MARKETING CAMPAIGN DATASET

DATA ANALYSIS BY - JITHIN JAYACHANDRAN

INTRODUCTION

OVERVIEW OF THE DATASET

THE DATASET IS A CUSTOMER ANALYTICS DATASET THAT PROVIDES INFORMATION ABOUT THE CUSTOMERS OF A RETAIL COMPANY.
THE DATASET INCLUDES VARIOUS ATTRIBUTES RELATED TO CUSTOMER DEMOGRAPHICS, PURCHASE BEHAVIOR, AND RESPONSE TO MARKETING CAMPAIGNS.
IT CAN BE USED FOR PERFORMING DETAILED DATA ANALYSIS AND GAINING INSIGHTS INTO CUSTOMER BEHAVIOR AND PREFERENCES.

TYPE OF DATA

THE DATASET CONTAINS BOTH NUMERICAL AND CATEGORICAL DATA, WITH EACH ROW REPRESENTING A UNIQUE CUSTOMER AND EACH COLUMN REPRESENTING A DIFFERENT ATTRIBUTE OF THE CUSTOMER.

NUMBER OF ROWS AND COLUMNS
ROWS: 2241 (EACH REPRESENTING A UNIQUE CUSTOMER)
COLUMNS: 28 (EACH REPRESENTING A DIFFERENT ATTRIBUTE OF THE CUSTOMER)

OBJECTIVE

THE PRIMARY GOAL OF THIS PROJECT IS TO ANALYZE CUSTOMER DATA FROM A MARKETING CAMPAIGN TO GAIN INSIGHTS THAT WILL HELP IN ENHANCING FUTURE MARKETING STRATEGIES AND IMPROVING CUSTOMER ENGAGEMENT.

PROJECT GOALS

UNDERSTAND CUSTOMER DEMOGRAPHICS: ANALYZE CUSTOMER AGE, INCOME, AND HOUSEHOLD COMPOSITION.
ASSESS MARKETING CAMPAIGN PERFORMANCE: EVALUATE THE EFFECTIVENESS OF DIFFERENT MARKETING CAMPAIGNS.
IDENTIFY KEY CUSTOMER SEGMENTS: SEGMENT CUSTOMERS BASED ON THEIR PURCHASING BEHAVIOR AND DEMOGRAPHICS.
PROVIDE ACTIONABLE INSIGHTS: OFFER RECOMMENDATIONS TO IMPROVE FUTURE MARKETING STRATEGIES BASED ON THE ANALYSIS.

Column Name	Description
ID	Unique identifier for each customer
Year_Birth	Year of birth of the customer
Education	Level of education attained by the customer
Marital_Status	Marital status of the customer
Income	Annual income of the customer
Kidhome	Number of children in the customer's household
Teenhome	Number of teenagers in the customer's household
Dt_Customer	Date when the customer was enrolled with the company
Recency	Number of days since the customer's last purchase
MntWines	Amount spent on wine products
MntFruits	Amount spent on fruit products
MntMeatProducts	Amount spent on meat products
MntFishProducts	Amount spent on fish products
MntSweetProducts	Amount spent on sweet products
MntGoldProds	Amount spent on gold products
NumDealsPurchases	Number of purchases made with a discount
NumWebPurchases	Number of purchases made through the company's website
NumCatalogPurchases	Number of purchases made using a catalog
NumStorePurchases	Number of purchases made directly in stores
NumWebVisitsMonth	Number of visits to the company's website
AcceptedCmp3	Whether the customer accepted the third campaign offer
AcceptedCmp4	Whether the customer accepted the fourth campaign offer
AcceptedCmp5	Whether the customer accepted the fifth campaign offer
AcceptedCmp1	Whether the customer accepted the first campaign offer
AcceptedCmp2	Whether the customer accepted the second campaign offer
Response	Whether the customer accepted any campaign offer
Complain	Whether the customer has complained
Country	Country of residence of the customer

DETAILED COLUMN DESCRIPTIONS

I HAVE CATEGORIZED THE COLUMNS AS FOLLOWS:

DEMOGRAPHICS

YEAR_BIRTH, EDUCATION, MARITAL_STATUS, INCOME, KIDHOME, TEENHOME

CUSTOMER RELATIONSHIP

DT_CUSTOMER, RECENCY

PURCHASING BEHAVIOR

MNTWINES, MNTFRUITS, MNTMEATPRODUCTS, MNTFISHPRODUCTS, MNTSWEETPRODUCTS, MNTGOLDPRODS

PURCHASE CHANNELS

NUMDEALSPURCHASES, NUMWEBPURCHASES, NUMCATALOGPURCHASES, NUMSTOREPURCHASES, NUMWEBVISITSMONTH

CAMPAIGN RESPONSES

ACCEPTEDCMP1, ACCEPTEDCMP2, ACCEPTEDCMP3, ACCEPTEDCMP4, ACCEPTEDCMP5, RESPONSE

CUSTOMER FEEDBACK

COMPLAIN

GEOGRAPHICAL INFORMATION

COUNTRY

LOADING THE DATA

```
IMPORT PANDAS AS PD

# LOAD THE DATASET INTO A DATAFRAME
DATA = PD.READ_CSV('/CONTENT/MARKETING_DATA.CSV')
```

DATA EXPLORATION

```
# 1. BASIC INFORMATION
PRINT("BASIC INFORMATION")
PRINT(DATA.INFO())
```

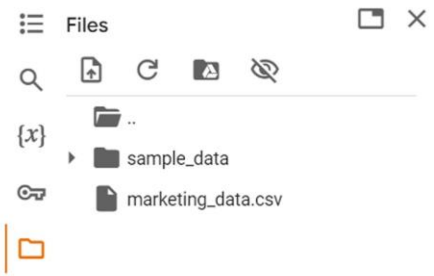
```
Basic Information
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2240 entries, 0 to 2239
Data columns (total 28 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   ID                   2240 non-null   int64
 1   Year_Birth           2240 non-null   int64
 2   Education            2240 non-null   object
 3   Marital_Status       2240 non-null   object
 4   Income               2216 non-null   object
 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  Response             2240 non-null   int64
 26  Complain             2240 non-null   int64
 27  Country              2240 non-null   object
dtypes: int64(23), object(5)
memory usage: 490.1+ KB
None
```

```
# 2. CHECK FOR MISSING VALUES
PRINT("\NMISSING VALUES")
PRINT(DATA.ISNULL().SUM())
```

```
Missing Values
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
Response                0
Complain                0
Country                 0
dtype: int64
```

```
# 3. CHECK FOR DUPLICATE ENTRIES
PRINT("\NDUPLICATE ENTRIES")
PRINT(DATA.DUPLICATED().SUM())
```

```
Duplicate Entries
0
```



DATA TYPE
CONVERSION

```
# 4. CHECK DATA TYPES OF COLUMNS
PRINT("\NDATA TYPES")
PRINT(DATA.DTYPES)
```

```
Data Types
ID                      int64
Year_Birth              int64
Education              object
Marital_Status         object
Income                 object
Kidhome                 int64
Teenhome                int64
Dt_Customer            object
Recency                 int64
MntWines                int64
MntFruits               int64
MntMeatProducts         int64
MntFishProducts         int64
MntSweetProducts        int64
MntGoldProds            int64
NumDealsPurchases       int64
NumWebPurchases         int64
NumCatalogPurchases     int64
NumStorePurchases       int64
NumWebVisitsMonth       int64
AcceptedCmp3            int64
AcceptedCmp4            int64
AcceptedCmp5            int64
AcceptedCmp1            int64
AcceptedCmp2            int64
Response                int64
Complain                int64
Country                object
dtype: object
```

```
# 5. CONVERTING THE DATATYPES

IMPORT PANDAS AS PD

# PREPROCESS THE 'INCOME' COLUMN TO REMOVE NON-NUMERIC CHARACTERS
DATA['INCOME'] = DATA['INCOME'].STR.REPLACE('$', '').STR.REPLACE(',', '')

# DEFINE THE MAPPINGS
DTYPE_MAPPING = {
  'ID': 'INT64',
  'YEAR_BIRTH': 'INT64',
  'EDUCATION': 'CATEGORY',
  'MARITAL_STATUS': 'CATEGORY',
  'INCOME': 'FLOAT64',
  'KIDHOME': 'INT64',
  'TEENHOME': 'INT64',
  'DT_CUSTOMER': 'DATETIME64[NS]',  # SPECIFY THE UNIT AS 'NS'
  'RECENCY': 'INT64',
  'MNTWINES': 'INT64',
  'MNTFRUITS': 'INT64',
  'MNTMEATPRODUCTS': 'INT64',
  'MNTFISHPRODUCTS': 'INT64',
  'MNTSWEETPRODUCTS': 'INT64',
  'MNTGOLDPRODS': 'INT64',
  'NUMDEALSPURCHASES': 'INT64',
  'NUMWEBPURCHASES': 'INT64',
  'NUMCATALOGPURCHASES': 'INT64',
  'NUMSTOREPURCHASES': 'INT64',
  'NUMWEBVISITSMONTH': 'INT64',
  'ACCEPTEDCMP3': 'BOOL',
  'ACCEPTEDCMP4': 'BOOL',
  'ACCEPTEDCMP5': 'BOOL',
  'ACCEPTEDCMP1': 'BOOL',
  'ACCEPTEDCMP2': 'BOOL',
  'RESPONSE': 'BOOL',
  'COMPLAIN': 'BOOL',
  'COUNTRY': 'CATEGORY'
}

# CONVERT DATA TYPES
DATA = DATA.ASTYPE(DTYPE_MAPPING)

# VERIFY THE DATA TYPES
PRINT(DATA.DTYPES)
```

```
Data Types
ID                             int64
Year_Birth                     int64
Education                   category
Marital_Status              category
Income                       float64
Kidhome                        int64
Teenhome                       int64
Dt_Customer           datetime64[ns]
Recency                        int64
MntWines                       int64
MntFruits                      int64
MntMeatProducts                int64
MntFishProducts                int64
MntSweetProducts               int64
MntGoldProds                   int64
NumDealsPurchases              int64
NumWebPurchases                int64
NumCatalogPurchases            int64
NumStorePurchases              int64
NumWebVisitsMonth              int64
AcceptedCmp3                    bool
AcceptedCmp4                    bool
AcceptedCmp5                    bool
AcceptedCmp1                    bool
AcceptedCmp2                    bool
Response                        bool
Complain                        bool
Country                     category
dtype: object
```

```
# 6. STATISTICAL SUMMARY OF NUMERICAL COLUMNS
PRINT("\NSTATISTICAL SUMMARY")
PRINT(DATA.DESCRIBE())
```

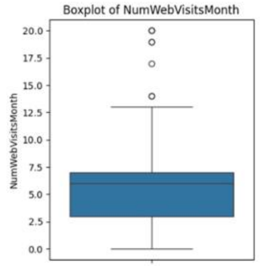
Statistic	Description
Count	The number of non-null (non-missing) entries in the column. It indicates how many values are present for each variable.
Mean	The average of all the values in the column. It is calculated as the sum of all values divided by the count of values.
Standard Deviation (std)	A measure of the amount of variation or dispersion in the values. A low standard deviation means the values tend to be close to the mean, while a high standard deviation means the values are spread out over a wider range.
Minimum (min)	The smallest value in the column. It represents the lowest observation.
25% (1st quartile)	The value below which 25% of the data fall. This is the first quartile, also known as Q1. It indicates the value at the 25th percentile.
50% (median)	The middle value when the data are ordered. This is the second quartile or median, representing the value at the 50th percentile. Half of the data points are below this value, and half are above.
75% (3rd quartile)	The value below which 75% of the data fall. This is the third quartile, also known as Q3. It indicates the value at the 75th percentile.
Maximum (max)	The largest value in the column. It represents the highest observation.

