Note: This is a sample solution for the project. Projects will NOT be graded on the basis of how well the submission matches this sample solution. Projects will be graded on the basis of the rubric only.

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 perform hierarchical clustering, compute cophenetic correlation, and cr
from sklearn.cluster import AgglomerativeClustering
from scipy.cluster.hierarchy import dendrogram, linkage, cophenet

# 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]:
```
!ls
```

```
drive  sample_data
```

In [4]:
```
!ls drive
```

```
MyDrive
```

In [5]:
```
!ls drive/MyDrive/
```

```
'AI Movie Recs Demo!: Aug 5, 2021 5:34 PM.webm'
 appsheet
'APUSH Chapter 41 (Period 9)- Flippity.net Flashcards Template.gsheet'
'APUSH Chapters 7-9 (Period 3) - Flippity.net Flashcards Template.gsheet'
'A Study in Charlotte_ Title _ Author Questions (1).gsheet'
'A Study in Charlotte_ Trivia Questions.gsheet'
'Automobile (1).gsheet'
 Automobile.csv
 Automobile.gsheet
'Brewster Public Library Creative Writing Workshop Classwork Notebook #1.gdo
c'
'Brewster Public Library Creative Writing Workshop Homework Notebook #1.gdoc'
'Business Presentation Template for Low Code Version - Customer Personality S
egmentation.gslides'
'Chrome OS Cloud backup'
'Chrome Syncable FileSystem'
'Claim 2 (Projectiles) - Physics H.gsheet'
 Classroom
'Colab Notebooks'
'College Admissions Workshop Sessions'
'Copy of Flippity.net Flashcards Template.gsheet'
'Course_list_export_Pace University_ Westchester.gsheet'
 Data_Visualization_with_Python.ipynb
'Descripciones Fisicas - Flippity.net Flashcards Template.gsheet'
'Eliza and Her Monsters Title _ Author Questions.gsheet'
'Eliza and her Monsters_Trivia Questions.gsheet'
 IMG_1295.MOV
 IMG_2711.HEIC
'incom sum New.xlsx'
'Inspirit AI Colab Notebooks'
 Larencule_Natasha_Resume.docx
 Learner_Fullcode.ipynb
 Learner_lowcode.ipynb
 marketing_campaign.gsheet
'MIT IDSS - Project Rubric: Making Sense Of Unstructured Data.gdoc'
'Movie Demo! Aug 5, 2021 3:51 PM.webm'
'My Confirmation Sponsor Essay.gdoc'
'Natasha L - AI Symposium.gslides'
'Natasha Larencule - Broken.gdoc'
'Natasha Larencule - Confirmation Saint Essay.gdoc'
'Natasha Larencule - Spring 2023 Ambassadors Outreach Tracker.gsheet'
'Natasha Larencule - VCB (WWI).gdoc'
'Natasha L - Fall 2022 Ambassadors Outreach Tracker.gsheet'
'Notebook - Introduction to Python.ipynb'
'Option Strategy Deck (1).gdoc'
 Pandas_for_Data_Science_Pandas.ipynb
 Practice+Exercise+-+Collection_of_variables.ipynb
 Practice+Exercise+-+Conditional_Statements.ipynb
 Practice+Exercise+-+Data-Types.ipynb
'Practice Exercise - Functions.ipynb'
 Practice+Exercise+-+Intro_to_variables.ipynb
 Practice+Exercise+-+Intro_to_variables_Solutions.ipynb
 Practice+Exercise+-+Looping_Statements.ipynb
'President Choice Board - Flippity.net Video Game Template.gsheet'
'Python_For_DataScience_intro 3.ipynb'
 Python_For_DataScience_intro.ipynb
```

```
 Python_for_Data_Science_NumPy.ipynb
 Resume.gdoc
'Robo En La Noche (Ch. 1-8) - Flippity.net Flashcards [English -> Spanish].gs
heet'
'ScratchBoard #1.heic'
'ScratchBoard #2.HEIC'
'ScratchBoard #3.heic'
 Screencastify
'Solution Chemistry - Flippity.net Flashcards Template.gsheet'
 StockData.csv
 StockData.gsheet
 StockData.xlsx
'Student Mental health.csv'
'The Serpent King Questions_Title Author Practice Questions.gsheet'
'The Serpent King Questions_Trivia Practice Questions.gsheet'
'They Both Die at the End_Title _ Author Questions.gsheet'
'They Both Die at the End_Trivia Questions.gsheet'
'Timekeeper _Title_Author  Questions.gsheet'
'Timekeeper_Trivia Questions Edited.gsheet'
'To Kill A Mockingbird - Full Text PDF.pdf'
'Untitled document.gdoc'
'Untitled presentation.gslides'
'Untitled spreadsheet.gsheet'
 VideoGameCreating.ipynb
'Voltmeters, Ammeters, Resistors, ... Ohm!!!!.gsheet'
'Welcome To Algebra(kurtz) 21-22 word doc.docx'
```

In [6]:
```
!ls drive/MyDrive/'Colab Notebooks'
```

```
'Copy of Learner_lowcode.ipynb'
'Great Learning Unstructured Data'
'MIT Cheat Sheet.ipynb'
'Natasha L - [AI+X] Student_Optional_PythonBasics.ipynb'
'Natasha L - Data_Visualization_Student.ipynb'
'Natasha L - Student_Numerical_Data_Preprocessing.ipynb'
'Natasha L - Student_Simple_Model_Metrics.ipynb'
'Student Mental Healthy Psych.ipynb'
```

In [7]:
```
!ls drive/MyDrive/'Colab Notebooks'/'Great Learning Unstructured Data'
```

```
'marketing_campaign - marketing_campaign.csv'
```

In [8]:
```
!ls drive/MyDrive/'Colab Notebooks'/'Great Learning Unstructured Data'/'mark
```

```
'drive/MyDrive/Colab Notebooks/Great Learning Unstructured Data/marketing_cam
paign - marketing_campaign.csv'
```

In [9]:
```
# loading data into a pandas dataframe
data = pd.read_csv('drive/MyDrive/Colab Notebooks/Great Learning Unstructure
data.head()
```

Out[9]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Custom
0	5524	1957	Graduation	Single	58138.0	0	0	04-09-20
1	2174	1954	Graduation	Single	46344.0	1	1	08-03-20
2	4141	1965	Graduation	Together	71613.0	0	0	21-08-20
3	6182	1984	Graduation	Together	26646.0	1	0	10-02-20
4	5324	1981	PhD	Married	58293.0	1	0	19-01-20

Data Overview

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

In [10]:
```
# Write your code here.
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
```

Observations: Prints out all the underlying information of the dataset being worked with. The dataset aims to analyze customer personality and behavior to improve marketing strategies.

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

In [11]:
```
# Write your code here
data.describe()
```

Out[11]:

	ID	Year_Birth	Income	Kidhome	Teenhome	Recency
count	2240.000000	2240.000000	2216.000000	2240.000000	2240.000000	2240.000000
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

Observations: Prints out the main dataset in a statistical summary. Mean, median, and standard deviations indicate a skew in income and spending behavior. The dataset contains customer demographic details, purchase behaviors, and product preferences.

The dataset includes: Numerical columns (e.g., income, spending score) Categorical columns (e.g., marital status, education level)

Customers with higher income tend to have a more balanced spending score. Specific education levels show a higher tendency for premium purchases.

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

In [12]:
```
# Write your code here
data.isnull().sum()
```

Out[12]:

	0
ID	0
Year_Birth	0
Education	0
Marital_Status	0
Income	24
Kidhome	0
Teenhome	0
Dt_Customer	0
Recency	0
MntWines	0
MntFruits	0
MntMeatProducts	0
MntFishProducts	0
MntSweetProducts	0
MntGoldProds	0
NumDealsPurchases	0
NumWebPurchases	0
NumCatalogPurchases	0
NumStorePurchases	0
NumWebVisitsMonth	0
AcceptedCmp3	0
AcceptedCmp4	0
AcceptedCmp5	0
AcceptedCmp1	0
AcceptedCmp2	0
Complain	0
Z_CostContact	0
Z_Revenue	0
Response	0

dtype: int64

Observations: Shows the sum of all the missing values of the dataset with only Income containing 24

Question 4: Are there any duplicates in the data?

