# Problem Statement

---

## Business Context

Understanding customer personality and behavior is pivotal for businesses to enhance customer satisfaction and increase revenue. Segmentation based on a customer's personality, demographics, and purchasing behavior allows companies to create tailored marketing campaigns, improve customer retention, and optimize product offerings.

A leading retail company with a rapidly growing customer base seeks to gain deeper insights into their customers' profiles. The company recognizes that understanding customer personalities, lifestyles, and purchasing habits can unlock significant opportunities for personalizing marketing strategies and creating loyalty programs. These insights can help address critical business challenges, such as improving the effectiveness of marketing campaigns, identifying high-value customer groups, and fostering long-term relationships with customers.

With the competition intensifying in the retail space, moving away from generic strategies to more targeted and personalized approaches is essential for sustaining a competitive edge.

---

## Objective

In an effort to optimize marketing efficiency and enhance customer experience, the company has embarked on a mission to identify distinct customer segments. By understanding the characteristics, preferences, and behaviors of each group, the company aims to:

1. Develop personalized marketing campaigns to increase conversion rates.
2. Create effective retention strategies for high-value customers.
3. Optimize resource allocation, such as inventory management, pricing strategies, and store layouts.

As a data scientist tasked with this project, your responsibility is to analyze the given customer data, apply machine learning techniques to segment the customer base, and provide actionable insights into the characteristics of each segment.

---

## Data Dictionary

The dataset includes historical data on customer demographics, personality traits, and purchasing behaviors. Key attributes are:

1. **Customer Information**

- **ID:** Unique identifier for each customer.
- **Year_Birth:** Customer's year of birth.
- **Education:** Education level of the customer.
- **Marital_Status:** Marital status of the customer.
- **Income:** Yearly household income (in dollars).
- **Kidhome:** Number of children in the household.
- **Teenhome:** Number of teenagers in the household.
- **Dt_Customer:** Date when the customer enrolled with the company.
- **Recency:** Number of days since the customer's last purchase.
- **Complain:** Whether the customer complained in the last 2 years (1 for yes, 0 for no).

2. **Spending Information (Last 2 Years)**

- **MntWines:** Amount spent on wine.
- **MntFruits:** Amount spent on fruits.
- **MntMeatProducts:** Amount spent on meat.
- **MntFishProducts:** Amount spent on fish.
- **MntSweetProducts:** Amount spent on sweets.
- **MntGoldProds:** Amount spent on gold products.

3. **Purchase and Campaign Interaction**

- **NumDealsPurchases:** Number of purchases made using a discount.
- **AcceptedCmp1:** Response to the 1st campaign (1 for yes, 0 for no).
- **AcceptedCmp2:** Response to the 2nd campaign (1 for yes, 0 for no).
- **AcceptedCmp3:** Response to the 3rd campaign (1 for yes, 0 for no).
- **AcceptedCmp4:** Response to the 4th campaign (1 for yes, 0 for no).
- **AcceptedCmp5:** Response to the 5th campaign (1 for yes, 0 for no).
- **Response:** Response to the last campaign (1 for yes, 0 for no).

4. **Shopping Behavior**

- **NumWebPurchases:** Number of purchases made through the company's website.
- **NumCatalogPurchases:** Number of purchases made using catalogs.
- **NumStorePurchases:** Number of purchases made directly in stores.
- **NumWebVisitsMonth:** Number of visits to the company's website in the last month.

# Let's start coding!

## Importing necessary libraries

In [1]:
```
# Libraries to help with reading and manipulating data
import pandas as pd
import numpy as np
```

```
# libaries to help with data visualization
import matplotlib.pyplot as plt
import seaborn as sns

# Removes the limit for the number of displayed columns
pd.set_option("display.max_columns", None)
# Sets the limit for the number of displayed rows
pd.set_option("display.max_rows", 200)

# to scale the data using z-score
from sklearn.preprocessing import StandardScaler

# to compute distances
from scipy.spatial.distance import cdist, pdist

# to perform k-means clustering and compute silhouette scores
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

# to visualize the elbow curve and silhouette scores
from yellowbrick.cluster import KElbowVisualizer, SilhouetteVisualizer

# to suppress warnings
import warnings

warnings.filterwarnings("ignore")
```

# Loading the data

In [2]:
```
# uncomment and run the following line if using Google Colab
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

In [3]:
```
# loading data into a pandas dataframe
data = pd.read_csv("/content/drive/MyDrive/classes/Summer 2025/MIT - Data Science a
```

# Data Overview

In [4]:
```
data.head()
data.tail()
```

Out[4]:

| | ID | Year_Birth | Education | Marital_Status | Income | Kidhome | Teenhome | Dt_Cust |
|---|---|---|---|---|---|---|---|---|
| **2235** | 10870 | 1967 | Graduation | Married | 61223.0 | 0 | 1 | 13-06 |
| **2236** | 4001 | 1946 | PhD | Together | 64014.0 | 2 | 1 | 10-06 |
| **2237** | 7270 | 1981 | Graduation | Divorced | 56981.0 | 0 | 0 | 25-01 |
| **2238** | 8235 | 1956 | Master | Together | 69245.0 | 0 | 1 | 24-01 |
| **2239** | 9405 | 1954 | PhD | Married | 52869.0 | 1 | 1 | 15-10 |

## **Question 1**: What are the data types of all the columns?

In [5]:
```
data.info()
# all columns except 2, 3, 4, 5, 6, and 7 are integers
# columns 2, 3, and 7 are of the object datatypes and therefore probably include di
# column 4 is of the float datatype
```

```
<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
```

**Observations:**

All columns except 2, 3, 4, 5, 6, and 7 are integers.

Columns 2, 3, and 7 are of the object datatypes and therefore probably include different datatypes such as arrays that can hold both strings and lists.

Column 4 is of the float datatype.

## **Question 2:** Check the statistical summary of the data. What is the average household income?

In [6]:
```
data.describe()
# "average" refers to the mean, and the mean of the income column is $52247.251354
```

Out[6]:

| | ID | Year_Birth | Income | Kidhome | Teenhome | Recency | |
|---|---|---|---|---|---|---|---|
| **count** | 2240.000000 | 2240.000000 | 2216.000000 | 2240.000000 | 2240.000000 | 2240.000000 | 2 |
| **mean** | 5592.159821 | 1968.805804 | 52247.251354 | 0.444196 | 0.506250 | 49.109375 | |
| **std** | 3246.662198 | 11.984069 | 25173.076661 | 0.538398 | 0.544538 | 28.962453 | |
| **min** | 0.000000 | 1893.000000 | 1730.000000 | 0.000000 | 0.000000 | 0.000000 | |
| **25%** | 2828.250000 | 1959.000000 | 35303.000000 | 0.000000 | 0.000000 | 24.000000 | |
| **50%** | 5458.500000 | 1970.000000 | 51381.500000 | 0.000000 | 0.000000 | 49.000000 | |
| **75%** | 8427.750000 | 1977.000000 | 68522.000000 | 1.000000 | 1.000000 | 74.000000 | |
| **max** | 11191.000000 | 1996.000000 | 666666.000000 | 2.000000 | 2.000000 | 99.000000 | 1 |

**Observations:**

"average" refers to the mean, and the mean of the income column is $52247.251354

## Question 3: Are there any missing values in the data? If yes, treat them using an appropriate method

In [7]:
```
data.isnull().sum()
# the income column has 24 null values

sns.heatmap(data.isnull(), cbar=False, cmap='viridis')
plt.show
# a heatmap helps us visulize where the null values are
# this heat map shows that there are null values around the 1350-1500 # of rows in
```

Out[7]:

**matplotlib.pyplot.show**

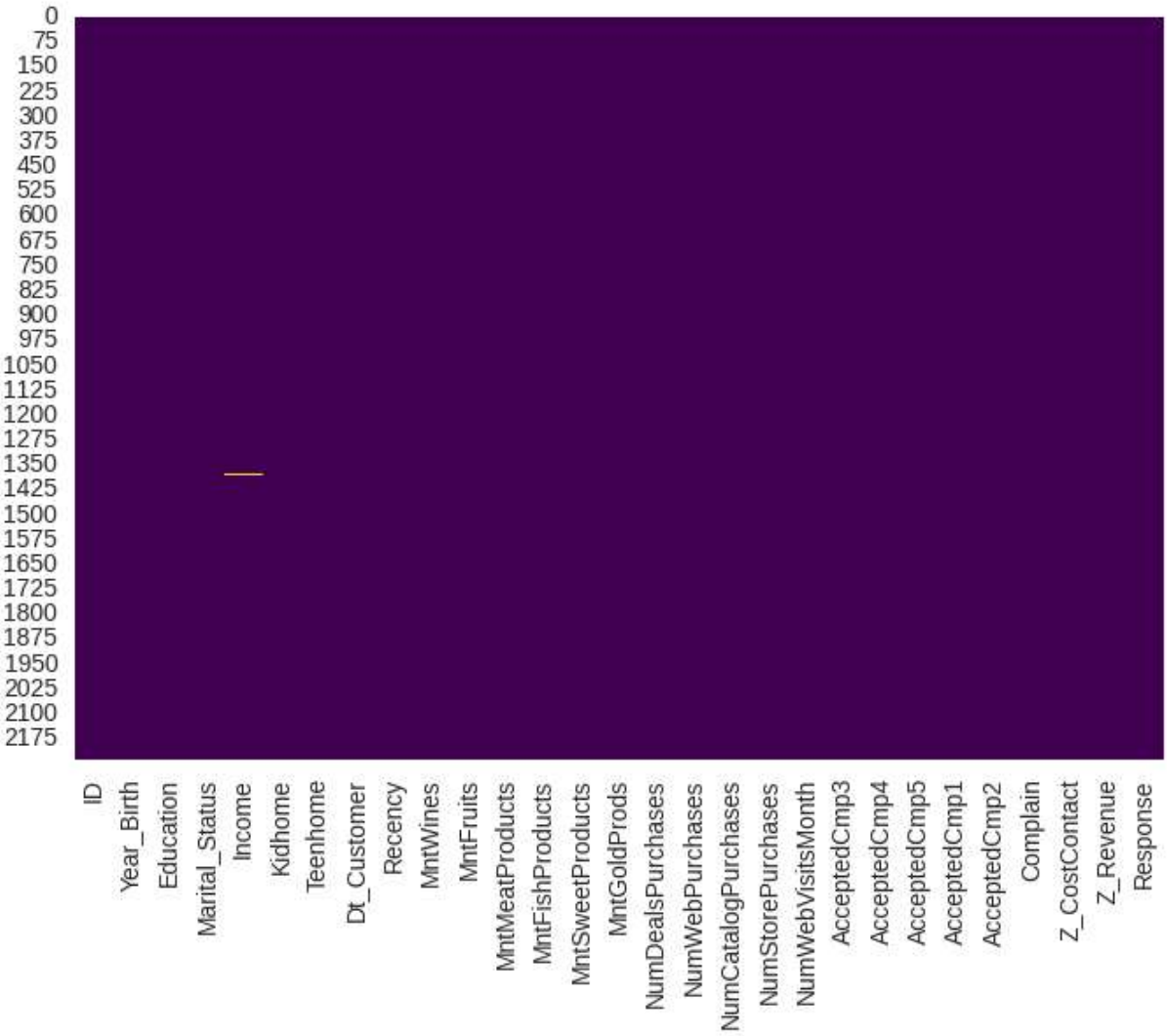
def show(*args, **kwargs) -> None

Display all open figures.

Parameters

---

block : bool, optional




In [8]:
```
data.dropna(inplace=True)
data.isnull().sum()

# to drop null values in a dataset without assigning the dataset to a new variable,
# there are now no null values in the income column of our dataset
```

Out[8]:

| | 0 |
|---|---|
| **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

**Observations:**

The income column has 24 null values, but now they are dropped.

### **Question 4**: Are there any duplicates in the data?

In [9]:
```
row_duplcates = data.duplicated().sum() # This function returns "np.int64(0)", and

# Some of the columns only had a limited possible data values (EX: 1 or 0), so it w
# Applying each column to the duplcate function individually found that there were
one = data['Education'].duplicated().sum()
two = data['AcceptedCmp2'].duplicated().sum()
print(row_duplcates, one, two)

# Assuming the question is refering to the duplicate of rows, there are no duplicat
```

```
0 2211 2214
```

**Observations:**

Assuming the question is refering to the duplicate of rows, there are no duplicates.

# Exploratory Data Analysis

## Univariate Analysis

In univariate analysis we are looking for these common takeaways:

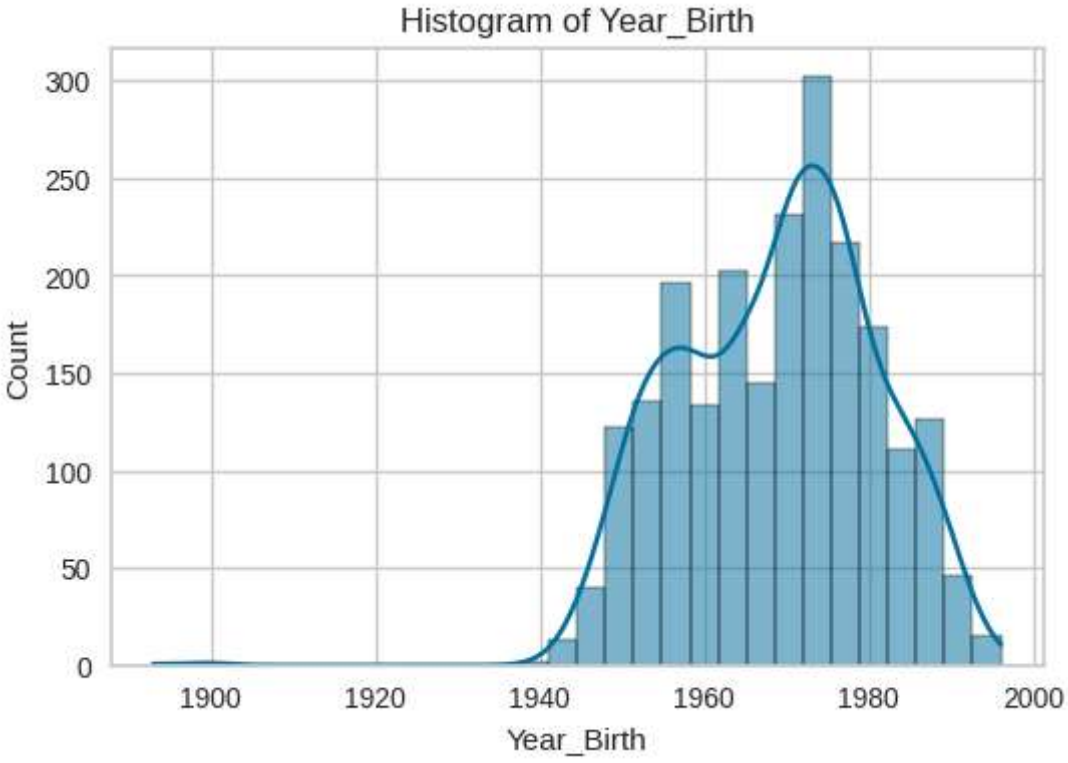
1. Distribution shape. Is is normal, skewed left or right, or multimodal? Do certain features need to be engineered (log transformed or combined with another feature)?
2. Central Tendancy. What is the mean/median?
3. Outliers. Are there outragous values that skew data? What needs to be cleaned? Are there people who don't spend anything? These could be inactive customers. Are there negative income values? These need to be dropped.
4. Spread. Is the data tightly clustered or really spread out?

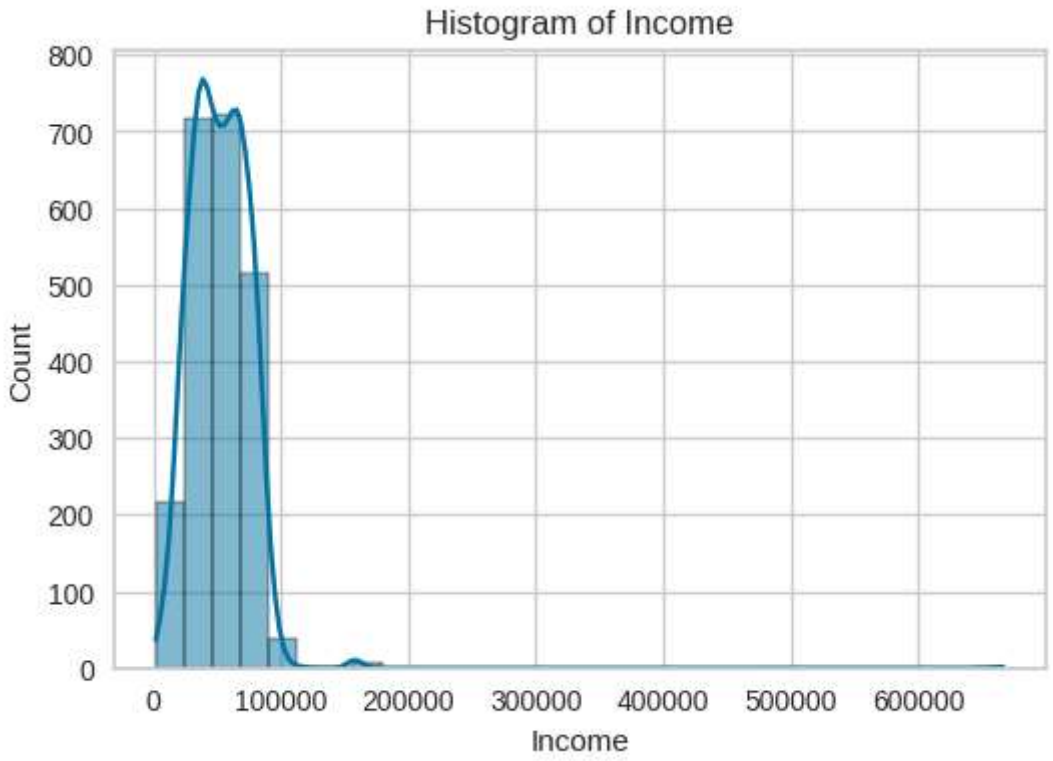
Conclusion: Univariate analysis helps with two aspects of data science: Understanding each variable before clustering. This allows the data scientist to look for and identify patterns before the ML algorithm. Data cleaning is the other aspect. Seeing each variable's distribution allows the data scientist to manicure the data so that it doesn't affect clustering and business recomendations down the line.

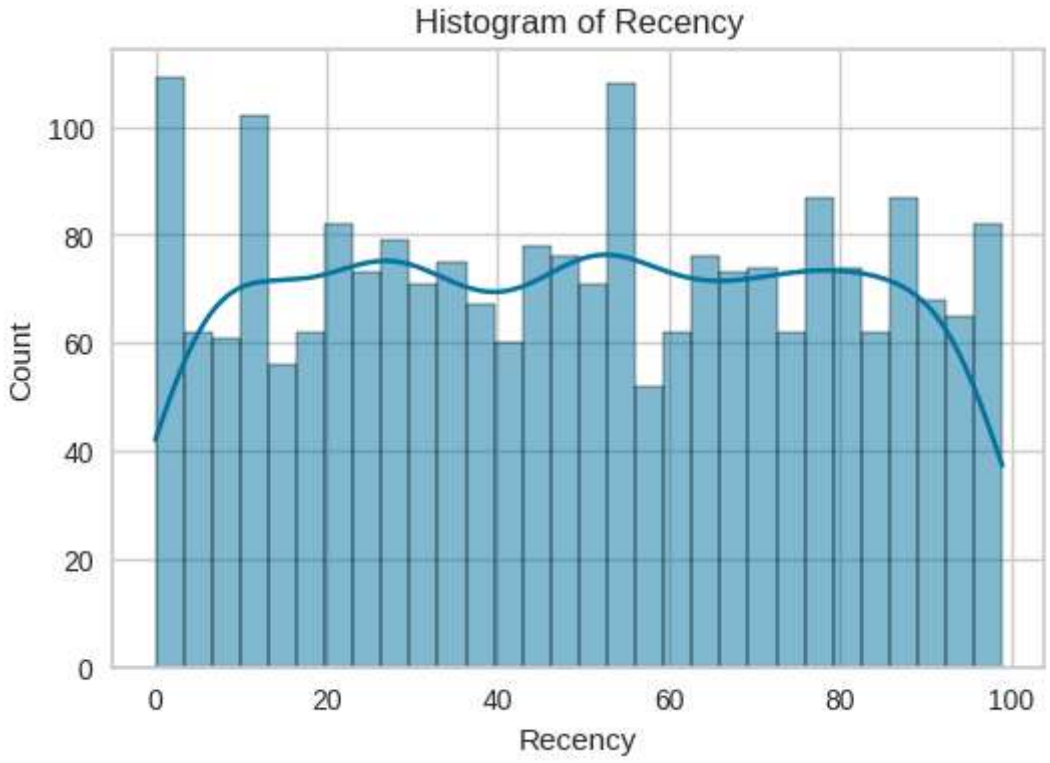
### **Question 5:** Explore all the variables and provide observations on their distributions. (histograms and boxplots)

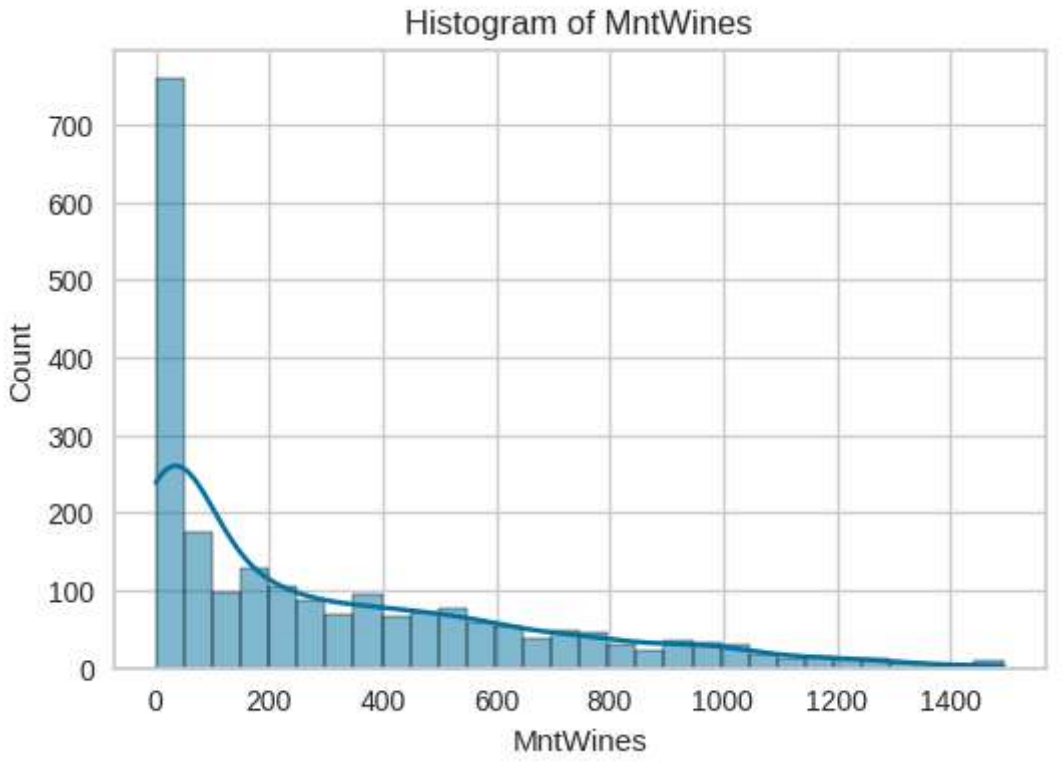
In [10]:
```
#Not all of the columns are useful to the business problem statement. I am interest
#"ID" is a unique identifier and is not helpful for analysis, so I will drop this c
```

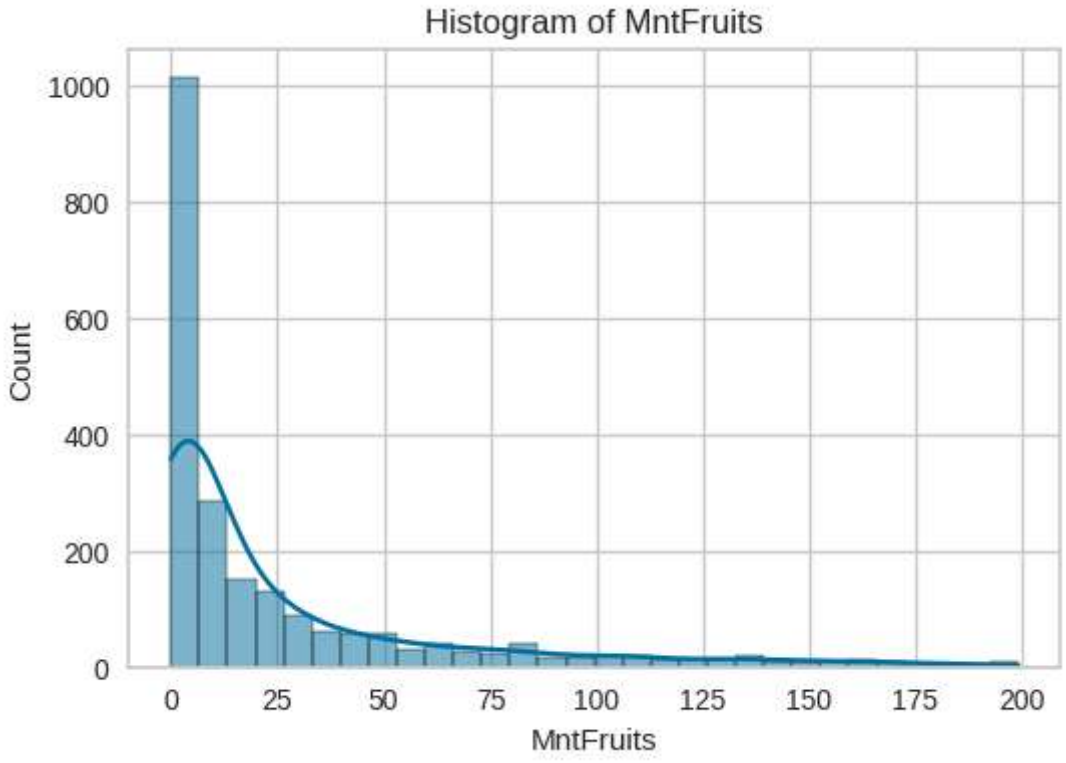
```
drop_columns = ['ID', 'Dt_Customer']
data.drop(columns=drop_columns, inplace=True)

continuous_features = data[['Year_Birth', 'Income', 'Recency', 'MntWines', 'MntFrui
categorical_binary_and_discrete_features = data[['Education', 'Marital_Status', 'Ki

#this for loop cycles through each column and creates a histogram for each column w
for col in continuous_features:
  plt.figure(figsize=(6, 4))
  sns.histplot(data, x=col, bins=30, kde=True)
  plt.title(f"Histogram of {col}")
  plt.show()

#this for loop cycles through each column and creats a boxplot for each column with
for col in continuous_features:
  plt.figure(figsize=(6, 4))
  sns.boxplot(y=data[col])
  plt.title(f"Boxplot of {col}")
  plt.show()


