In [61]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import preprocessing
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
```

In [62]:

```
data=pd.read_csv("D:/DataSets/marketing_campaign.csv",sep="\t")
```

In [63]:

```
data
```

Out[63]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines
0	5524	1957	Graduation	Single	58138.0	0	0	04-09-2012	58	635
1	2174	1954	Graduation	Single	46344.0	1	1	08-03-2014	38	11
2	4141	1965	Graduation	Together	71613.0	0	0	21-08-2013	26	426
3	6182	1984	Graduation	Together	26646.0	1	0	10-02-2014	26	11
4	5324	1981	PhD	Married	58293.0	1	0	19-01-2014	94	173
...	...	...	...	...	...	...	...	...	...	...
2235	10870	1967	Graduation	Married	61223.0	0	1	13-06-2013	46	709
2236	4001	1946	PhD	Together	64014.0	2	1	10-06-2014	56	406
2237	7270	1981	Graduation	Divorced	56981.0	0	0	25-01-2014	91	908
2238	8235	1956	Master	Together	69245.0	0	1	24-01-2014	8	428
2239	9405	1954	PhD	Married	52869.0	1	1	15-10-2012	40	84

2240 rows × 29 columns

In [64]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2240 entries, 0 to 2239
Data columns (total 29 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   ID                   2240 non-null   int64  
 1   Year_Birth           2240 non-null   int64  
 2   Education            2240 non-null   object 
 3   Marital_Status       2240 non-null   object 
 4   Income               2216 non-null   float64
 5   Kidhome              2240 non-null   int64  
 6   Teenhome             2240 non-null   int64  
 7   Dt_Customer          2240 non-null   object 
 8   Recency              2240 non-null   int64  
 9   MntWines             2240 non-null   int64  
 10  MntFruits            2240 non-null   int64  
 11  MntMeatProducts      2240 non-null   int64  
 12  MntFishProducts      2240 non-null   int64  
 13  MntSweetProducts     2240 non-null   int64  
 14  MntGoldProds         2240 non-null   int64  
 15  NumDealsPurchases    2240 non-null   int64  
 16  NumWebPurchases      2240 non-null   int64  
 17  NumCatalogPurchases  2240 non-null   int64  
 18  NumStorePurchases    2240 non-null   int64  
 19  NumWebVisitsMonth    2240 non-null   int64  
 20  AcceptedCmp3         2240 non-null   int64  
 21  AcceptedCmp4         2240 non-null   int64  
 22  AcceptedCmp5         2240 non-null   int64  
 23  AcceptedCmp1         2240 non-null   int64  
 24  AcceptedCmp2         2240 non-null   int64  
 25  Complain             2240 non-null   int64  
 26  Z_CostContact        2240 non-null   int64  
 27  Z_Revenue            2240 non-null   int64  
 28  Response             2240 non-null   int64  
dtypes: float64(1), int64(25), object(3)
memory usage: 507.6+ KB
```

In [65]:
```
data.columns
```

Out[65]:
```
Index(['ID', 'Year_Birth', 'Education', 'Marital_Status', 'Income', 'Kidhome',
       'Teenhome', 'Dt_Customer', 'Recency', 'MntWines', 'MntFruits',
       'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts',
       'MntGoldProds', 'NumDealsPurchases', 'NumWebPurchases',
       'NumCatalogPurchases', 'NumStorePurchases', 'NumWebVisitsMonth',
       'AcceptedCmp3', 'AcceptedCmp4', 'AcceptedCmp5', 'AcceptedCmp1',
       'AcceptedCmp2', 'Complain', 'Z_CostContact', 'Z_Revenue', 'Response'],
      dtype='object')
```

In [66]:
```
data.isnull().sum()
```

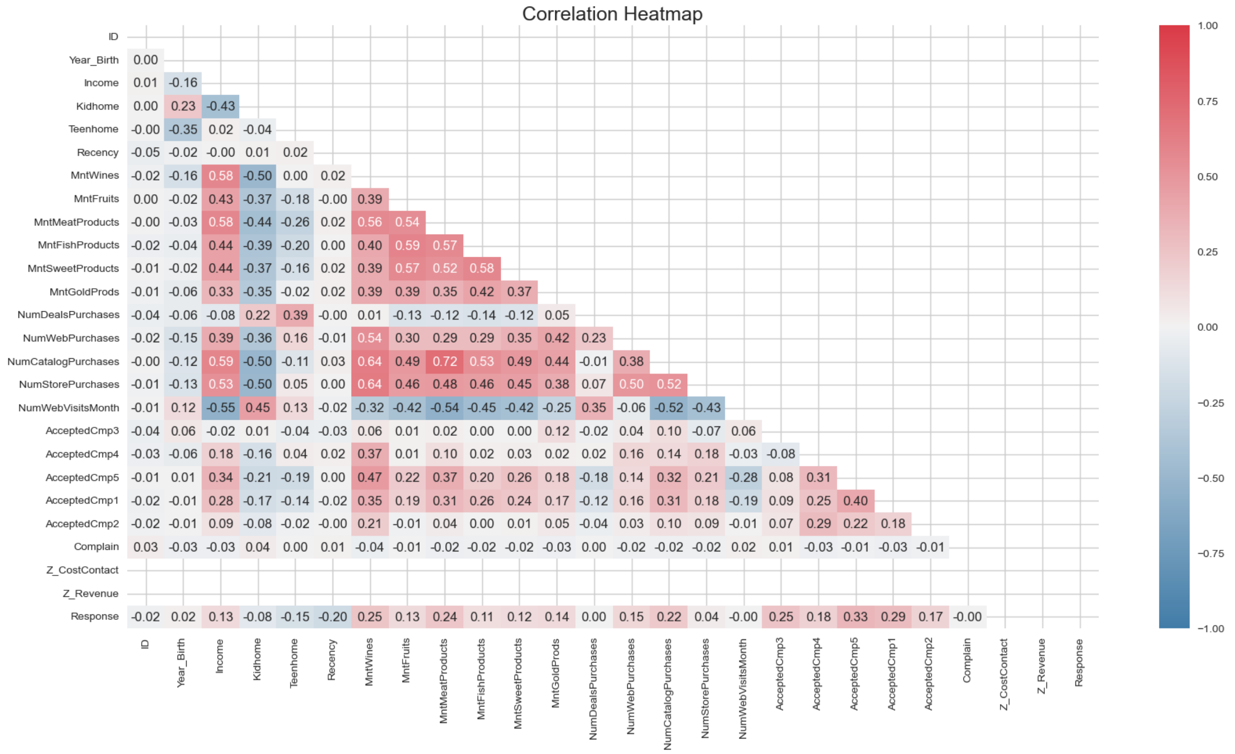
Out[66]:
```
ID                      0
Year_Birth              0
Education               0
Marital_Status          0
Income                 24
Kidhome                 0
Teenhome                0
Dt_Customer             0
Recency                 0
MntWines                0
MntFruits               0
MntMeatProducts         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
```