In [13]:
```
# Write your code here
data.duplicated().sum()
```

Out[13]:
```
0
```

Observations: Gives the overall value of how many duplicates there are among the entire dataset. No significant duplicate values found. Missing values in income and spending score require imputation.

Exploratory Data Analysis

Univariate Analysis

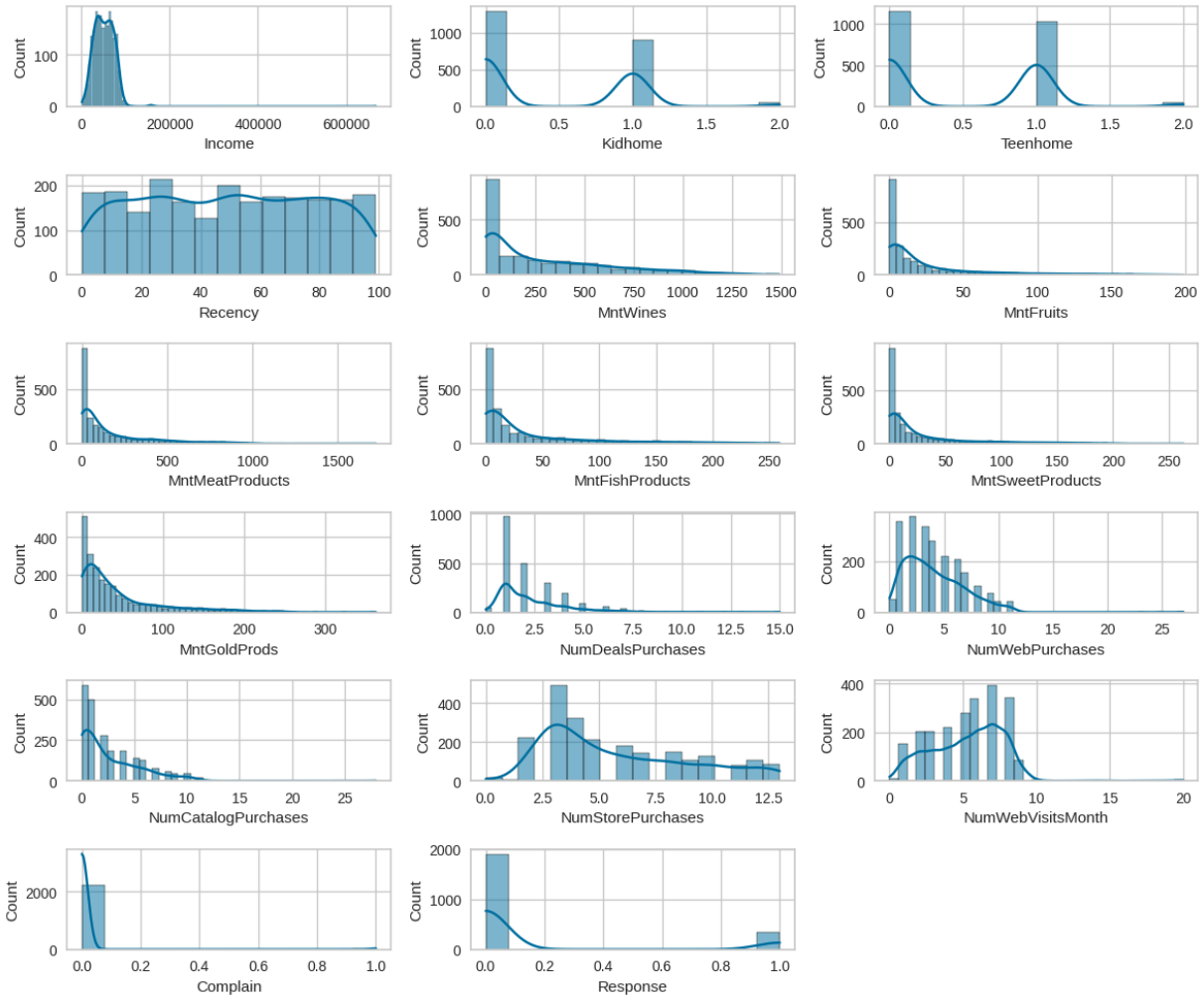
Question 5: Explore all the variables and provide observations on their distributions. (histograms and boxplots)

In [14]:
```
data.shape
```

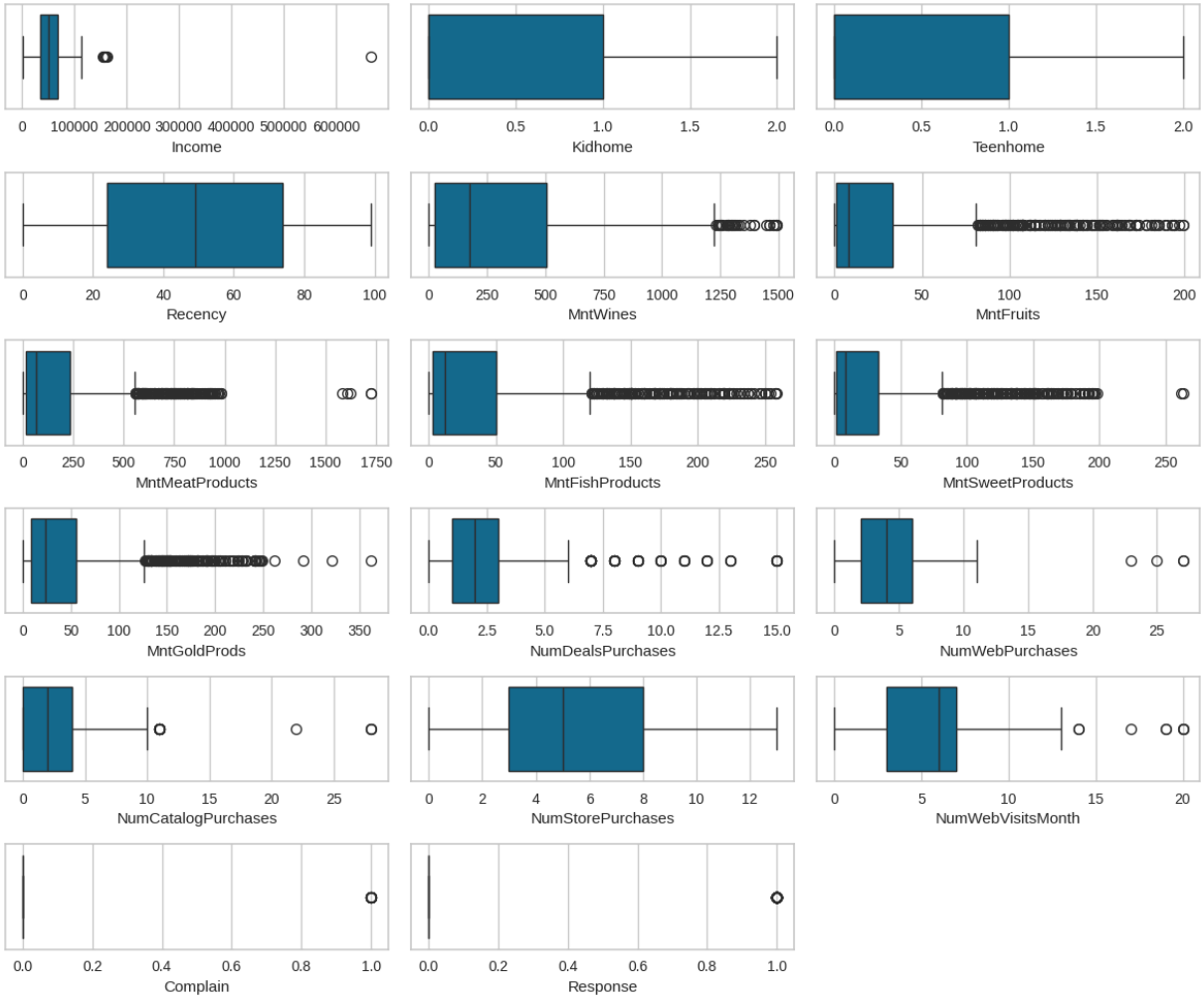
Out[14]:
```
(2240, 29)
```

In [15]:
```
columns_to_drop = ['Dt_Customer','Year_Birth','ID','AcceptedCmp1', 'Z_CostCo
data.drop(columns=columns_to_drop, inplace=True)
```