#this for loop cycles through each column and creats a countplot for each column of
for col in categorical_binary_and_discrete_features:
  plt.figure(figsize=(6, 4))
  sns.countplot(x=col, data=data)
  plt.title(f"Countplot of {col}")
  plt.show()
```

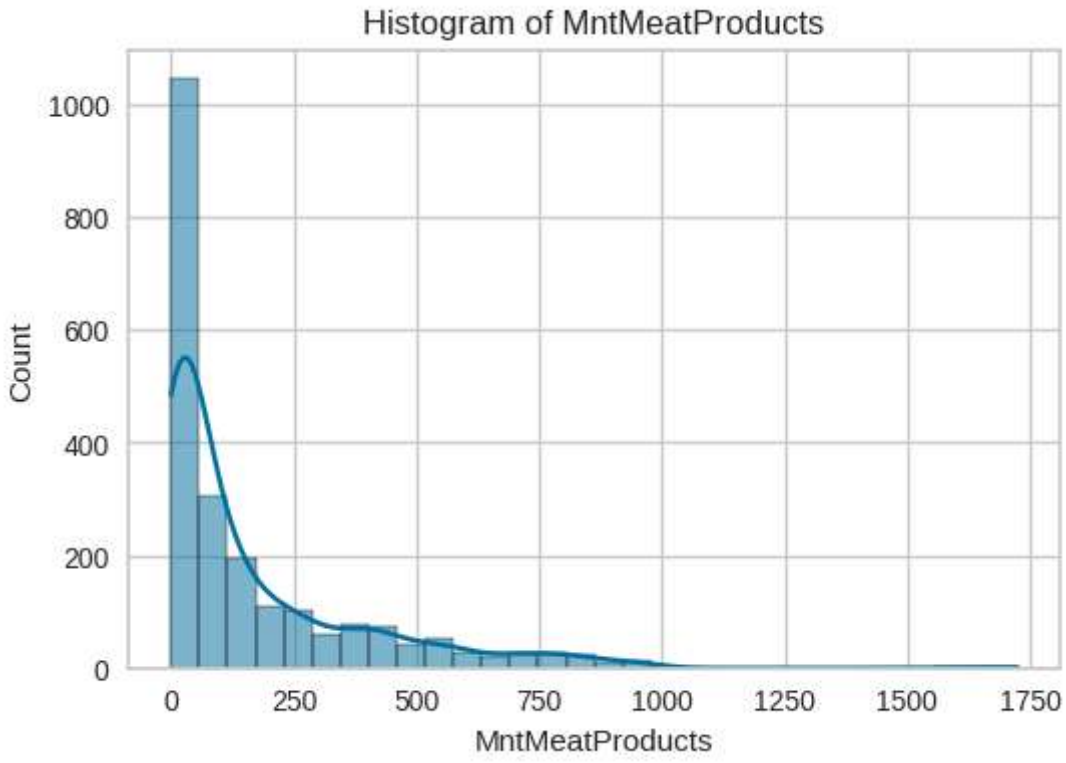


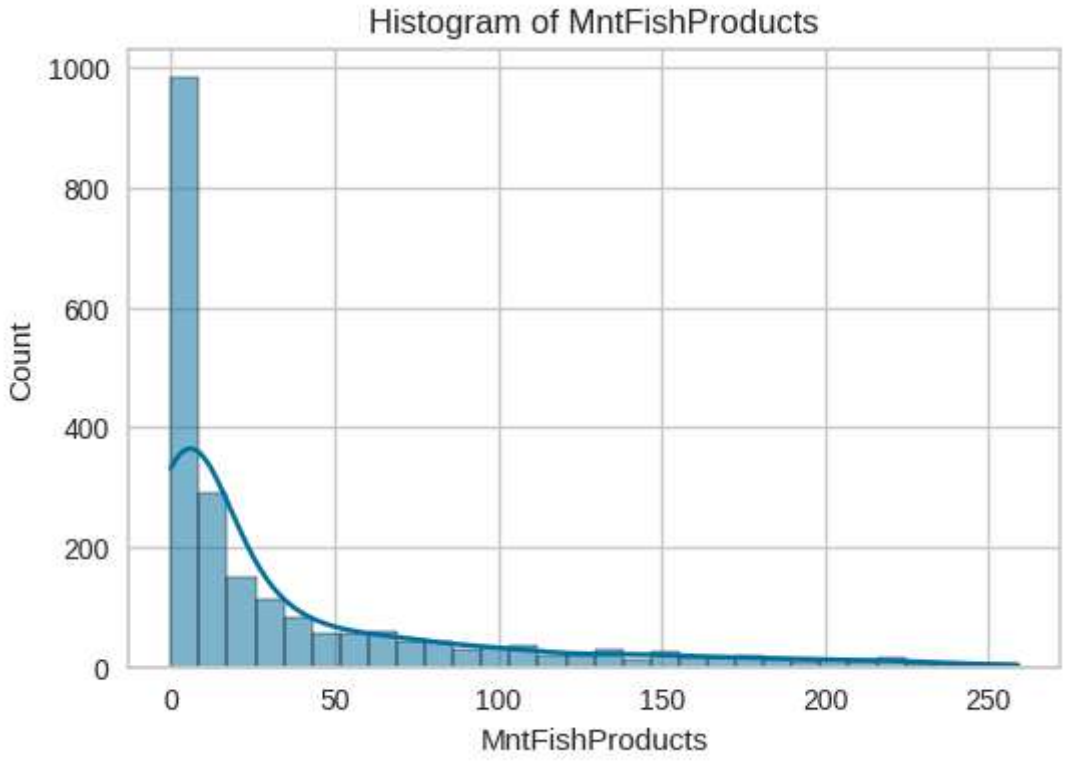

Histogram of Income
Count
Income
0
100
200
300
400
500
600
700
800
0
100000
200000
300000
400000
500000
600000

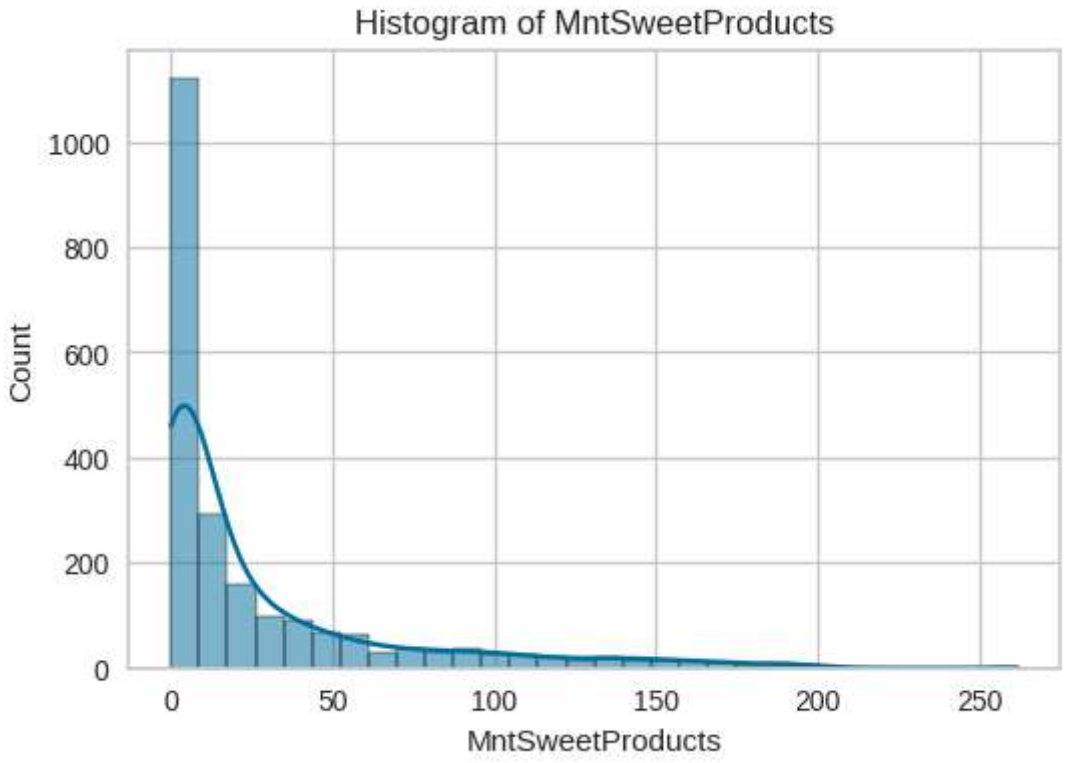

Histogram of Recency
Count
Recency
0
20
40
60
80
100
0
20
40
60
80
100

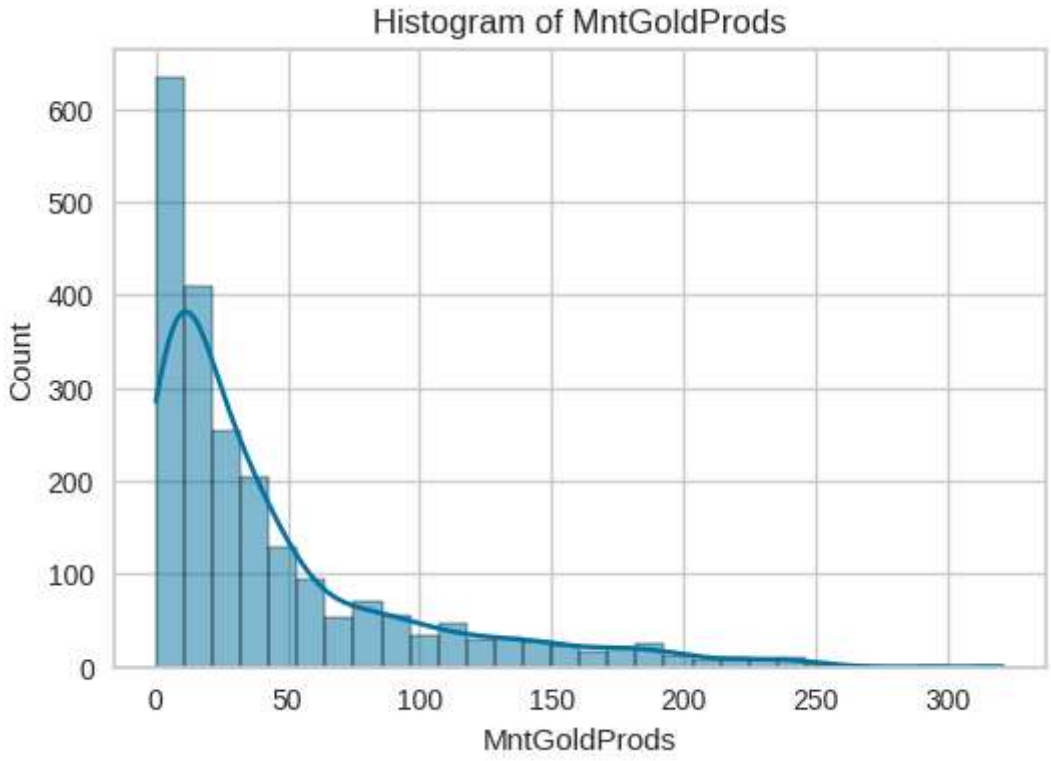

Histogram of MntWines
Count
MntWines
0
100
200
300
400
500
600
700
0
200
400
600
800
1000
1200
1400

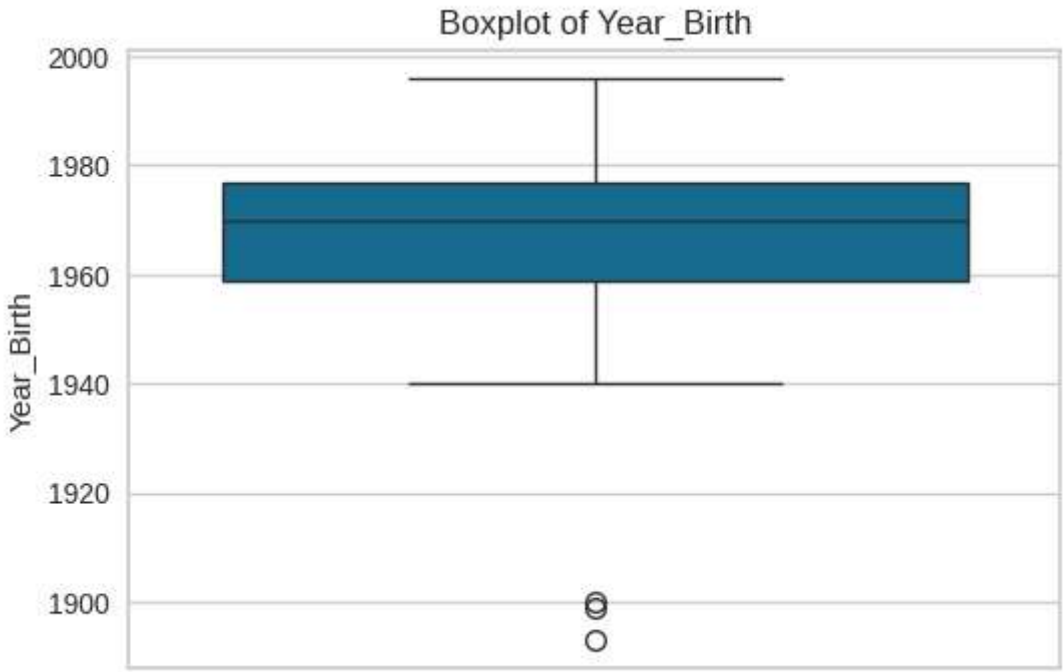

Histogram of MntFruits
Count
MntFruits
0
200
400
600
800
1000
0
25
50
75
100
125
150
175
200

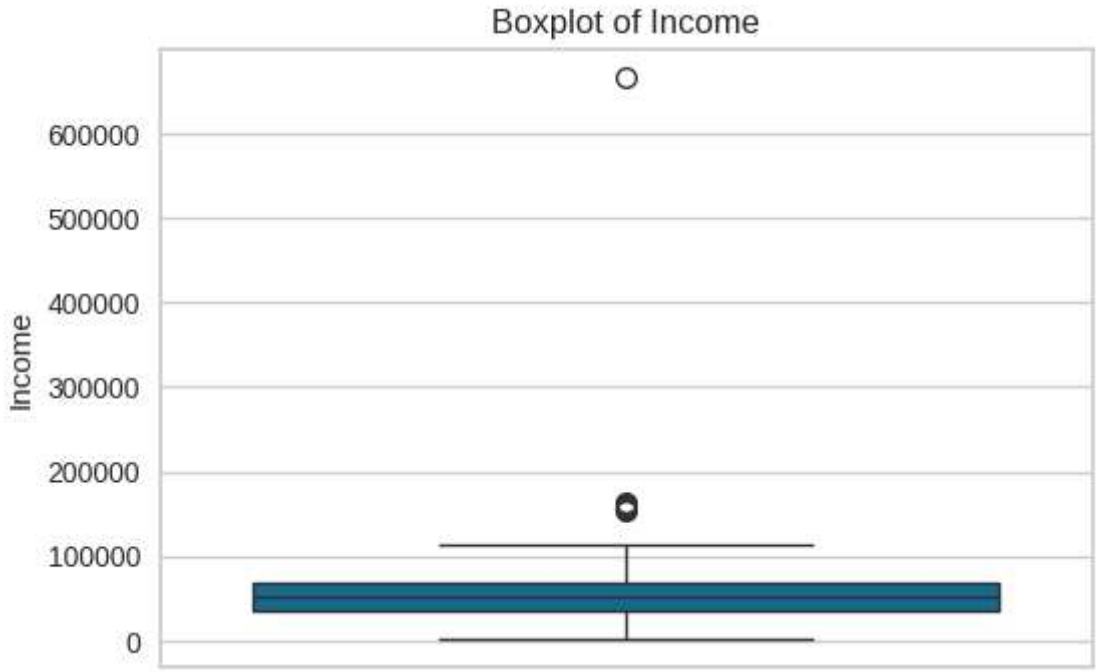

Histogram of MntMeatProducts
Count
MntMeatProducts
0
200
400
600
800
1000
0
250
500
750
1000
1250
1500
1750

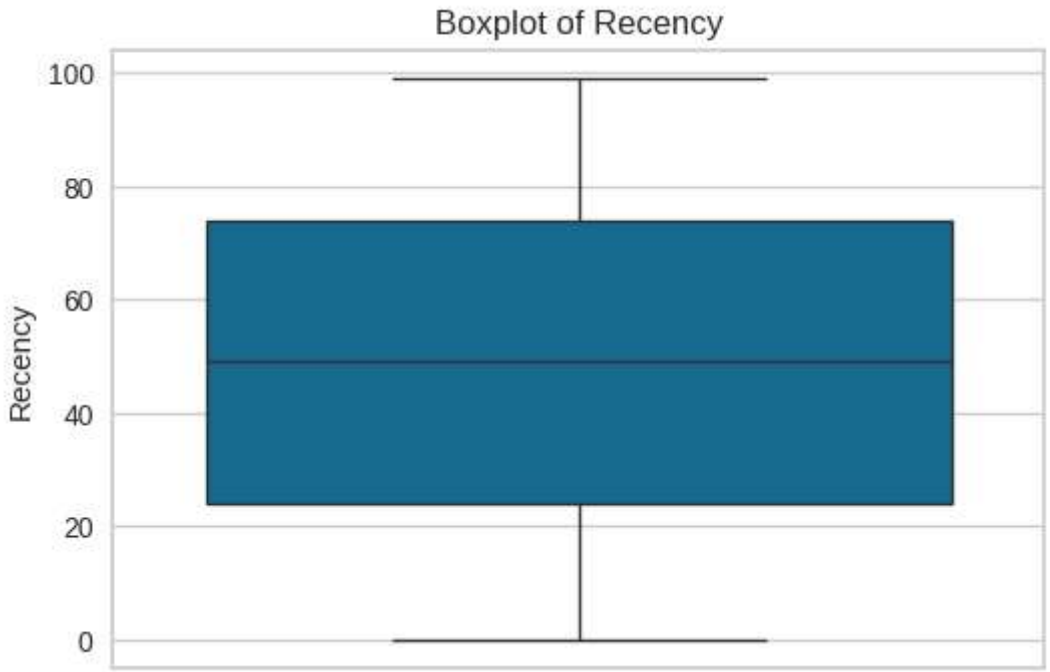

Histogram of MntFishProducts
Count
MntFishProducts
0
200
400
600
800
1000
0
50
100
150
200
250

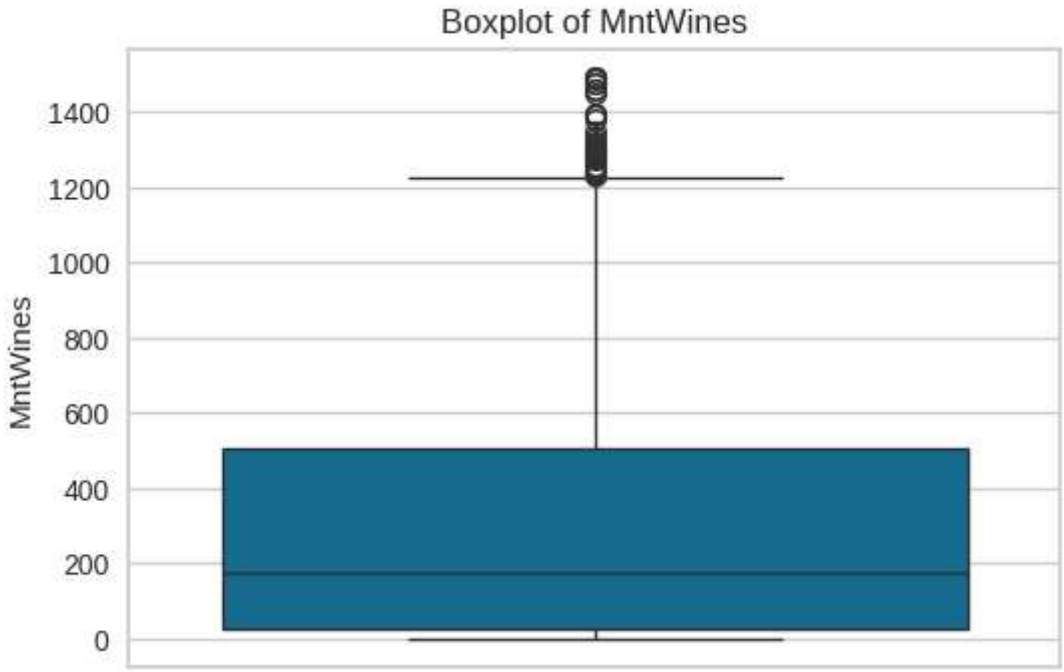

Histogram of MntSweetProducts
Count
MntSweetProducts
0
200
400
600
800
1000
0
50
100
150
200
250

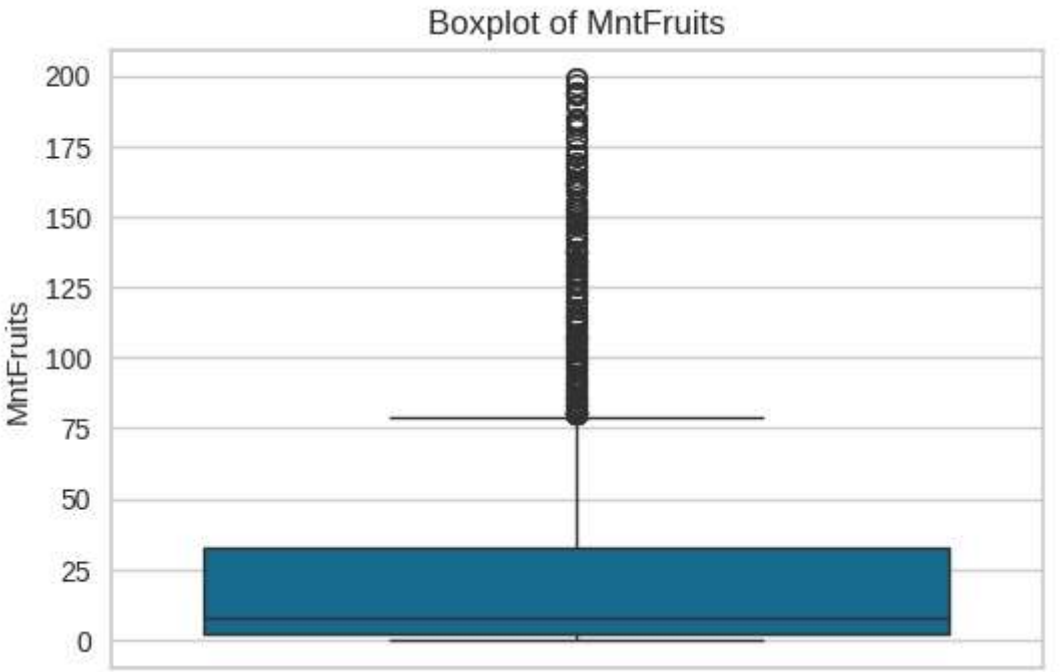

Histogram of MntGoldProds
Count
MntGoldProds
0
100
200
300
400
500
600
0
50
100
150
200
250
300

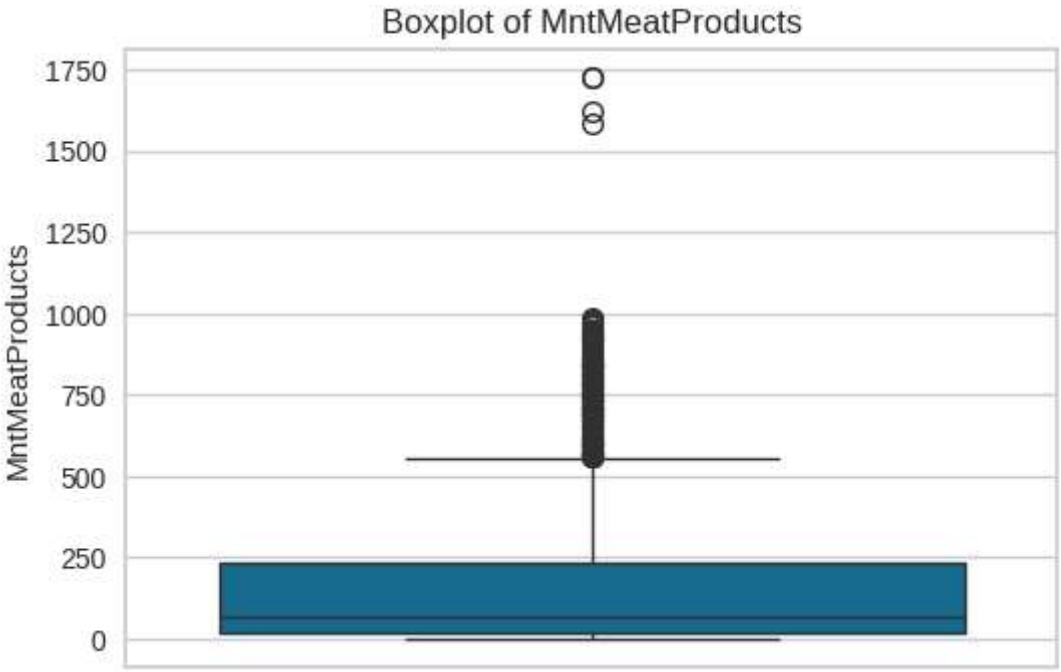

Boxplot of Year_Birth
Year_Birth
2000
1980
1960
1940
1920
1900

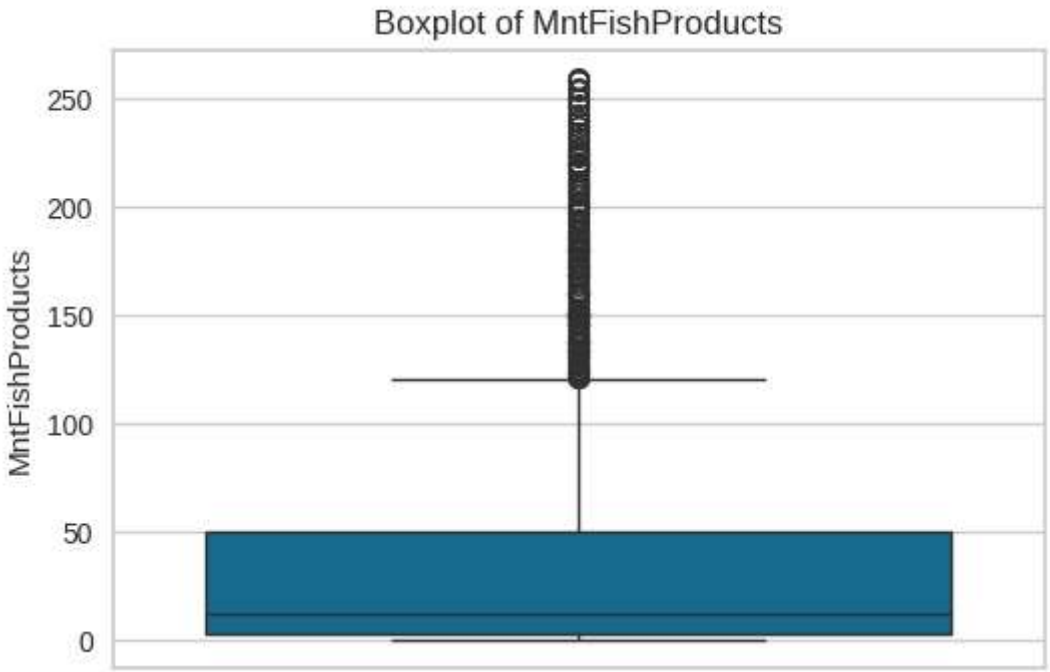

Boxplot of Income
Income
600000
500000
400000
300000
200000
100000
0

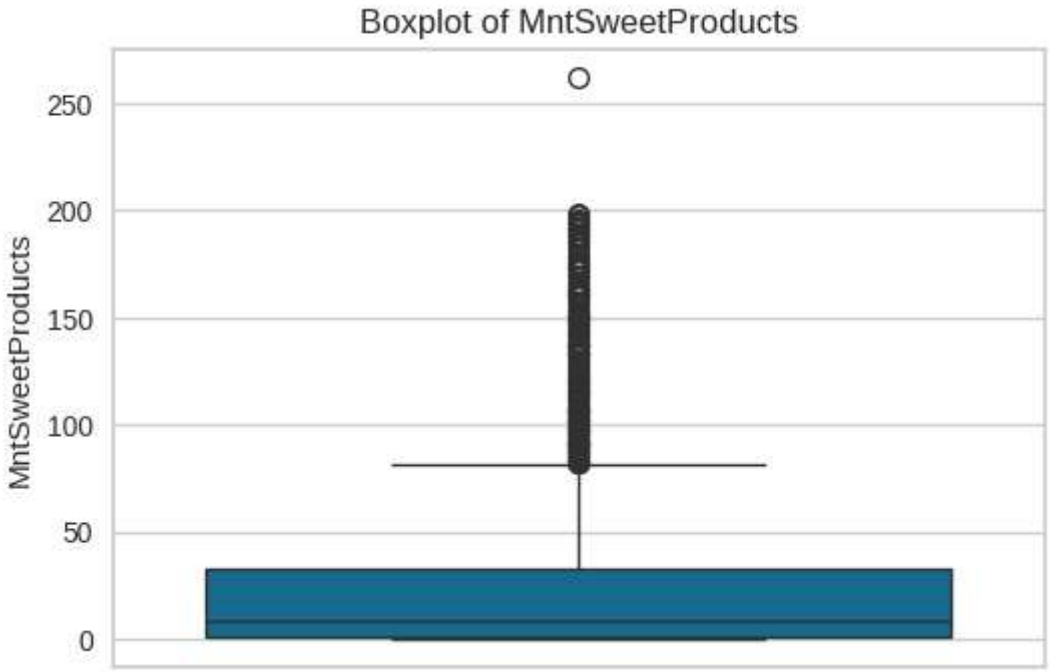

Boxplot of Recency
Recency
100
80
60
40
20
0

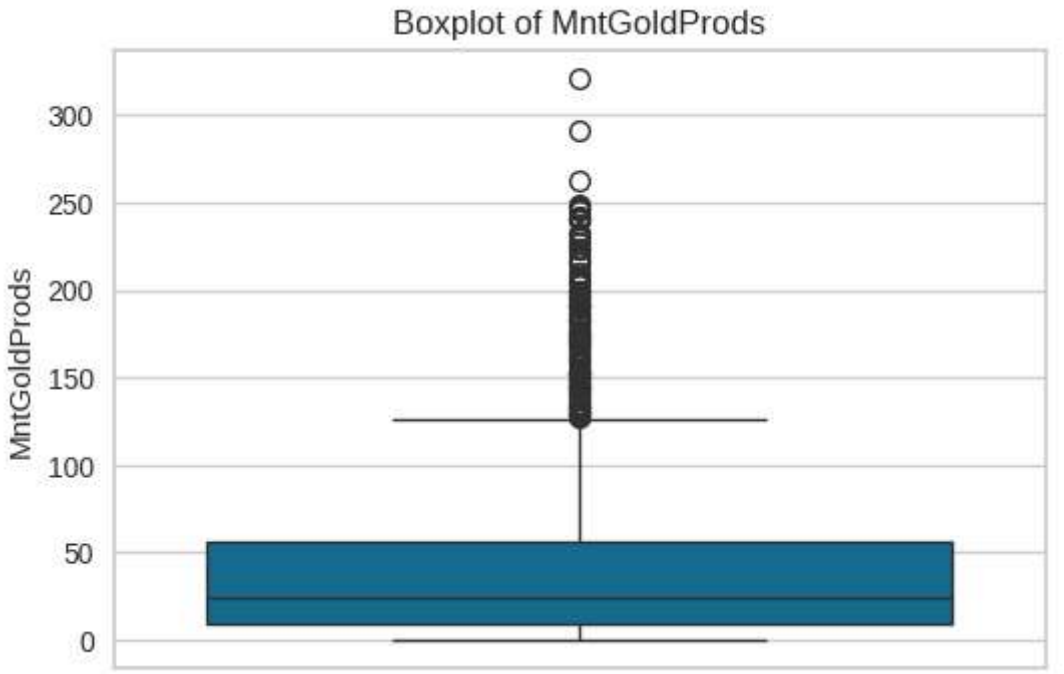

Boxplot of MntWines
MntWines
1400
1200
1000
800
600
400
200
0

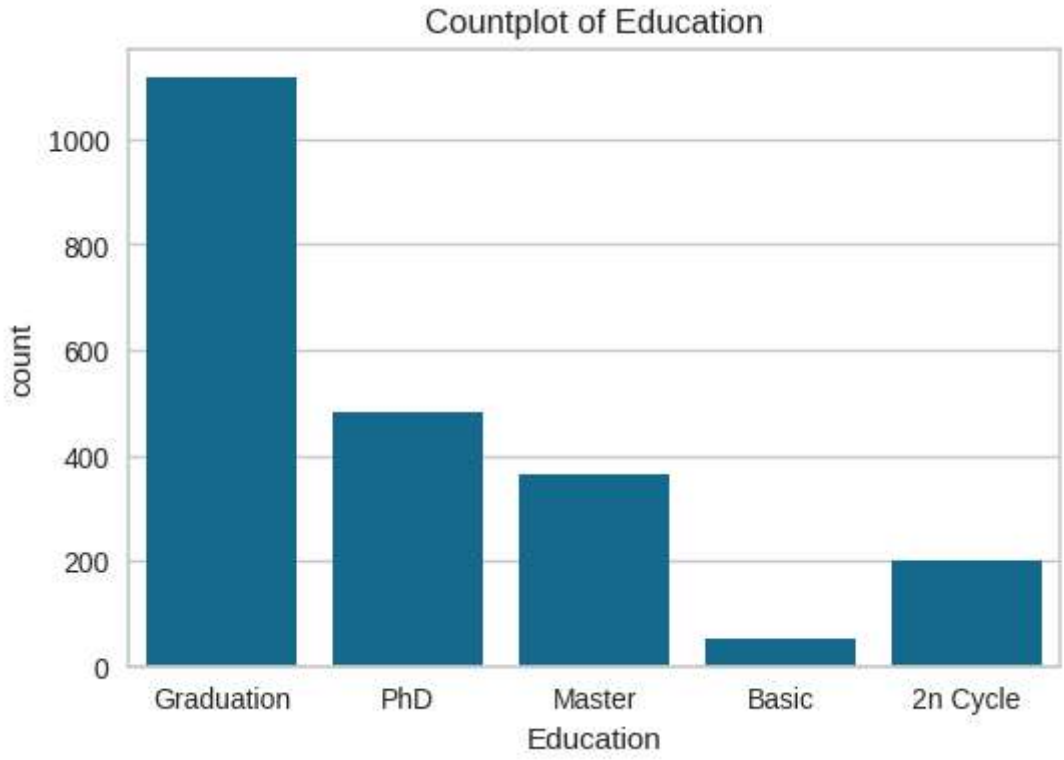

Boxplot of MntFruits
MntFruits
200
175
150
125
100
75
50
25
0

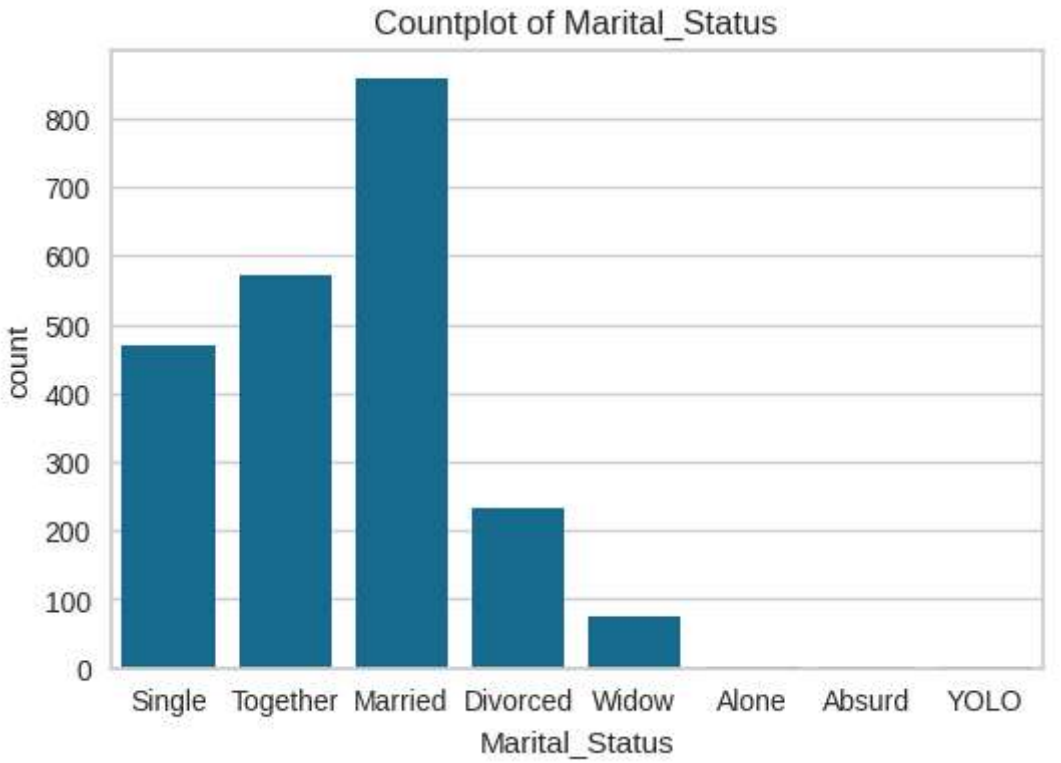

Boxplot of MntMeatProducts
MntMeatProducts
1750
1500
1250
1000
750
500
250
0

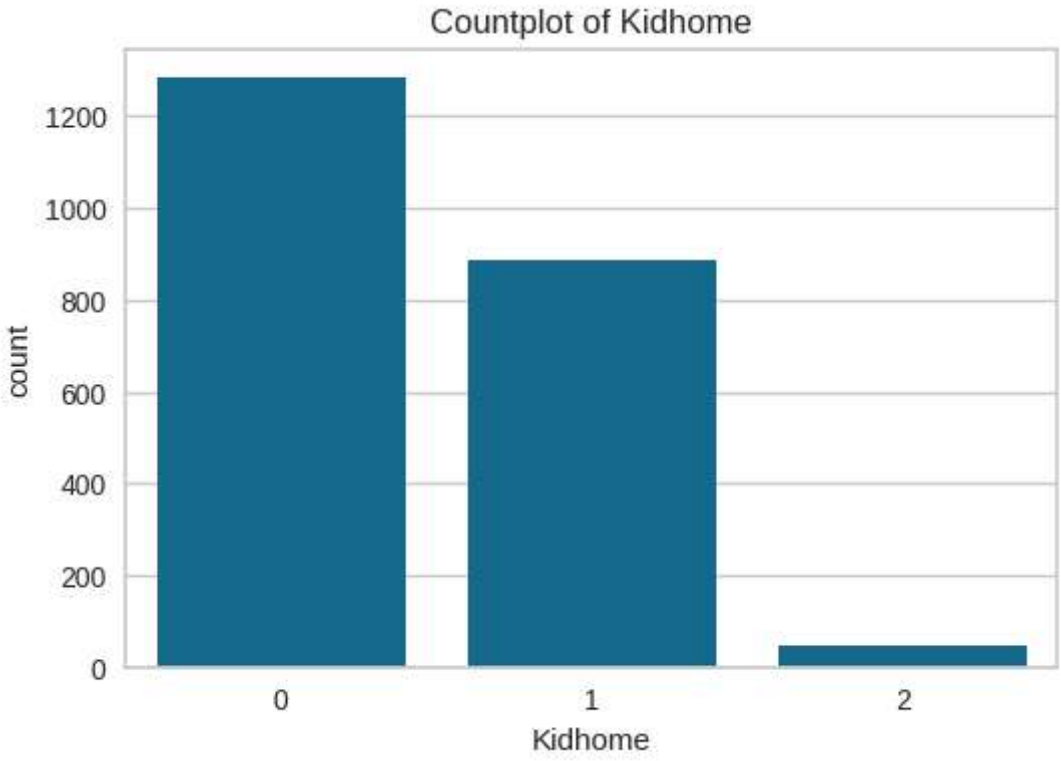

Boxplot of MntFishProducts
MntFishProducts
250
200
150
100
50
0

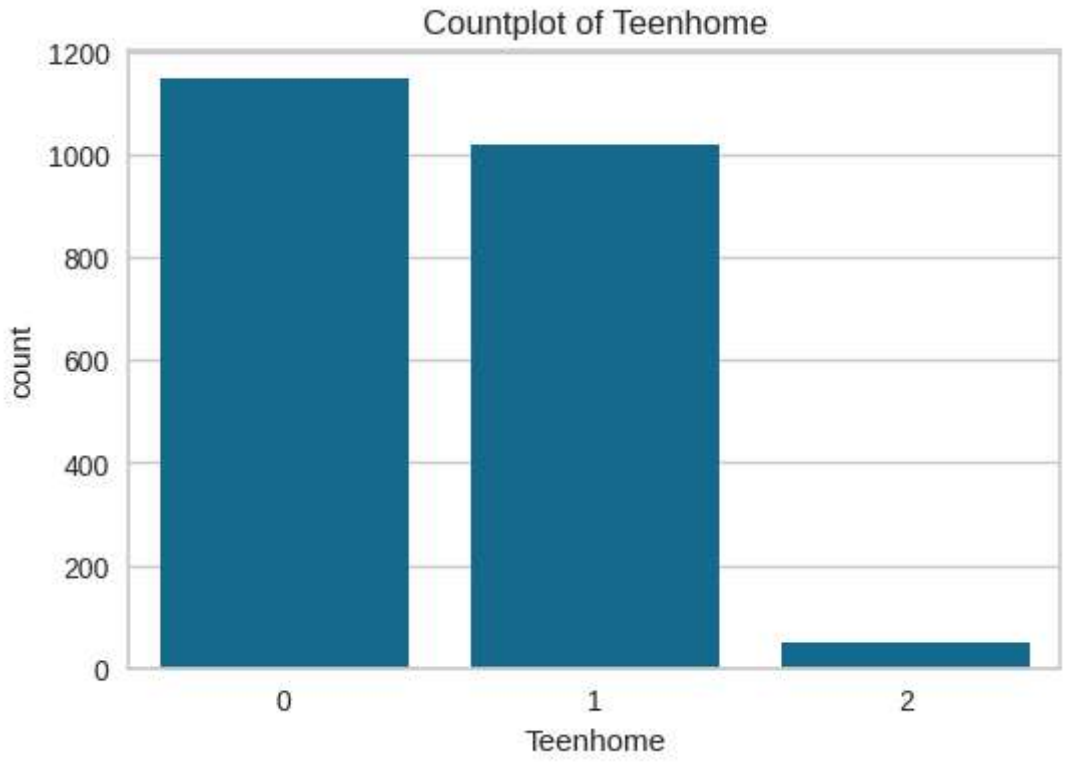

Boxplot of MntSweetProducts
MntSweetProducts
250
200
150
100
50
0

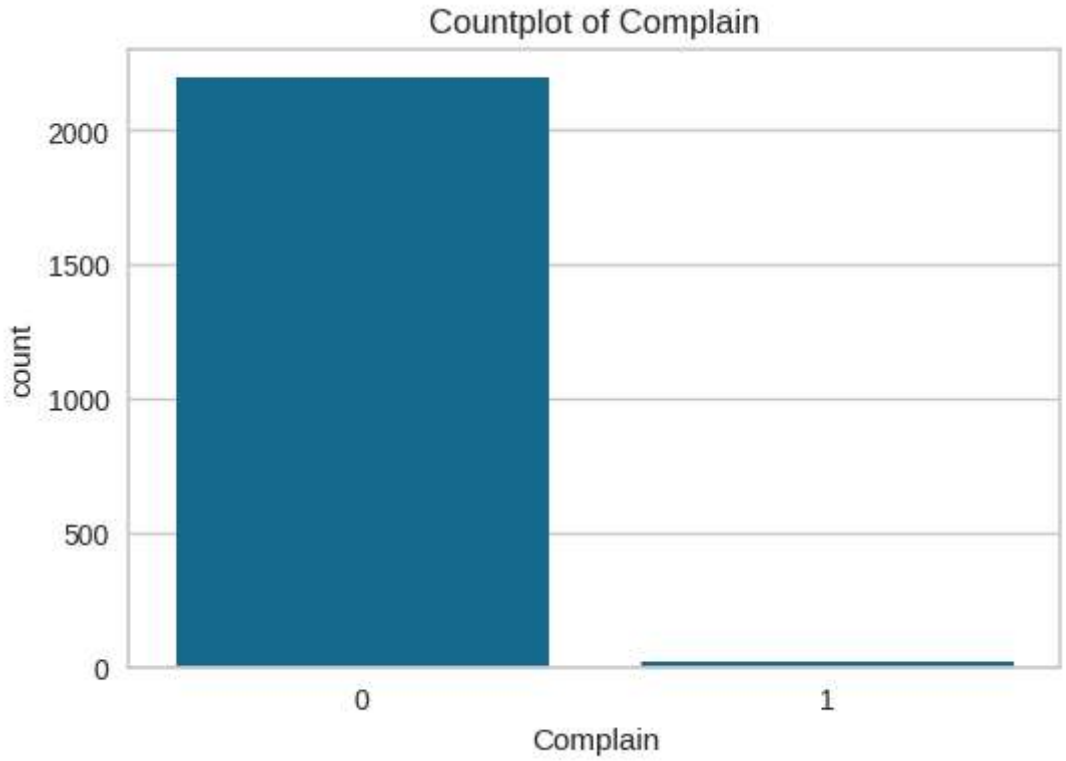

Boxplot of MntGoldProds
MntGoldProds
300
250
200
150
100
50
0

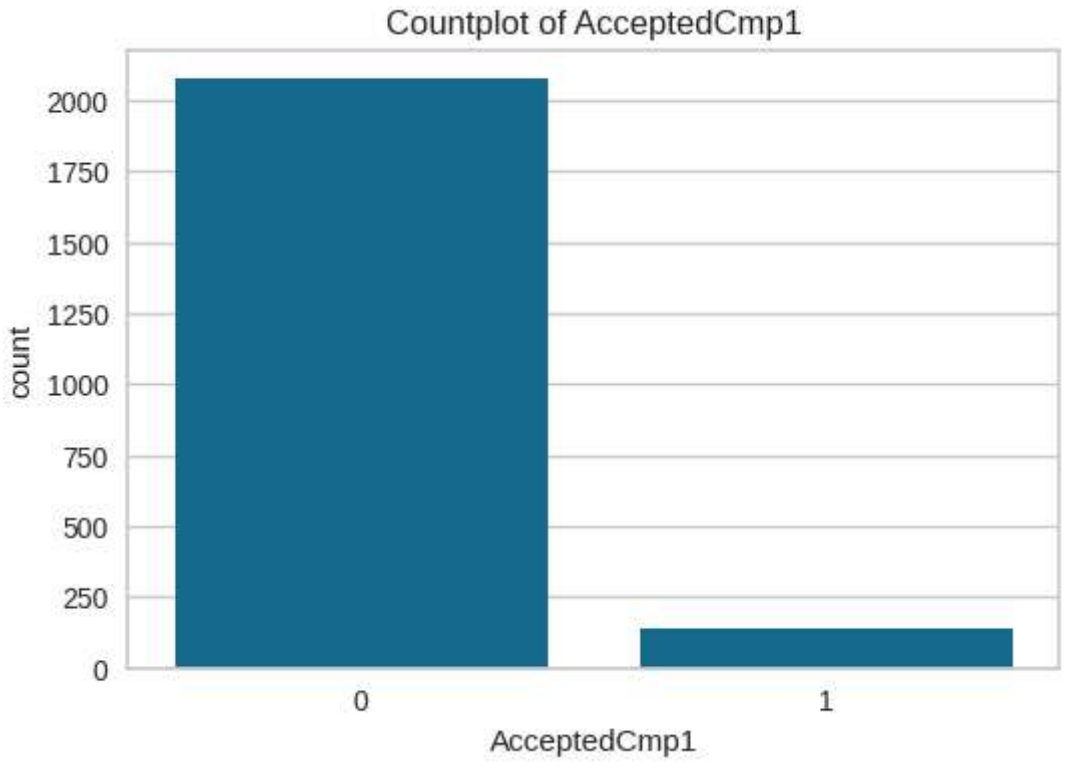

Countplot of Education
count
1000
800
600
400
200
0
Graduation
PhD
Master
Basic
2n Cycle
Education

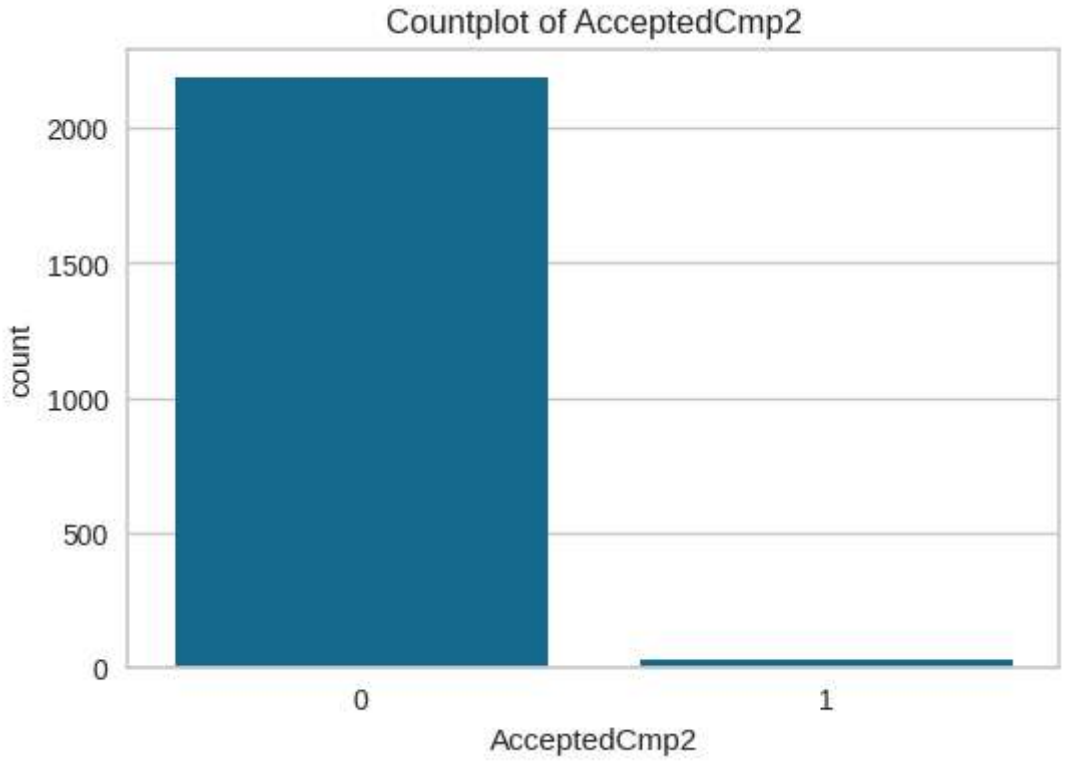

Countplot of Marital_Status
count
0
100
200
300
400
500
600
700
800
Single
Together
Married
Divorced
Widow
Alone
Absurd
YOLO
Marital_Status

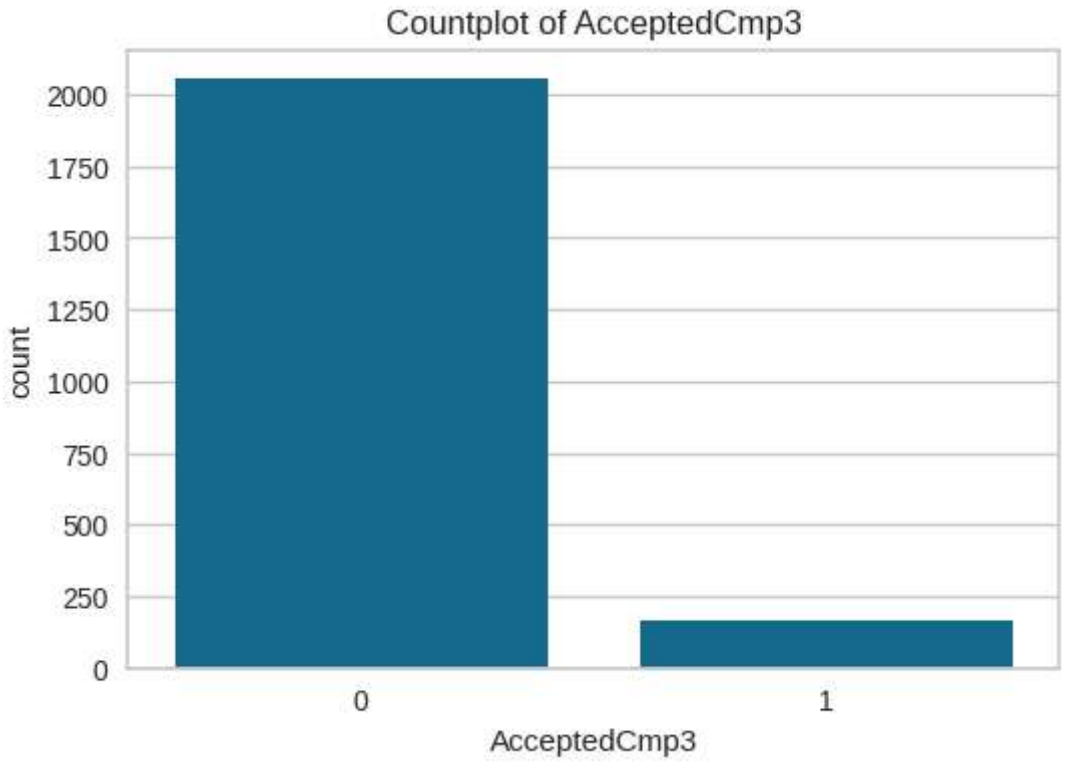

Countplot of Kidhome
count
0
200
400
600
800
1000
1200
0
1
2
Kidhome

