```
#level 1 project 2

#customer segmentation analysis

#importing necessary libraries

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns


print("1.data collection \n")

# 1.data loading & cleaning

file_path = 'C:/Users/rahul/OneDrive/Desktop/python/task-2/ifood_df.csv'

df = pd.read_csv(file_path)

print(df.head())


#printing all data columns

print("checking all columns in our dataframe \n")

print(df.columns )
```

output—

```
1.data collection


   Income  Kidhome  Teenhome  ...  MntTotal  MntRegularProds  AcceptedCmpOverall

0  58138.0        0         0  ...      1529             1441                   0

1  46344.0        1         1  ...        21               15                   0

2  71613.0        0         0  ...       734              692                   0

3  26646.0        1         0  ...        48               43                   0

4  58293.0        1         0  ...       407              392                   0


[5 rows x 39 columns]

checking all columns in our dataframe
```

```
Index(['Income', 'Kidhome', 'Teenhome', 'Recency', 'MntWines', 'MntFruits',
       'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts',
       'MntGoldProds', 'NumDealsPurchases', 'NumWebPurchases',
       'NumCatalogPurchases', 'NumStorePurchases', 'NumWebVisitsMonth',
       'AcceptedCmp3', 'AcceptedCmp4', 'AcceptedCmp5', 'AcceptedCmp1',
       'AcceptedCmp2', 'Complain', 'Z_CostContact', 'Z_Revenue', 'Response',
       'Age', 'Customer_Days', 'marital_Divorced', 'marital_Married',
       'marital_Single', 'marital_Together', 'marital_Widow',
       'education_2n Cycle', 'education_Basic', 'education_Graduation',
       'education_Master', 'education_PhD', 'MntTotal', 'MntRegularProds',
       'AcceptedCmpOverall'],
      dtype='object')
```

```
#2.data exploration and cleaning


#looking for missing value
print("looking for missing value \n")
print(df.isna().sum())


#uniqueness
print("uniqueness \n")
print(df.nunique())


#Data Exploration


print("data exploration")
plt.figure(figsize=(6, 4))
sns.boxplot(data=df, y='MntTotal')
plt.title('Box Plot for MntTotal')
plt.ylabel('MntTotal')
plt.show()
```

```
#Outliers


print("outliers")
Q1 = df['MntTotal'].quantile(0.25)
Q3 = df['MntTotal'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
outliers = df[(df['MntTotal'] < lower_bound) | (df['MntTotal'] > upper_bound)]
print(outliers.head())
```