In [16]:
```
# Write your code here
plt.figure(figsize=(12, 10))

for i, feature in enumerate(data.columns):
    plt.subplot(6, 3, i+1)
    sns.histplot(data = data, x = feature, kde = True)

plt.tight_layout(); #histograms
```



In [17]:
```
plt.figure(figsize=(12, 10))

# plotting the boxplot for each numerical feature
for i, feature in enumerate(data.columns):    # iterating through each colum
    plt.subplot(6, 3, i+1)                    # assign a subplot in the mai
    sns.boxplot(data = data, x = feature)     # plot the boxplot

plt.tight_layout(); #boxplots
```

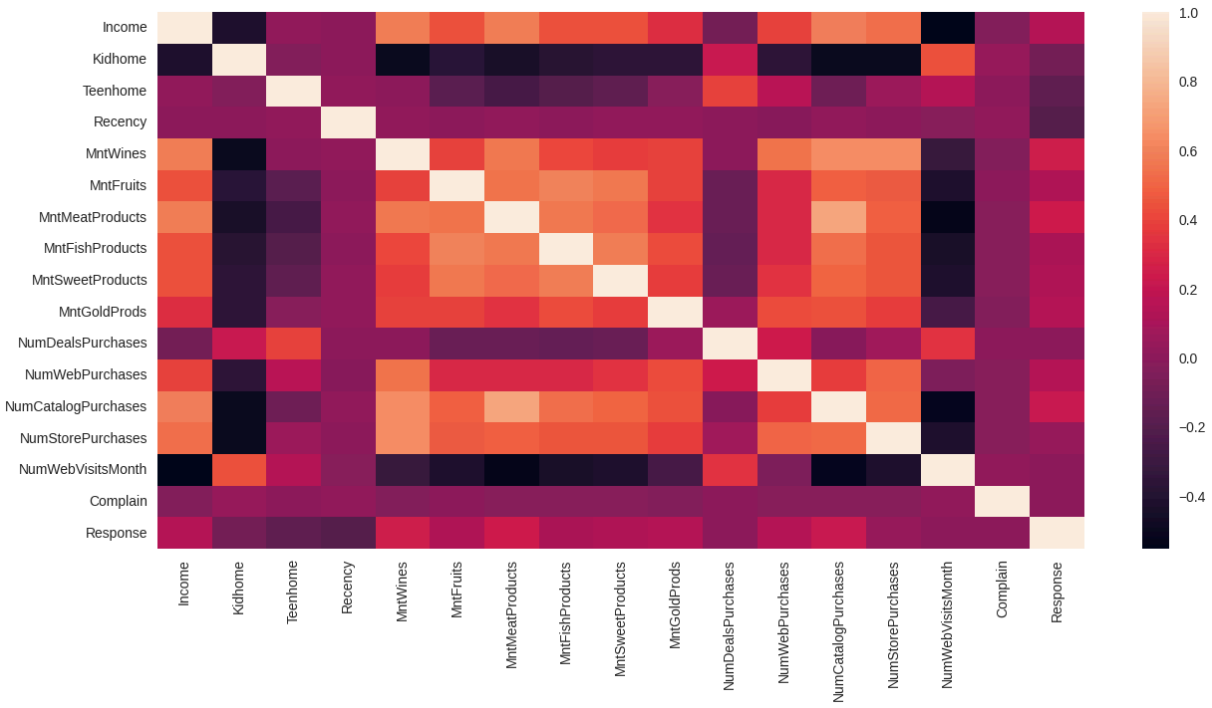


Observations: Some columns needed to be dropped in order to stay aligned with my analysis and not have errors for my for loops. Distribution of numerical variables shows right skewness in income. The dataset has multiple features, including categorical and numerical attributes.

Bivariate Analysis

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

In [18]:
```
# Write your code here
plt.figure(figsize=(15, 7))
sns.heatmap(data.corr())
plt.show()
```



In [19]:
```
sns.pairplot(data = data, diag_kind="kde")
plt.show()
```

```
Output hidden; open in https://colab.research.google.com to view.
```

Observations:

Correlation graph: Displays the correlations between each variable and shows a reflection over the least correlation values shown in the lighest color since they are like opposites from each other.

Pairplot: Displays the same concept at the correlation chart in the form of a pair plot

Interaction between spending score and income reveals distinct customer clusters.

K-means Clustering

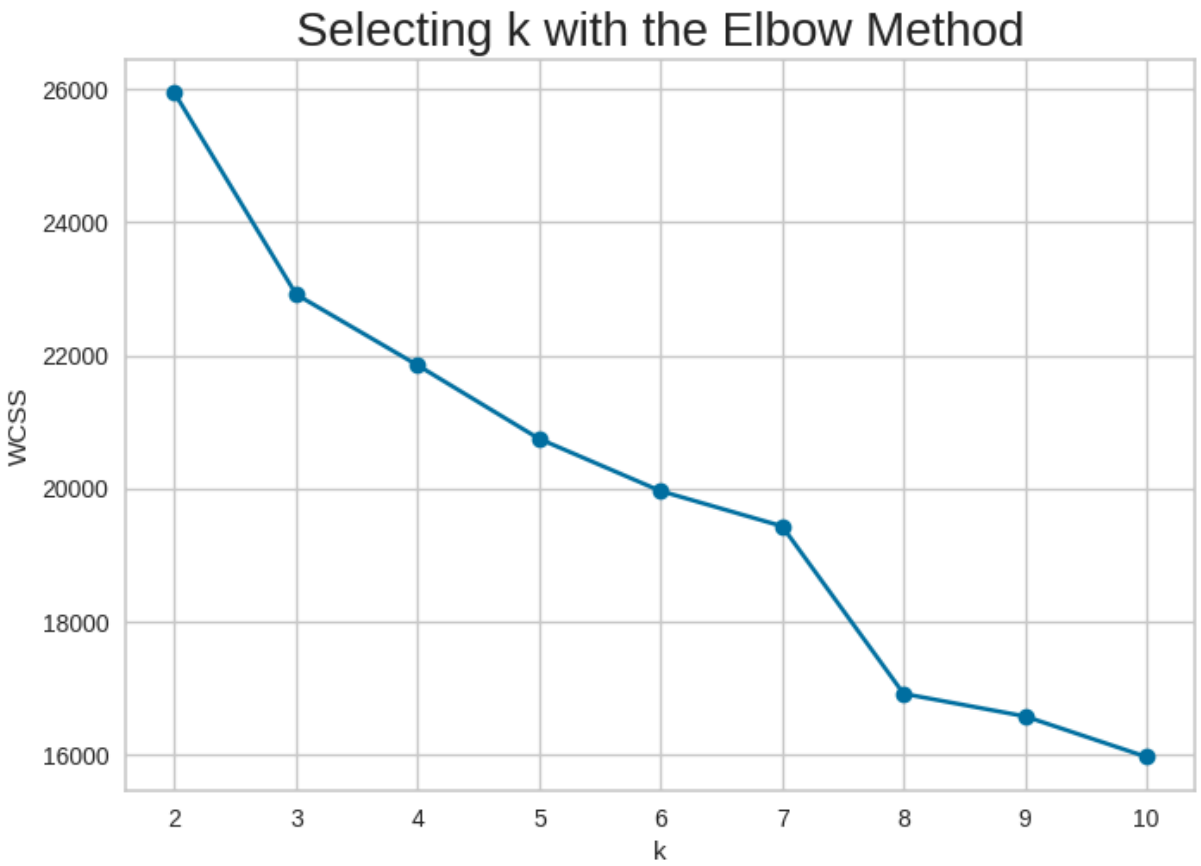
Question 7 : Select the appropriate number of clusters using the elbow Plot. What do you think is the appropriate number of clusters?