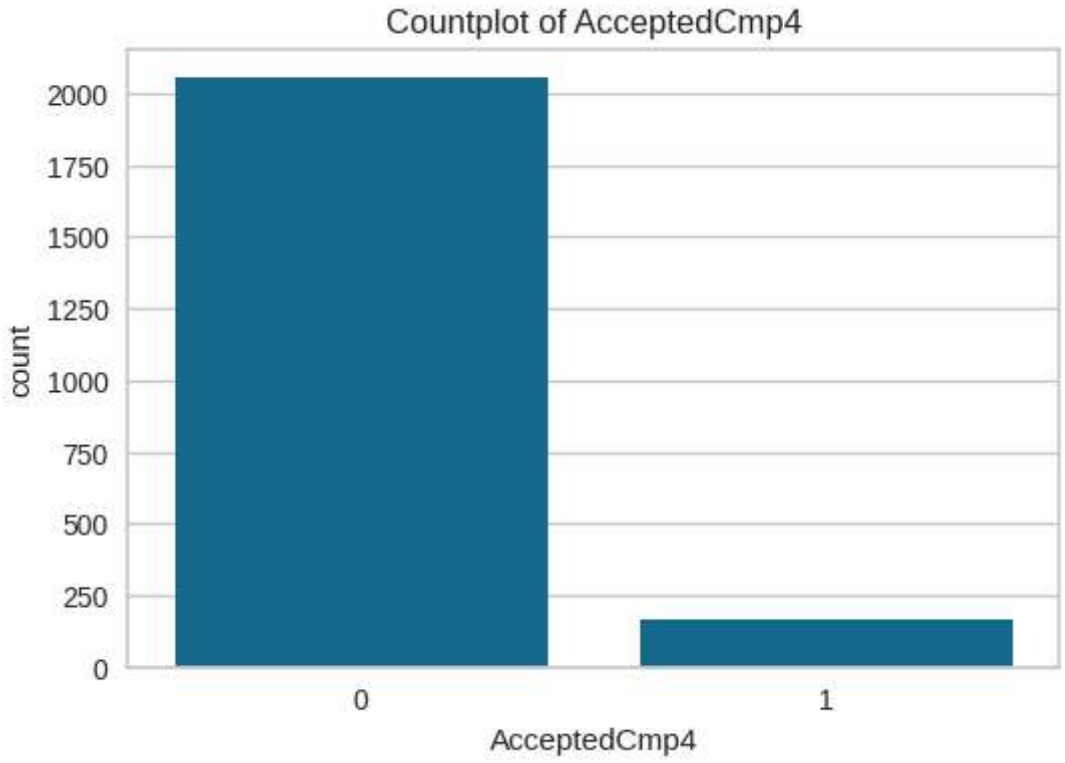

Countplot of Teenhome
1200
1000
800
600
400
200
0
count
0
1
2
Teenhome

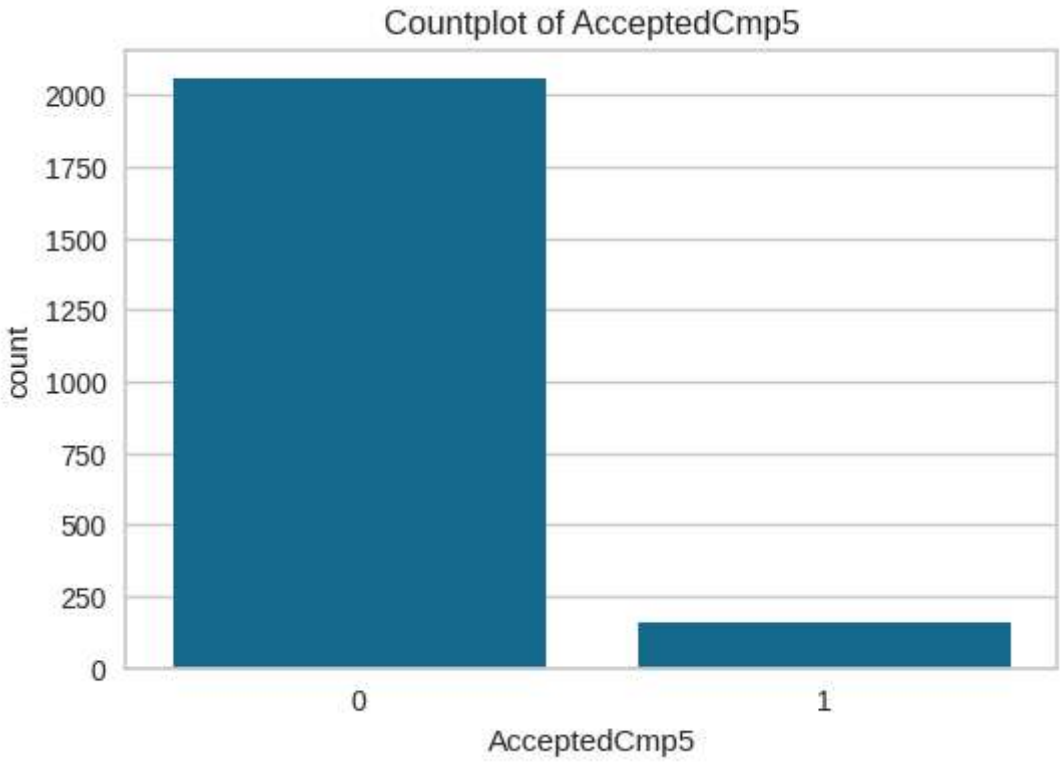

Countplot of Complain
2000
1500
1000
500
0
count
0
1
Complain

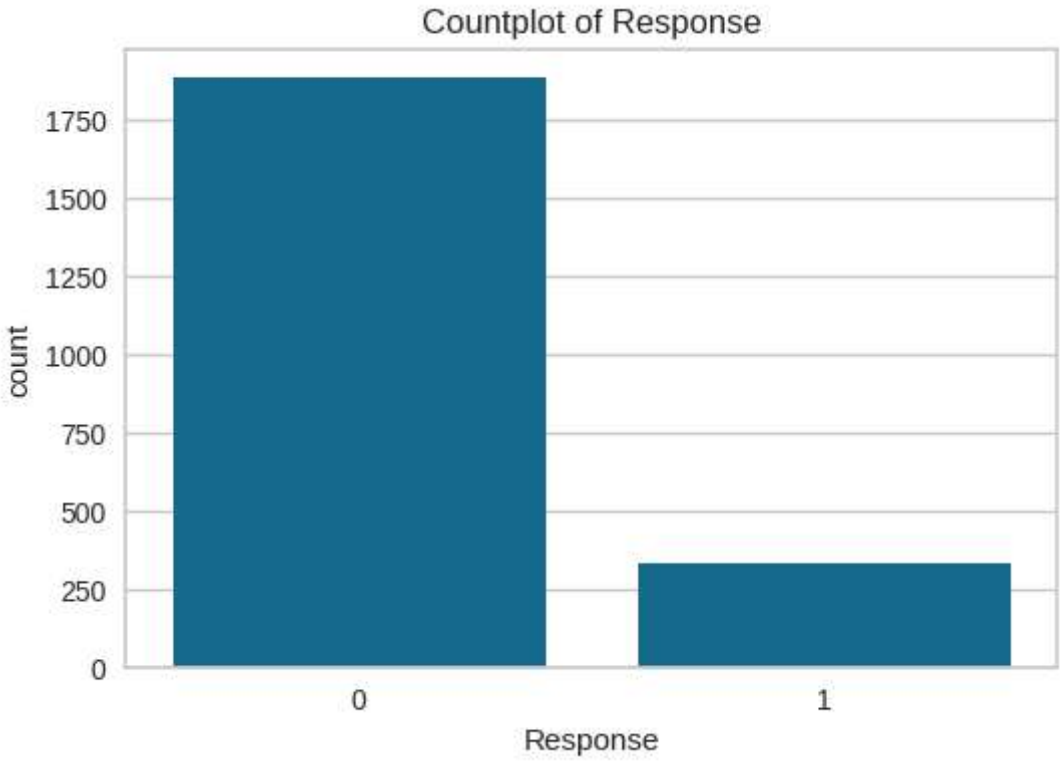

Countplot of AcceptedCmp1
count
2000
1750
1500
1250
1000
750
500
250
0
0
1
AcceptedCmp1


Countplot of AcceptedCmp2
count
2000
1500
1000
500
0
0
1
AcceptedCmp2


Countplot of AcceptedCmp3
count
2000
1750
1500
1250
1000
750
500
250
0
0
1
AcceptedCmp3


Countplot of AcceptedCmp4
count
2000
1750
1500
1250
1000
750
500
250
0
0
1
AcceptedCmp4




**Observations:**

Year_Birth: There are a few outliers near the 1900's and 1920's that should be omitted from the data since those people are probably no longer active customers of the company. Most customers were born between the 1950's and 1980's, peaking around the 1970's, with very few being bron before the 1940's or after the 2000's.

Income: This graph is heaviy right skewed, and would be clearer with a log-fit transformation. Majority of customers earn an annual income between 0 - 100,000 dollars.

Recency: This distribution apears faily uniform, with spikes near the ends and middle of the data range. A fairly uniform distribution would suggest that this feature does not follow a specific trend.

Mnt: Amount of money spent on specific products all form about the same right-skewed distribution pattern. All six of these graphs peak below the $250 amount spent per customer, with a lot of outliers spending various large amounts on each product.

Education: About half of the customers have a graduation level education, and the other half have a higher level of education. The "Basic" bin is so small it could be categorized with "graduation" to reduce the dimensionability of the data.

Martial_Status: Majoity of customers are either married or together, and some are single. The "Divorced", "Widow", "Alone", "Absurd", and "YOLO" bins are so small these could be categorized with "Single" to reduce the dimensionability of the data.

Kidhome/ Teenhome: About half of the customers have a kid at home, and majority of these kids are teenagers in the house.

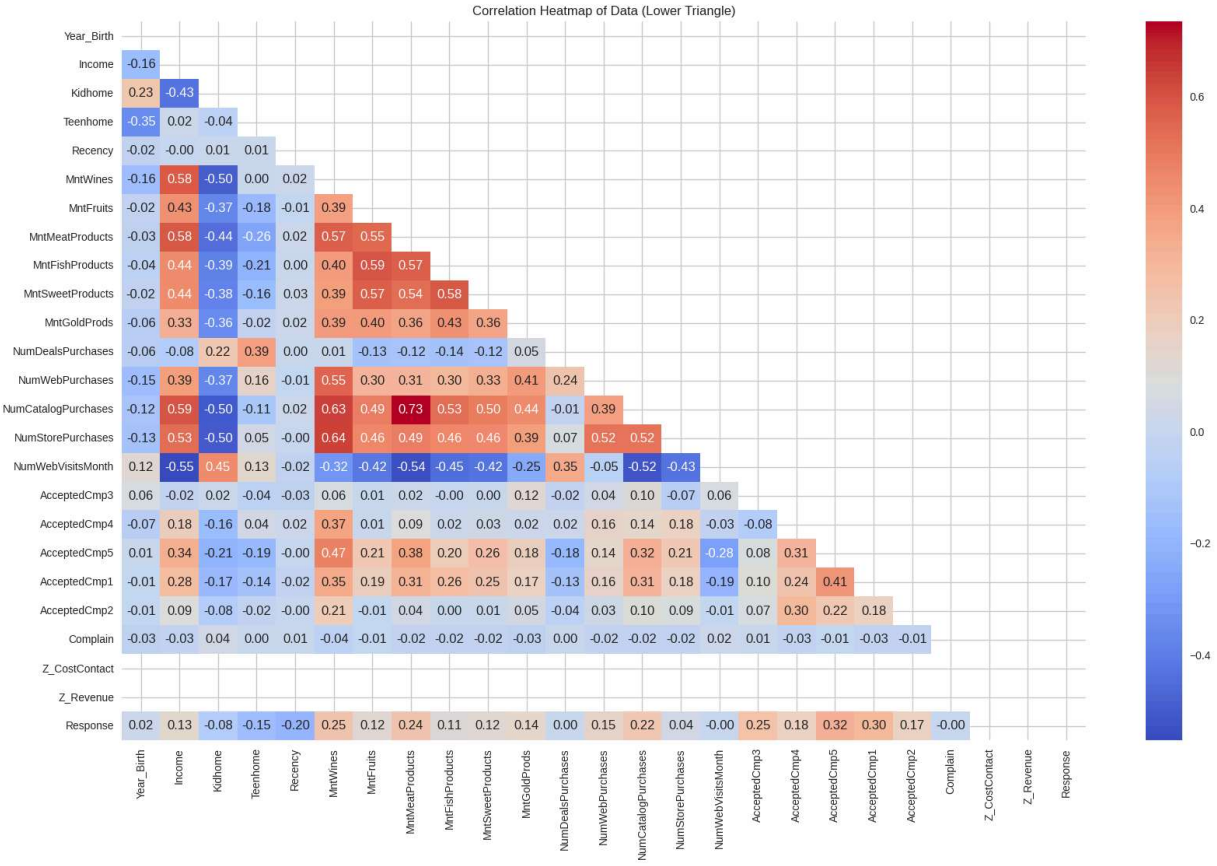
Complain: Majority of customers have not made a complaint.

Accepted: Majority of customers have not interacted with any of the six campaigns, but cutomer engagement does increase with each new campaign.

## Bivariate Analysis

### **Question 6:** Perform multivariate analysis to explore the relationsips between the variables.

In [11]:
```
# Starting with a heatmap of the features is helpful to visualize what features hav
corr_matrix = data.corr(numeric_only=True)
plt.figure(figsize=(20, 12))
mask = np.triu(np.ones(corr_matrix.shape, dtype=bool)) #this creates a mask for the

