```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df=pd.read_csv("marketing_campaign.csv")

print(df.head())
```

```
     ID  Year_Birth   Education Marital_Status   Income  Kidhome
Teenhome  \
0  5524        1957  Graduation         Single  58138.0        0
0
1  2174        1954  Graduation         Single  46344.0        1
1
2  4141        1965  Graduation       Together  71613.0        0
0
3  6182        1984  Graduation       Together  26646.0        1
0
4  5324        1981         PhD        Married  58293.0        1
0

  Dt_Customer  Recency  MntWines  ...  NumWebVisitsMonth  AcceptedCmp3
\
0  04/09/2012       58       635  ...                  7             0

1  08/03/2014       38        11  ...                  5             0

2  21/08/2013       26       426  ...                  4             0

3  10/02/2014       26        11  ...                  6             0

4  19/01/2014       94       173  ...                  5             0


   AcceptedCmp4  AcceptedCmp5  AcceptedCmp1  AcceptedCmp2  Complain  \
0             0             0             0             0         0
1             0             0             0             0         0
2             0             0             0             0         0
3             0             0             0             0         0
4             0             0             0             0         0

   Z_CostContact  Z_Revenue  Response
0              3         11         1
1              3         11         0
2              3         11         0
3              3         11         0
4              3         11         0

[5 rows x 29 columns]
```

```
print(df.tail())
```

```
         ID  Year_Birth   Education Marital_Status   Income
Kidhome  \
2235  10870        1967  Graduation        Married  61223.0         0

2236   4001        1946         PhD       Together  64014.0         2

2237   7270        1981  Graduation       Divorced  56981.0         0

2238   8235        1956      Master       Together  69245.0         0

2239   9405        1954         PhD        Married  52869.0         1


      Teenhome Dt_Customer  Recency  MntWines  ...  NumWebVisitsMonth
\
2235         1  13/06/2013       46       709  ...                  5

2236         1  10/06/2014       56       406  ...                  7

2237         0  25/01/2014       91       908  ...                  6

2238         1  24/01/2014        8       428  ...                  3

2239         1  15/10/2012       40        84  ...                  7


      AcceptedCmp3  AcceptedCmp4  AcceptedCmp5  AcceptedCmp1
AcceptedCmp2  \
2235             0             0             0             0
0
2236             0             0             0             1
0
2237             0             1             0             0
0
2238             0             0             0             0
0
2239             0             0             0             0
0

      Complain  Z_CostContact  Z_Revenue  Response
2235         0              3         11         0
2236         0              3         11         0
2237         0              3         11         0
2238         0              3         11         0
2239         0              3         11         1

[5 rows x 29 columns]
```

```
print(df.describe())
```

```
                 ID   Year_Birth         Income      Kidhome
Teenhome  \
count   2240.000000  2240.000000    2216.000000  2240.000000
2240.000000
mean    5592.159821  1968.805804   52247.251354     0.444196
0.506250
std     3246.662198    11.984069   25173.076661     0.538398
0.544538
min        0.000000  1893.000000    1730.000000     0.000000
0.000000
25%     2828.250000  1959.000000   35303.000000     0.000000
0.000000
50%     5458.500000  1970.000000   51381.500000     0.000000
0.000000
75%     8427.750000  1977.000000   68522.000000     1.000000
1.000000
max    11191.000000  1996.000000  666666.000000     2.000000
2.000000

           Recency     MntWines    MntFruits  MntMeatProducts  \
count  2240.000000  2240.000000  2240.000000      2240.000000
mean     49.109375   303.935714    26.302232       166.950000
std      28.962453   336.597393    39.773434       225.715373
min       0.000000     0.000000     0.000000         0.000000
25%      24.000000    23.750000     1.000000        16.000000
50%      49.000000   173.500000     8.000000        67.000000
75%      74.000000   504.250000    33.000000       232.000000
max      99.000000  1493.000000   199.000000      1725.000000

       MntFishProducts  ...  NumWebVisitsMonth  AcceptedCmp3
AcceptedCmp4  \
count      2240.000000  ...        2240.000000   2240.000000
2240.000000
mean         37.525446  ...           5.316518      0.072768
0.074554
std          54.628979  ...           2.426645      0.259813
0.262728
min           0.000000  ...           0.000000      0.000000
0.000000
25%           3.000000  ...           3.000000      0.000000
0.000000
50%          12.000000  ...           6.000000      0.000000
0.000000
75%          50.000000  ...           7.000000      0.000000
0.000000
max         259.000000  ...          20.000000      1.000000
1.000000

       AcceptedCmp5  AcceptedCmp1  AcceptedCmp2     Complain
Z_CostContact  \
```

```
count   2240.000000   2240.000000   2240.000000  2240.000000
2240.0
mean       0.072768      0.064286      0.013393     0.009375
3.0
std        0.259813      0.245316      0.114976     0.096391
0.0
min        0.000000      0.000000      0.000000     0.000000
3.0
25%        0.000000      0.000000      0.000000     0.000000
3.0
50%        0.000000      0.000000      0.000000     0.000000
3.0
75%        0.000000      0.000000      0.000000     0.000000
3.0
max        1.000000      1.000000      1.000000     1.000000
3.0

       Z_Revenue     Response
count     2240.0  2240.000000
mean        11.0     0.149107
std          0.0     0.356274
min         11.0     0.000000
25%         11.0     0.000000
50%         11.0     0.000000
75%         11.0     0.000000
max         11.0     1.000000

[8 rows x 26 columns]
```

```
print(df.isna().sum())
```

```
ID                     0
Year_Birth             0
Education              0
Marital_Status         0
Income                24
Kidhome                0
Teenhome               0
Dt_Customer            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
dtype: int64
```

```
print(df.isnull().sum())
```

```
ID                     0
Year_Birth             0
Education              0
Marital_Status         0
Income                24
Kidhome                0
Teenhome               0
Dt_Customer            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
dtype: int64
```

```
print(type(df))
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
df.dropna(inplace=True)
```

```
print(df.isna().sum())
```

```
ID                     0
Year_Birth             0
Education              0
Marital_Status         0
Income                 0
Kidhome                0
Teenhome               0
Dt_Customer            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
dtype: int64
```

```
df['total_child'] = df['Kidhome'] + df['Teenhome']
```

