In [103…

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [104…

```
df=pd.read_csv("ifood_df.csv")
```

In [105…

```
df.head()
```

Out[105…

	Income	Kidhome	Teenhome	Recency	MntWines	MntFruits	MntMeatProducts	MntFishProducts	MntSweetProducts	MntGoldF
0	58138.0	0	0	58	635	88	546	172	88	
1	46344.0	1	1	38	11	1	6	2	1	
2	71613.0	0	0	26	426	49	127	111	21	
3	26646.0	1	0	26	11	4	20	10	3	
4	58293.0	1	0	94	173	43	118	46	27	

5 rows × 39 columns

In [106…

```
df.tail()
```

Out[106…

	Income	Kidhome	Teenhome	Recency	MntWines	MntFruits	MntMeatProducts	MntFishProducts	MntSweetProducts	MntGo
2200	61223.0	0	1	46	709	43	182	42	118	
2201	64014.0	2	1	56	406	0	30	0	0	
2202	56981.0	0	0	91	908	48	217	32	12	
2203	69245.0	0	1	8	428	30	214	80	30	
2204	52869.0	1	1	40	84	3	61	2	1	

5 rows × 39 columns

In [107…

```
df.shape
```

Out[107…

```
(2205, 39)
```

In [108…

```
#CHECKING COLUMNS
df.columns.tolist()
```

Out[108…

```
['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']
```

In [9]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2205 entries, 0 to 2204
Data columns (total 39 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Income                2205 non-null   float64
 1   Kidhome               2205 non-null   int64  
 2   Teenhome              2205 non-null   int64  
 3   Recency               2205 non-null   int64  
 4   MntWines              2205 non-null   int64  
 5   MntFruits             2205 non-null   int64  
 6   MntMeatProducts       2205 non-null   int64  
 7   MntFishProducts       2205 non-null   int64  
 8   MntSweetProducts      2205 non-null   int64  
 9   MntGoldProds          2205 non-null   int64  
 10  NumDealsPurchases     2205 non-null   int64  
 11  NumWebPurchases       2205 non-null   int64  
 12  NumCatalogPurchases   2205 non-null   int64  
 13  NumStorePurchases     2205 non-null   int64  
 14  NumWebVisitsMonth     2205 non-null   int64  
 15  AcceptedCmp3          2205 non-null   int64  
 16  AcceptedCmp4          2205 non-null   int64  
 17  AcceptedCmp5          2205 non-null   int64  
 18  AcceptedCmp1          2205 non-null   int64  
 19  AcceptedCmp2          2205 non-null   int64  
 20  Complain              2205 non-null   int64  
 21  Z_CostContact         2205 non-null   int64  
 22  Z_Revenue             2205 non-null   int64  
 23  Response              2205 non-null   int64  
 24  Age                   2205 non-null   int64  
 25  Customer_Days         2205 non-null   int64  
 26  marital_Divorced      2205 non-null   int64  
 27  marital_Married       2205 non-null   int64  
 28  marital_Single        2205 non-null   int64  
 29  marital_Together      2205 non-null   int64  
 30  marital_Widow         2205 non-null   int64  
 31  education_2n Cycle    2205 non-null   int64  
 32  education_Basic       2205 non-null   int64  
 33  education_Graduation  2205 non-null   int64  
 34  education_Master      2205 non-null   int64  
 35  education_PhD         2205 non-null   int64  
 36  MntTotal              2205 non-null   int64  
 37  MntRegularProds       2205 non-null   int64  
 38  AcceptedCmpOverall    2205 non-null   int64  
dtypes: float64(1), int64(38)
memory usage: 672.0 KB
```

In [109…

```
df.describe(include='all')
```

Out[109…

	Income	Kidhome	Teenhome	Recency	MntWines	MntFruits	MntMeatProducts	MntFishProducts	MntS
count	2205.000000	2205.000000	2205.000000	2205.000000	2205.000000	2205.000000	2205.000000	2205.000000	
mean	51622.094785	0.442177	0.506576	49.009070	306.164626	26.403175	165.312018	37.756463	
std	20713.063826	0.537132	0.544380	28.932111	337.493839	39.784484	217.784507	54.824635	
min	1730.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	35196.000000	0.000000	0.000000	24.000000	24.000000	2.000000	16.000000	3.000000	
50%	51287.000000	0.000000	0.000000	49.000000	178.000000	8.000000	68.000000	12.000000	
75%	68281.000000	1.000000	1.000000	74.000000	507.000000	33.000000	232.000000	50.000000	
max	113734.000000	2.000000	2.000000	99.000000	1493.000000	199.000000	1725.000000	259.000000	

8 rows × 39 columns

In [110…