sns.heatmap(corr_matrix,
            mask=mask, #only shows bottom triangle
            annot=True, #annot being toggled shows the correlation coeficients in t
            fmt=".2f", #formats to 2 decimal places
            cmap="coolwarm")

plt.title("Correlation Heatmap of Data (Lower Triangle)")
plt.show()
```



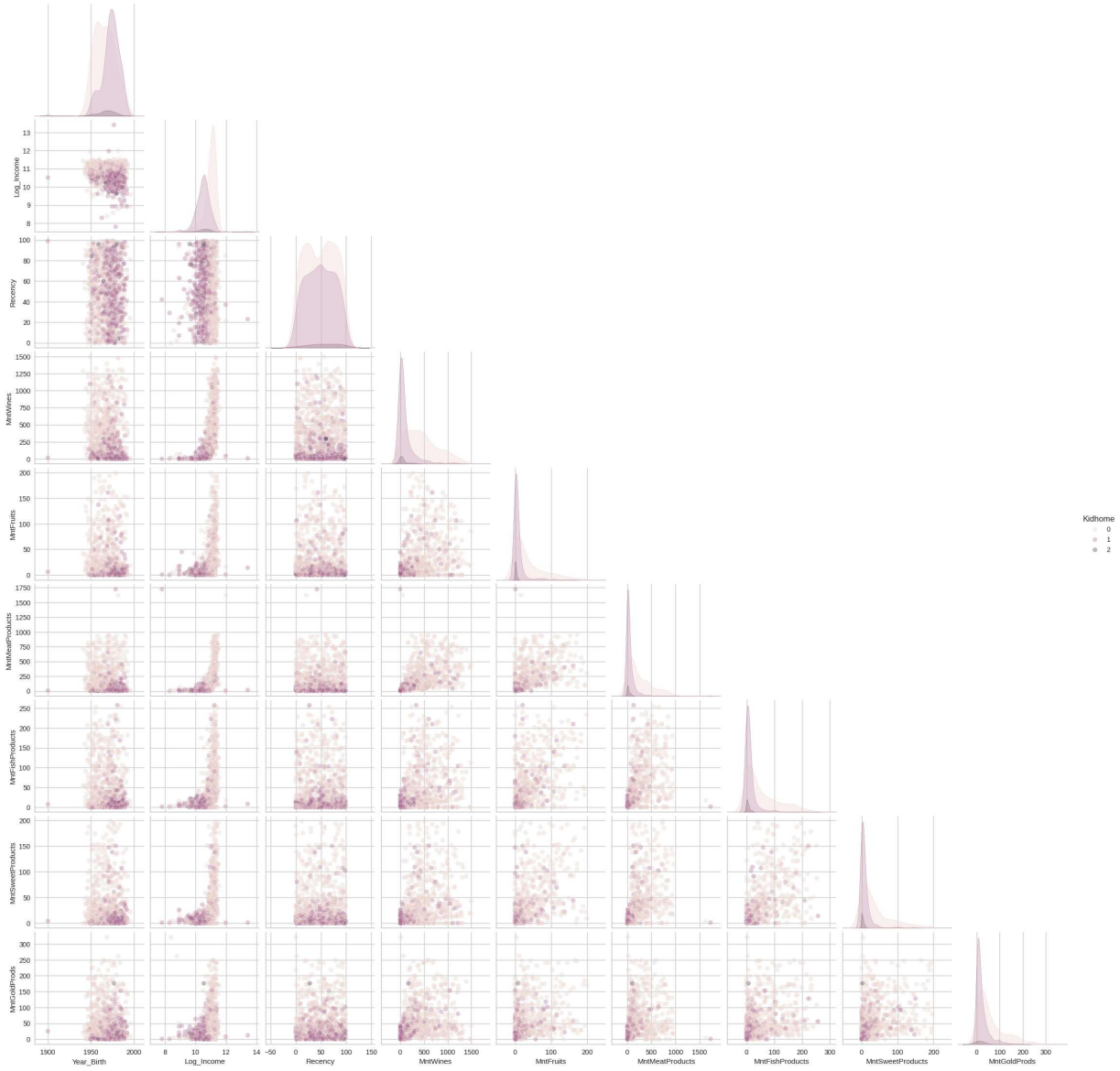

Heatmap Observations: ** above 0.5 is a strong correlation, between 0.3 and 0.5 is moderate, and below 0.3 is weak

1. Income has a strong positive correlation with MntWines, MntMeatProducts, NumCatalogPurhases, and NumstorePurchases (r>0.5), and similar levels with other purchasing related categories such as MntFruits and NumWebPurchases.
2. NumCatalogPurchases, NumStorePurchases, and NumWebPurchases all have strong positive correlations with eachother, which suggests that customers who engage in one type of channel probably also engage with others.
3. NumCatalogPurchases show the strongest correlation with amount spent on products, but a strong negative correlation with NumDealsPurchases. Catalog optimization and increased catalog deals might yield the best ROI.
4. Amount spent on one product (such as MmtMeatProducts) have strong correlations with other products. This may indicate the presense of 'premium' shoppers who make large purchases of a variety of products.
5. NumWebVisitsMonth has a strong negative correlation with the amount of products bought for all products, possibly due to indecision or abandoned carts. This indicates that web-based visits are not translating to purchases, suggesting a need for optimized checkout UX or retargeting efforts.
6. AcceptedCmp categories have strong correlations with purchasing categories and other campaign categories, but negative correlations with NumDealsPurchased, NumWebVisitsMonth, and Kidhome and Teenhome. This could suggest that campaigns are only reaching a limited number of customers through one type of media, and

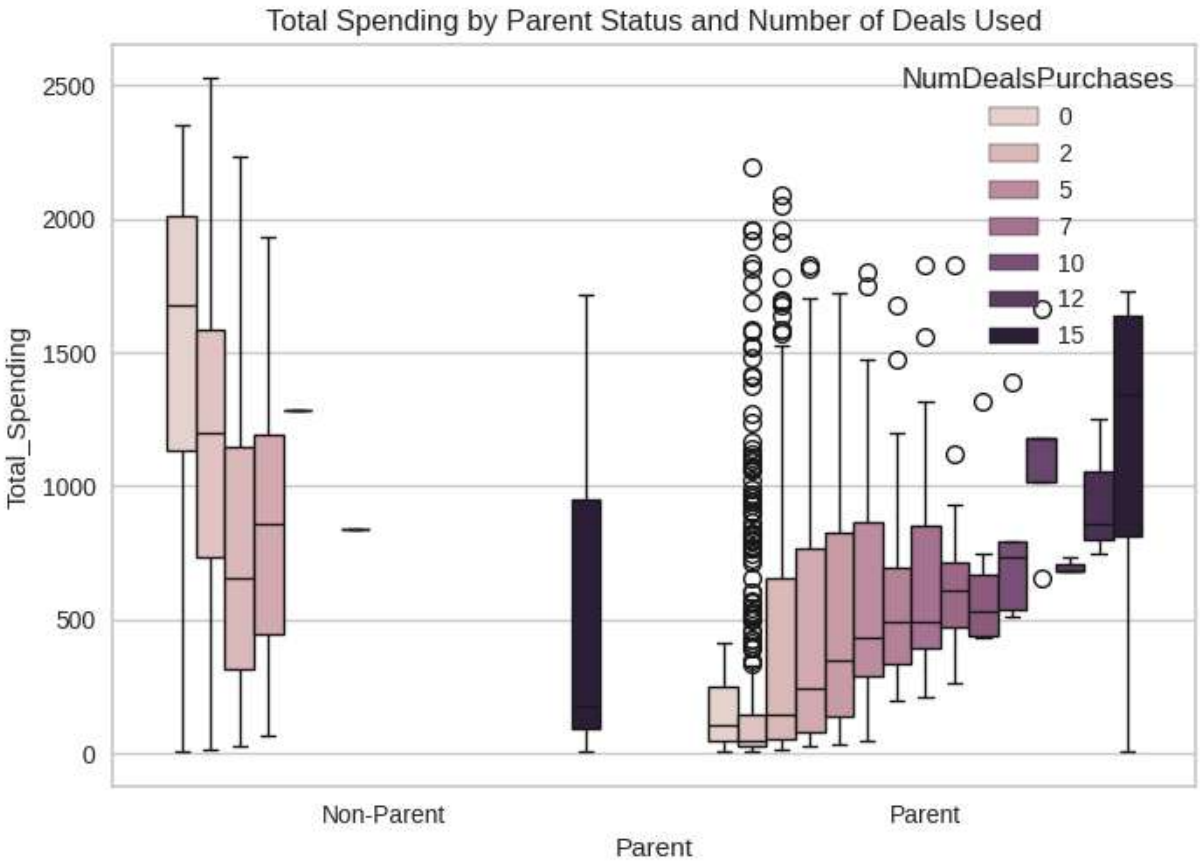
indicate that customers with extra responsibility at home have limited time to watch media for campaigns.

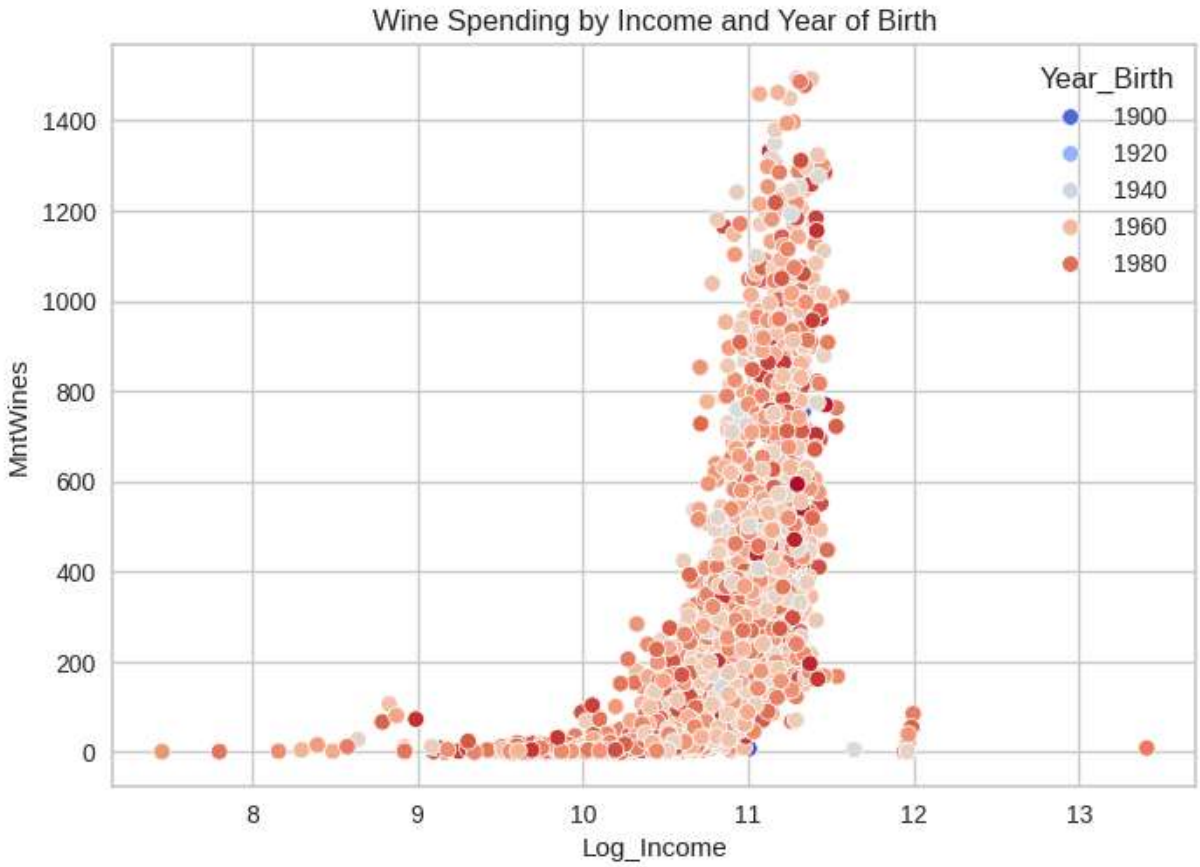
7. Kidhome and Teenhome have strong negative correlations with almost every category, besides NumDealsPurchases and NumWebVisitsMonth. This suggests it's possible that customers with extra responsbility at home are looking to save money and time by shopping online with deals.
8. YearBirth only has moderate correlations with Kidhome, Teenhome, and Income.
9. Interestingly, Complain has weak correlations wth all other categories. This pattern may suggest that complaints are a random, sporadic event rather issues with spending habits.
10. Response is moderately correlated with past campaign acceptances, suggesting a customer's history of engagement is predictive of future responses.

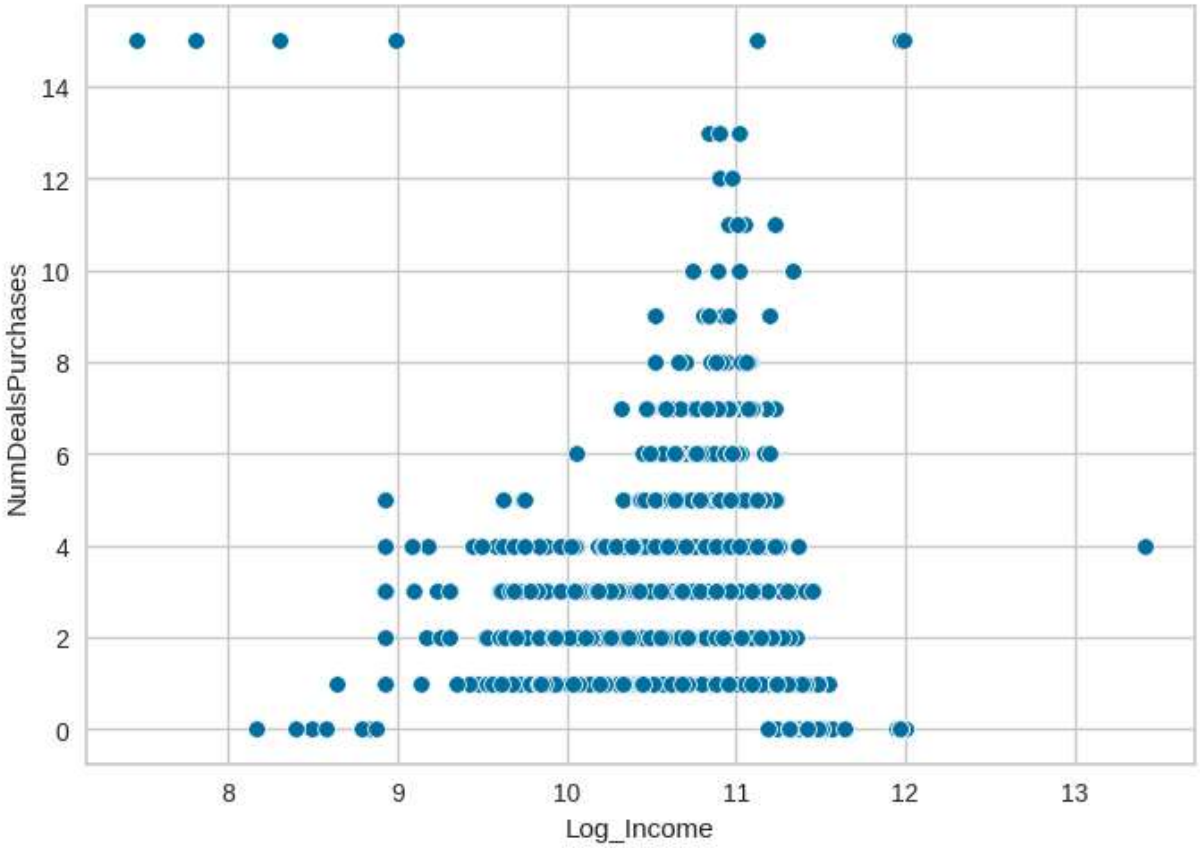
In [12]:
```
#pairplots of all numerical (excluding binary or discrete data) features shows what
data['Log_Income'] = np.log1p(data['Income'])
numerical_columns = ['Year_Birth', 'Log_Income', 'Recency', 'MntWines', 'MntFruits'
data_sample = data.sample(n=1000, random_state=42) #take only a sample of the data
sns.pairplot(data=data_sample[numerical_columns + ['Kidhome']], hue='Kidhome', corn
plt.show()
```




In [13]:
```
#Going back to the observations of the heatmap, creating different types of plots c
#1. Do parents spend more or less, and do deals affect this?
data['Parent'] = data['Kidhome'] + data['Teenhome'] #add Kidhome and Teenhome colum
data['Parent'] = data['Parent'].apply(lambda x: 'Parent' if x>0 else 'Non-Parent')

product_cols = ['MntFishProducts', 'MntMeatProducts', 'MntGoldProds', 'MntFruits',
data['Total_Spending'] = data[product_cols].sum(axis=1) #sum all of the amount spen

sns.boxplot(x='Parent', y='Total_Spending', hue='NumDealsPurchases', data=data)
plt.title('Total Spending by Parent Status and Number of Deals Used')
plt.show()

#2. Do income and year of birth affect wine buying habits?
sns.scatterplot(x=data['Log_Income'], y=data['MntWines'], hue=data['Year_Birth'], p
plt.title('Wine Spending by Income and Year of Birth')
plt.show()

#3 Does income affect how many deals customers use?
sns.scatterplot(x=data['Log_Income'], y=data['NumDealsPurchases'])
plt.show()
```


Total Spending by Parent Status and Number of Deals Used
NumDealsPurchases
0
2
5
7
10
12
15
Total_Spending
2500
2000
1500
1000
500
0
Non-Parent
Parent
Parent


Wine Spending by Income and Year of Birth
Year_Birth
1900
1920
1940
1960
1980
MntWines
1400
1200
1000
800
600
400
200
0
8
9
10
11
12
13
Log_Income




1. The first graph shows the amount of money spent by parents vs. non-parents, categorized by deals used through different colors. Non-parents spend more than parents regardless if they use deals, which most do not use deals. For parents, there is a positive relation between deals used and money spent. As the number of deals used increases, so does the amount of money spent by parents. These observations may suggest that non-parents buy more luxury or high-ticket items, while parents are more deal-conscience and spend more money with more available deals, which may indicate buying in bulk while deals are available. Marketing strategy suggestions may include targeting parents with deal-based promotions, or targeting non-parents with premium product advertising.
2. The second graph shows the amount spent on wine versus income level, with data points colored by year of birth. The general diagonal trend from the bottom left to the top right indicates that as income increases, so does wine spending. However, there's a wide variation in wine spending at higher income levels, suggesting that while income and wine spending are positively correlated, higher income doesn't guarantee increased wine purchases. Notably, the darker red data points - representing younger customers - tend to appear higher on the y-axis. This suggests that younger individuals are more likely to spend more on wine at similar income levels compared to older generations.
3. The third graph shows the income of customers versus how many deals they use when purchasing products. For lower income customers there are two ends of the spectrum; they either use no deals, or they use a lot of deals. This may suggest that those with a lower income do not spend money on products unless there are deals available. For

others, while there is a general positive correlation between income and deals used, there is also a lot of variation, which indicates that as income increases there is no garantee of more deals used by customers.

**Observations:**

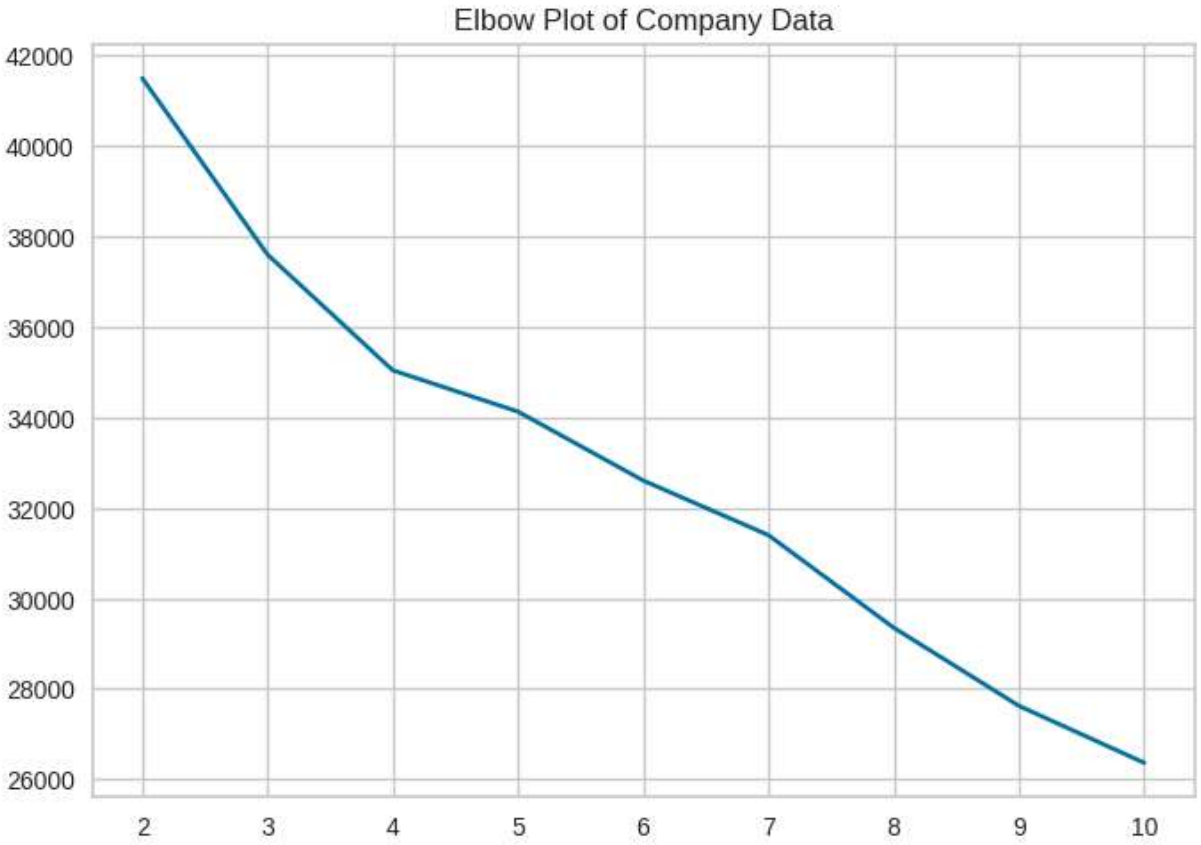
Personality traits that affect spending from observations of multi-variate analysis include:

1. If the customer is a parent or not affects what items they buy, how many deals they use, and how often they purchase items, and what channel they use to buy products due to limited or plentyful time.
2. Age of the customer affects what products they buy and how much they are willing to spend on each item.

# K-means Clustering

### Question 7 : Select the appropriate number of clusters using the elbow Plot. What do you think is the appropriate number of clusters?

In [14]:
```
data_copy = data.copy() #before transforming or fitting data it is a good idea to m
data_copy_numerical = data_copy.select_dtypes(include=['number']) #KMeans does not
scaler = StandardScaler() #standarized the data since this is easier for KMeans to
data_scaled = scaler.fit_transform(data_copy_numerical)
df = pd.DataFrame(data_scaled, columns=data_copy_numerical.columns) #fit_transform
df_copy = df.copy() #make another copy for KMeans purpose, in order to save the sta
WCSS = [] #initialize a list for the Within-Cluster Sum of Squares, which will be p

silhouette = [] #for question 8, initialize a list to keep the score for each numbe

#the "range is not callable" error kept popping up, so this resets the range functi
clusters = range(2,11)

for n in clusters: #the for loop applies KMeans to the dataset using each number of
  kmeans = KMeans(n_clusters=n, random_state=1)
  kmeans.fit(df_copy)
  WCSS.append(kmeans.inertia_)
  labels = kmeans.labels_
  silhouette.append(silhouette_score(df_copy, labels))

plt.plot(clusters, WCSS)
plt.title('Elbow Plot of Company Data')
plt.show()
#k optimal = 4 clusters
```




**Observations:**

The optimal number of clusters is k=4 since the slope of the elbow curve is steep from k=2 to k=4, then levels off.

## Question 8 : finalize appropriate number of clusters by checking the silhoutte score as well. Is the answer different from the elbow plot?

In [15]:
```
optimal_k = silhouette.index(max(silhouette)) + 2
print(optimal_k)
#according to the silouhette score, the optimal k is 2 clusters, which is different
```

2

**Observations:**

According to the silouhette score, the optimal k is 2 clusters, which is different from the elbow plot. The Elbow plot suggests the "ideal" clusters, but the silhouette score suggests the number of clusters that are the most seperated. When plotted, the PCA graph with 2 clusters visually looked the same as 4 clusters, so for clarity and uniformity, 2 clusters will be used.

## Question 9: Do a final fit with the appropriate number of clusters. How much total time does it take for the model to fit the data?

In [16]:
```
kmeans_2 = KMeans(n_clusters=2, random_state=0)
```