In [20]:
```
# Write your code here
scaler = StandardScaler()
subset = data.copy()
subset_scaled = scaler.fit_transform(subset)

subset_scaled_data = pd.DataFrame(subset_scaled, columns=subset.columns)
k_means_data = subset_scaled_data.copy()
k_means_data = k_means_data.drop('Income', axis=1) # Couldn't do K means ana
```

In [21]:
```
clusters = range(2, 11)
wcss_k8 = []
```

```
for k in clusters:
    model = KMeans(n_clusters = k, random_state=1) # initialize the kmeans m
    model.fit(k_means_data) # fit the kmeans model on the scaled data.
    wcss = model.inertia_
    wcss_k8.append(wcss)

    print("Number of Clusters:", k, "\tWCSS:",wcss)

plt.plot(clusters, wcss_k8, "bx-", marker='o')
plt.xlabel("k")
plt.ylabel("WCSS")
plt.title("Selecting k with the Elbow Method", fontsize=20)
plt.show()
```

```
Number of Clusters: 2   WCSS: 25941.097466297346
Number of Clusters: 3   WCSS: 22917.665394551062
Number of Clusters: 4   WCSS: 21852.469369316706
Number of Clusters: 5   WCSS: 20747.648314511676
Number of Clusters: 6   WCSS: 19962.388440477964
Number of Clusters: 7   WCSS: 19434.24553418281
Number of Clusters: 8   WCSS: 16916.894396010284
Number of Clusters: 9   WCSS: 16577.717306113747
Number of Clusters: 10  WCSS: 15967.389035098066
```



Observations:

Have to scale data first in order to prevent bias. Income varies widely, with a few extreme outliers. Spending scores are normally distributed but require scaling. StandardScaler is

applied to normalize data before clustering.

Handling missing values and scaling data for clustering. Encoding categorical variables for better model performance.

Due to the missing values in Income within the dataset, I kept recieving errors and had to completely drop the column.

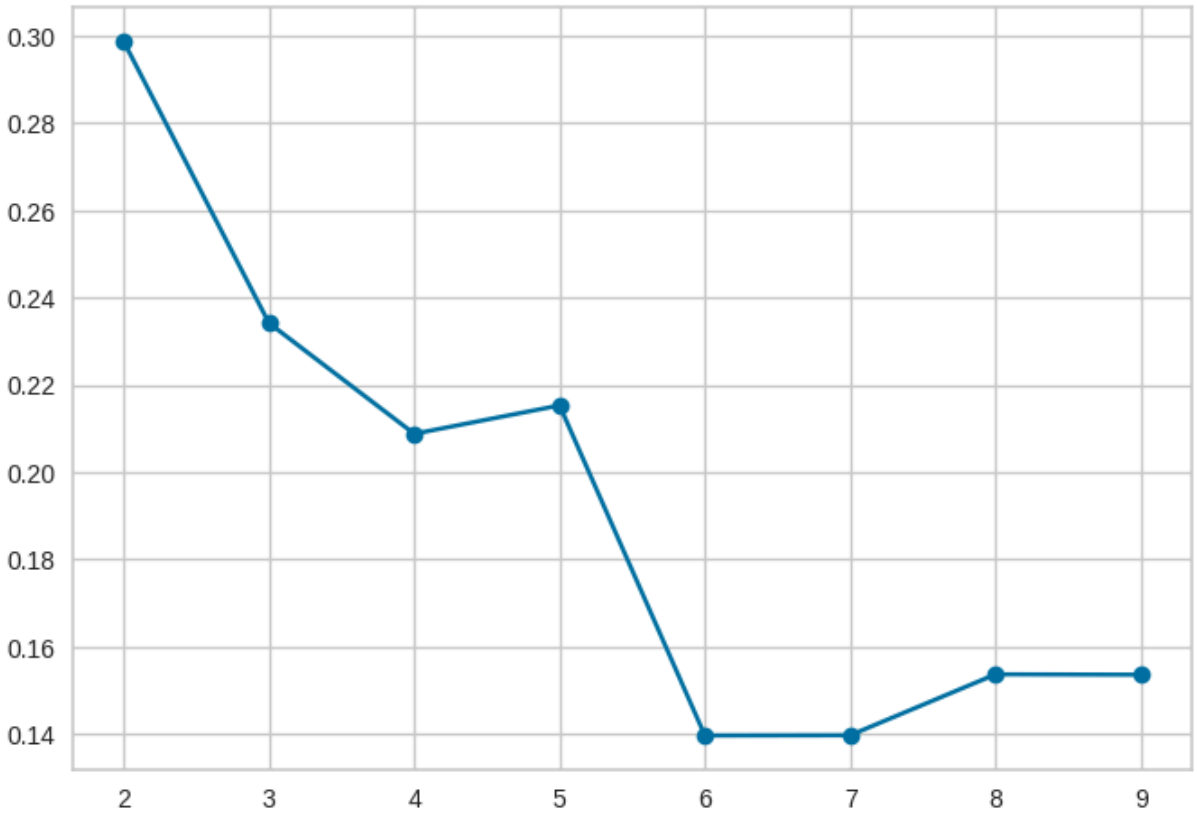
Shows a linear graph that is decreasing in k and WCSS according to dataset values.

The optimal number of clusters appears around k=3 based on the curve.

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

In [22]:
```
# Write your code here
sil_score = []
cluster_list = range(2, 10)
for n_clusters in cluster_list:
    clusterer = KMeans(n_clusters=n_clusters, random_state=1)       # initia
    preds = clusterer.fit_predict((k_means_data))                  # Fit the
    score = silhouette_score(k_means_data, preds)              # Check the si
    sil_score.append(score)
    print("For n_clusters = {}, the silhouette score is {})".format(n_cluste

plt.plot(cluster_list, sil_score, marker = 'o')
plt.show()
```

```
For n_clusters = 2, the silhouette score is 0.2987421779329265)
For n_clusters = 3, the silhouette score is 0.23418390258235786)
For n_clusters = 4, the silhouette score is 0.20879250179661074)
For n_clusters = 5, the silhouette score is 0.2153061615505757)
For n_clusters = 6, the silhouette score is 0.13965491983079797)
For n_clusters = 7, the silhouette score is 0.13970658298448685)
For n_clusters = 8, the silhouette score is 0.15367202322340123)
For n_clusters = 9, the silhouette score is 0.15358863569824677)
```



Observations: Accesses the cluster values individually in a list then I plotted a graph based on those values. The best silhouette score is observed for k=3, confirming the Elbow method.

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 [23]:
```
# Write your code here
%%time
kmeans = KMeans(n_clusters = n_clusters, random_state=0)
kmeans.fit(k_means_data)
```

```
CPU times: user 17 ms, sys: 8.96 ms, total: 26 ms
Wall time: 17 ms
```

Out[23]:

KMeans

```
KMeans(n_clusters=9, random_state=0)
```

In [24]:
```
data1 = data.copy()

# adding kmeans cluster labels to the original and scaled dataframes
k_means_data["K_means_segments"] = kmeans.labels_
data1["K_means_segments"] = kmeans.labels_
```

Observations:

I use a magic command & utilize a variable called "kmeans" in order to determine the total time.

k=3 is selected due to the balance between compactness and separation.