```
#CHECKING FOR MISSING VALUES
df.isnull().sum()
```

Out[110…

```
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
```

In [15]:

```
#CHECKING FOR DUPLICATE VALUES
df.nunique()
```

Out[15]:
```
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
```

In [111…
```
df.size
```

Out[111…
```
85995
```

In [112…
```
df.isnull()
```

Out[112…

	Income	Kidhome	Teenhome	Recency	MntWines	MntFruits	MntMeatProducts	MntFishProducts	MntSweetProducts	MntGc
0	False	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	...	
2200	False	False	False	False	False	False	False	False	False	
2201	False	False	False	False	False	False	False	False	False	
2202	False	False	False	False	False	False	False	False	False	
2203	False	False	False	False	False	False	False	False	False	
2204	False	False	False	False	False	False	False	False	False	

2205 rows × 39 columns

In [22]:
```
df.dtypes
```

Out[22]:

```
Income                  float64
Kidhome                   int64
Teenhome                  int64
Recency                   int64
MntWines                  int64
MntFruits                 int64
MntMeatProducts           int64
MntFishProducts           int64
MntSweetProducts          int64
MntGoldProds              int64
NumDealsPurchases         int64
NumWebPurchases           int64
NumCatalogPurchases       int64
NumStorePurchases         int64
NumWebVisitsMonth         int64
AcceptedCmp3              int64
AcceptedCmp4              int64
AcceptedCmp5              int64
AcceptedCmp1              int64
AcceptedCmp2              int64
Complain                  int64
Z_CostContact             int64
Z_Revenue                 int64
Response                  int64
Age                       int64
Customer_Days             int64
marital_Divorced          int64
marital_Married           int64
marital_Single            int64
marital_Together          int64
marital_Widow             int64
education_2n Cycle        int64
education_Basic           int64
education_Graduation      int64
education_Master          int64
education_PhD             int64
MntTotal                  int64
MntRegularProds           int64
AcceptedCmpOverall        int64
dtype: object
```

In [113…

```
#EXPLORATORY DATA ANALYSIS
#lets find average age of customers

average = df.Age.mean()
average
```

Out[113…

```
np.float64(51.09569160997732)
```

In [114…

```
# Let's check as per the age increase do the customer increases
age_df = df['Age'].value_counts().sort_values(ascending=True)
age_df
```

Out[114…

```
Age
79     1
80     1
24     2
26     3
25     5
27     5
77     6
76     7
75     8
28    13
29    15
74    16
73    16
30    18
72    21
33    27
70    29
32    29
31    29
71    29
35    32
67    35
59    35
36    38
39    38
40    39
63    41
56    41
34    41
37    41
69    42
38    43
53    44
57    44
58    44
65    48
66    49
60    49
54    50
61    50
43    50
52    51
68    52
62    52
41    53
64    55
46    69
51    70
47    71
55    74
50    75
42    76
48    78
45    82
49    85
44    88
Name: count, dtype: int64
```

In [115…

```
age_df = df.groupby('Age')[['Age']].count()
age_df
```

Out[115…

	Age
Age	
24	2
25	5
26	3
27	5
28	13
29	15
30	18
31	29
32	29
33	27
34	41

35	32
36	38
37	41
38	43
39	38
40	39
41	53
42	76
43	50
44	88
45	82
46	69
47	71
48	78
49	85
50	75
51	70
52	51
53	44
54	50
55	74
56	41
57	44
58	44
59	35
60	49
61	50
62	52
63	41
64	55
65	48
66	49
67	35
68	52
69	42
70	29
71	29
72	21
73	16
74	16
75	8
76	7
77	6
79	1
80	1

In [116…

```
#Campaign Acquisition
# how many customers are accepted offer in campaign 1 as per the total customers
campaign1 = df.AcceptedCmp1.sum()/len(df)*100
print('Customer Offer Aquisition on campaign 1 is {:.5f}%'.format(campaign1))
```

```
Customer Offer Aquisition on campaign 1 is 6.43991%
```

In [117…

```
# how many customers are accepted offer in campaign 2 as per the total
```

```
Campaign2 = df.AcceptedCmp2.sum()/len(df)*100
print('Customer Offer Aquisition on campaign 2 is {:.3f}%'.format(campaign2))
```

