXVY9U1T5gL3B2kuXAyVX1xaoq4IPARTN
0DJKkYzQTc/pvAD7d2iuABWeW7W99K1p7Yv9QSTYlGU8yPjExQbjubc3UYlGU8yfujQoRkoUZIEI4Z+krcDjwLXHu4aMqyOOj9UVW2tqnVVtW5
sbGyUEiVJAxZPd8UkG4FXAee3KRvozuDPGBi2Enio9a8c0n9UW17aNfTn9i6E9m6Al8ygA2A4FXnT44+6Tj24XU9e6zqHo8VzeKFCcv64TUs6A5t0fMkmFOHS+qLwnmR1X15e+Ue4+2cykHwKJbxc1tHGeo4ZvVHeKbB/hYR5WK7I3Ps8gNkXxi97JXYQWzVVdn1/zn1iBX8KKIqY3nR
6xNCXpB4x9CWpRwx9SeoRQ1+SesTQl6QeMfQlqUcMfUnqEUNfknrE0JekHjH0JalHDH1J6hFDX5J6ZNLQT/K+JAeT3D3
Qd2qSG5Pc1+6XDizbnGRvkj1JLhjof2GSu9Sqy9yTJzB+OJ0lojuVM/wPAhRP6Lgd2VtUaYGd7TJK1wAbgzLbOVXUkXW
uBjYBa9pt4jYlScfZpKFfVfVTcB35rQvR7Y1trbgIsG5Kr6YlUV8MGBdSRJs2S6c/qnxVsVd0oGv50M8D
```

```
/WutfATw4MG5/61vR2hP7h0qyKcl4kvFDhw5Ns0RJ0kQz/ULusHn6Okr/UFW1tarWVdW6sbGxGStOkvpuuqH/cJuyod0
fbP37gTMGxq0EHmr9K4f0S5Jm0XRDfwewsbU3AtcP9G9IsiTJaroXbG9pU0DfSXJOe9fO6wfWkSTNksWTDUjyYeAlwLI
k+4ErgCuB7UkuAR4ALgaoql1JtgO7gUeBS6vqsbapN9G9E+gpwKfbTZI0iyYN/ap6zREWnXeE8VuALUP6x4GzplSdJGl
G+YlcSeoRQ1+SesTQl6QeMfQlqUcMfUnqEUNfknrE0JekHjH0JalHDH1J6hFDX5J6xNCXpB4x9CWpRwx9SeoRQ1+SesT
Ql6QeMfQlqUcMfUnqEUNfknrE0JekHjH0JalHRgr9JL+RZFeSu5N8OMmTk5ya5MYk97X7pQPjNyfZm2RPkgtGL1+SNBX
TDv0kK4A3A+uq6ixgEbABuZYWVVrgJ3tMUnWtuVnAhcCVyVZNFr5kqSpGHV6ZzHwlCSLgacDwHrgW1t+TbgotZeD1x
XVY9U1T5gL3D2iPuXJE3BtEO/qr4O/D7wAHAA++NuqugE4vaoOtDEHgNPaKiuABwc2sb/1PU6STUnGk4wfOnRouiVKkiY
YZXpnKd3Z+2rgHwNPS/K6o6o60ypK+GDayqrVW1rqrWjY2NTbdESdIEo0zvnA/sq6pDVfUD4OPAzwAPJ1kOO04PtvH7gTM
G1l9JNx0kSZolo4T+A8A5SZ6aJMB5wD3ADmBjG7MRuL61dwAbkixJshpYA9wywv4lSVO0eLorVtXtSTOeLorVtXNST4K3AY8CtwObAV
OArYnuYTuieHiNn5Xku3A7jb+0qp6bMT6JUlTMO3QB6iqK4ArJnQ/QnfWP2z8FmDLKPuUJE2fn8iVpB4x9CWpRwx9Seo
RQ1+SesTQl6QeMfQlqUcMfUnqEUNfknrE0JekHjH0JalHDH1J6hFDX5J6xNCXpB4x9CWpRwx9SeoRQ1+SesTQl6QeMfQ
lqUcMfUnqkZFCP8kpST6a5N4k9yR5UZJTk9yY5L52v3Rg/OYke5PsSXLB6OVLkqZi1DP9/wZ8pqqeC/wUcA9wObCzqtY
AO9tjkqwFNgBnAhcCVyVZNOL+JUlTMO3QT3IycC7wXoCq+n5VfRtYD2xrw7YBF7X2euC6qnqkvYBe4Gzp7t/SdLUjXK
m/yzgEPD+JLcnuSbJO4DTq+oAQLs/rY1fATw4sP7+1idJmiWjhP5i4AXA1VX1fOB7tKmcI8iQ1vho6MNmUZDzJ+KFDh0Y
oUZI0aJTQ3w/sr6qb2+OP0j0JPJxkOUC7Pzgw/oyB9VcCDw3bcFVtrap1VbVubGxshBIlSYOmHfpV9Q3gwSTPaV3nAbu
BHcDG1rcRuL61dwAbkixJshpYA9wy3f1LkqZu8Yjr/zpwbKzBCZkCZkLAapqV5LtdE8MjwKXVtVjI+5
fkjQFI4V+VX0FWDdk0XlHGL8F2DLKPiVJ0+cnciWpRwx9SeoRQ1+SesTQl6QeMfQlqUcMfUnqEUNfknrE0JekHjH0Jal
HDH1J6hFDX5J6xNCXpB4x9CWpRwx9SeoRQ1+SesTQl6QeMfQlqUcMfUnqEUNfknrE0JekHhk59JMsSnJ7kk+2x6cmuTH
Jfe1+6cDYzUn2JtmT5IJR9y1JmpqZONN/C3DPwOPLgZ1VtQbY2R6TZC2wATgTgTgTuBC4KsmiGdi/i/J0kYjRT6SVYCrwSuGeh
eD2xr7W3ARQP911XVI1W1D9gLnD3K/iVJUzPqmf5VV7ff67gbcBPxzoO72qDgC0+9Na/wrgwYFX+1vf4yTZlGQ8yfihQ4dGLFG
SdNiOQz/Jq4CDVXXrsa4ypK+GDayqrVW1rqrWjY2NTbdESdIEi0dY98XAq58O8AngycHKSDwEPJ1leVQeLSLAcOtvQmLAcOtvH7gTM
G1l8JPDTC/iVJUzTtM/2q2lxVK6tqFd0LtJ+vqtcBO4CNbdhG4PrW3gFsSLIkyWpgDXDLtCuXXJE3ZKGf6R3IlsD3JJcA
DwMUAVbUryXZgN/AocGlVPXYc9i9+q9rIZCf2q+jzw+db+a+JPWIoS9I4a+JPWIoS9JPWIoS9JPWLoS1KPGPrSUS1COGviT
oS1KPGPqS1COGviT1iKEvST1i6EtSjxj6ktQjhr4k9YihLUo8Y+pLUI4a+JPWIoS9JPWLoS1KPGPqS1COGviT
1iKEvST0y7dBPckaSzyW2W5J8muQJG9puJG9p/acmuTHJfe1+6cA6m5PsTe+o8C/qap/CpwDXJpkLXA5sA5sL
q1gA722Pasg3AmcCFwFVJFo1SvCRpaqYd+lV1oKpua+3vAPcAK4D1wLY2bBtwUWuvB66rqkeqah+wFzh7uvuXJE3djMz
pJ1kFPB+4GTi9qg5A98QAnNaGrQAeHFhtf+sbtr1NScz9QA2uRJi26RKxPYW6vq7442dEhfDRtYVVural1
VrRsbGxu1RElSM1LoJ3kiXeBfW1Ufb90PJ1neli8HDrb+/cAZA6uvBB4aZf+SpKkZ5d07Ad4L3lNVfxnuG5VfRW9kl0AJPy5VXP4iP8aMtpZWSW9cZM1aBZxYrBAGVY1
H+jckWZJkNbAGuGW6+5ckTd3iEdZ9MfBLwF1JvtL6L6/j1wJbA9ySXAA8DFAFW1K8l2YDfdO38urarHRti/JGmKph36Vf
XDJ+nBzjvCOtsAbZMd5+SpNH4iVxJ6hFDX5J6xNCXpB4Z5YVcacpWX6f6puS5BQ9x/5SvnugTNEs/0JalHDH1J6hFDX5J
6xNCXpB4x9CWpRwx9SeoRQ1+SesTQl6QeMfQlqUcMfUnqEUNfknrE0JekHjH0JalHDH1J6hFDX5J6xNCXpB6Z9dBPcmG
SPUn2Jrl8tvcvSX02q9+clWQR8L7gS8n2VFVu2ezDkk/zm80m3+O17eZzfaZ/tnA3Oq6R1
2d+SuAB4ceLwf+OmJg5JsJzAja1h99Nsmea+1sGfHOa686lhVo3LNzaF2rdsHBrX6h1wyzUnneOvIl/MqxztkM/Q/rqcR1