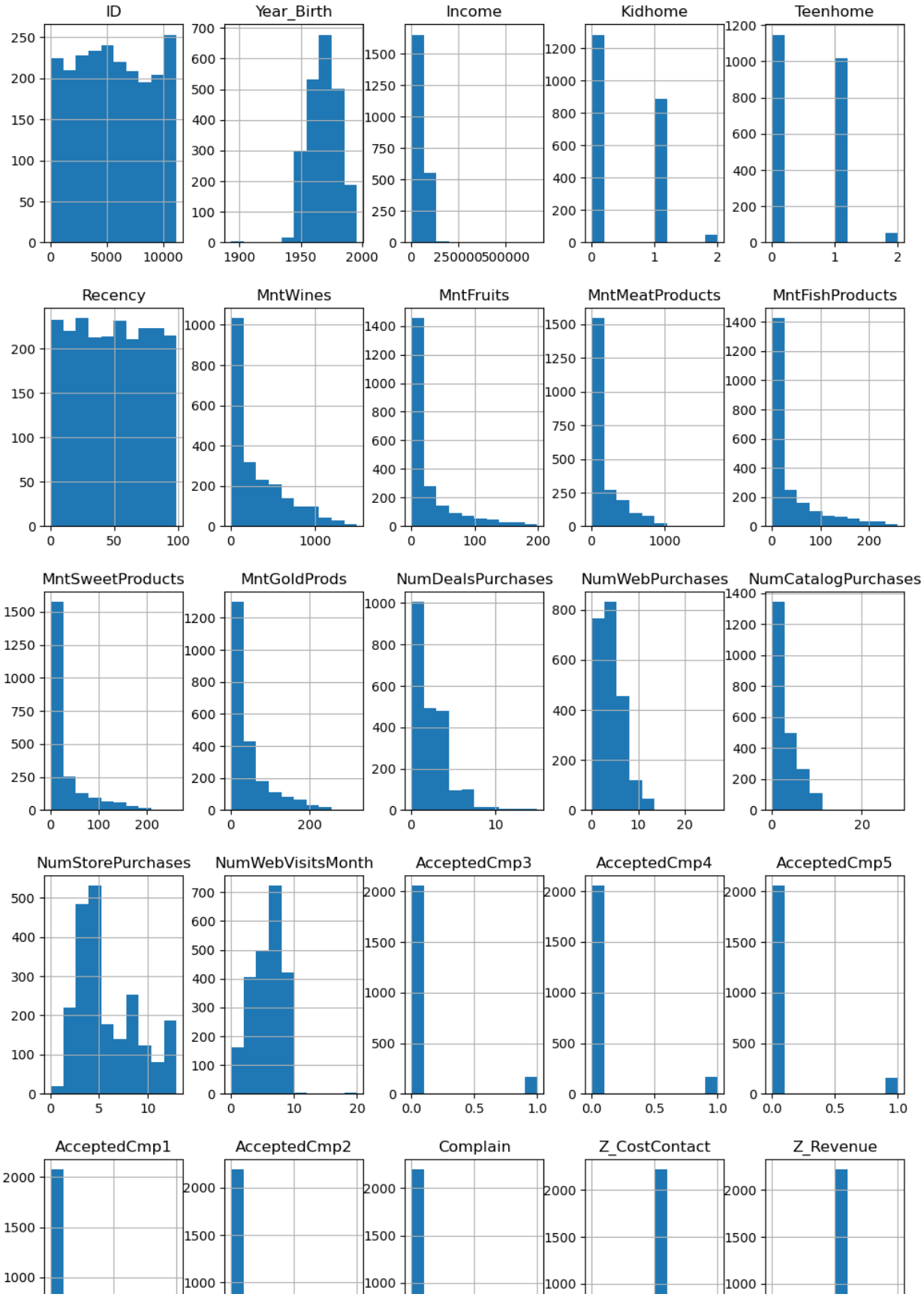
```
def to_married(x):
    if x == 'Together':
        return 'Married'
    elif x == 'Alone':
        return 'Single'
    elif x == 'Divorced':
        return 'Single'
    elif x == 'Widow':
        return 'Single'
    else:
        return x
df['Marital_Status'] =  df['Marital_Status'].apply(to_married)
```

```
df['Marital_Status'].value_counts()
```

```
Marital_Status
Married    1430
Single      782
Absurd        2
YOLO          2
Name: count, dtype: int64
```

```
df.hist(figsize=(12,23))
```

```
array([[<Axes: title={'center': 'ID'}>,
        <Axes: title={'center': 'Year_Birth'}>,
        <Axes: title={'center': 'Income'}>,
        <Axes: title={'center': 'Kidhome'}>,
        <Axes: title={'center': 'Teenhome'}>],
       [<Axes: title={'center': 'Recency'}>,
        <Axes: title={'center': 'MntWines'}>,
        <Axes: title={'center': 'MntFruits'}>,
        <Axes: title={'center': 'MntMeatProducts'}>,
        <Axes: title={'center': 'MntFishProducts'}>],
       [<Axes: title={'center': 'MntSweetProducts'}>,
        <Axes: title={'center': 'MntGoldProds'}>,
        <Axes: title={'center': 'NumDealsPurchases'}>,
        <Axes: title={'center': 'NumWebPurchases'}>,
        <Axes: title={'center': 'NumCatalogPurchases'}>],
       [<Axes: title={'center': 'NumStorePurchases'}>,
        <Axes: title={'center': 'NumWebVisitsMonth'}>,
        <Axes: title={'center': 'AcceptedCmp3'}>,
        <Axes: title={'center': 'AcceptedCmp4'}>,
        <Axes: title={'center': 'AcceptedCmp5'}>],
       [<Axes: title={'center': 'AcceptedCmp1'}>,
        <Axes: title={'center': 'AcceptedCmp2'}>,
        <Axes: title={'center': 'Complain'}>,
        <Axes: title={'center': 'Z_CostContact'}>,
        <Axes: title={'center': 'Z_Revenue'}>],
       [<Axes: title={'center': 'Response'}>,
        <Axes: title={'center': 'total_child'}>, <Axes: >, <Axes: >,
        <Axes: >]], dtype=object)
```