```
Customer Offer Aquisition on campaign 2 is 6.440%
```

In [118…

```
campaign3 = df.AcceptedCmp1.sum()/len(df)*100
print('Customer Offer Aquisition on campaign 3 is {:.2f}%'.format(campaign3))
```

```
Customer Offer Aquisition on campaign 3 is 6.44%
```

In [119…

```
campaign4 = df.AcceptedCmp1.sum()/len(df)*100
print('Customer Offer Aquisition on campaign 4 is {:.4f}%'.format(campaign4))
```

```
Customer Offer Aquisition on campaign 4 is 6.4399%
```

In [120…

```
campaign5 = df.AcceptedCmp1.sum()/len(df)*100
print('Customer Offer Aquisition on campaign 5 is {:.6f}%'.format(campaign5))
```

```
Customer Offer Aquisition on campaign 5 is 6.439909%
```

In [121…

```
#average Complaint rate
complaint_ratio = len(df[df['Complain']==1]/len(df))
print('complaint ratio of company is {:.3f}%'.format(complaint_ratio*100))
```

```
complaint ratio of company is 2000.000%
```

In [122…

```
#Average reccency
averagerecency = df.Recency.mean()
print('Average purchase of Day is {:.5f}%'.format(averagerecency))
```

```
Average purchase of Day is 49.00907%
```

In [123…

```
#Average Purchase on various Outlets
discount_purchase = df['NumDealsPurchases'].mean()
catalog_purchase  = df['NumCatalogPurchases'].mean()
web_purchase      = df['NumWebPurchases'].mean()
store_purchase    = df['NumStorePurchases'].mean()
print('''Average purchase with Discount : {}
Average purchase with Catalog : {}
Average purchase on Website : {}
Average purchase On Store : {}'''. format(discount_purchase, catalog_purchase, web_purchase, store_purchase))
```

```
Average purchase with Discount : 2.3183673469387753
Average purchase with Catalog : 2.6453514739229025
Average purchase on Website : 4.100680272108844
Average purchase On Store : 5.823582766439909
```

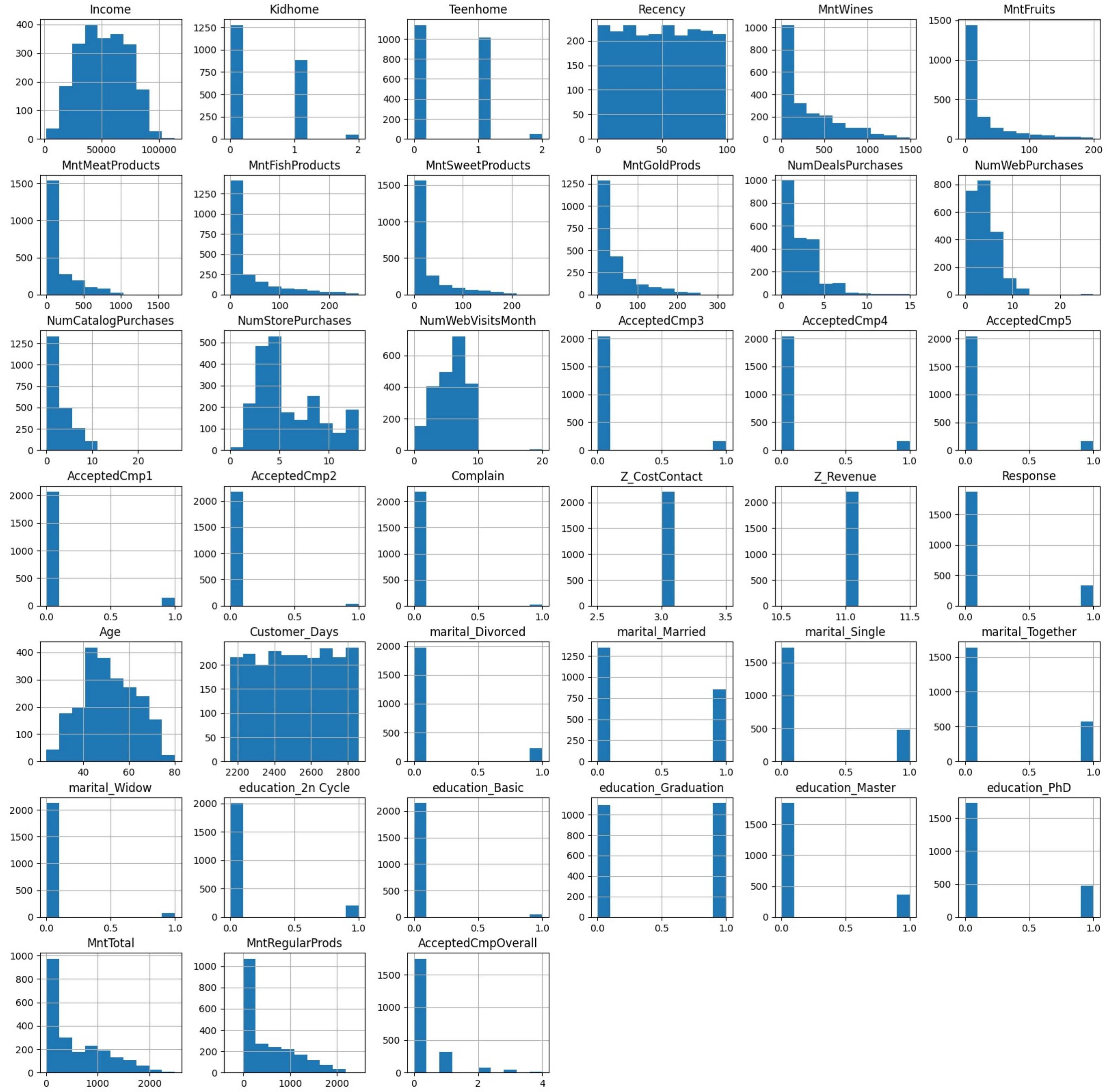
In [124…