VW4GtI+8sGa+qdaNuZ7Yt1Lph4da+UOuGhVv7Qq0bFnbntsz29sx84Y+DxSuChWa5BknprtkP/y8CaJKuTPAnYN7TP
kqbdmdXqnqh5N8mvAnwKLgPdV1a7juS2UNTbTn+m/qSFmBVKNbWnpSc4dpJc1r7UbZn+xjRUWqJTUZ4jgqs7+Y5tSa/2ws2bC4DZH4Q1ATaSaTQRFzGiVIBsUK61KF1RV2O2SGJZsWQj3SPg+ZSSHUNU
ST7bH8772JPcnuSvJV5KML5S6Wy2nJPloknvb//nzLl+bngKFkrtk6qqE+pG9wLxV4FnAU8C7gDWznbPdA3J4c6b3+8Cl7f
25cA7W3tt04YlwOp2bIvmqO7lwAta+yTg71p987t2IMAzWvt6wM03zH7lfSfmeDZhL55X3er2Zxv
wxtZ+EnDKQql9sduJeKY/7y/1VfIVe0D9tId46yrqgNVdVtrfwe4h1N5Xl+7vTbw8v4U2Pnjn8
8YrsV87zuw5KsBF4JXDPQvRDqhnPeO3yk7P4ReryAIpC2W/PqAJmAe+x8sH3aNjGqqUyp/0Mz0UHw8DhDwXeirBfGeT+xhd+4kpb+NJmE
3x5PkJOBjwFur6tQS1XR5ebZ10/J7PbXF+nLuXqOYahs/MqNjvaiv7f+uq8jRxcuTi7C5r+uZGR4Sbiv5rVs7uOUFUUFjU/TX2ke
X002/Xl1Vzwe+RzedcyTzqXYAXnXuW1UNYmD6bHT9WY3BmMgGl24k04YNvvoLuzK8gVxCLQhL9AvjLvf6rs0LfW4NZdBydYoGFZ0cYPGMIajd8f23uTWN9tA8mdWadqqaznUnRcrNY1DbhlLVb9v8
bqvz5JB9iAdReVQ+1+4PAJ+imgZjZ93a2W/e2vCdmB2ja28Erh/o35BkgSZLVwmojyS
hm+e8p6reNbBoXeeZCzJKa39F0OB84F7meLCd6wRlzXCTqKOJ3lZ9B8p3cdP3J7e1Cm+NC6Od/ARD7QfRtf+8ZLAg==
WdR6wmwVQ+zGZ61eSj8cNeAXdOOu+Crx9rusZUt+HgQPAD+jOEi4BfgLYCdzX7k8dGP/2dix7gJfPYd0/S/dn653AV9r
tFfO9duAngdtb3XcDv9X653XdQ47jJfzo3Tvzuna6efE72m3X4f+H873ugVqeB4y335k/BpYulNonu3kZBknqkRNxeke
SdASGviT1iKEvST1i6EtSjxj6ktQjhr6OqyRvblcpvHauaznekjy3XVHTPnqV9fnfyUdKP+JZNHVdJ7qV73/K+Cf2
Lq+rROSrruEh3Ge+nVNUVM7zdI/6skny3qk6ayf3pxOaZvo6bJP+L7kM6O5L8RpJ3tNMa5Abgg+2Tsh9L8uRGqwlzDbQe7e
S3NDOmP8wyV8lWZZkVX78OwguS/KO1n52ks+Oi3v9eZLtnv4PJ2JHlPki8k+VqSXxxY/23teDcjnyuR/UeGjG12geVVrkjzumjd
JnpfkSOnuTPPKJJEuTvAJ4K4K/DGJJ8bss5A3k2uLv1j4ib1IA9kEd6n0N3
WdgxfSnJ66/uFJDe3n+H/Gej/ufbXyOgD/vGwDXVgX2An9ddKUNN3ffifbe1n0l3
eAeA9/OhTs6+k+xTwMmAVP/4dBJcB72jtncCa5v4oK19r5H8w5eDMpSAHwJjAJ+cc9Sl5lSpVmvAHwA+AjdCc5austuQ3cRsC8A4T22PT233i3nwOe19r/Gfj1Icd
OJ/Bzrf07wLsHju+wI/wcCnhta/8W8D9a+/PAutZeBt5zf2r9M92ntw3W98/B+2uO1a9v9hdb+Xe+A/HF7Oj/6SfyPwB63
9J3QXQgM4ie7iYi+j+6LvtJ/TJ4Fz5/p3x9vxuSOe9kQgHUc7quofWvt8YG13SR8ATm5nnucCvwBQVZ9K8v+OUscLgKcPVK
8GeAjA9taMjDkj6vqh8Duw2e8bd/vr6q/b/s5/P0G1wC/kuQ3gJkj3GcEpV/Vnr2kb3pDKZHwJ/1NofAj5+lG
H3ThQ1/lO196h1Xv4Z/J9upCG7gn1pa29EvijdBcGexJwepHrt/wLvaq+xfLyq9id5GV3w397GnER3/ZibjqFGLTCGvmb
b9wbaTwBeNPAkAEAL7mEvNj3Kj09JPnlgO9+u7tLJwzwyuPmB+2H7+BhwBfBZ4Naq+usjbHNUh/c9eExPnjBm8Gd1pHp
/UFWH+x/jR/+n/zvwrqrakeQlwP3QldH+FUFVXJvkU3TTvJdp3TWTvpTk/Lbt/ZT1cm5ptRK0yyPvPDhHlnD4o55W7rA3xjAvkvV+3ztI+TK0dLJp6LTnGLEeVL5oXU91j4rdPE1jAo
KgIeB09qc/xLgVQDV83HpmT0/s8qz93eTDapdH3JAAEW+5qc/B/wpBJDXwPqRV8fXxbtJAT+5YTIxnIP8
fA3/R2vcDL2ztX+TIv6slh5lLMAzgK+39sYxPrzNeRPJbD4esHTKakgGf3WmRSbE87i/7r5oDxbF2X6kZ9+Ogxn5zcLq+uMh5Z8qzv1UiWiA
38Kut/7eBc9uLqi8DHgCoqh/QzaLY0FUdj/oVmVX1GbpgGk+ZFRbA/k0OO4EkeQzVp98h2
djfF3juFYvwec2V4Y/vmBdX4feFOSL9DhN6R/JfwKWJrm7HeMHeM/n2R/76Cb7vpzYLh/W8d2MY/AJ+uqhvoXl/5YpK76w59Mjt+wbp35fr
f//RjOCYtQL5lU/Nee0fLuqr65mRjZ2h/lwHPqKr/OIPb9K2Vm6hec05cGJPkE8Gy6s3HphOOZviT1iHP6ktQjhr4k9Yi
hL0k9YuhLUo8Y+pLUI/8f4RNR88BuvgcAAAAASUVORK5CYII=\n",
      "text/plain": [
       "<Figure size 432x288 with 1 Axes>"
      ]
     },
     "metadata": {
      "needs_background": "light"
     },
     "output_type": "display_data"
    }
   ],
   "source": [
    "# 1. Profile the customers based on their frequency of purchase - calculate frequency
of purchase for\n",
    "# each customer and plot a histogram to get the threshold for Low/Mid/High frequency
customers\n",
    "\n",
    "plt.hist(df_customer_new['Frequency'], bins=3)\n",
    "plt.xlabel('frequency of purchase')\n",
    "plt.show()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 28,
   "id": "c076ed59",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "Low     1442\n",
       "Mid      138\n",
       "High      10\n",
       "Name: freq_range, dtype: int64"
      ]
     },