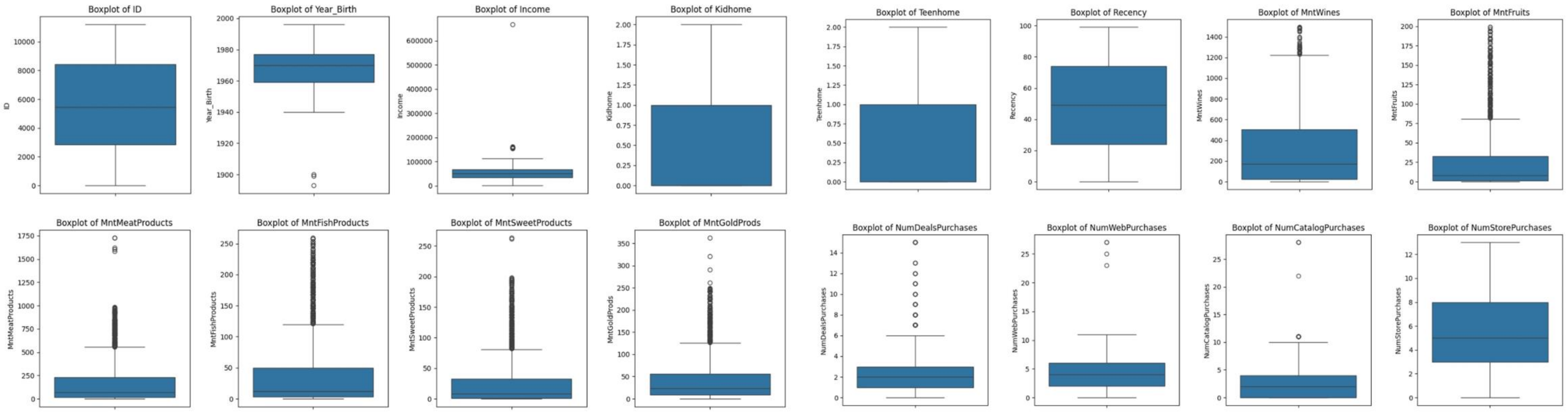
```
Statistical Summary
                 ID   Year_Birth         Income      Kidhome     Teenhome  \
count   2240.000000  2240.000000    2216.000000  2240.000000  2240.000000
mean    5592.159821  1968.805804   52247.251354     0.444196     0.506250
min        0.000000  1893.000000    1730.000000    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

                         Dt_Customer      Recency     MntWines    MntFruits  \
count                           2240  2240.000000  2240.000000  2240.000000
mean   2013-07-10 10:01:42.857142784    49.109375   303.935714    26.302232
min              2012-07-30 00:00:00     0.000000     0.000000     0.000000
25%              2013-01-16 00:00:00    24.000000    23.750000     1.000000
50%              2013-07-08 12:00:00    49.000000   173.500000     8.000000
75%              2013-12-30 06:00:00    74.000000   504.250000    33.000000
max              2014-06-29 00:00:00    99.000000  1493.000000   199.000000
std                              NaN    28.962453   336.597393    39.773434

       MntMeatProducts  MntFishProducts  MntSweetProducts  MntGoldProds  \
count      2240.000000      2240.000000       2240.000000   2240.000000
mean        166.950000        37.525446         27.062946     44.021875
min           0.000000         0.000000          0.000000      0.000000
25%          16.000000         3.000000          1.000000      9.000000
50%          67.000000        12.000000          8.000000     24.000000
75%         232.000000        50.000000         33.000000     56.000000
max        1725.000000       259.000000        263.000000    362.000000
std         225.715373        54.628979         41.280498     52.167439

       NumDealsPurchases  NumWebPurchases  NumCatalogPurchases  \
count        2240.000000      2240.000000          2240.000000
mean            2.325000         4.084821             2.662054
min             0.000000         0.000000             0.000000
25%             1.000000         2.000000             0.000000
50%             2.000000         4.000000             2.000000
75%             3.000000         6.000000             4.000000
max            15.000000        27.000000            28.000000
std             1.932238         2.778714             2.923101

       NumStorePurchases  NumWebVisitsMonth
count        2240.000000        2240.000000
mean            5.790179           5.316518
min             0.000000           0.000000
25%             3.000000           3.000000
50%             5.000000           6.000000
75%             8.000000           7.000000
max            13.000000          20.000000
std             3.250958           2.426645
```

```
# 6. IDENTIFY AND VISUALIZE POTENTIAL OUTLIERS
# PLOT BOXPLOTS FOR NUMERICAL COLUMNS TO IDENTIFY OUTLIERS

IMPORT PANDAS AS PD
IMPORT MATPLOTLIB.PYPLOT AS PLT
IMPORT SEABORN AS SNS

# IDENTIFY NUMERICAL COLUMNS IN THE DATASET
NUMERICAL_COLUMNS = DATA.SELECT_DTYPES(INCLUDE=['FLOAT64', 'INT64']).COLUMNS
NUM_COLS = LEN(NUMERICAL_COLUMNS) # COUNT THE NUMBER OF NUMERICAL COLUMNS
NUM_ROWS = (NUM_COLS // 4) + (1 IF NUM_COLS % 4 != 0 ELSE 0)

PLT.FIGURE(FIGSIZE=(16, 4 * NUM_ROWS))
FOR I, COLUMN IN ENUMERATE(NUMERICAL_COLUMNS, 1):
   PLT.SUBPLOT(NUM_ROWS, 4, I) # CREATE SUBPLOT IN THE FIGURE
   SNS.BOXPLOT(DATA[COLUMN]) # PLOT BOXPLOT FOR THE CURRENT NUMERICAL COLUMN
   PLT.TITLE(F'BOXPLOT OF {COLUMN}')
   PLT.TIGHT_LAYOUT()

PLT.SHOW()
```

OUTLIERS

AN OUTLIER IS A DATA POINT THAT IS NOTICEABLY DIFFERENT FROM THE OTHER VALUES IN A DATASET, OFTEN BEING MUCH HIGHER OR LOWER THAN THE MAJORITY.





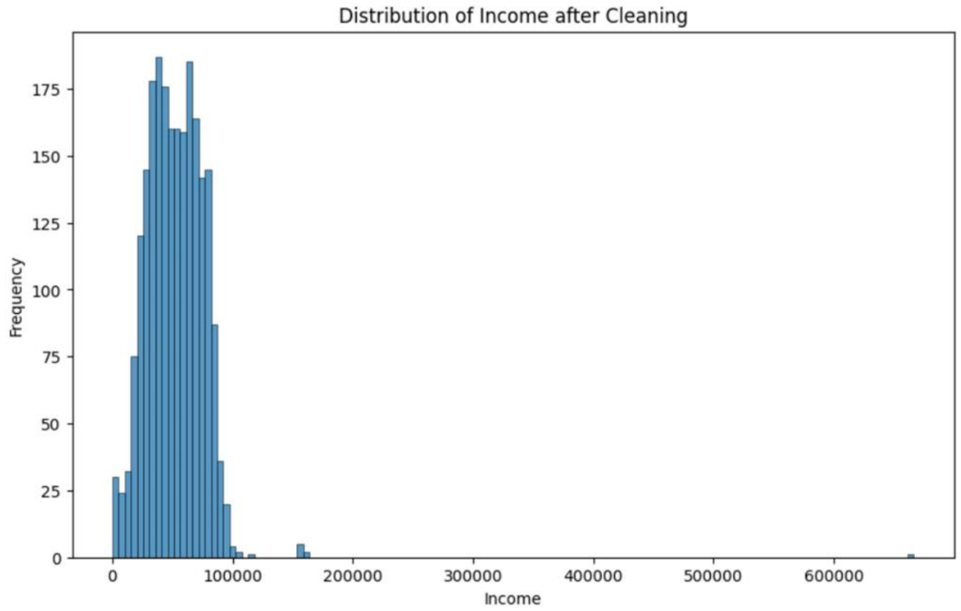
DATA CLEANING
HANDLING MISSING VALUES

```
# 1. IDENTIFY MISSING VALUES
MISSING_VALUES = DATA.ISNULL().SUM()
PRINT("MISSING VALUES:")
PRINT(MISSING_VALUES)
```

```
Missing Values:
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
Response               0
Complain               0
Country                0
dtype: int64
```

```
# 2. FILL MISSING VALUES WITH 0 FOR NUMERICAL COLUMNS
NUMERICAL_COLUMNS = DATA.SELECT_DTYPES(INCLUDE=['FLOAT64', 'INT64']).COLUMNS
DATA[NUMERICAL_COLUMNS] = DATA[NUMERICAL_COLUMNS].FILLNA(0)
```

```
# 3. VISUALIZE THE DISTRIBUTION OF CLEANED NUMERICAL COLUMNS [INCOME].
PLT.FIGURE(FIGSIZE=(10, 6))
SNS.HISTPLOT(DATA['INCOME'])
PLT.TITLE('DISTRIBUTION OF INCOME AFTER CLEANING')
PLT.XLABEL('INCOME')
PLT.YLABEL('FREQUENCY')
PLT.SHOW()
```



HANDLING DUPLICATES

```
# 1. CHECK FOR DUPLICATE ROWS
DUPLICATE_ROWS = DATA[DATA.DUPLICATED()]
PRINT("DUPLICATE ROWS:")
PRINT(DUPLICATE_ROWS)
```

```
Duplicate Rows:
Empty DataFrame
Columns: [ID, Year_Birth, Education, Marital_Status, Income, Kidhome, Teenhome, Dt_Customer, Recency, MntWines, MntFruits, MntMeatProducts,
          MntMeatProducts, MntFishProducts, MntSweetProducts, MntGoldProds, NumDealsPurchases, NumWebPurchases, NumCatalogPurchases, NumStorePurchases,
          NumStorePurchases, NumWebVisitsMonth, AcceptedCmp3, AcceptedCmp4, AcceptedCmp5, AcceptedCmp1, AcceptedCmp2, Response, Complain, Country]
Index: []

[0 rows x 28 columns]
```

```
# 2. PRINT THE ORIGINAL NUMBER OF ROWS
PRINT(F"ORIGINAL NUMBER OF ROWS: {LEN(DATA)}")
```

```
Original number of rows: 2240
```

```
# 3. REMOVE DUPLICATE ROWS (IF ANY)
DATA.DROP_DUPLICATES(INPLACE=TRUE)
```