```
#Average Amount spent on different products
fish_products = df['MntFishProducts'].mean()
meat_products = df['MntMeatProducts'].mean()
fruits        = df['MntFruits'].mean()
sweet_products= df['MntSweetProducts'].mean()
gold_products = df['MntGoldProds'].mean()
wines         = df['MntWines'].mean()
print('''Average amount on fish products : {}
Average amount on meat products : {}
Average amount on fruits  : {}
Average amount on sweet products : {}
Average amount on gold products : {}
Average amount on wines : {}'''. format(fish_products, meat_products, fruits, sweet_products, gold_products, wi
```

```
Average amount on fish products : 37.756462585034015
Average amount on meat products : 165.31201814058957
Average amount on fruits  : 26.4031746031746
Average amount on sweet products : 27.128344671201813
Average amount on gold products : 44.05714285714286
Average amount on wines : 306.16462585034014
```

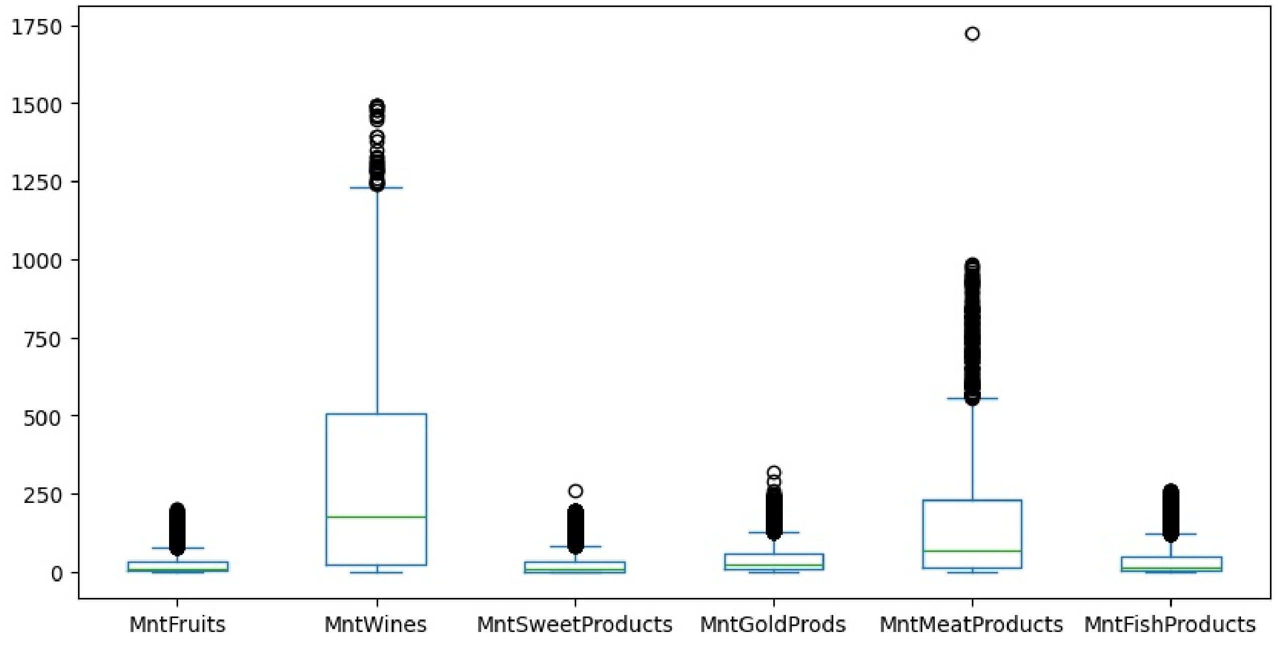
In [133…