```
kmeans_2.fit(df_copy)
```

Out[16]:

KMeans

```
KMeans(n_clusters=2, random_state=0)
```

**Observations:**

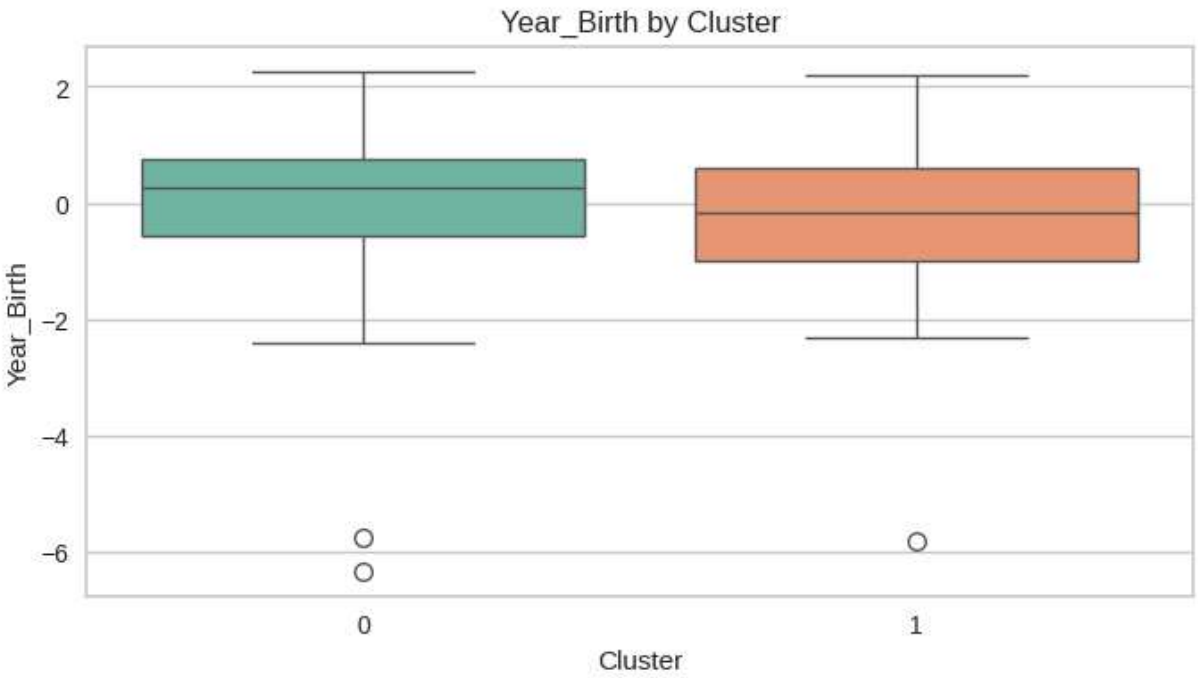
It took zero seconds for the model to fit the data.

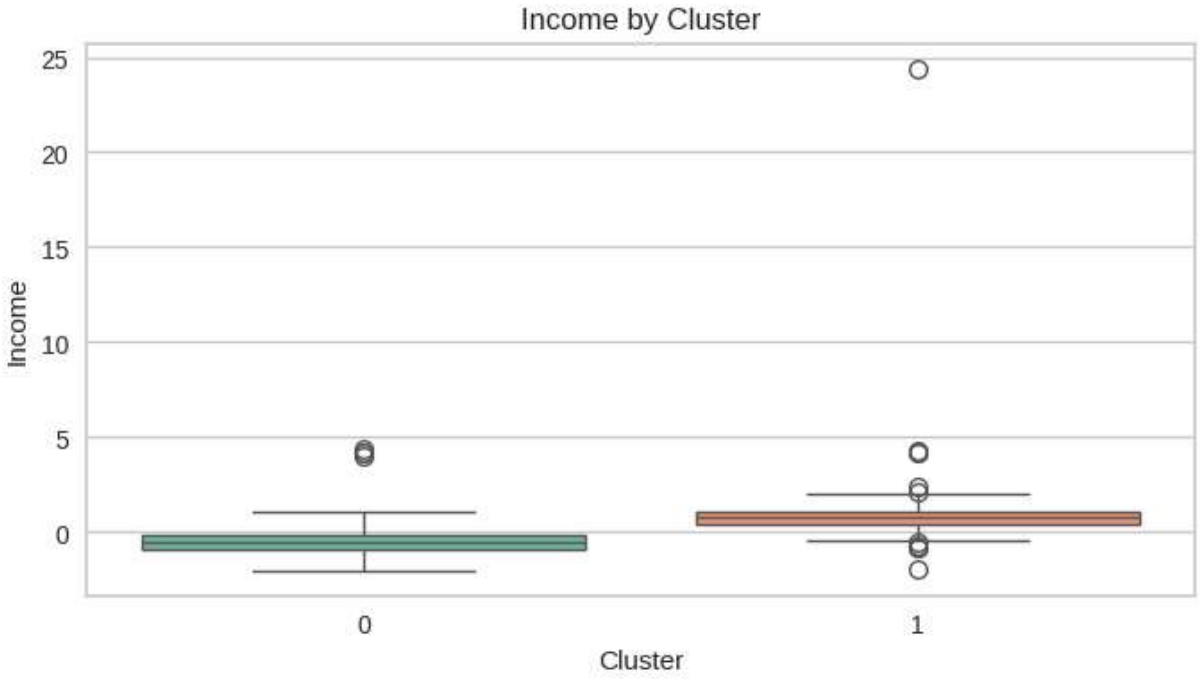
# Cluster Profiling and Comparison

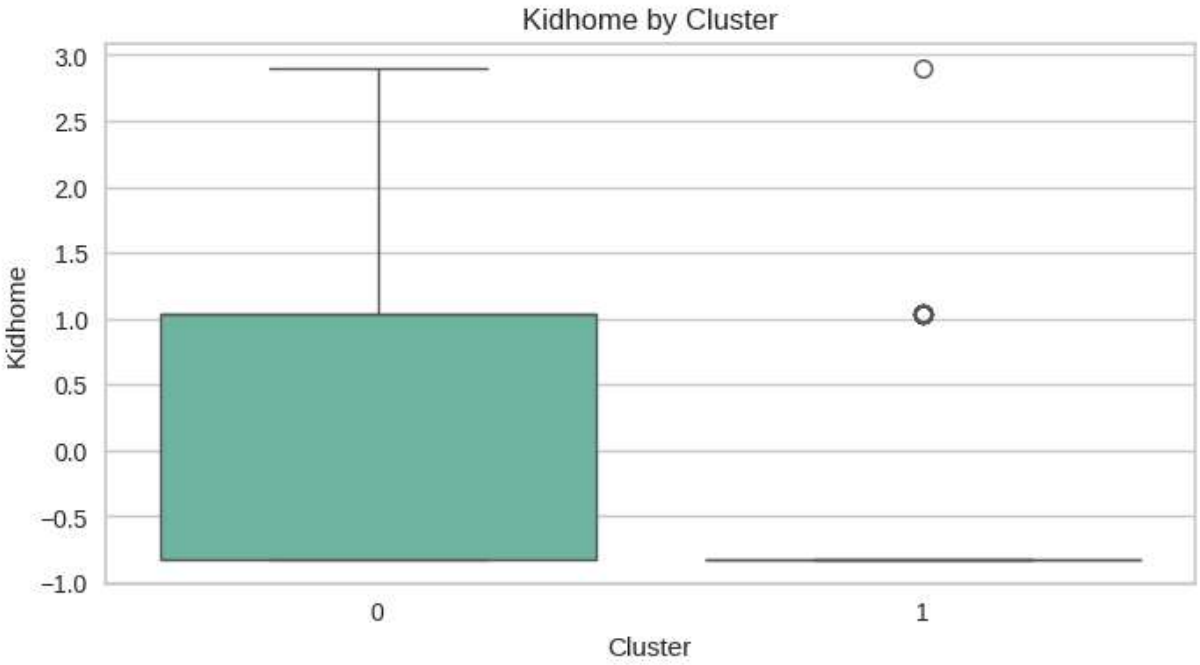
### **Question 10**: Perform cluster profiling using boxplots for the K-Means algorithm. Analyze key characteristics of each cluster and provide detailed observations.

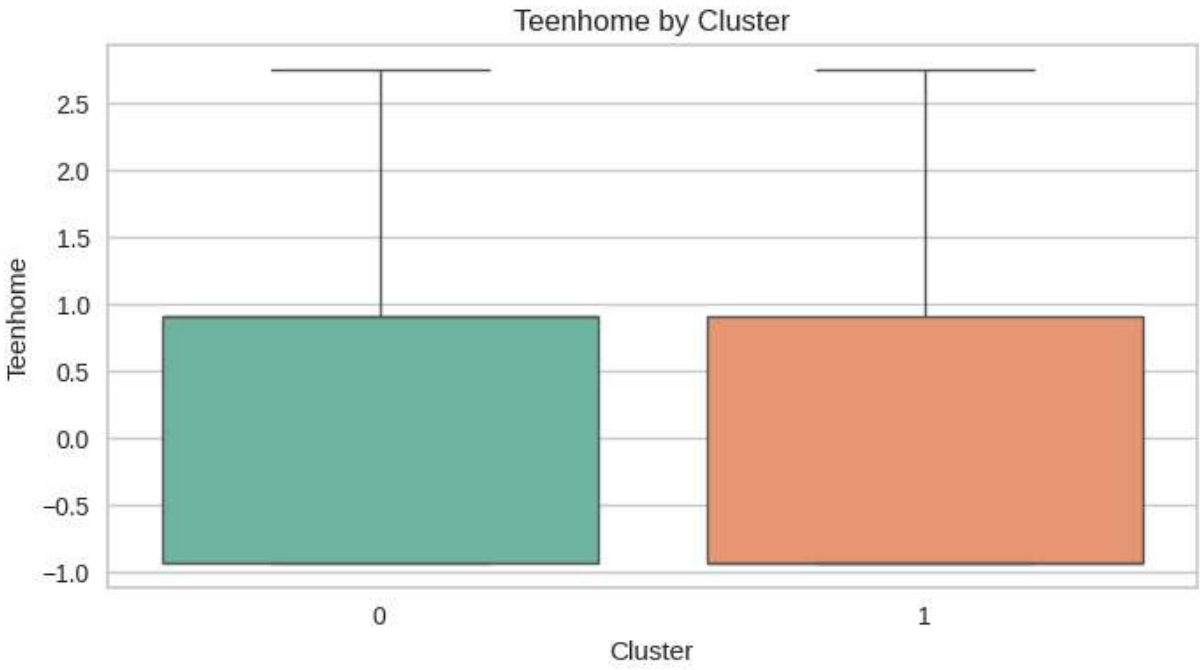
In [17]:

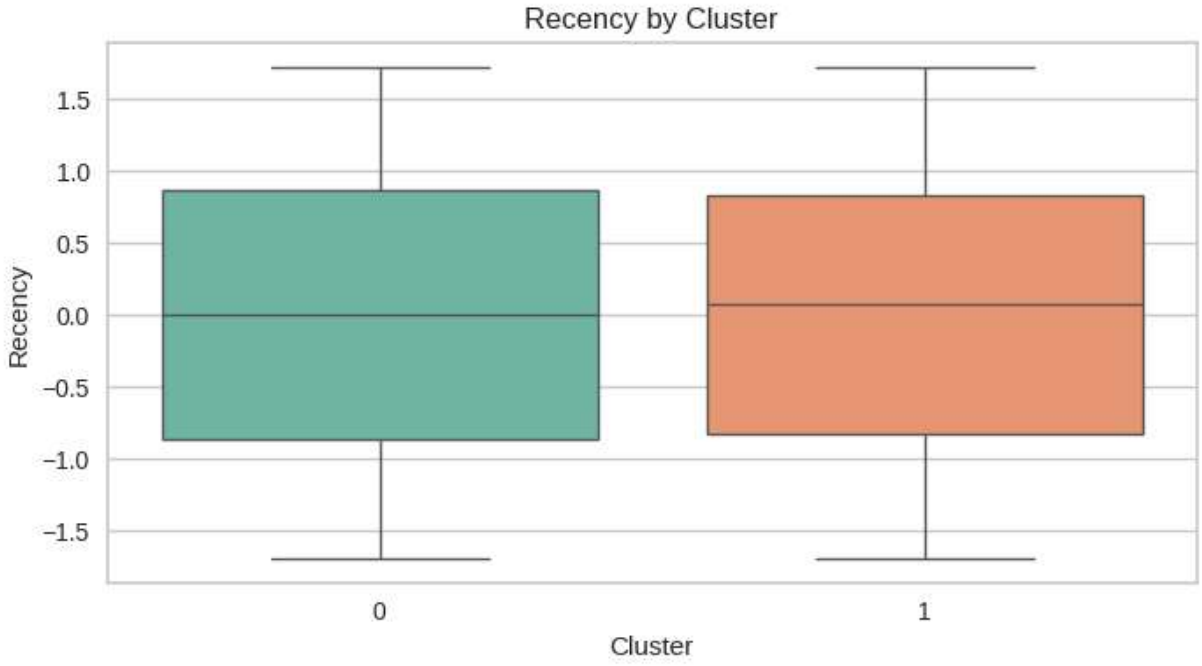
```
df_copy['cluster'] = kmeans_2.labels_ #include the cluster labels assigned by the K
features = df_copy.columns.drop('cluster') #exclude 'cluster' from feature list
for feature in features:
  plt.figure(figsize=(8,4))
  sns.boxplot(data=df_copy, x='cluster', y=feature, palette='Set2')
  plt.title(f'{feature} by Cluster')
  plt.xlabel('Cluster')
  plt.ylabel(feature)
  plt.show()
```

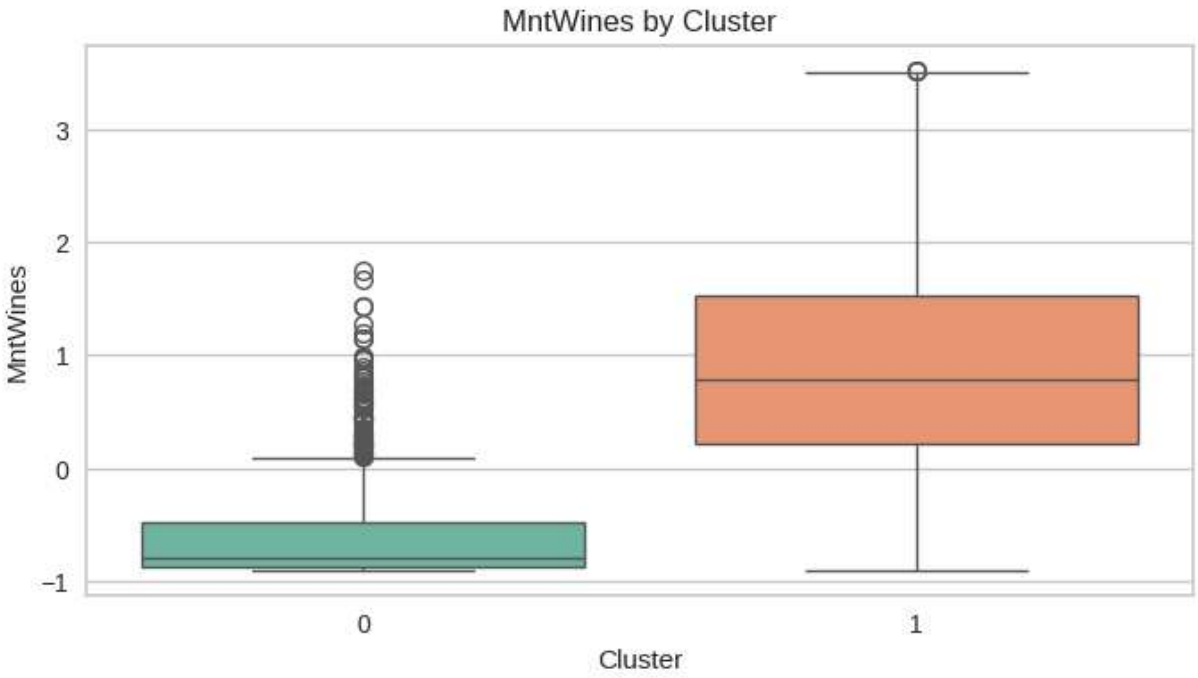


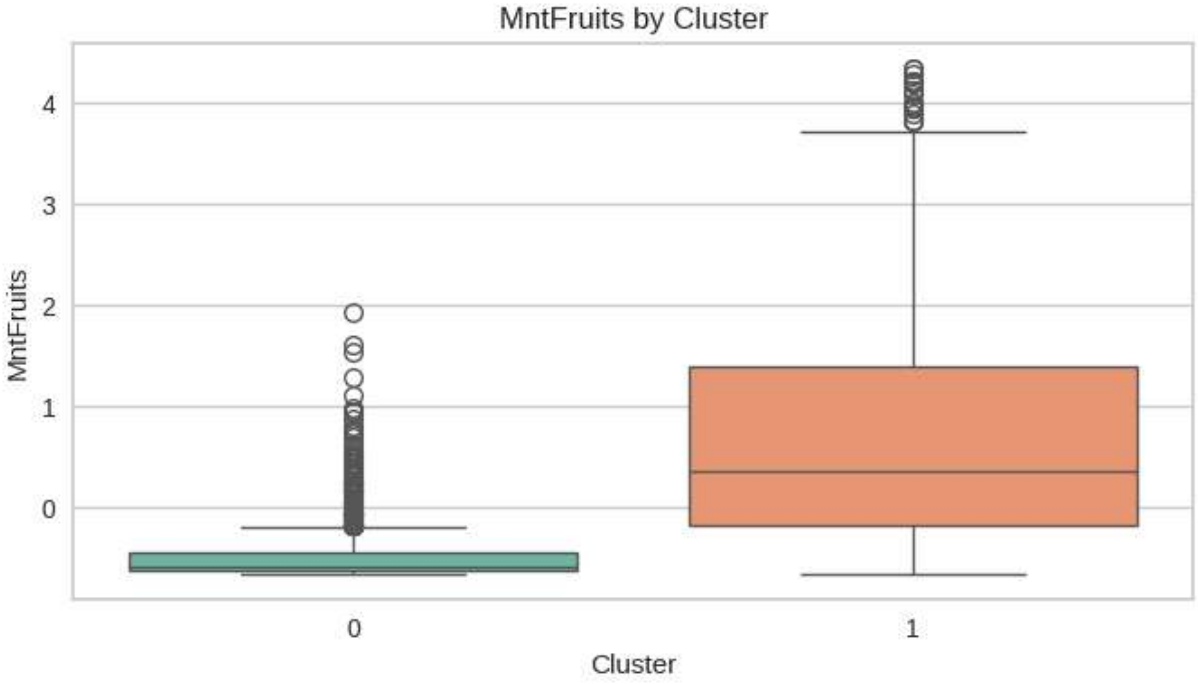

Income by Cluster
Income
25
20
15
10
5
0
0
1
Cluster

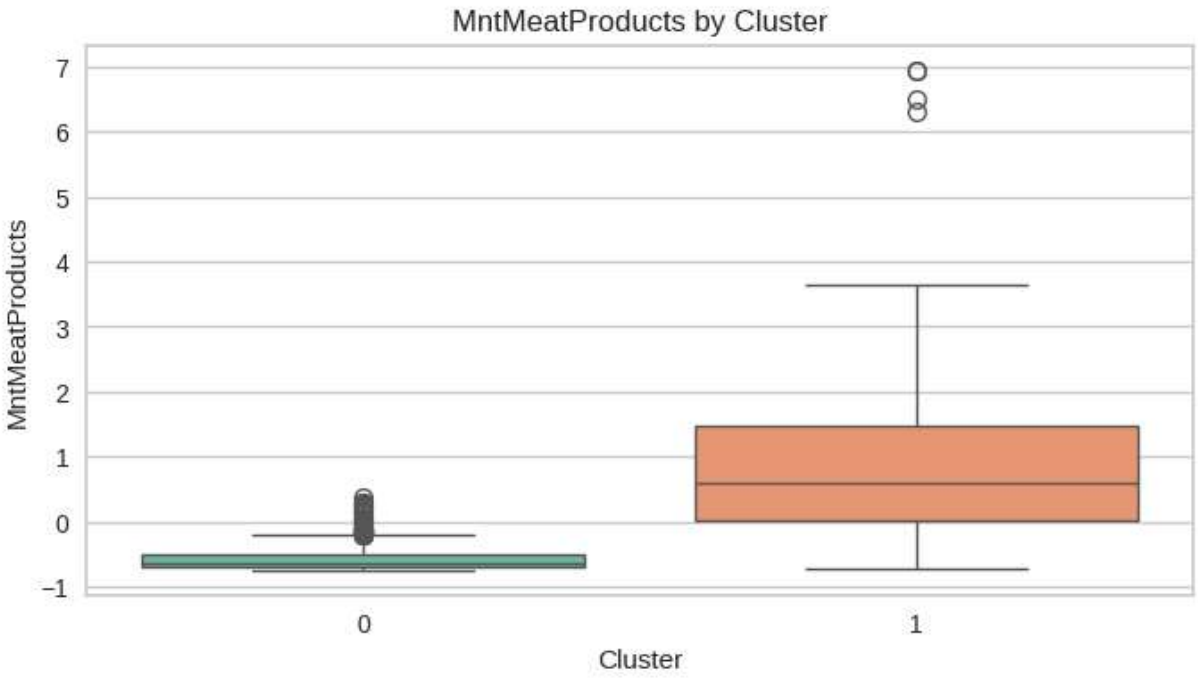

Kidhome by Cluster
Kidhome
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
0
1
Cluster

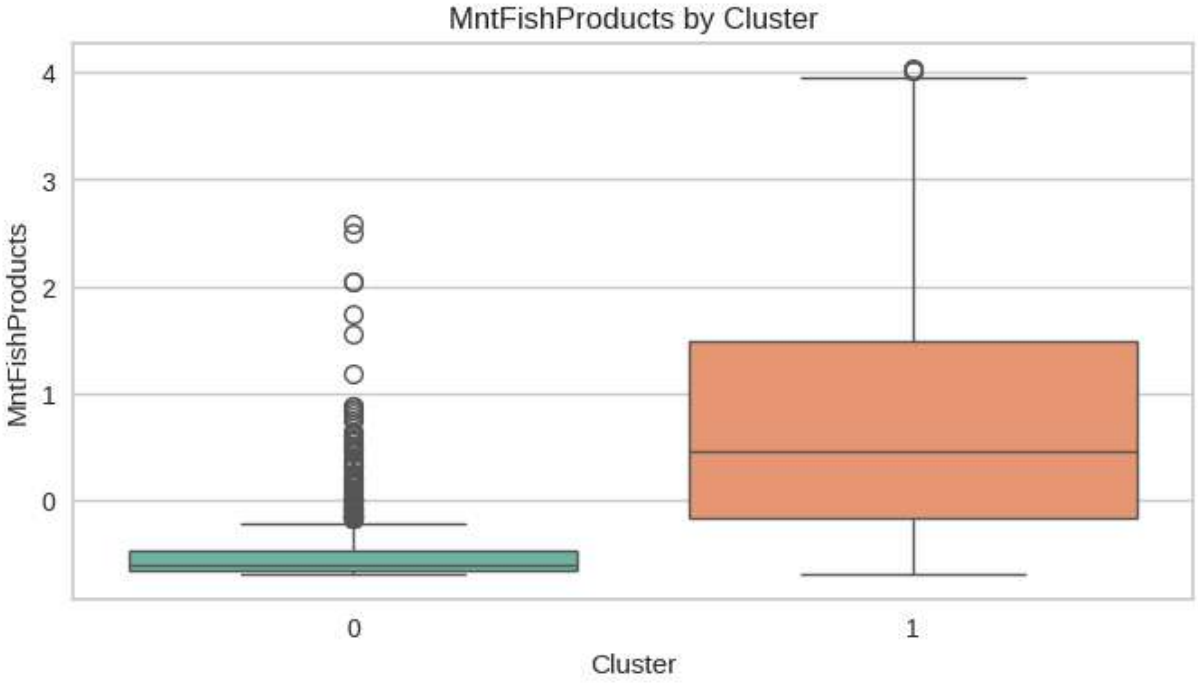

Teenhome by Cluster
Teenhome
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
0
1
Cluster

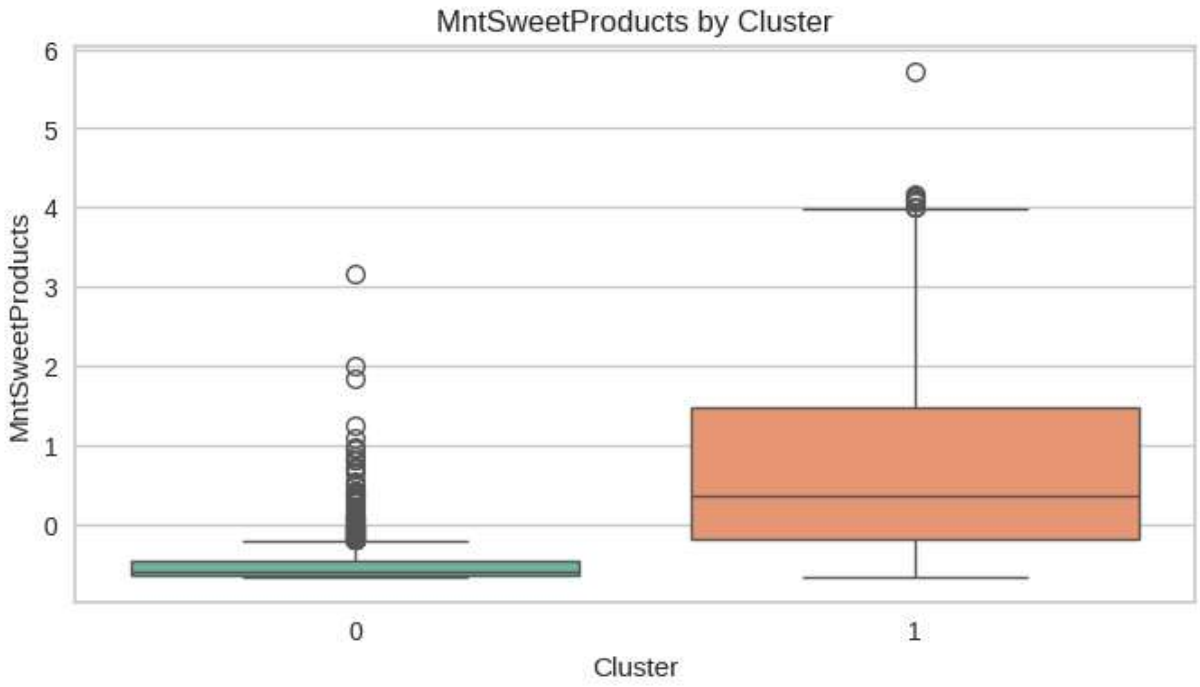

Recency by Cluster
Recency
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0
1
Cluster

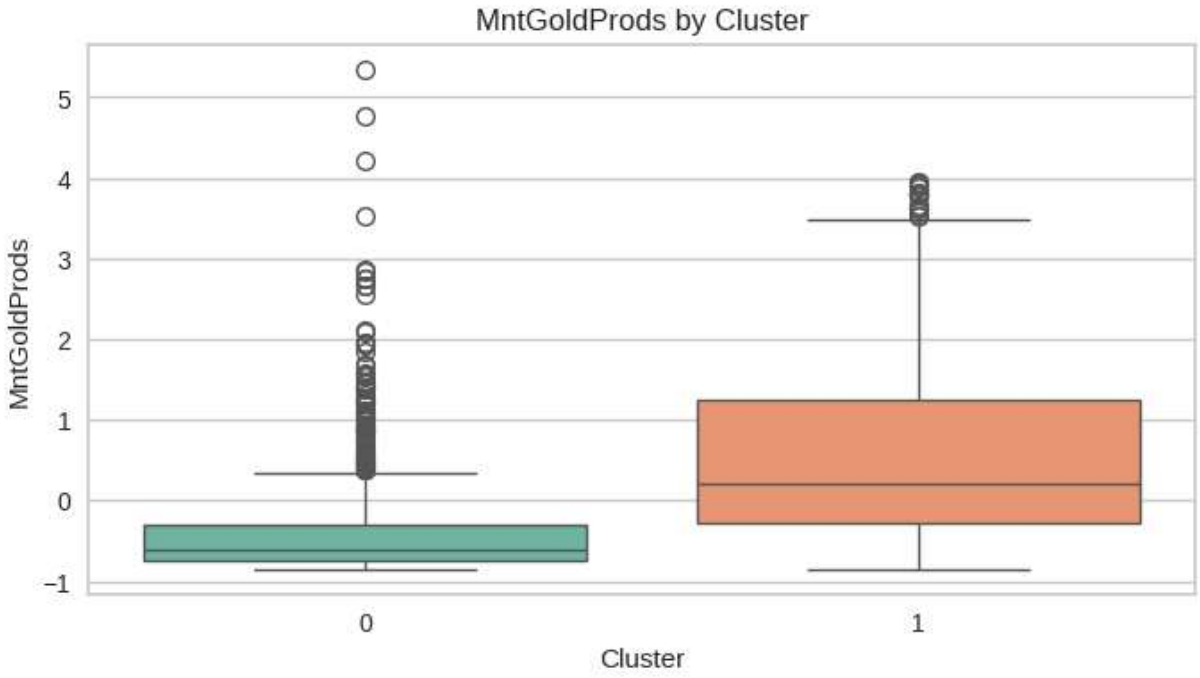

MntWines by Cluster
MntWines
3
2
1
0
−1
0
1
Cluster


MntFruits by Cluster
MntFruits
4
3
2
1
0
0
1
Cluster

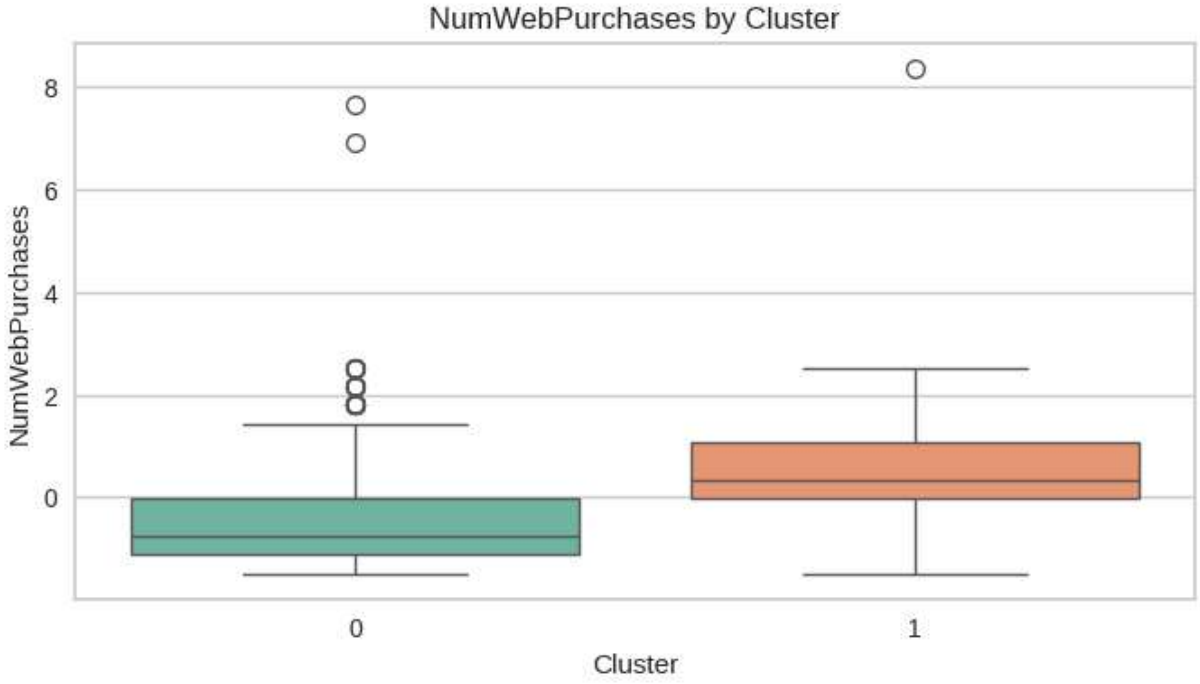

MntMeatProducts by Cluster
MntMeatProducts
7
6
5
4
3
2
1
0
−1
0
1
Cluster

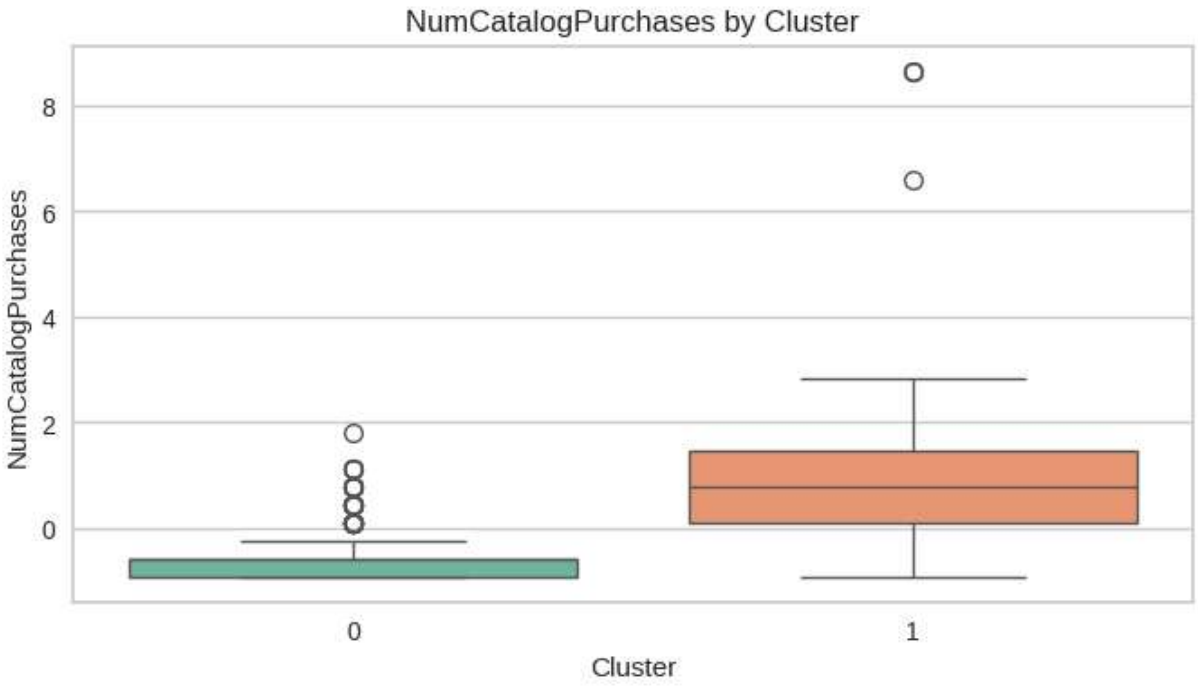

MntFishProducts by Cluster
MntFishProducts
4
3
2
1
0
0
1
Cluster

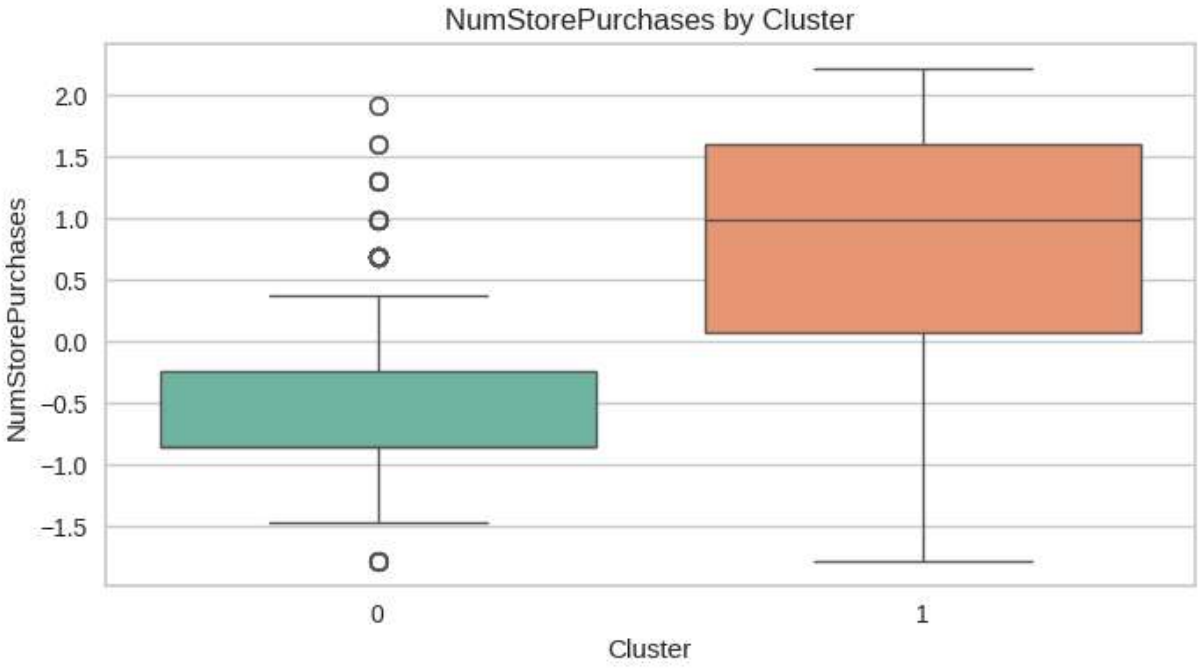

MntSweetProducts by Cluster
MntSweetProducts
Cluster
0
1
2
3
4
5
6

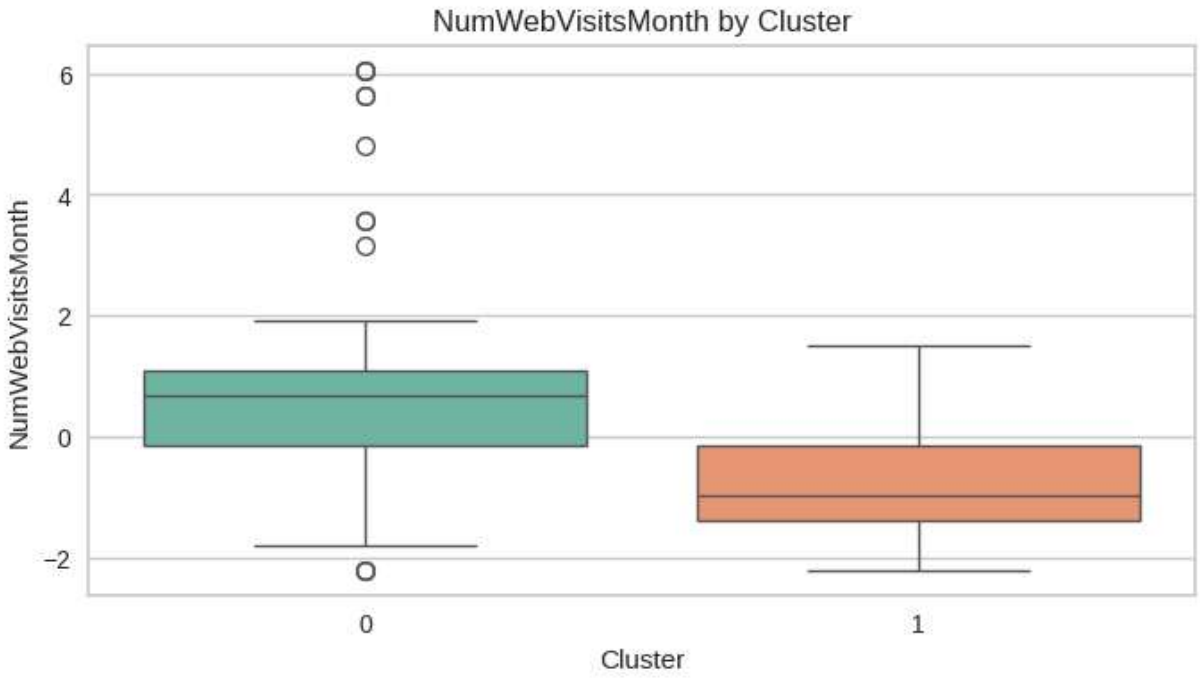

MntGoldProds by Cluster
MntGoldProds
Cluster
0
1
−1
0
1
2
3
4
5

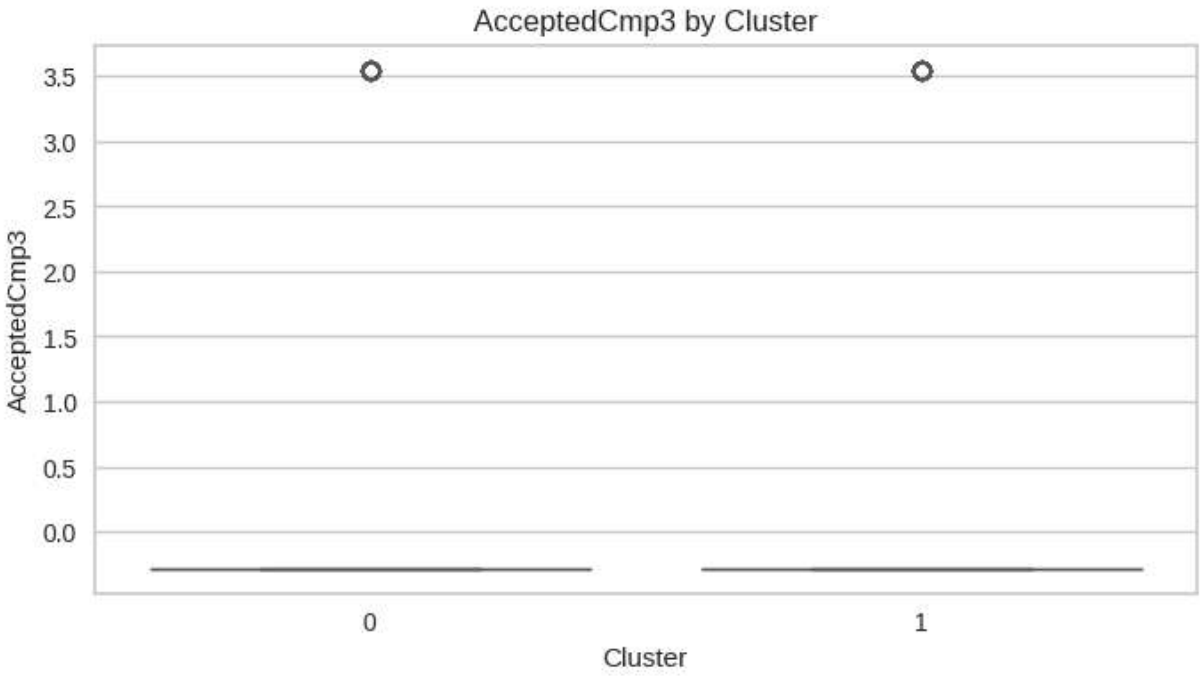

NumDealsPurchases by Cluster
NumDealsPurchases
6
5
4
3
2
1
0
−1
0
1
Cluster

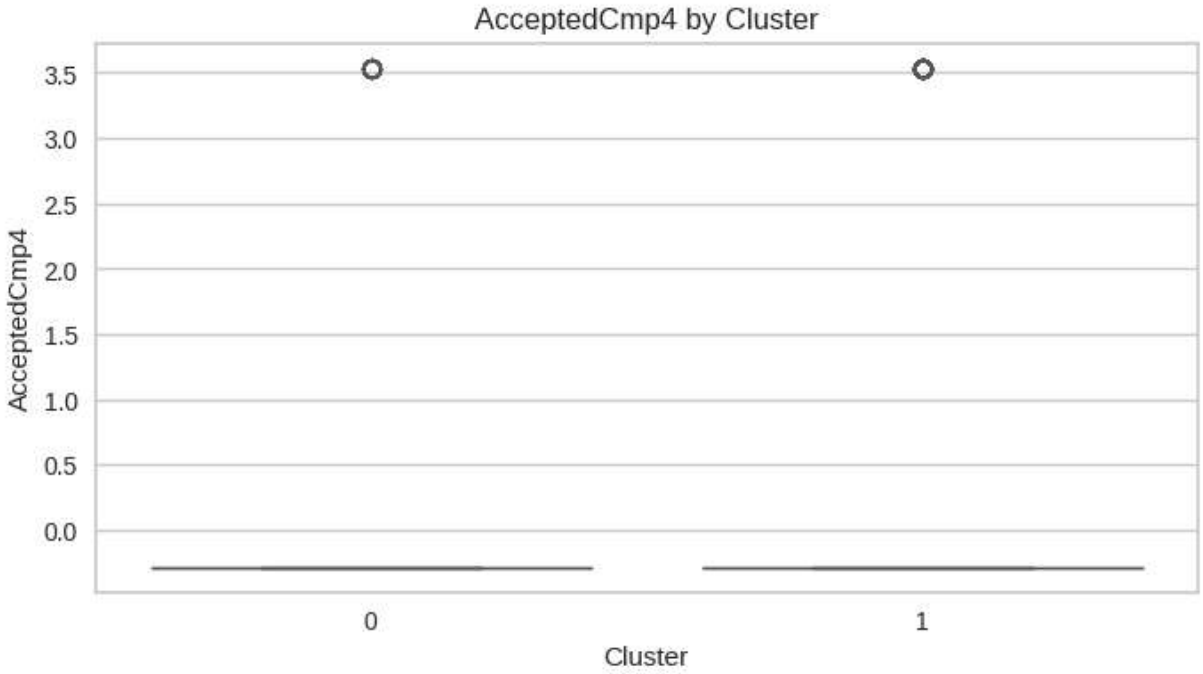

NumWebPurchases by Cluster
NumWebPurchases
8
6
4
2
0
0
1
Cluster

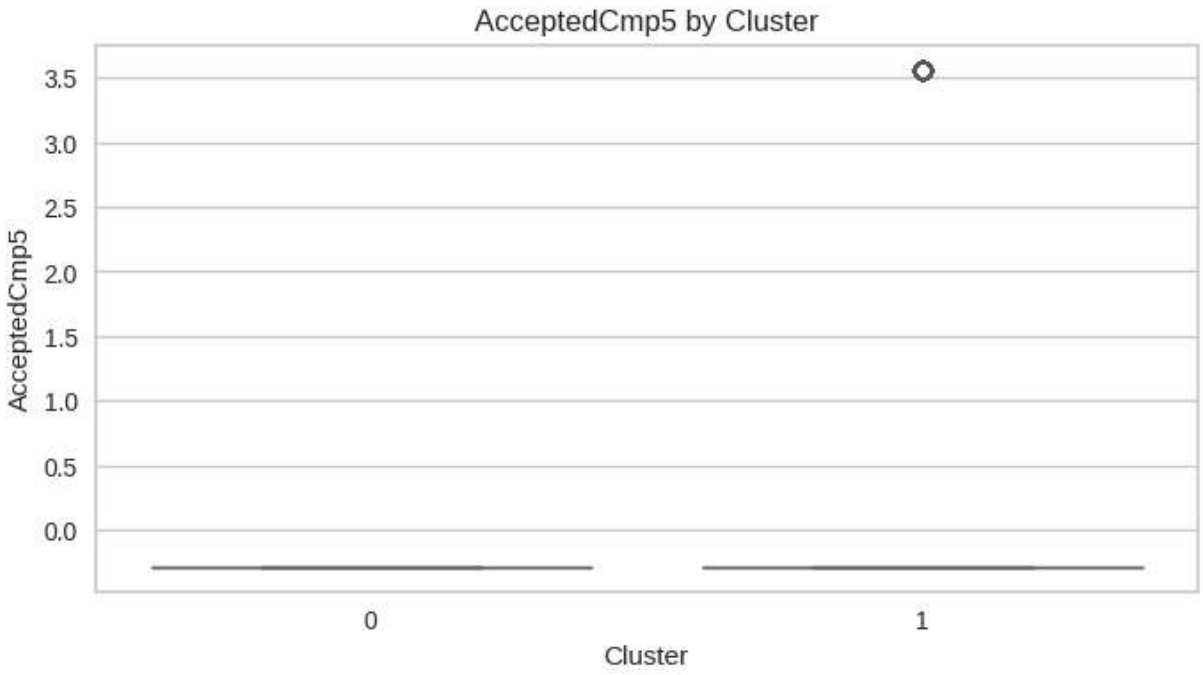

NumCatalogPurchases by Cluster
NumCatalogPurchases
8
6
4
2
0
0
1
Cluster

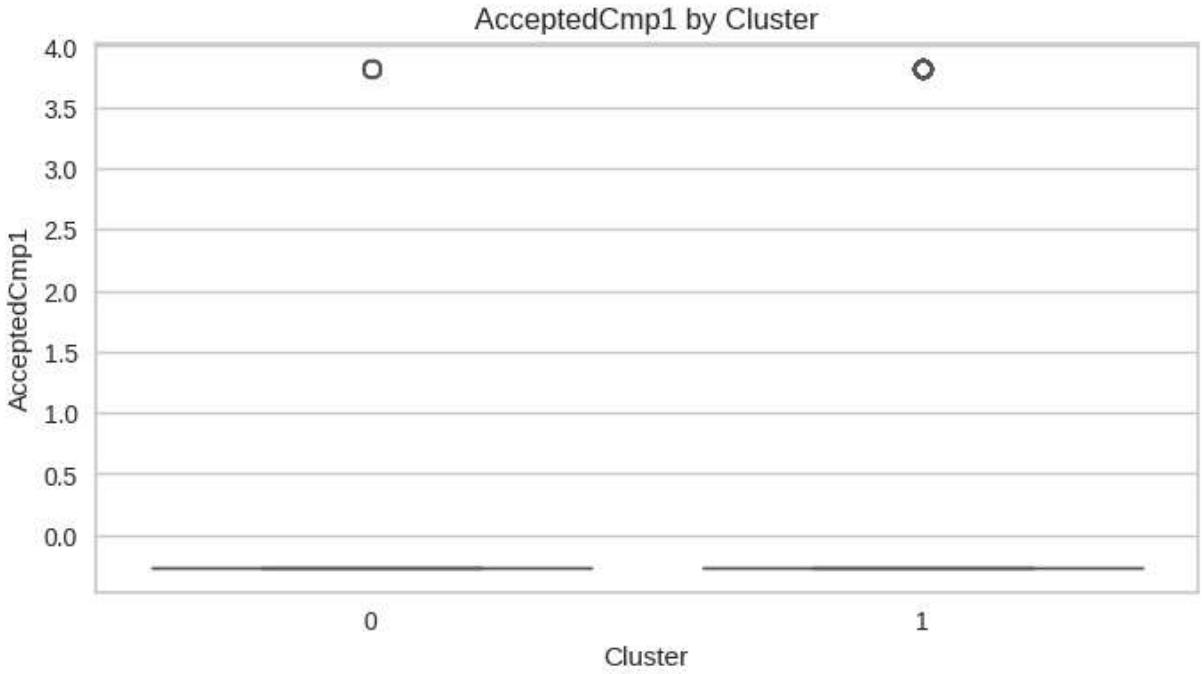

NumStorePurchases by Cluster
NumStorePurchases
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0
1
Cluster

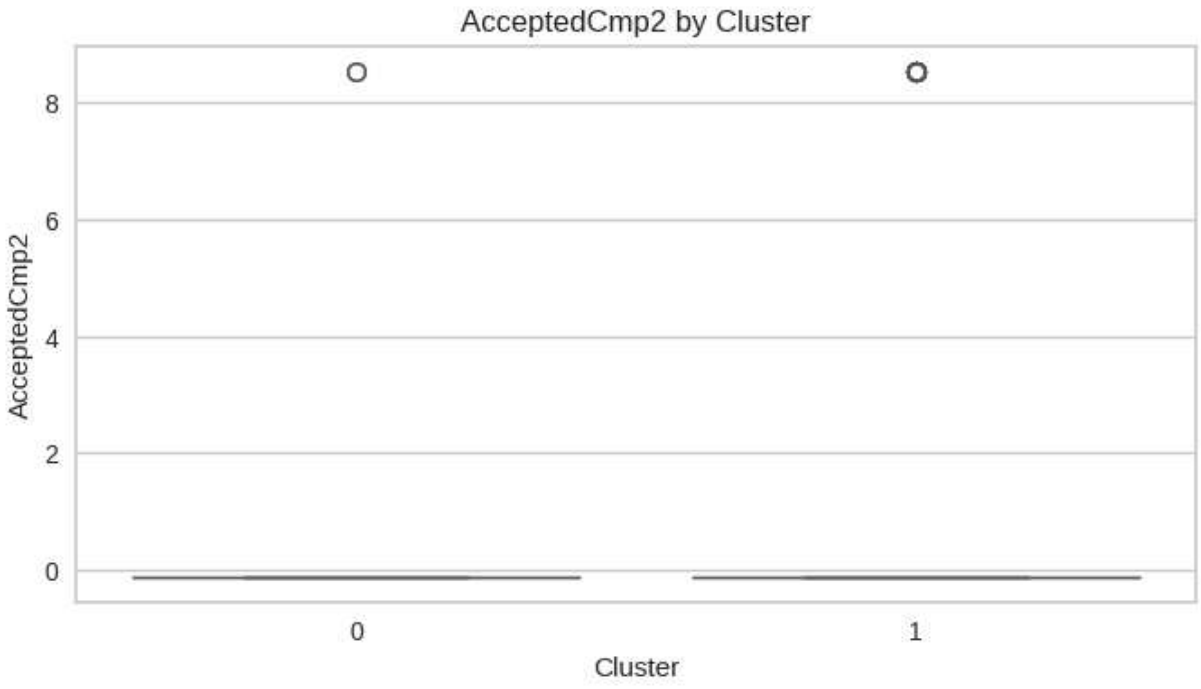

NumWebVisitsMonth by Cluster
NumWebVisitsMonth
6
4
2
0
−2
0
1
Cluster

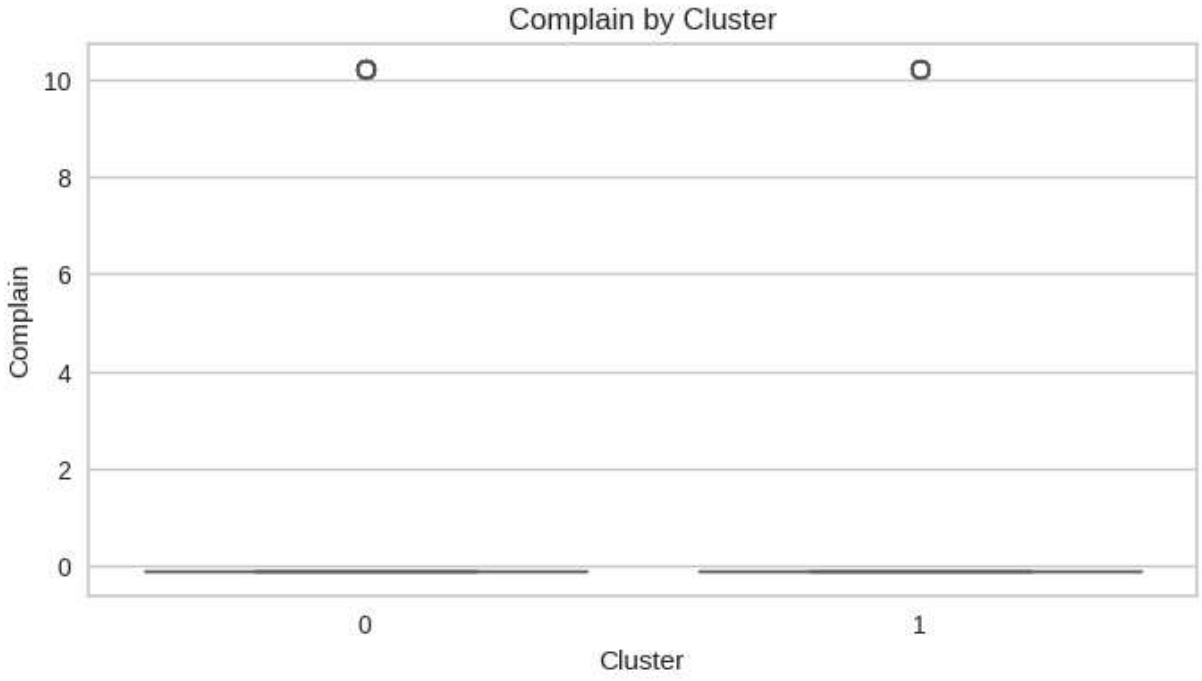

AcceptedCmp3 by Cluster
AcceptedCmp3
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
1
Cluster

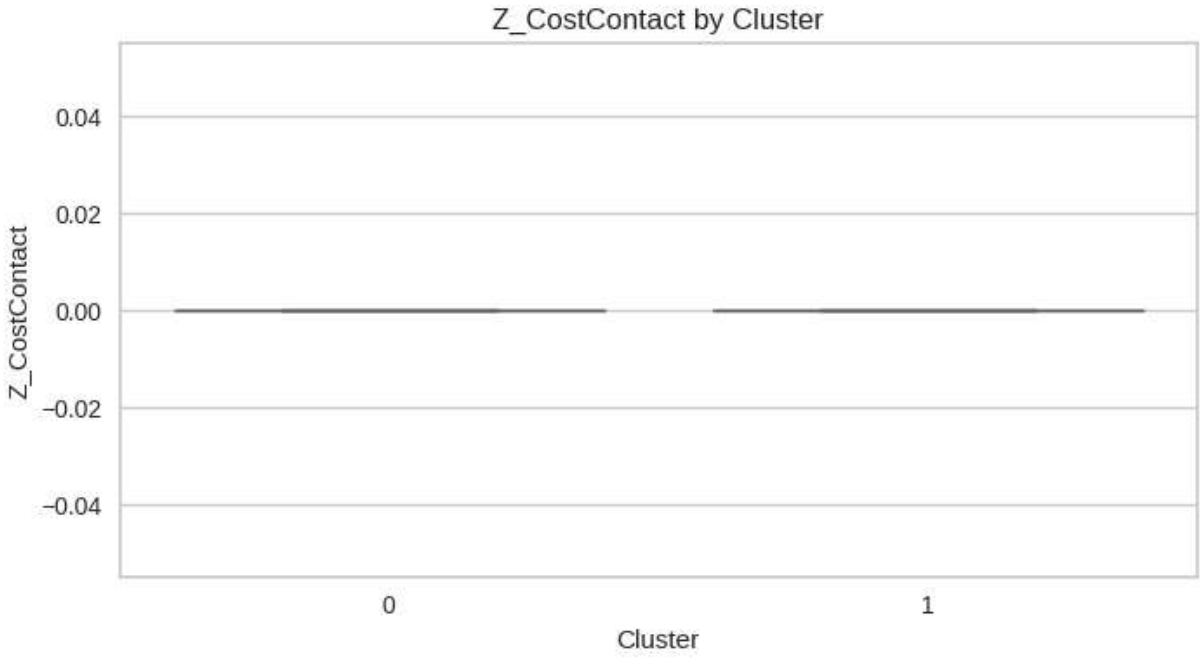

AcceptedCmp4 by Cluster
AcceptedCmp4
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
1
Cluster

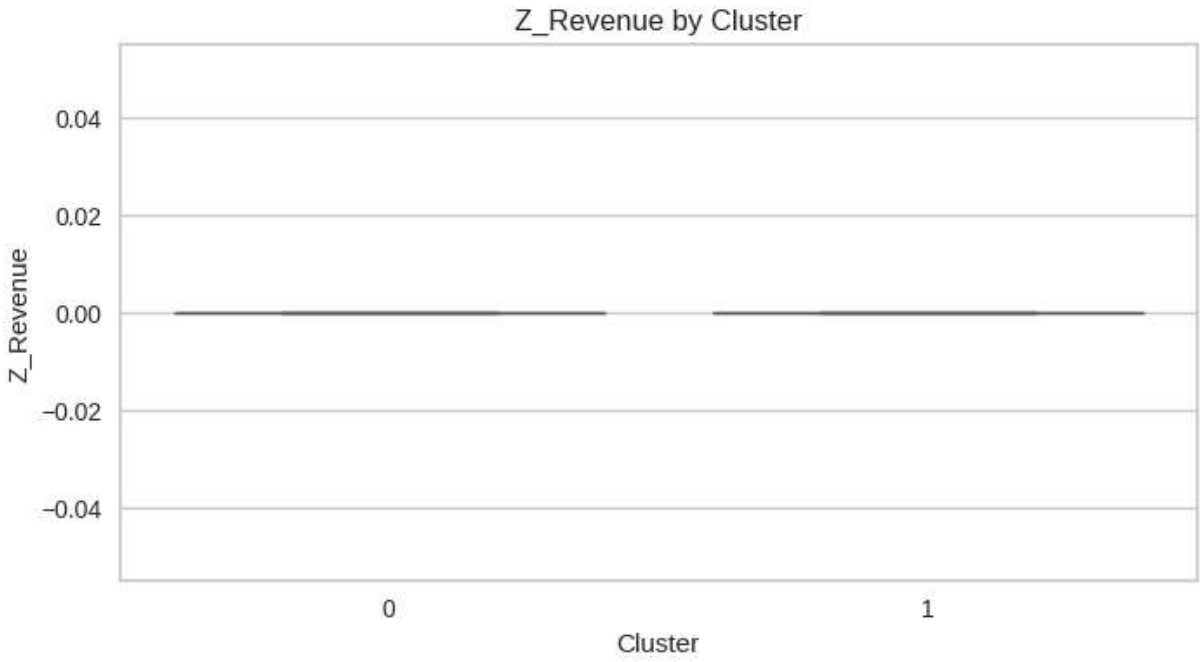

AcceptedCmp5 by Cluster
AcceptedCmp5
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
1
Cluster

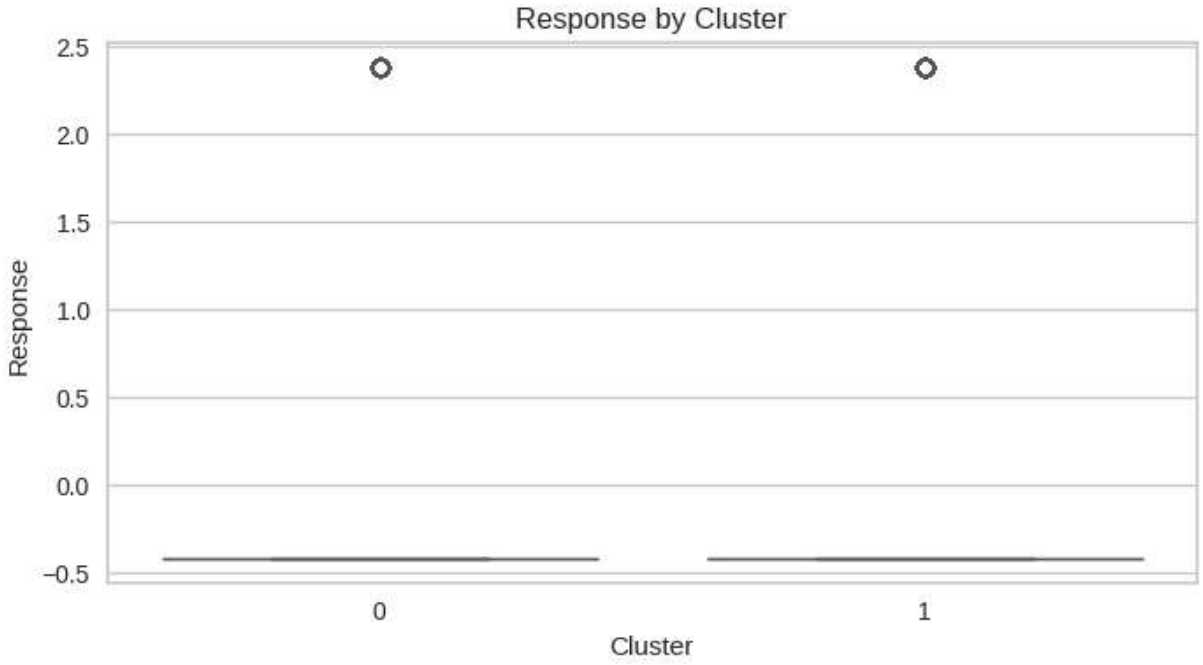

AcceptedCmp1 by Cluster
AcceptedCmp1
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
1
Cluster

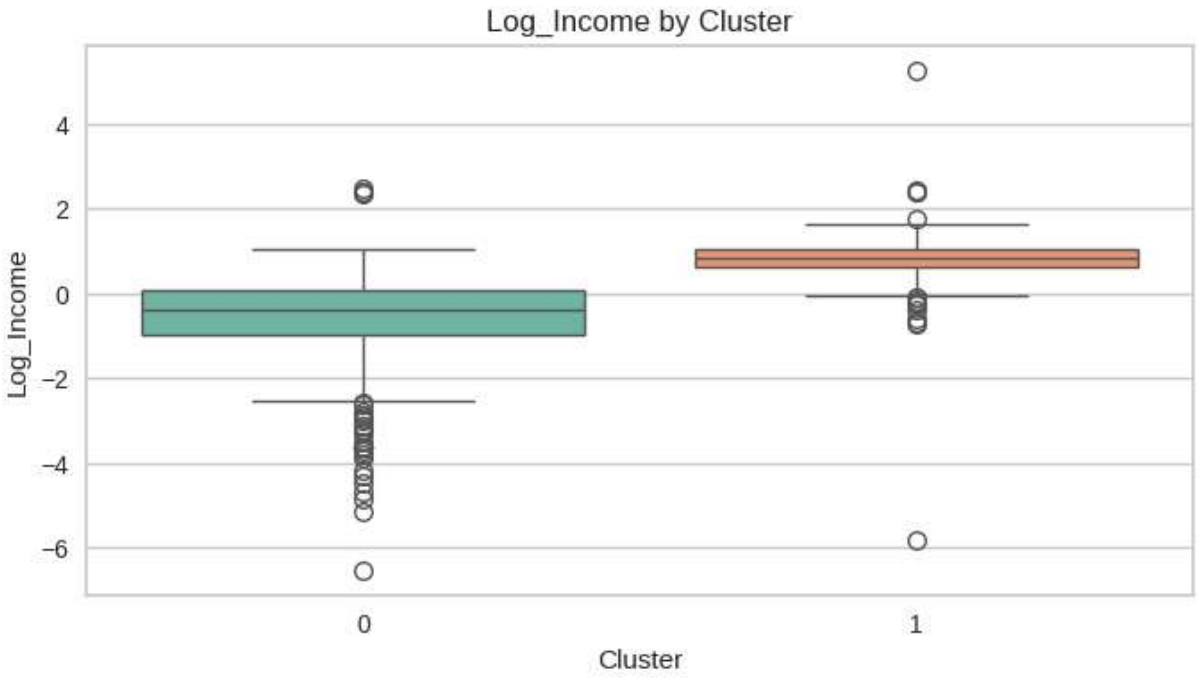

AcceptedCmp2 by Cluster
AcceptedCmp2
8
6
4
2
0
0
1
Cluster

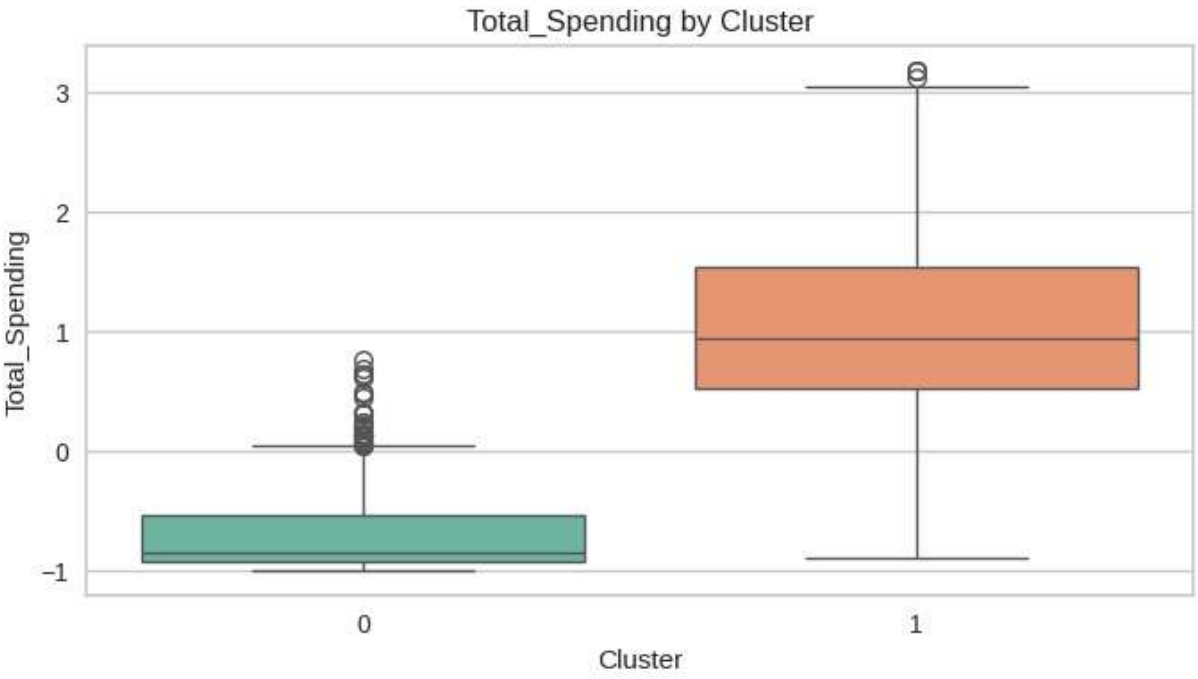

Complain by Cluster
Complain
10
8
6
4
2
0
0
1
Cluster


Z_CostContact by Cluster
Z_CostContact
0.04
0.02
0.00
−0.02
−0.04
0
1
Cluster


Z_Revenue by Cluster
Z_Revenue
0.04
0.02
0.00
−0.02
−0.04
0
1
Cluster


Response by Cluster
Response
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
0
1
Cluster




**Observations:**

Cluster 0:

- slightly older generation
- lower income
- more kids home
- same amount of teens home
- about same recenecy
- less amount spent on wine, fruit, meat, fish, and sweets, and gold with lots of high outliers

- about same deals used
- lower web purchases
- less catalog purchases
- less store purchases
- more web visits monthly

Cluster 1:

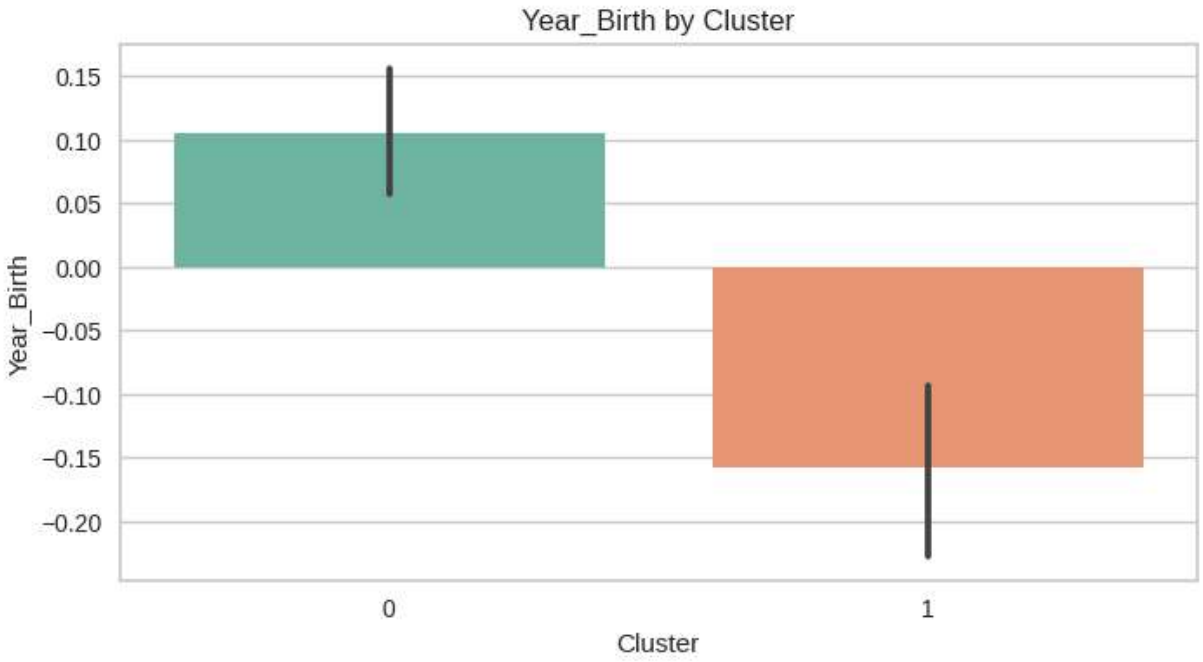
- slightly younger generation
- higher income
- less kids home
- same amount of teens home
- about same recenecy