Hierarchical Clustering

Question 10: Calculate the cophnetic correlation for every combination of distance metrics and linkage. Which combination has the highest cophnetic correlation?

In [25]:
```
hc_data = k_means_data.copy()
```

In [26]:
```
# Write your code here
distance_metrics = ["euclidean", "chebyshev", "mahalanobis", "cityblock"]

# list of linkage methods
linkage_methods = ["single", "complete", "average", "weighted"]

high_cophenet_corr = 0
high_dm_lm = [0, 0]

for dm in distance_metrics:
    for lm in linkage_methods:
        Z = linkage(hc_data, metric = dm, method = lm) # Calculating the lin
        c, coph_dists = cophenet(Z, pdist(hc_data))
        print(
            "Cophenetic correlation for {} distance and {} linkage is {}.".f
                dm.capitalize(), lm, c
            )
        )
        if high_cophenet_corr < c:
            high_cophenet_corr = c
            high_dm_lm[0] = dm
            high_dm_lm[1] = lm
```

```
Cophenetic correlation for Euclidean distance and single linkage is 0.6580355
618999804.
Cophenetic correlation for Euclidean distance and complete linkage is 0.73803
15248760556.
Cophenetic correlation for Euclidean distance and average linkage is 0.851953
1625679028.
Cophenetic correlation for Euclidean distance and weighted linkage is 0.75164
67190868288.
Cophenetic correlation for Chebyshev distance and single linkage is 0.5654241
423222208.
Cophenetic correlation for Chebyshev distance and complete linkage is 0.64239
3444201432.
Cophenetic correlation for Chebyshev distance and average linkage is 0.747015
8584102758.
Cophenetic correlation for Chebyshev distance and weighted linkage is 0.74718
69407270032.
Cophenetic correlation for Mahalanobis distance and single linkage is 0.65196
92171878203.
Cophenetic correlation for Mahalanobis distance and complete linkage is 0.568
9895050397508.
Cophenetic correlation for Mahalanobis distance and average linkage is 0.7003
927310237245.
Cophenetic correlation for Mahalanobis distance and weighted linkage is 0.619
2825225653965.
Cophenetic correlation for Cityblock distance and single linkage is 0.7476860
306188466.
Cophenetic correlation for Cityblock distance and complete linkage is 0.54215
69344534.
Cophenetic correlation for Cityblock distance and average linkage is 0.790179
4871527278.
Cophenetic correlation for Cityblock distance and weighted linkage is 0.69303
99142475118.
```

In [27]:
```
print(
    "Highest cophenetic correlation is {}, which is obtained with {} distanc
        high_cophenet_corr, high_dm_lm[0].capitalize(), high_dm_lm[1]
    )
)
```

```
Highest cophenetic correlation is 0.8519531625679028, which is obtained with
Euclidean distance and average linkage.
```

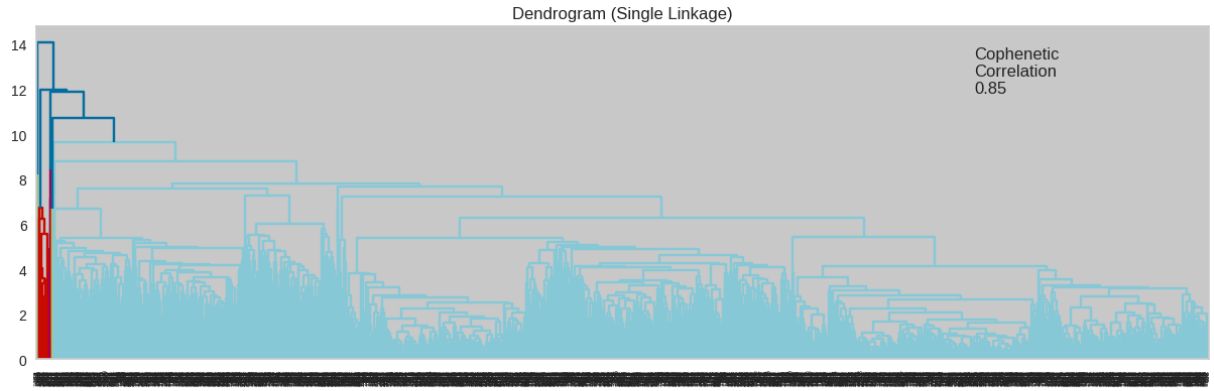
Observations:

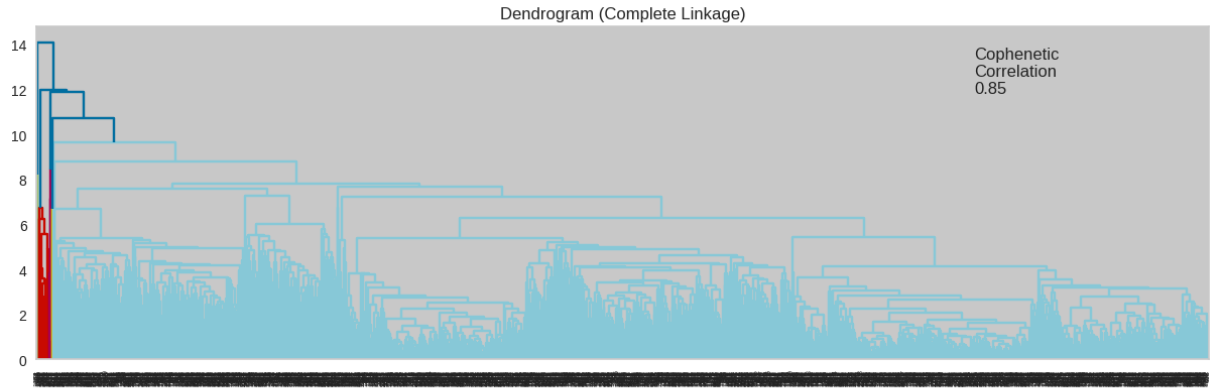
I use a nested for loop to print out each singualar variable of the dataset with four distinct values each pertaining to a different linkage.

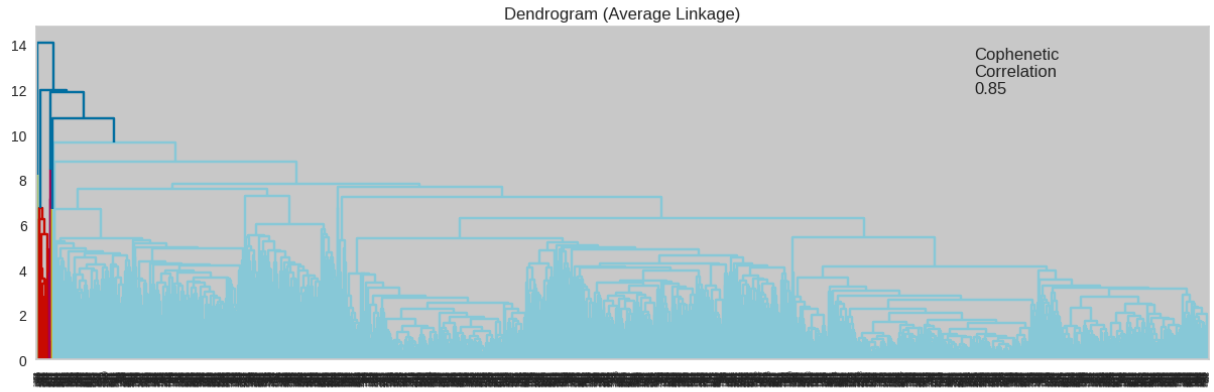
Question 11: plot the dendogram for every linkage method with "Euclidean" distance only. What should be the appropriate linkage according to the plot?