```
#plot histograms for each variable
df.hist(figsize=(20,20))
plt.show()
```



In [150…

```
numerical_columns = ['MntFruits', 'MntWines', 'MntSweetProducts', 'MntGoldProds', 'MntMeatProducts', 'MntFishPr
df[numerical_columns].plot(kind = 'box', figsize=(10,5))
plt.show()
```

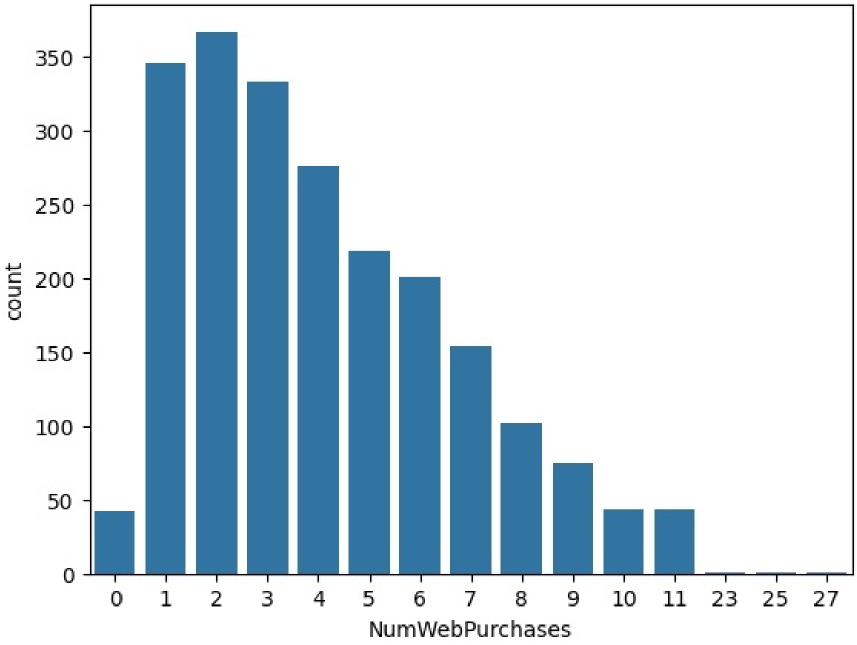


In [161…

```
sns.countplot(x ='NumWebPurchases',  data = df)
```

```
plt.show
```

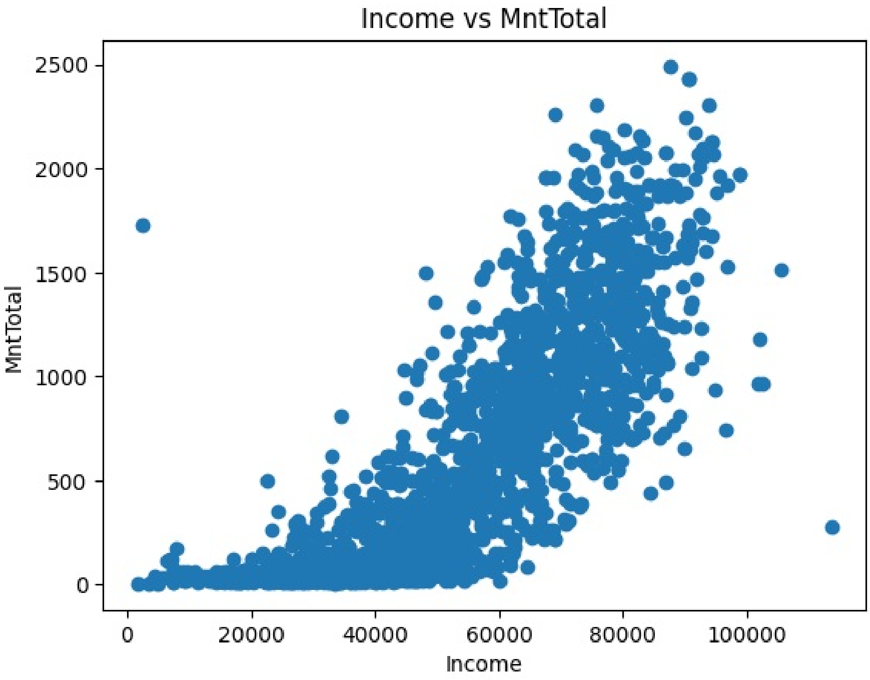
Out[161… `<function matplotlib.pyplot.show(close=None, block=None)>`



In [137…