     "execution_count": 28,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "# bucketing continuous data\n",
    "def freq(x):\n",
    "    if x < 219:\n",
    "        return 'Low' \n",
    "    elif x < 436:\n",
    "        return 'Mid' \n",
    "    else:\n",
    "        return 'High'\n",
    "\n",
    "df_customer_new['freq_range'] = df_customer_new.Frequency.apply(freq)\n",
    "\n",
    "df_customer_new['freq_range'].value_counts()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 29,
   "id": "b9da0ca5",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/html": [
"<div>\n",
"<style scoped>\n",
"    .dataframe tbody tr th:only-of-type {\n",
"        vertical-align: middle;\n",
"    }\n",
"\n",
"    .dataframe tbody tr th {\n",
"        vertical-align: top;\n",
"    }\n",
"\n",
"    .dataframe thead th {\n",
"        text-align: right;\n",
"    }\n",
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",
"      <th></th>\n",
"      <th>Purchases</th>\n",
"      <th>Total_Sales</th>\n",
"      <th>Total_Cost</th>\n",
"      <th>First_Purchase_Date</th>\n",
"      <th>Latest_Purchase_Date</th>\n",
"      <th>Location_Count</th>\n",
"      <th>Duration</th>\n",
"      <th>Frequency</th>\n",
"      <th>freq_range</th>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>Customer ID</th>\n",
"      <th></th>\n",
"      <th></th>\n",
"      <th></th>\n",
"      <th></th>\n",
"      <th></th>\n",
"      <th></th>\n",
"      <th></th>\n",
"      <th></th>\n",
"      <th></th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>AA-10315</th>\n",
"      <td>19</td>\n",
"      <td>13747.41300</td>\n",
"      <td>1236.15</td>\n",
"      <td>2011-03-31</td>\n",
"      <td>2014-12-23</td>\n",
"      <td>18</td>\n",
"      <td>1363</td>\n",
"      <td>71.736842</td>\n",
"      <td>Low</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>AA-10375</th>\n",
"      <td>23</td>\n",
"      <td>5884.19500</td>\n",
"      <td>903.92</td>\n",
"      <td>2011-04-21</td>\n",
"      <td>2014-12-25</td>\n",
"      <td>23</td>\n",
"      <td>1344</td>\n",
"      <td>58.434783</td>\n",
"      <td>Low</td>\n",
 "    </tr>\n",
 "    <tr>\n",
 "      <th>AA-10480</th>\n",
 "      <td>20</td>\n",
 "      <td>17695.58978</td>\n",
 "      <td>1633.67</td>\n",
 "      <td>2011-01-11</td>\n",
 "      <td>2014-09-05</td>\n",
 "      <td>20</td>\n",
 "      <td>1333</td>\n",
 "      <td>66.650000</td>\n",
 "      <td>Low</td>\n",
 "    </tr>\n",
 "    <tr>\n",
 "      <th>AA-10645</th>\n",
 "      <td>36</td>\n",
 "      <td>15343.89070</td>\n",
 "      <td>1752.27</td>\n",
 "      <td>2011-01-12</td>\n",
 "      <td>2014-12-05</td>\n",
 "      <td>35</td>\n",
 "      <td>1423</td>\n",
 "      <td>39.527778</td>\n",
 "      <td>Low</td>\n",
 "    </tr>\n",
 "    <tr>\n",
 "      <th>AA-315</th>\n",
 "      <td>7</td>\n",
 "      <td>2243.25600</td>\n",
 "      <td>215.80</td>\n",
 "      <td>2011-08-06</td>\n",
 "      <td>2014-12-29</td>\n",
 "      <td>7</td>\n",
 "      <td>1241</td>\n",
 "      <td>177.285714</td>\n",
 "      <td>Low</td>\n",
 "    </tr>\n",
 "  </tbody>\n",
 "</table>\n",
 "</div>"
],
"text/plain": [
 "             Purchases  Total_Sales  Total_Cost First_Purchase_Date  \\\n",
 "Customer ID                                                           \n",
 "AA-10315            19  13747.41300     1236.15          2011-03-31   \n",
 "AA-10375            23   5884.19500      903.92          2011-04-21   \n",
 "AA-10480            20  17695.58978     1633.67          2011-01-11   \n",
 "AA-10645            36  15343.89070     1752.27          2011-01-12   \n",
 "AA-315               7   2243.25600      215.80          2011-08-06   \n",