In [67]:
```
data.describe().T
```

Out[67]:

	count	mean	std	min	25%	50%	75%	max
ID	2240.0	5592.159821	3246.662198	0.0	2828.25	5458.5	8427.75	11191.0
Year_Birth	2240.0	1968.805804	11.984069	1893.0	1959.00	1970.0	1977.00	1996.0
Income	2216.0	52247.251354	25173.076661	1730.0	35303.00	51381.5	68522.00	666666.0
Kidhome	2240.0	0.444196	0.538398	0.0	0.00	0.0	1.00	2.0
Teenhome	2240.0	0.506250	0.544538	0.0	0.00	0.0	1.00	2.0
Recency	2240.0	49.109375	28.962453	0.0	24.00	49.0	74.00	99.0
MntWines	2240.0	303.935714	336.597393	0.0	23.75	173.5	504.25	1493.0
MntFruits	2240.0	26.302232	39.773434	0.0	1.00	8.0	33.00	199.0
MntMeatProducts	2240.0	166.950000	225.715373	0.0	16.00	67.0	232.00	1725.0
MntFishProducts	2240.0	37.525446	54.628979	0.0	3.00	12.0	50.00	259.0
MntSweetProducts	2240.0	27.062946	41.280498	0.0	1.00	8.0	33.00	263.0
MntGoldProds	2240.0	44.021875	52.167439	0.0	9.00	24.0	56.00	362.0
NumDealsPurchases	2240.0	2.325000	1.932238	0.0	1.00	2.0	3.00	15.0
NumWebPurchases	2240.0	4.084821	2.778714	0.0	2.00	4.0	6.00	27.0
NumCatalogPurchases	2240.0	2.662054	2.923101	0.0	0.00	2.0	4.00	28.0
NumStorePurchases	2240.0	5.790179	3.250958	0.0	3.00	5.0	8.00	13.0
NumWebVisitsMonth	2240.0	5.316518	2.426645	0.0	3.00	6.0	7.00	20.0
AcceptedCmp3	2240.0	0.072768	0.259813	0.0	0.00	0.0	0.00	1.0
AcceptedCmp4	2240.0	0.074554	0.262728	0.0	0.00	0.0	0.00	1.0
AcceptedCmp5	2240.0	0.072768	0.259813	0.0	0.00	0.0	0.00	1.0
AcceptedCmp1	2240.0	0.064286	0.245316	0.0	0.00	0.0	0.00	1.0
AcceptedCmp2	2240.0	0.013393	0.114976	0.0	0.00	0.0	0.00	1.0
Complain	2240.0	0.009375	0.096391	0.0	0.00	0.0	0.00	1.0
Z_CostContact	2240.0	3.000000	0.000000	3.0	3.00	3.0	3.00	3.0
Z_Revenue	2240.0	11.000000	0.000000	11.0	11.00	11.0	11.00	11.0
Response	2240.0	0.149107	0.356274	0.0	0.00	0.0	0.00	1.0

```
In [68]: #finding coorelation between columns..
         plt.figure(figsize=(20, 10))
         cmap = sns.diverging_palette(240, 10, as_cmap=True)
         mask = np.triu(np.ones_like(data.corr()))
         corr = sns.heatmap(data.corr(), fmt='.2f',vmin=-1, vmax=1, annot=True,cmap=cmap,mask=mask)
         corr.set_title('Correlation Heatmap', fontdict={'fontsize':18}, pad=5);
         cmap = sns.diverging_palette(230, 20, as_cmap=True)
```



In [69]:
```
data.isnull().sum()/data.shape[0] * 100
# 1% from income is null value
```

Out[69]:
```
ID                     0.000000
Year_Birth             0.000000
Education              0.000000
Marital_Status         0.000000
Income                 1.071429
Kidhome                0.000000
Teenhome               0.000000
Dt_Customer            0.000000
Recency                0.000000
MntWines               0.000000
MntFruits              0.000000
MntMeatProducts        0.000000
MntFishProducts        0.000000
MntSweetProducts       0.000000
MntGoldProds           0.000000
NumDealsPurchases      0.000000
NumWebPurchases        0.000000
NumCatalogPurchases    0.000000
NumStorePurchases      0.000000
NumWebVisitsMonth      0.000000
AcceptedCmp3           0.000000
AcceptedCmp4           0.000000
AcceptedCmp5           0.000000
AcceptedCmp1           0.000000
AcceptedCmp2           0.000000
Complain               0.000000
Z_CostContact          0.000000
Z_Revenue              0.000000
Response               0.000000
dtype: float64
```

In [70]:

```
#filling the null values with mean..
data['Income']=data['Income'].fillna(data['Income'].mean())
data.isnull().sum()
```

Out[70]:

```
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
```

In [71]:

```
data.duplicated().sum()
```

Out[71]:

```
0
```

In [72]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2240 entries, 0 to 2239
Data columns (total 29 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   ID                   2240 non-null   int64  
 1   Year_Birth           2240 non-null   int64  
 2   Education            2240 non-null   object 
 3   Marital_Status       2240 non-null   object 
 4   Income               2240 non-null   float64
 5   Kidhome              2240 non-null   int64  
 6   Teenhome             2240 non-null   int64  
 7   Dt_Customer          2240 non-null   object 
 8   Recency              2240 non-null   int64  
 9   MntWines             2240 non-null   int64  
 10  MntFruits            2240 non-null   int64  
 11  MntMeatProducts      2240 non-null   int64  
 12  MntFishProducts      2240 non-null   int64  
 13  MntSweetProducts     2240 non-null   int64  
 14  MntGoldProds         2240 non-null   int64  
 15  NumDealsPurchases    2240 non-null   int64  
 16  NumWebPurchases      2240 non-null   int64  
 17  NumCatalogPurchases  2240 non-null   int64  
 18  NumStorePurchases    2240 non-null   int64  
 19  NumWebVisitsMonth    2240 non-null   int64  
 20  AcceptedCmp3         2240 non-null   int64  
 21  AcceptedCmp4         2240 non-null   int64  
 22  AcceptedCmp5         2240 non-null   int64  
 23  AcceptedCmp1         2240 non-null   int64  
 24  AcceptedCmp2         2240 non-null   int64  
 25  Complain             2240 non-null   int64  
 26  Z_CostContact        2240 non-null   int64  
 27  Z_Revenue            2240 non-null   int64  
 28  Response             2240 non-null   int64  
dtypes: float64(1), int64(25), object(3)
memory usage: 507.6+ KB
```

In [73]:

```
#convert the columns of DT_Customer to date type..
data["Dt_Customer"] = pd.to_datetime(data["Dt_Customer"])
```

In [74]:

```
#to know the last day..
data['Dt_Customer'].max()
```

Out[74]:

```
Timestamp('2014-12-06 00:00:00')
```

In [75]:

```
data['last_day']=pd.to_datetime('2014-12-06')
data['No_Days']=(data['last_day']-data['Dt_Customer']).dt.days
```

In [76]:
```
data
```

Out[76]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines
0	5524	1957	Graduation	Single	58138.0	0	0	2012-04-09	58	635
1	2174	1954	Graduation	Single	46344.0	1	1	2014-08-03	38	11
2	4141	1965	Graduation	Together	71613.0	0	0	2013-08-21	26	426
3	6182	1984	Graduation	Together	26646.0	1	0	2014-10-02	26	11
4	5324	1981	PhD	Married	58293.0	1	0	2014-01-19	94	173
...	...	...	...	...	...	...	...	...	...	...
2235	10870	1967	Graduation	Married	61223.0	0	1	2013-06-13	46	709
2236	4001	1946	PhD	Together	64014.0	2	1	2014-10-06	56	406
2237	7270	1981	Graduation	Divorced	56981.0	0	0	2014-01-25	91	908
2238	8235	1956	Master	Together	69245.0	0	1	2014-01-24	8	428
2239	9405	1954	PhD	Married	52869.0	1	1	2012-10-15	40	84

2240 rows × 31 columns

In [77]:
```
data['No_Days'].max()
```

Out[77]:
```
1063
```

In [78]:
```
#make columns of age ..
data['age']=2023-data["Year_Birth"]
```

In [79]:

```
data
```

Out[79]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines
0	5524	1957	Graduation	Single	58138.0	0	0	2012-04-09	58	635
1	2174	1954	Graduation	Single	46344.0	1	1	2014-08-03	38	11
2	4141	1965	Graduation	Together	71613.0	0	0	2013-08-21	26	426
3	6182	1984	Graduation	Together	26646.0	1	0	2014-10-02	26	11
4	5324	1981	PhD	Married	58293.0	1	0	2014-01-19	94	173
...	...	...	...	...	...	...	...	...	...	...
2235	10870	1967	Graduation	Married	61223.0	0	1	2013-06-13	46	709
2236	4001	1946	PhD	Together	64014.0	2	1	2014-10-06	56	406
2237	7270	1981	Graduation	Divorced	56981.0	0	0	2014-01-25	91	908
2238	8235	1956	Master	Together	69245.0	0	1	2014-01-24	8	428
2239	9405	1954	PhD	Married	52869.0	1	1	2012-10-15	40	84

2240 rows × 32 columns

In [80]:

```
pd.set_option('display.max_columns', None)
data.head(10)
```

Out[80]:

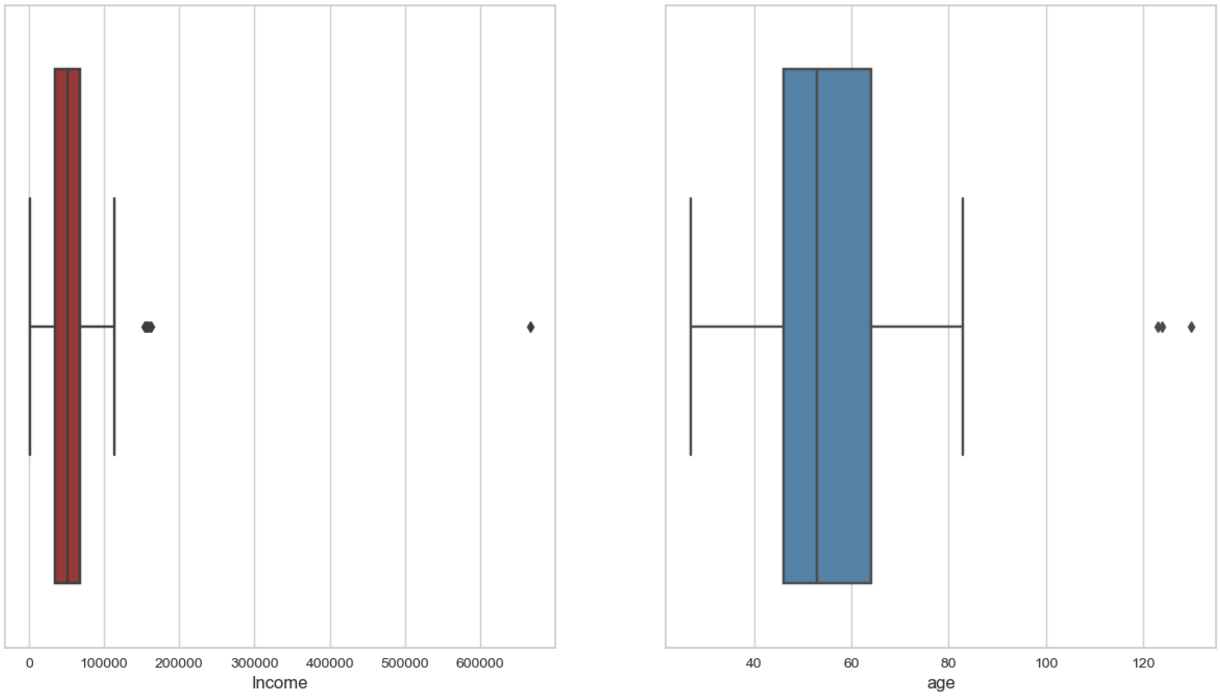
	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines	Mr
0	5524	1957	Graduation	Single	58138.0	0	0	2012-04-09	58	635	
1	2174	1954	Graduation	Single	46344.0	1	1	2014-08-03	38	11	
2	4141	1965	Graduation	Together	71613.0	0	0	2013-08-21	26	426	
3	6182	1984	Graduation	Together	26646.0	1	0	2014-10-02	26	11	
4	5324	1981	PhD	Married	58293.0	1	0	2014-01-19	94	173	
5	7446	1967	Master	Together	62513.0	0	1	2013-09-09	16	520	
6	965	1971	Graduation	Divorced	55635.0	0	1	2012-11-13	34	235	
7	6177	1985	PhD	Married	33454.0	1	0	2013-08-05	32	76	
8	4855	1974	PhD	Together	30351.0	1	0	2013-06-06	19	14	
9	5899	1950	PhD	Together	5648.0	1	1	2014-03-13	68	28	

In [81]:

```
#Checking the ouliers in age and income columns.
plt.figure(figsize=(15,8))
plt.subplot(1,2,1)
plt.xlabel='income'
sns.boxplot(data=data,x='Income',color='brown')
plt.subplot(1,2,2)
plt.xlabel='age'
sns.boxplot(data=data,x='age',color='steelblue')
```

Out[81]:

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



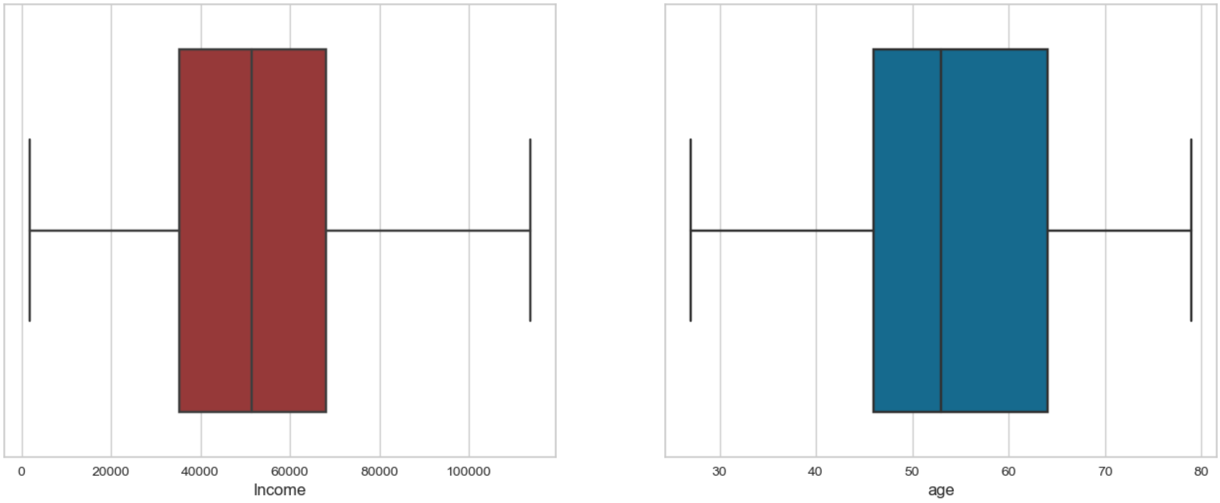
So, there are outliers visible in graph

In [82]:

```
#delete the outliers..
#from age column
data = data[data['age'] < 80]
#from income column
data=data[data['Income']<150000]
```

In [83]:
```
plt.figure(figsize=(16,6))
plt.subplot(1,2,1)
plt.xlabel='income'
sns.boxplot(data=data,x='Income',color = "brown")
plt.subplot(1,2,2)
plt.xlabel='age'
sns.boxplot(data=data,x='age')
```

Out[83]:
```
<Axes: xlabel='age'>
```



In [84]:
```
data['Marital_Status'].value_counts()
```

Out[84]:
```
Married     858
Together    575
Single      477
Divorced    228
Widow        75
Alone         3
Absurd        2
YOLO          2
Name: Marital_Status, dtype: int64
```

In [85]:
```
#handling Marital_Status column
data['relationship']=data['Marital_Status'].replace({'Married':'in_relationship' ,
                'Together':'in_relationship' , 'Single':'single' ,  'Divorced':'single',
                    'YOLO':'single' , 'Absurd':'single' , 'Widow':'single' ,'Alone':'single'})
```

In [86]:
```
data.columns
```

Out[86]:
```
Index(['ID', 'Year_Birth', 'Education', 'Marital_Status', 'Income', 'Kidhome',
       'Teenhome', 'Dt_Customer', 'Recency', 'MntWines', 'MntFruits',
       'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts',
       'MntGoldProds', 'NumDealsPurchases', 'NumWebPurchases',
       'NumCatalogPurchases', 'NumStorePurchases', 'NumWebVisitsMonth',
       'AcceptedCmp3', 'AcceptedCmp4', 'AcceptedCmp5', 'AcceptedCmp1',
       'AcceptedCmp2', 'Complain', 'Z_CostContact', 'Z_Revenue', 'Response',
       'last_day', 'No_Days', 'age', 'relationship'],
      dtype='object')
```

In [87]:
```
data['members_home']=data['Kidhome']+data['Teenhome']+data['relationship'].replace({'single':0
```

In [88]:
```
data['AcceptedCmp'] = data['AcceptedCmp1'] + data['AcceptedCmp2'] + data['AcceptedCmp3']
+ data['AcceptedCmp4'] + data['AcceptedCmp5'] + data['Response']
```

Out[88]:
```
0       1
1       0
2       0
3       0
4       0
       ..
2235    0
2236    0
2237    1
2238    0
2239    1
Length: 2220, dtype: int64
```

In [89]:
```
data['num_purchases'] = data['NumWebPurchases'] + data['NumCatalogPurchases'] + data['NumStore
+ data['NumDealsPurchases']
```

Out[89]:
```
0       3
1       2
2       1
3       2
4       5
       ..
2235    2
2236    7
2237    1
2238    2
2239    3
Name: NumDealsPurchases, Length: 2220, dtype: int64
```

In [90]:
```
data['expenses'] = data['MntWines'] + data['MntFruits'] + data['MntMeatProducts']
+ data['MntFishProducts'] + data['MntSweetProducts'] + data['MntGoldProds']
```

Out[90]:
```
0       348
1         9
2       174
3        18
4        88
       ...
2235    407
2236      8
2237     68
2238    171
2239     24
Length: 2220, dtype: int64
```

In [91]:
```
#dropping unnecessary columns
data.drop(labels=['Marital_Status','ID','last_day','Year_Birth','Dt_Customer','last_day', 'Kid
                  'MntWines', 'MntFruits','MntMeatProducts', 'MntFishProducts','MntSweetProduc
                  'NumDealsPurchases', 'NumWebPurchases','NumCatalogPurchases', 'NumStorePurch
                 'AcceptedCmp3', 'AcceptedCmp4', 'AcceptedCmp5', 'AcceptedCmp1',
                'AcceptedCmp2','Z_CostContact', 'Z_Revenue',"Recency", "Complain"], axis=1, in
```

In [92]:
```
data.columns
```

Out[92]:
```
Index(['Education', 'Income', 'Response', 'No_Days', 'age', 'relationship',
       'members_home', 'AcceptedCmp', 'num_purchases', 'expenses'],
      dtype='object')
```

In [93]:

```
data
```

Out[93]:

	Education	Income	Response	No_Days	age	relationship	members_home	AcceptedCmp	num_purchases
0	Graduation	58138.0	1	971	66	single	0	0	22
1	Graduation	46344.0	0	125	69	single	2	0	4
2	Graduation	71613.0	0	472	58	in_relationship	1	0	20
3	Graduation	26646.0	0	65	39	in_relationship	2	0	6
4	PhD	58293.0	0	321	42	in_relationship	2	0	14
...	...	...	...	...	...	...	...	...	...
2235	Graduation	61223.0	0	541	56	in_relationship	2	0	16
2236	PhD	64014.0	0	61	77	in_relationship	4	1	15
2237	Graduation	56981.0	0	315	42	single	0	0	18
2238	Master	69245.0	0	316	67	in_relationship	2	0	21
2239	PhD	52869.0	1	782	69	in_relationship	3	0	8

2220 rows × 10 columns

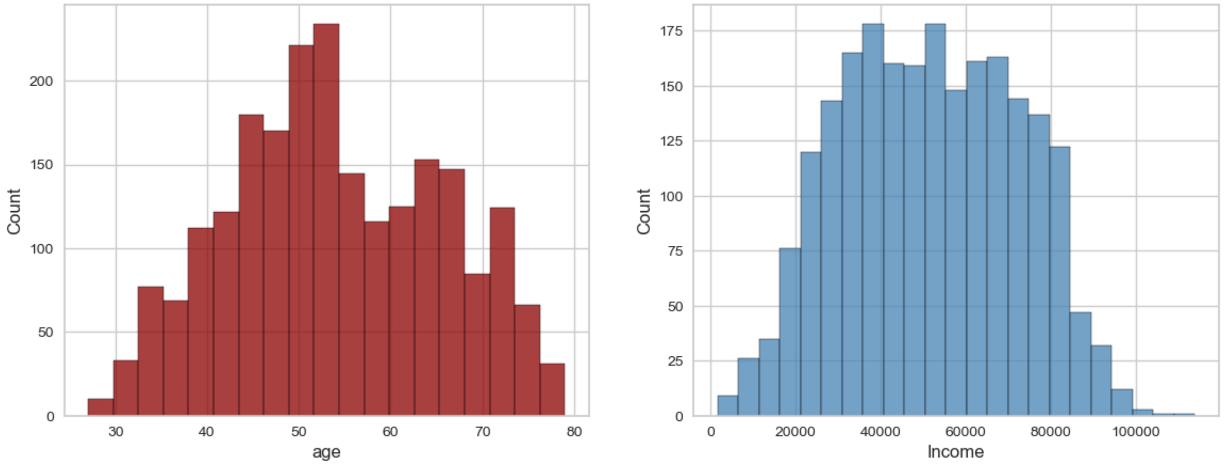
Make some plots

In [94]:

```
plt.figure(figsize=(14, 5))
plt.subplot(1,2,1)
sns.histplot(data,x='age',color = "darkred")
plt.subplot(1,2,2)
sns.histplot(data,x='Income',color='steelblue')
```

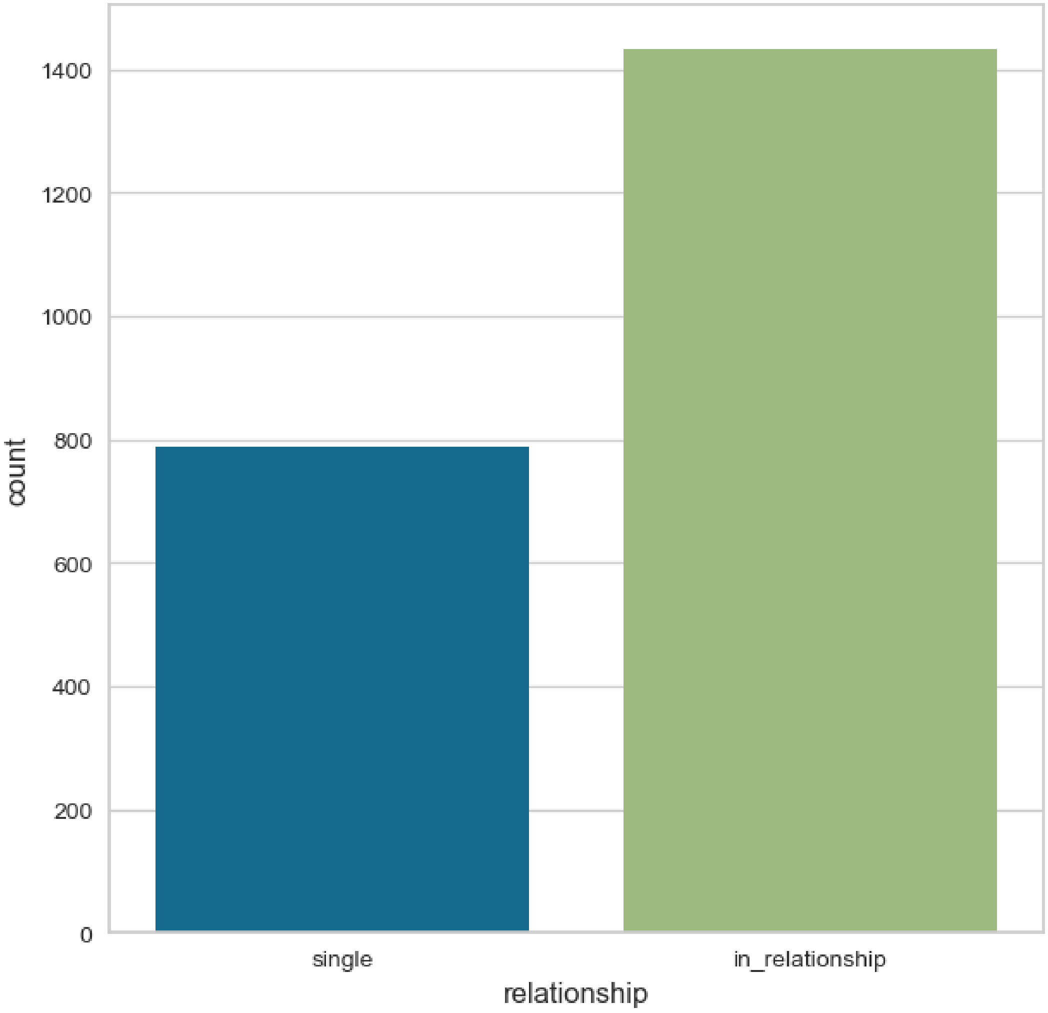
Out[94]:

```
<Axes: xlabel='Income', ylabel='Count'>
```



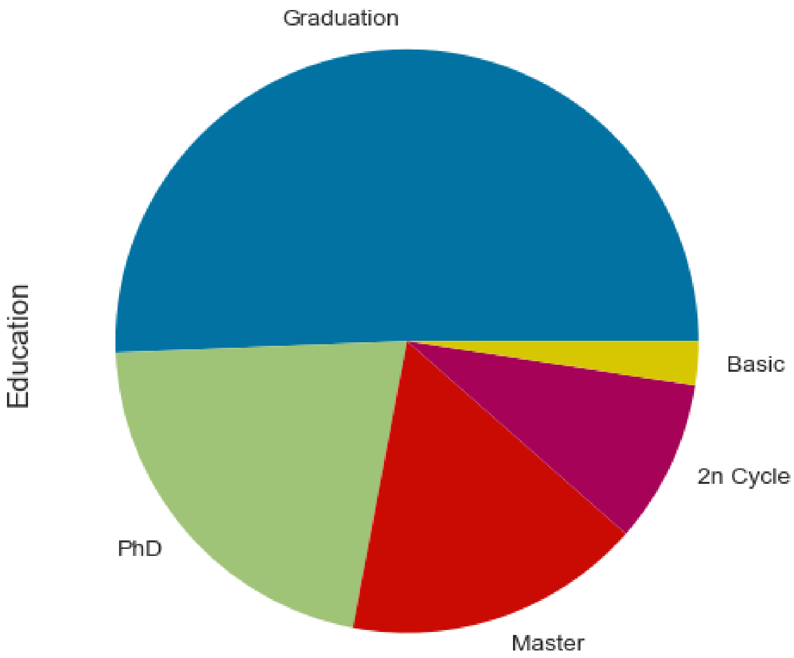
In [95]:
```
plt.figure(figsize=(7,7))
sns.countplot(data,x='relationship')
```

Out[95]:
```
<Axes: xlabel='relationship', ylabel='count'>
```

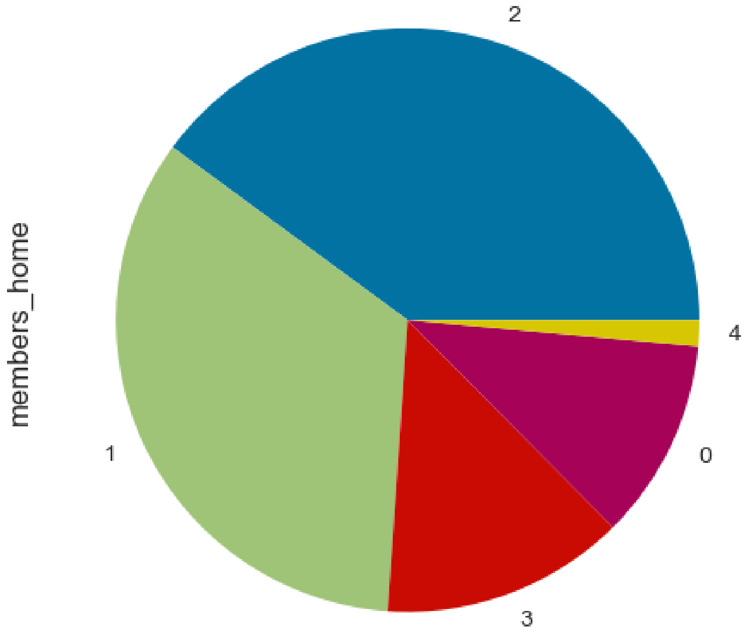


```
In [96]: #pieplot of eduction
         plt.plot(figsize=(10,10))
         data.Education.value_counts().plot(kind='pie')

Out[96]: <Axes: ylabel='Education'>
```



```
In [97]: #numbers of members in family.
         plt.plot(figsize=(10,10))
         data.members_home.value_counts().plot(kind='pie')
         plt.show()
```



In [98]:
```
data.columns
```

Out[98]:
```
Index(['Education', 'Income', 'Response', 'No_Days', 'age', 'relationship',
       'members_home', 'AcceptedCmp', 'num_purchases', 'expenses'],
      dtype='object')
```

Preprocess the data

In [99]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2220 entries, 0 to 2239
Data columns (total 10 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Education      2220 non-null   object 
 1   Income         2220 non-null   float64
 2   Response       2220 non-null   int64  
 3   No_Days        2220 non-null   int64  
 4   age            2220 non-null   int64  
 5   relationship   2220 non-null   object 
 6   members_home   2220 non-null   int64  
 7   AcceptedCmp    2220 non-null   int64  
 8   num_purchases  2220 non-null   int64  
 9   expenses       2220 non-null   int64  
dtypes: float64(1), int64(7), object(2)
memory usage: 190.8+ KB
```

In [100]:
```
#convert education and relationship to num values..
data['Education']=  preprocessing.LabelEncoder().fit_transform(data['Education'])
data['relationship']=  preprocessing.LabelEncoder().fit_transform(data['relationship'])
```

In [101]:
```
#education after converting
data.head()
```

Out[101]:

	Education	Income	Response	No_Days	age	relationship	members_home	AcceptedCmp	num_purchases	expe
0	2	58138.0	1	971	66	1	0	0	22	
1	2	46344.0	0	125	69	1	2	0	4	
2	2	71613.0	0	472	58	0	1	0	20	
3	2	26646.0	0	65	39	0	2	0	6	
4	4	58293.0	0	321	42	0	2	0	14	

In [102]:
```
scaler=StandardScaler()
scaled_features = scaler.fit_transform(data.values)
scaled_data = pd.DataFrame(scaled_features, index=data.index, columns=data.columns)
```

In [103]:
```
#reduce features of the data to 4 ..
pca = PCA(n_components=4)
data_pca = pca.fit_transform(scaled_data)
```

In [104]:
```
data_pca.shape
```

Out[104]:
```
(2220, 4)
```

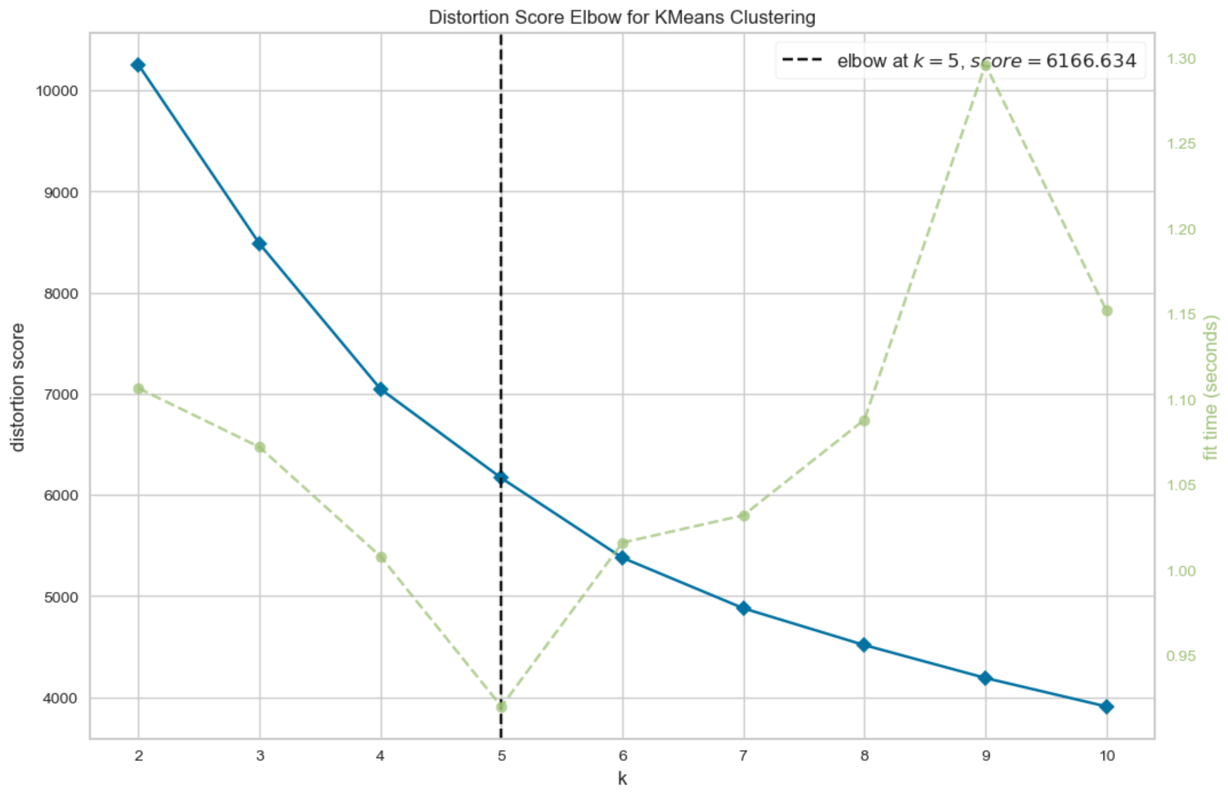
clustering Time

In [105]:

```
from yellowbrick.cluster import KElbowVisualizer
from sklearn.cluster import KMeans
```

In [106]:

```
plt.figure(figsize=(12, 8))
elbow_graph = KElbowVisualizer(KMeans(random_state=123), k=10)
elbow_graph.fit(data_pca)
elbow_graph.show()
```



Out[106]:

```
<Axes: title={'center': 'Distortion Score Elbow for KMeans Clustering'}, xlabel='k', ylabel
='distortion score'>
```

In [107]:

```
import warnings
warnings.filterwarnings('ignore')
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

number of clusters is 5

In [117]:

```
kmeans = KMeans(n_clusters =5 )
Cluster = kmeans.fit_predict(data_pca)
```

In [118]:

```
data['Cluster']=Cluster
```

In [119]:

```
Cluster.min(),Cluster.max()
```

Out[119]:

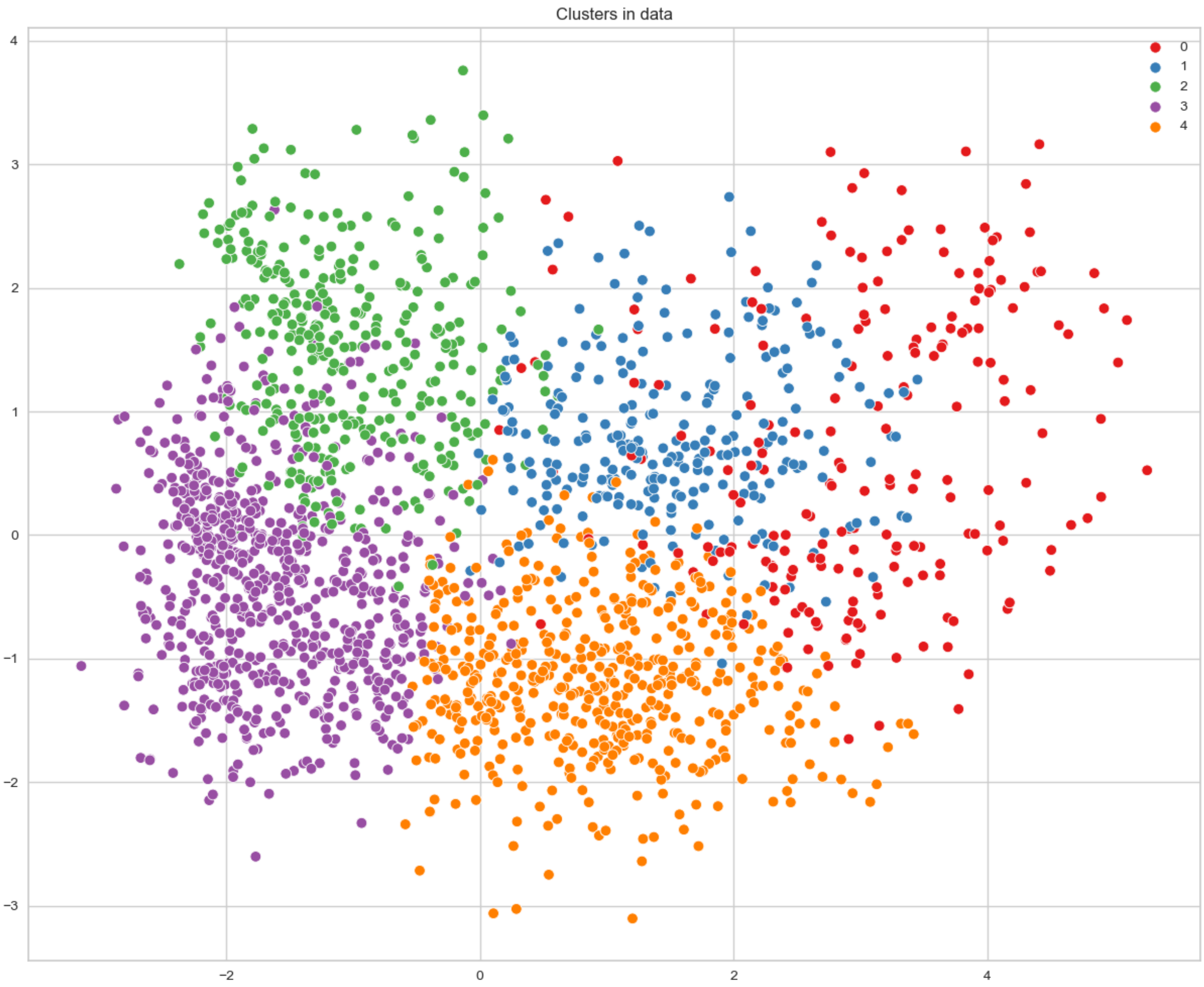
```
(0, 4)
```