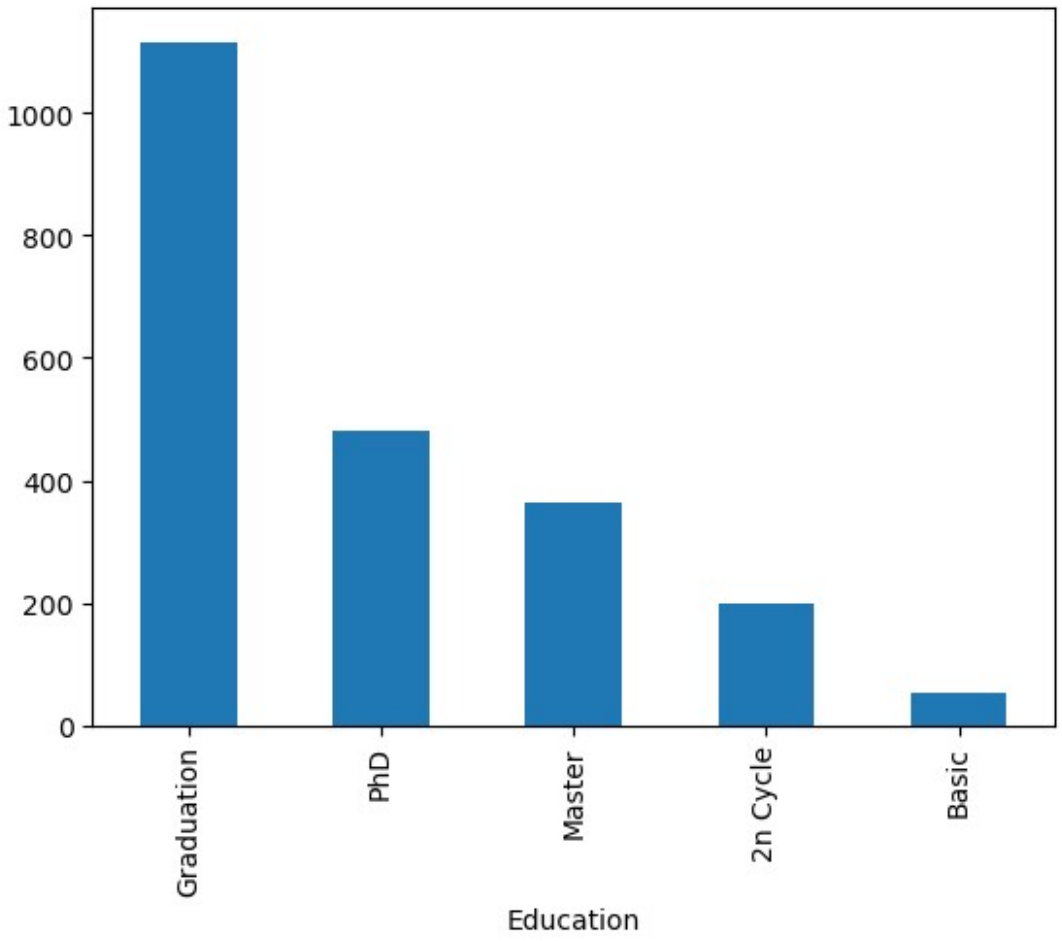

ID
Year_Birth
Income
Kidhome
Teenhome
Recency
MntWines
MntFruits
MntMeatProducts
MntFishProducts
MntSweetProducts
MntGoldProds
NumDealsPurchases
NumWebPurchases
NumCatalogPurchases
NumStorePurchases
NumWebVisitsMonth
AcceptedCmp3
AcceptedCmp4
AcceptedCmp5
AcceptedCmp1
AcceptedCmp2
Complain
Z_CostContact
Z_Revenue

```
df.Education.value_counts()
```

```
Education
Graduation    1116
PhD            481
Master         365
2n Cycle       200
Basic           54
Name: count, dtype: int64
```

```
df.Education.value_counts().plot(kind = "bar")
```

```
<Axes: xlabel='Education'>
```

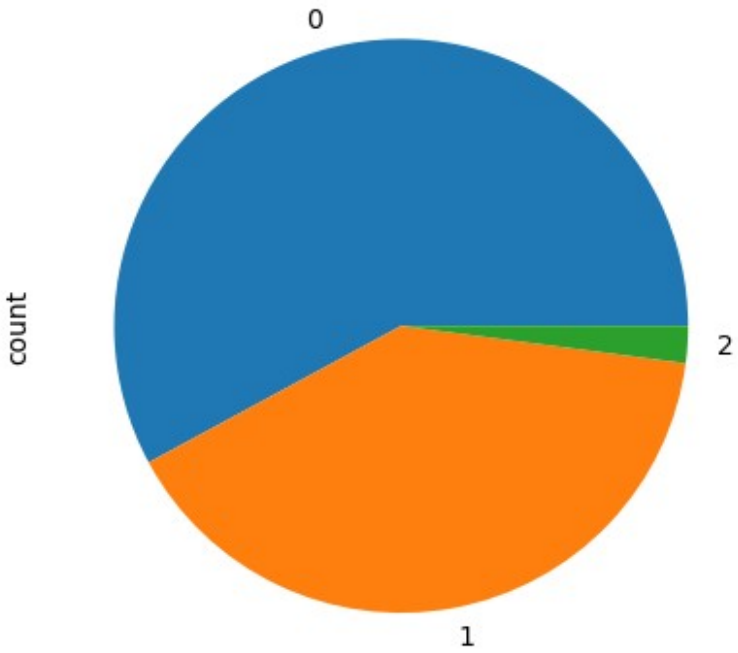



```
df.describe()[["Income"]].agg(["mean", "std","median"])
```

```
                Income
mean    112904.853502
std     225013.172775
median   43342.250000
```

```
df.Kidhome.value_counts().plot(kind= "pie")
```

```
<Axes: ylabel='count'>
```




```
df["AcceptedCmp3"].value_counts()
```

```
AcceptedCmp3
0    2053
1     163
Name: count, dtype: int64
```

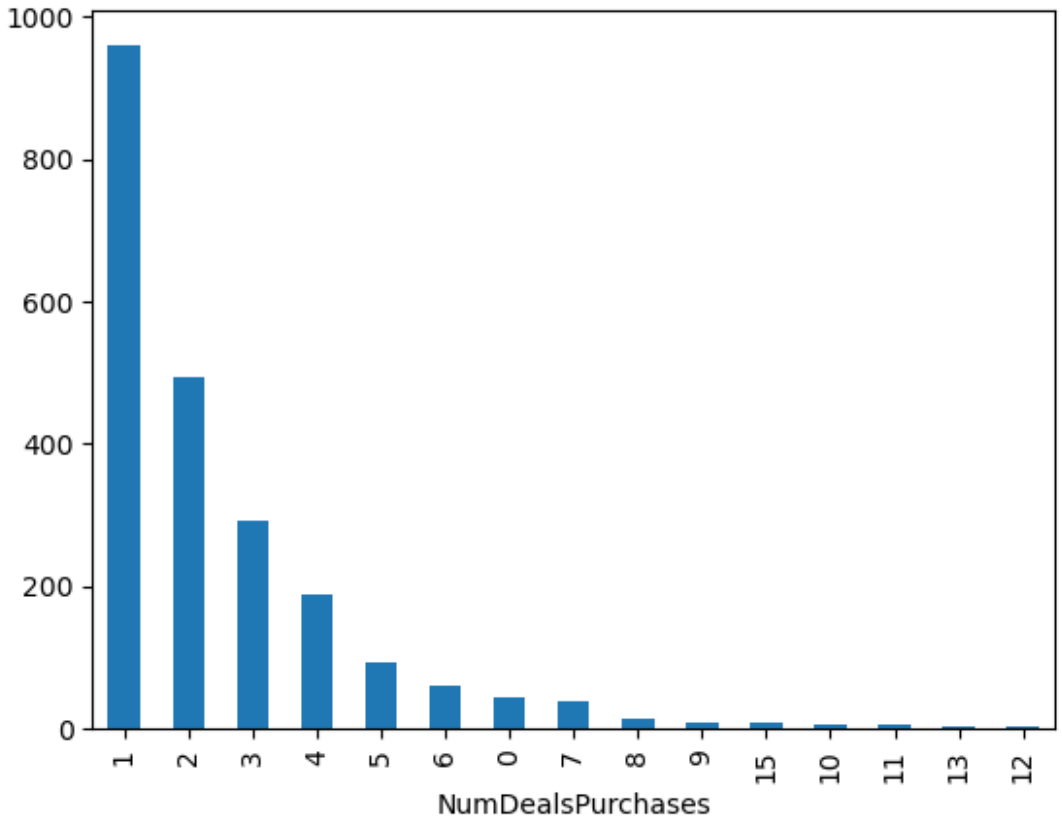
```
df['NumDealsPurchases'].value_counts()
```

```
NumDealsPurchases
1     960
2     493
3     293
4     188
5      94
6      60
0      44
7      39
8      14
9       8
15      7
10      5
11      5
```

```
13      3
12      3
Name: count, dtype: int64
```

```
df['NumDealsPurchases'].value_counts().plot(kind = "bar")
```

```
<Axes: xlabel='NumDealsPurchases'>
```