 "\n",
 "            Latest_Purchase_Date  Location_Count  Duration   Frequency  \\\n",
 "Customer ID                                                              \n",
 "AA-10315              2014-12-23              18      1363   71.736842   \n",
 "AA-10375              2014-12-25              23      1344   58.434783   \n",
 "AA-10480              2014-09-05              20      1333   66.650000   \n",
 "AA-10645              2014-12-05              35      1423   39.527778   \n",
 "AA-315                2014-12-29               7      1241  177.285714   \n",
 "\n",
 "            freq_range  \n",
 "Customer ID             \n",
 "AA-10315           Low  \n",
 "AA-10375           Low  \n",
 "AA-10480           Low  \n",
 "AA-10645           Low  \n",
 "AA-315             Low  "
```

```
       ]
      },
      "execution_count": 29,
      "metadata": {},
      "output_type": "execute_result"
     }
    ],
    "source": [
     "# profiling based on purchase frequency\n",
     "df_customer_new.head()"
    ]
   },
   {
    "cell_type": "markdown",
    "id": "d224c3cb",
    "metadata": {},
    "source": [
     "2.Are the high frequent customers contributing more revenue"
    ]
   },
   {
    "cell_type": "code",
    "execution_count": 32,
    "id": "6594a521",
    "metadata": {},
    "outputs": [],
    "source": [
     "import numpy as np\n",
     "result = df.groupby([\"Customer Name\"])
['Profit'].aggregate(np.sum).reset_index().sort_values('Profit',ascending = False).head(20)"
    ]
   },
   {
    "cell_type": "code",
    "execution_count": 33,
    "id": "effa8263",
    "metadata": {},
    "outputs": [],
    "source": [
     "df_customer_new['Revenue'] = df_customer_new['Purchases'] *
df_customer_new['Total_Sales']"
    ]
   },
   {
    "cell_type": "code",
    "execution_count": 34,
    "id": "3a03d92c",
    "metadata": {},
    "outputs": [
     {
      "data": {
       "image/png":
"iVBORw0KGgoAAAANSUhEUgAAAWAAAAFgCAYAAACFYaNMAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24
zLjUuMSwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy/YYfK9AAAACXBIWXMAAAsTAAALEwEAmpwYAAAWjUlEQVR4nO3de7S
ldX3f8fcHplw0QrgMl3ARotQEDWAYERcxmkUyjElbiEUdm8iYRTPVhYlZraHSZoxWIiywlaSgGymVFcZaXOm0DOsU
2RIPASCgIhjIVgQFGIDOOI1HqDN/+sX9H9xzPHPZc9vmdy/u1l772ARotQEDWAYERcxmkUyjElbiEUdm8iYRTPVhYlZraHSZoWIiywlaVixKSgGymVFcZaXOm0DOsU
2RIPASCgIhjIVgQFGIDOOI1HqDN/+sX9H9xzPHPZc9vmdy/u11l772d/9/J793Rv4nIff8+xnp6qQJM28vXo3IEkLlQE
sSZ0YwJLUiQEsSZ0YwJLUyaLeDcwWy5Ytq5tvvrl3G5mp0xVdA+4efrpp3u3IGmBMYAlqRMDWJI6MYAlqRMDWJI6MYA
lqRMDWJI6MYAlqRMDWJI6MYAlqRMDWJI6MYAlqRMDWJI68Wpoc9yFF17Ihg0bOOKII7jssst6tyNJtmNpLxjAc9yGDRt47LH
HerchaRc4BSFJnRjAktSJASxJnRjAktSJASxJnRjAktSJASxJnRjAktSJASxJnRjAktSJASxJnRjAktS
Jl6PcSaf+7vW9W9jOi57ewt7AI09vmVW9feWPzuvdgjTruQcsSZ0YwJLUiQEsSZ0YwJLUiQEsSZ0YwJLUiQEsSZ0YwJLUiQEsSZ0YwJL
UiQEsSZ0YwJLUiQEsSZ0YwJLUiQEsSZ2MLYCTHJPkfyb5WpL7kry71Q9OsibJg+3rQa2eJH+SZG2Se5P85LaNm+LuVBCCt
J0ur7JvlEq9+e5LihMSvaazyYZMW43qck7apx7ck7apx7gFvBf5NVf00cDpwQZKTgFpvQZIdSPhF8MhIpwQbVnJZ/bvm5FJv3Rv4buhJ
YCZzQbsta/XxgU1W9FLgc+GDb1sHAxcCrgdOAi4eDXpJmg7EFcFWtr6q/bdvfAe4HjgLOBq5rd7sOOKdtnw1cXwNuBw5KciTwBm
IWAecluRI4ICquq2qCrh+0pi2ZiKmqjVVW1CVjDDDN7XeqQN0N7Xlnepf4XknK6qDNRteVRpdbxgajX4S2ZR/rZqno
HV9UTMAjpJIe11Y4wCvjwbH97tT8T65HPHjHmRK4mbKaxYk7aKxH4RL8mP
```

Ap4DfqapvT7fqFLWapr6rY35YqLqqqpZU1ZLFixdP05ok7XljDeAk/4hB+P5FVX26lb/ZphVo90+2+nrgmKHhRwOPt/r
RU9S3G5NkEXAgsHGabUnSrDHOsyACXA18rar+ZOip1cDEWQkrgM8O1Ze3MxuOZ3Cw7Y42XbElyeltm+dNGjOxrXOBL7R
54s8BS5Mc1A6+LW01SZo1xjkHfAbwNuDeJHe32r8DPgCsSnI+8AjwJoCqui/JKuB+BmdQXFBV29q4dwLXAvsDN7UbDAL
+hiTrGOz5Lm/b2pjk/cCdbb1LqmrjmN6nJO2SsQVwVX2RqediAc7cwZhLgUunqK8FXjFF/Xu0AJ/iuWuAa0btV5Jmmt+
Ek6RODGBJ6sQAlqRODGBJ6sQAlqRODGBJ6sQAlqRODGBJ6sQAlqRODGBJ6sQAlqRODGBJ6sQAlqRODGBJ6sQAlqRODGB