```
plt.scatter(df['Income'], df['MntTotal'])
plt.xlabel('Income')
plt.ylabel('MntTotal')
plt.title('Income vs MntTotal')
```

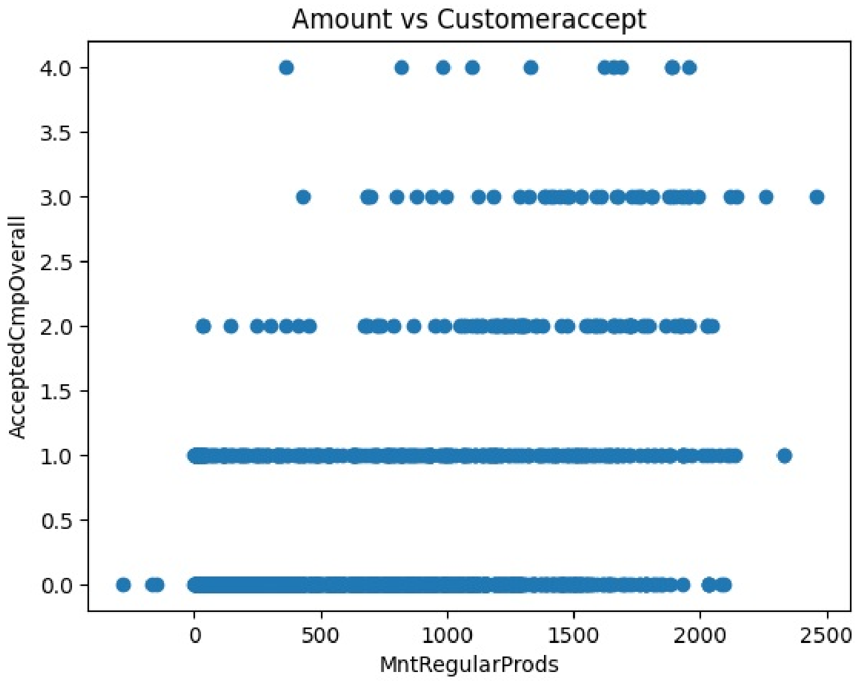
Out[137… `Text(0.5, 1.0, 'Income vs MntTotal')`



In [139…

```
plt.scatter(df['MntRegularProds'], df['AcceptedCmpOverall'])
plt.xlabel('MntRegularProds')
plt.ylabel('AcceptedCmpOverall')
plt.title('Amount vs Customeraccept')
```

Out[139… `Text(0.5, 1.0, 'Amount vs Customeraccept')`

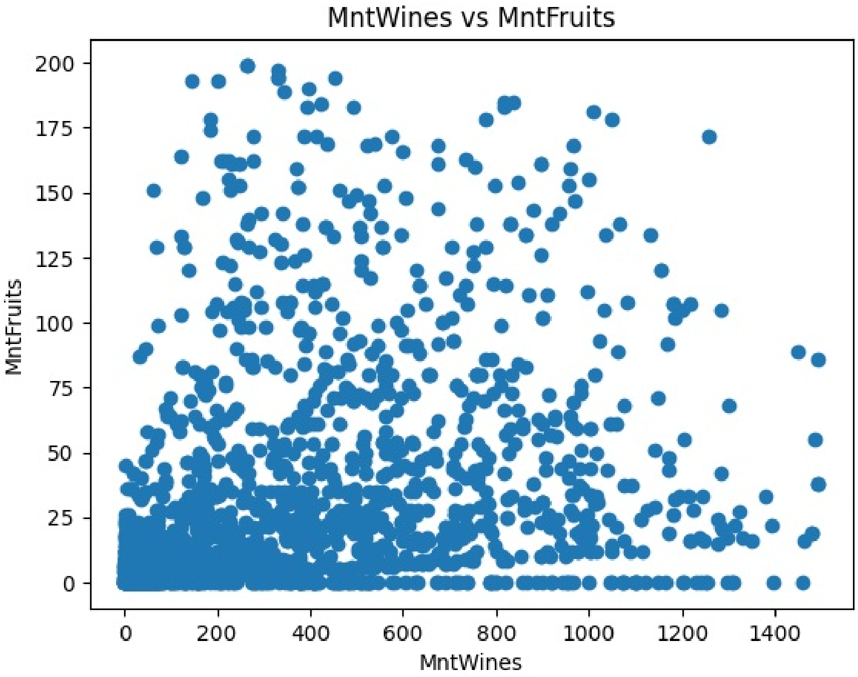


In [142…