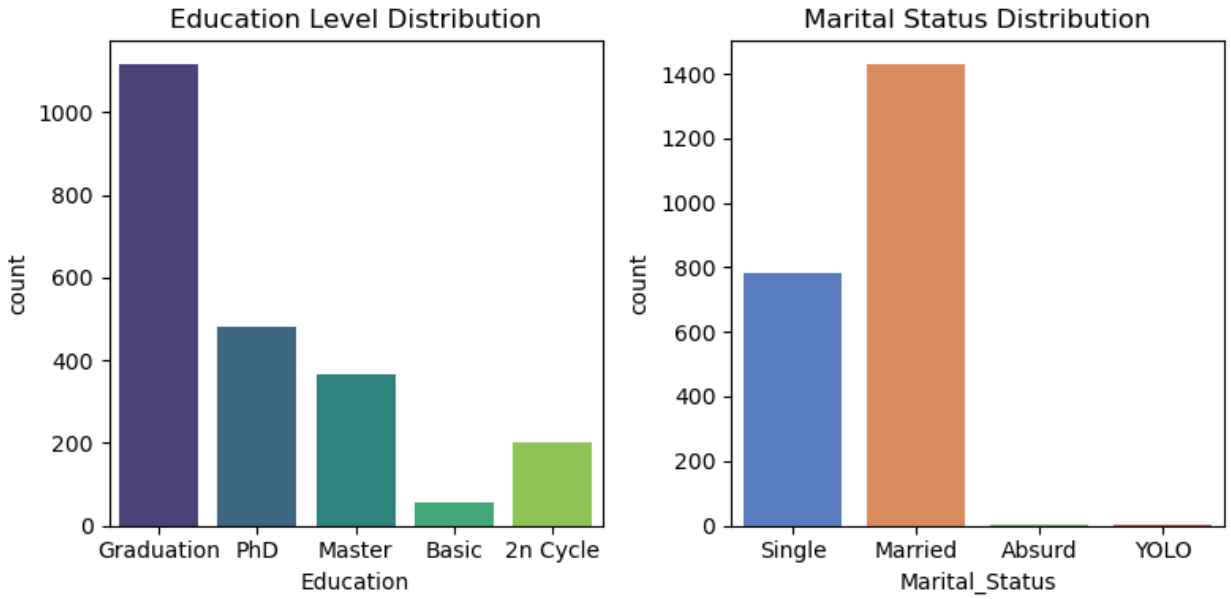

```
pd.crosstab(df.Education, df.Income)
```

Income	1730.0	2447.0	3502.0	4023.0	4428.0	4861.0
\						
Education						
2n Cycle	0	0	0	0	0	0
Basic	0	0	0	0	0	0
Graduation	1	1	1	0	1	1
Master	0	0	0	0	0	0
PhD	0	0	0	1	0	0

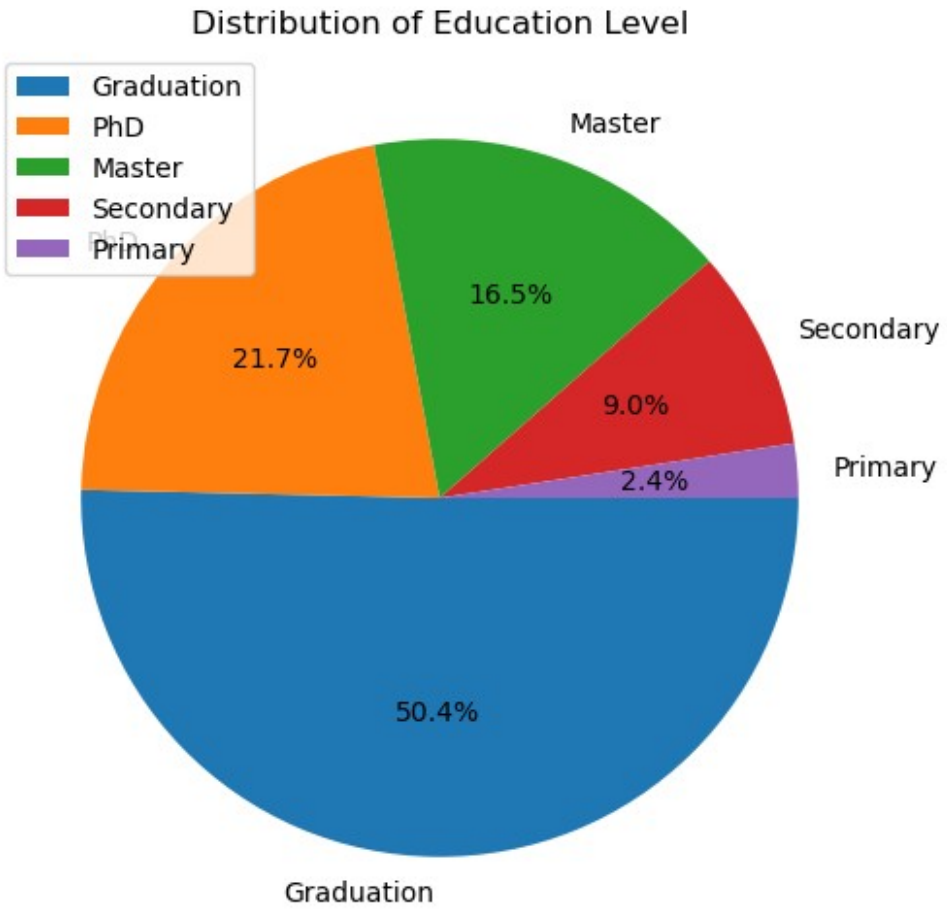
```
Income      5305.0    5648.0    6560.0    6835.0    ...  105471.0
113734.0  \
Education                                             ...

2n Cycle           0         0         0         0  ...         0
0
Basic              0         0         0         0  ...         0
0
Graduation         1         0         0         0  ...         1
0
Master             0         0         1         0  ...         0
0
PhD                0         1         0         1  ...         0
1

Income      153924.0  156924.0  157146.0  157243.0  157733.0  160803.0
\
Education

2n Cycle           0         0         0         0         0         0

Basic              0         0         0         0         0         0

Graduation         1         0         1         0         0         0

Master             0         0         0         0         1         0

PhD                0         1         0         1         0         1


Income      162397.0  666666.0
Education
2n Cycle           0         0
Basic              0         0
Graduation         0         1
Master             0         0
PhD                1         0

[5 rows x 1974 columns]
```

```
plt.figure(figsize=(8, 4))
plt.subplot(1, 2, 1)


sns.countplot(x='Education', data=df, palette='viridis')
plt.title('Education Level Distribution')
plt.subplot(1, 2, 2)

sns.countplot(x='Marital_Status', data=df, palette='muted')
plt.title('Marital Status Distribution')
```