output—

2.data exploration and cleaning

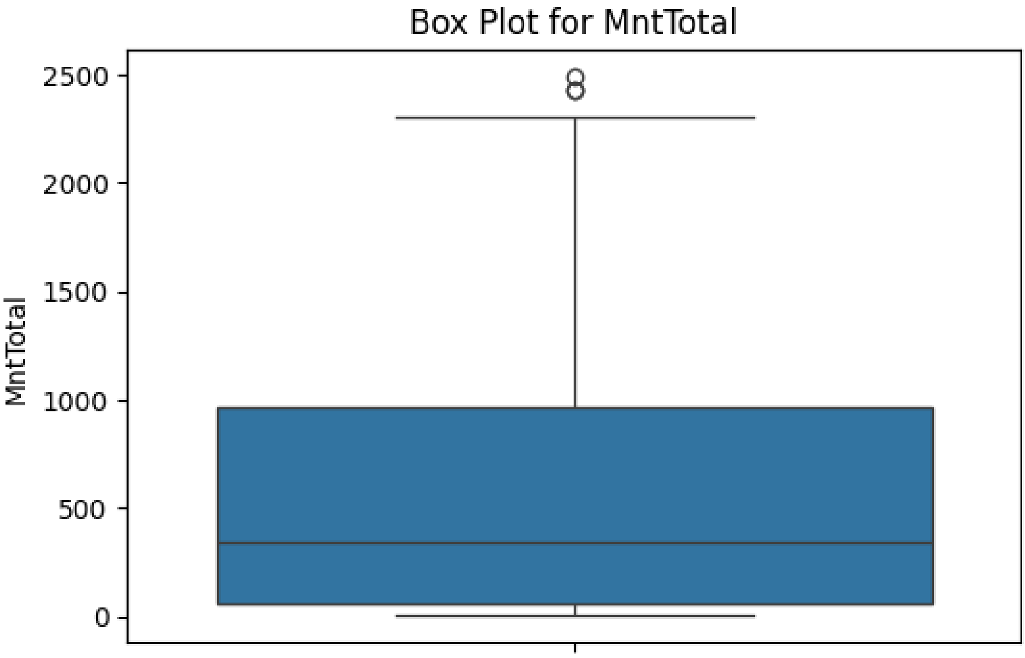
looking for missing value

```
Income                0
Kidhome               0
Teenhome              0
Recency               0
MntWines              0
MntFruits             0
MntMeatProducts       0
MntFishProducts       0
MntSweetProducts      0
MntGoldProds          0
NumDealsPurchases     0
NumWebPurchases       0
NumCatalogPurchases   0
NumStorePurchases     0
NumWebVisitsMonth     0
AcceptedCmp3          0
```

```
AcceptedCmp4          0
AcceptedCmp5          0
AcceptedCmp1          0
AcceptedCmp2          0
Complain              0
Z_CostContact         0
Z_Revenue             0
Response              0
Age                   0
Customer_Days         0
marital_Divorced      0
marital_Married       0
marital_Single        0
marital_Together      0
marital_Widow         0
education_2n Cycle    0
education_Basic       0
education_Graduation  0
education_Master      0
education_PhD         0
MntTotal              0
MntRegularProds       0
AcceptedCmpOverall    0
dtype: int64
uniqueness

Income                 1963
Kidhome                   3
Teenhome                  3
Recency                 100
MntWines                775
```

```
MntFruits             158
MntMeatProducts        551
MntFishProducts        182
MntSweetProducts       176
MntGoldProds           212
NumDealsPurchases       15
NumWebPurchases         15
NumCatalogPurchases      13
NumStorePurchases       14
NumWebVisitsMonth        16
AcceptedCmp3             2
AcceptedCmp4             2
AcceptedCmp5             2
AcceptedCmp1             2
AcceptedCmp2             2
Complain                 2
Z_CostContact            1
Z_Revenue                1
Response                 2
Age                     56
Customer_Days          662
marital_Divorced         2
marital_Married          2
marital_Single           2
marital_Together         2
marital_Widow            2
education_2n Cycle       2
education_Basic          2
education_Graduation      2
education_Master         2
education_PhD            2
```

```
MntTotal            897
MntRegularProds       974
AcceptedCmpOverall      5
dtype: int64
data exploration
```



```
outliers
      Income  Kidhome  Teenhome  ...  MntTotal  MntRegularProds  AcceptedCmpOverall
1159  90638.0        0         0  ...      2429             2333                   1
1467  87679.0        0         0  ...      2491             2458                   3
1547  90638.0        0         0  ...      2429             2333                   1

[3 rows x 39 columns]
```