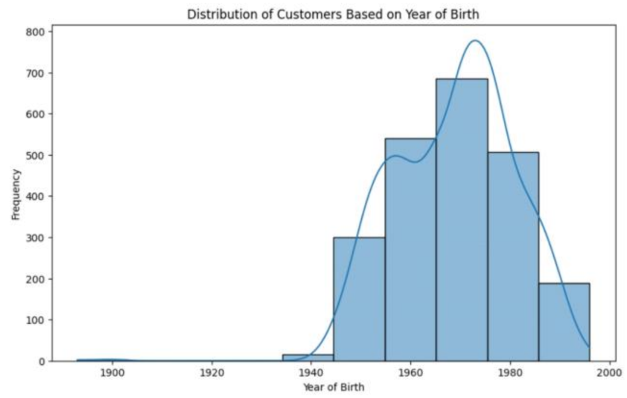
```
# 4. PRINT THE NEW NUMBER OF ROWS
PRINT(F"NEW NUMBER OF ROWS AFTER REMOVING DUPLICATES: {LEN(DATA)}")
```

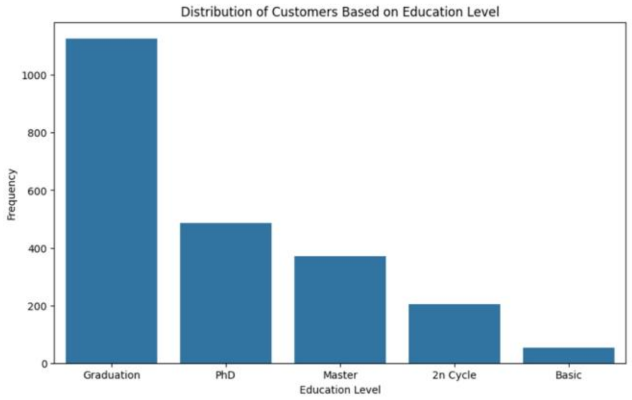
```
New number of rows after removing duplicates: 2240
```

DATA ANALYSIS

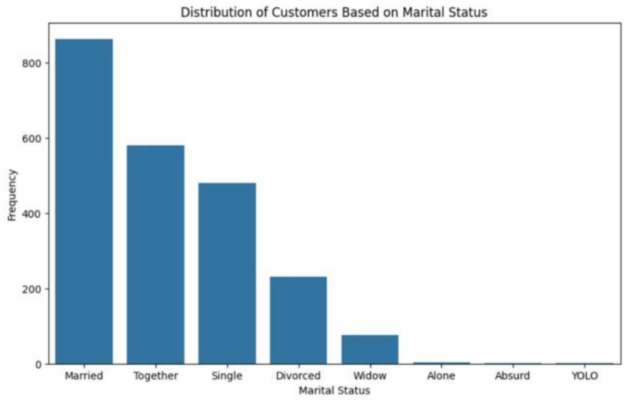
1. WHAT IS THE DISTRIBUTION OF CUSTOMERS BASED ON THEIR YEAR OF BIRTH, EDUCATION LEVEL, AND MARITAL STATUS?

```
# YEAR OF BIRTH DISTRIBUTION
PLT.FIGURE(FIGSIZE=(10, 6))
SNS.HISTPLOT(DATA['YEAR_BIRTH'], BINS=10, KDE=TRUE)
PLT.TITLE('DISTRIBUTION OF CUSTOMERS BASED ON YEAR OF BIRTH')
PLT.XLABEL('YEAR OF BIRTH')
PLT.YLABEL('FREQUENCY')
PLT.SHOW()
```






```
# EDUCATION LEVEL DISTRIBUTION
PLT.FIGURE(FIGSIZE=(10, 6))
SNS.COUNTPLOT(X='EDUCATION', DATA=DATA, ORDER=DATA['EDUCATION'].VALUE_COUNTS().INDEX)
PLT.TITLE('DISTRIBUTION OF CUSTOMERS BASED ON EDUCATION LEVEL')
PLT.XLABEL('EDUCATION LEVEL')
PLT.YLABEL('FREQUENCY')
PLT.SHOW()
```




```
# MARITAL STATUS DISTRIBUTION
PLT.FIGURE(FIGSIZE=(10, 6))
SNS.COUNTPLOT(X='MARITAL_STATUS', DATA=DATA, ORDER=DATA['MARITAL_STATUS'].VALUE_COUNTS().INDEX)
PLT.TITLE('DISTRIBUTION OF CUSTOMERS BASED ON MARITAL STATUS')
PLT.XLABEL('MARITAL STATUS')
PLT.YLABEL('FREQUENCY')
PLT.SHOW()
```

DEMOGRAPHIC ANALYSIS

```
# 2. WHAT IS THE PREDOMINANT MARITAL STATUS AMONG CUSTOMERS?

PREDOMINANT_MARITAL_STATUS = DATA['MARITAL_STATUS'].VALUE_COUNTS().IDXMAX()
PRINT(F'THE PREDOMINANT MARITAL STATUS AMONG CUSTOMERS IS {PREDOMINANT_MARITAL_STATUS}.')
```

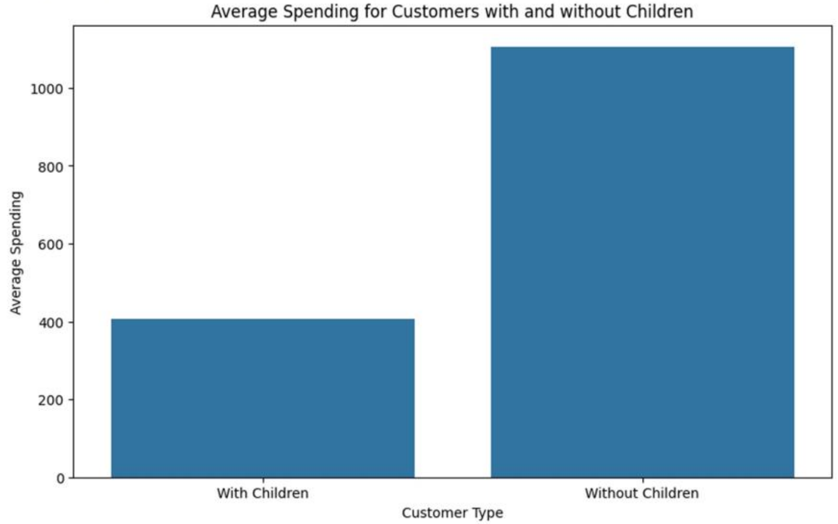
```
The predominant marital status among customers is Married.
```

```
# 3. IS THERE A CORRELATION BETWEEN INCOME LEVEL AND THE NUMBER OF CHILDREN IN THE HOUSEHOLD?

CORRELATION_INCOME_CHILDREN = DATA['INCOME'].CORR(DATA['KIDHOME'] + DATA['TEENHOME'])
PRINT(F'THE CORRELATION BETWEEN INCOME LEVEL AND THE NUMBER OF CHILDREN IS {CORRELATION_INCOME_CHILDREN:.2F}.')
```

```
The correlation between income level and the number of children is -0.29.
```

```
# 4. HOW DOES THE PURCHASING BEHAVIOR DIFFER BETWEEN CUSTOMERS WITH AND WITHOUT CHILDREN?

DATA['TOTAL_PURCHASES'] = DATA[['MNTWINES', 'MNTFRUITS', 'MNTMEATPRODUCTS', 'MNTFISHPRODUCTS',
                                'MNTSWEETPRODUCTS', 'MNTGOLDPRODS']].SUM(AXIS=1)

DATA['HAS_CHILDREN'] = DATA['KIDHOME'] + DATA['TEENHOME'] > 0

PURCHASES_WITH_CHILDREN = DATA[DATA['HAS_CHILDREN']]['TOTAL_PURCHASES'].MEAN()
PURCHASES_WITHOUT_CHILDREN = DATA[~DATA['HAS_CHILDREN']]['TOTAL_PURCHASES'].MEAN()

PRINT(F'AVERAGE SPENDING FOR CUSTOMERS WITH CHILDREN: {PURCHASES_WITH_CHILDREN:.2F}')
PRINT(F'AVERAGE SPENDING FOR CUSTOMERS WITHOUT CHILDREN: {PURCHASES_WITHOUT_CHILDREN:.2F}')

# PLOTTING
PLT.FIGURE(FIGSIZE=(10, 6))
SNS.BARPLOT(X=['WITH CHILDREN', 'WITHOUT CHILDREN'], Y=[PURCHASES_WITH_CHILDREN,
                                                        PURCHASES_WITHOUT_CHILDREN])
PLT.TITLE('AVERAGE SPENDING FOR CUSTOMERS WITH AND WITHOUT CHILDREN')
PLT.XLABEL('CUSTOMER TYPE')
PLT.YLABEL('AVERAGE SPENDING')
PLT.SHOW()
```

```
Average spending for customers with children: 406.58
Average spending for customers without children: 1106.03
```

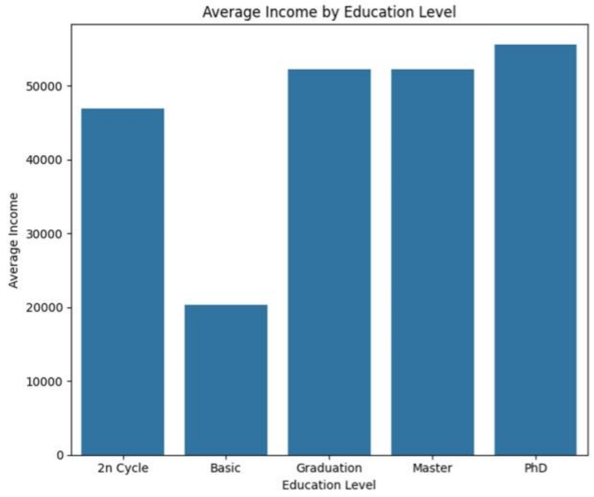


INCOME AND DEMOGRAPHIC SEGMENT

1: HOW DOES INCOME VARY ACROSS DIFFERENT DEMOGRAPHIC SEGMENTS?