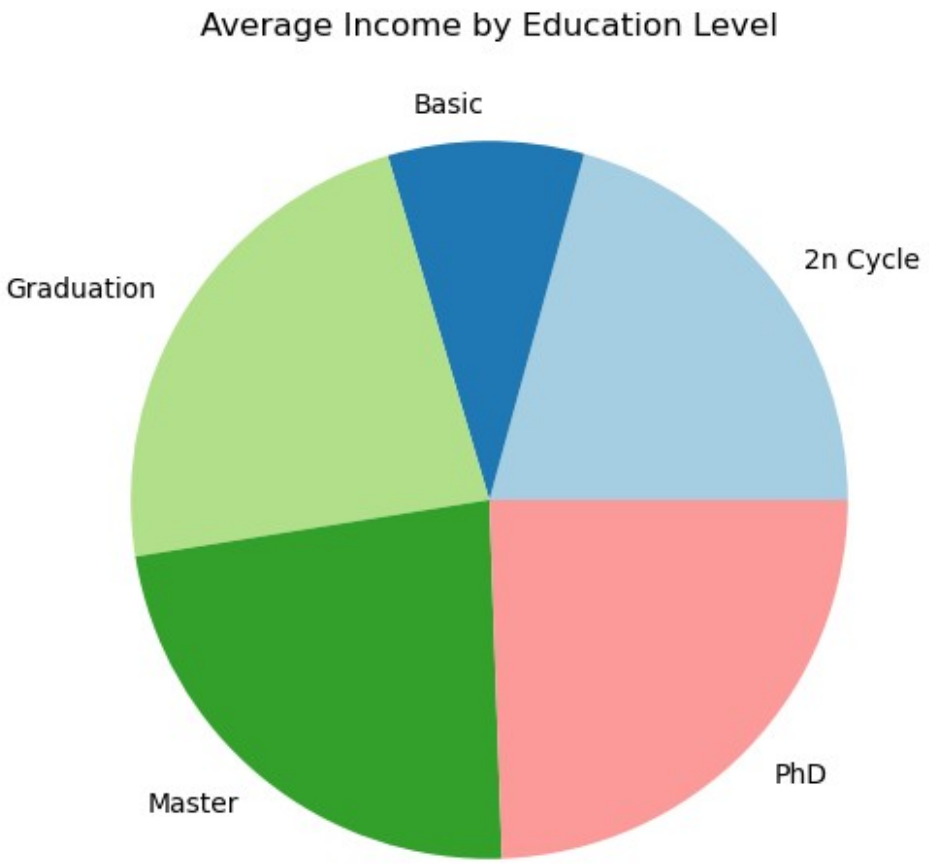
```
plt.tight_layout()
plt.show()
```



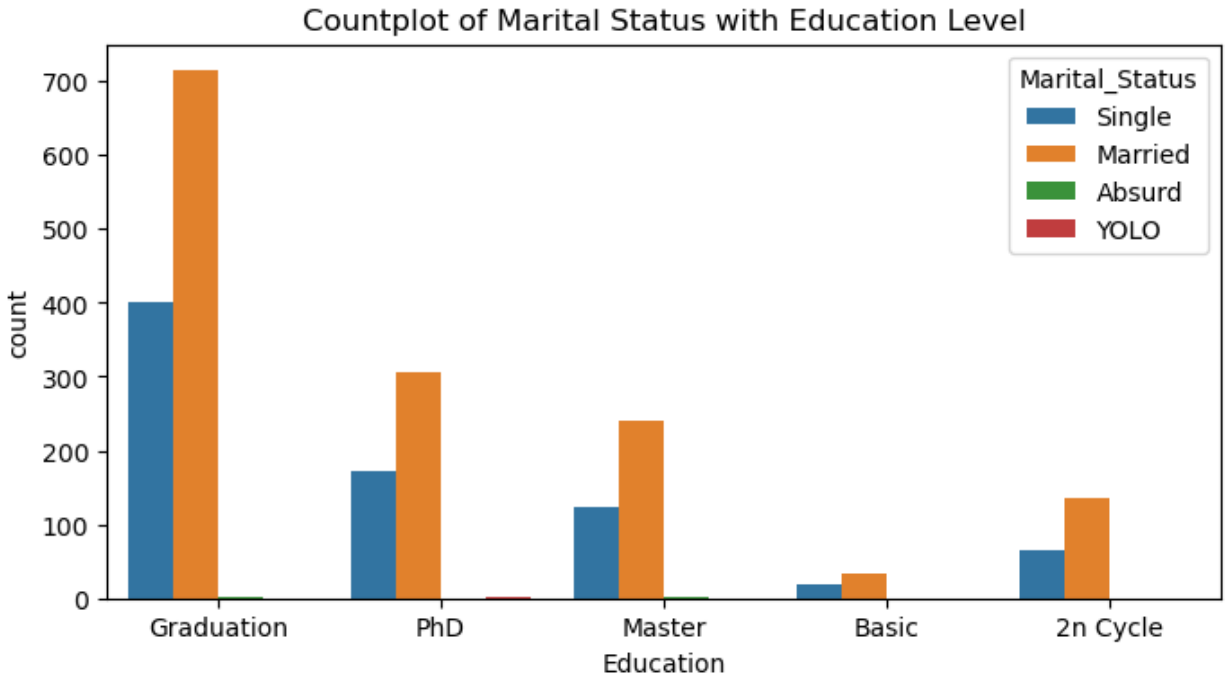

```
plt.figure(figsize=(8,6))
plt.pie(df["Education"].value_counts(), labels = ["Graduation", "PhD",
"Master", "Secondary", "Primary"], autopct='%1.1f%%',
counterclock=False)
plt.legend()
plt.title("Distribution of Education Level")
plt.show()
```