J6sQAlqRODGBJ6sQAlqRODGBJ6sQAlqRODGBJ6sQAlqRODGBJ6sQAlqRODGBJ6sQAlqRODGBJ6sQAlqRODGBJ6sQAlqR
ODGBJ6sQAlqRODGBJ6sQAlqRODGBJ6sQAlqRODGBJ6sQAlqRODGBJ6sQAlqRODGBJ6sQAlqRODGBJ6sQAlqRODGBJ6sQ
AlqRODGBJ6sQAlqRODGBJ6sQAlqRODGBJ6sQAlqRODGBJ6sQAlqRODGBJ6mRsAZzkmiRPJvnqUOOPkjyW5O52++Wh5y5
Ksi7JA0nOGqqfmuTe9tyHkqTV903yiVa/PclxQ2NWJHmw3VaM6z1K0u4Y5x7wtcCyKeqXV9Up7faXAElOBJYDL29jrki
yd1v/SmAlcEK7TWzzfGBTVb0UuBz4YNvWwcDFwKuB04CLkxy059+eJO2esQVwVd0KbBxx9bOBG6vq2ap6CFgHnJbkSOC
Aqrqtqgq4HjhnaMx1bfmTwJlt7/gsYE1VbayqTcAapv5DIEld9ZgDfleSe9oUxcSe6VHAo0PrrG+1o9ry5Pp2Y6pqK7A
ZOGSabf2IJCuTrE2y9qmnntq9dyVJO2mmA/hK4CXAKcATwH9o9Uyxbk1T39Ux2xerrqqqJVW1ZPHixdO0LUl73owGcFV
9s6q2VdVzwEcZzNHCYC/1mKFVjwYeb/Wjp6hvNybJIuBABlMeO9qWJM0qMxrAbU53wq8CE2dIrAaWtzMbjmdwsO2Oqno
C2JLk9Da/ex7w2aExE2c4nAt8oc0Tfw5YmuSgNsWxtNUkaVZZNK4NJ/k48Hrg0CTrGZyZ8PokpzCYEvgG8K8Aquq+JKu
A+4GtwAVVta1t6p0MzqjYH7ip3QCuBm5Iso7Bnu/ytq2NSd4P3NnWu6SqRj0YKEkzZmwBXFVvnaJ89TTrXwpcOkV9LfC
KKerfA960g21dA1wzcrOS1IHfhJOkTgxgSerEAJakTgxgSerEAJakTgxgSerEAJakTgxgSerEAJakTgxgSerEAJakTgx
gSerEAJakTgxgSerEAJakTgxgSerEAJakTkYK4CSHJ7k6yU3t8YlJzh9va5IOv426B3wtgx+2/In2+P8AvzOGfiRpwRg
1gA+tqlXAcwBVtRXYNv0QSdJ0Rg3gZ5IcwuDXjElyOrB5bF1J0gIw6q8i/2tgNfCSJF8CFgPnjq0rSVoARgrgqroryeu
AlwEBHqiq74+1M0ma50YK4CTnTSr9bBKq6vox9CRJC8KoUxCvGlreDzgTuAswgCVpF406BfFbw4+THAjcMJaOJGmB2NV
vwv0DcMKebESSFppR54D/K+0UNAahfSKwalxNSdJCMOoc8B8PLW8FHq6q9WPoR5IWjFHngP9q3I1I0kIz6sV43pjkwSS
bk3w7yZYk3x53c5IOn406BXEZ8E+r6mvjbEaSFpJRz4L4puErSXvWqHvAa5N8AvgvwLMTxar69DiakqSFYNQAPoDBub9
Lh2oFGMCStItGPQviN8bdiCQtNKOeBfGPk9yS5Kvt8UlJfm+8rUnS/DbqQbiPAhcB3weoqnuA5eNqSpIWglED+AVVdce
k2tY93YwkLSSjBvDTSV7CD3+S6FzgibF1JUkLwKhnQVwAXAX8VJLHgIeAXxtbV5K0AIwawA9X1S8meSGwV1VtGWdTkrQ
QjDoF8VCSq4DTge+MsR9JWjBGDeCXAf+DwVTEQ0n+LMnPja8tSZr/RgrgqvpuVa2qqjcCr2TwzTgvUSlJu2HknyRK8ro
kVzD4Mc79gDePrStJWgBG/Umih4C7GfwM0e9W1TPjbEqSFoJRz4I4uaq8ALsk7UGjTkEc4bUgJGnP8loQktSJ14KQpE6
8FoQkdeK1ICSpk1F/EePrwA+uBQF8F3gL8PAYe5OkeW3aKYgkByS5qH31+JcY/C7cCmAdfhFDknbL8+0B3wBsAm4DfhO
4ENgHOKeq7h5va5IOvz3fQbifrKq3V9VHgLcCS4B/Mkr4JrkmyZMT5w632sFJ1iR5sN0fNPTcRUnWJXkgyVlD9VOT3Nu
e+1CStPq+ST7R6rcnOW5ozIr2Gg8mWTHqhyFJM+n5Avj7EwtVtQ14aCeuBXwtsGxS7b3ALVV1AnBLe0ySExmcV/zyNua
KJHu3MVcCK4ET2m1im+cDm6rqpcDlwAfbtg4GLgZeDZwGXDwc9JI0WzxfAJ+c5NvttgU4aWI5ybRfTa6qW4GNk8pnA9e
15euAc4bqN1bVs1X1EIM55tOSHAkcUFW3VVUB108aM7GtTwJntr3js4A1VbWxqjYBa/jRPwSS1N20c8BVtfd0z++Cw6v
qibbtJ5Ic1upHAV8eWm99q32/LU+uT4x5tG1ra5LNwCHD9SnGbCfJSgZ71xx77LG7/q4kaReMfDnKMcsUtZqmvqtjti9
WXVVVS6pqyeLFi0dqVJL2lJkO4G+2aQXa/ZOtvh44Zmi9o4HHW/3oKerbjUmyCDiQwZTHjrYlSbPKTAfwagbnEdPuPzt
UX97ObDiewcG2O9p0xZYkp7f53fMmjZnY1rnAF9o88eeApUkOagfflraaJM0qo34Veacl+TjweuDQJOsZnJnwAWBVkvO
BR4A3AVTVfUlWAfczuMjPBe2sC4B3MjijYn/gpnYDuBq4Ick6Bnu+y9u2NiZ5P3BnW++Sqpp8MFCSuhtbAFfVW3fw1Jk
7WP9S4NIp6muBV0xR/x4twKd47hrgmpGblaQOZstBOElacAxgSerEAJakTgxgSerEAJakTgxgSerEAJakTgxgSerEAJa
kTgxgSerEAJakTgxgSerEAJakTgxgSerEAJakTgxgSerEAJakTgxgSerEAJakTgxgSerEAJakTgxgSerEAJakTgxgSer
EAJakTgxgSerEAJakTgxgSerEAJakTgxgSerEAJakTgxgSerEAJakTgxgSerEAJakTgxgSerEAJakTgxgSerEAJakTgx
gSerEAJakTgxgSerEAJakTgxgSerEAJakTgxgSerEAJakTgxgSerEAJakTgxgSerEAJakTgxgSerEAJakTgxgSerEAJa
kTgxgSeqkSwAn+UaSe5PcnWRtqx2cZE2SB9v9QUPrX5RkXZIHkpw1VD+1bWddkg8lSavvm+QTrX57kuNm/E1K0vPouQf
8C1V1SlUtaY/fC9xSVScAt7THJDkRWA68HFgGXJFk7zbmSmAlcEK7LWv184FNVfVS4HLggzPwfiRpp8ymKYizgeva8nX
AOUP1G6vq2ap6CFgHnJbkSOCAqrqtqgq4ftKYiW19EjhzYu9YkmaLXgFcwOeTfCXJylY7vKqeAGj3h7X6UcCjQ2PXt9p
RbXlyfbsxVbUV2AwcMrmJJCuTrE2y9qmnntojb0ySRrWo0+ueUVWPJzkMWJPk76ZZd6o915qmPt2Y7QtVVwFXASxZsuR
HnpekceqyB1xVj7f7J4HPAKcB32zTCrT7J9vq64FjhoYfDTze6kdPUd9uTJJFwIHAxnG8F0naVTMewElemORFE8vAUuC
rwGpgRVttBfDZtrwaWN7ObDiewcG2O9o0xZYkp7f53fMmjZnY1rnAF9o8sSTNGj2mIA4HPtOOiSOCPlZVNye5E1iV5Hz