- more amount spent on wine, fruit, meat, fish, and sweets, and gold with lots of high outliers
- about same deals used
- higher web purchases
- more catalog purchases
- more store purchases
- less web visits monthly

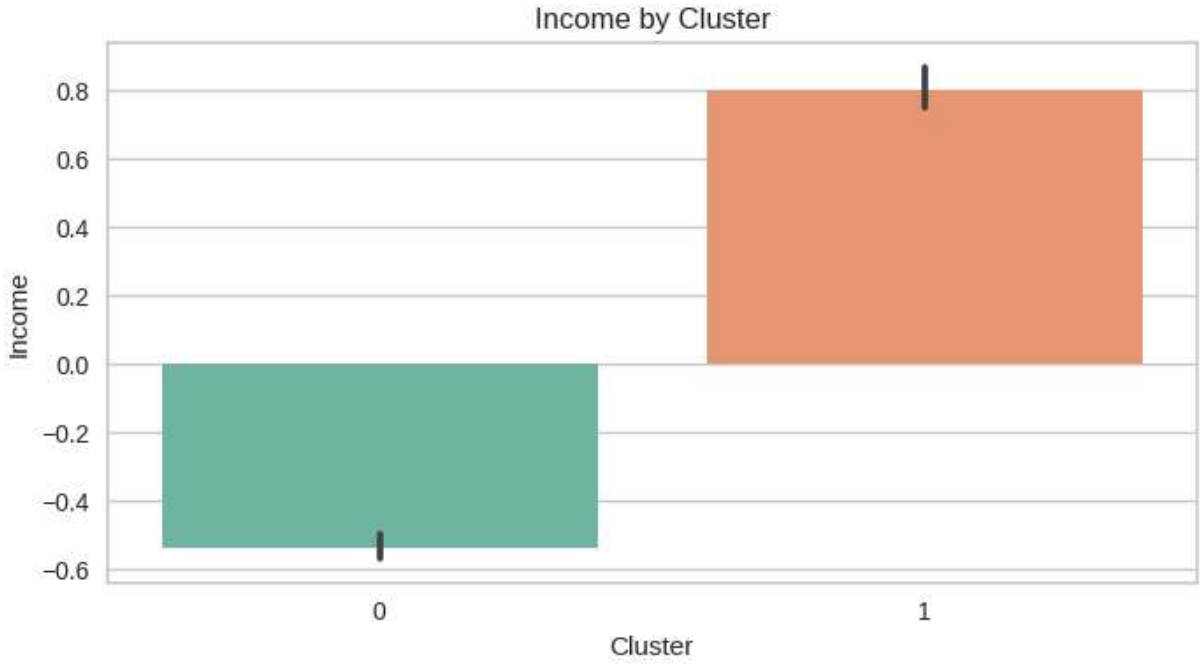
**Behavioral Interpretation Based on Data Type:** Cluster 0: This clsutering of customers represents financially constrained families who browse frequently but spend less. Likely shoppinng for needs and not wants. They are deal-sensitive and time-sensitive.

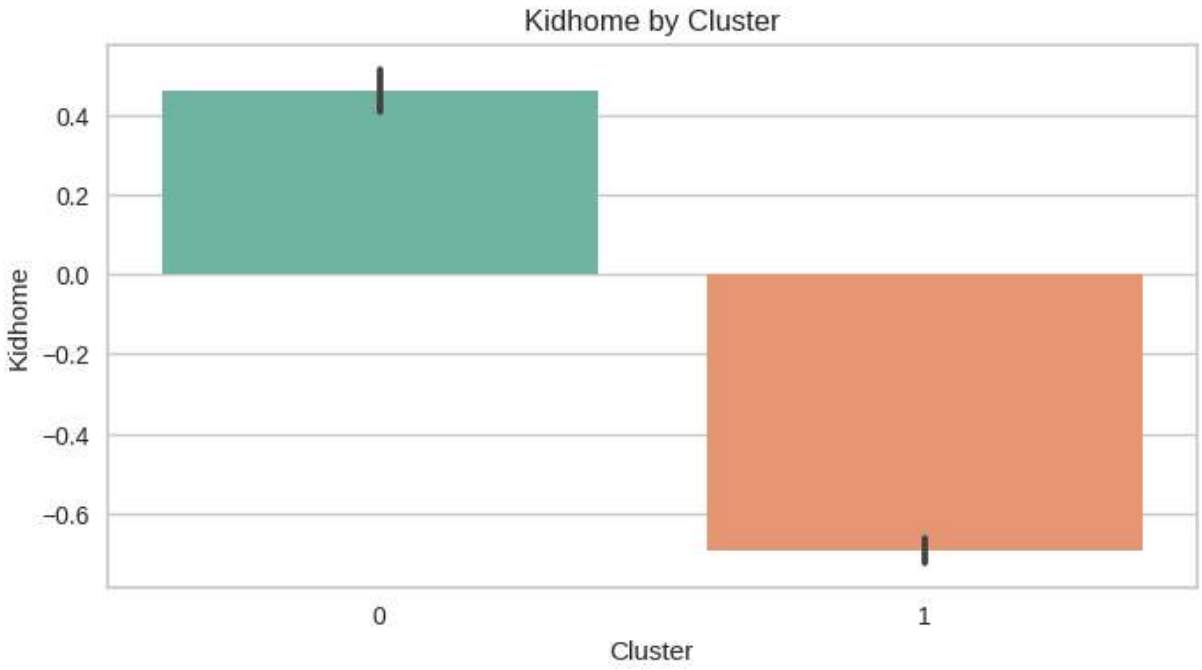
Cluster 1: This clustering of customers represents a younger generation with discretionary spending, who spend more on items regarless of deals available.

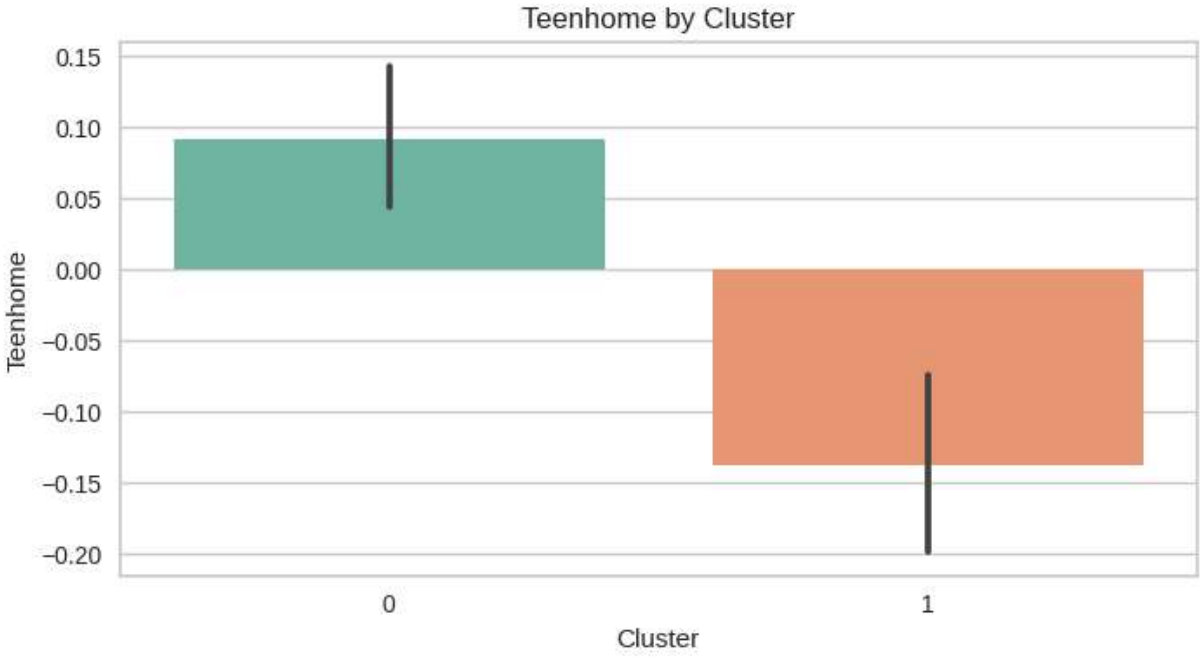
## **Question 11**: Perform cluster profiling on the data using a barplot for the K-Means algorithm. Provide insights and key observations for each cluster based on the visual analysis.

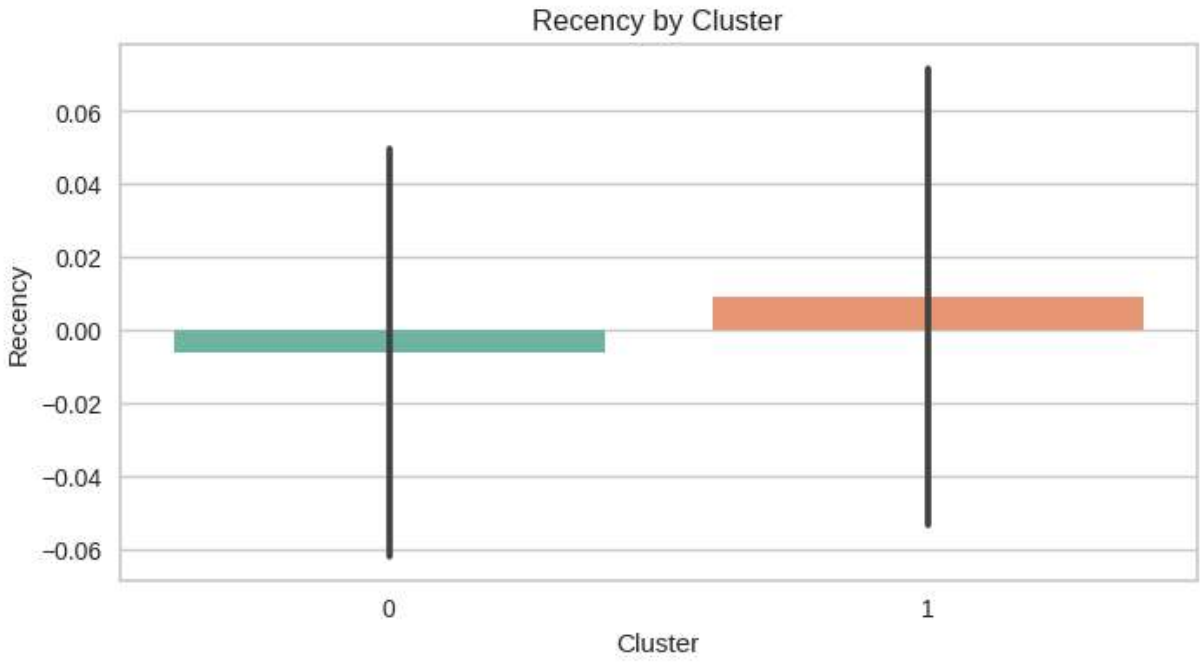
In [18]:
```
for feature in features:
  plt.figure(figsize=(8,4))
  sns.barplot(data=df_copy, x='cluster', y=feature, palette='Set2')
  plt.title(f'{feature} by Cluster')
  plt.xlabel('Cluster')
  plt.ylabel(feature)
  plt.show()
```

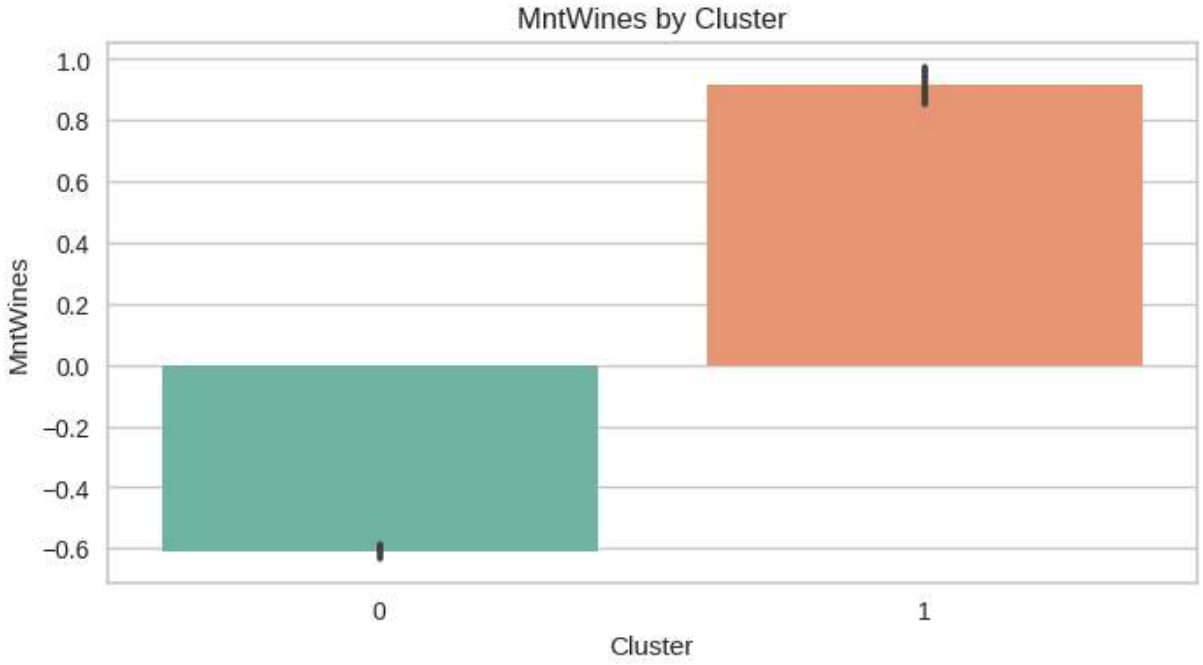

Year_Birth by Cluster
Year_Birth
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
0
1
Cluster

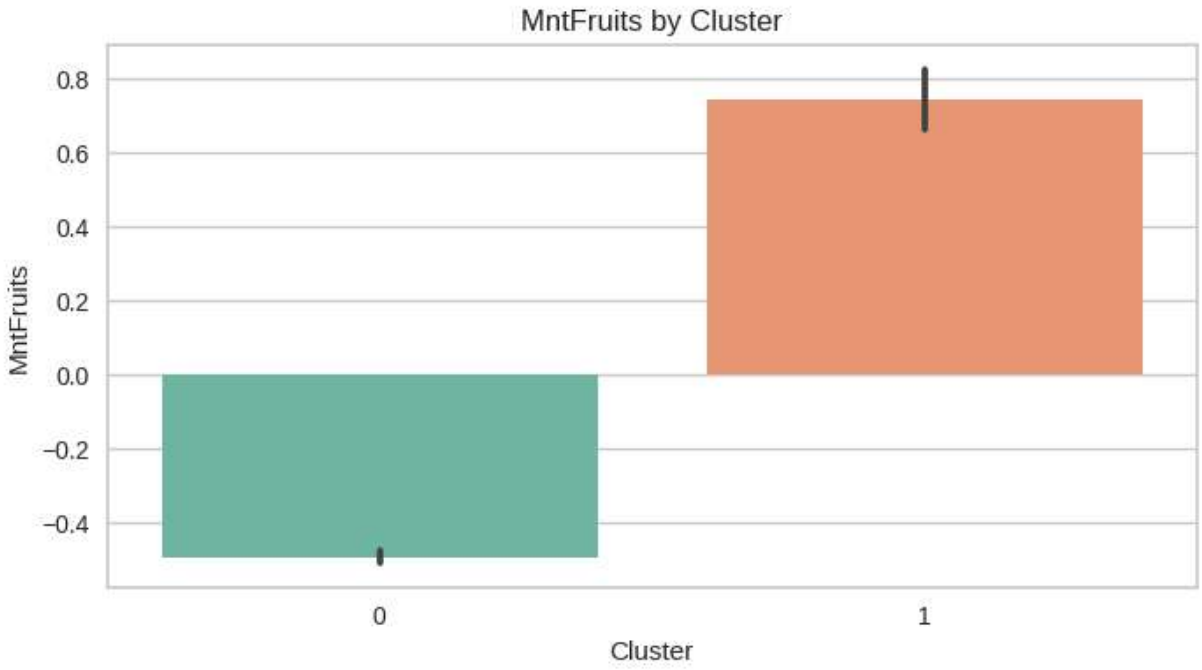

Income by Cluster
Income
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
0
1
Cluster

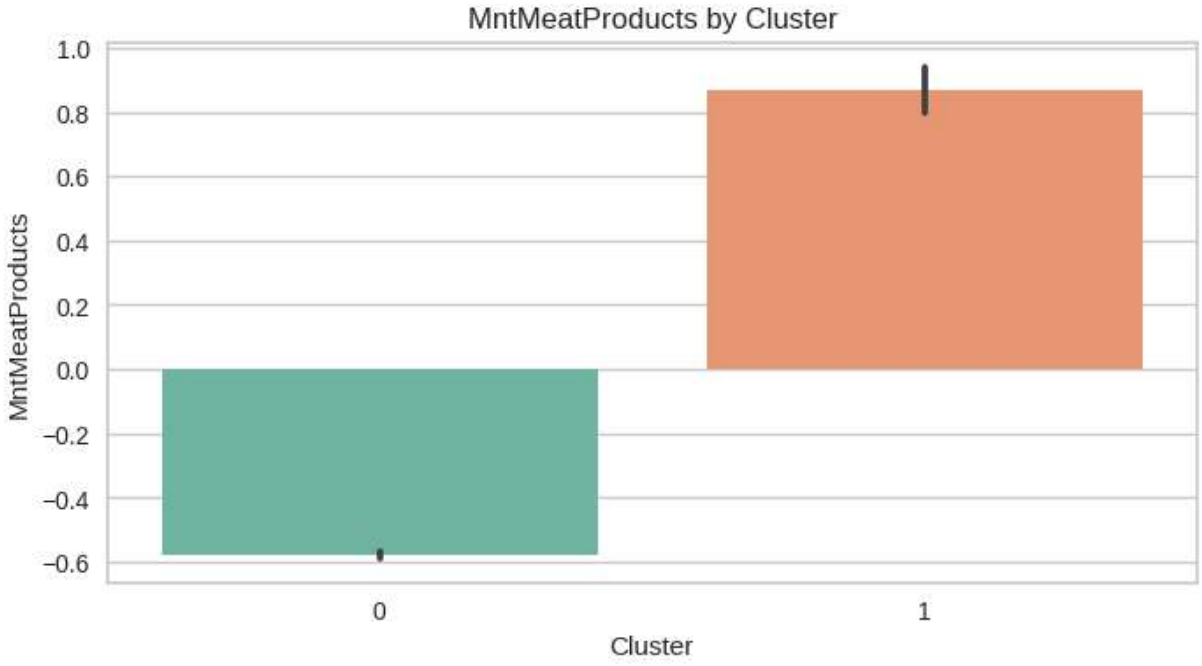

Kidhome by Cluster
Kidhome
0.4
0.2
0.0
−0.2
−0.4
−0.6
0
1
Cluster

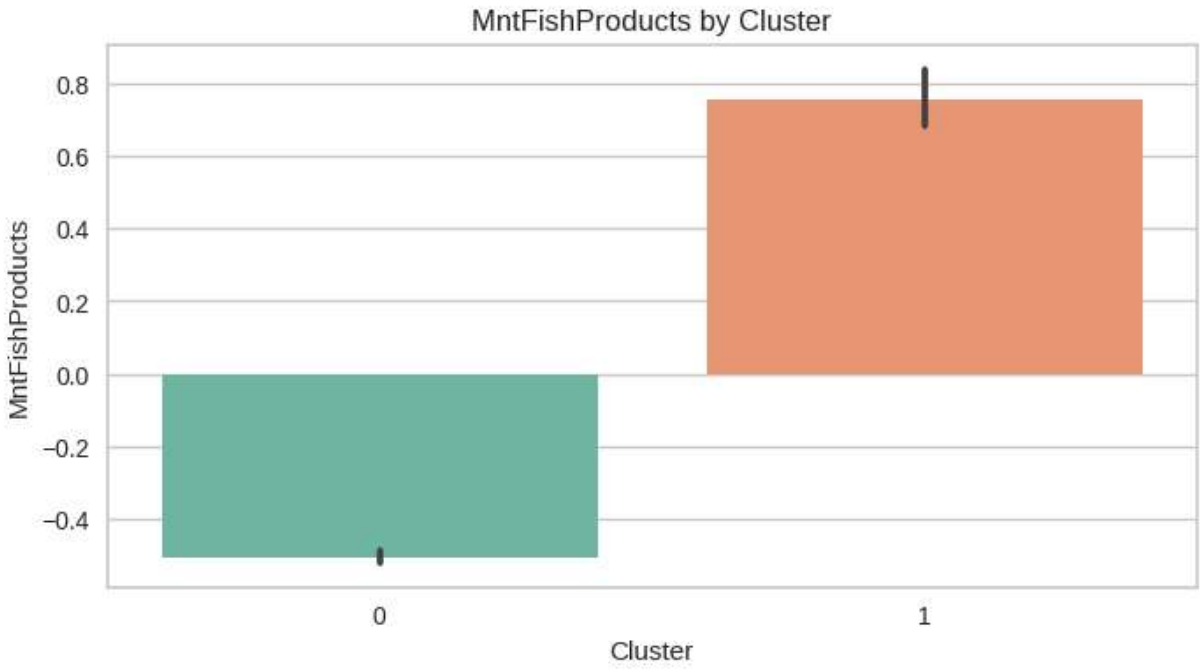

Teenhome by Cluster
Teenhome
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
0
1
Cluster

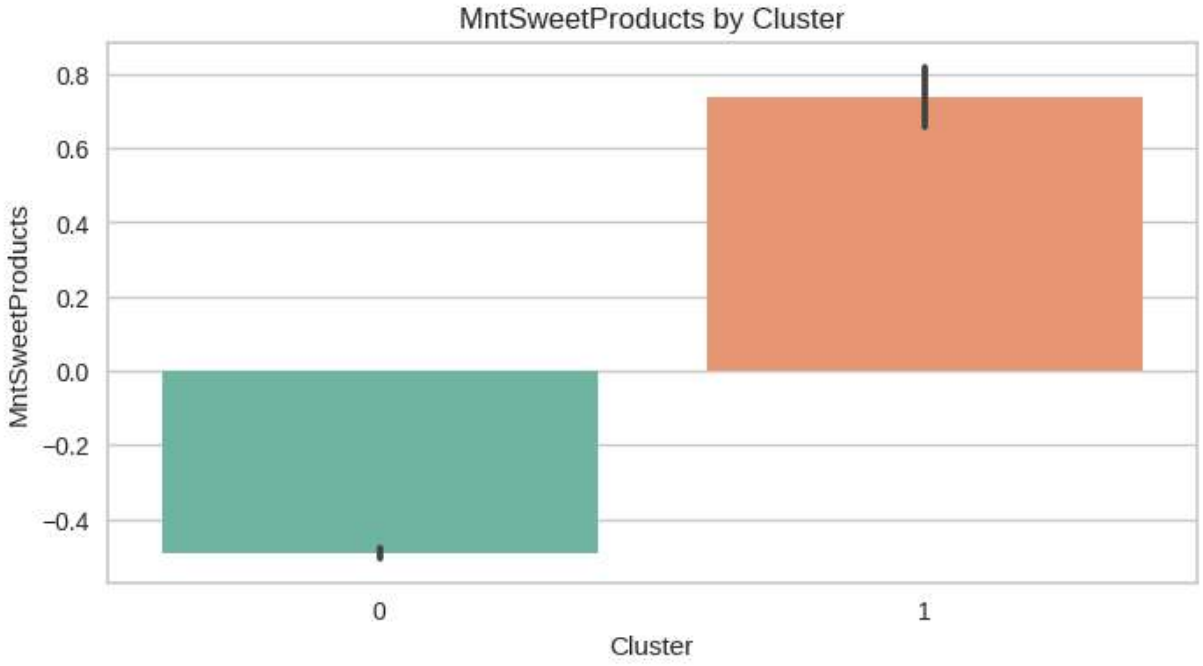

Recency by Cluster
Recency
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
0
1
Cluster

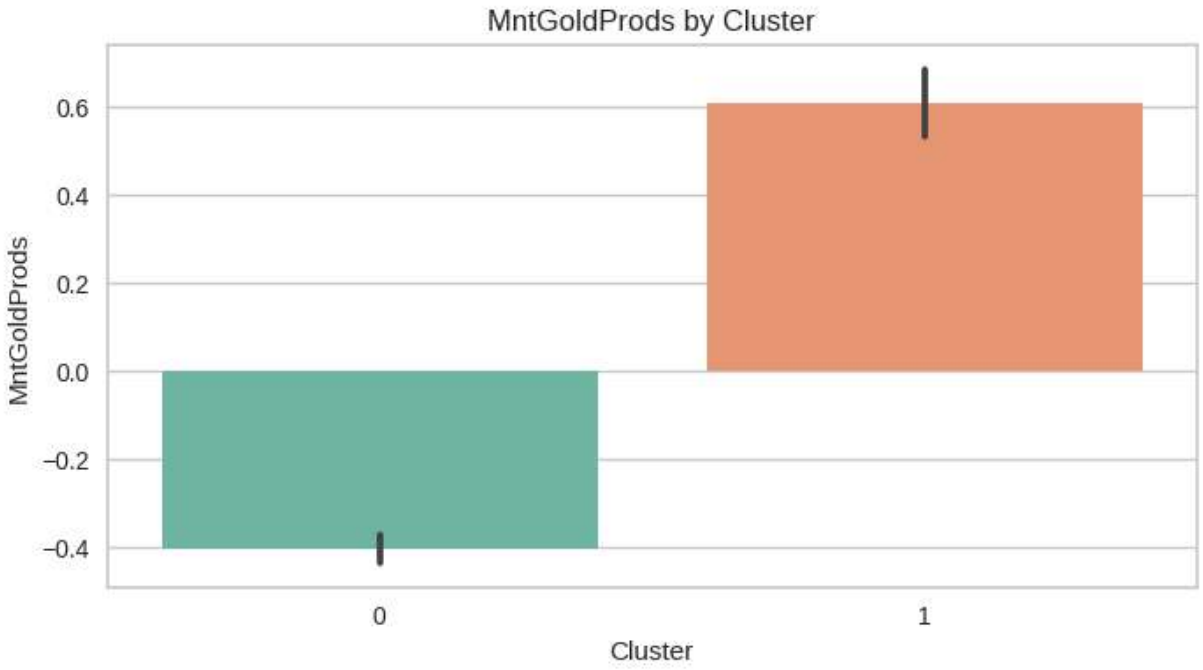

MntWines by Cluster
MntWines
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
0
1
Cluster

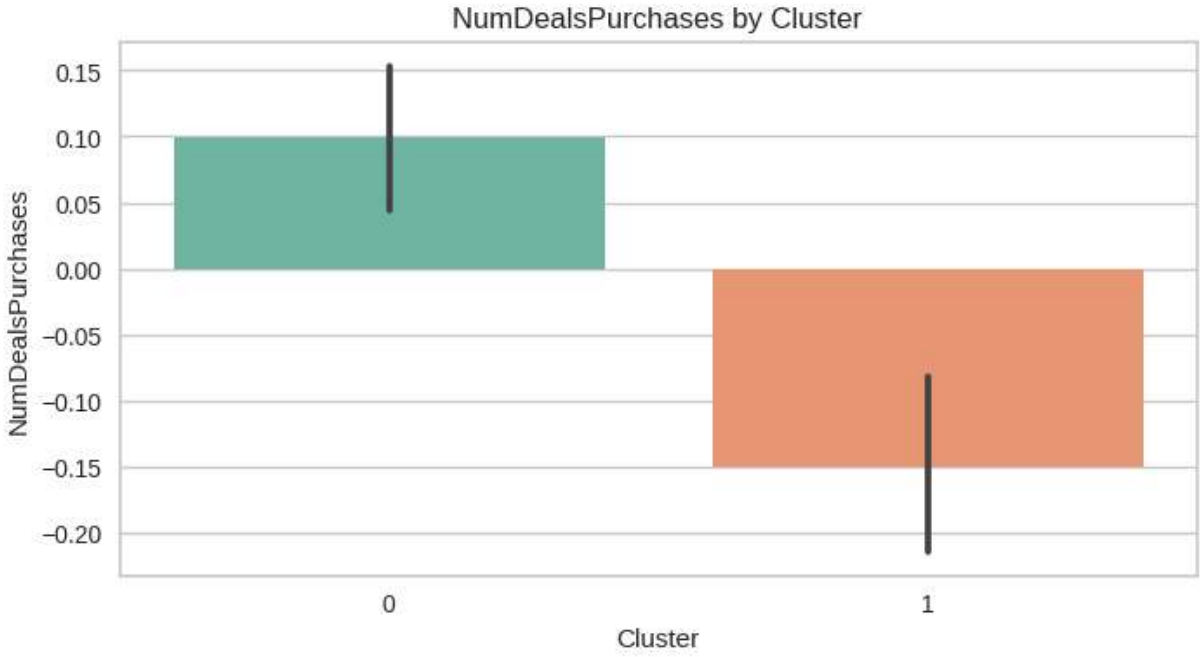

MntFruits by Cluster
MntFruits
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0
1
Cluster


MntMeatProducts by Cluster
MntMeatProducts
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
0
1
Cluster

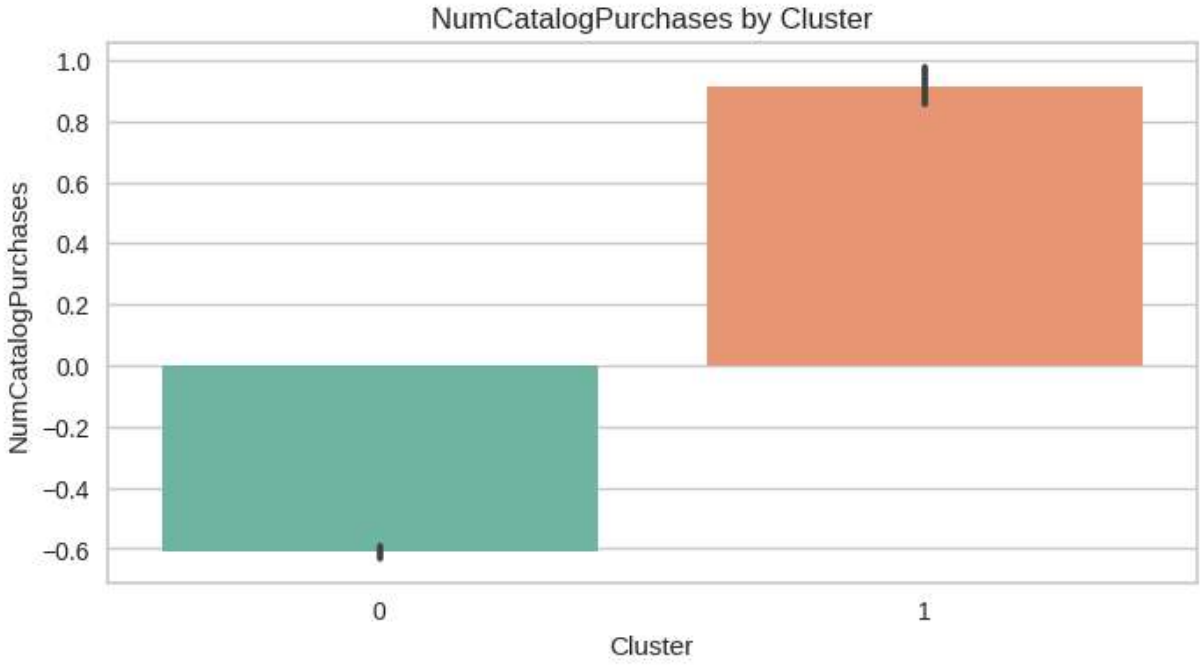

MntFishProducts by Cluster
MntFishProducts
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0
1
Cluster

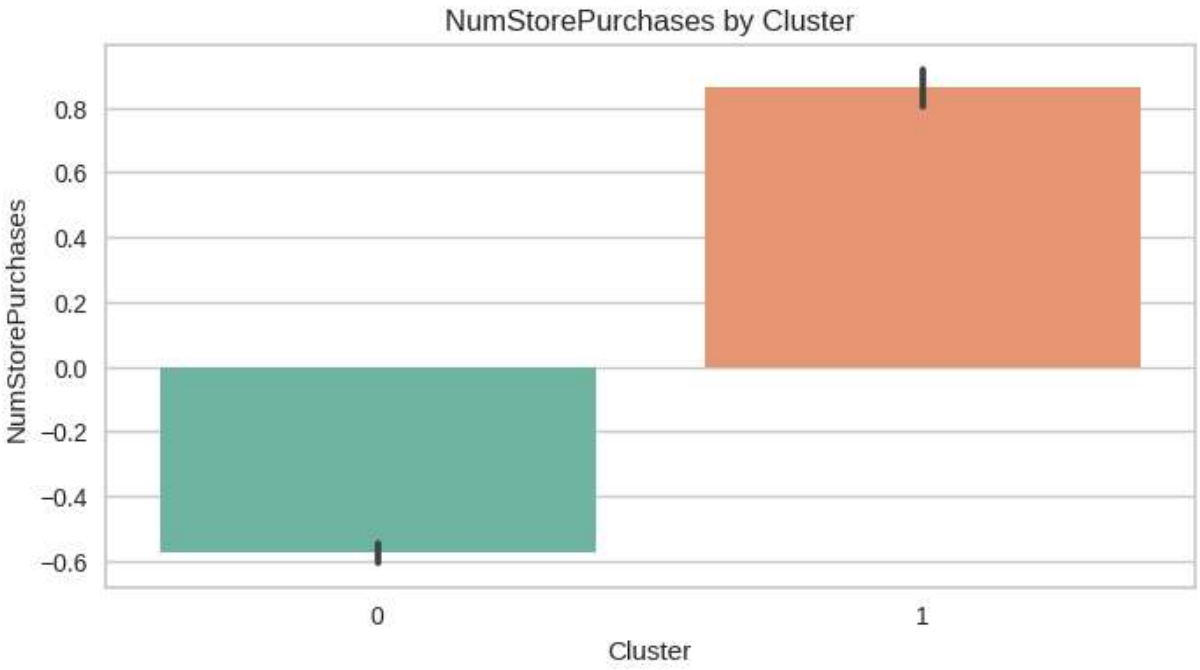

MntSweetProducts by Cluster
MntSweetProducts
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0
1
Cluster


MntGoldProds by Cluster
MntGoldProds
0.6
0.4
0.2
0.0
−0.2
−0.4
0
1
Cluster

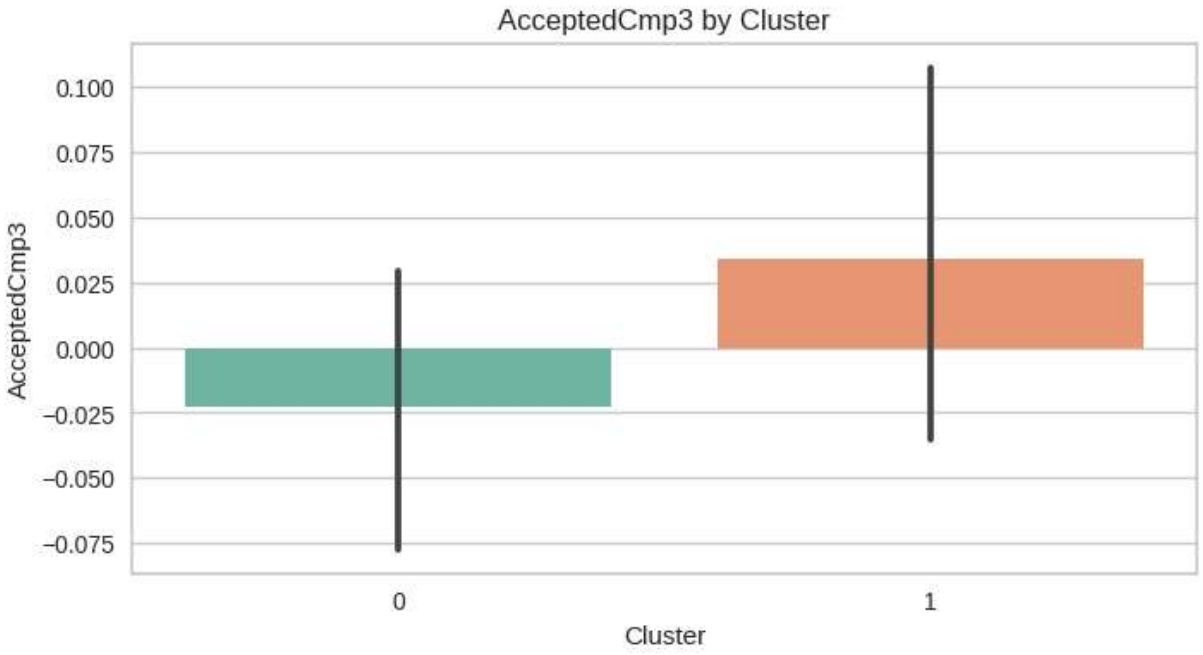

NumDealsPurchases by Cluster
NumDealsPurchases
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
0
1
Cluster

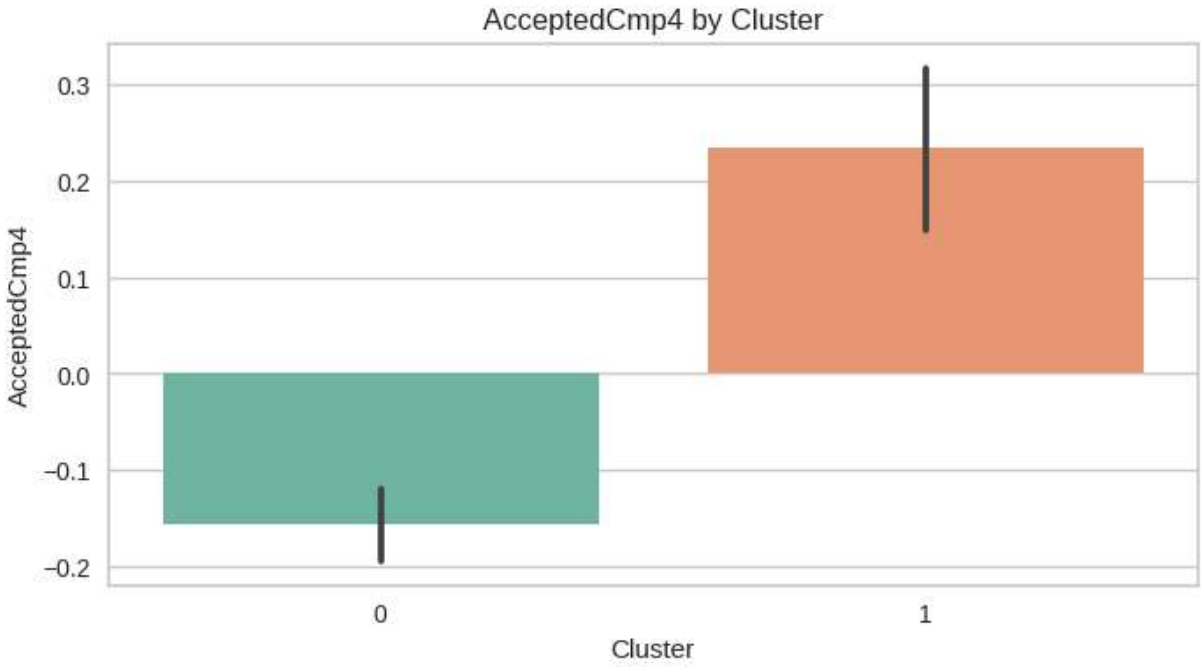

NumWebPurchases by Cluster
NumWebPurchases
0.6
0.4
0.2
0.0
−0.2
−0.4
0
1
Cluster

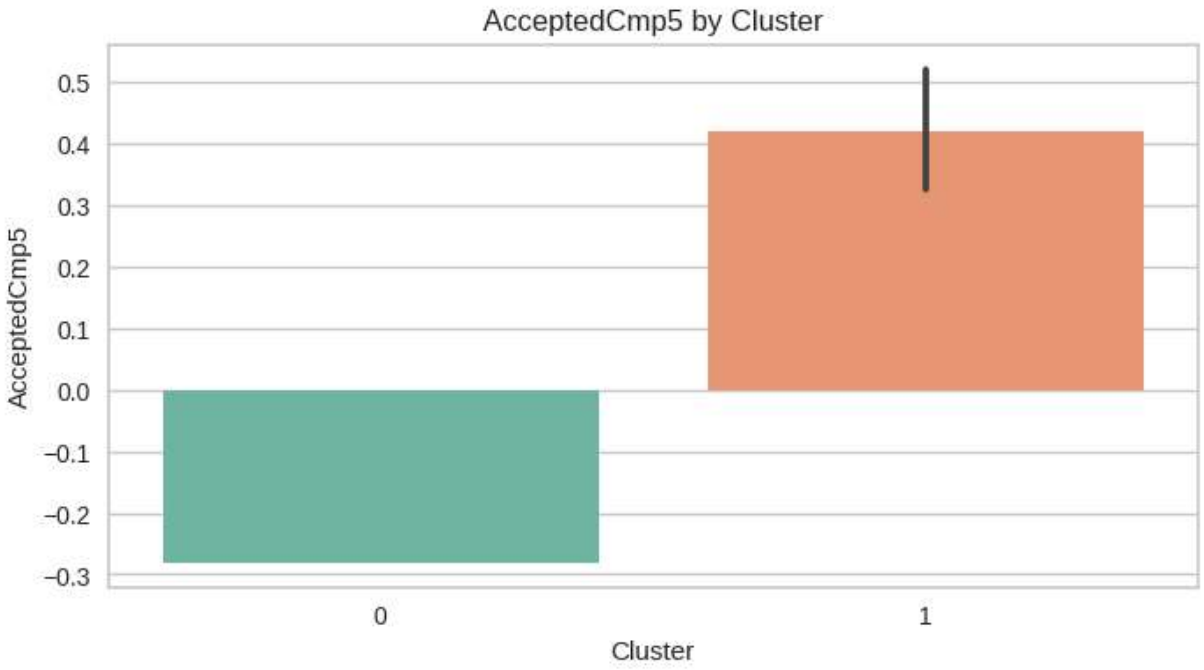

NumCatalogPurchases by Cluster
NumCatalogPurchases
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
0
1
Cluster

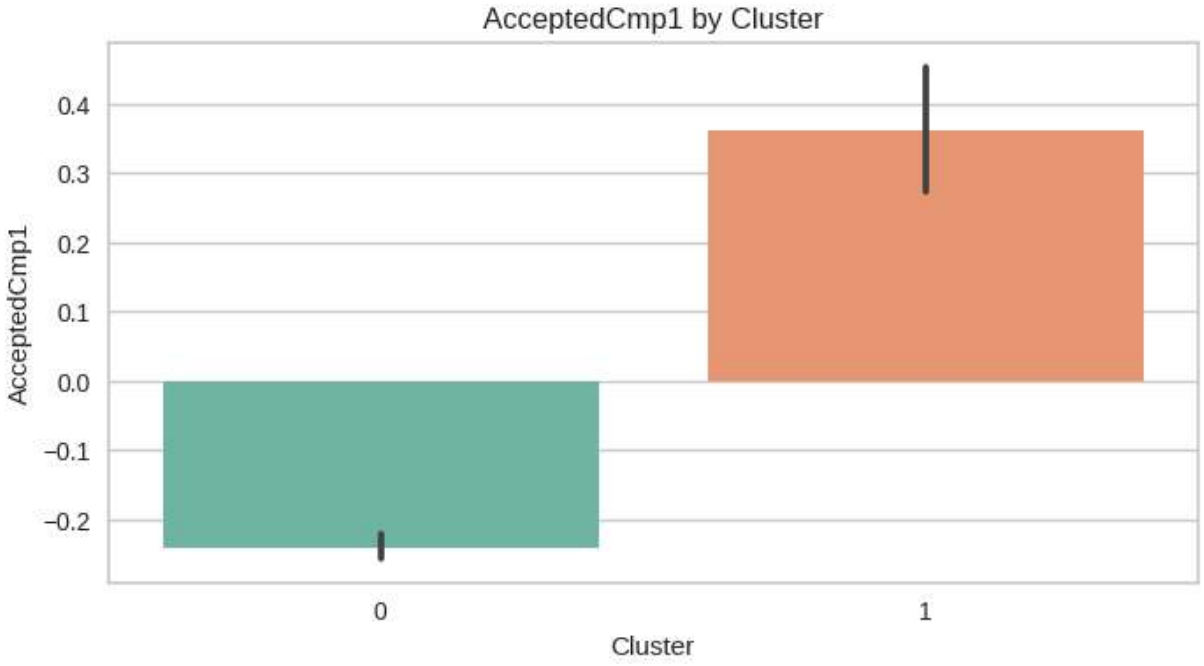

NumStorePurchases by Cluster
NumStorePurchases
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
0
1
Cluster

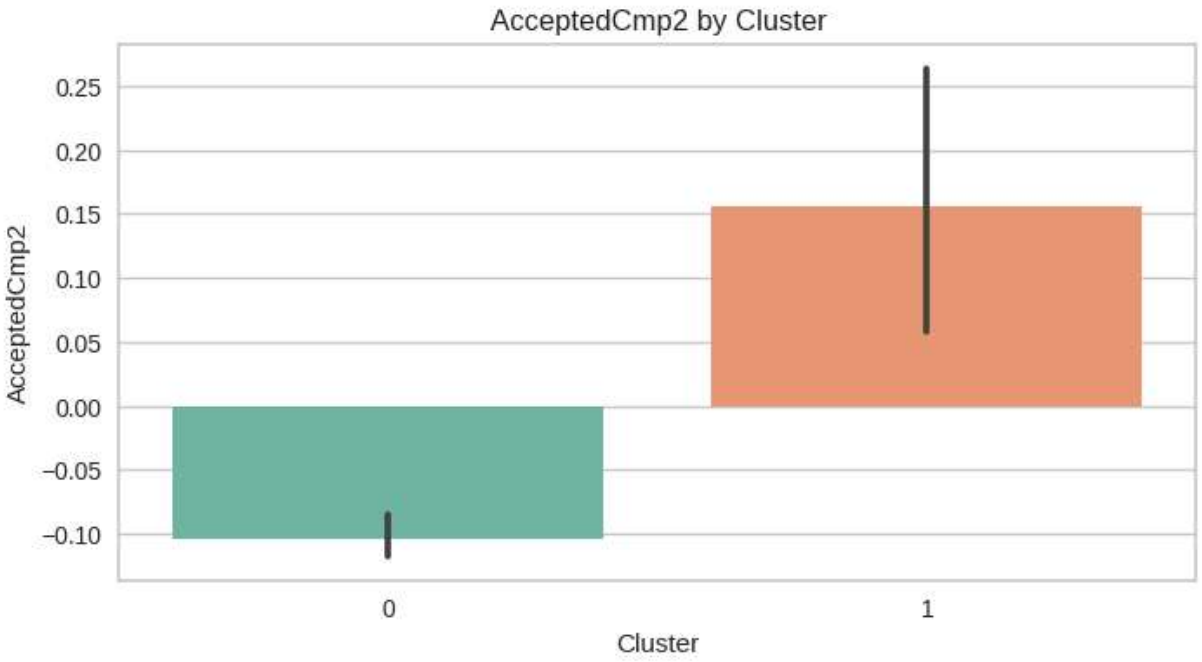

NumWebVisitsMonth by Cluster
NumWebVisitsMonth
0.4
0.2
0.0
−0.2
−0.4
−0.6
0
1
Cluster

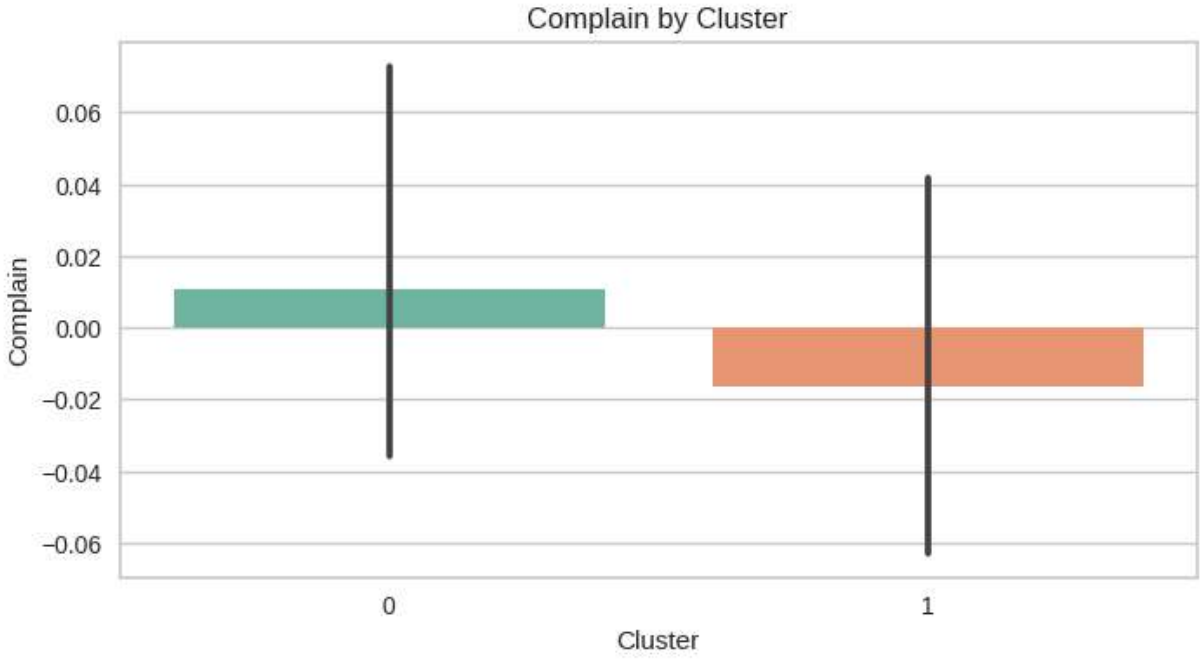

AcceptedCmp3 by Cluster
AcceptedCmp3
0.100
0.075
0.050
0.025
0.000
−0.025
−0.050
−0.075
0
1
Cluster

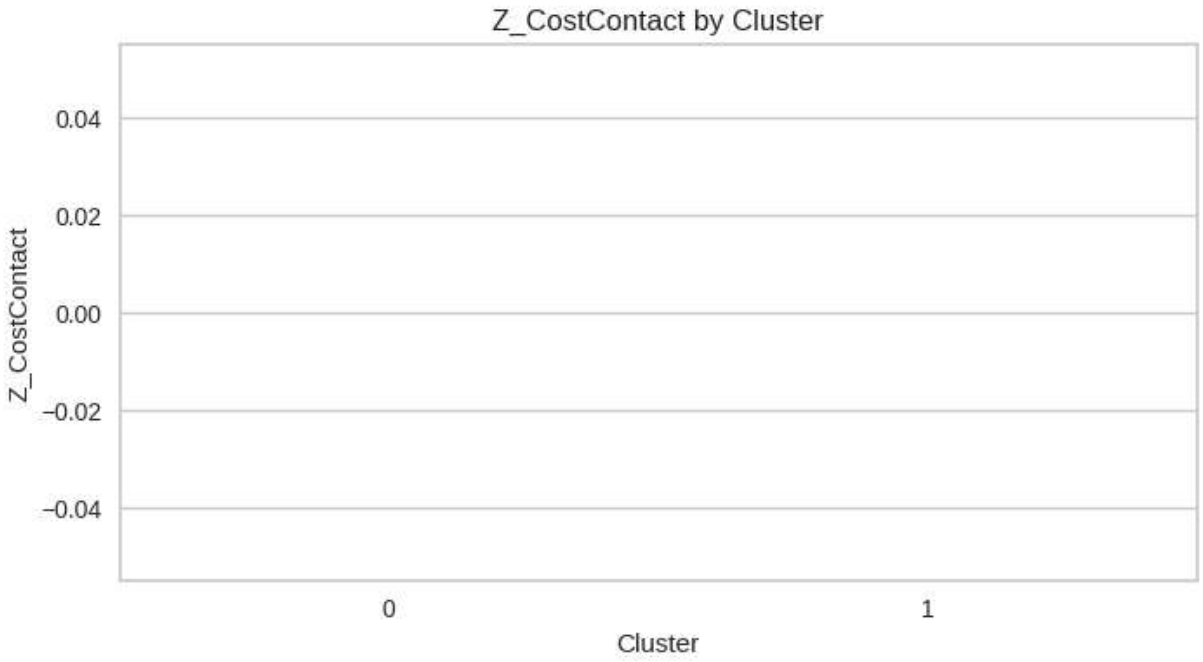

AcceptedCmp4 by Cluster
AcceptedCmp4
0.3
0.2
0.1
0.0
−0.1
−0.2
0
1
Cluster


AcceptedCmp5 by Cluster
AcceptedCmp5
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
0
1
Cluster

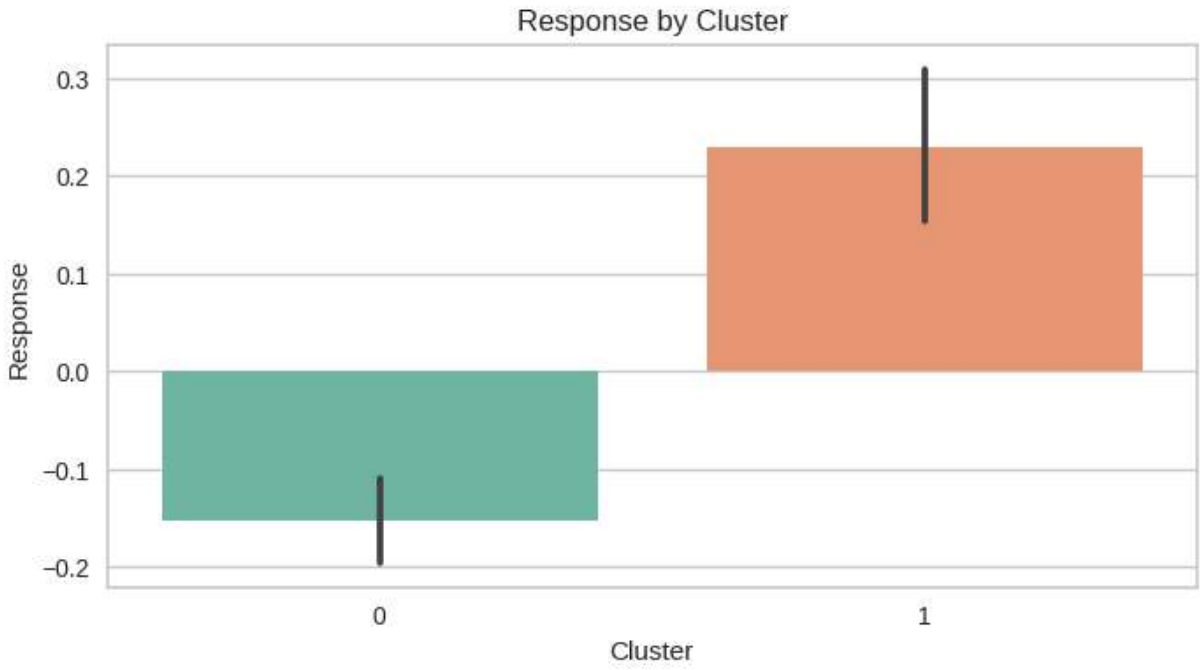

AcceptedCmp1 by Cluster
AcceptedCmp1
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
0
1
Cluster

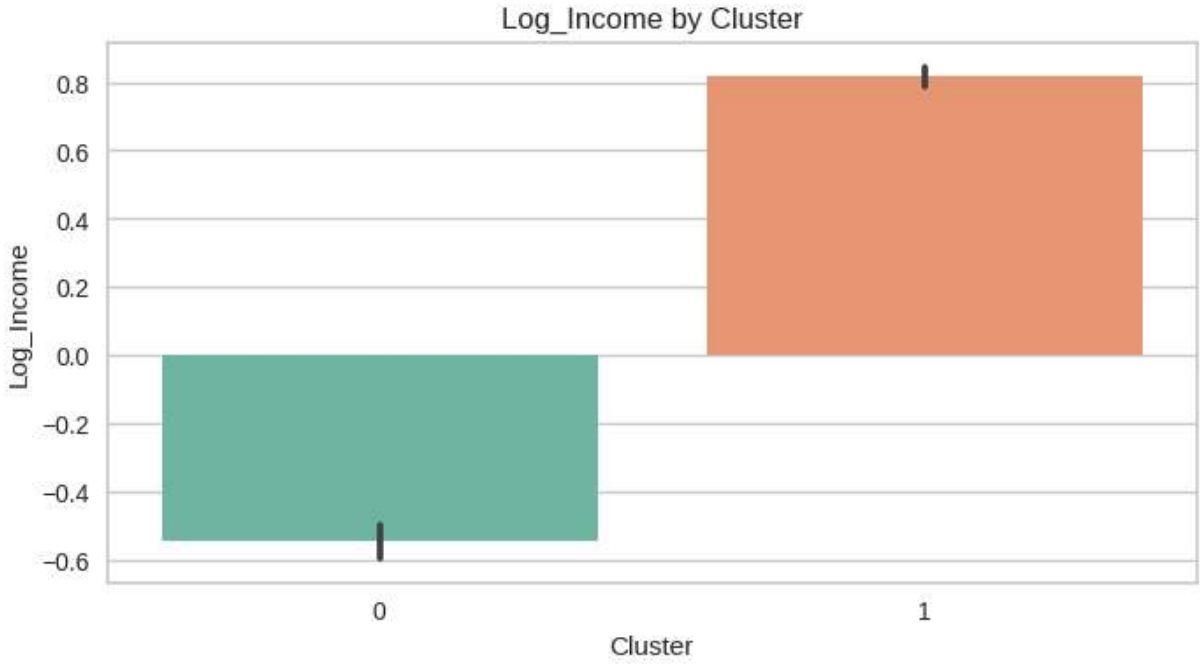

AcceptedCmp2 by Cluster
AcceptedCmp2
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
0
1
Cluster

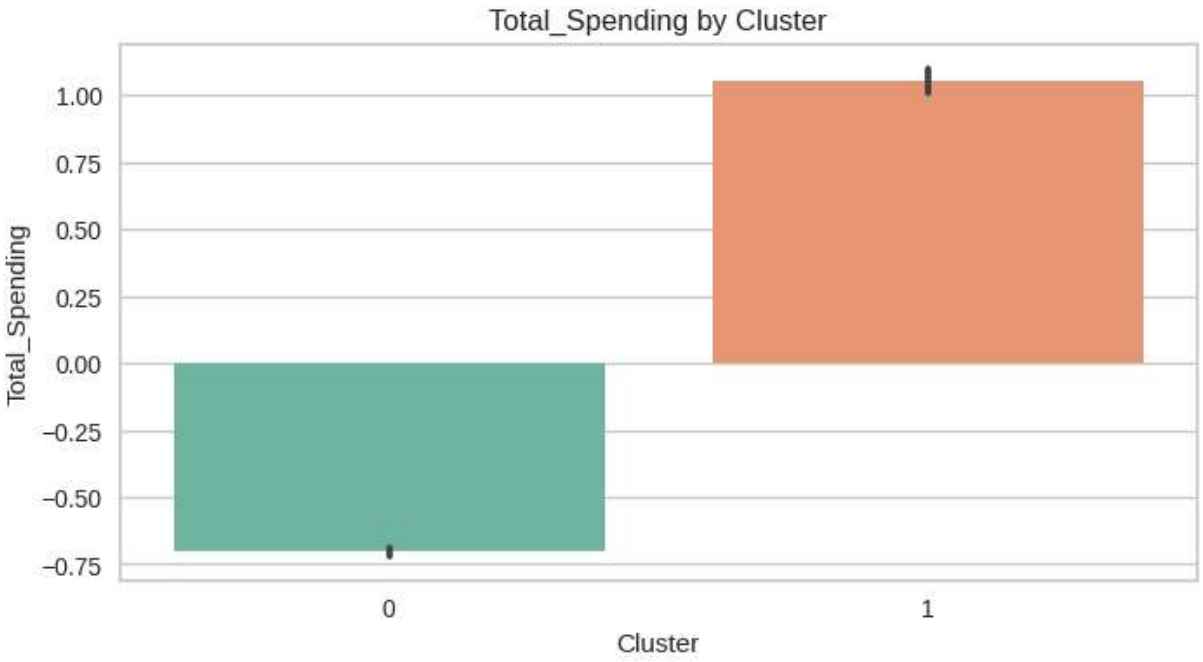

Complain by Cluster
Complain
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
0
1
Cluster


Z_CostContact by Cluster
Z_CostContact
0.04
0.02
0.00
−0.02
−0.04
0
1
Cluster


Z_Revenue by Cluster
Z_Revenue
0.04
0.02
0.00
−0.02
−0.04
0
1
Cluster


Response by Cluster
Response
0.3
0.2