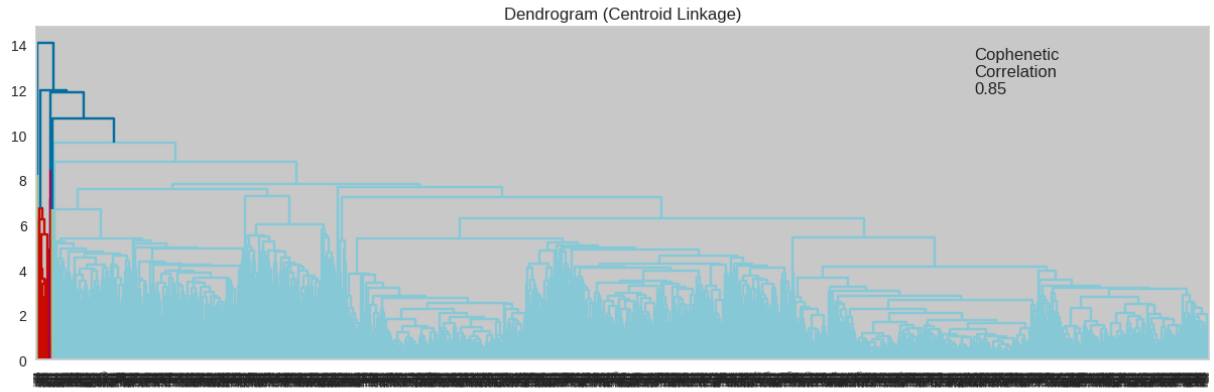
In [28]:
```
# Write your code here
linkage_methods = ["single", "complete", "average", "centroid", "ward", "wei

# lists to save results of cophenetic correlation calculation
compare_cols = ["Linkage", "Cophenetic Coefficient"]
```

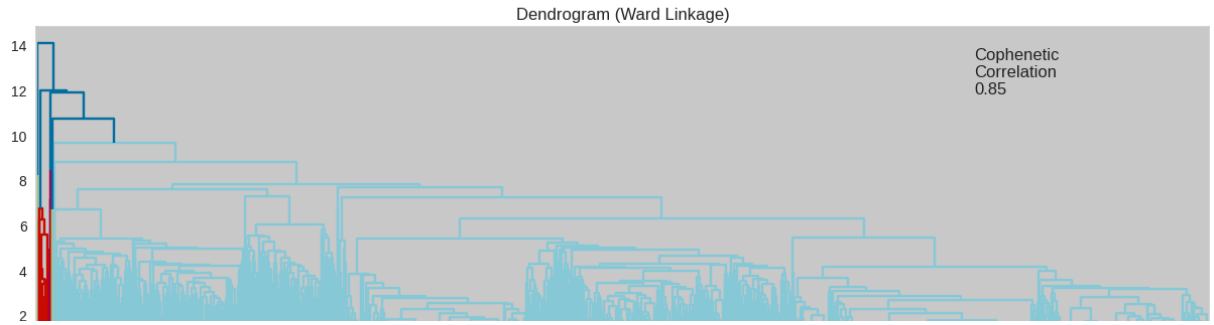
```
compare = []

# to create a subplot image
fig, axs = plt.subplots(len(linkage_methods), 1, figsize=(15, 30))

# We will enumerate through the list of linkage methods above
# For each linkage method, we will plot the dendrogram and calculate the cop
for i, method in enumerate(linkage_methods):


    Z = linkage(hc_data, metric="euclidean", method= 'average') # Calculatin

    dendrogram(Z, ax=axs[i])                            # Visualizing the De


    axs[i].set_title(f"Dendrogram ({method.capitalize()} Linkage)")

    coph_corr, coph_dist = cophenet(Z, pdist(hc_data))
    axs[i].annotate(
        f"Cophenetic\nCorrelation\n{coph_corr:0.2f}",
        (0.80, 0.80),
        xycoords="axes fraction",
    )
```

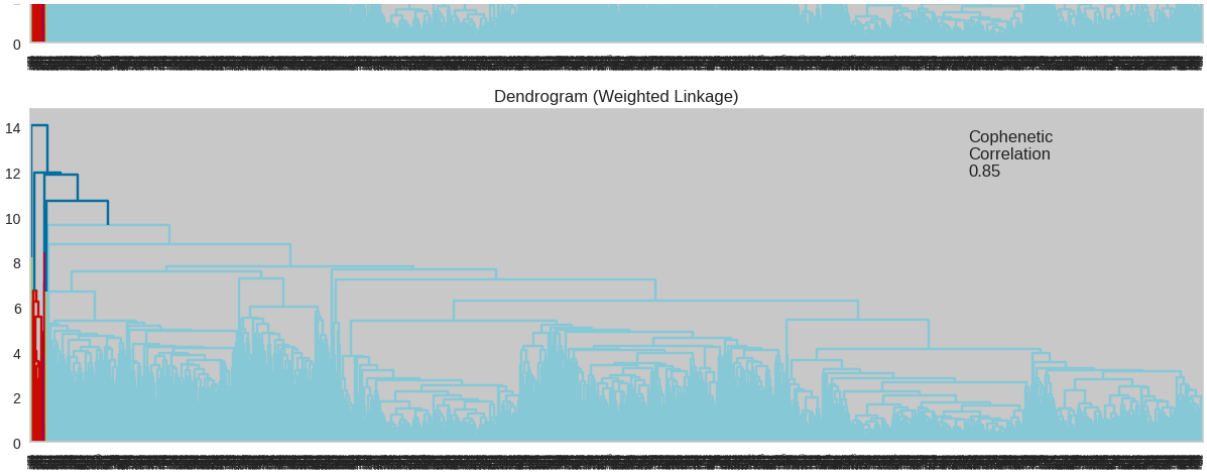

Dendrogram (Single Linkage)
Cophenetic
Correlation
0.85
14
12
10
8
6
4
2
0


Dendrogram (Complete Linkage)
Cophenetic
Correlation
0.85
14
12
10
8
6
4
2
0


Dendrogram (Average Linkage)
Cophenetic
Correlation
0.85
14
12
10
8
6
4
2
0


Dendrogram (Centroid Linkage)
Cophenetic
Correlation
0.85
14
12
10
8
6
4
2
0


Dendrogram (Ward Linkage)
Cophenetic
Correlation
0.85
14
12
10
8
6
4
2



Observations: Clusters into many clusters. Ward's method provides the best clustering separation.

Question 12: Check the silhoutte score for the hierchial clustering. What should be the appropriate number of clusters according to this plot?

In [29]:
```
# Write your code here
sil_score_hc = []
cluster_list = list(range(2, 10))
for n_clusters in cluster_list:
    clusterer = AgglomerativeClustering(n_clusters= n_clusters) # Initialize
    preds = clusterer.fit_predict((k_means_data))                # Fit the m
    score = silhouette_score(hc_data, preds)              # Calculate the s
    sil_score_hc.append(score)
    print("For n_clusters = {}, silhouette score is {}".format(n_clusters, s
```

```
For n_clusters = 2, silhouette score is 0.2752083036759842
For n_clusters = 3, silhouette score is 0.2821742003818839
For n_clusters = 4, silhouette score is 0.2913889880211373
For n_clusters = 5, silhouette score is 0.3051635245368833
For n_clusters = 6, silhouette score is 0.32003720102655475
For n_clusters = 7, silhouette score is 0.2826707444348891
For n_clusters = 8, silhouette score is 0.2830801755867989
For n_clusters = 9, silhouette score is 0.2764234706313624
```

Observations: Similar to K-Means, 3 clusters offer the best segmentation.

Question 13: Fit the Hierarchial clustering model with the appropriate parameters finalized above. How much time does it take to fit the model?