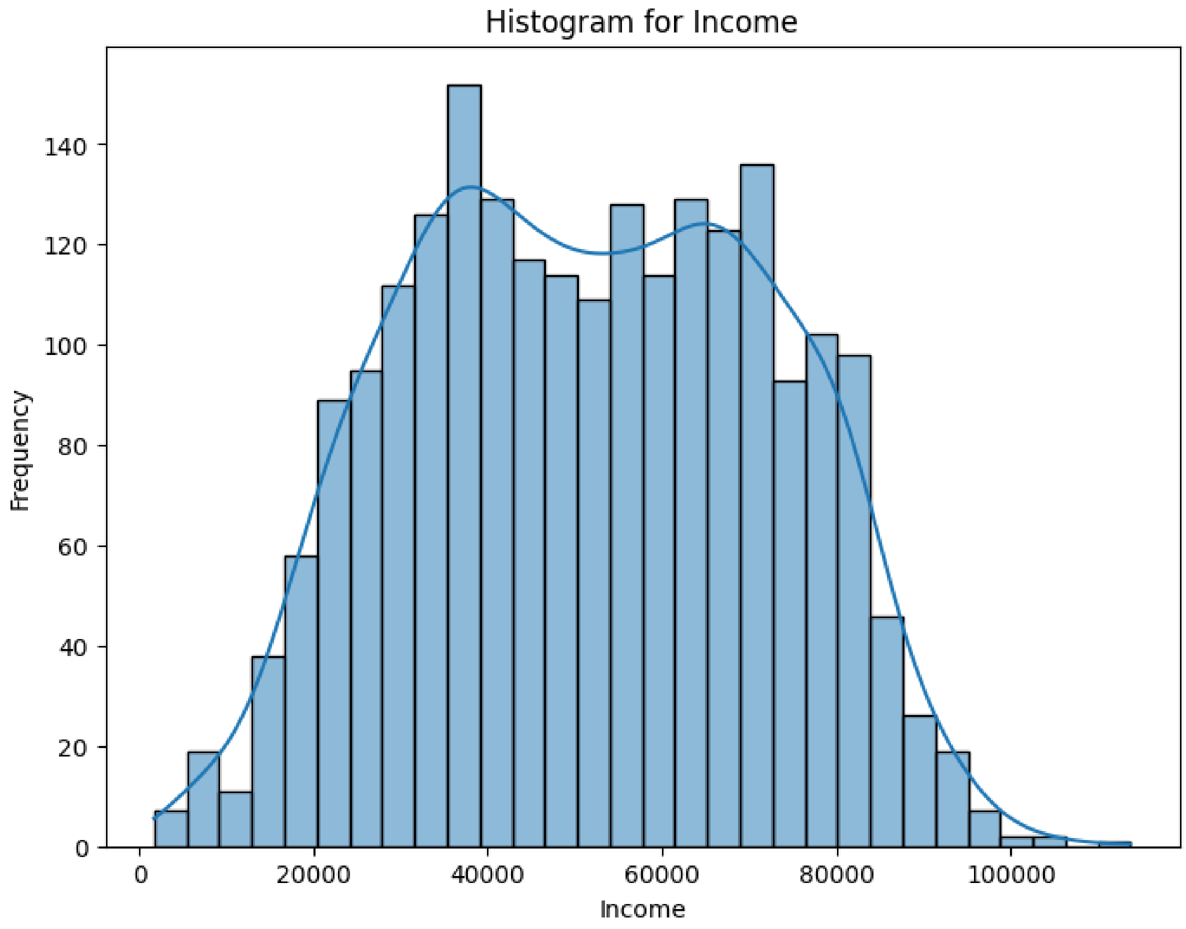
```
# 3.Calculate Average Purchase Value
print("3.Descriptive Statistics")
transactions = pd.DataFrame(df)
total_amount_spent = transactions['Income'].sum()
total_transactions = transactions.shape[0]
average_purchase_value = total_amount_spent / total_transactions
```

```
print("Average Purchase Value:", average_purchase_value)
```

