In [442]:

```
import numpy as np
import pandas as pd
import datetime
import matplotlib
import matplotlib.pyplot as plt
from matplotlib import colors
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from yellowbrick.cluster import KElbowVisualizer
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt, numpy as np
from mpl_toolkits.mplot3d import Axes3D
from sklearn.cluster import AgglomerativeClustering
from matplotlib.colors import ListedColormap
from sklearn.preprocessing import RobustScaler
from sklearn import metrics
import warnings
import sys
if not sys.warnoptions:
    warnings.simplefilter("ignore")
np.random.seed(42)
```

In [ ]:

```
```

In [443]:

```
data= pd.read_csv("C:\\Users\\User\\Desktop\\samsung\\project-samsung\\Final p
```

In [444]:

```
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 [445]:

```
data
```

Out[445]:

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

2240 rows × 29 columns

Content

ID: Customer's unique identifier

Year_Birth: Customer's birth year

Education: Customer's education level

Marital_Status: Customer's marital status

Income: Customer's yearly household income

Kidhome: Number of children in customer's household

Teenhome: Number of teenagers in customer's household

Dt_Customer: Date of customer's enrollment with the company

Recency: Number of days since customer's last purchase

Complain: 1 if the customer complained in the last 2 years, 0 otherwise

MntWines: Amount spent on wine in last 2 years

MntFruits: Amount spent on fruits in last 2 years

MntMeatProducts: Amount spent on meat in last 2 years

MntFishProducts: Amount spent on fish in last 2 years

MntSweetProducts: Amount spent on sweets in last 2 years

MntGoldProds: Amount spent on gold in last 2 years

NumDealsPurchases: Number of purchases made with a discount

AcceptedCmp1: 1 if customer accepted the offer in the 1st campaign, 0 otherwise

AcceptedCmp2: 1 if customer accepted the offer in the 2nd campaign, 0 otherwise

AcceptedCmp3: 1 if customer accepted the offer in the 3rd campaign, 0 otherwise

AcceptedCmp4: 1 if customer accepted the offer in the 4th campaign, 0 otherwise

AcceptedCmp5: 1 if customer accepted the offer in the 5th campaign, 0 otherwise

Response: 1 if customer accepted the offer in the last campaign, 0 otherwise

NumWebPurchases: Number of purchases made through the company’s website

NumCatalogPurchases: Number of purchases made using a catalogue

NumStorePurchases: Number of purchases made directly in stores

NumWebVisitsMonth: Number of visits to company’s website in the last month

Questions:

1-Relationship between Date of customer's enrollment and marital status?

2-The relationship between the number of purchases and marital status?

3-The relationship between the number of purchases and the number of children and the family size?

4-What does age have to do with the number of purchases?

5-What is the relationship between education and income?

6-What is the relationship between income, family size and number of children?

7-What is the relationship between income and the number of purchases?

8-What is the relationship between the number of purchases from the website and the number of website visits?

9-What is the relationship between the number of purchases from a Deal with the number of purchases from the website, the number of purchases from the catalog, and the number of purchases from the store?

10-What is the relationship between the number of purchases from a Deal with accepted cmp 1 ,accepted cmp 2,accepted cmp 3 ,accepted cmp 4 ,accepted cmp 5 and Response?

11-What is the relationship between the complaint and Date of customer's enrollment?

In [446]:

```
data
```

Out[446]:

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

2240 rows × 29 columns

Add new features or modify features to better clarify the data

In [447]:

```
data["Dt_Customer"] = pd.to_datetime(data["Dt_Customer"])
dates = []
for i in data["Dt_Customer"]:
    i = i.date()
    dates.append(i)
#Dates of the newest and oldest recorded customer
print("Date of registration of the company's newest client:",max(dates))
print("Date of registration of the company's oldest client:",min(dates))
```

```
Date of registration of the company's newest client: 2014-12-06
Date of registration of the company's oldest client: 2012-01-08
```

In [ ]:

In [448]:

```
#Created a feature "Customer_From_days"
days = []
d1 = max(dates) #taking it to be the newest customer
for i in dates:
    delta = d1 - i
    days.append(delta)
data["Customer_From_days"] = days
data["Customer_From_days"] = pd.to_numeric(data["Customer_From_days"], errors=
for i in range(len(data['Customer_From_days'])):
    t=0
    t=data['Customer_From_days'][i]
    data['Customer_From_days'][i]=t/60/60/24/1000000000
```

In [ ]:

In [449]:

```
# Create a feature that shows the age of the customer based on the date of bir
data["Age"] = 2021-data["Year_Birth"]
```

In [450]:

```
data
```

Out[450]:

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

2240 rows × 31 columns

In [451]:

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

Out[451]:

```
Married     864
Together    580
Single      480
Divorced    232
Widow        77
Alone         3
Absurd        2
YOLO          2
Name: Marital_Status, dtype: int64
```

In [452]:

```
# We will change the values[Alone,Absurd,YOLO] because there are few of them a
data['Marital_Status'].replace('Alone','Single',inplace=True)
data['Marital_Status'].replace('Absurd','Single',inplace=True)
data['Marital_Status'].replace('YOLO','Single',inplace=True)

data["Living_With"]=data["Marital_Status"].replace({"Married":"Partner", "Toge

#Feature indicating total children living in the household
data["Num_Children"]=data["Kidhome"]+data["Teenhome"]

#Feature for total members in the householde
data["Family_Size"] = data["Living_With"].replace({"Alone": 1, "Partner":2})+
```

In [ ]:

In [ ]:

Q1: Relationship between Date of customer's enrollment and marital status ?

In [453]:

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

Out[453]:

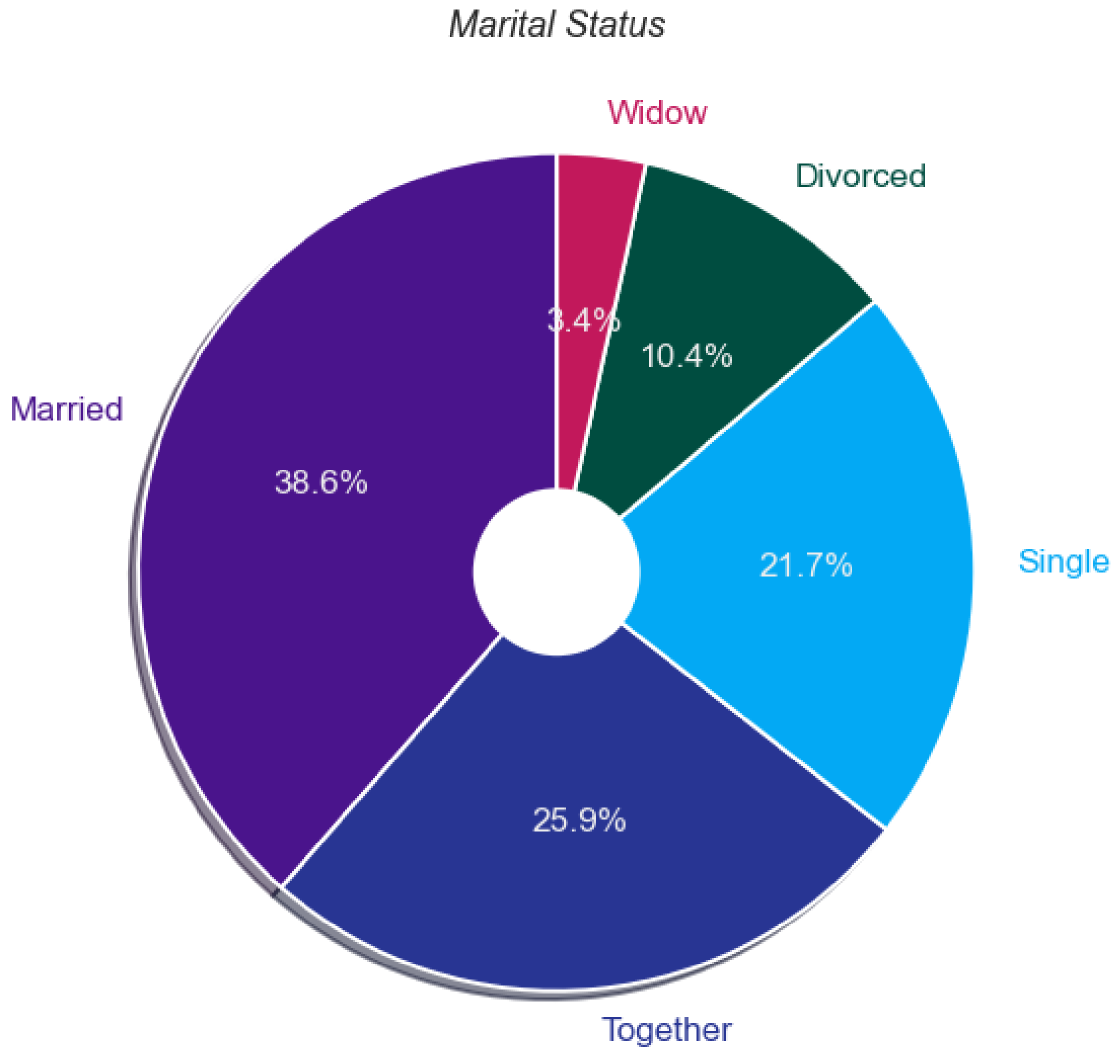
```
Married     864
Together    580
Single      487
Divorced    232
Widow        77
Name: Marital_Status, dtype: int64
```

In [454]:

```
fig, ax = plt.subplots(figsize=(15, 8))
colors2=['#4a148c','#283593','#03a9f4','#004d40','#c2185b']
patches, texts, pcts = ax.pie(
    data['Marital_Status'].value_counts(), labels=[*data['Marital_Status'].val
    ,wedgeprops={'linewidth': 2.0, 'edgecolor': 'white'},
    textprops={'size': 'x-large'},
    startangle=90)

for i, patch in enumerate(patches):
    texts[i].set_color(patch.get_facecolor())
plt.setp(pcts, color='#EEEEEE')
plt.setp(texts, fontweight=200)
centre_circle = plt.Circle((0,0),0.20,fc='white')
plt.gcf().gca().add_artist(centre_circle)
plt.tight_layout()
plt.title(label='Marital Status',fontsize=18,fontstyle='italic')

plt.tight_layout()
```



The largest proportion of the company's customers are people who live with partners

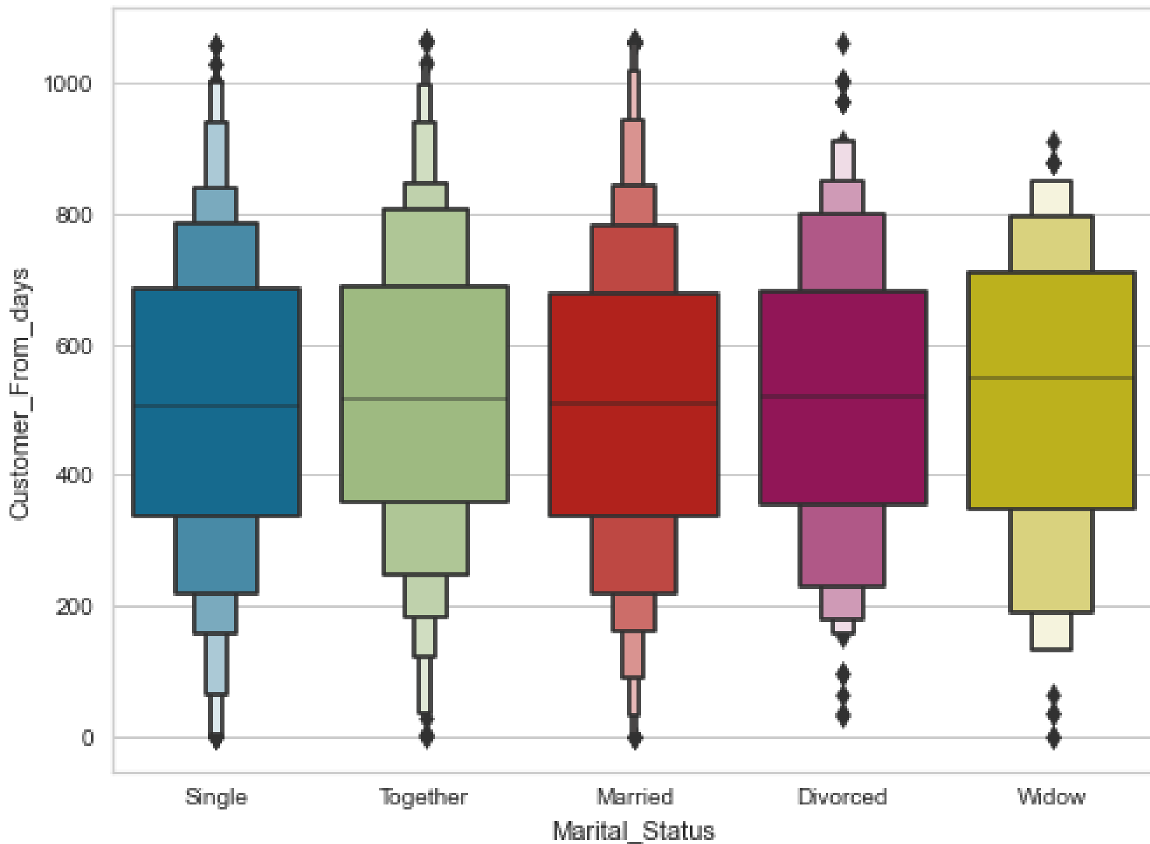
In [ ]:

In [455]:

```
plt.figure(figsize=(8,6))
sns.boxenplot(data=data,x='Marital_Status',y='Customer_From_days')
```

Out[455]:

<AxesSubplot:xlabel='Marital_Status', ylabel='Customer_From_day
s'>



There is no relationship between marital status and the date of joining the company

In [ ]:

```

```

Q2: The relationship between the number of purchases and marital status?

In [456]:

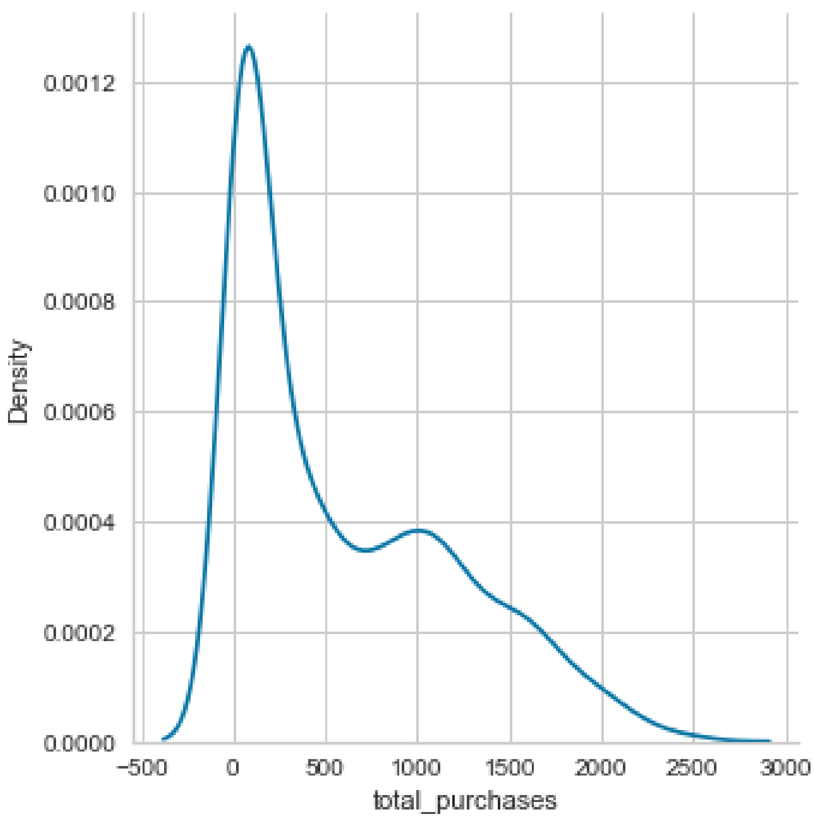
```
# Create a feature that shows the number of purchases for customers
data['total_purchases']=data['MntFishProducts']+data["MntFruits"]+data['MntGol
```

In [457]:

```
sns.displot(data,x='total_purchases',kind='kde')
```

Out[457]:

```
<seaborn.axisgrid.FacetGrid at 0x1e87fd32250>
```



In [458]:

```
print('Min:'+str(min(data['total_purchases'])),'| Max: '+str(max(data['total_p
```

Min:5 | Max: 2525

In [459]:

```
inter=pd.interval_range(start=5,freq=210, end= 2525)
inter
```

Out[459]:

```
IntervalIndex([(5, 215], (215, 425], (425, 635], (635, 845], (8
45, 1055] ... (1475, 1685], (1685, 1895], (1895, 2105], (2105,
2315], (2315, 2525]], dtype='interval[int64, right]')
```

In [460]:

```
# We will classify the number of purchases into more than one category
s=5
name_class=[]
for i in range(12):
    t='class ' + str(i) +" : ("+str(s)+ ", " +str(s+210) +')'
    name_class.append(t)
    s=s+210
inter=[5,215,425,635,845,1055,1265,1475,1685,1895,2105,2315,2525]
data['purchase_quantity']=pd.cut(data['total_purchases'],bins=inter,labels=nam
```

In [461]:

```
data['purchase_quantity'].value_counts()
```

Out[461]:

```
class 0 : (5, 215)          920
class 1 : (215, 425)        246
class 4 : (845, 1055)       184
class 2 : (425, 635)        177
class 5 : (1055, 1265)      172
class 3 : (635, 845)        161
class 6 : (1265, 1475)      117
class 7 : (1475, 1685)      117
class 8 : (1685, 1895)       67
class 9 : (1895, 2105)       50
class 10 : (2105, 2315)      20
class 11 : (2315, 2525)       8
Name: purchase_quantity, dtype: int64
```

In [462]:

```
d={'num_clas':data['purchase_quantity'].value_counts(),'clas':name_class}

import plotly.express as px

fig = px.pie(d,values='num_clas', names='clas',labels='clas',color_discrete_se
fig.update_traces(textposition='inside', textinfo='percent+label')
fig.update_traces(textposition='inside', hole=.4, hoverinfo="label+percent+nam
fig.show()
```

That the largest percentage of the number of purchases made by customers and up to 41.1% was between 5 to 215 purchases and the more purchases the less the percentage

In [463]:

```
pre=[]
total=d['num_clas'].sum()
for i in d['num_clas']:
    n=i/total
    pre.append(n.round(3)*100)
```

In [464]:

```
d_1={'number':d['num_clas'] ,'pre_clas':pre}
df=pd.DataFrame(data=d_1)
```

In [465]:

```
df
```

Out[465]:

	number	pre_clas
class 0 : (5, 215)	920	41.1
class 1 : (215, 425)	246	11.0
class 4 : (845, 1055)	184	8.2
class 2 : (425, 635)	177	7.9
class 5 : (1055, 1265)	172	7.7
class 3 : (635, 845)	161	7.2
class 6 : (1265, 1475)	117	5.2
class 7 : (1475, 1685)	117	5.2
class 8 : (1685, 1895)	67	3.0
class 9 : (1895, 2105)	50	2.2
class 10 : (2105, 2315)	20	0.9
class 11 : (2315, 2525)	8	0.4

In [466]:

```
data.head()
```

Out[466]:

	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

5 rows × 36 columns

In [467]:

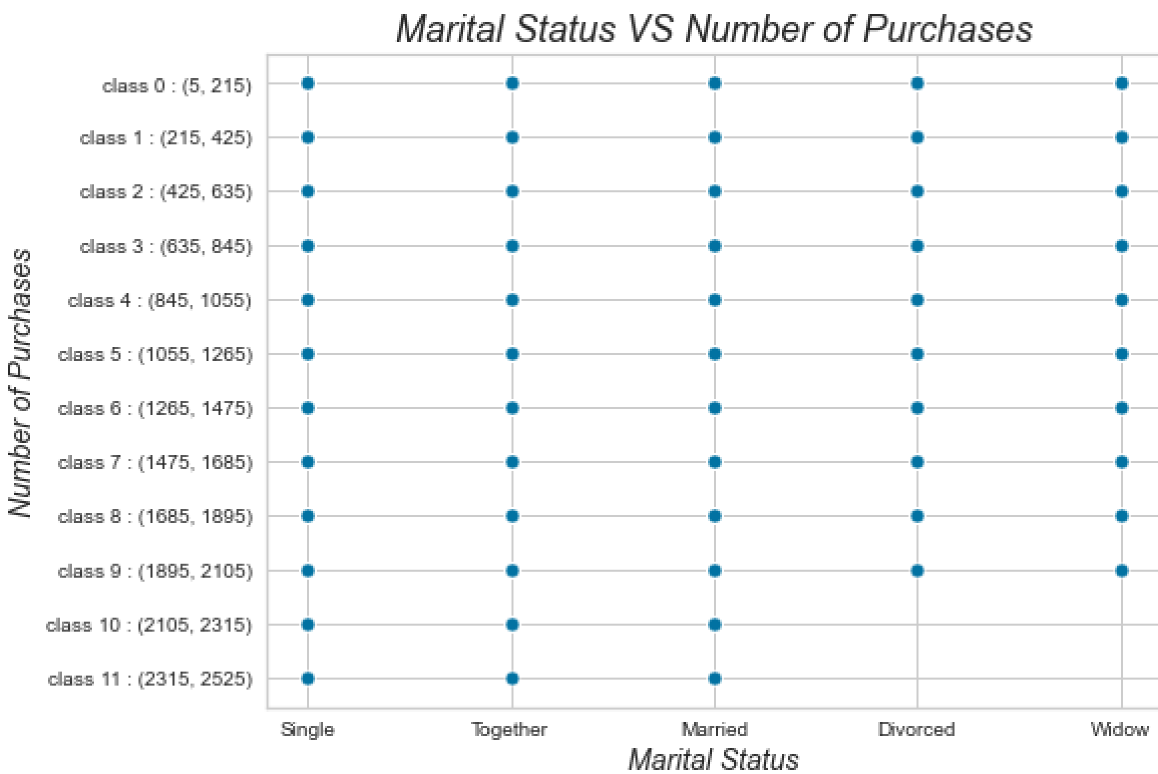
```
name_class
```

Out[467]:

```
['class 0 : (5, 215)',
 'class 1 : (215, 425)',
 'class 2 : (425, 635)',
 'class 3 : (635, 845)',
 'class 4 : (845, 1055)',
 'class 5 : (1055, 1265)',
 'class 6 : (1265, 1475)',
 'class 7 : (1475, 1685)',
 'class 8 : (1685, 1895)',
 'class 9 : (1895, 2105)',
 'class 10 : (2105, 2315)',
 'class 11 : (2315, 2525)']
```

In [468]:

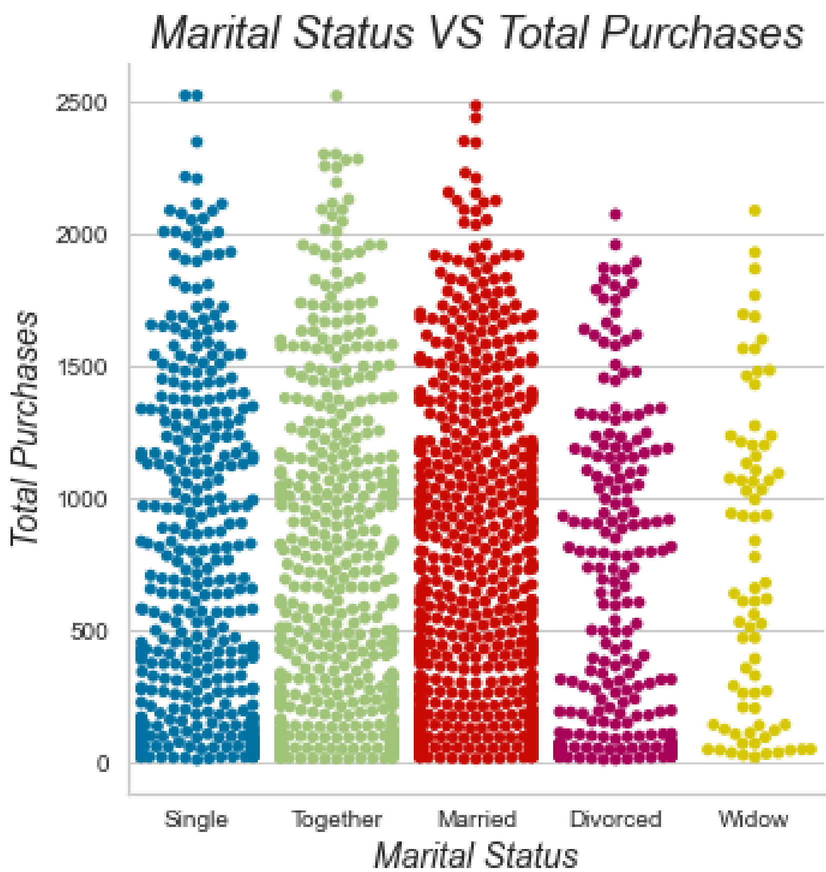
```
plt.figure(figsize=(8,6))
sns.scatterplot(data=data,x='Marital_Status',y='purchase_quantity')
plt.xlabel(fontsize=14,xlabel='Marital Status',fontstyle='italic')
plt.ylabel(fontsize=14,ylabel='Number of Purchases',fontstyle='italic')
plt.title(label='Marital Status VS Number of Purchases',fontsize=18,fontstyle=
plt.show()
```



Divorced and widowed clients are not included in the categories 10 and 11

In [469]:

```
sns.catplot(data=data,x='Marital_Status',y='total_purchases',kind='swarm')
plt.xlabel(fontsize=14,xlabel='Marital Status',fontstyle='italic')
plt.ylabel(fontsize=14,ylabel='Total Purchases',fontstyle='italic')
plt.title(label='Marital Status VS Total Purchases',fontsize=18,fontstyle='ita
plt.show()
```



In [ ]:

```

```

Q3: The relationship between the number of purchases and the number of children and the age them?

In [470]:

```
data['Kidhome'].value_counts()
```

Out[470]:

```
0    1293
1     899
2      48
Name: Kidhome, dtype: int64
```

In [471]:

```
data['Teenhome'].value_counts()
```

Out[471]:

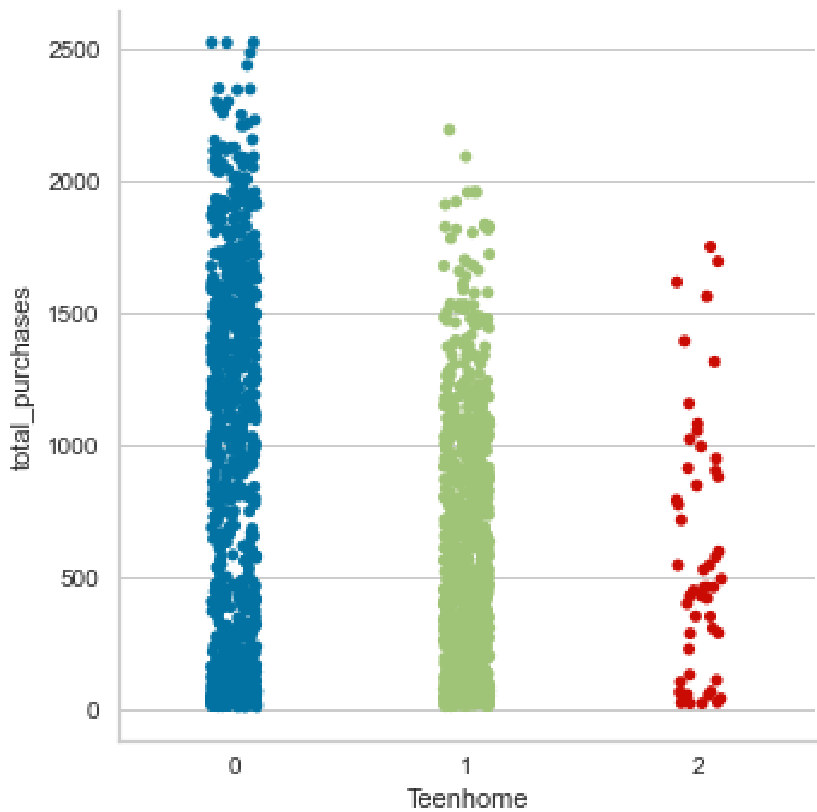
```
0    1158
1    1030
2      52
Name: Teenhome, dtype: int64
```

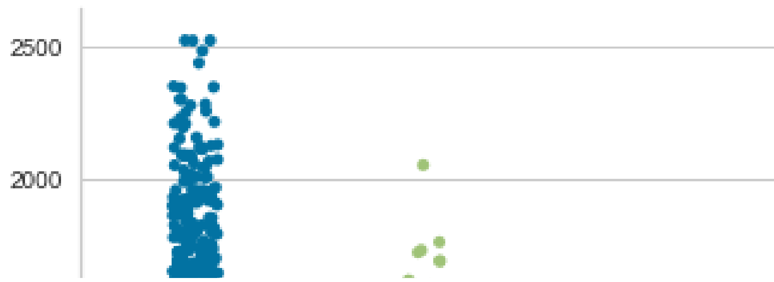
In [472]:

```
sns.catplot(data=data,x='Teenhome',y='total_purchases',kind='strip')
sns.catplot(data=data,x='Kidhome',y='total_purchases',kind='strip')
plt.subplot()
```

Out[472]:

<AxesSubplot:xlabel='Kidhome', ylabel='total_purchases'>





We see that customers with teenagers have more purchases than customers with children

In [473]:

```
data['Family_Size'].value_counts()
```

Out[473]:

```
3    889
2    764
4    301
1    254
5     32
Name: Family_Size, dtype: int64
```

In [474]:

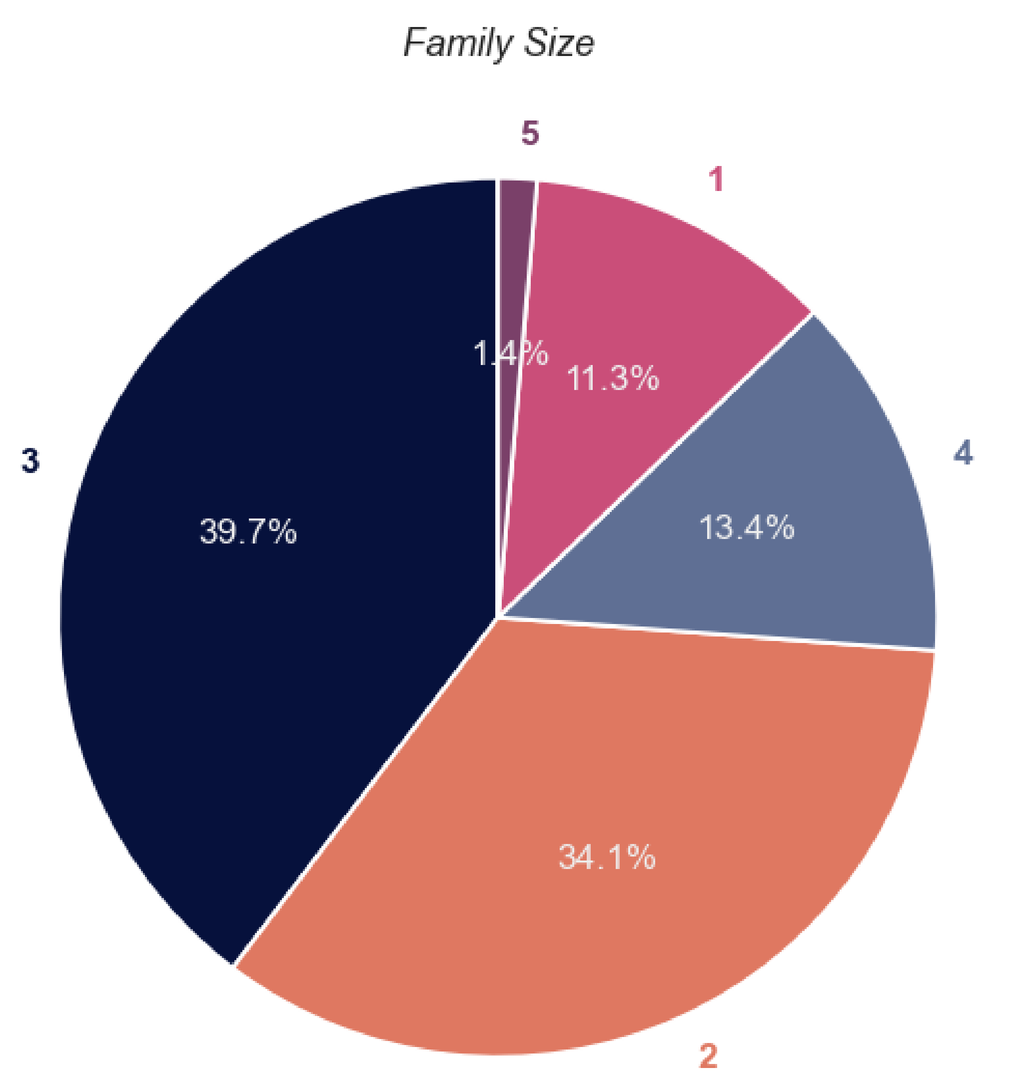
```
color_3=['#06113C','#DF7861','#5F6F94','#CA4E79','#7A4069']
```

In [475]:

```
fig, ax = plt.subplots(figsize=(15, 8))
patches, texts, pcts = ax.pie(
    data['Family_Size'].value_counts(), labels=[*data['Family_Size'].value_cou
    ,wedgeprops={'linewidth': 2.0, 'edgecolor': 'white'},
    textprops={'size': 'x-large'},
    startangle=90)

for i, patch in enumerate(patches):
    texts[i].set_color(patch.get_facecolor())
plt.setp(pcts, color='#EEEEEE')
plt.setp(texts, fontweight=600)
plt.tight_layout()
plt.title(label='Family Size',fontsize=18,fontstyle='italic')

plt.tight_layout()
```



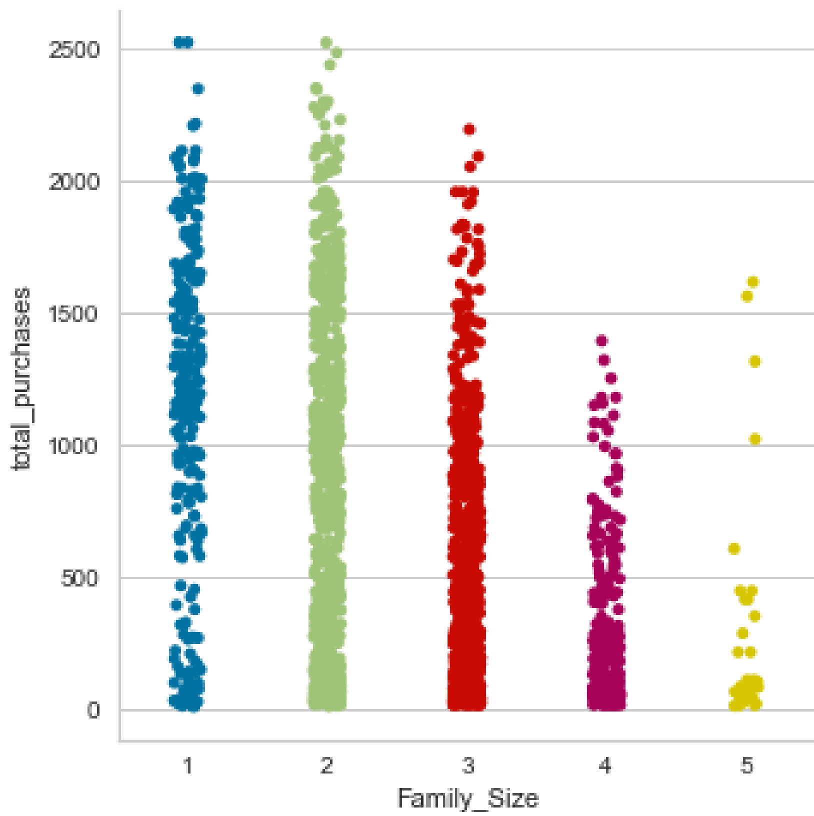
Most of the company's clients are families of two or 3

In [476]:

```
sns.catplot(data=data,x='Family_Size',y='total_purchases',kind='strip')
```

Out[476]:

```
<seaborn.axisgrid.FacetGrid at 0x1e8077d5610>
```



The figure shows that the higher the number of family members, the fewer purchases

In [ ]:

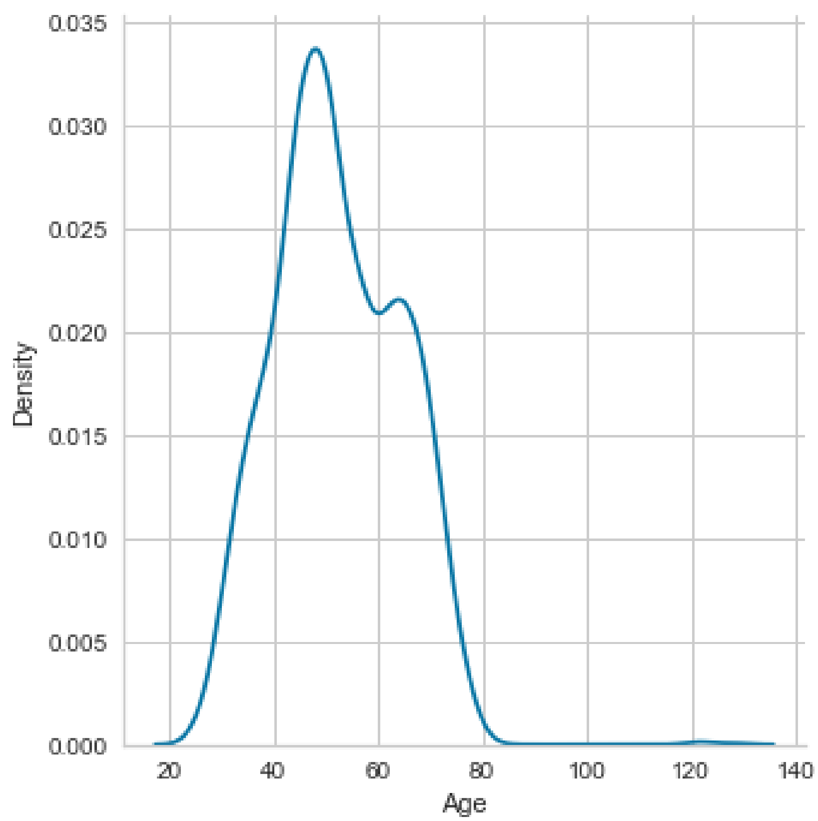
Q4: What does age have to do with the number of purchases?

In [477]:

```
sns.displot(data,x='Age',kind='kde')
```

Out[477]:

```
<seaborn.axisgrid.FacetGrid at 0x1e860fedac0>
```



In [478]:

```
print("Oldest customer:",max(data['Age']))
print("Youngest customer:",min(data['Age']))
```

```
Oldest customer: 128
Youngest customer: 25
```

In [479]:

```
data['Age'].mean()
```

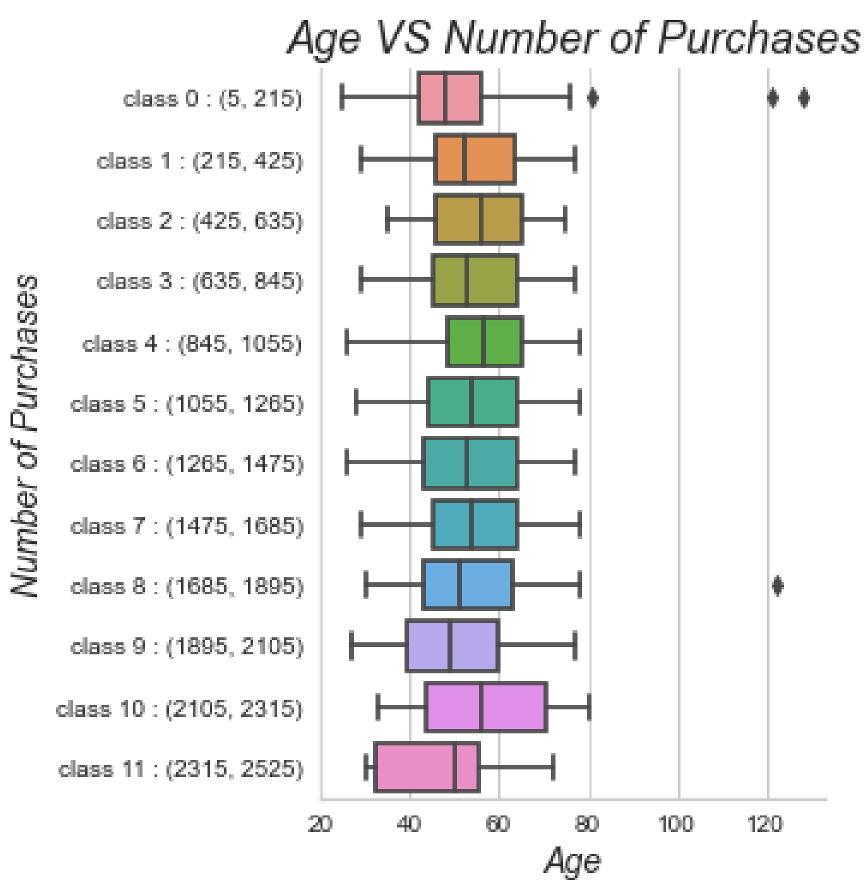
Out[479]:

```
52.19419642857143
```

Average customer age 52

In [480]:

```
sns.catplot(data=data,x='Age',y='purchase_quantity',kind='box')
plt.xlabel(fontsize=14,xlabel='Age',fontstyle='italic')
plt.ylabel(fontsize=14,ylabel='Number of Purchases',fontstyle='italic')
plt.title(label='Age VS Number of Purchases',fontsize=18,fontstyle='italic')
plt.show()
```



There is no relationship between age and number of purchases

In [ ]:

```

```

Q5: What is the relationship between education and income?

In [481]:

```
data['Education'].value_counts()
```

Out[481]:

```
Graduation    1127
PhD            486
Master         370
2n Cycle       203
Basic           54
Name: Education, dtype: int64
```

In [482]:

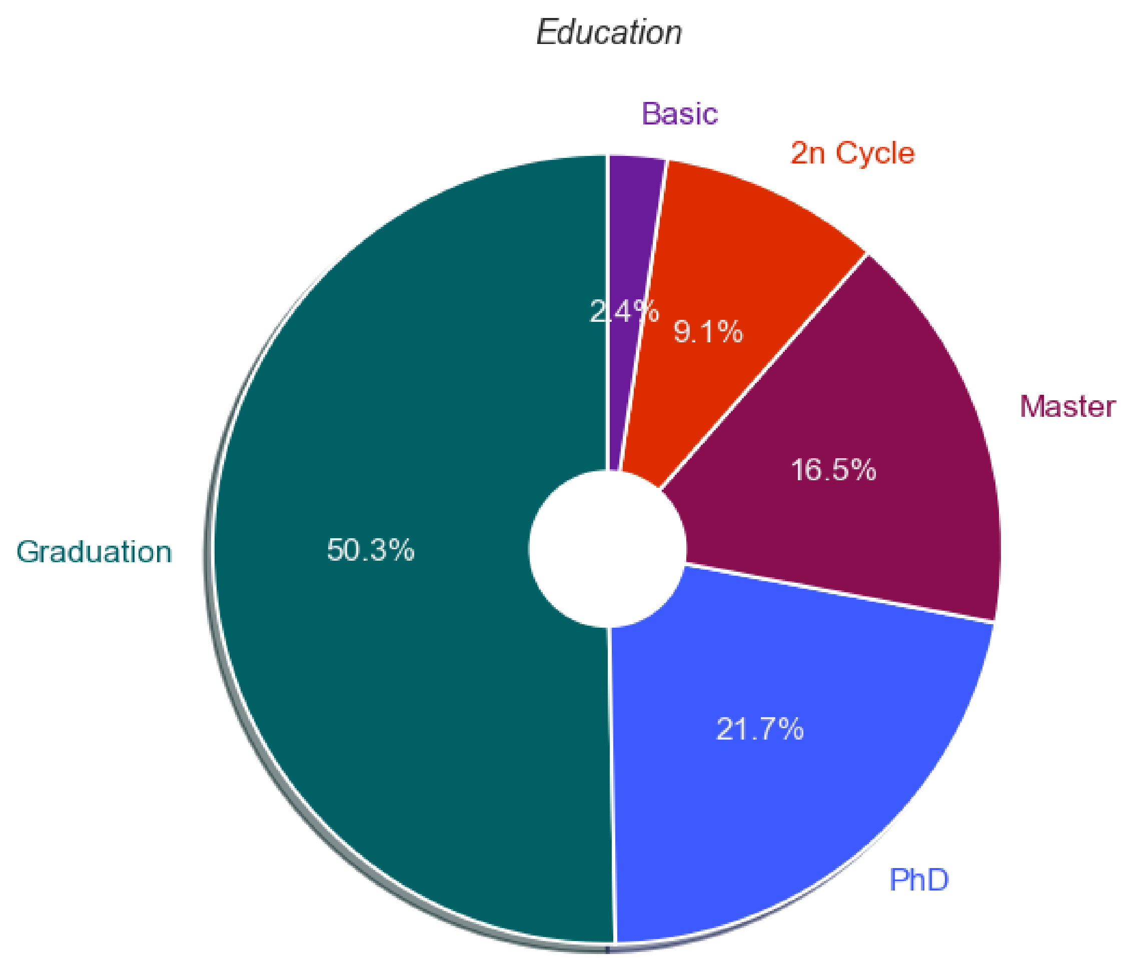
```
color_1=['#006064', '#3d5afe', '#880e4f','#dd2c00','#6a1b9a']
```

In [483]:

```
fig, ax = plt.subplots(figsize=(15, 8))
patches, texts, pcts = ax.pie(
    data['Education'].value_counts(), labels=[*data['Education'].value_counts(
    ,wedgeprops={'linewidth': 2.0, 'edgecolor': 'white'},
    textprops={'size': 'x-large'},
    startangle=90)

for i, patch in enumerate(patches):
    texts[i].set_color(patch.get_facecolor())
plt.setp(pcts, color='#EEEEEE')
plt.setp(texts, fontweight=200)
centre_circle = plt.Circle((0,0),0.20,fc='white')
plt.gcf().gca().add_artist(centre_circle)
plt.tight_layout()
plt.title(label='Education',fontsize=18,fontstyle='italic')

plt.tight_layout()
```



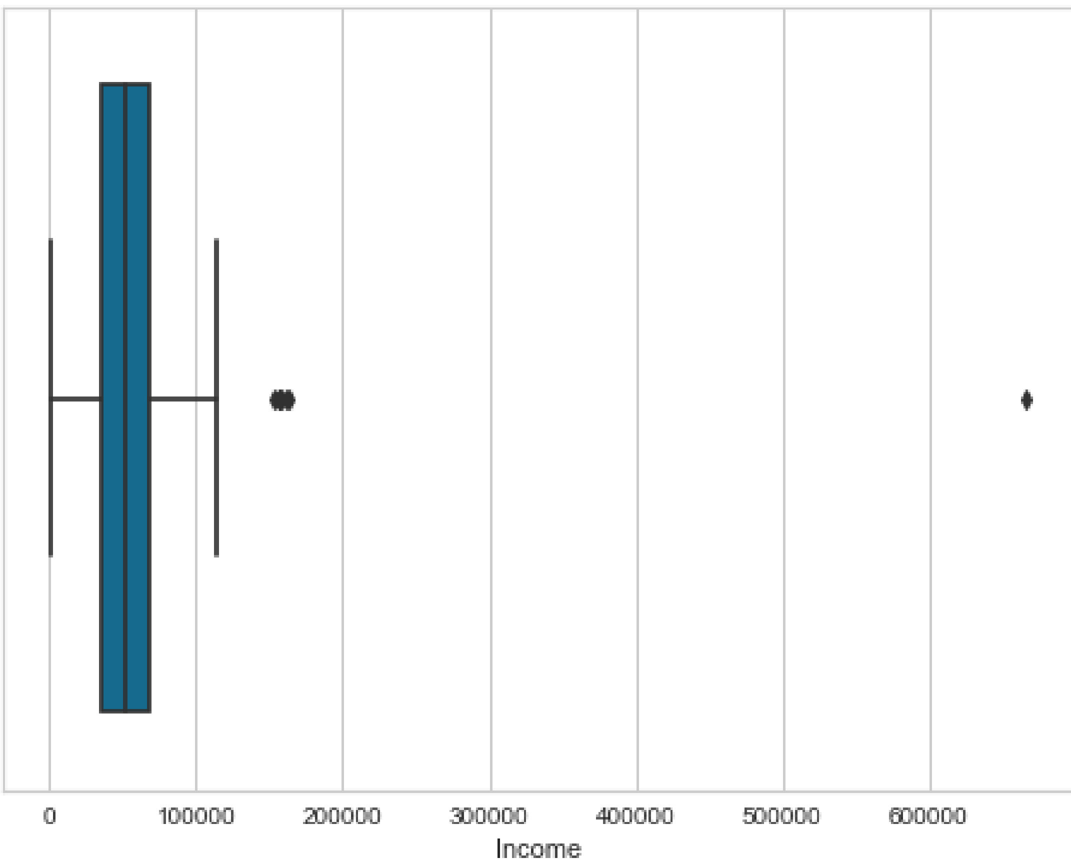
The percentage of clients with university degrees reaches 97.6%

In [484]:

```
plt.figure(figsize=(8,6))
sns.boxplot(data=data,x='Income')
```

Out[484]:

```
<AxesSubplot:xlabel='Income'>
```



In [485]:

```
data=data[data['Income']<200000]
```

In [486]:

```
fig = px.box(data, x="Education", y="Income",)
fig.show()
```

The average income of all clients is very similar except for Basic

In [ ]:

Q6: What is the relationship between income, family size and number of children?

In [487]:

```
fig = px.scatter(data, x="Num_Children", y="Income")
fig.show()
```

All clients have approximately the same income, despite the difference in the number of children, except for those who have 3

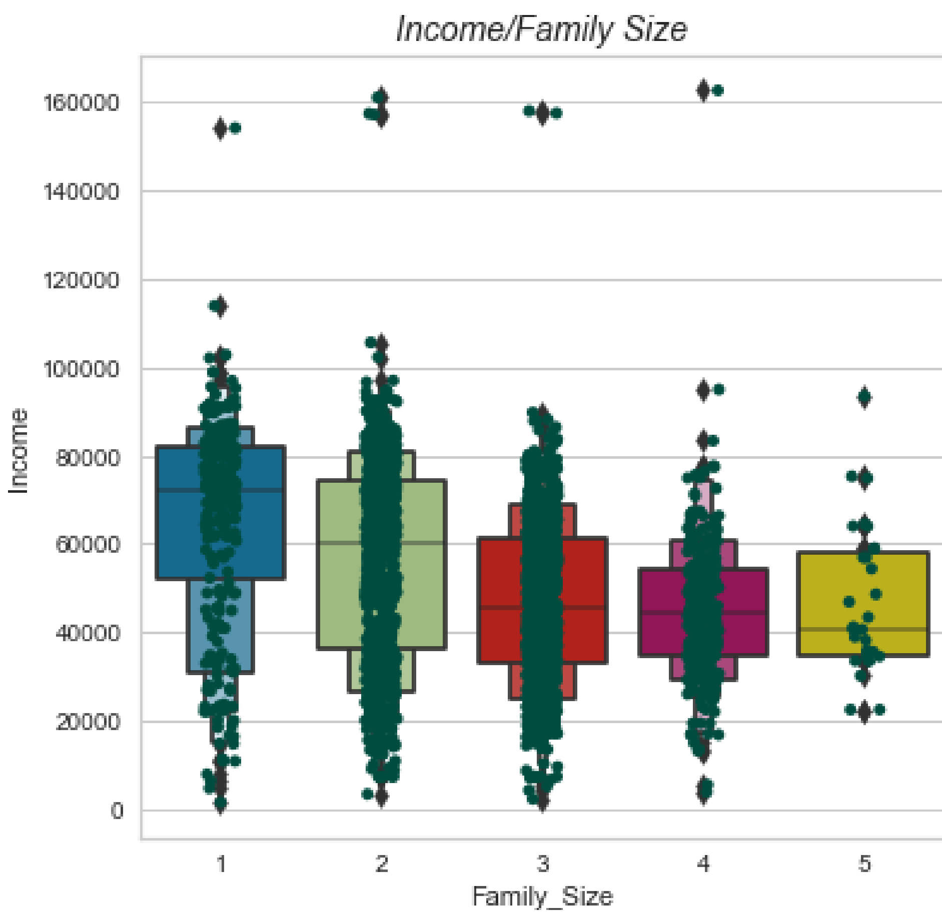
تقريباً جميع العملاء دخلهم متقارب رغم أختلاف عدد الأبناء باستثناء الذين يملكون 3 لأنهم يبداء دخلهم من ال20 ألف وأكثر على خلاف البقية

In [488]:

```
plt.figure(figsize=(6,6))
sns.boxenplot(data=data, x="Family_Size", y="Income")
sns.stripplot(data=data, x="Family_Size", y="Income",color='#004d40')
plt.title(label='Income/Family Size',fontsize=14,fontstyle='italic')
```

Out[488]:

```
Text(0.5, 1.0, 'Income/Family Size')
```



All clients receive roughly the same income, despite the difference in family size, except for those whose share size is 5

In [ ]:

```

```

Q7: What is the relationship between income and the number of purchases?

In [489]:

```
fig = px.scatter(data, x="total_purchases", y="Income")
fig.show()
```

There is a near-linear relationship between income and the number of purchases

العملاء الذين بلغ عدد عمليات الشراء الخاصة بهم ٥٠٠ اغلبهم دخلهم السنوي يقل عن ال٥٠ ألف أما باقي العملاء الذين تزيد عدد عمليات الشراء الخاصة بهم ال ٥٠٠ اغلبهم دخلهم يزيد عن ٥٠ الف وقد يصل إلى ال١٠٠ ألف

أحكيها بالبريزنتيشن

In [ ]:

```

```

Q8: What is the relationship between the number of purchases from the website and the number of website visits?

In [490]:

```
data['NumWebPurchases'].value_counts()
```

Out[490]:

```
2      368
1      348
3      333
4      277
5      219
6      201
7      154
8      102
9       75
0       48
11      44
10      43
23       1
27       1
25       1
Name: NumWebPurchases, dtype: int64
```

In [491]:

```
data['NumWebVisitsMonth'].value_counts()
```

Out[491]:

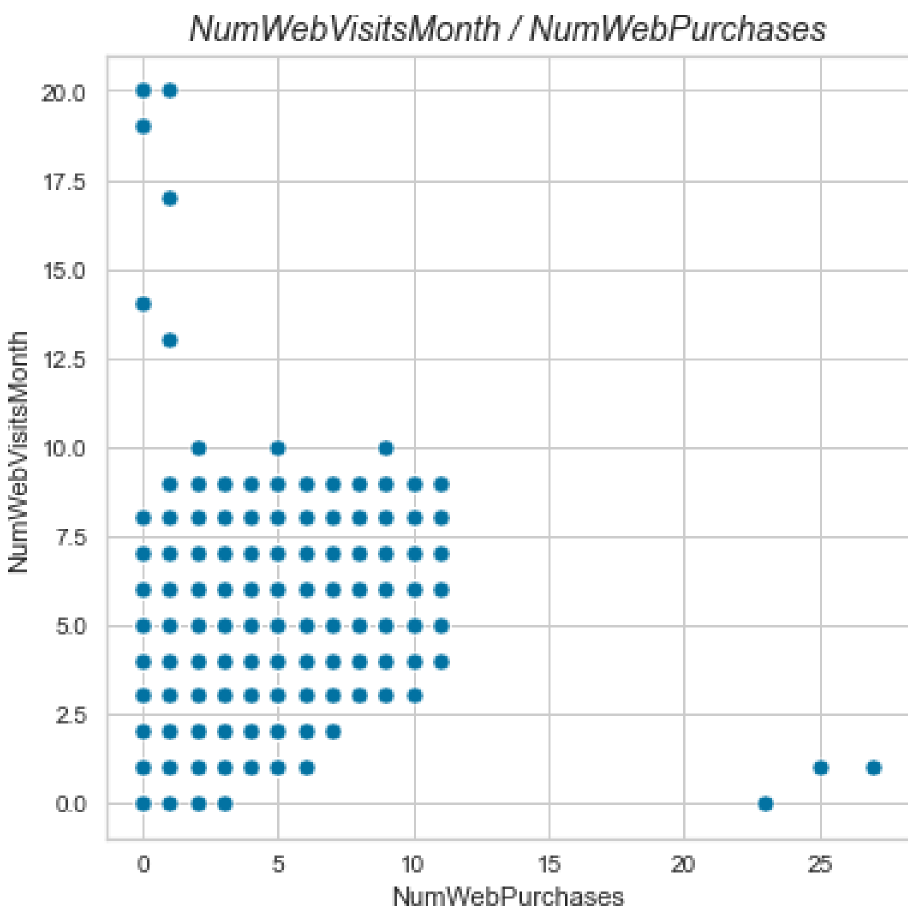
```
7     387
8     340
6     334
5     279
4     217
3     203
2     201
1     150
9      82
0      10
20      3
10      3
14      2
19      2
17      1
13      1
Name: NumWebVisitsMonth, dtype: int64
```

In [492]:

```
plt.figure(figsize=(6,6))
sns.scatterplot(data=data, x="NumWebPurchases", y="NumWebVisitsMonth")
plt.title(label='NumWebVisitsMonth / NumWebPurchases',fontsize=14,fontstyle='i
```

Out[492]:

```
Text(0.5, 1.0, 'NumWebVisitsMonth / NumWebPurchases')
```

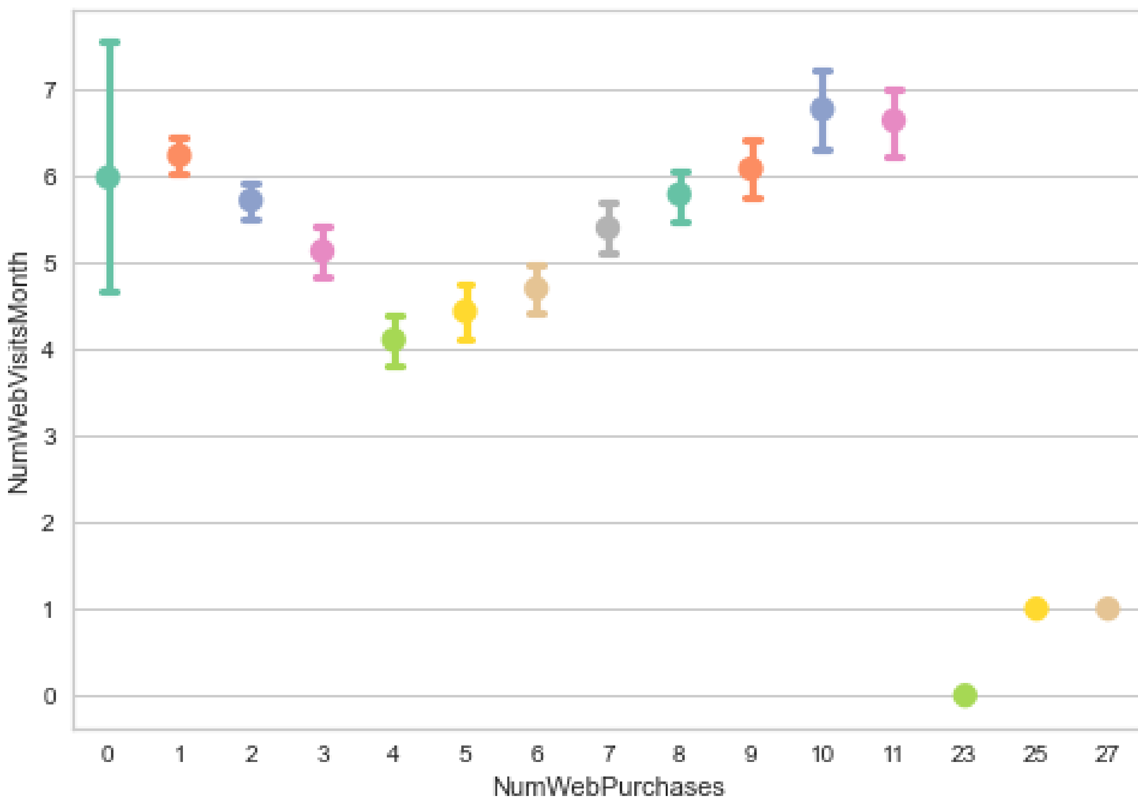


In [493]:

```
sns.pointplot(data=data,x='NumWebPurchases',y='NumWebVisitsMonth',palette='Set
```

Out[493]:

```
<AxesSubplot:xlabel='NumWebPurchases', ylabel='NumWebVisitsMont
h'>
```



In general, there is a linear relationship between the number of visits to the site and the number of purchases from it

طبعاً باستثناء بعض الحالات الغريبة مثل أنه يوجد عميل أشترى من الموقع 23 مرة دون زيارة الموقع

In [ ]:

```

```

Q9: What is the relationship between the number of purchases from a Deal with the number of purchases from the website, the number

of purchases from the catalog, and the number of purchases from the store?

In [494]:

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

Out[494]:

```
1     960
2     493
3     293
4     187
5      94
6      60
0      44
7      39
8      14
9       8
15      7
10      5
11      5
13      3
12      3
Name: NumDealsPurchases, dtype: int64
```

In [495]:

```
data['NumWebPurchases'].value_counts()
```

Out[495]:

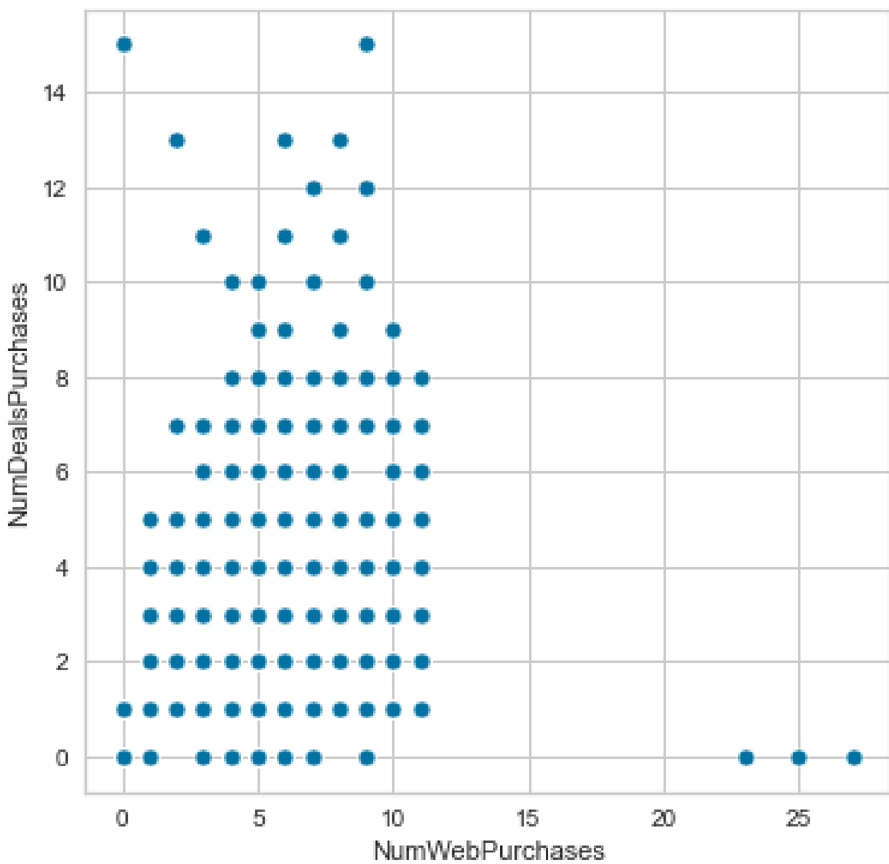
```
2     368
1     348
3     333
4     277
5     219
6     201
7     154
8     102
9      75
0      48
11     44
10     43
23      1
27      1
25      1
Name: NumWebPurchases, dtype: int64
```

In [496]:

```
plt.figure(figsize=(6,6))
sns.scatterplot(data=data,x='NumWebPurchases',y='NumDealsPurchases')
```

Out[496]:

<AxesSubplot:xlabel='NumWebPurchases', ylabel='NumDealsPurchase
s'>



In [497]:

```
data['NumCatalogPurchases'].value_counts()
```

Out[497]:

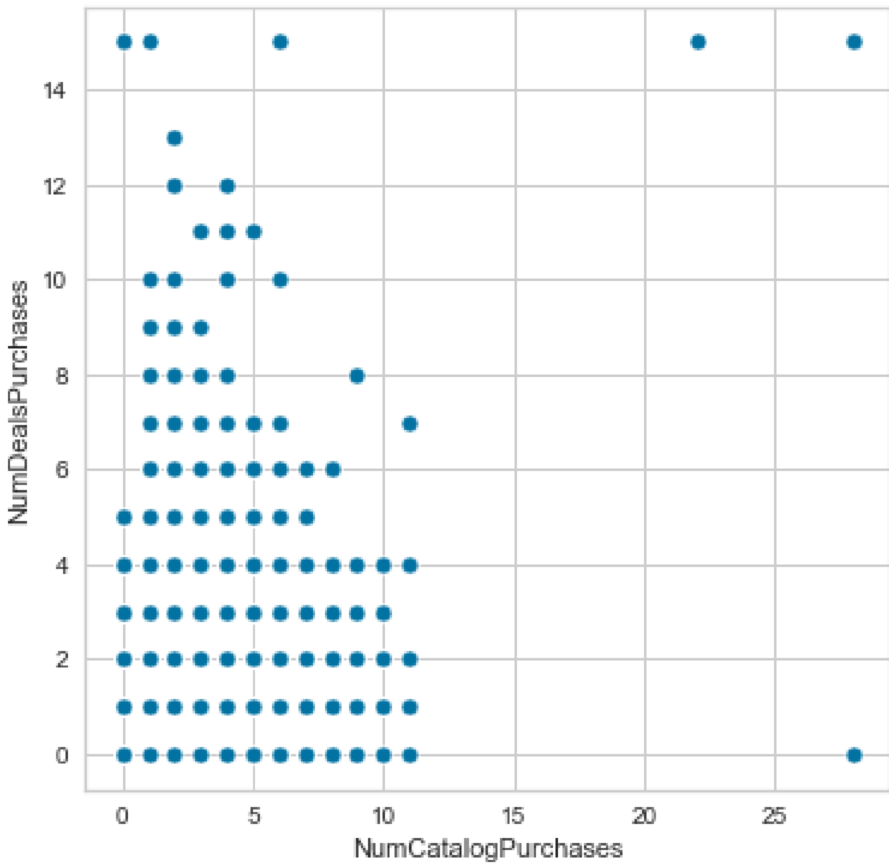
```
0     576
1     491
2     274
3     182
4     181
5     137
6     128
7      79
8      55
10     47
9      42
11     19
28      3
22      1
Name: NumCatalogPurchases, dtype: int64
```

In [498]:

```
plt.figure(figsize=(6,6))
sns.scatterplot(data=data,x='NumCatalogPurchases',y='NumDealsPurchases')
```

Out[498]:

<AxesSubplot:xlabel='NumCatalogPurchases', ylabel='NumDealsPurc
hases'>



In [499]:

```
data['NumStorePurchases'].value_counts()
```

Out[499]:

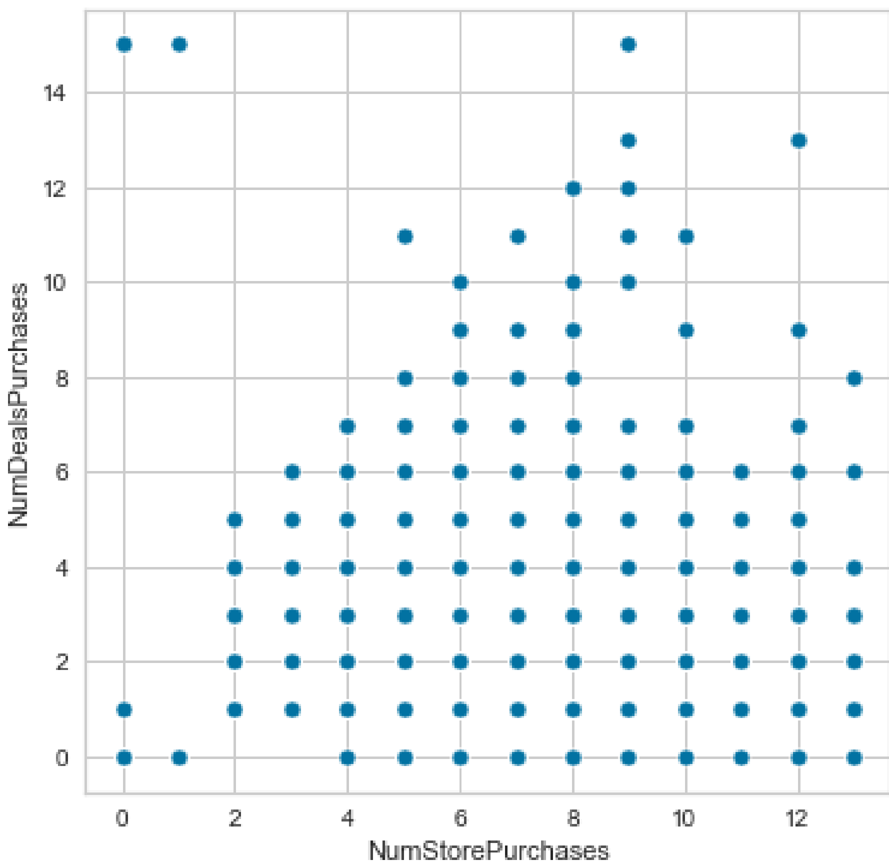
```
3     483
4     319
2     220
5     211
6     177
8     147
7     141
10    124
9     106
12    104
13     83
11     80
0      14
1       6
Name: NumStorePurchases, dtype: int64
```

In [500]:

```
plt.figure(figsize=(6,6))
sns.scatterplot(data=data,x='NumStorePurchases',y='NumDealsPurchases')
```

Out[500]:

```
<AxesSubplot:xlabel='NumStorePurchases', ylabel='NumDealsPurcha
ses'>
```



In [ ]:

Q10: What is the relationship between the number of purchases from a Deal with accepted cmp 1 ,accepted cmp 2,accepted cmp 3 ,accepted cmp 4 ,accepted cmp 5 and Response?

In [501]:

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

Out[501]:

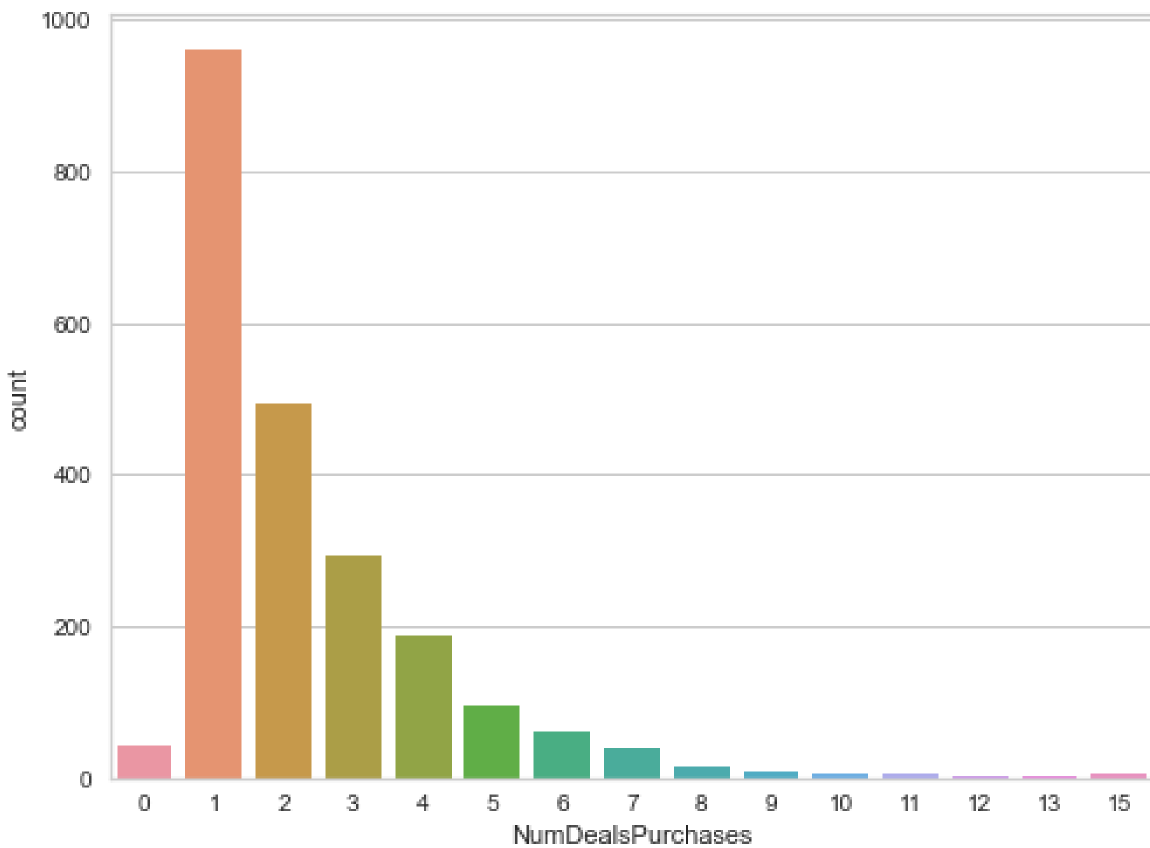
```
1     960
2     493
3     293
4     187
5      94
6      60
0      44
7      39
8      14
9       8
15      7
10      5
11      5
13      3
12      3
Name: NumDealsPurchases, dtype: int64
```

In [502]:

```
plt.figure(figsize=(8,6))
sns.countplot(data=data,x='NumDealsPurchases')
```

Out[502]:

```
<AxesSubplot:xlabel='NumDealsPurchases', ylabel='count'>
```



Almost all customers have made a purchase at least once

In [503]:

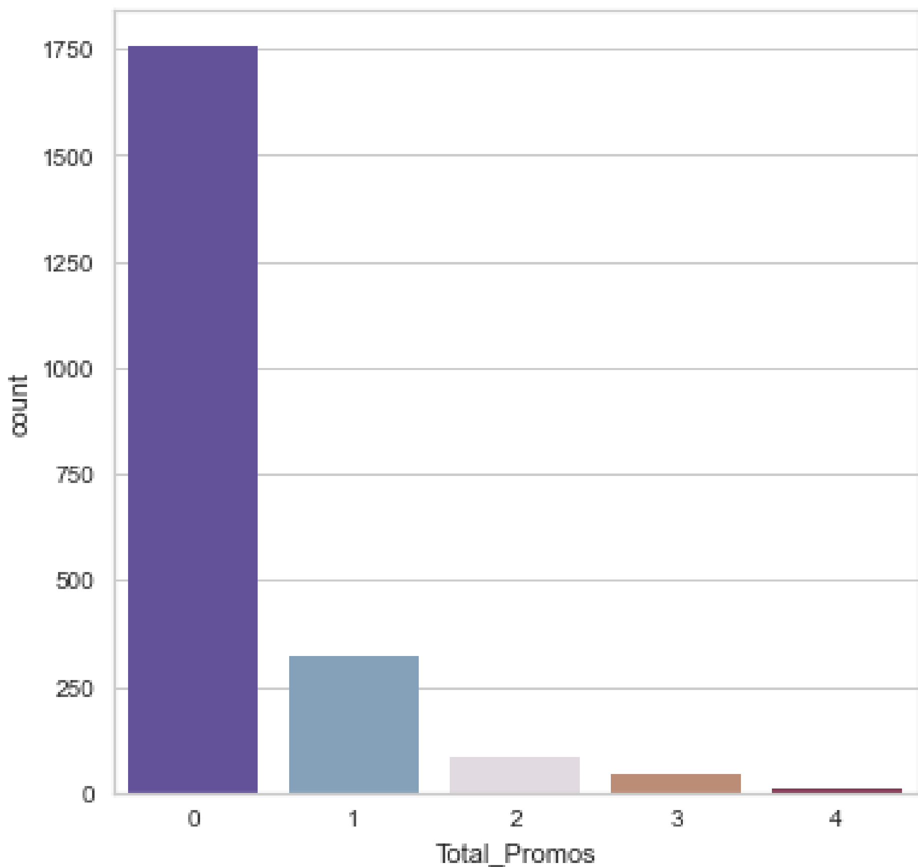
```
# Create a feature that collects all offers
data["Total_Promos"] = data["AcceptedCmp1"]+ data["AcceptedCmp2"]+ data["Accep
```

In [504]:

```
plt.figure(figsize=(6,6))
sns.countplot(data=data,x='Total_Promos',palette='twilight_shifted')
```

Out[504]:

```
<AxesSubplot:xlabel='Total_Promos', ylabel='count'>
```



The number of customers who did not accept the offers is very large, up to 80%

In [ ]:

Q11: What is the relationship between the complaint and Date of customer's enrollment?

In [505]:

```
data['Complain'].value_counts()
```

Out[505]:

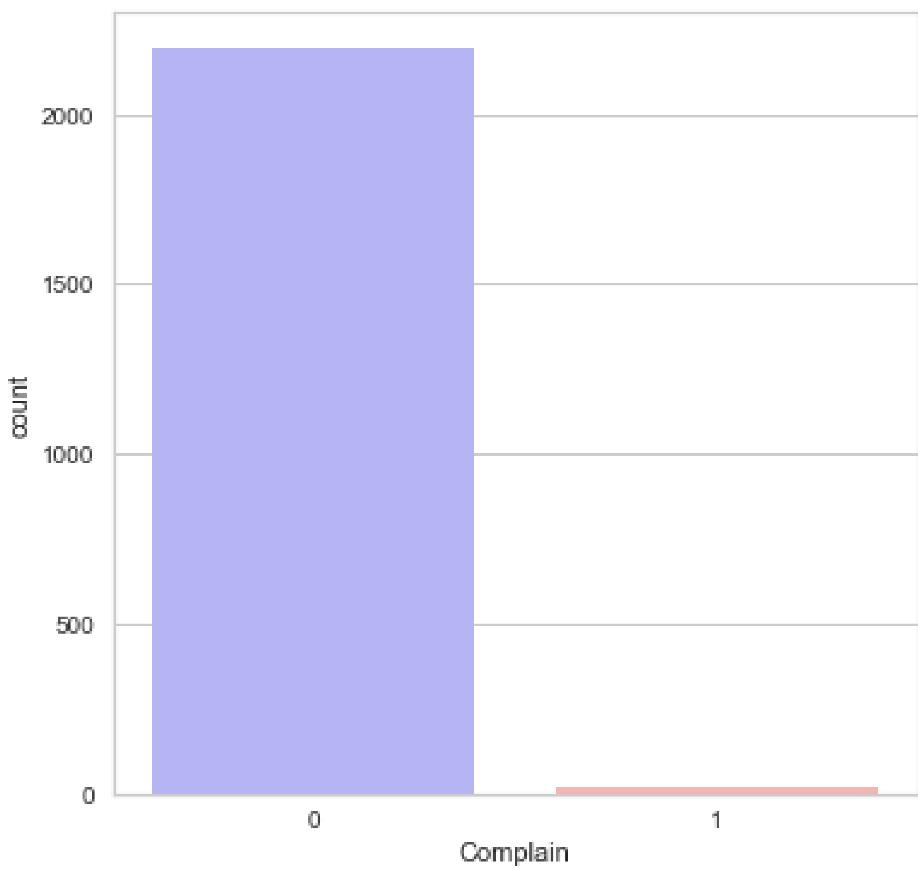
```
0    2194
1      21
Name: Complain, dtype: int64
```

In [506]:

```
plt.figure(figsize=(6,6))
sns.countplot(data=data,x='Complain',palette='bwr')
```

Out[506]:

```
<AxesSubplot:xlabel='Complain', ylabel='count'>
```



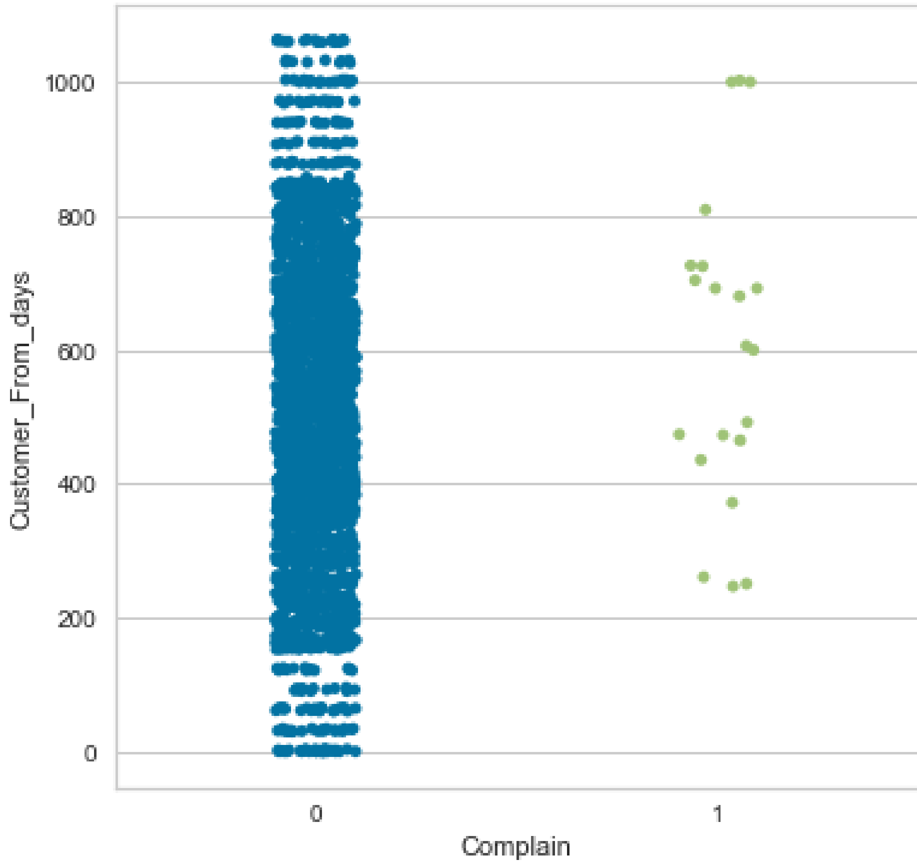
Very few customers who complained

In [507]:

```
plt.figure(figsize=(6,6))
sns.stripplot(data=data,x='Complain',y='Customer_From_days')
```

Out[507]:

```
<AxesSubplot:xlabel='Complain', ylabel='Customer_From_days'>
```



All customers who filed a complaint as if they were with the company more than 200 days ago

In [ ]:

```

```

Data Preprocessing

In [508]:

```
data.isnull().sum()
```

Out[508]:

```
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
Customer_From_days     0
Age                    0
Living_With            0
Num_Children           0
Family_Size            0
total_purchases        0
purchase_quantity      1
Total_Promos           0
dtype: int64
```

In [511]:

```
fill_tobed=data['purchase_quantity'].dropna()
data['purchase_quantity']=data['purchase_quantity'].fillna(pd.Series(np.random
```

In [512]:

```
data.isnull().sum()
```

Out[512]:

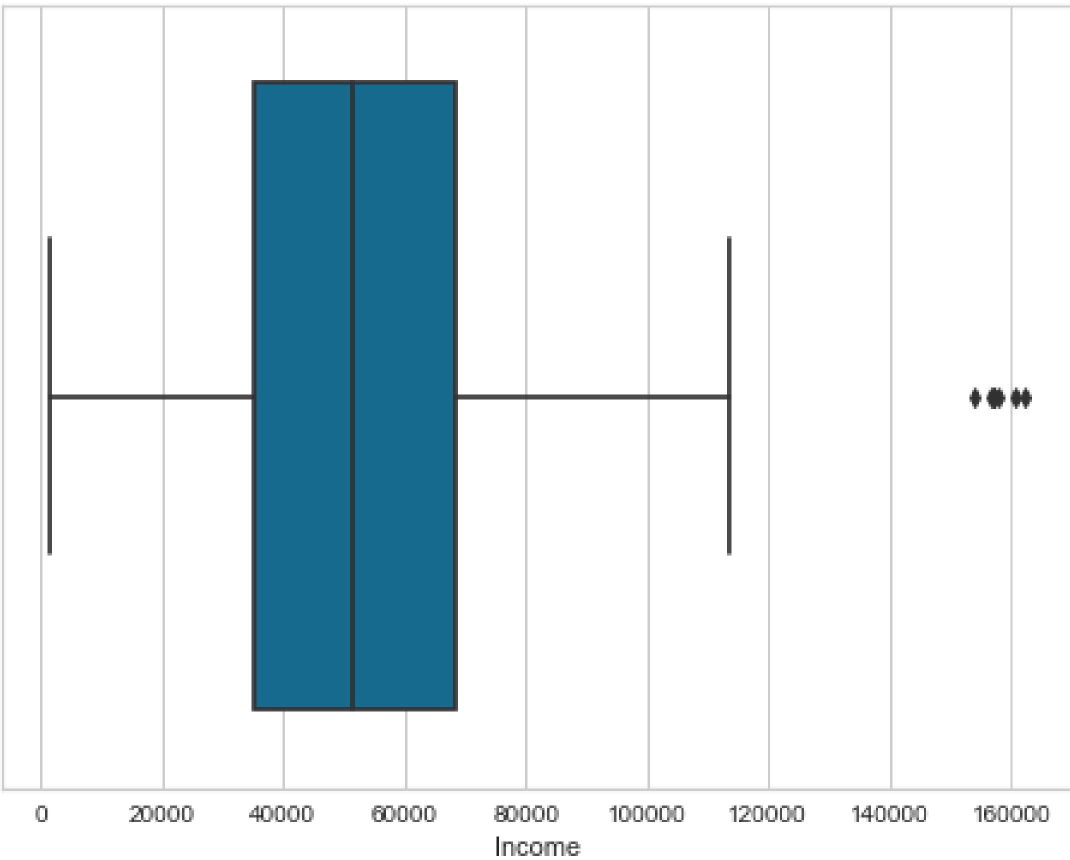
```
ID                     0
Year_Birth             0
Education              0
Marital_Status         0
Income                 0
Kidhome                0
Teenhome               0
Dt_Customer            0
Recency                0
MntWines               0
MntFruits              0
MntMeatProducts        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
Customer_From_days     0
Age                    0
Living_With            0
Num_Children           0
Family_Size            0
total_purchases        0
purchase_quantity      0
Total_Promos           0
dtype: int64
```

In [514]:

```
plt.figure(figsize=(8,6))
sns.boxplot(data=data,x='Income')
```

Out[514]:

```
<AxesSubplot:xlabel='Income'>
```



In [515]:

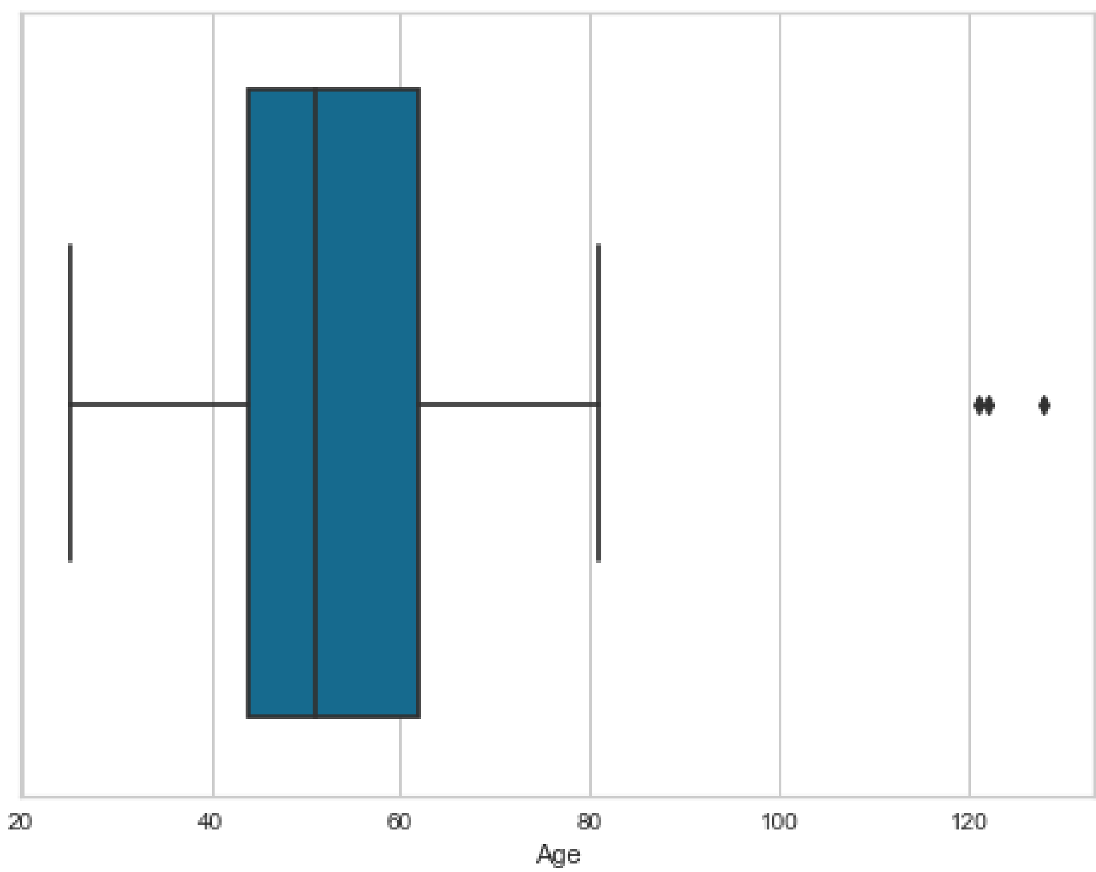
```
# Remove outliers
data=data[data['Income']<150000]
```

In [516]:

```
plt.figure(figsize=(8,6))
sns.boxplot(data=data,x='Age')
```

Out[516]:

<AxesSubplot:xlabel='Age'>



In [517]:

```
# Remove outliers
data=data[data['Age']<100]
```

In [519]:

```
data.columns
```

Out[519]:

```
Index(['ID', 'Year_Birth', 'Education', 'Marital_Status', 'Inco
me', 'Kidhome',
       'Teenhome', 'Dt_Customer', 'Recency', 'MntWines', 'MntFr
uits',
       'MntMeatProducts', 'MntFishProducts', 'MntSweetProduct
s',
       'MntGoldProds', 'NumDealsPurchases', 'NumWebPurchases',
       'NumCatalogPurchases', 'NumStorePurchases', 'NumWebVisit
sMonth',
       'AcceptedCmp3', 'AcceptedCmp4', 'AcceptedCmp5', 'Accepte
dCmp1',
       'AcceptedCmp2', 'Complain', 'Z_CostContact', 'Z_Revenu
e', 'Response',
       'Customer_From_days', 'Age', 'Living_With', 'Num_Childre
n',
       'Family_Size', 'total_purchases', 'purchase_quantity',
'Total_Promos'],
      dtype='object')
```

In [520]:

```
# Drop columns we don't need them
data=data[['Education', 'Marital_Status', 'Income','Recency', 'MntWines', 'Mnt
       'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts', 'Kidhome',
       'Teenhome','Num_Children','MntGoldProds', 'NumDealsPurchases', 'NumWebP
       'NumCatalogPurchases', 'NumStorePurchases', 'NumWebVisitsMonth',
       'Customer_From_days', 'Age', 'Family_Size', 'total_purchases','purchase
```

In [521]:

```
data
```

Out[521]:

	Education	Marital_Status	Income	Recency	MntWines	MntFruits	MntM
0	Graduation	Single	58138.0	58	635	88	
1	Graduation	Single	46344.0	38	11	1	
2	Graduation	Together	71613.0	26	426	49	
3	Graduation	Together	26646.0	26	11	4	
4	PhD	Married	58293.0	94	173	43	
...	...	...	...	...	...	...	
2235	Graduation	Married	61223.0	46	709	43	
2236	PhD	Together	64014.0	56	406	0	
2237	Graduation	Divorced	56981.0	91	908	48	
2238	Master	Together	69245.0	8	428	30	
2239	PhD	Married	52869.0	40	84	3	

2205 rows × 24 columns

In [522]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2205 entries, 0 to 2239
Data columns (total 24 columns):
 #   Column               Non-Null Count  Dtype   
---  ------               --------------  -----   
 0   Education            2205 non-null   object  
 1   Marital_Status       2205 non-null   object  
 2   Income               2205 non-null   float64 
 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   Kidhome              2205 non-null   int64   
 10  Teenhome             2205 non-null   int64   
 11  Num_Children         2205 non-null   int64   
 12  MntGoldProds         2205 non-null   int64   
 13  NumDealsPurchases    2205 non-null   int64   
 14  NumWebPurchases      2205 non-null   int64   
 15  NumCatalogPurchases  2205 non-null   int64   
 16  NumStorePurchases    2205 non-null   int64   
 17  NumWebVisitsMonth    2205 non-null   int64   
 18  Customer_From_days   2205 non-null   int64   
 19  Age                  2205 non-null   int64   
 20  Family_Size          2205 non-null   int64   
 21  total_purchases      2205 non-null   int64   
 22  purchase_quantity    2205 non-null   category
 23  Total_Promos         2205 non-null   int64   
dtypes: category(1), float64(1), int64(20), object(2)
memory usage: 416.0+ KB
```

In [523]:

```
s = (data.dtypes == 'object')
n = (data.dtypes == 'category')
object_cols = list(s[s].index)
category_col = list(n[n].index)
print("Categorical variables in the dataset:", object_cols, category_col)
```

```
Categorical variables in the dataset: ['Education', 'Marital_St
atus'] ['purchase_quantity']
```

In [524]:

```
LE=LabelEncoder()
for i in object_cols:
    object_le=LE
    data[i]=object_le.fit_transform(data[i])
    print(object_le.classes_)
for i in category_col:
    category_le=LE
    data[i]=category_le.fit_transform(data[i])
print("All features are now numerical")
```

```
['2n Cycle' 'Basic' 'Graduation' 'Master' 'PhD']
['Divorced' 'Married' 'Single' 'Together' 'Widow']
All features are now numerical
```

In [ ]:

```

```

In [525]:

```
#Creating a copy of data
ds = data.copy()
#Scaling
scaled=StandardScaler()
scaled.fit(ds)
scaled_ds = pd.DataFrame(scaled.transform(ds),columns= ds.columns )
print("All features are now scaled")
```

```
All features are now scaled
```

In [ ]:

```

```

In [526]:

```
ds.isnull().sum()
```

Out[526]:

```
Education              0
Marital_Status         0
Income                 0
Recency                0
MntWines               0
MntFruits              0
MntMeatProducts        0
MntFishProducts        0
MntSweetProducts       0
Kidhome                0
Teenhome               0
Num_Children           0
MntGoldProds           0
NumDealsPurchases      0
NumWebPurchases        0
NumCatalogPurchases    0
NumStorePurchases      0
NumWebVisitsMonth      0
Customer_From_days     0
Age                    0
Family_Size            0
total_purchases        0
purchase_quantity      0
Total_Promos           0
dtype: int64
```

In [527]:

```
ds
```

Out[527]:

	Education	Marital_Status	Income	Recency	MntWines	MntFruits	MntM
0	2	2	58138.0	58	635	88	
1	2	2	46344.0	38	11	1	
2	2	3	71613.0	26	426	49	
3	2	3	26646.0	26	11	4	
4	4	1	58293.0	94	173	43	
...	...	...	...	...	...	...	
2235	2	1	61223.0	46	709	43	
2236	4	3	64014.0	56	406	0	
2237	2	0	56981.0	91	908	48	
2238	3	3	69245.0	8	428	30	
2239	4	1	52869.0	40	84	3	

2205 rows × 24 columns

In [528]:

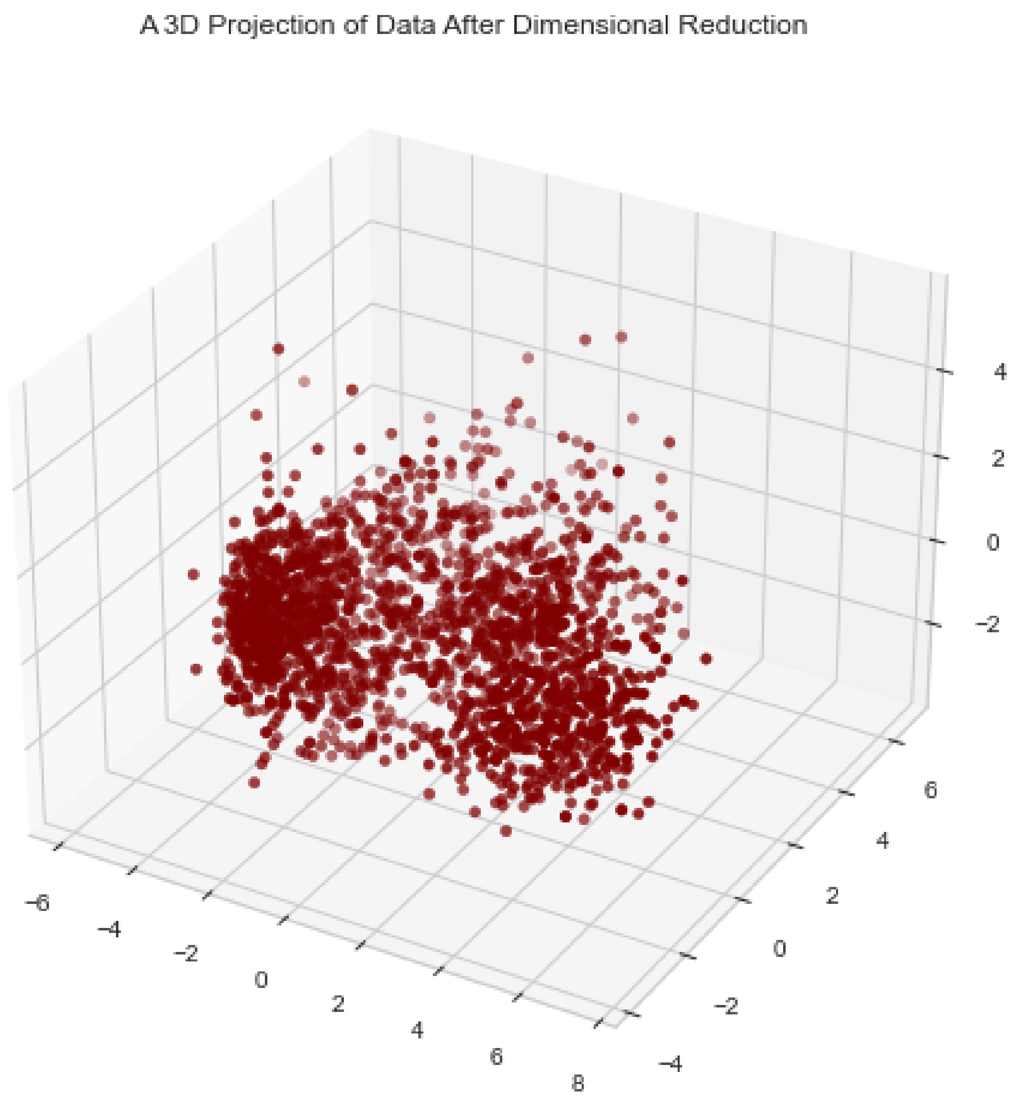
```
# Using PCA to reduce the dimensions of the data to 3 dimensions
pca = PCA(n_components=3)
pca.fit(scaled_ds)
PCA_ds = pd.DataFrame(pca.transform(scaled_ds), columns=(["col1","col2", "col3
PCA_ds.describe().T
```

Out[528]:

	count	mean	std	min	25%	50%	75%
col1	2205.0	-1.691768e-17	2.992294	-5.853830	-2.719203	-0.885906	2.618322
col2	2205.0	1.208406e-17	1.648515	-3.723516	-1.362039	-0.170453	1.167380
col3	2205.0	2.723949e-17	1.247571	-3.408792	-0.860292	0.016014	0.800394

In [529]:

```
x =PCA_ds["col1"]
y =PCA_ds["col2"]
z =PCA_ds["col3"]
fig = plt.figure(figsize=(10,8))
ax = fig.add_subplot(111, projection="3d")
ax.scatter(x,y,z, c="maroon", marker="o" )
ax.set_title("A 3D Projection of Data After Dimensional Reduction")
plt.show()
```



In [ ]:

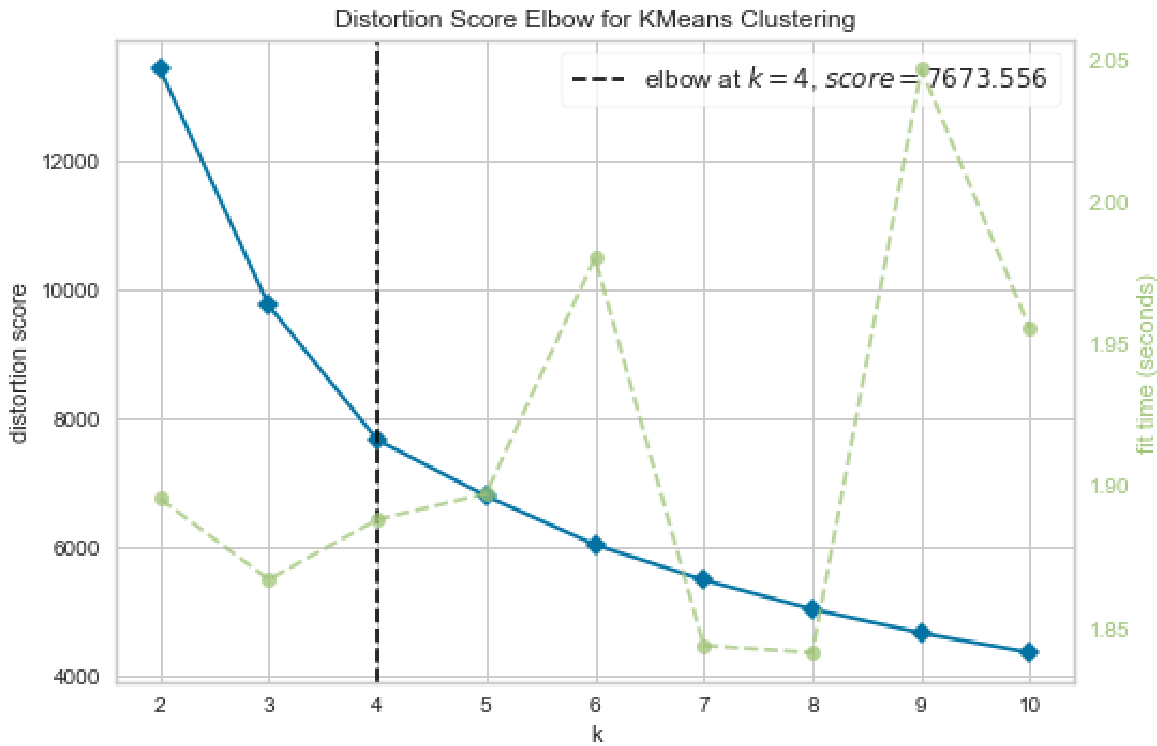
```

```

CLUSTERING

In [530]:

```
# Using the elbow method to find the optimal number of clusters'
Elbow_M = KElbowVisualizer(KMeans(), k=10)
Elbow_M.fit(PCA_ds)
Elbow_M.show()
```



Out[530]:

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

In [531]:

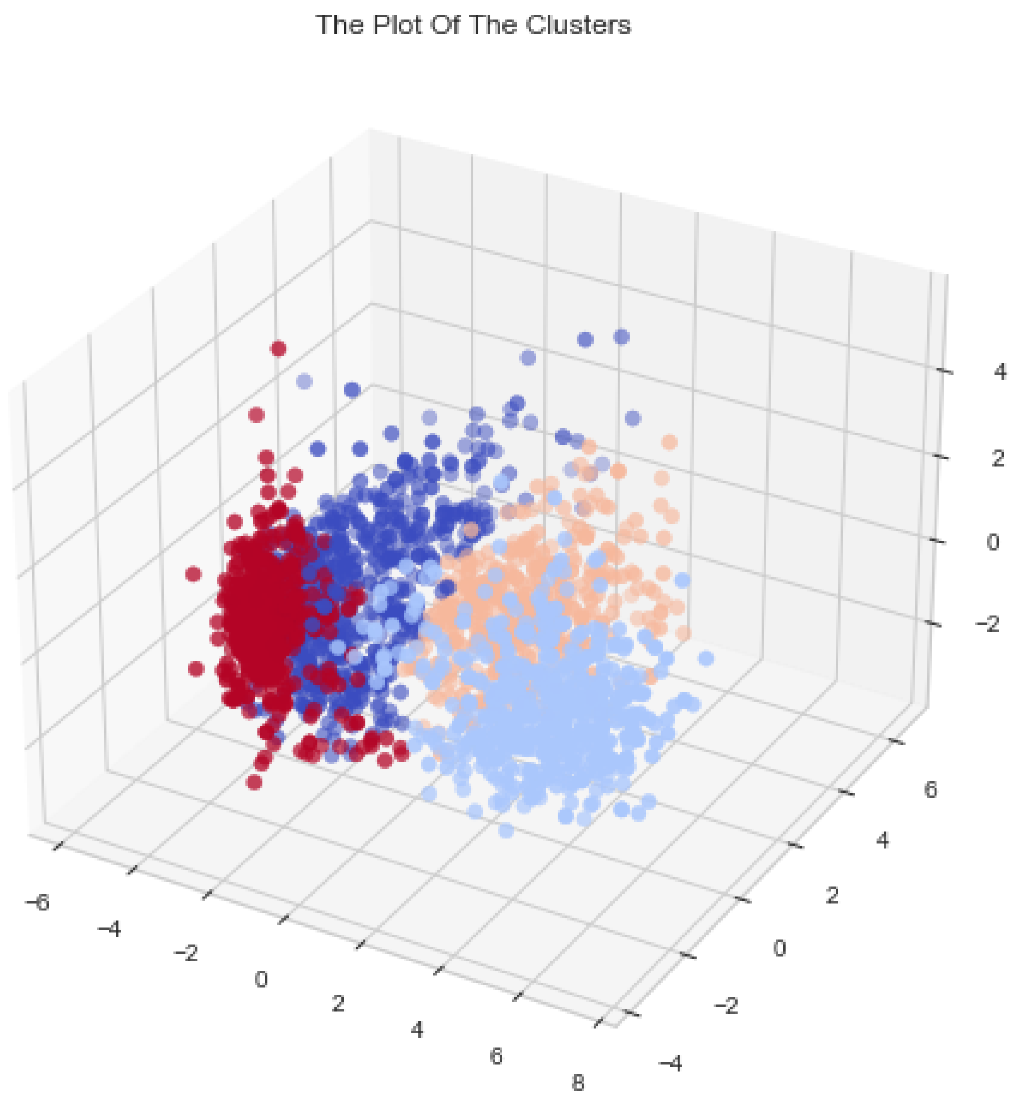
```
PCA_KM=PCA_ds.copy()
PCA_AC=PCA_ds.copy()
data_KM=data.copy()
data_AC=data.copy()
```

In [532]:

```
#Initiating the Agglomerative Clustering model
AC = AgglomerativeClustering(n_clusters=4)
# fit model and predict clusters
yhat_AC = AC.fit_predict(PCA_AC)
PCA_AC["Clusters"] = yhat_AC
#Adding the Clusters feature to the orignal dataframe.
data_AC["Clusters"]= yhat_AC
```

In [533]:

```
#Plotting the clusters
fig = plt.figure(figsize=(10,8))
ax = plt.subplot(111, projection='3d', label="bla")
ax.scatter(x, y, z, s=40, c=PCA_AC["Clusters"], marker='o', cmap = 'coolwarm'
ax.set_title("The Plot Of The Clusters")
plt.show()
```



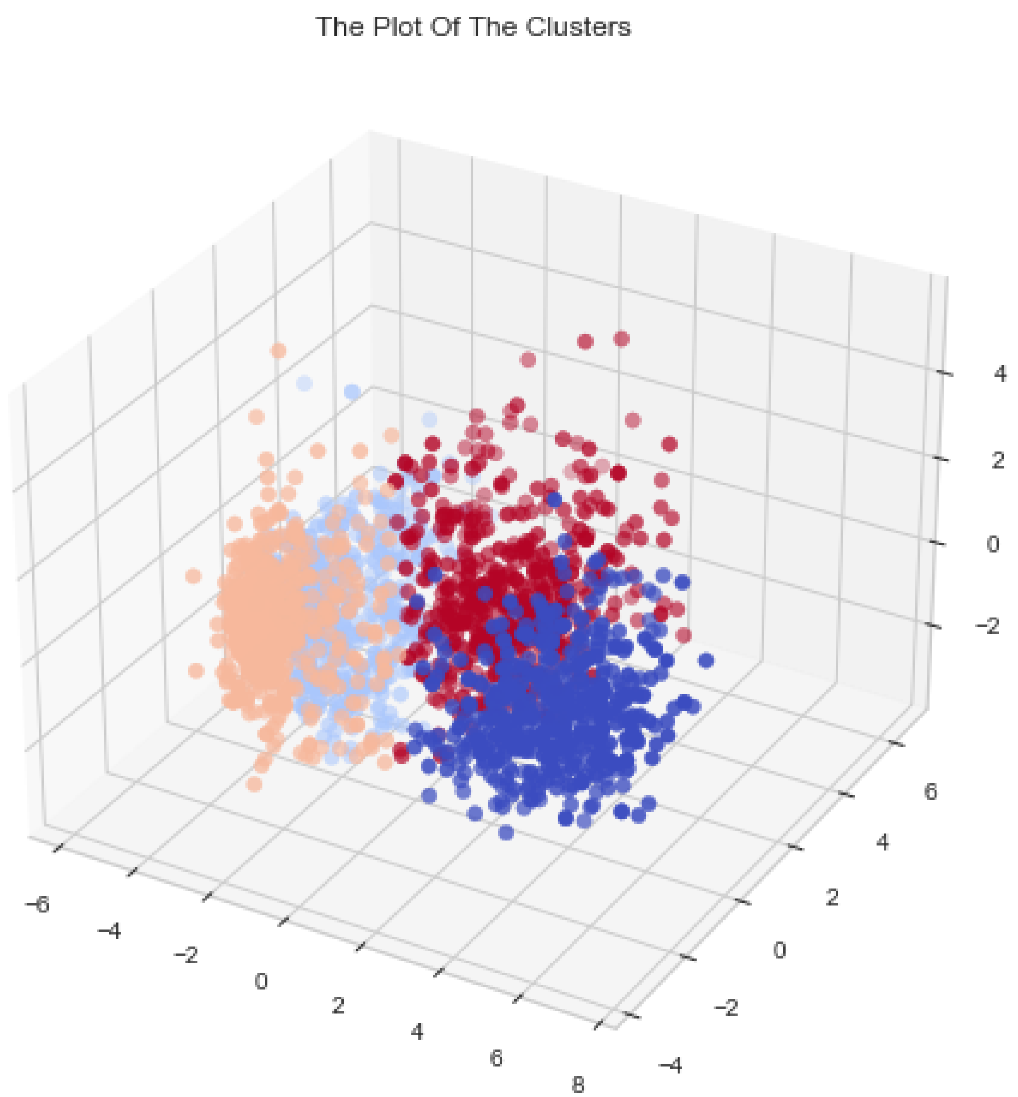
In [534]:

```
#Initiating the K-Means model
KM = KMeans(n_clusters=4)
# fit model and predict clusters
yhat_KM = KM.fit_predict(PCA_KM)
PCA_KM["Clusters"] = yhat_KM
#Adding the Clusters feature to the orignal dataframe.
data_KM["Clusters"]= yhat_KM
```

In [ ]:

In [535]:

```
#Plotting the clusters
fig = plt.figure(figsize=(10,8))
ax = plt.subplot(111, projection='3d', label="bla")
ax.scatter(x, y, z, s=40, c=PCA_KM["Clusters"], marker='o', cmap = 'coolwarm'
ax.set_title("The Plot Of The Clusters")
plt.show()
```



In [ ]:

```

```

Evaluation Models.