gEeBNAFV1X5JVwP3AVuCCqtrWtvVO4Fpgf+CmdgO4GrghyToGe77LZ+KNSdLOmPEArqqvAydPUf974MwdjLkUuHSK+lr
gFVPUv0cLcEmarWbTaWiStKAYwJLUiQEsSZ0YwJLUiQEsSZ0YwJLUiQEsSZ0YwJLUiQEsSZ0YwJLUiQEsSZ0YwJLUiQE
sSZ0YwJLUiQEsSZ0YwJLUiQEsSZ0YwJLUiQEsSZ0YwJLUiQEsSZ0YwJLUiQEsSZ0YwJLUiQEsSZ0YwJLUiQEsSZ0YwJL
UiQEsSZ0YwJLUiQEsSZ0YwJLUiQEsSZ0YwJLUiQEsSZ0YwJLUiQEsSZ0YwJLUiQEsSZ0YwJLUiQEsSZ0YwJLUiQEsSZ0
YwJLUiQEsSZ0YwJLUiQEsSZ0YwJLUiQEsSZ0s6t2ANNdceOGFbNiwgSOOOILLLrusdzuawwxgaSdt2LCBxx57rHcbmgc
MYM16j1zyM71b2M7WjQcDi9i68eFZ1duxv39v7xa0k5wDlqRO3AOWdtKh+z0HbG330q4zgKWd9J6TvtW7Bc0TTkFIUif
zeg84yTLgT4G9gT+vqg90bknSLpivp/7N2wBOsjfwn4BfAtYDdyZZXVX39+1M0s6ar6f+zdsABk4D1lXV1wGS3AicDRj
A0vM44z+e0buF7ezzrX3Yi7149FuPzqrevvRbX9qt8amqPdTK7JLkXGBZVf3L9vhtwKur6l1D66wEVraHLwMemPFG94x
Dgad7N7HA+JnPvLn8mT9dVcsmF+fzHnCmqG3316aqrgKumpl2xifJ2qpa0ruPhcTPfObNx898Pp8FsR44Zujx0cDjnXq
RpB8xnwP4TuCEJMcn2QdYDqzu3JMk/cC8nYKoqq1J3gV8jsFpaNdU1X2d2xqXOT+NMgf5mc+8efeZz9uDcJI0283nKQh
JmtUMYEnqxACe5ZJ8p3cPC1GSSnLDOONFSZ5K8t/a43+W5L07GOs/s50w+fNK8vYkf9aW35HkvOcZ/4P155p5exBO2k3
PAK9Isn9VfZfBV9p/8F3YqlqNZ9WMXVV9uHcP4+Qe8ByU5JQkX05yT5LPJDkoyWFJvtKeP7ntwR3bHv/fJC/o2/WcdBP
wK235rcDHJ56YtJd2fJLbktyZ5P0d+py3kvxBkve05Ve1f+dvS/JHSb46tOpPJLk5yYNJ5szVegzguel64N9W1UnAvcD
FVfUksF+SA4DXAmuB1yZ5MfBkVf1Dv3bnrBuB5Un2A04Cbt/Ben8KXFlVrwI2zFRz88j+Se6euAGX7GC9/wy8o6peA2y
b9NwpwFuAnwHekuQY5gADeI5JciDw41X1V610HfDzbflvgDPa4z9s968F/nqm+5wPquoe4DgGe79/Oc2qZ/DDveMbpll
PU/tuVZ0ycQN+f/IKSX4ceFFV/U0rfWzSKrdU1eaq+h6DC269eJwN7ykG8Pzy1wwC98XAZ4GTgZ8Dbu3Z1By3GvhjhqY
fdsAT6sdrqmu7DHt2aHkbc+T4lgE8x1TVZmBTkte20tuAib3hW4FfBx6squeAjcAvA7t3zbyF7Rrgkqqa7ieHv8Tgq+4
Avzb+lhaeqtoEbElyeistn279uWJO/JVY4F6QZP3Q4z8BVgAfbgfWvg78BkBVfSMJ/HCP94vA0e1fXu2CqlrPYI53Ou8
GPpbk3cCnxt/VgnU+8NEkzwD/C9jct53d51eRJc0JSX6sqr7Tlt8LHFlV7+7c1m5xD1jSXPErSS5ikFsPA2/v287ucw9
YkjrxIJwkdWIAS1InBrAkdWIAS1InBrDmlSS/neRrSf6idy/S8/EsCM0rSf4OeENVPTRUW1RVW/fw64TBfz/P7cntamF
xD1jzRpIPAz8JrE6yOclVST4PXJ9kcZJPtUtG3pnkjDbmkCSfT/K3ST6S5OEkh+5g+8e1vesrgLuAY5JcmWRtkvuSvG9

```
o3W8keV+Su5Lcm+SnWn1xkjWtvt3rJfn1JHeOq4J9JMneY/7I1JkBrHmjqt4BPA78AnA5cCpwdlX9CwZfJ768XTLynwN
/3oZdDHyxql7J4MI7xz7Py7wMuL6qXllVDwP/vqqWMLhc5euSnDSO7tNV9bPAlcB7hl7vC63+mYnXS/LTDC6neEa7Itg
2vK7EvOc34TSfrW6/ZgHwi8CJ7VoZAAckeRGDS3a+EaCq/nuS57tuxsNV9eWhx29OspLBf0tHAicC97TnPt3uvzLxGgy
uTver7fVuHnq9Mxn8wbiz9bg/8OROvFfNQQaw5rNnhpb3Al4zFMgAtLDbmQMhP9hmkuMZ7Nm+qqo2JbkW2G9o3YlLJA5
fHnFHl1UMcF1VXbQTvWiOcwpCC8XngXdNPEhySlu8lfa/+kneAByOE9s8gEEgb05yOPCGEcZ8EXhze72lQ693C3BuksP
acwe3XzPRPGYAa6H4bWBJ+O2x+4F3tPr7gJ9PchewFHhk1A1W1f8G/ha4j8F1g0e57vL7gKXt9d4APAFsqar7gd8DPp/
kHmANgykNzWOehiYNSfINYElVPT2m7e8LbKuqrUlew+C35E4Zx2tp9nMOWJpZxwKrkIgzn
Zyc6sqr+f6X4OfxnAktSJB+EkqRMDWJI6MYAlqRMDWJI6+f8TL/Jt5T/JRQAAAABJRU5ErkJggg==\n",
```

```
      "text/plain": [
       "<Figure size 360x360 with 1 Axes>"
      ]
     },
     "metadata": {
      "needs_background": "light"
     },
     "output_type": "display_data"
    }
   ],
   "source": [
    "sns.catplot(x=\"freq_range\", y=\"Revenue\", kind=\"bar\", data=df_customer_new);"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "9d46ab11",
   "metadata": {},
   "source": [
    "From the graph we can see that the high frequent customers are contrinuting less\n",
    "\n",
    "1.Are they also profitable - what is the profit margin across the buckets"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 35,
   "id": "9af30e97",
   "metadata": {},
   "outputs": [],
   "source": [
    "profit_margin = df_customer_new.groupby('freq_range')"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 36,
   "id": "4838b34e",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "<pandas.core.groupby.generic.DataFrameGroupBy object at 0x000001A31BB8F730>"
      ]
     },