In [30]:
```
# Write your code here
%%time
HCmodel = AgglomerativeClustering(n_clusters=n_clusters, metric=dm, linkage=
HCmodel.fit(hc_data)
```

```
CPU times: user 353 ms, sys: 3.8 ms, total: 357 ms
Wall time: 382 ms
```

Out[30]:

```
AgglomerativeClustering(linkage='average', metric='cityblock', n_clusters=9)
```

Observations: Since Hierarchical clustering is useful for visualizing relationships, this is implemented by allowing us to see how long it takes to fit the model shown in CPU by user, system, and total time and Wall time presented in milliseconds.

Cluster Profiling and Comparison

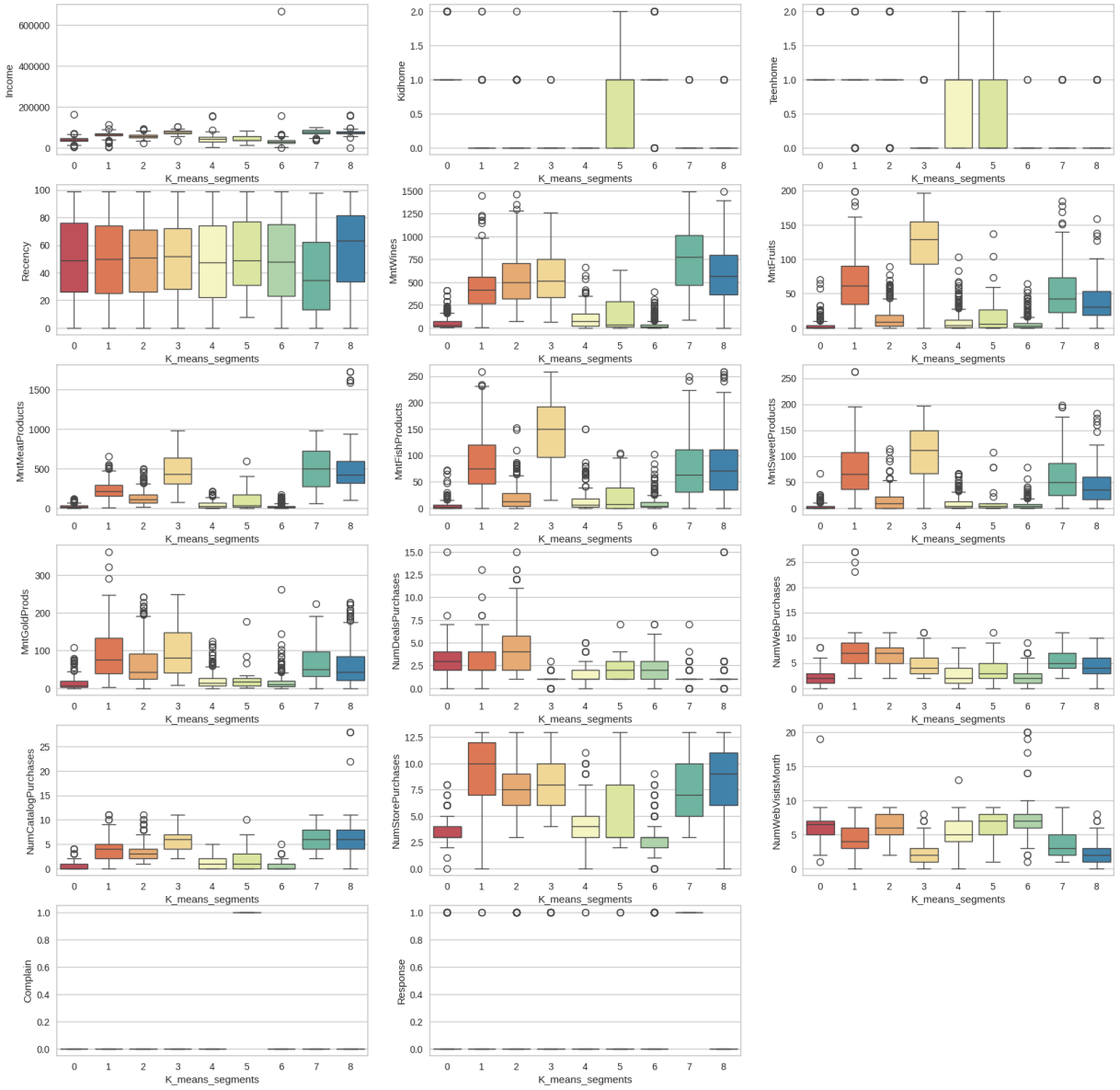
K-Means Clustering vs Hierarchical Clustering Comparison

Question 14: Perform and compare Cluster profiling on both algorithms using boxplots. Based on the all the observaions Which one of them provides better clustering?

In [31]:

```
# Write your code here
data2 = data.copy()

# adding hierarchical cluster labels to the original and scaled dataframes
hc_data["HC_segments"] = HCmodel.labels_
data2["HC_segments"] = HCmodel.labels_

plt.figure(figsize=(20, 20))  # Set the figure size for the plot
plt.suptitle("Boxplot of numerical variables for each cluster in Kmeans Clus

# Iterate over each numerical variable in the dataframe
for i, variable in enumerate(data1.columns.to_list()[:-1]):
    plt.subplot(6, 3, i + 1)  # Create subplots in a 3x4 grid, starting from
    sns.boxplot(data=data1, x = 'K_means_segments', y = variable, palette='S

subset_scaled_data["HC_Clusters"] = HCmodel.labels_
data["HC_Clusters"] = HCmodel.labels_
```

Boxplot of numerical variables for each cluster in Kmeans Clustering



Observations: K-Means is computationally efficient and provides clear clusters.

Hierarchical clustering is useful for visualizing relationships.

Question 15: Perform Cluster profiling on the data with the appropriate algorithm determined above using a barplot. What observations can be derived for each cluster from this plot?

In [32]:
```
# Write your code here
plt.figure(figsize=(20, 20))  # Set the figure size for the plot
plt.suptitle("Barplots of all variables for each cluster")  # Set the main t

for i, variable in enumerate(data.columns.to_list()[:-1]):
    plt.subplot(6, 3, i + 1)
    sns.barplot(data=data1, x = "K_means_segments", y = variable, palette='S
```

```
plt.tight_layout(pad=2.0)
```

Observations: Both methods reveal similar customer segments:

Low-income, low-spending High-income, high-spending Middle-income, balanced spending

Business Recommedations

- We have seen that 3 clusters are distinctly formed using both methodologies and the clusters are analogous to each other.
- Cluster 1 has premium customers with a high credit limit and more credit cards, indicating that they have more purchasing power. The customers in this group have a preference for online banking.

- Cluster 0 has customers who prefer to visit the bank for their banking needs than doing business online or over the phone. They have an average credit limit and a moderate number of credit cards.
- Cluster 2 has more overhead of customers calling in, and the bank may need to spend money on call centers.

Here are **5–7 actionable business recommendations** based on the cluster profiling:

---

1. Focus on Retaining High-Value Customers (Cluster 3)

- **Offer Exclusive Loyalty Programs**: Provide tailored loyalty benefits, early access to products, and exclusive discounts to maintain engagement and drive repeat purchases.
- **Upsell and Cross-Sell**: Introduce premium products or bundles targeting their high spending patterns across product categories like wines, gold products, and meats.
- **Personalized Campaigns**: Use their high response rate to create personalized campaigns highlighting products they prefer.

---

2. Activate Potential in Moderate-Spending Customers (Cluster 2)

- **Incentivize Higher Engagement**: Offer targeted discounts or special offers to encourage increased spending and purchases across channels.
- **Educate About Products**: Provide content (emails, guides, or social media) showcasing the value and uniqueness of products they don't purchase frequently.
- **Improve Campaign Effectiveness**: Refine campaign messaging based on their moderate response rate to increase acceptance.