In [ ]:

```

```

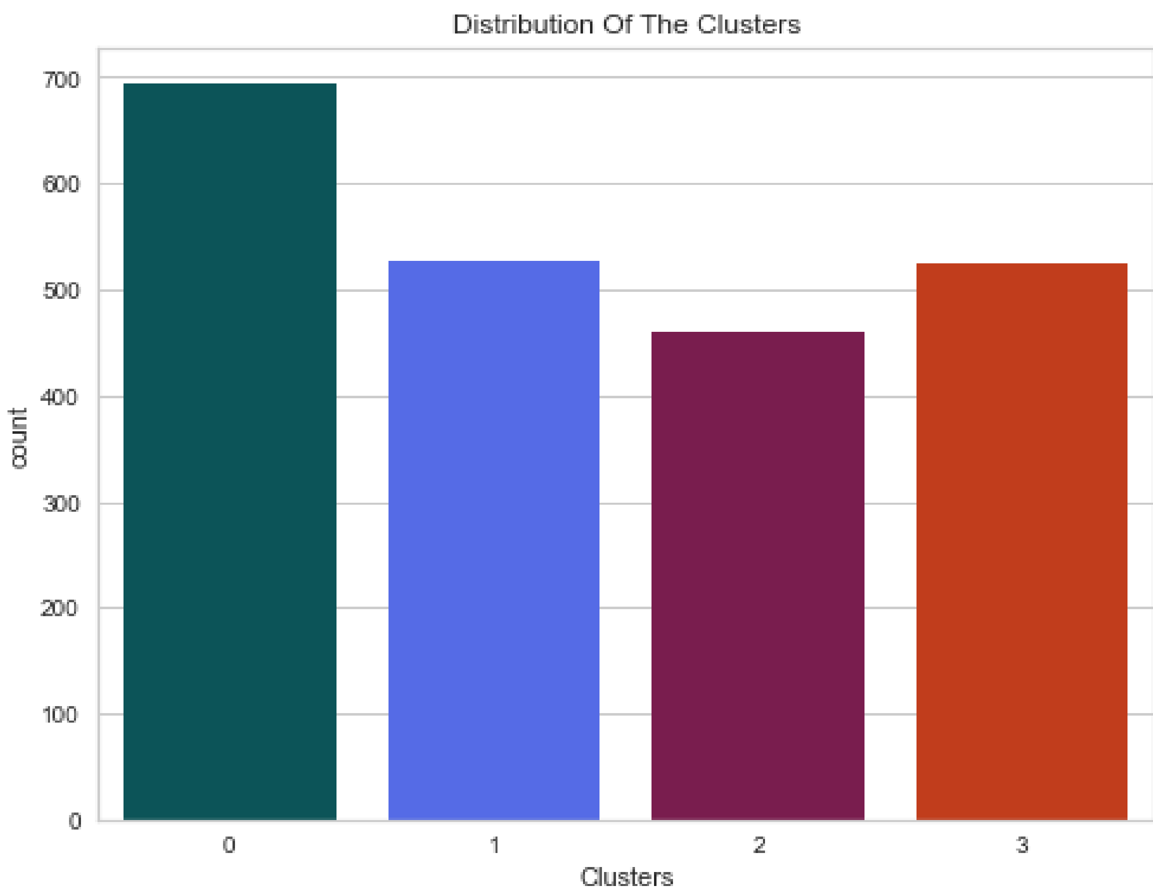
AgglomerativeClustering

In [536]:

```
color_2=['#006064', '#3d5afe', '#880e4f','#dd2c00']
```

In [537]:

```
#Plotting countplot of clusters
plt.figure(figsize=(8,6))
pl = sns.countplot(x=data_AC["Clusters"], palette=color_2 )
pl.set_title("Distribution Of The Clusters")
plt.show()
```

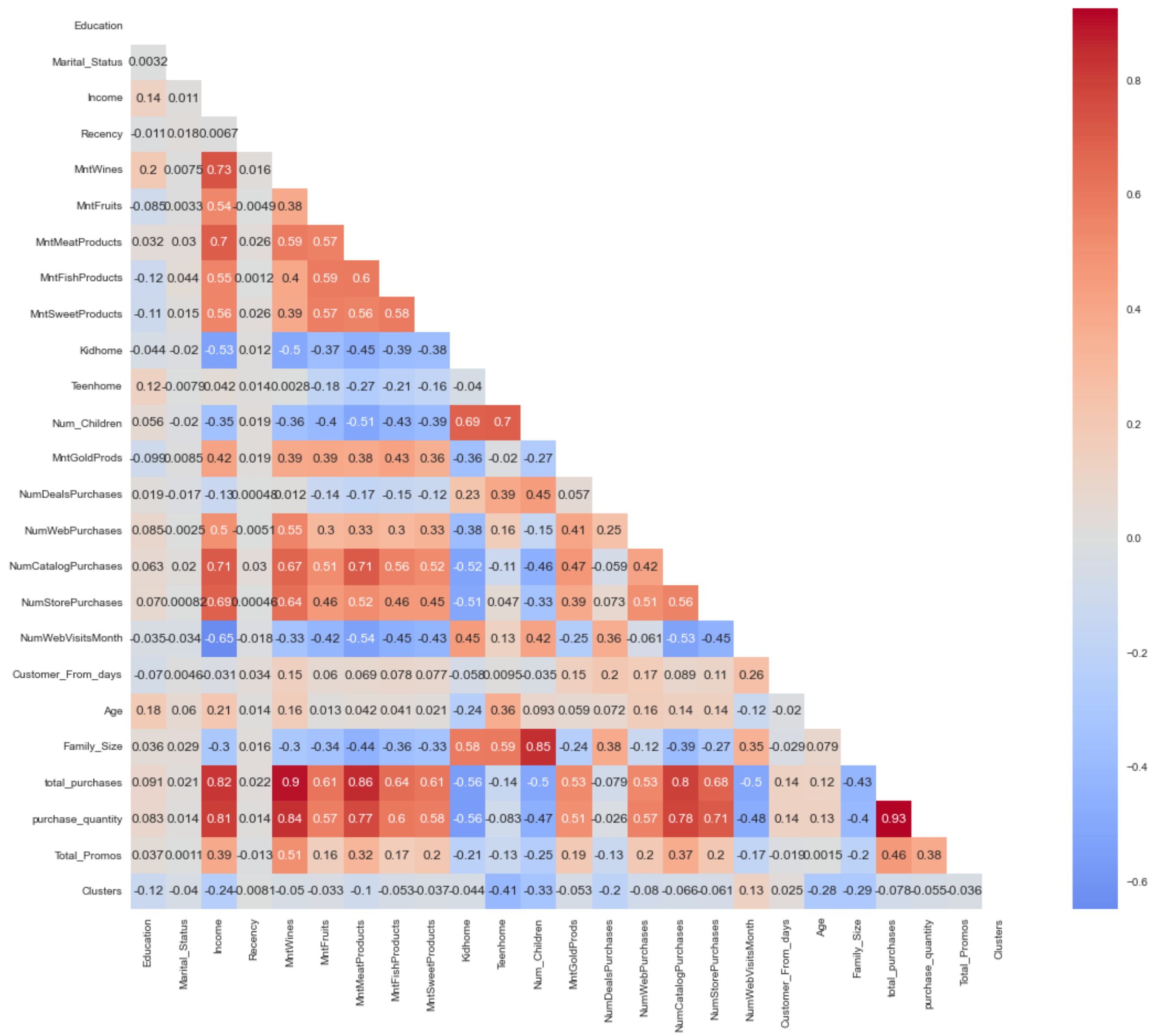


In [538]:

```
corrmat= data_AC.corr()
mask = np.triu(np.ones_like(corrmat, dtype=bool))
plt.figure(figsize=(18,15))
sns.heatmap(corrmat,annot=True, cmap='coolwarm', center=0,mask=mask)
```

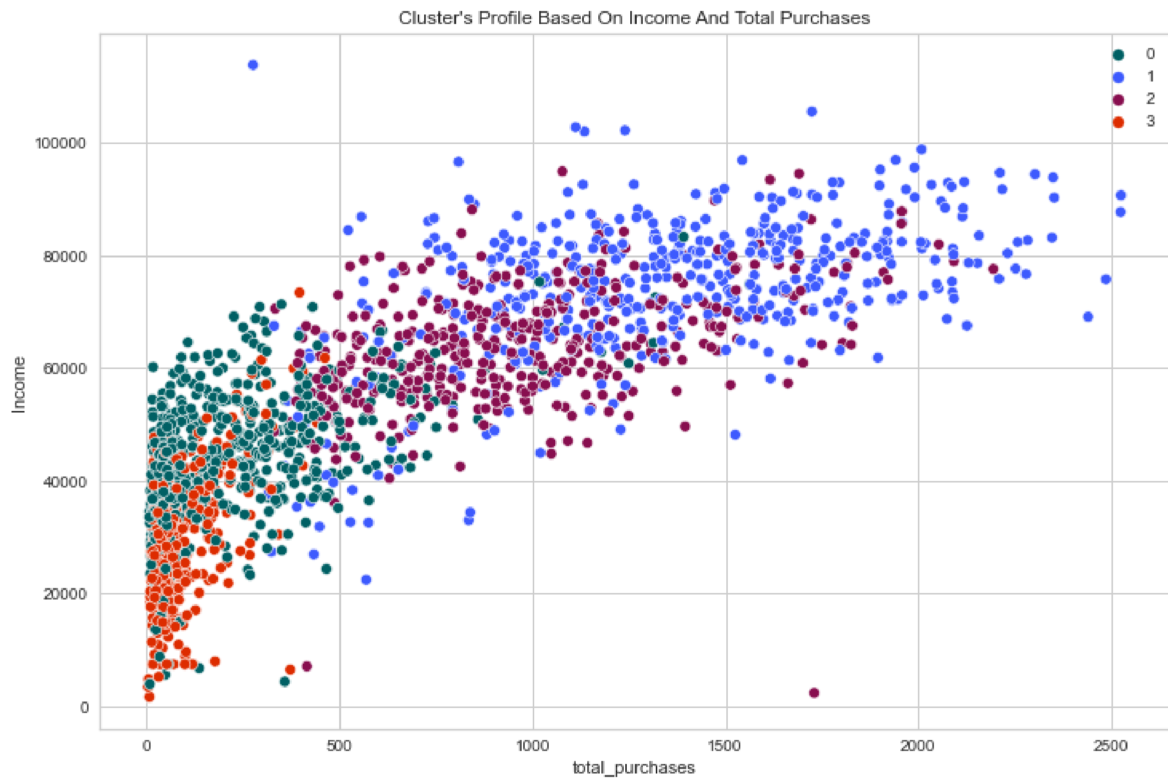
Out[538]:

<AxesSubplot:>



In [539]:

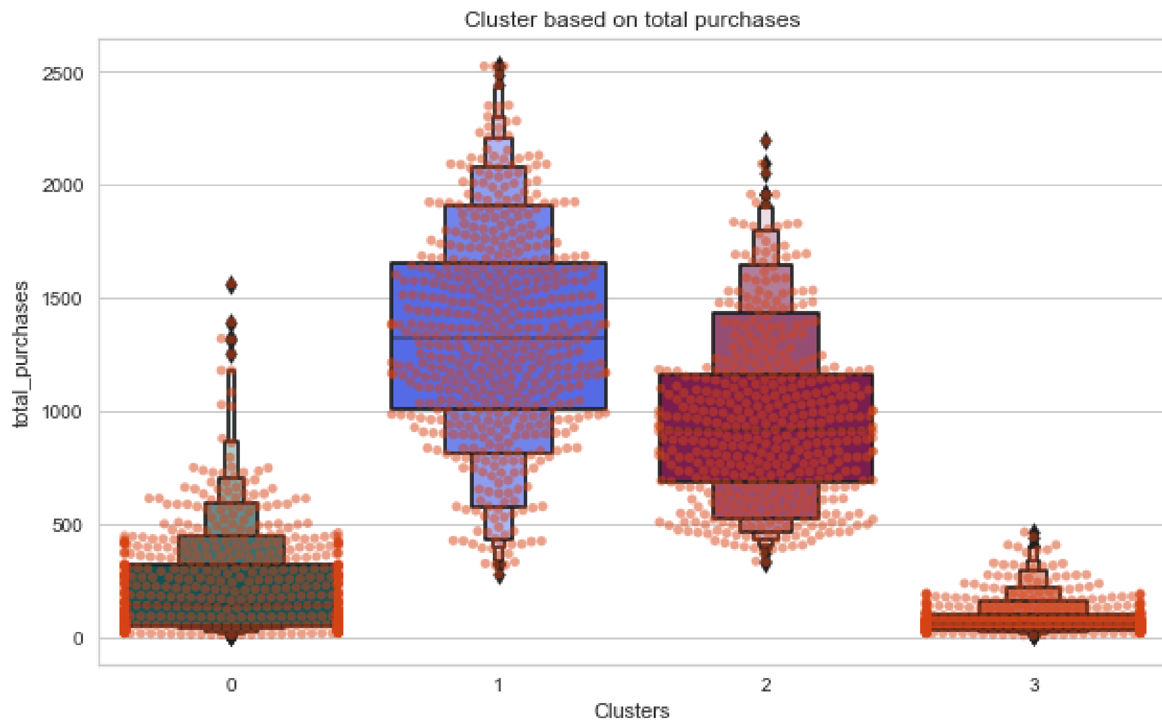
```
plt.figure(figsize=(12,8))
pl = sns.scatterplot(data = data_AC,x=data_AC["total_purchases"], y=data_AC["I
pl.set_title("Cluster's Profile Based On Income And Total Purchases")
plt.legend()
plt.show()
```



The relationship between income and the number of purchases by clusters

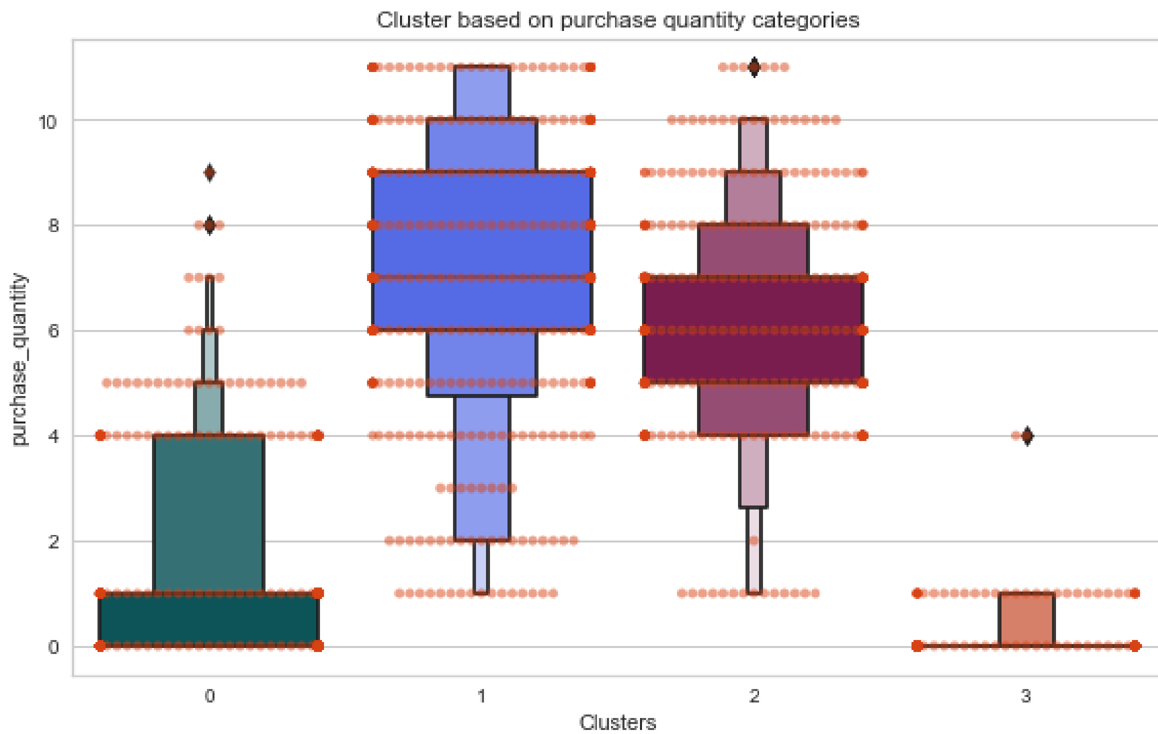
In [540]:

```
plt.figure(figsize=(10,6))
pl=sns.swarmplot(x=data_AC["Clusters"], y=data_AC["total_purchases"], color= '
pl=sns.boxenplot(x=data_AC["Clusters"], y=data_AC["total_purchases"], palette=
pl.set_title("Cluster based on total purchases")
plt.show()
```



In [541]:

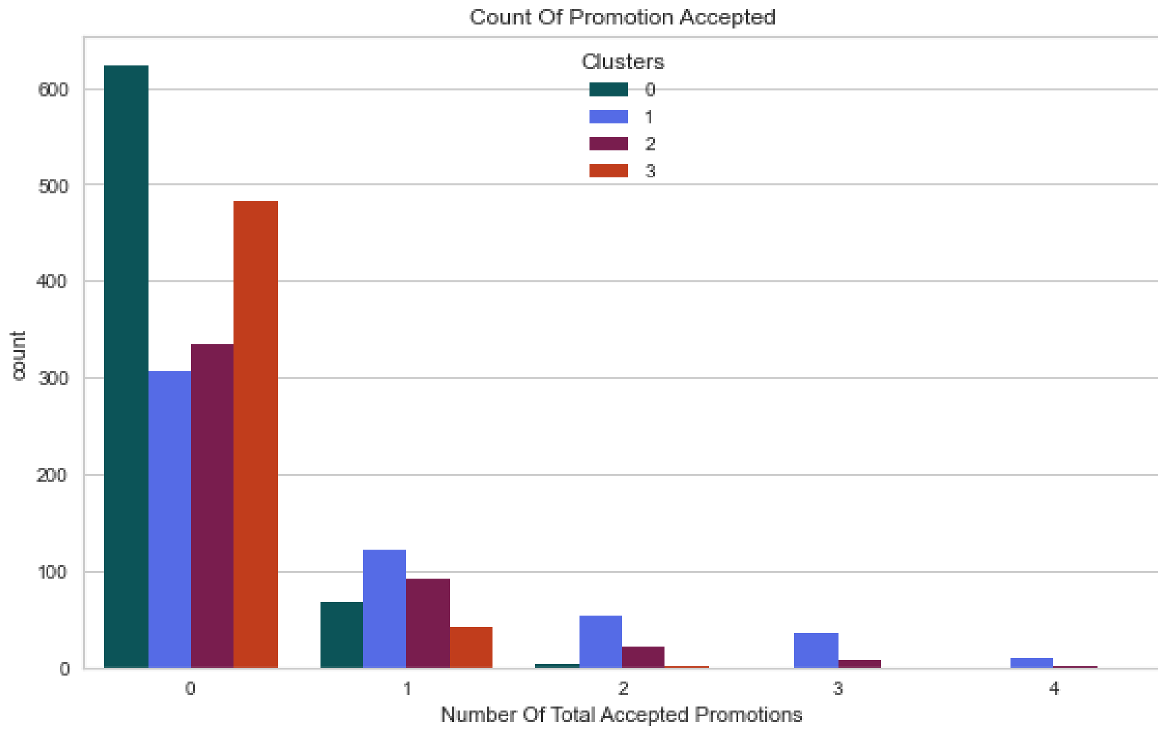
```
plt.figure(figsize=(10,6))
pl=sns.swarmplot(x=data_AC["Clusters"], y=data_AC["purchase_quantity"], color=
pl=sns.boxenplot(x=data_AC["Clusters"], y=data_AC["purchase_quantity"], palett
pl.set_title("Cluster based on purchase quantity categories")
plt.show()
```



In [ ]:

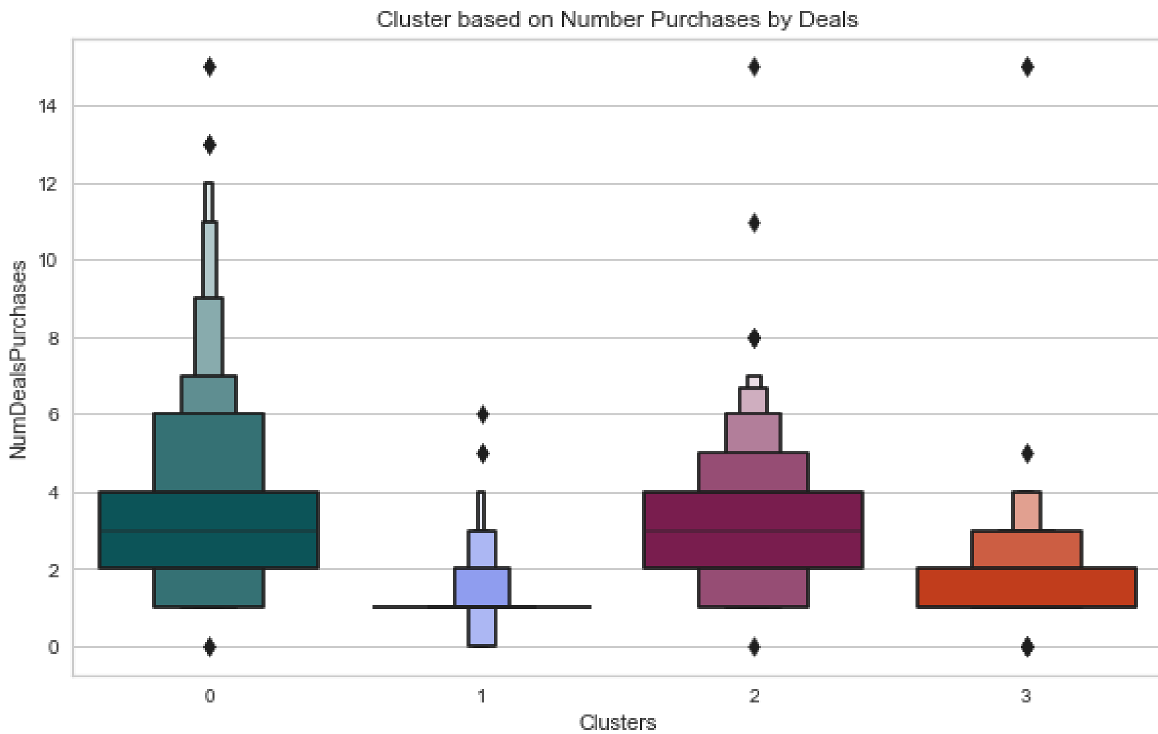
In [542]:

```
plt.figure(figsize=(10,6))
pl = sns.countplot(x=data_AC["Total_Promos"],hue=data_AC["Clusters"], palette=
pl.set_title("Count Of Promotion Accepted")
pl.set_xlabel("Number Of Total Accepted Promotions")
plt.show()
```