```
plt.scatter(df['MntWines'], df['MntFruits'])
plt.xlabel('MntWines')
plt.ylabel('MntFruits')
plt.title('MntWines vs MntFruits')
```

Out[142…

```
Text(0.5, 1.0, 'MntWines vs MntFruits')
```

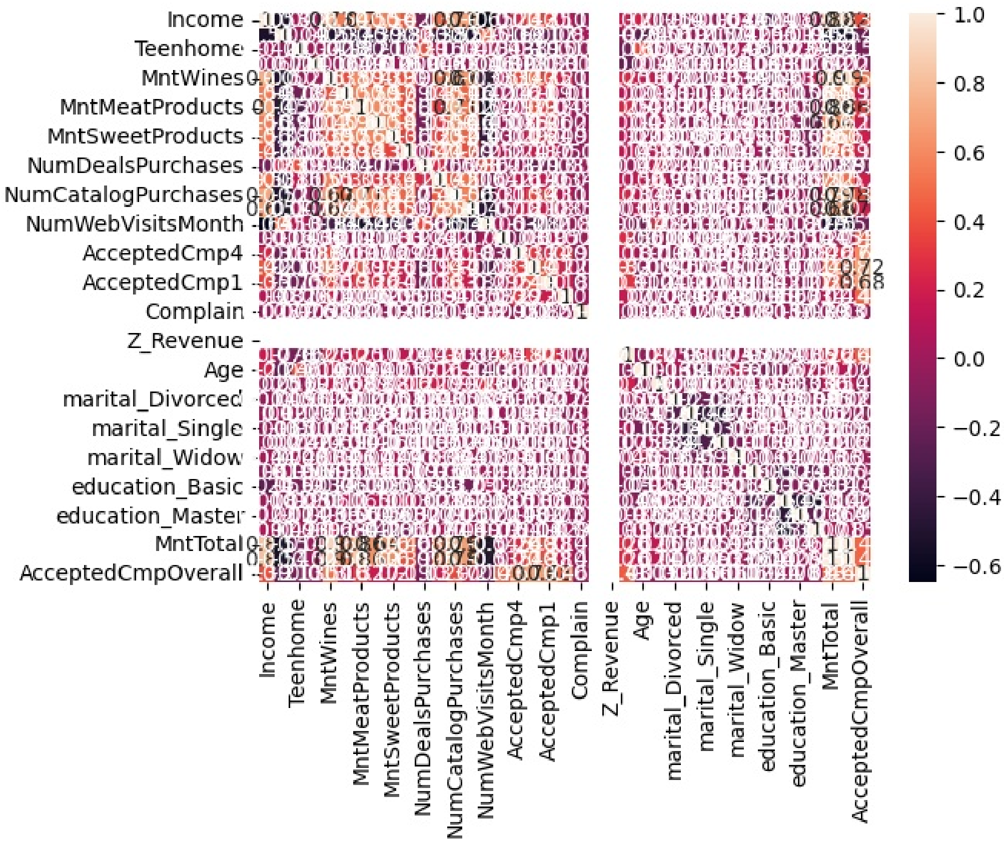


In [170…