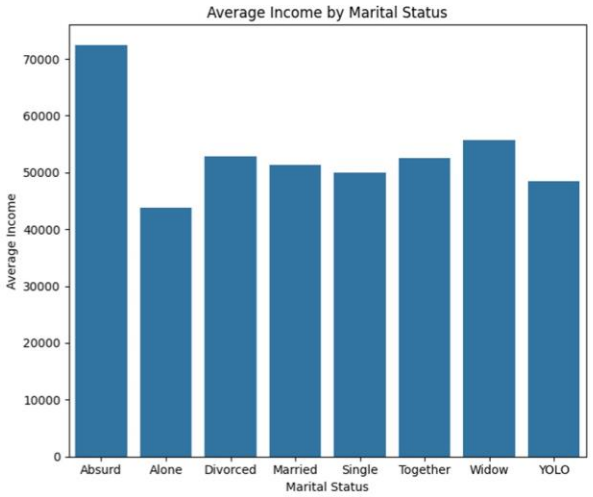
```
INCOME_EDUCATION = DATA.GROUPBY('EDUCATION')['INCOME'].MEAN()
INCOME_MARITAL_STATUS = DATA.GROUPBY('MARITAL_STATUS')['INCOME'].MEAN()

PRINT('AVERAGE INCOME BY EDUCATION LEVEL:')
PRINT(INCOME_EDUCATION)

PRINT('\NAVERAGE INCOME BY MARITAL STATUS:')
PRINT(INCOME_MARITAL_STATUS)
```

```
Average income by education level:
Education
2n Cycle      46929.251232
Basic         20306.259259
Graduation    52205.800355
Master        52202.432432
PhD           55567.687243
Name: Income, dtype: float64
```




```
# PLOTTING
PLT.FIGURE(FIGSIZE=(14, 6))

PLT.SUBPLOT(1, 2, 1)
SNS.BARPLOT(X=INCOME_EDUCATION.INDEX, Y=INCOME_EDUCATION.VALUES)
PLT.TITLE('AVERAGE INCOME BY EDUCATION LEVEL')
PLT.XLABEL('EDUCATION LEVEL')
PLT.YLABEL('AVERAGE INCOME')

PLT.SUBPLOT(1, 2, 2)
SNS.BARPLOT(X=INCOME_MARITAL_STATUS.INDEX, Y=INCOME_MARITAL_STATUS.VALUES)
PLT.TITLE('AVERAGE INCOME BY MARITAL STATUS')
PLT.XLABEL('MARITAL STATUS')
PLT.YLABEL('AVERAGE INCOME')

PLT.TIGHT_LAYOUT()
PLT.SHOW()
```

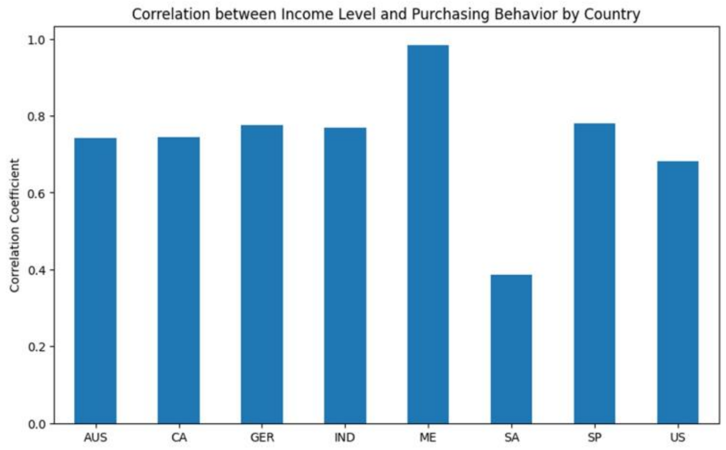
```
Average income by marital status:
Marital_Status
Absurd      72365.500000
Alone       43789.000000
Divorced    52834.228448
Married     51305.910880
Single      50039.187500
Together    52602.915517
Widow       55748.025974
YOLO        48432.000000
Name: Income, dtype: float64
```



2. IS THERE A CORRELATION BETWEEN INCOME LEVEL AND PURCHASING BEHAVIOR ACROSS DIFFERENT COUNTRIES?

```
CORRELATION_INCOME_PURCHASES = DATA.GROUPBY('COUNTRY')['INCOME'].CORR(DATA['TOTAL_PURCHASES'])
PRINT('CORRELATION BETWEEN INCOME LEVEL AND PURCHASING BEHAVIOR ACROSS DIFFERENT COUNTRIES:')
PRINT(CORRELATION_INCOME_PURCHASES)

# PLOTTING
PLT.FIGURE(FIGSIZE=(10, 6))
CORRELATION_INCOME_PURCHASES.PLOT(KIND='BAR')
PLT.TITLE('CORRELATION BETWEEN INCOME LEVEL AND PURCHASING BEHAVIOR BY COUNTRY')
PLT.XLABEL('COUNTRY')
PLT.XTICKS(ROTATION=0)
PLT.YLABEL('CORRELATION COEFFICIENT')
PLT.SHOW()
```

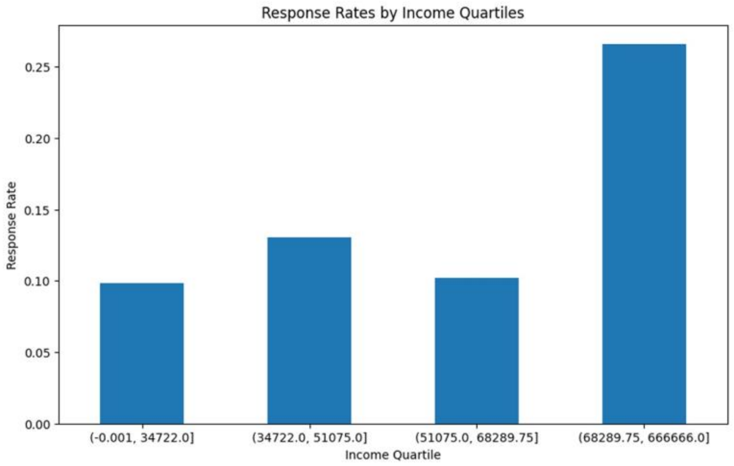
```
AUS    0.742545
CA     0.745360
GER    0.775466
IND    0.769846
ME     0.984205
SA     0.385333
SP     0.779565
US     0.682400
```



3. DO CUSTOMERS WITH HIGHER INCOME LEVELS TEND TO RESPOND MORE TO MARKETING CAMPAIGNS?

```
RESPONSE_RATE_BY_INCOME = DATA.GROUPBY(PD.QCUT(DATA['INCOME'], 4))['RESPONSE'].MEAN()
PRINT('RESPONSE RATES BY INCOME QUARTILES:')
PRINT(RESPONSE_RATE_BY_INCOME)

# PLOTTING
PLT.FIGURE(FIGSIZE=(10, 6))
RESPONSE_RATE_BY_INCOME.PLOT(KIND='BAR')
PLT.TITLE('RESPONSE RATES BY INCOME QUARTILES')
PLT.XLABEL('INCOME QUARTILE')
PLT.XTICKS(ROTATION=0)
PLT.YLABEL('RESPONSE RATE')
PLT.SHOW()
```

```
Response rates by income quartiles:
Income
(-0.001, 34722.0]       0.098214
(34722.0, 51075.0]      0.130357
(51075.0, 68289.75]     0.101786
(68289.75, 666666.0]    0.266071
Name: Response, dtype: float64
```

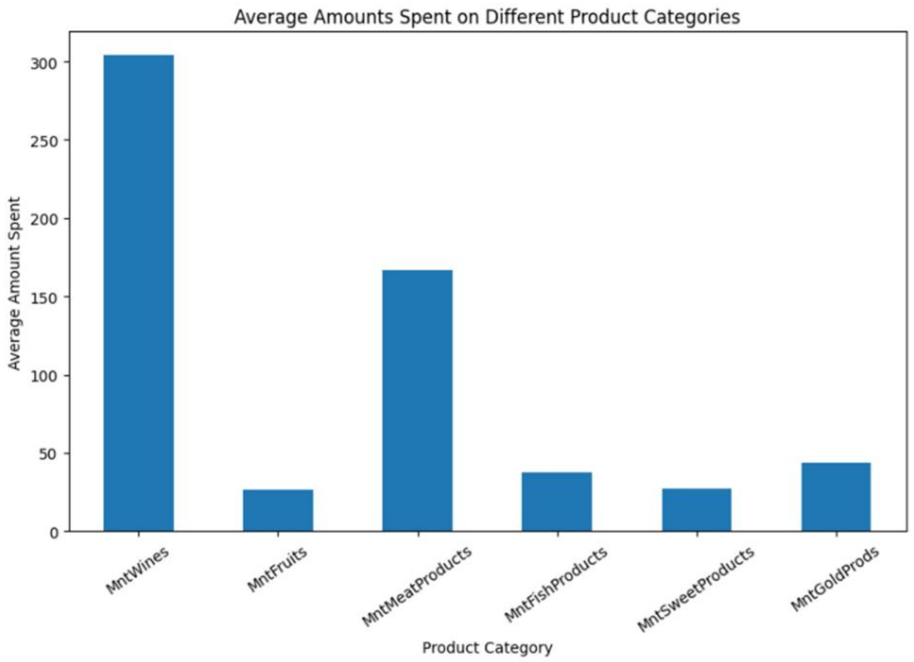


SPENDING & PRODUCT CATEGORIES

1. WHAT ARE THE AVERAGE AMOUNTS SPENT ON DIFFERENT PRODUCT CATEGORIES LIKE WINES, FRUITS, MEAT PRODUCTS, ETC.?

```
AVERAGE_SPENT_PER_CATEGORY = DATA[['MNTWINES', 'MNTFRUITS', 'MNTMEATPRODUCTS',
                                   'MNTFISHPRODUCTS', 'MNTSWEETPRODUCTS',
                                   'MNTGOLDPRODS']].MEAN()
PRINT('AVERAGE AMOUNTS SPENT ON DIFFERENT PRODUCT CATEGORIES:')
PRINT(AVERAGE_SPENT_PER_CATEGORY)

# PLOTTING
PLT.FIGURE(FIGSIZE=(10, 6))
AVERAGE_SPENT_PER_CATEGORY.PLOT(KIND='BAR')
PLT.TITLE('AVERAGE AMOUNTS SPENT ON DIFFERENT PRODUCT CATEGORIES')
PLT.XLABEL('PRODUCT CATEGORY')
PLT.XTICKS(ROTATION=35)
PLT.YLABEL('AVERAGE AMOUNT SPENT')
PLT.SHOW()
```

```
Average amounts spent on different product categories:
MntWines            303.935714
MntFruits            26.302232
MntMeatProducts     166.950000
MntFishProducts      37.525446
MntSweetProducts     27.062946
MntGoldProds         44.021875
dtype: float64
```



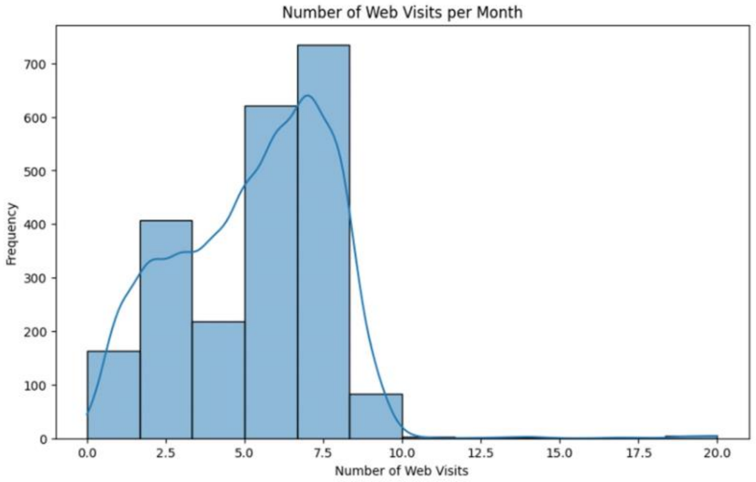
2. WHICH PRODUCT CATEGORY HAS THE HIGHEST AVERAGE SPENDING AMONG CUSTOMERS?