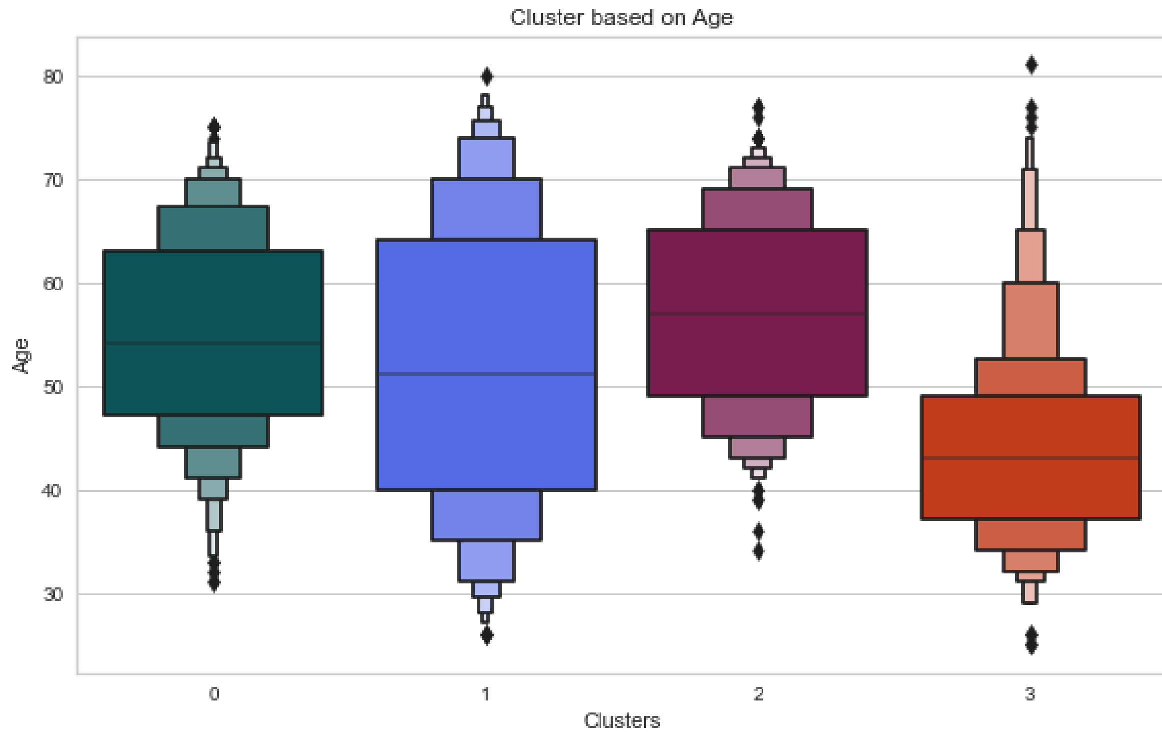
In [543]:

```
plt.figure(figsize=(10,6))
pl=sns.boxenplot(y=data_AC["NumDealsPurchases"],x=data_AC["Clusters"], palette
pl.set_title("Cluster based on Number Purchases by Deals")
plt.show()
```



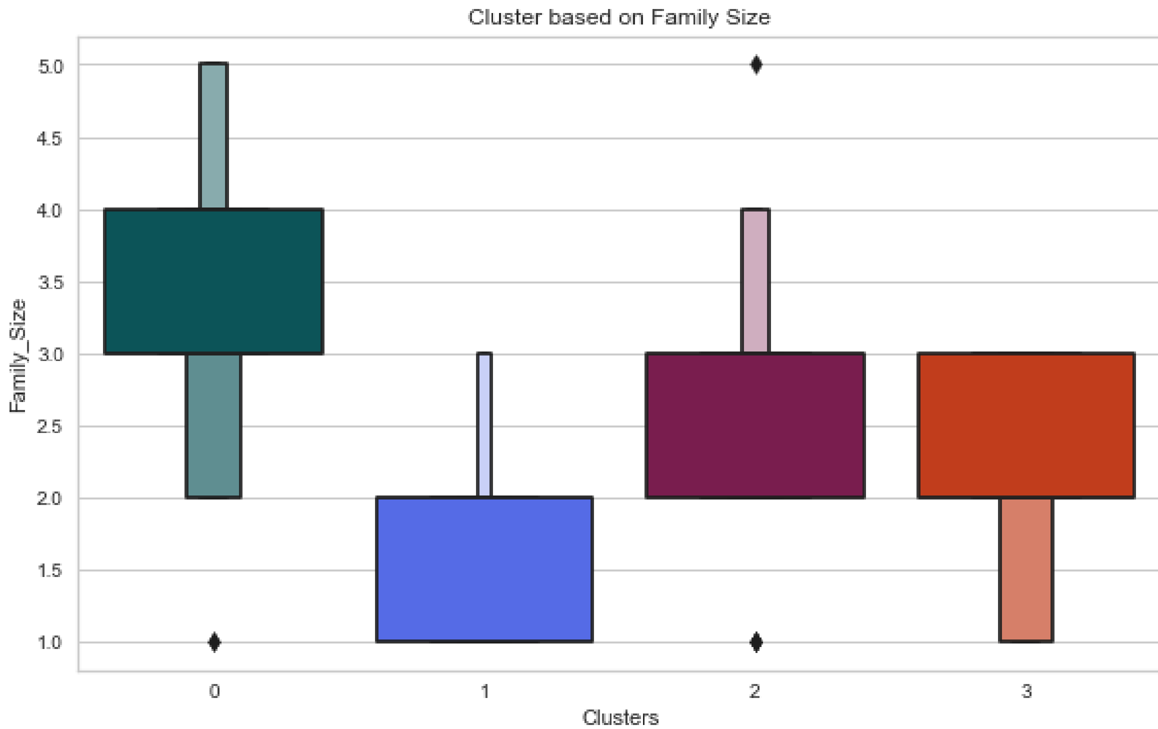
In [544]:

```
plt.figure(figsize=(10,6))
pl=sns.boxenplot(y=data_AC["Age"],x=data_AC["Clusters"], palette= color_2)
pl.set_title("Cluster based on Age")
plt.show()
```



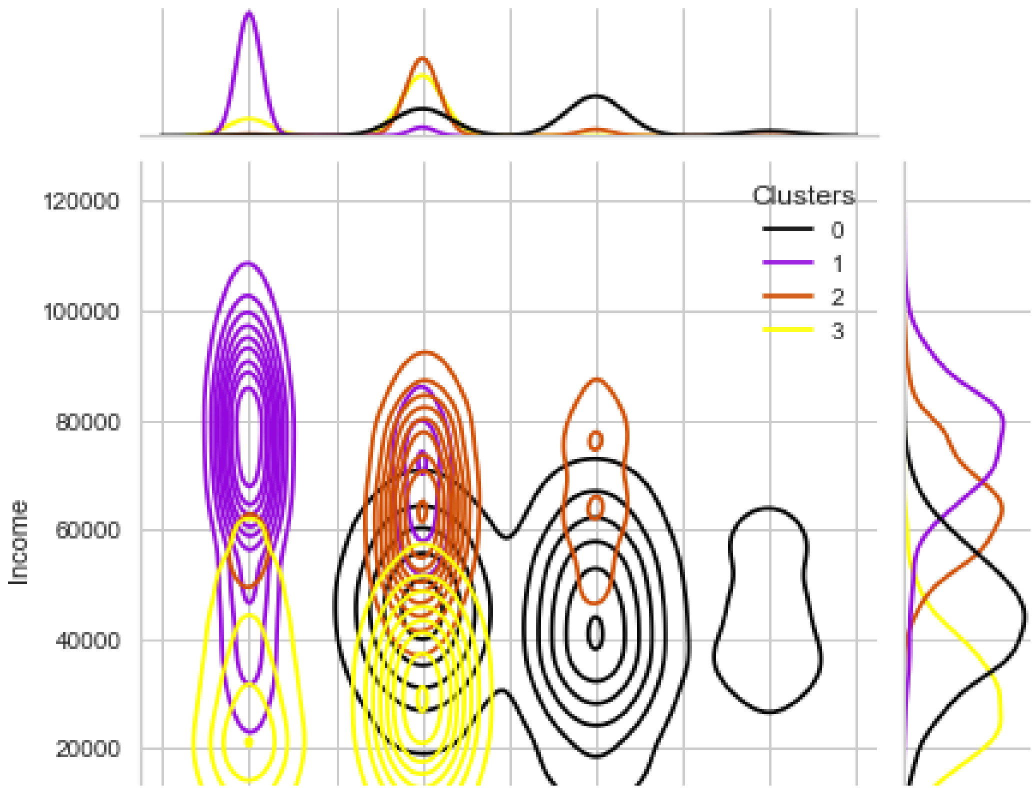
In [557]:

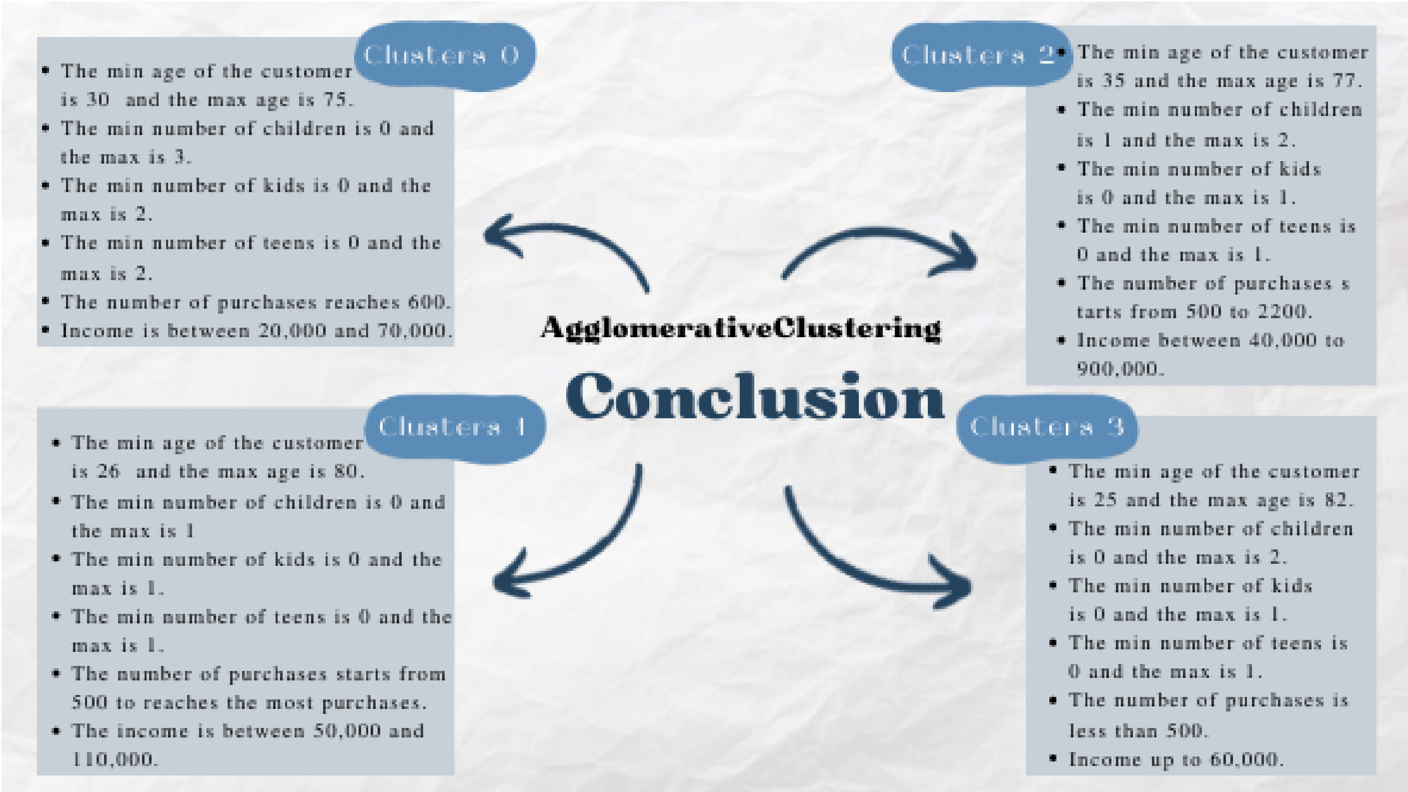
```
plt.figure(figsize=(10,6))
pl=sns.boxenplot(y=data_AC["Family_Size"],x=data_AC["Clusters"], palette= colc
pl.set_title("Cluster based on Family Size")
plt.show()
```



In [546]:

```
Personal = [ "Kidhome","Teenhome","Customer_From_days", "Age", "Num_Children",
sns.jointplot(x=data_AC['Num_Children'], y=data_AC["Income"], hue =data_AC["Cl
for i in Personal:
    plt.figure()
    sns.jointplot(x=data[i], y=data_AC["total_purchases"], hue =data_AC["Clust
    plt.show()
```





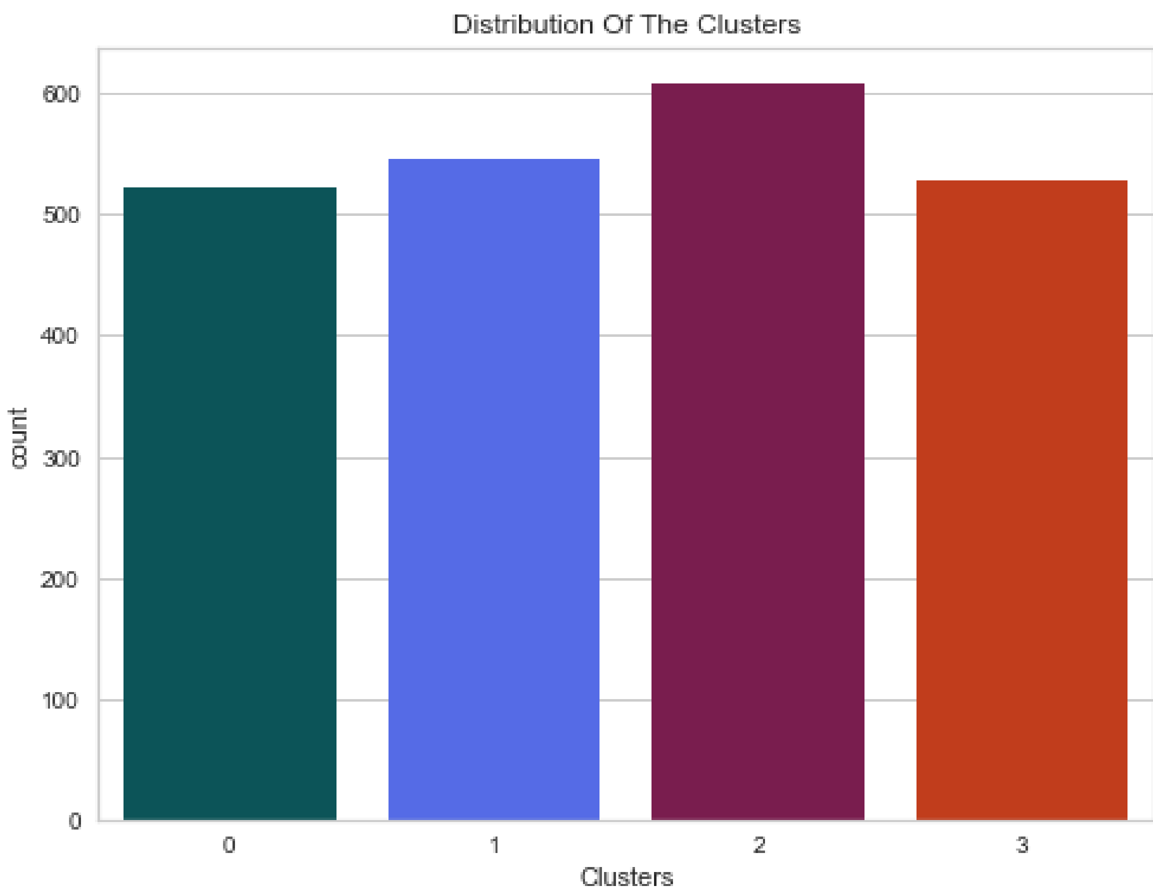
https://www.canva.com/design/DAFRH7dSxMc/Z5q7lD3JD7F1qKQFLCDPgQ/view?utm_content=DAFRH7dSxMc&utm_campaign=designshare&utm_medium=link&utm_source=hor
(https://www.canva.com/design/DAFRH7dSxMc/Z5q7lD3JD7F1qKQFLCDPgQ/view?utm_content=DAFRH7dSxMc&utm_campaign=designshare&utm_medium=link&utm_source=hor

In [ ]:

K-Means

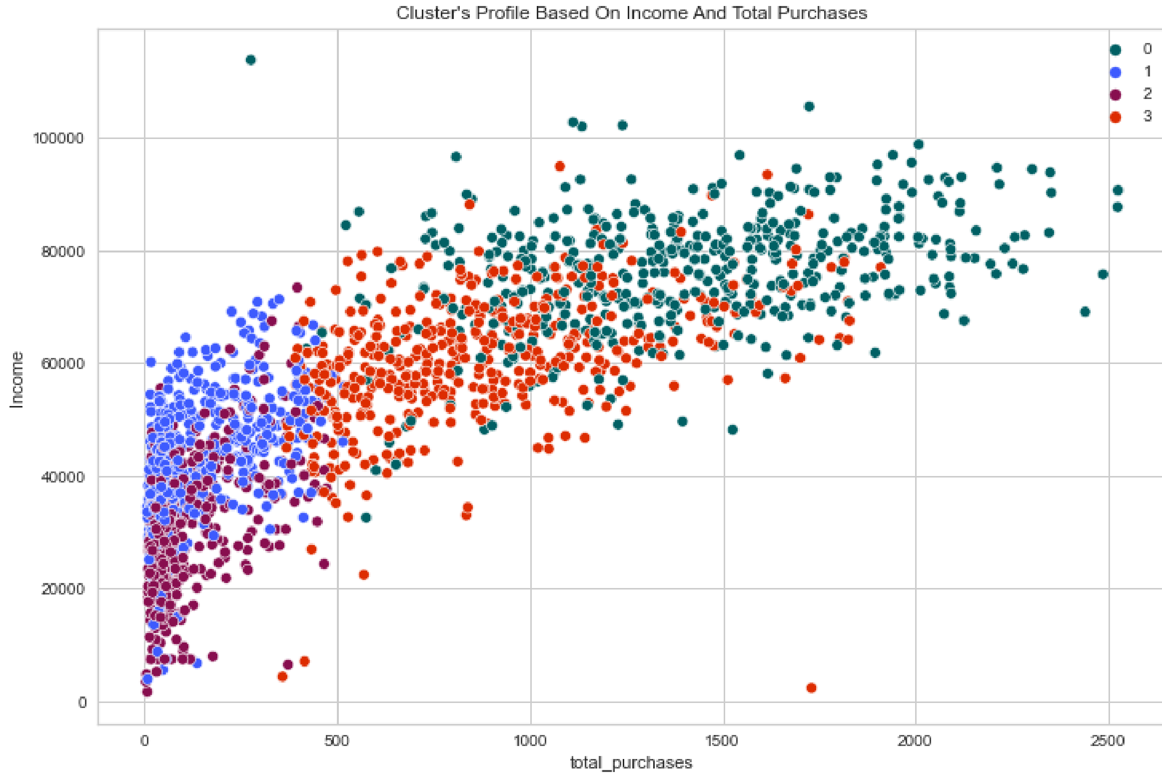
In [547]:

```
#Plotting countplot of clusters
plt.figure(figsize=(8,6))
pl = sns.countplot(x=data_KM["Clusters"], palette=color_2 )
pl.set_title("Distribution Of The Clusters")
plt.show()
```



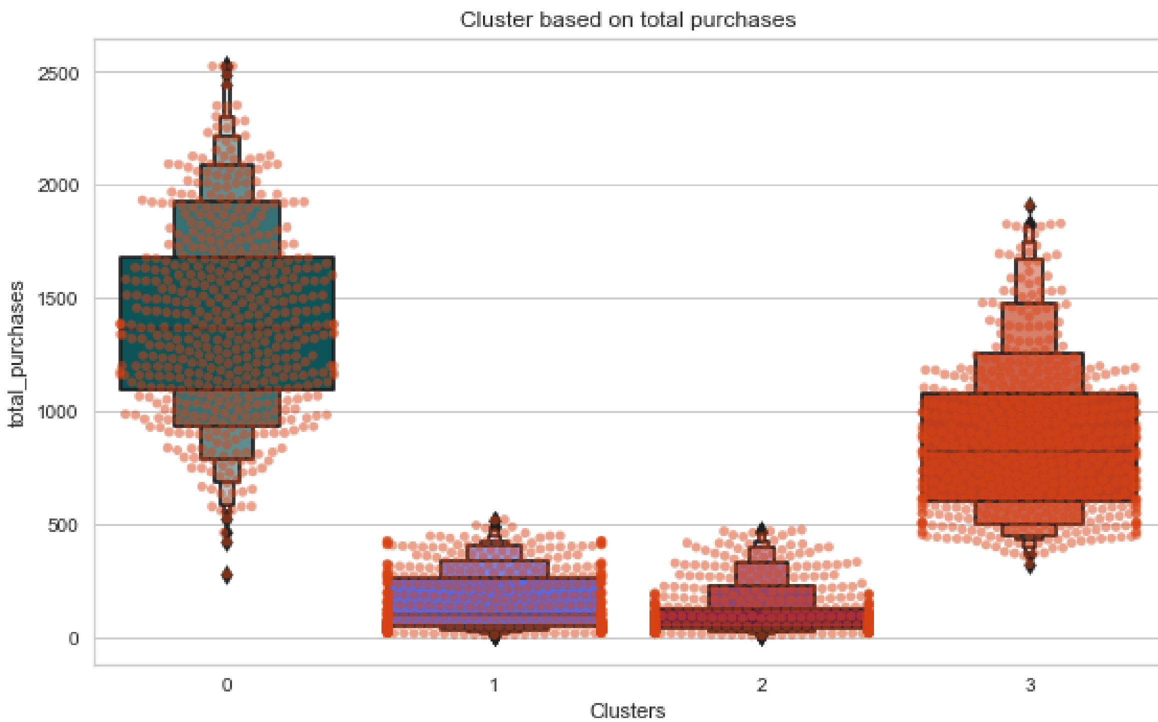
In [549]:

```
plt.figure(figsize=(12,8))
pl = sns.scatterplot(data = data_KM,x=data_KM["total_purchases"], y=data_KM["I
pl.set_title("Cluster's Profile Based On Income And Total Purchases")
plt.legend()
plt.show()
```