     "execution_count": 36,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "profit_margin"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "af4d4e1c",
   "metadata": {},
   "source": [
    "4. Which customer segment is most profitable in each year ( there is a column called 
customer segment)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 37,
   "id": "8eb744bd",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "Consumer       26518\n",
       "Corporate      15429\n",
       "Home Office     9343\n",
       "Name: Segment, dtype: int64"
      ]
     },
     "execution_count": 37,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "df.Segment.value_counts()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 38,
   "id": "b3b981e9",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "<AxesSubplot:xlabel='Segment', ylabel='count'>"
      ]
     },
     "execution_count": 38,
     "metadata": {},
     "output_type": "execute_result"
    },
    {
     "data": {
      "image/png": 
"iVBORw0KGgoAAAANSUhEUgAAAZIAAAEGCAYAAABPdROvAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24
zLjUuMSwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy/YYfK9AAAACXBIWXMAAAsTAAALEwEAmpwYAAAX50lEQVR4nO3dfbR
ddX3n8fcHgojyIEjEmKChGrVANZaIKO1Ui0uZTmdAB2rQClqmsQzUh+nDQMextiNr6VDFAYUZrJjAiEixCjpqZQE+Ixg
sEgJSU1GIRIiF8mCFruB3/ti/q4fLyc1N9j33cjuMuw0SpLnNIJEk9WK
QSJJ6MUgkSb0YJJKkXuaNuw0zbe+9967FixePuxuSNKdce+21n4e+21P66q+cPu2+6CZPHixaxevXrc3ZCk0SXJDzZ3n4e2JEm
9GCSSpF4MEklSLwaJJKkXg0SS1ItBIknqxSCRJPVikEiSejFIJEm9bHefbN8aB/3JeePuwnbh2t0OHXcXJPXgHokkqR
DRJLUi0EiSerFIJEk9WKQSJJ6MUgkSb0YJJKkXkYWJEn2TXJlkpuSrE3yllZ/Z5IfJrmuTb810OaBP1g5K
safedkSStvn0Sj7f61UkWj+r5SJKGG+UeySbgj6rql4FDgBOT7N/uO72qlrbpswDtvuXAAcDhwFlJdmzLnw2sAJa06fB
WPx64u6qeBZwvGeEz0eSNMTIgqSqNlTVt9r8fcBNwMIpmhwBXFhVD1bVLcA64OA7CTnGGE
TeyuSpNkxK+dI2iGnFwBXt9JJSa5Pcm6SPVttIXDbQLP1rbawzU+uP6xNVW0C7gGePIrnIEkabuRBkmRX4BPAW6vqXrr
DVM8ElgIbgPdOLDqkeU1Rn6N5D6sSLI6yeqNGZdu3ROQJE1ppEGSZCe6EPloVf0tQFXdUVUPVdXPgA8BB7fF1wP7DjR
fBNze6ouG1B/WJsk8YA/grsk/LjlJ94TxHWE3fSG6rRKMN33sf5yCsV2dcm
q2gDcl+SQts5jgUsG2hzX5o8CrmjnUSRJs2SUw8gfCrweWWZb7M+CYJEvpngxEvpDkF9H3gTQFWtTXIRcCPdFV8nVtVDrd0
JwEpgF+BzbYIuqM5Pso5uT2RjPpVVwMHDqk/ABzdo5uSpJ78ZLskqReDRJLUi0E
iSerFIJEk9WKQSJJ6MUgkSb0YJJKkXgwSSVIvBokkqReDRJLUi0EiSerFIJEk9WKQSJJ6MUgkSb0YJJKkXgwSSVIvBok
```

kqReDRJLUi0EiSerFIJEk9WKQSJJ6MUgkSb0YJJKkXgwSSVIvBokkqReDRJLUi0EiSerFIJEk9WKQSJJ6MUgkSb2MLEi
S7JvkyiQ3JVmb5C2tvleSy5J8t/3cc6DNKUnWJbk5ySsH6gclWdPuOyNJWn3nJB9v9auTLB7V85EkDTfKPZJNwB9V1S8
DhwAnJtkfOBm4vKqWAJe327T7lgMHAIcDZyXZsa3rbGAFsKRNh7f68cDdVfUs4HTgPSN8PpKkIUYWJFW1oaq+1ebvA24
CFgJHAKvaYquAI9v8EcCFVfVgVd0CrAMOTrIA2L2qrqqqAs6b6b1GZiXRcDh03srUiSZsesnCNph5xeAFwN7FNVG6ALG+A
pbbGFwG0Dzda32sI2P7n+sDZVtQm4B3jySJ6EJGmokQdJkl2BTwBvrap7p1p0SK2mqE/VZnIfViRZnWT1xo0bt9RlSdJ
WGGmQJNmJLkQ+WlV/28p3tMNVtJ93tvp6YN+B5ouA21t90ZD6w9okmQfsAdw1uR9VdU5VLauqZfPnz5+JpyZJakZ51Va
ADwM3VdX7Bu66FDiuzR8HXDJQX96uxNqP7qT6Ne3w131JDmnrPHZSm4l1HQVc0c6jSJJmybwRrvtQ4PXAmiTXtdqfAe8
GLkpyPHArcDRAVa1NchFwI90VXydW1UOt3QnASmAX4HNtgi6ozk+yjm5PZPkIn48kaYiRBUlVfZXh5zAADttMm1OBU4f
UVwMHDqk/QAsiSdJ4+Ml2SVIvBokkqReDRJLUi0EiSerFIJEk9WKQSJJ6MUgkSb0YJJKkXgwSSVIvBokkqReDRJLUi0E
iSerFIJEk9WKQSJJ6MUgkSb0YJJKkXgwSSVIvBokkqReDRJLUi0EiSerFIJEk9WKQSJJ6MUgkSb0YJJKkXgwSSVIvBok
kqReDRJLUi0EiSerFIJEk9WKQSJJ6mTfuDkijcutf/sq4u/CY9/R3rBl3F/QoMK09kiSXT6c26f5zk9yZ5IaB2juT/DD
JdW36rYH7TkmyLsnNSV45UD8oyZp23xlJ0uo7J/l4q1+dZPF0noskaWZNGSRJHp9kL2DvJHjsm2atNi4GnbWHdK4HDh9R
Pr6qlbfpse5z9geXAAa3NWUl2bMufDawAlrRpYp3HA3dX1bOA04H3bKE/kqQR2NIeyZuAa4Hntp8T0yXAB6dqWFVfVfBu6
aZj+OAC6sqger6hZgHXBwkgXA7lV1VVUVcB5w5ECbVW3+YuCwib0VSdLsmTJIqup/VdV+wB9X1TOrar82vVdV/+wB9X1S9V1X5ten5VfWAbH/O
kJNe3Q197ttpC4LaBZda32sI2P7n+sDZVtQm4B3jysAdMsiLJ6iSrN27cuI3dliQNM61zJFV1ZpKXJHltkmMnpm14vLO