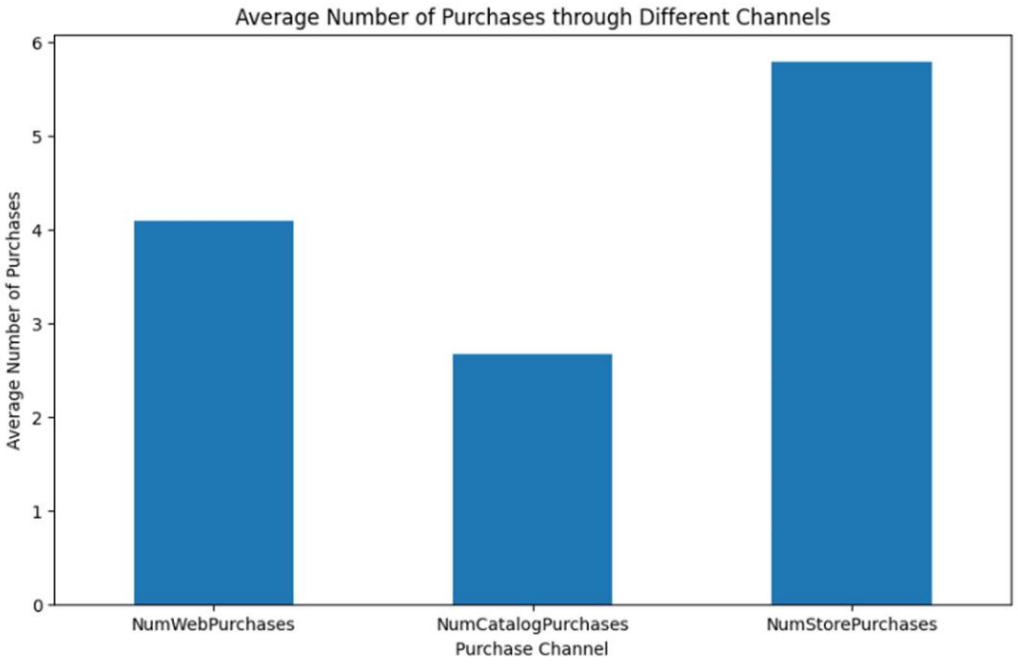
```
HIGHEST_SPENDING_CATEGORY = AVERAGE_SPENT_PER_CATEGORY.IDXMAX()
PRINT(F'THE PRODUCT CATEGORY WITH THE HIGHEST AVERAGE SPENDING IS {HIGHEST_SPENDING_CATEGORY}.')
```

```
The product category with the highest average spending is MntWines.
```

PURCHASING CHANNEL & BEHAVIOR

```
#1. HOW DO THE NUMBER OF PURCHASES THROUGH DIFFERENT CHANNELS (WEB, CATALOG, STORE) COMPARE?

PLT.FIGURE(FIGSIZE=(10, 6))
DATA[['NUMWEBPURCHASES', 'NUMCATALOGPURCHASES', 'NUMSTOREPURCHASES']].MEAN().PLOT(KIND='BAR')
PLT.TITLE('AVERAGE NUMBER OF PURCHASES THROUGH DIFFERENT CHANNELS')
PLT.XLABEL('PURCHASE CHANNEL')
PLT.XTICKS(ROTATION=0)
PLT.YLABEL('AVERAGE NUMBER OF PURCHASES')
PLT.SHOW()

# 2. ARE THERE ANY NOTICEABLE TRENDS IN THE NUMBER OF WEB VISITS PER MONTH OVER TIME?

PLT.FIGURE(FIGSIZE=(10, 6))
SNS.HISTPLOT(DATA['NUMWEBVISITSMONTH'], BINS=12, KDE=TRUE)
PLT.TITLE('NUMBER OF WEB VISITS PER MONTH')
PLT.XLABEL('NUMBER OF WEB VISITS')
PLT.YLABEL('FREQUENCY')
PLT.SHOW()
```

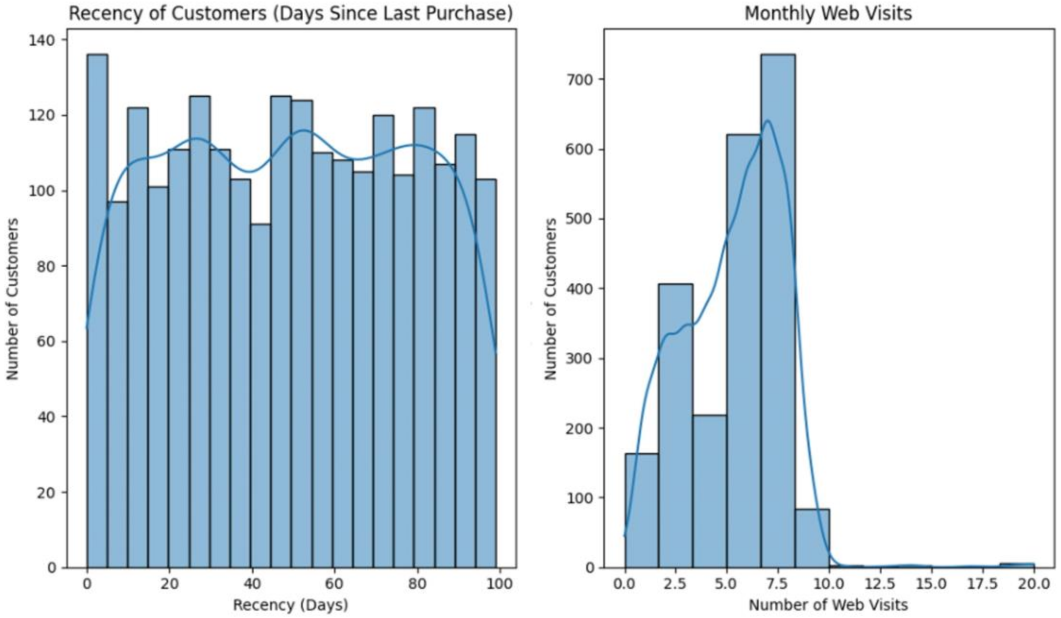




3. HOW RECENT ARE THE CUSTOMERS IN TERMS OF THEIR LAST PURCHASE (RECENCY) AND MONTHLY WEB VISITS?

```
RECENCY = DATA['RECENCY'].DESCRIBE()
MONTHLY_WEB_VISITS = DATA['NUMWEBVISITSMONTH'].DESCRIBE()

PRINT('RECENCY OF CUSTOMERS (DAYS SINCE LAST PURCHASE):')
PRINT(RECENCY)

PRINT('\NMONTHLY WEB VISITS:')
PRINT(MONTHLY_WEB_VISITS)

# PLOTTING
PLT.FIGURE(FIGSIZE=(10, 6))

PLT.SUBPLOT(1, 2, 1)
SNS.HISTPLOT(DATA['RECENCY'], BINS=20, KDE=TRUE)
PLT.TITLE('RECENCY OF CUSTOMERS (DAYS SINCE LAST PURCHASE)')
PLT.XLABEL('RECENCY (DAYS)')
PLT.YLABEL('NUMBER OF CUSTOMERS')

PLT.SUBPLOT(1, 2, 2)
SNS.HISTPLOT(DATA['NUMWEBVISITSMONTH'], BINS=12, KDE=TRUE)
PLT.TITLE('MONTHLY WEB VISITS')
PLT.XLABEL('NUMBER OF WEB VISITS')
PLT.YLABEL('NUMBER OF CUSTOMERS')

PLT.TIGHT_LAYOUT()
PLT.SHOW()
```

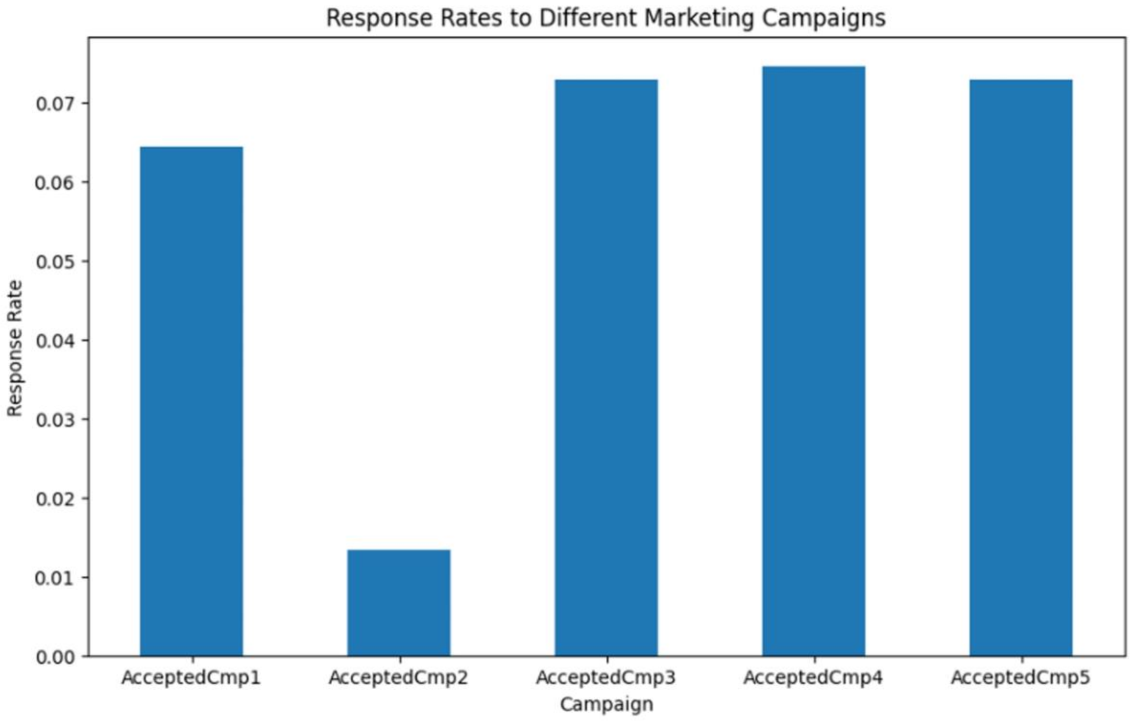
```
Recency of customers (days since last purchase):
count    2240.000000
mean       49.109375
std        28.962453
min         0.000000
25%        24.000000
50%        49.000000
75%        74.000000
max        99.000000
Name: Recency, dtype: float64

Monthly web visits:
count    2240.000000
mean        5.316518
std         2.426645
min         0.000000
25%         3.000000
50%         6.000000
75%         7.000000
max        20.000000
Name: NumWebVisitsMonth, dtype: float64
```