0.1
0.0
−0.1
−0.2
0
1
Cluster


Log_Income by Cluster
Log_Income
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
0
1
Cluster




**Observations:**

# Business Recommedations

### **Question 12**: Based on the cluster insights, what business recommendations can be provided?

Based on cluster analysis, heatmap observations, and targeted visualizations, clear insights per segment can be drawn to align with the objective of developing personalized marketing campaigns to increase conversion rates, create effective retention strategies for high-value customers, and optimize resource allocation.

# Marketing Strategy Recommendations

**Deal-Based Targeting for Families:** Introduce personalized, limited-time online bundles tailored to families (ex: back-to-school, meal prep kits). Focus delivery via web & mobile for budget-conscious, family-oriented customers.

**Premium Positioning for Non-Parents / High-Spending Shoppers:** Target non-parents and high-spending catalog users with premium product advertising through catalog and in-store channels.

**Catalog-Centric Campaigns:** Invest in high-quality catalog content and design; leverage catalogs as a primary channel for high-value customers because catalog purchases strongly correlate with total spend.

**Cross-Channel Integration (Omnichannel Loyalty):** Integrate campaigns across web, catalog, and store - customers engaging in one channel are likely to engage in others. Use omnichannel loyalty programs to amplify lifetime value.

**Segmented Messaging Based on Responsibility:** Create concise, time-efficient ("snackable") campaign content for customers with kids/teens at home who have limited attention - avoid long, media-heavy formats.

**"Young Professional" / Luxury Lifestyle Branding:** Launch brand messaging for younger, higher-income customers around self-care, celebrations, and gifting to promote via web & social, including influencer partnerships.

**Deal Visibility / Bulk Promotion for Lower-Income Segments:** Highlight bulk savings and limited-time offers to price-sensitive customers. Use web/email to surface value packs.

**Trend-Based Targeting for Luxury Purchases:** Use lifestyle-oriented social media and targeted ads to reach young luxury shoppers, emphasizing premium wine/meat/sweets.

**Retargeting & Abandoned Cart Promotions:** Implement exit-intent popups, abandoned cart email triggers, and dynamic retargeting ads to convert high web-visit but low-purchase customers (abandoned cart browsers - web channel).

**Campaign Differentiation by Engagement History:** Use historical campaign engagement data to tailor future outreach; a customer who responded before is more likely to respond again.

**Re-Engagement Funnels:** After initial campaign response, progressively offer deeper tiers of incentives to keep high-engagement customers active.

**Referral Program Expansion:** Introduce referral incentives to broaden reach beyond the currently engaged base.

# Customer Retention & Targeting

**Loyalty Programs for Multi-Channel & Premium Buyers:** Create VIP or loyalty programs rewarding customers who purchase across channels (web, catalog, store) and those with premium cross-category spending patterns.

**Segment-Specific Targeting:**

1. Cluster 0 / Budget-Conscious Families: Focus on web and mobile - they browse frequently but convert less. Use deal-based retargeting, simplify messaging, and promote family-value bundles.

2. High-Spending Catalog Users: Engage through catalog + in-store with exclusive premium campaigns, loyalty perks, and curated bundle offers.

3. Young Luxury Shoppers: Retain via web & social with trend-based, lifestyle branding and personalized premium upsells.

4. Abandoned Cart Browsers: Re-engage primarily on web with UX improvements and urgency-based prompts to reduce churn.