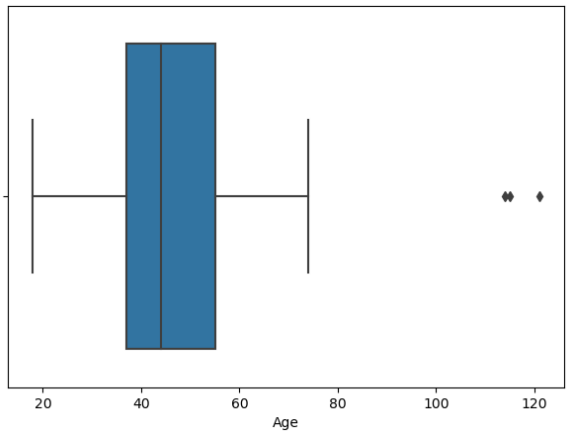

```
education_income = df.groupby('Education')
['Income'].mean().reset_index()


plt.figure(figsize=(10, 6))
plt.pie(education_income['Income'],
labels=education_income['Education'],colors=plt.cm.Paired.colors)
plt.title('Average Income by Education Level')
plt.show();
```

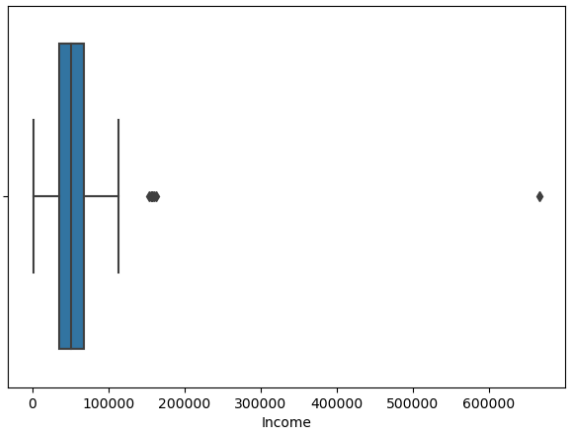



```
plt.figure(figsize=(8, 4))
sns.countplot(x='Education', hue='Marital_Status', data=df)
plt.title('Countplot of Marital Status with Education Level')
plt.show()
```




```
categorical_columns = []

for i in df.columns:
    if (df[i].dtypes == "object"):
        categorical_columns.append(i)
df['Age'] = 2014-df['Year_Birth']

fig, ax = plt.subplots(1, 2, figsize=(16,5))

sns.boxplot(x=df['Age'], ax=ax[0])
sns.boxplot(x=df['Income'], ax=ax[1])
plt.show()
```






```
df.describe()
```

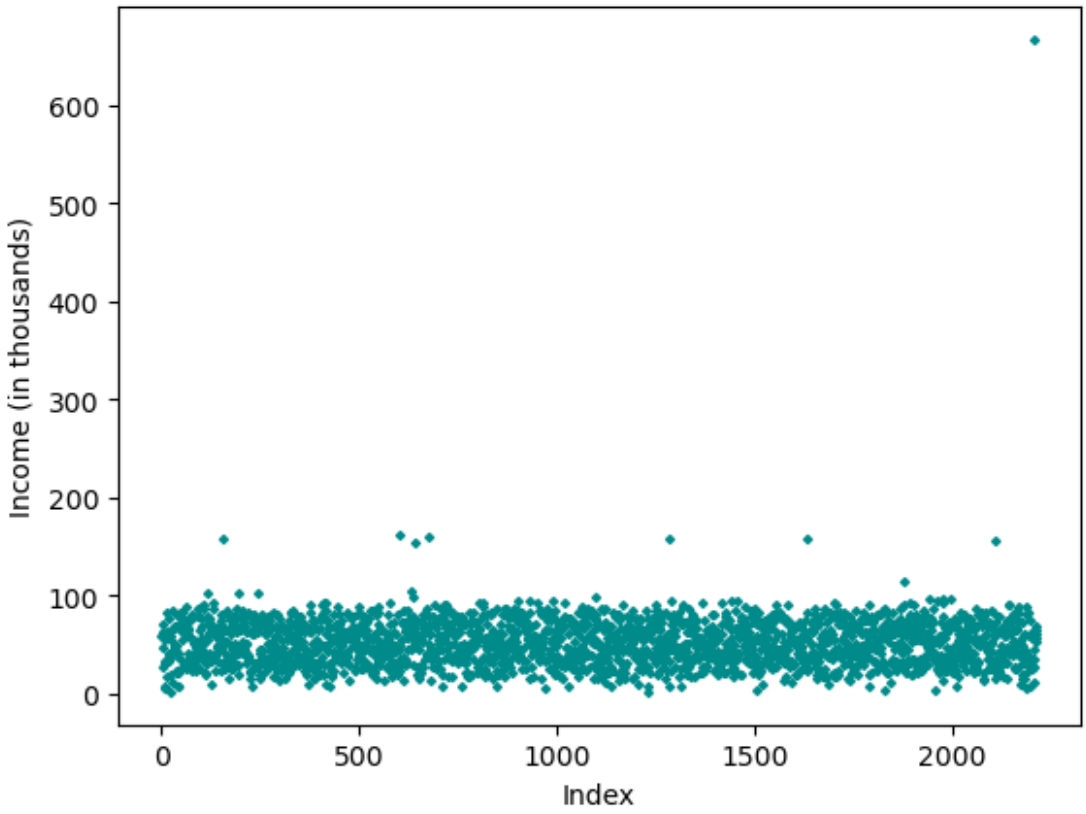
```
                ID   Year_Birth         Income      Kidhome
Teenhome  \
count   2216.000000  2216.000000    2216.000000  2216.000000
2216.000000
mean    5588.353339  1968.820397   52247.251354     0.441787
0.505415
std     3249.376275    11.985554   25173.076661     0.536896
0.544181
min        0.000000  1893.000000    1730.000000     0.000000
0.000000
25%     2814.750000  1959.000000   35303.000000     0.000000
0.000000
50%     5458.500000  1970.000000   51381.500000     0.000000
0.000000
75%     8421.750000  1977.000000   68522.000000     1.000000
1.000000
max    11191.000000  1996.000000  666666.000000     2.000000
2.000000

           Recency     MntWines    MntFruits  MntMeatProducts  \
count  2216.000000  2216.000000  2216.000000      2216.000000
mean     49.012635   305.091606    26.356047       166.995939
std      28.948352   337.327920    39.793917       224.283273
min       0.000000     0.000000     0.000000         0.000000
25%      24.000000    24.000000     2.000000        16.000000
50%      49.000000   174.500000     8.000000        68.000000
75%      74.000000   505.000000    33.000000       232.250000
max      99.000000  1493.000000   199.000000      1725.000000

       MntFishProducts  ...  AcceptedCmp4  AcceptedCmp5  AcceptedCmp1
\
count      2216.000000  ...   2216.000000   2216.000000   2216.000000

mean         37.637635  ...      0.074007      0.073105      0.064079

std          54.752082  ...      0.261842      0.260367      0.244950

min           0.000000  ...      0.000000      0.000000      0.000000

25%           3.000000  ...      0.000000      0.000000      0.000000

50%          12.000000  ...      0.000000      0.000000      0.000000

75%          50.000000  ...      0.000000      0.000000      0.000000

max         259.000000  ...      1.000000      1.000000      1.000000


       AcceptedCmp2     Complain  Z_CostContact  Z_Revenue
Response  \
count   2216.000000  2216.000000         2216.0     2216.0
```

```
2216.000000
mean       0.013538     0.009477             3.0       11.0
0.150271
std        0.115588     0.096907             0.0        0.0
0.357417
min        0.000000     0.000000             3.0       11.0
0.000000
25%        0.000000     0.000000             3.0       11.0
0.000000
50%        0.000000     0.000000             3.0       11.0
0.000000
75%        0.000000     0.000000             3.0       11.0
0.000000
max        1.000000     1.000000             3.0       11.0
1.000000

       total_child          Age
count  2216.000000  2216.000000
mean      0.947202    45.179603
std       0.749062    11.985554
min       0.000000    18.000000
25%       0.000000    37.000000
50%       1.000000    44.000000
75%       1.000000    55.000000
max       3.000000   121.000000

[8 rows x 28 columns]
```