output—

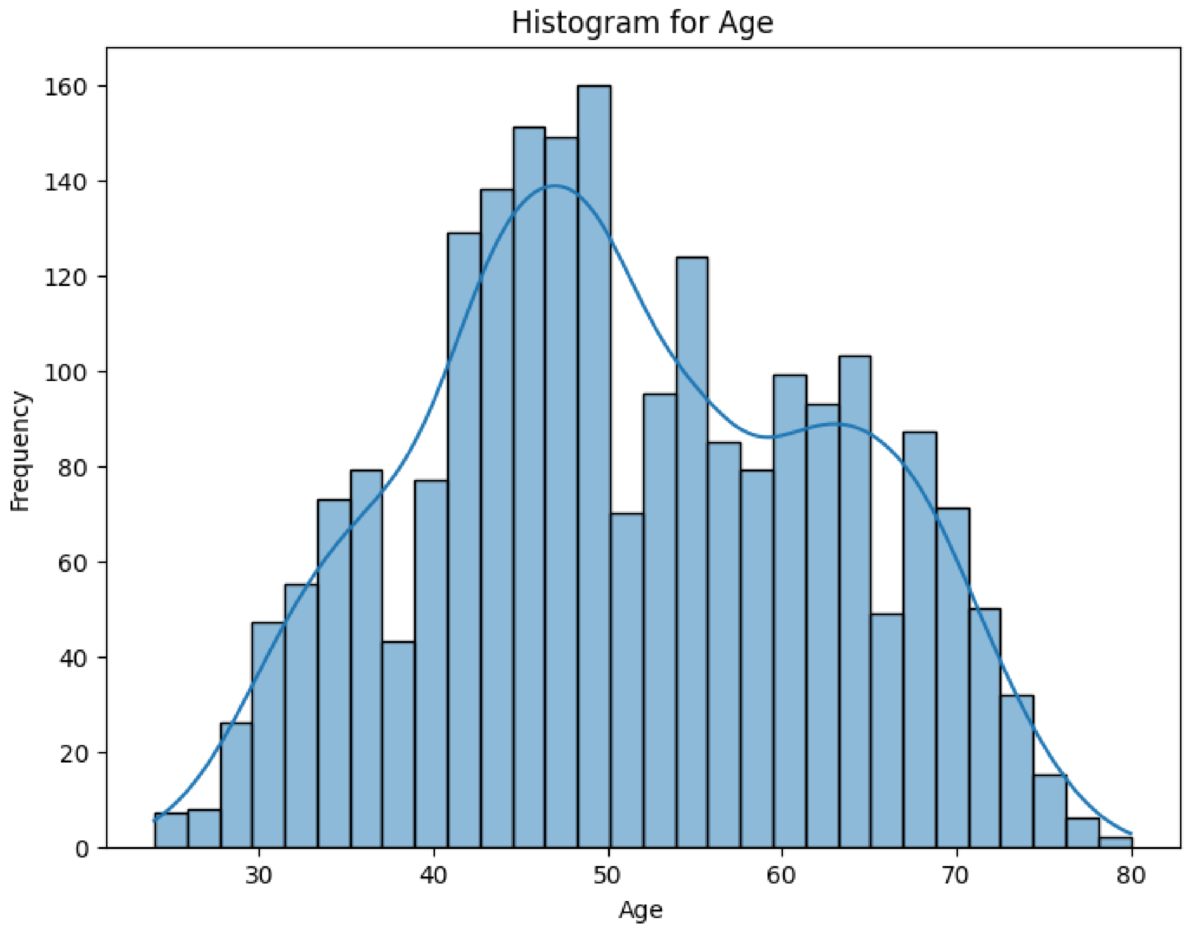
```
Average Purchase Value: 51622.0947845805
```

```
#4.visualization
print("4.visualization")
#histogram for income
print("Hisotogram for income")
plt.figure(figsize=(8, 6))
sns.histplot(data=df, x='Income', bins=30, kde=True)
plt.title('Histogram for Income')
plt.xlabel('Income')
plt.ylabel('Frequency')
plt.show()
```



```
#histogram for age

print("histogram for age")

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

sns.histplot(data=df, x='Age', bins=30, kde=True)

plt.title('Histogram for Age')

plt.xlabel('Age')

plt.ylabel('Frequency')

plt.show()
```

output—



```
#K-Means Clustering

print("5.k-means clustering")

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

cols_for_clustering = ['Income', 'MntTotal']

data_scaled = df.copy()

data_scaled[cols_for_clustering] = scaler.fit_transform(df[cols_for_clustering])

print(data_scaled[cols_for_clustering].describe())
```

output—

```
5.k-means clustering

             Income      MntTotal

count  2.205000e+03  2.205000e+03

mean   2.255691e-17 -3.705778e-17
```

```
std    1.000227e+00  1.000227e+00
min   -2.409272e+00 -9.704038e-01
25%   -7.932106e-01 -8.800957e-01
50%   -1.618161e-02 -3.816642e-01
75%    8.044529e-01  6.968235e-01
max    2.999363e+00  3.348757e+00
```

```
print("6.insights and recommendations")


print("1.We can Calculate the average purchase value by summing up all purchase amounts and dividing by the total number of transactions")
print("2.We can Visualize the distribution using histograms or box plots to identify any patterns or anomalies")
```

output—

```
6.insights and recommendations
1.We can Calculate the average purchase value by summing up all purchase amounts and dividing by the total number of transactions
2.We can Visualize the distribution using histograms or box plots to identify any patterns or anomalies
```