MARKETING CAMPAIGNS

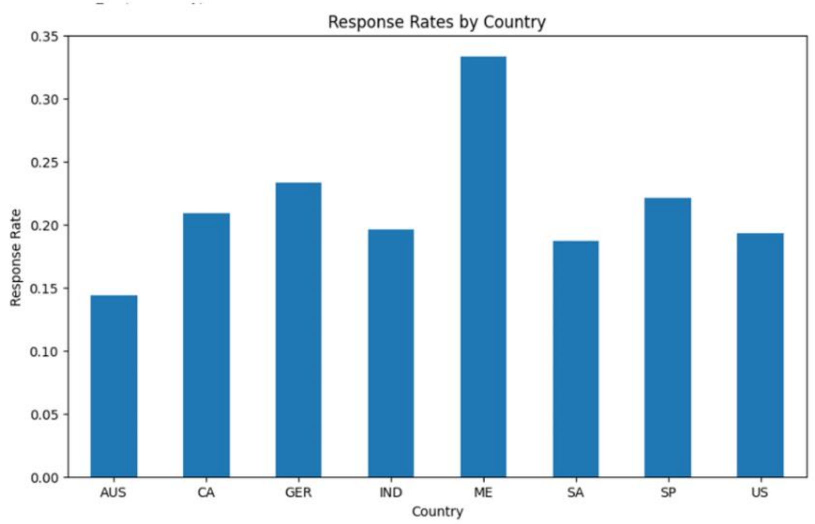
```
# WHAT IS THE RESPONSE RATE TO DIFFERENT MARKETING CAMPAIGNS (CMP1-CMP5) AND THE OVERALL RESPONSE RATE?

IMPORT PANDAS AS PD
IMPORT MATPLOTLIB.PYPLOT AS PLT

CAMPAIGN_COLUMNS = ['ACCEPTEDCMP1', 'ACCEPTEDCMP2', 'ACCEPTEDCMP3', 'ACCEPTEDCMP4', 'ACCEPTEDCMP5']
DATA.COLUMNS = DATA.COLUMNS.STR.STRIP()
DATA['OVERALL_RESPONSE'] = DATA[CAMPAIGN_COLUMNS].SUM(AXIS=1) > 0

RESPONSE_RATES = DATA[CAMPAIGN_COLUMNS].MEAN()
OVERALL_RESPONSE_RATE = DATA['OVERALL_RESPONSE'].MEAN()

PRINT('RESPONSE RATES TO DIFFERENT MARKETING CAMPAIGNS:')
PRINT(RESPONSE_RATES)

PRINT(F'\NOVERALL RESPONSE RATE: {OVERALL_RESPONSE_RATE:.2%}')

# PLOTTING THE RESPONSE RATES
PLT.FIGURE(FIGSIZE=(10, 6))
RESPONSE_RATES.PLOT(KIND='BAR')
PLT.TITLE('RESPONSE RATES TO DIFFERENT MARKETING CAMPAIGNS')
PLT.XLABEL('CAMPAIGN')
PLT.XTICKS(ROTATION=0)
PLT.YLABEL('RESPONSE RATE')
PLT.SHOW()
```

```
Response rates to different marketing campaigns:
AcceptedCmp1    0.064286
AcceptedCmp2    0.013393
AcceptedCmp3    0.072768
AcceptedCmp4    0.074554
AcceptedCmp5    0.072768
dtype: float64

Overall response rate: 20.67%
```



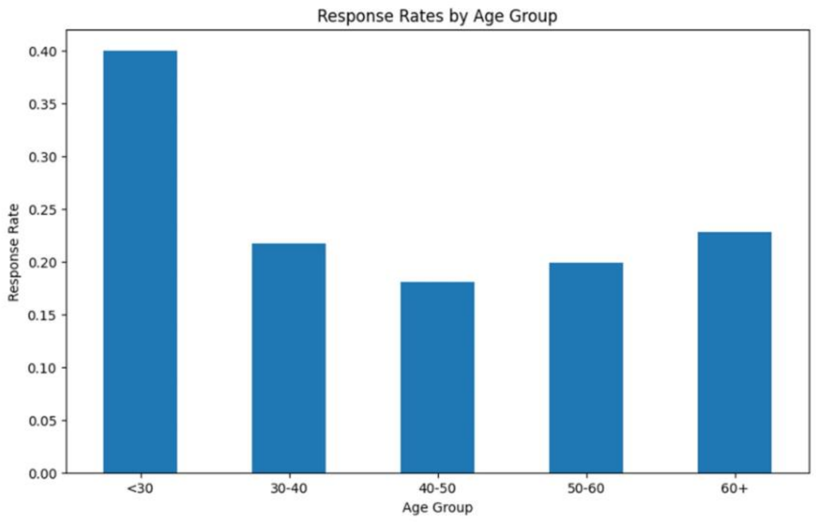
```
# 2. ARE THERE ANY NOTABLE DIFFERENCES IN RESPONSE RATES ACROSS DIFFERENT COUNTRIES?

RESPONSE_RATE_BY_COUNTRY = DATA.GROUPBY('COUNTRY')['OVERALL_RESPONSE'].MEAN()
PRINT('RESPONSE RATES BY COUNTRY:')
PRINT(RESPONSE_RATE_BY_COUNTRY)

# PLOTTING
PLT.FIGURE(FIGSIZE=(10, 6))
RESPONSE_RATE_BY_COUNTRY.PLOT(KIND='BAR')
PLT.TITLE('RESPONSE RATES BY COUNTRY')
PLT.XLABEL('COUNTRY')
PLT.XTICKS(ROTATION=0)
PLT.YLABEL('RESPONSE RATE')
PLT.SHOW()
```

```
Response rates by country:
Country
AUS    0.143750
CA     0.208955
GER    0.233333
IND    0.195946
ME     0.333333
SA     0.186944
SP     0.221005
US     0.192661
Name: Overall_Response, dtype: float64
```




```
# 3. HOW DOES THE ACCEPTANCE RATE OF MARKETING CAMPAIGNS VARY ACROSS DIFFERENT AGE GROUPS?

DATA['AGE'] = 2024 - DATA['YEAR_BIRTH']
DATA['AGE_GROUP'] = PD.CUT(DATA['AGE'],
BINS=[0, 30, 40, 50, 60, 100], LABELS=['<30', '30-40', '40-50', '50-60', '60+'])
RESPONSE_RATE_BY_AGE_GROUP = DATA.GROUPBY('AGE_GROUP')['OVERALL_RESPONSE'].MEAN()
PRINT('RESPONSE RATES BY AGE GROUP:')
PRINT(RESPONSE_RATE_BY_AGE_GROUP)

# PLOTTING
PLT.FIGURE(FIGSIZE=(10, 6))
RESPONSE_RATE_BY_AGE_GROUP.PLOT(KIND='BAR')
PLT.TITLE('RESPONSE RATES BY AGE GROUP')
PLT.XLABEL('AGE GROUP')
PLT.XTICKS(ROTATION=0)
PLT.YLABEL('RESPONSE RATE')
PLT.SHOW()
```

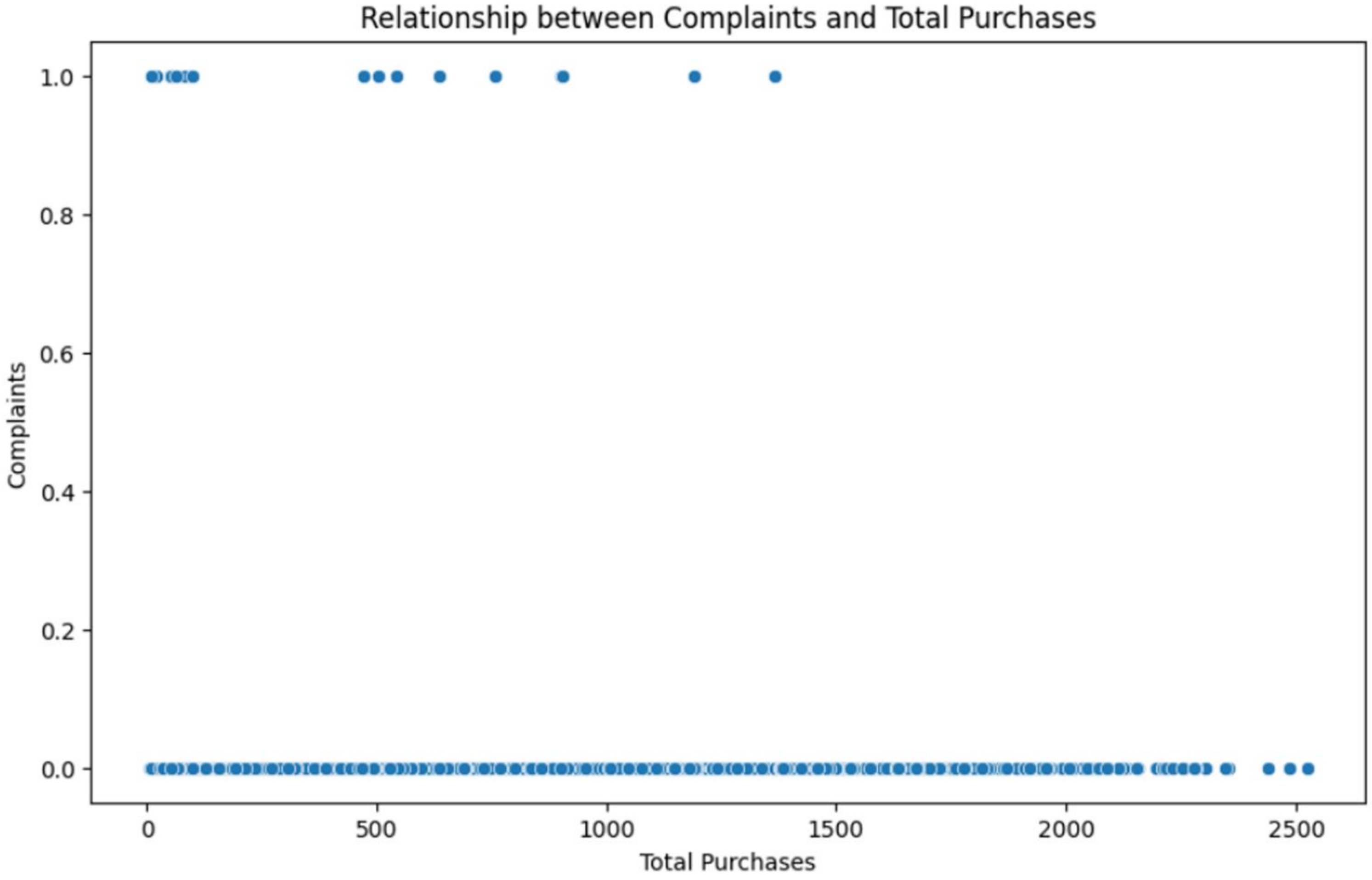
```
Response rates by age group:
Age_Group
<30      0.400000
30-40    0.216867
40-50    0.180272
50-60    0.198767
60+      0.228070
Name: Overall_Response, dtype: float64
```