```
In [120]: Cluster
```

```
Out[120]: array([0, 2, 4, ..., 1, 4, 3])
```

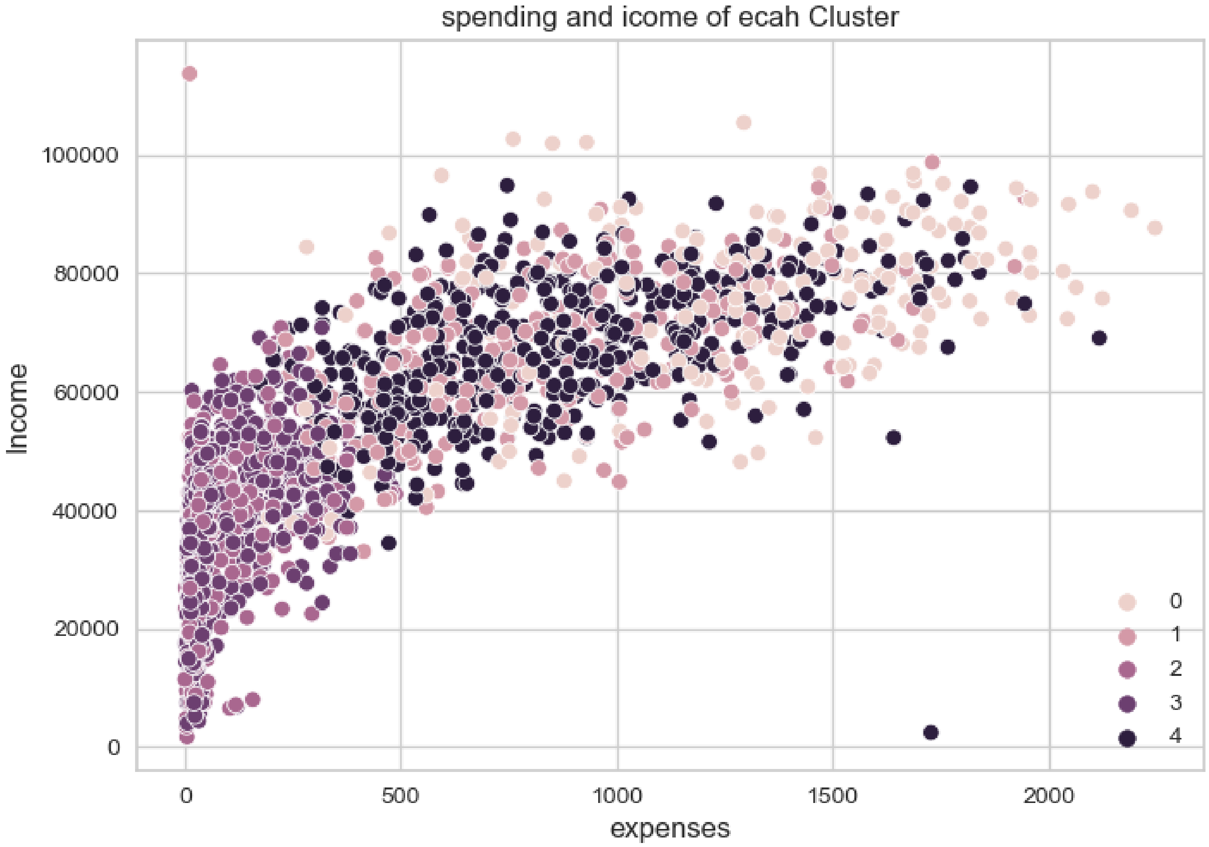
```
In [121]: #ploting cluster...
          plt.figure(figsize=(15,12))
          sns.scatterplot(x=data_pca[:, 0], y=data_pca[:, 1], hue=Cluster,s=60, palette='Set1')
          plt.title('Clusters in data')
```