BZwJLgQ3Ae1t92J5ETVGfqs0ji1XnVNWyqlo2f/78reqwJGlq07pqK8n5dAFwHfBQK08capq2qrpjYJ0fAj7Tbq4H9h1
YdBFwe6svGlIfbLM+ytTxgD6Z/KE2SNEOme/nvMmD/dp5imyVZUFUb2s1XARNXdF0KXJDkfXQn8S3IIcDV
wLHDmQJvjgKuAo4Ar+vZPkrT1phskNwBPpTscNS1JPga8l06Kr/XAnwMvTbkKUbm/m+3Qn86mqtUkuAm4ENgEnVtXEns8
JdFeA7QJ8rk0AHwbOT7KObk9k+XT7JkmaOdMNkr2BG5C5NcAzw4Uayq/7C5BlV1zJDyh6dY/lTg1CH11cCBQ+oPAEdP3W1
J0qhNN0jeOcpOSJLmrmkFSV9adQdkSTNTdO9aus+fnFp7eOAnYCfVNXuo+qYJGlumO0Z4jeeW6Dt5McCRw8kuaWbRp
Gvqo+BfzmzHZFkjQXTffQ1qsHbj6O7nslW/PJCknSDmq6V2399sD8JuBHdN8+lyS9xk33HMkbRt0RSdLcNN3fSFiU5JNJ7kzy4yY+kWTRlltKknZ00z3Z/hG6gRWfRjeQ4j+0miTpMW66QTK/qj5SVZvatBJw3HVJ0rSD5CdJXpfkcW16HfBvo+yYJGlumG6Q/D7wO8C/0P1uyFF0Y3tJkh7jpnv5718A/1RVdwO0Hqe4Iu62lmrTpOH8n8aG+fyw2zV7G1LPNP96hK0nOC9l7Vdc7XFEhOHO5RnVr6UmZHD/3KYuWtyvQDkAAlCqVSoP4D4W8N/Iox8ZXCuY/RQeIR2amuTR6N2uf1rO+IQrPQ
IFMw46ZZs9Wb0lUDDq6zw5FgdUjmHjwPDbdbhtbqrjWJA10ewSgqeDFEcvUDBZy58DMKUxbxHuB8ByngyLbLWdD3PkyADJkhj9aXSDWu1lDDkRmMFA+qRSO0CE8s9rJbJAbYRRhcMI7OnNXO6NaYyqDZ7rfkWQzJwMgdtQYzwj5I+U1a4h7A8gA
ULQQ/sbv63bh+2ZxPc4fOx7mXmb2uByYkiQ8Dn99AhOsndfgGRhxIGfzvTYf4R8fnpXkSBY6AzdSi12e/cLGgHAuZnBgfjjsz
5I9qJchLWRA0fXqq6brPgTk/3PE4h3PxM8pOqN4w3YpL9Vn8/ygI9ABJsLqScXu6wYfrVvVkApfYH0Eal9jqzI/AxZLiAfh8
b4N1g5xc8dAj5w2DjxKPbfY4TnthNLZL41ZBOYgJ6+Fvd8rGBQf3Zuvp8u8AXlHl1NxPPxAK6/GJSPJ5gQpixE+Xh1p/9vwBjbz3gBv8Lh2bQUjFfLUCwtVWKVu1jcO5X6LIngHYKwJ8Eh91jRDEFUeQwbN0jr2ShjB/bHTKgVhk+cZgbnrUudzb2x
QKBhkfZ1T0IT8+hdfe9LqGtIBDYnW0y4gHm8UrzMfhYTXBdewWv5xjBwMmezRcYEyq2ZqDGnSYuf5szHp8rqHpGNLT9e6P8
JFU2FEVoL7mPYMAffdEhhBBGkIHthmTiQJ8+WU6h1Nic1q7SFDfUuTUxqF/IjmNoz8fx4Aiu7W+gb9MrOaZM3hoBeUYlEdDX
Pdj0XfZDuZ6mKnwz9Q2NRE6wMqVJBmHJ+VbN3w5pAUdOTUvoQJe7m1VIyshpzNwRs8BQVHHcUHkmy7YvPLdV4Xw9B0kQYe
DV5jANvIAZhYlQFmSIQkYs5Ev/BZQGWe87cbv57ILhiXkETkdk+xZgNMrkcwkvHFCvtfk/Dj7SrSSEtnP7Xqwf4kl7gZ5nUn
hNo7gxUQqP+5qEhR3fyvJeNNZu3Ob7QHgUnRa8VVTTY9Xn2kz3sp+Fr9Lj7SOrYrt7GZMYPfYbEx1bFHv6wZtUQnRjpV

```
R7lDUPLoTqf97M93478DVwA/oDktOHHq8EPhQkjfT7TW9Djg7ydvp3lBc2Pq3vXszcG6SPwE2Am+cbsOq2tj2Aj+WZOJ
Q4dt55BV7ZwJ7AmuSPAT8CDhiaw4NV9W/JnkNcGYLpJ/SHXr+a7rDbN9qRx02Ol2xN1YOkSLNkiSLgc9U1YHj7os0kzy
OJUnqxTOSSVIv7pFIknoxSCRJvRgkkqReDBJpGyX5b0nbOnWtpFbrvvyonH3aH+H3aUKSxUleO+5+DBrVZhrJLHNZNSZDxPpCL+aqulZWlHqY
J9ubhIwmM22LgtWzdh+CkbeIeibRtFgA/rqoHAarqx1V1e5KDknwpybbVJ/i7JAug+Ad/2XK5KclqSG1r9DUk+leTTSW5
JclKS/5Lk75N8I8lebblnJvl8W+9XJj7smGRlkjOSfD3J9wbGYHo38OttT+9ZaL+ltj+i9NIMMEmnbfAHYN90XF52V5DfasBV
nOo3CexDd6L6ntuU/AvxBVb2YbsymQQfS7TOc3Jb/l6p6Ad2AgMe2Zc6hG2PpILoRg88aaL+AbgiO36YLEICTga9U1dK
qOn3GnrUOhIe2pG1QVfcnOYjuOyReRjdC77voQuGyNtDljnTjYj0J2K2qvt6aXOD3oj/hyqq6D7gvyT10Q51DNwTK86Y
xmu+nqupnwI1J9pnZZyptmUEibaM2bPwBgS+2ocdPBNa2vY6L5nFla1pZGZtzSa72D7LY0kLMO4D21J2yDJc5IsGSg
tpRuQcf7E6L5Jdkr39ap30+1tHNKWXc5W2MbRfH8+GrA0agaJtG12BVYlubGNlrw/3deoHgW8J8m36b6d7yVt+eOBc5J
cRbfXcM9WPt7rgOPbetfSfU3AVK4HNiX5X5tifbNWqOtSXNgiS7VtX9bf5kYEFVvWXM3ZJmhOdIpNnx75JUi8GiSSpF4NEktSLQSJJ6sUgkST18v8B3No4R6Dc2mkAAAAASUVORK5CYII=\n",
      "text/plain": [
       "<Figure size 432x288 with 1 Axes>"
      ]
     },
     "metadata": {
      "needs_background": "light"
     },
     "output_type": "display_data"
    }
   ],
   "source": [
    "sns.countplot(x=\"Segment\",data = df)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "ecd53c42",
   "metadata": {},
   "outputs": [],
   "source": []
  }
 ],
 "metadata": {
  "kernelspec": {
   "display_name": "Python 3 (ipykernel)",
   "language": "python",
   "name": "python3"
  },
  "language_info": {
   "codemirror_mode": {
    "name": "ipython",
    "version": 3
   },
   "file_extension": ".py",
   "mimetype": "text/x-python",
   "name": "python",
   "nbconvert_exporter": "python",
   "pygments_lexer": "ipython3",
   "version": "3.9.12"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 5
}
```