CUSTOMERS COMPLAINTS & SATISFACTION

1. ARE THERE ANY PATTERNS IN CUSTOMER COMPLAINTS AND THEIR RELATIONSHIP WITH PURCHASES OR DEMOGRAPHICS?

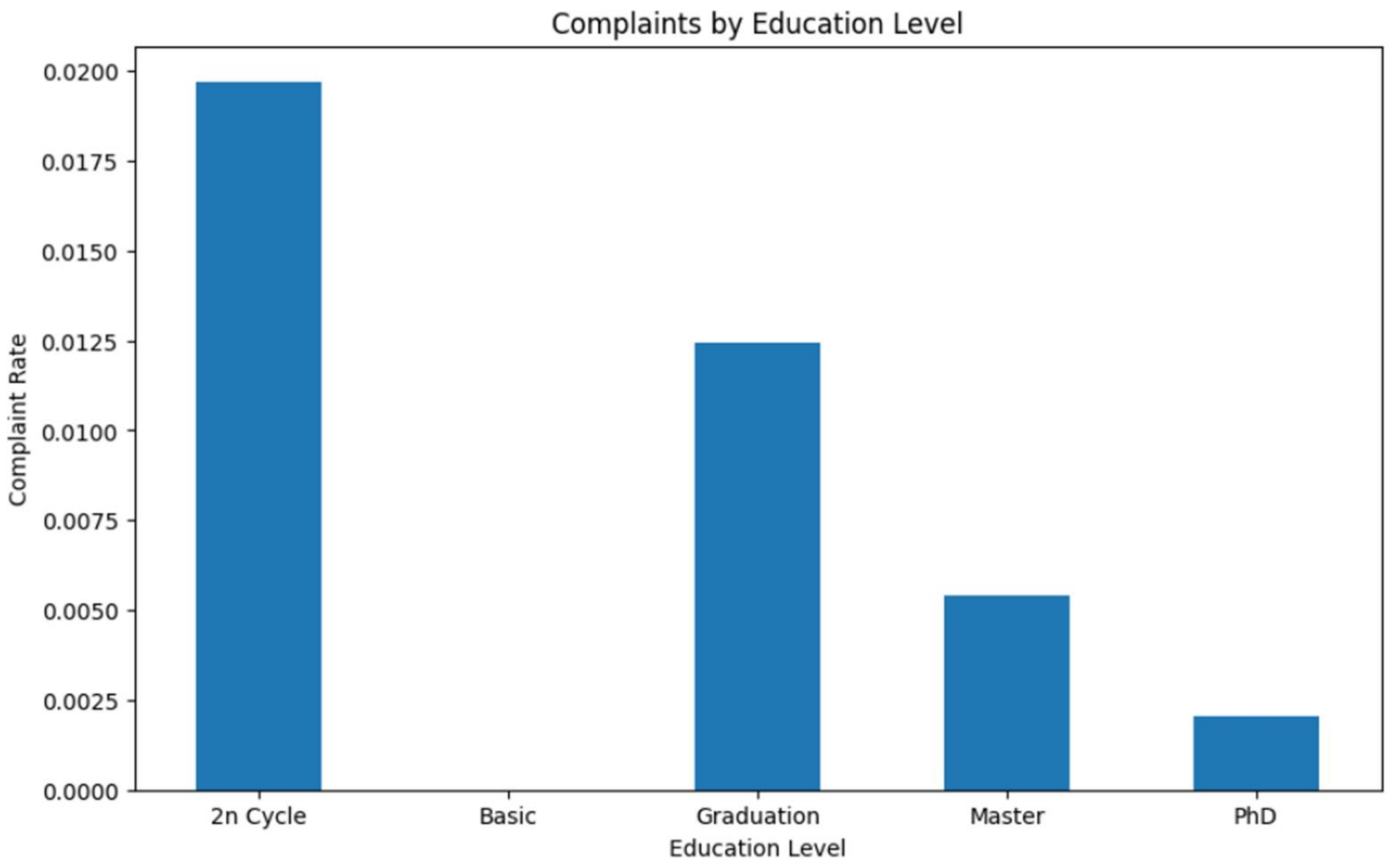
```
Correlation between complaints and total purchases: -0.04
```



2. WHAT PROPORTION OF CUSTOMERS HAVE PREVIOUSLY MADE COMPLAINTS?

```
Proportion of customers who have made complaints: 0.94%
```

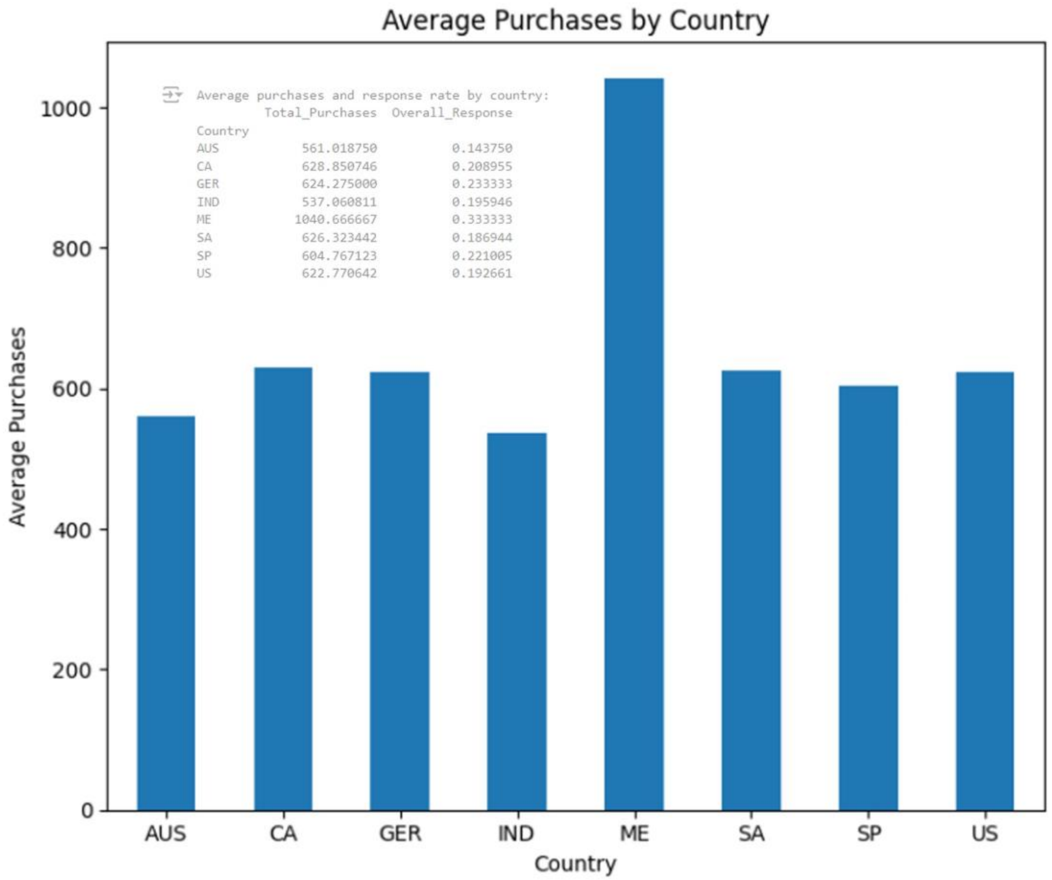
3. IS THERE A RELATIONSHIP BETWEEN CUSTOMER SATISFACTION (MEASURED THROUGH COMPLAINTS) AND EDUCATION LEVEL?

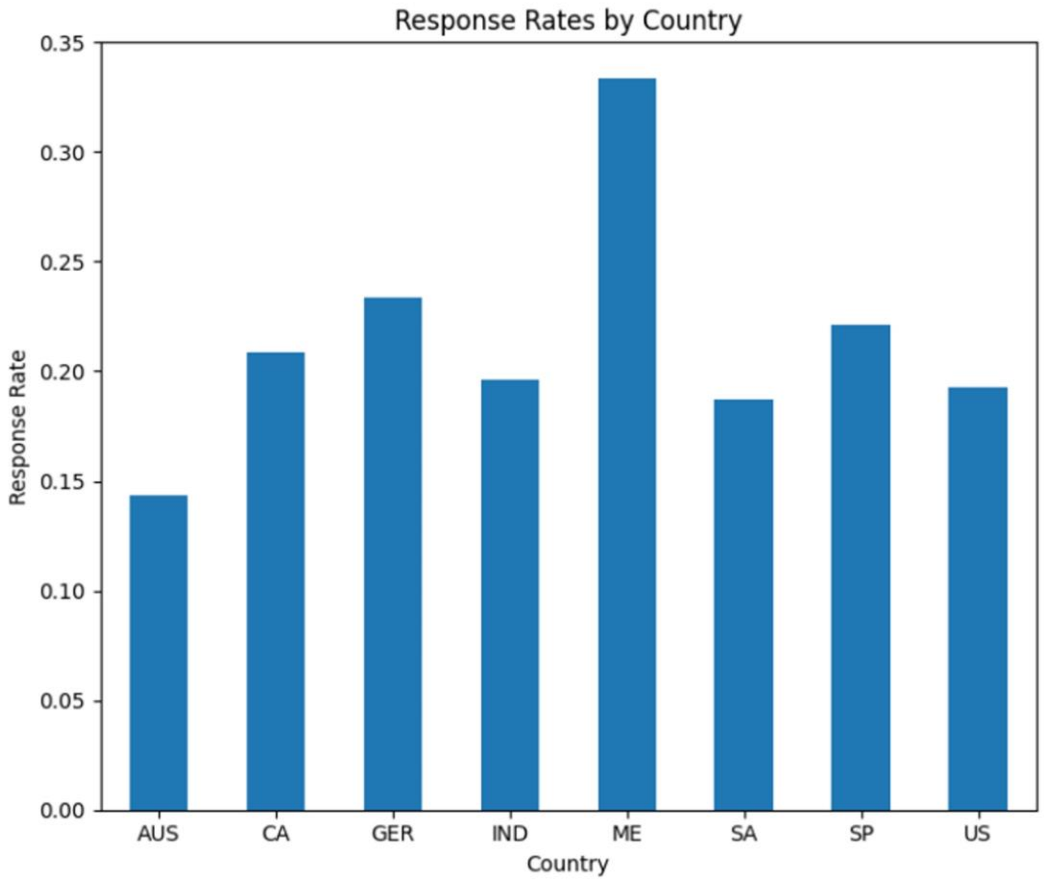



```
Complaints by education level:
Education
2n Cycle      0.019704
Basic         0.000000
Graduation    0.012422
Master        0.005405
PhD           0.002058
Name: Complain, dtype: float64
```