```
sns.heatmap(df.corr(), annot=True)
```

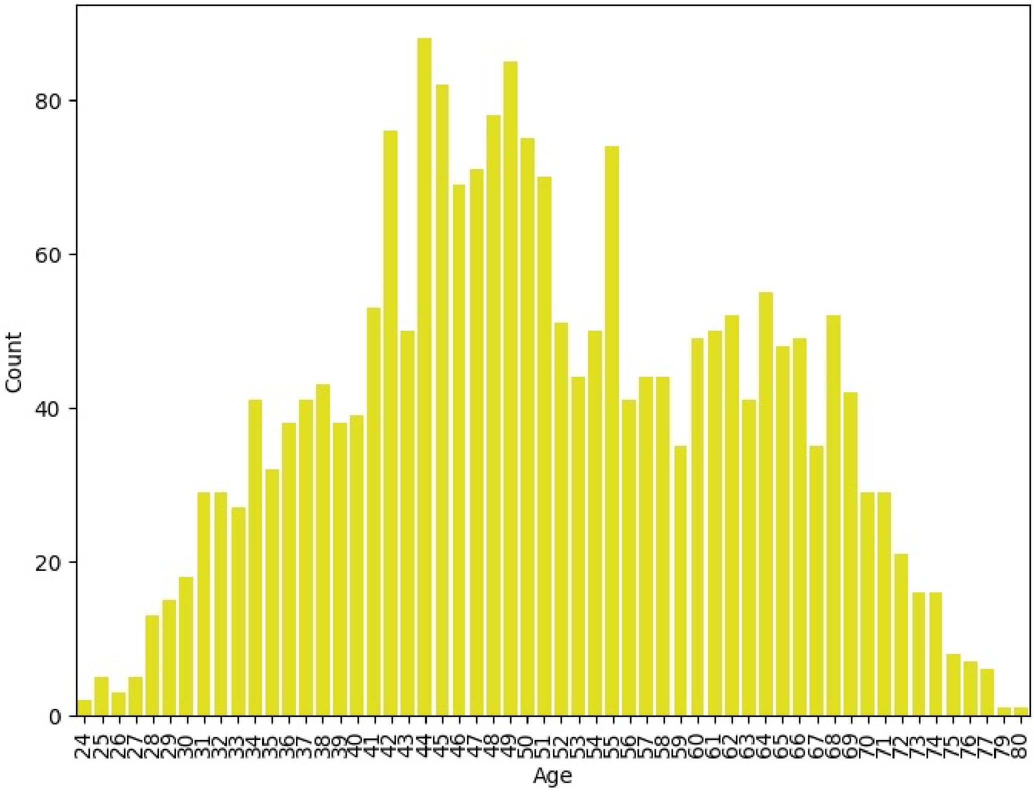
Out[170…

```
<Axes: >
```



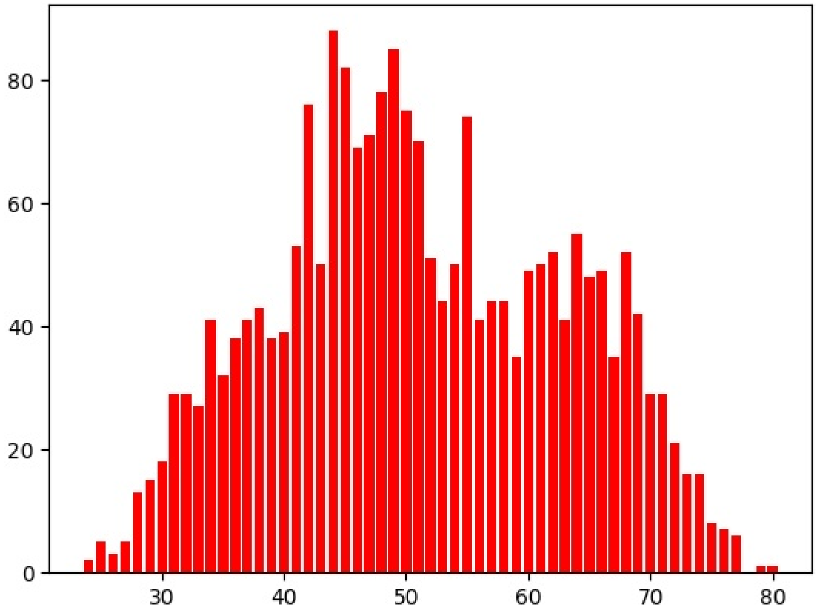
In [171…

```
age_graph = plt.figure(figsize=(8,6))
sns.barplot(x=age_df.index,y=age_df.Age, color = 'yellow')
plt.ylabel('Count')
plt.xticks(rotation = 90);
```



In [172…

```
plt.legend
plt.bar(age_df.index,age_df.Age, color='red');
```



```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```