---

3. Reengage Low-Value Customers (Cluster 1)

- **Win-Back Campaigns**: Implement campaigns specifically aimed at bringing back inactive customers, such as offering steep discounts or limited-time offers.
- **Understand Barriers to Engagement**: Conduct surveys or collect feedback to identify reasons for their low purchases and disengagement.
- **Promote Entry-Level Products**: Introduce affordable or trial-sized products to ease them into higher spending.

---

4. Convert Browsers into Buyers (Cluster 0)

- **Optimize Website Experience**: Since Cluster 0 has high website visits but low spending, improve website navigation, showcase popular products, and streamline the checkout process.

- **Targeted Digital Campaigns**: Retarget these users with ads or emails featuring products they browsed but didn't purchase.
- **Offer Online-Exclusive Discounts**: Provide web-only discounts or promotions to convert visits into purchases.

---

5. Strengthen Digital and Multi-Channel Strategies

- **Seamless Omni-Channel Experience**: Ensure a consistent shopping experience across all channels (web, catalog, and store) to encourage cross-channel engagement, especially for Clusters 2 and 3.
- **Digital Campaigns for All Clusters**: Focus on targeted digital campaigns, particularly for Clusters 0 and 2, as they have moderate to high online engagement.

---

6. Develop Campaigns to Boost Responses

- Use the insights from Clusters 2 and 3 (which show higher response rates) to refine campaign targeting and messaging. Emulate successful strategies used for Cluster 3 to increase responses across other segments.

---

7. Leverage Product-Specific Insights

- Promote popular categories (e.g., wines, gold products) to high-value clusters, while running introductory campaigns for less-engaged clusters to familiarize them with premium products.

By focusing on these strategies, the company can enhance engagement, increase revenue, and strengthen customer loyalty across all clusters.

Observations:

Insights on Each Cluster:

The high-spending cluster can be targeted for premium product promotions. The middle-income group responds well to discounts and loyalty programs. The low-income group benefits from budget-friendly options.

Business Recommendations:

Implement personalized marketing campaigns based on customer clusters. Consider loyalty programs for mid-to-high spenders. Collect additional behavioral data to refine segmentation.

In [66]:
```
import nbformat
import nbconvert
from google.colab import files

try:
    # Read the notebook content
    with open('drive/MyDrive/Learner_Fullcode.ipynb', 'r', encoding='utf-8')
        notebook_content = f.read()

    # Convert to HTML
    notebook = nbformat.reads(notebook_content, as_version=4)
    html_exporter = nbconvert.HTMLExporter()
    (body, resources) = html_exporter.from_notebook_node(notebook)

    # Create a temporary file
    with open('temp_Learner_Fullcode.html', 'w', encoding='utf-8') as temp_f
        temp_file.write(body)

    # Download the temporary file
    files.download('temp_Learner_Fullcode.html')

    # (Optional) Remove the temporary file
    import os
    os.remove('temp_Learner_Fullcode.html')

    print("Conversion and download successful.")

except Exception as e:
    print(f"An error occurred: {e}")
```

```
Conversion and download successful.
```

In [55]:
```
!pwd
```

```
/content
```

In [56]:
```
!ls
```

```
drive  sample_data
```

In [57]:
```
!ls drive
```

```
MyDrive
```

In [58]:
```
!ls drive/MyDrive
```

```
'AI Movie Recs Demo!: Aug 5, 2021 5:34 PM.webm'
 appsheet
'APUSH Chapter 41 (Period 9)- Flippity.net Flashcards Template.gsheet'
'APUSH Chapters 7-9 (Period 3) - Flippity.net Flashcards Template.gsheet'
'A Study in Charlotte_ Title _ Author Questions (1).gsheet'
'A Study in Charlotte_ Trivia Questions.gsheet'
'Automobile (1).gsheet'
 Automobile.csv
 Automobile.gsheet
'Brewster Public Library Creative Writing Workshop Classwork Notebook #1.gdo
c'
'Brewster Public Library Creative Writing Workshop Homework Notebook #1.gdoc'
'Business Presentation Template for Low Code Version - Customer Personality S
egmentation.gslides'
'Chrome OS Cloud backup'
'Chrome Syncable FileSystem'
'Claim 2 (Projectiles) - Physics H.gsheet'
 Classroom
'Colab Notebooks'
'College Admissions Workshop Sessions'
'Copy of Flippity.net Flashcards Template.gsheet'
'Course_list_export_Pace University_ Westchester.gsheet'
 Data_Visualization_with_Python.ipynb
'Descripciones Fisicas - Flippity.net Flashcards Template.gsheet'
'Eliza and Her Monsters Title _ Author Questions.gsheet'
'Eliza and her Monsters_Trivia Questions.gsheet'
 IMG_1295.MOV
 IMG_2711.HEIC
'incom sum New.xlsx'
'Inspirit AI Colab Notebooks'
 Larencule_Natasha_Resume.docx
 Learner_Fullcode.ipynb
 Learner_lowcode.ipynb
 marketing_campaign.gsheet
'MIT IDSS - Project Rubric: Making Sense Of Unstructured Data.gdoc'
'Movie Demo! Aug 5, 2021 3:51 PM.webm'
'My Confirmation Sponsor Essay.gdoc'
'Natasha L - AI Symposium.gslides'
'Natasha Larencule - Broken.gdoc'
'Natasha Larencule - Confirmation Saint Essay.gdoc'
'Natasha Larencule - Spring 2023 Ambassadors Outreach Tracker.gsheet'
'Natasha Larencule - VCB (WWI).gdoc'
'Natasha L - Fall 2022 Ambassadors Outreach Tracker.gsheet'
'Notebook - Introduction to Python.ipynb'
'Option Strategy Deck (1).gdoc'
 Pandas_for_Data_Science_Pandas.ipynb
 Practice+Exercise+-+Collection_of_variables.ipynb
 Practice+Exercise+-+Conditional_Statements.ipynb
 Practice+Exercise+-+Data-Types.ipynb
'Practice Exercise - Functions.ipynb'
 Practice+Exercise+-+Intro_to_variables.ipynb
 Practice+Exercise+-+Intro_to_variables_Solutions.ipynb
 Practice+Exercise+-+Looping_Statements.ipynb
'President Choice Board - Flippity.net Video Game Template.gsheet'
'Python_For_DataScience_intro 3.ipynb'
 Python_For_DataScience_intro.ipynb
```

```
 Python_for_Data_Science_NumPy.ipynb
 Resume.gdoc
'Robo En La Noche (Ch. 1-8) - Flippity.net Flashcards [English -> Spanish].gs
heet'
'ScratchBoard #1.heic'
'ScratchBoard #2.HEIC'
'ScratchBoard #3.heic'
 Screencastify
'Solution Chemistry - Flippity.net Flashcards Template.gsheet'
 StockData.csv
 StockData.gsheet
 StockData.xlsx
'Student Mental health.csv'
'The Serpent King Questions_Title Author Practice Questions.gsheet'
'The Serpent King Questions_Trivia Practice Questions.gsheet'
'They Both Die at the End_Title _ Author Questions.gsheet'
'They Both Die at the End_Trivia Questions.gsheet'
'Timekeeper _Title_Author  Questions.gsheet'
'Timekeeper_Trivia Questions Edited.gsheet'
'To Kill A Mockingbird - Full Text PDF.pdf'
'Untitled document.gdoc'
'Untitled presentation.gslides'
'Untitled spreadsheet.gsheet'
 VideoGameCreating.ipynb
'Voltmeters, Ammeters, Resistors, ... Ohm!!!!.gsheet'
'Welcome To Algebra(kurtz) 21-22 word doc.docx'
```

In [60]:
```
!ls drive/MyDrive/'Learner_Fullcode.ipynb'
```

```
drive/MyDrive/Learner_Fullcode.ipynb
```