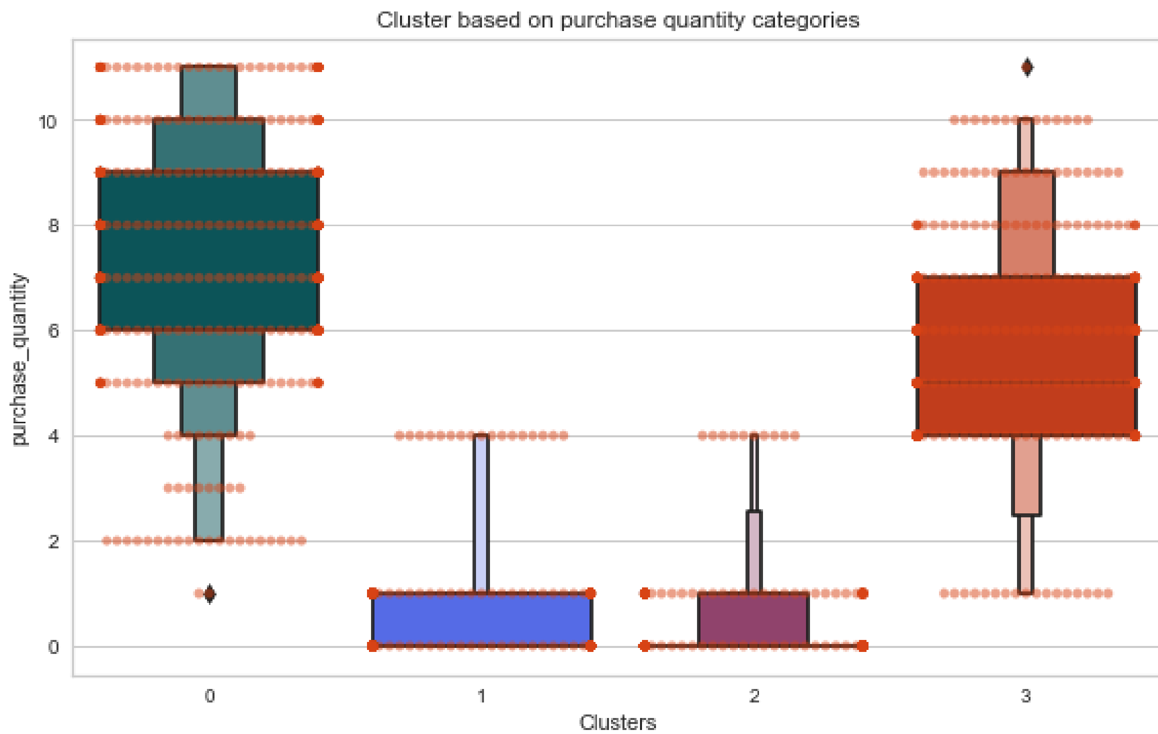
In [550]:

```
plt.figure(figsize=(10,6))
pl=sns.swarmplot(x=data_KM["Clusters"], y=data_KM["total_purchases"], color= '
pl=sns.boxenplot(x=data_KM["Clusters"], y=data_KM["total_purchases"], palette=
pl.set_title("Cluster based on total purchases")
plt.show()
```



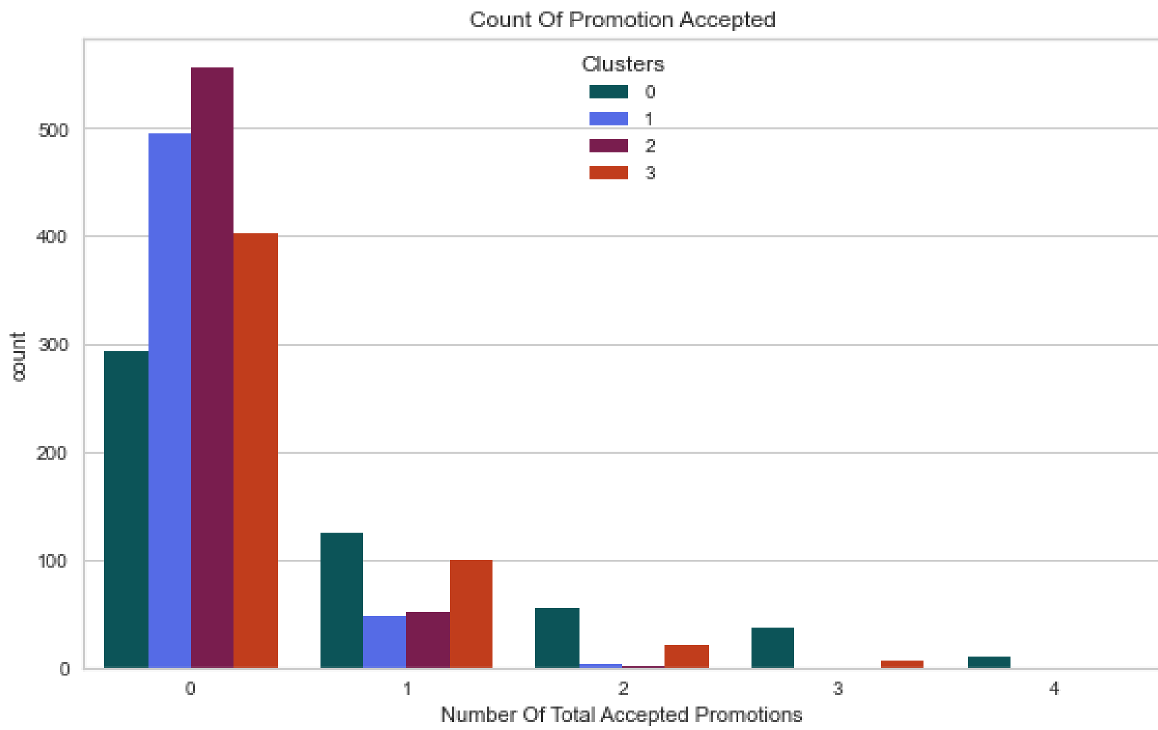
In [551]:

```
plt.figure(figsize=(10,6))
pl=sns.swarmplot(x=data_KM["Clusters"], y=data_KM["purchase_quantity"], color=
pl=sns.boxenplot(x=data_KM["Clusters"], y=data_KM["purchase_quantity"], palett
pl.set_title("Cluster based on purchase quantity categories")
plt.show()
```



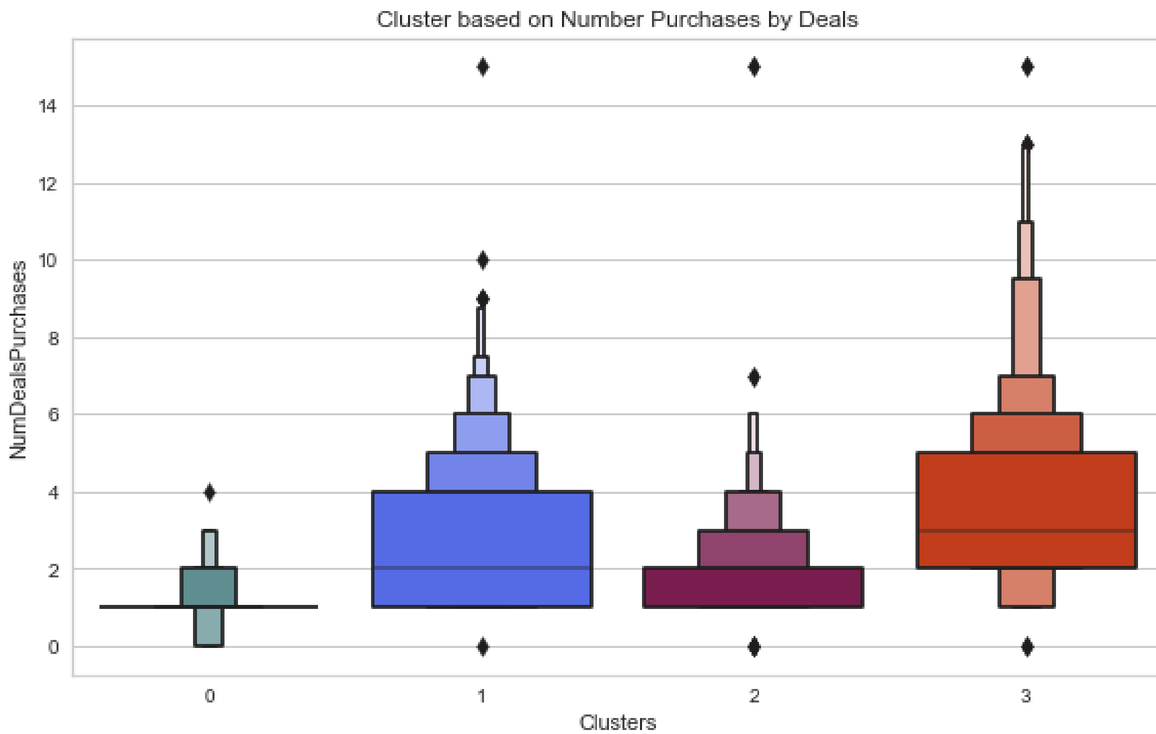
In [552]:

```
plt.figure(figsize=(10,6))
pl = sns.countplot(x=data_KM["Total_Promos"],hue=data_KM["Clusters"], palette=
pl.set_title("Count Of Promotion Accepted")
pl.set_xlabel("Number Of Total Accepted Promotions")
plt.show()
```



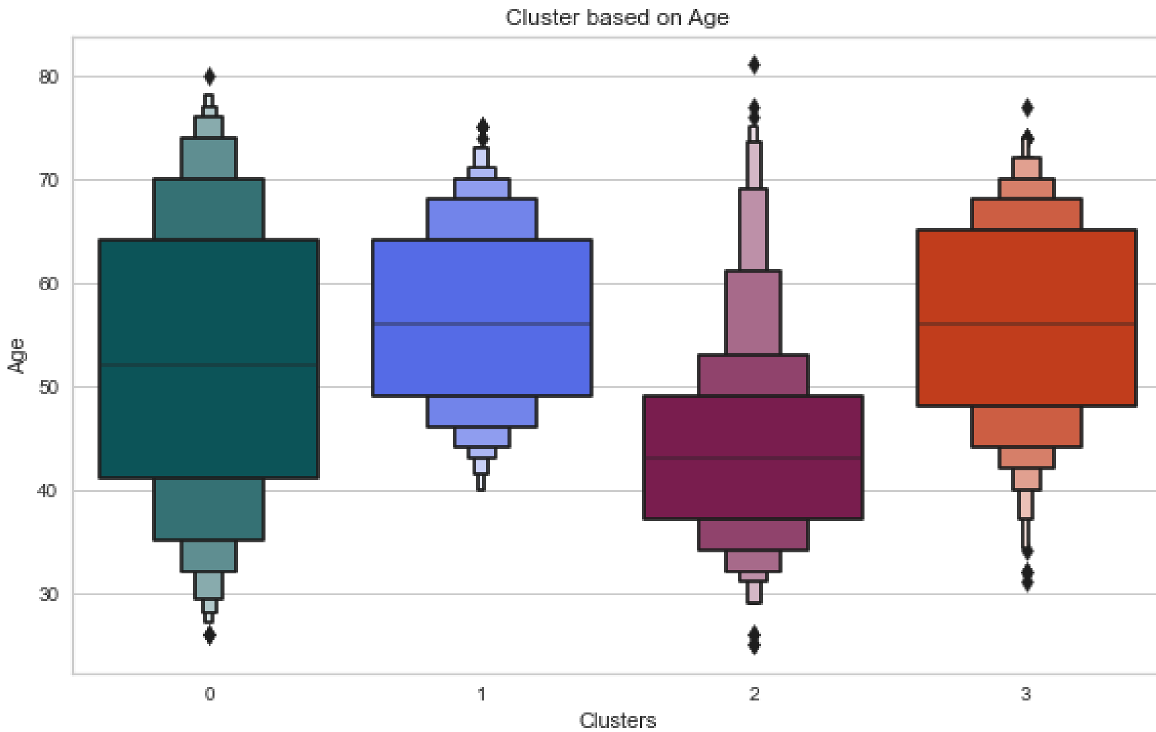
In [553]:

```
plt.figure(figsize=(10,6))
pl=sns.boxenplot(y=data_KM["NumDealsPurchases"],x=data_KM["Clusters"], palette
pl.set_title("Cluster based on Number Purchases by Deals")
plt.show()
```



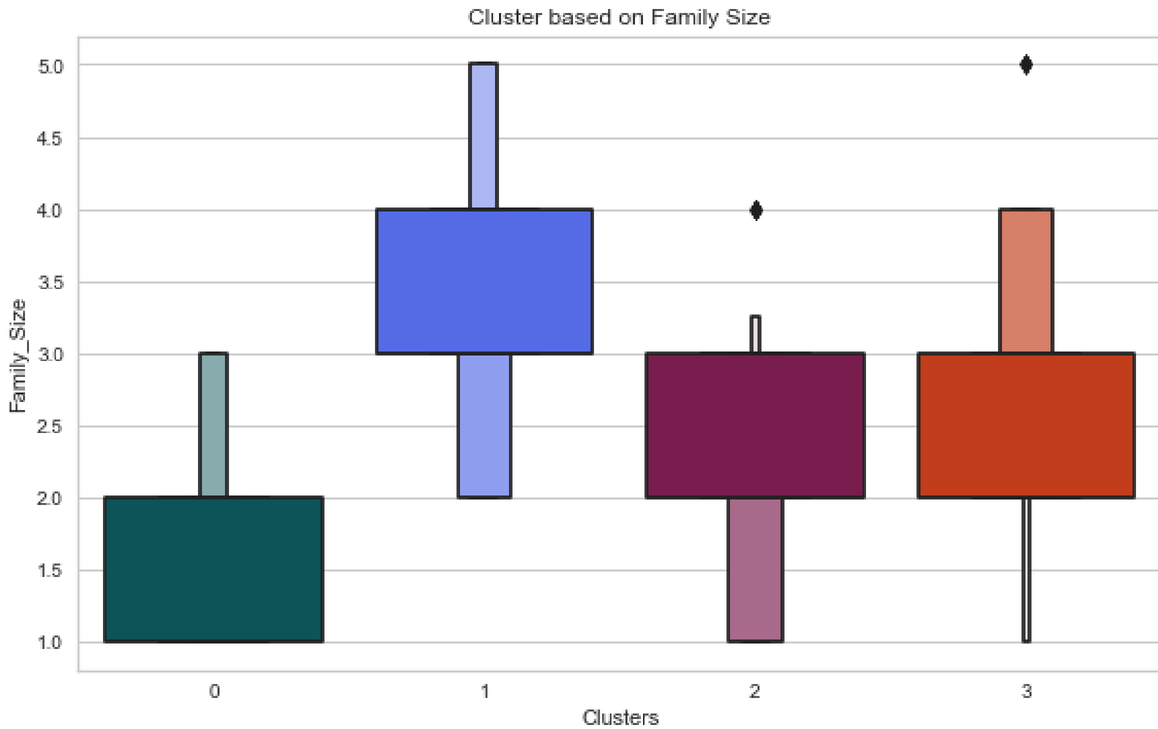
In [554]:

```
plt.figure(figsize=(10,6))
pl=sns.boxenplot(y=data_KM["Age"],x=data_KM["Clusters"], palette= color_2)
pl.set_title("Cluster based on Age")
plt.show()
```



In [558]:

```
plt.figure(figsize=(10,6))
pl=sns.boxenplot(y=data_KM["Family_Size"],x=data_KM["Clusters"], palette= colo
pl.set_title("Cluster based on Family Size")
plt.show()
```



In [633]:

```
# It is used to verify the data
t=data_KM[data_KM['Clusters']==3][data_KM['total_purchases']<400]
t
```

Out[633]:

	Education	Marital_Status	Income	Recency	MntWines	MntFruits	MntM
257	2	2	45989.0	97	138	33	
570	2	1	44989.0	26	98	0	
637	2	1	56181.0	6	121	103	
1375	2	2	48904.0	1	283	10	
1751	2	1	44989.0	26	98	0	
1796	2	2	62535.0	13	163	48	
1957	2	2	46998.0	55	172	41	
1975	2	1	4428.0	0	16	4	
2035	2	3	60905.0	27	208	17	

9 rows × 25 columns

In [600]:

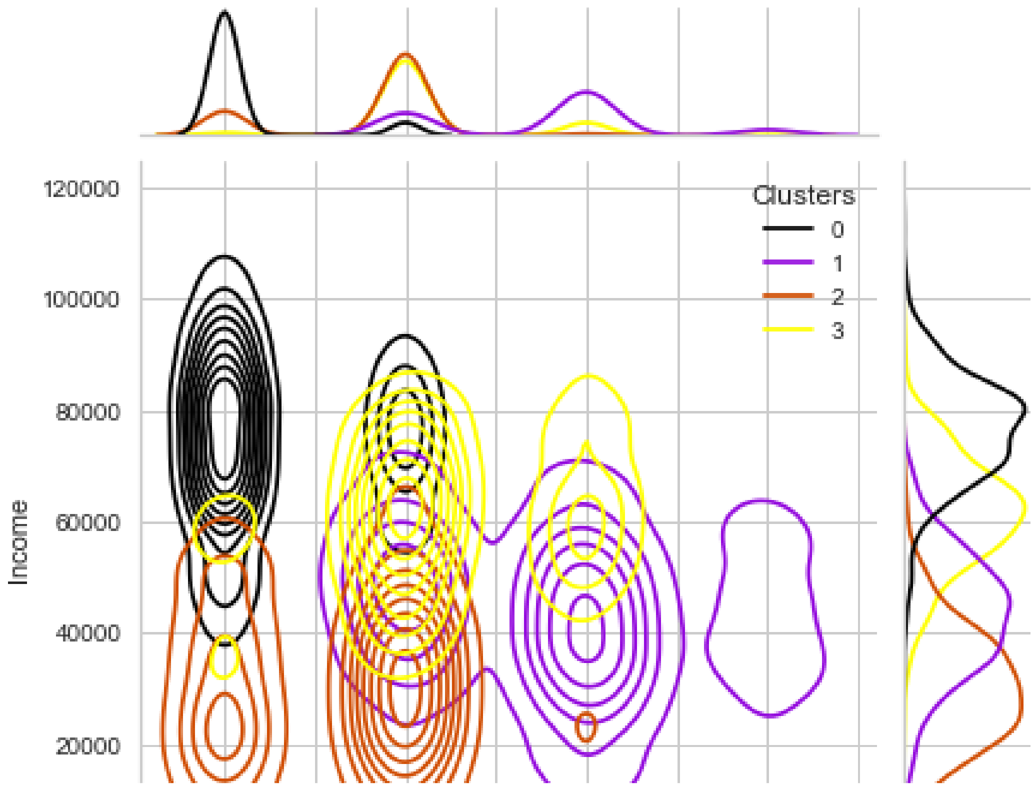
```
```

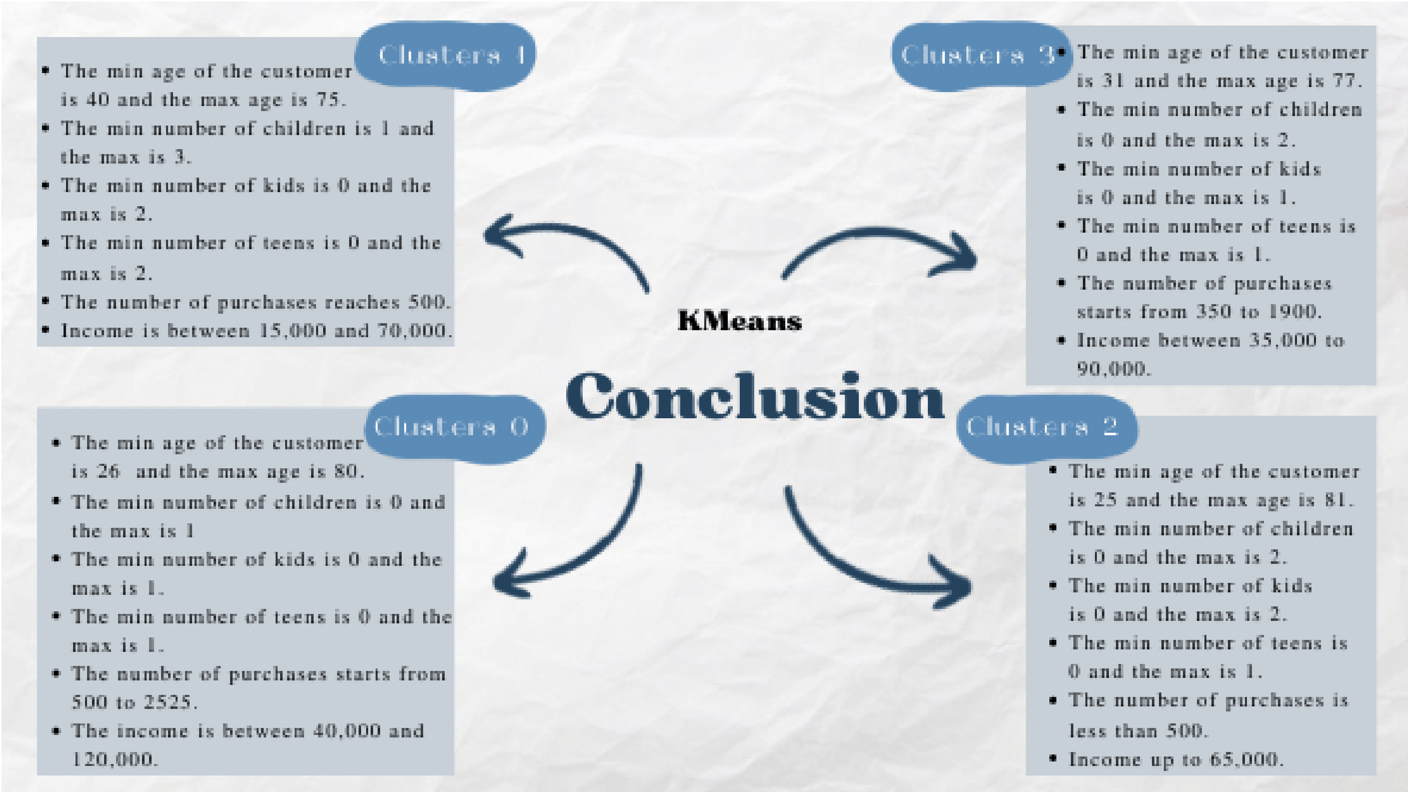
Out[600]:

```
14299.947368421053
```

In [ ]:

```
Personal = [ "Kidhome","Teenhome","Customer_From_days", "Age", "Num_Children",
sns.jointplot(x=data_KM['Num_Children'], y=data_KM["Income"], hue =data_KM["Cl
for i in Personal:
    plt.figure()
    sns.jointplot(x=data_KM[i], y=data_KM["total_purchases"], hue =data_KM["Cl
    plt.show()
```





https://www.canva.com/design/DAFRH7dSxMc/Z5q7lD3JD7F1qKQFLCDPgQ/view?utm_content=DAFRH7dSxMc&utm_campaign=designshare&utm_medium=link&utm_source=hor
(https://www.canva.com/design/DAFRH7dSxMc/Z5q7lD3JD7F1qKQFLCDPgQ/view?utm_content=DAFRH7dSxMc&utm_campaign=designshare&utm_medium=link&utm_source=hor

In [ ]:

```

```

In [ ]:

```

```