```
Out[121]: Text(0.5, 1.0, 'Clusters in data')
```



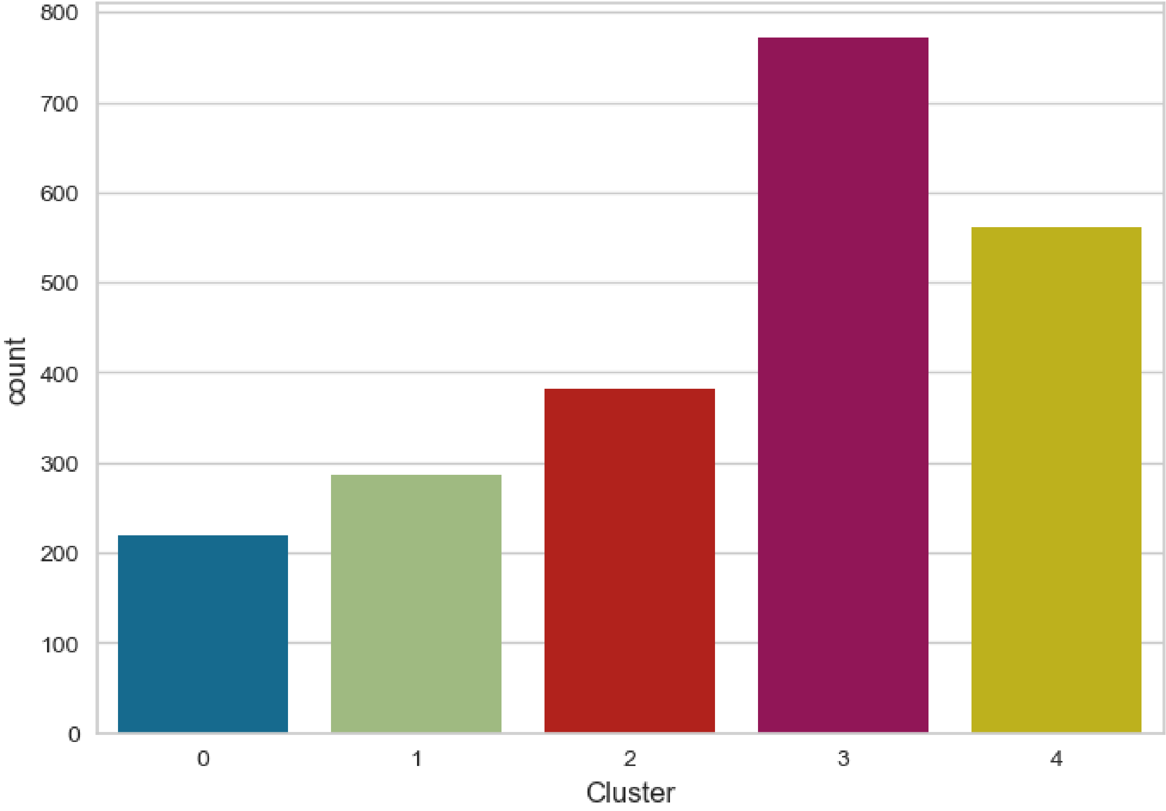
Make some plots and identify the spending capabilities and income for each cluster

In [122]:
```
pl = sns.scatterplot(data = data, x=data["expenses"], y=data["Income"], hue=data["Cluster"])
pl.set_title("spending and icome of ecah Cluster")
plt.legend()
plt.show()
```



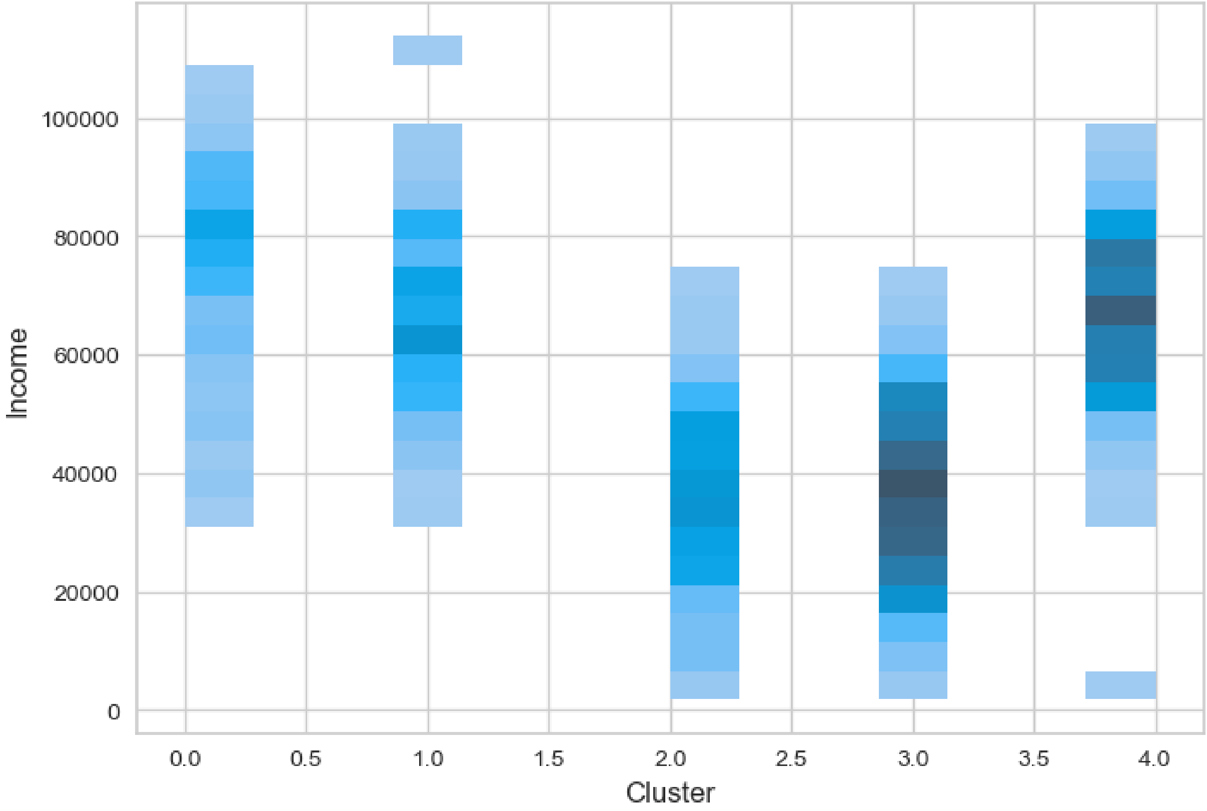
```
In [123]: sns.countplot(x=data['Cluster'])
```

```
Out[123]: <Axes: xlabel='Cluster', ylabel='count'>
```



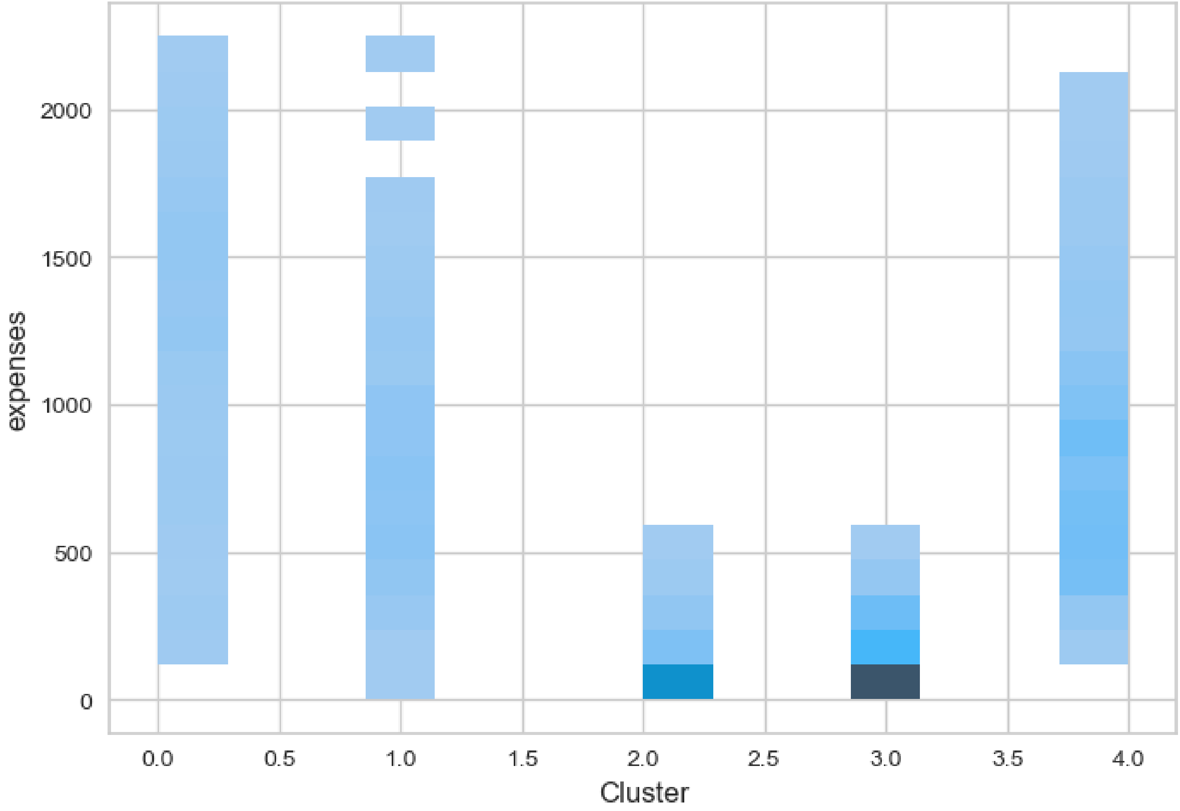
```
In [124]: sns.histplot(x=data['Cluster'],y=data['Income'])
```

```
Out[124]: <Axes: xlabel='Cluster', ylabel='Income'>
```



```
In [125]: sns.histplot(x=data['Cluster'],y=data['expenses'])
```

```
Out[125]: <Axes: xlabel='Cluster', ylabel='expenses'>
```



```
In [ ]:
```

```
In [ ]:
```