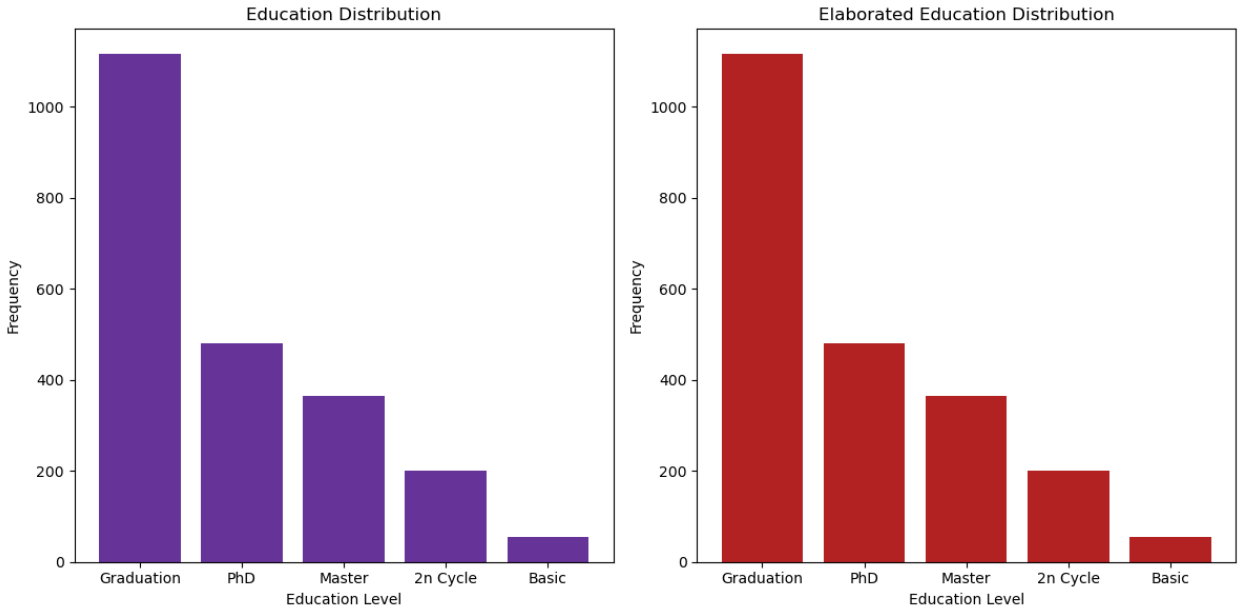
```
df.duplicated()
```

```
0       False
1       False
2       False
3       False
4       False
        ...
2235    False
2236    False
2237    False
2238    False
2239    False
Length: 2216, dtype: bool
```

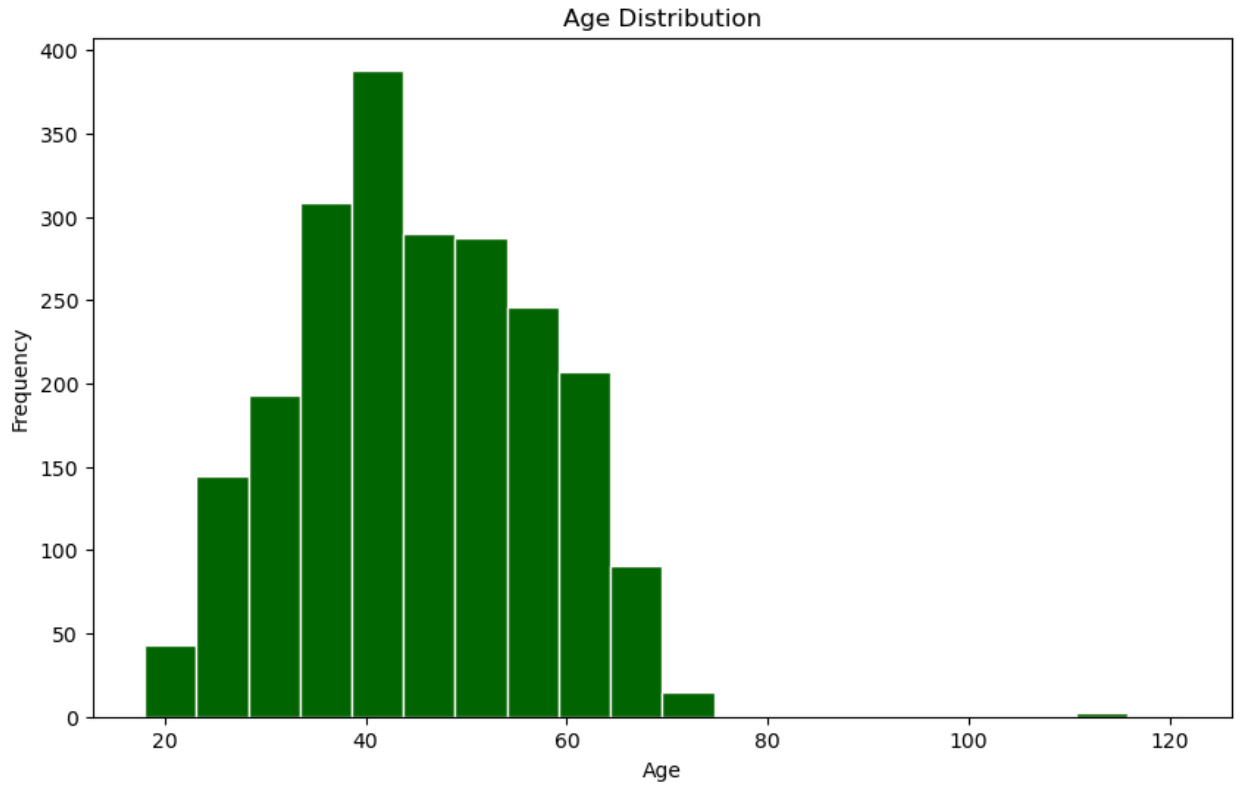
```
income_data = df['Income']
income_div = income_data / 1000
plt.scatter(range(len(income_div)), income_div, marker='D', s = 4, c =
'darkcyan')
plt.xlabel('Index')
plt.ylabel('Income (in thousands)')
plt.show()
```