**Behavior-Based Segmentation:** Use campaign acceptance history as a predictor - prior responders should be prioritized in retention pipelines.

**Value Reinforcement for Deal-Responsive Customers:** For lower-income or deal-using segments, reinforce that spending is efficient via personalized savings reminders and time-limited deals.

**Preventing Attrition Among Browsers:** Identify frequent visitors who don't convert and apply tailored retention nudges (ex: incentive popups, personalized reminders).

# Operational Improvements

**UX & Checkout Optimization:** Simplify the web checkout flow, reduce friction, and optimize mobile experience to improve conversion for high-visit but low-purchase segments.

**Catalog Optimization:** Refine catalog layout, content, and timing - since catalog engagement is linked to high spend, ensure premium content is targeted at non-deal seekers and high-value customers.

**Inventory & Product Bundling Strategy:**

1. Stock and promote cross-category premium products together (wine, meat, sweets, gold) for "premium lifestyle" buyers.

2. Design family-value packs and bulk-friendly bundles for budget-conscious families.

**Store Layout Adjustments:** For budget-conscious families that shop in-store, highlight "value corners" or end-caps with discounted essentials to display affordability.

**Channel-Focused Resource Allocation:**

1. Web & Mobile: Prioritize for budget-conscious families and abandoned-cart segments - improve targeting, loading speed, and personalization.

2. Catalog + In-Store: Prioritize for high-spending and premium segments with exclusive drops and curated premium bundle displays.

3. Web & Social: Prioritize for younger luxury shoppers with influencer-aligned campaigns and lifestyle product placements.

**Cross-Sell Enablement:** Use observed strong co-spending patterns to programmatically recommend complementary products (ex:, wine with meat or sweets) across channels.

**Time-Saving Offer Engineering:** For customers with high home responsibilities (those with kids/teens), develop operational flows like automated reorders, delivery discounts, and pre-packed offerings to reduce time cost.

# Future Recommendations:

More data features could provide more insights on customer purchasing habits by creating further in-depth relationships between added aspects of customers life to their purchasing habits. Here is a list of a few of these features:

1. Channel-Specific Purchase Type (Online vs. in-store, mobile vs. desktop) -> Crucial for tailoring marketing channels and clustering based on behavior.
2. Purchase Timing (Day of week, time of day, seasonality) -> Helps segment customers by lifestyle, schedule, and seasonal demand.
3. Basket Composition (Average items, product variety per order) -> Reveals bundle potential and lifestyle indicators (ex: families vs. singles).
4. Churn Risk Indicators (Drop in frequency, declining engagement, etc.) -> Essential for proactive retention strategies and dynamic cluster updates.
5. Price Sensitivity (Coupon usage, response to discounts) -> Helps identify value-driven vs. quality-driven customers.
6. Customer Service Interactions (Complaints, returns, feedback) -> Allows clustering based on satisfaction and service needs.
7. Web Analytics (Session duration, pages viewed, scroll depth) -> Improves understanding of abandoned carts and digital hesitation.
8. Referral & Social Influence (Referrals, social media tagging, reviews) -> Segments based on brand advocacy and word-of-mouth potential.
9. Email Engagement Metrics (Open rates, click-throughs, campaign preferences) -> Supports smarter campaign design and engagement-based clustering.
10. Return Behavior (Frequency, timing, reason for returns) -> Indicates satisfaction, product fit, and logistical preferences.
11. Life Events (New child, retirement, marriage - opt-in data) -> Enables lifecycle marketing and long-term engagement strategy.

12. Proximity to Store / Regional Analysis (ZIP code clustering, rural vs. urban) -> Useful for tailoring store-based campaigns and local offers.

13. Payment Method (Mobile wallet, credit vs. debit, loyalty points use) -> Can reveal digital savviness, income levels, or preferences.

14. Occupation or Work Type (Remote vs. in-office vs. shift work) -> Helps time campaigns or target needs (e.g., fast prep meals for nurses).

15. In-store Location Tracking (Time in store, aisle heat maps) -> Supports store layout optimization but requires advanced infrastructure.

16. Outdoor vs. Indoor Buying Context (Tied to weather, outdoor events) -> Helps build seasonal campaigns but has limited daily impact.

17. Hobbies/Interests (via surveys or behavior) (Fitness, cooking, sustainability) -> Best for premium personalization but hard to collect without opt-in.