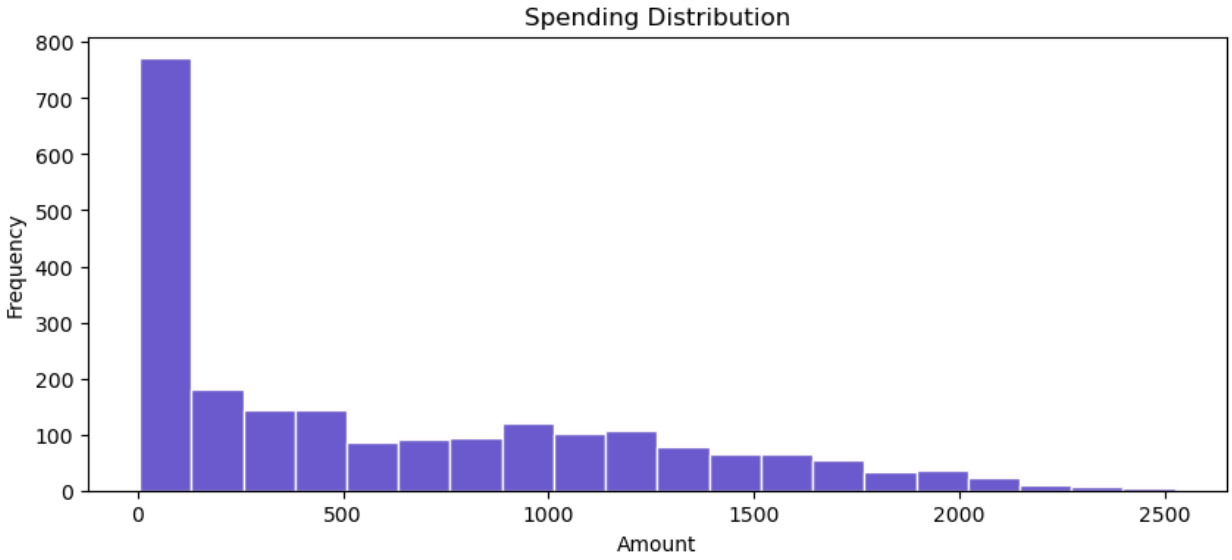

```
income_greater_than_200000 = df[df['Income'] >= 200000].index
df.drop(income_greater_than_200000, inplace=True)
education_dist = df['Education'].value_counts()
plt.figure(figsize=(12,6))
plt.subplot(1,2,1)
plt.bar(education_dist.index, education_dist.values,
color='rebeccapurple')
plt.title('Education Distribution')
plt.xlabel('Education Level')
plt.ylabel('Frequency')

elaborated_dist = df['Education'].value_counts()
plt.subplot(1,2,2)
plt.bar(elaborated_dist.index, elaborated_dist.values, color =
'firebrick')
plt.title('Elaborated Education Distribution')
plt.xlabel('Education Level')
plt.ylabel('Frequency')

plt.tight_layout()
plt.show()
```



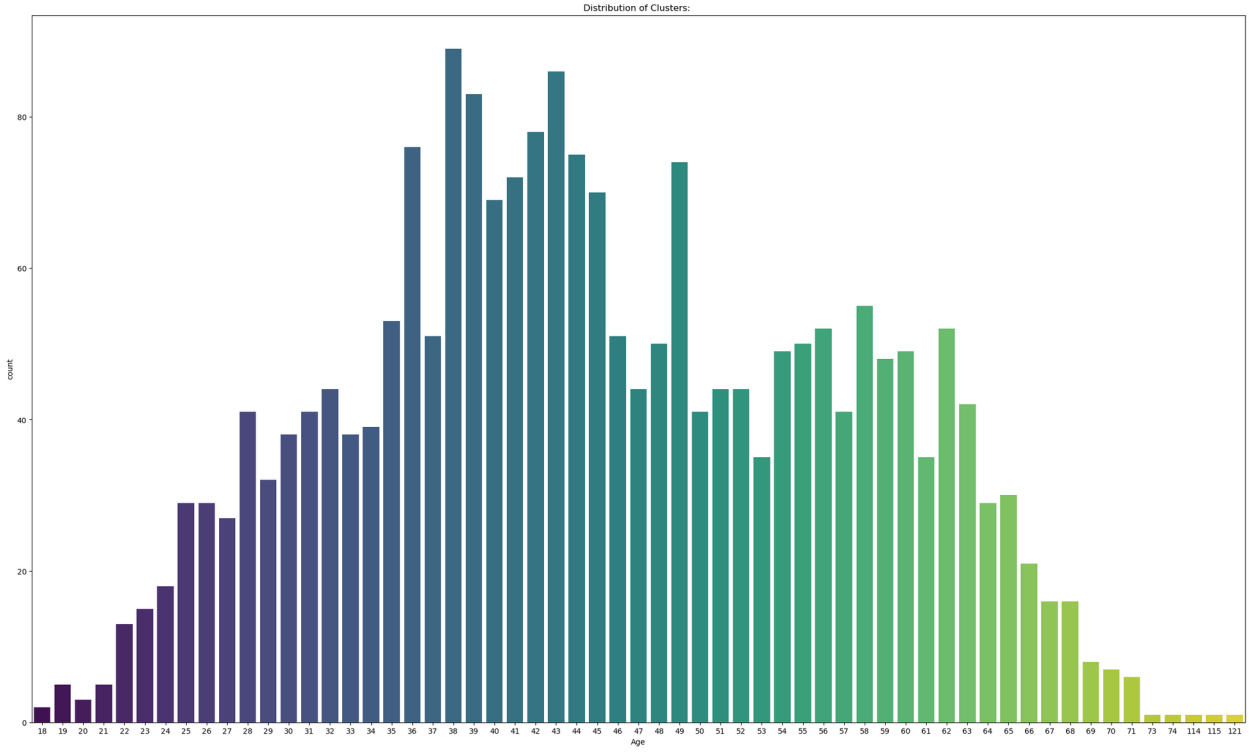

```
plt.figure(figsize=(10, 6))
plt.hist(df['Age'], bins=20, color='darkgreen', edgecolor='snow')
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```



```
df['TotalMntSpent'] = df[['MntWines', 'MntFruits',
       'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts',
       'MntGoldProds']].sum(axis=1)

plt.figure(figsize=(10, 4))
plt.hist(df['TotalMntSpent'], bins=20, color='slateblue',
edgecolor='snow')
plt.title('Spending Distribution')
plt.xlabel('Amount')
plt.ylabel('Frequency')
plt.show()
```




```
plt.figure(figsize=(29,17))
sns.countplot(x=df['Age'], palette='viridis')
plt.title("Distribution of Clusters: ")
plt.show()
```


Distribution of Clusters:
count
Age
0
20
40
60
80
18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 73 74 114 115 121