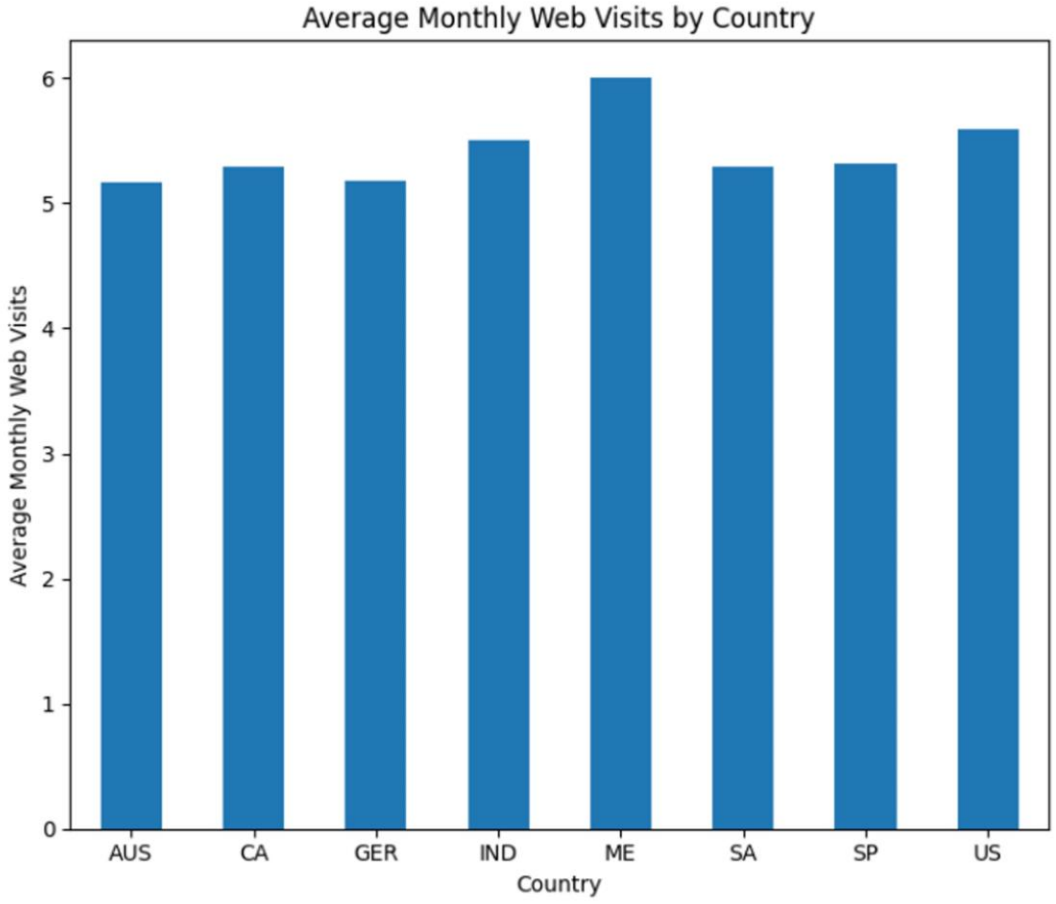
COUNTRY-SPECIFIC ANALYSIS

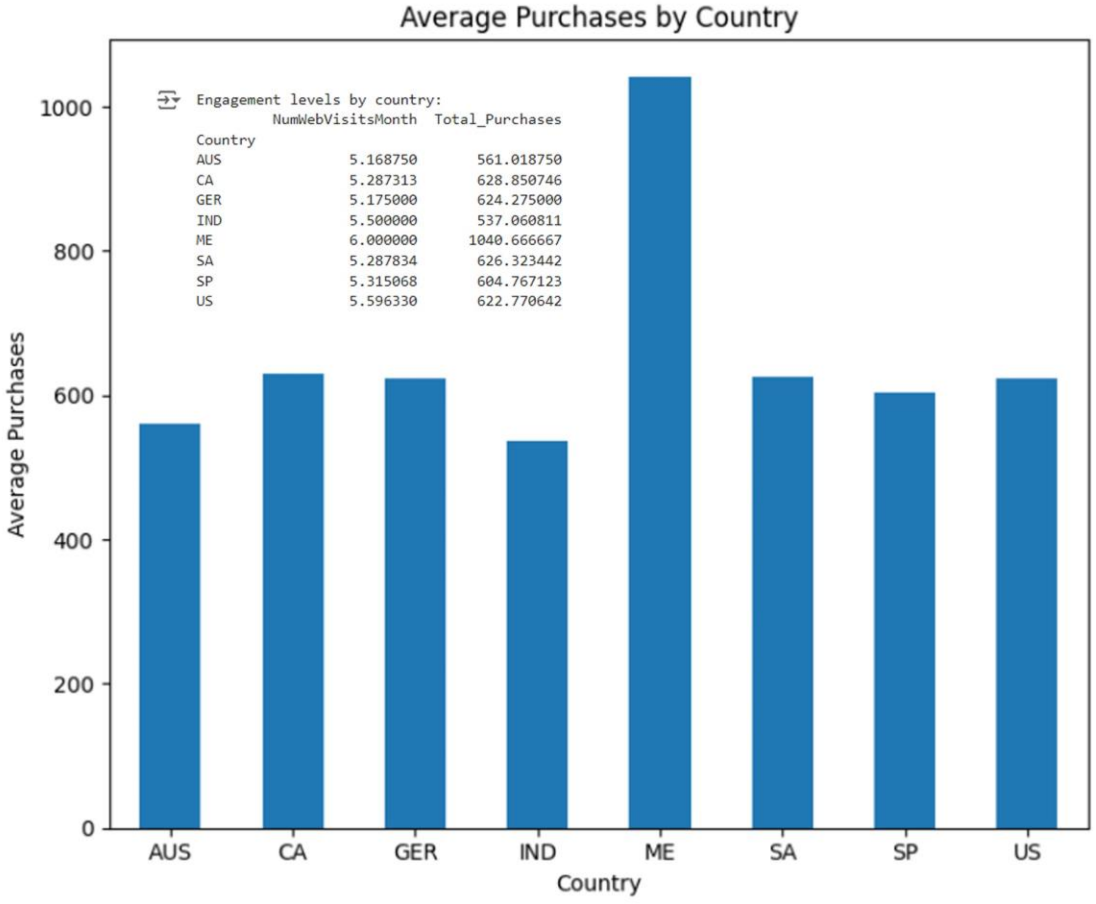
1. HOW DO THE CUSTOMERS VARY ACROSS DIFFERENT COUNTRIES IN TERMS OF THEIR BEHAVIOR AND RESPONSE TO MARKETING CAMPAIGNS?





2. ARE THERE ANY NOTABLE DIFFERENCES IN PURCHASING PATTERNS OR ENGAGEMENT LEVELS BASED ON THE COUNTRY OF RESIDENCE?





CORRELATION ANALYSIS

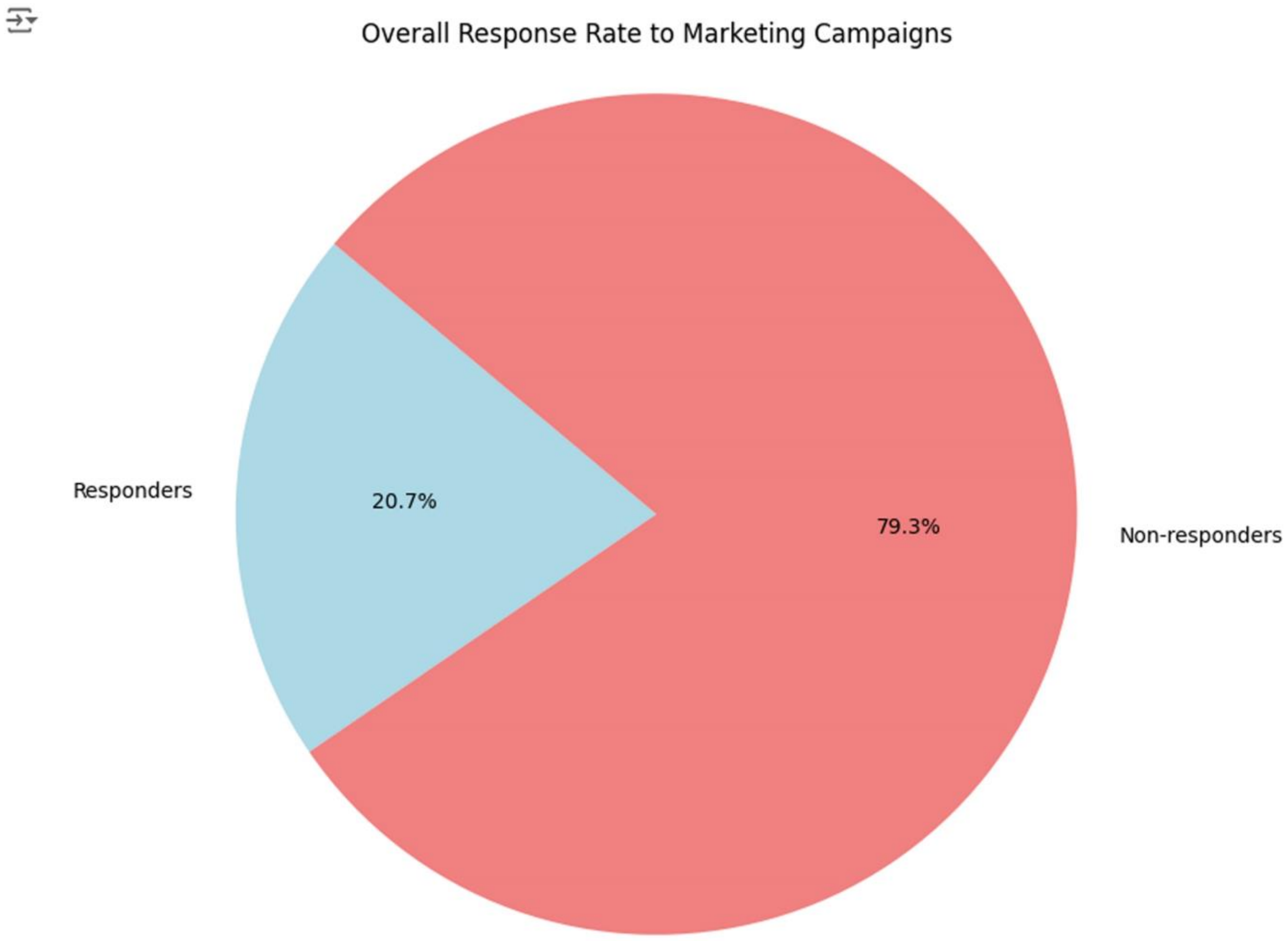
IS THERE A CORRELATION BETWEEN RECENCY, WEB VISITS, AND THE NUMBER OF DEALS OR PURCHASES MADE?


```
CORRELATION_MATRIX = DATA[['RECENCY', 'NUMWEBVISITSMONTH', 'TOTAL_PURCHASES']].CORR()
PRINT('CORRELATION MATRIX BETWEEN RECENCY, WEB VISITS, AND NUMBER OF PURCHASES:')
PRINT(CORRELATION_MATRIX)

# PLOTTING
PLT.FIGURE(FIGSIZE=(10, 6))
SNS.HEATMAP(CORRELATION_MATRIX, ANNOT=TRUE, CMAP='COOLWARM')
PLT.TITLE('CORRELATION MATRIX BETWEEN RECENCY, WEB VISITS, AND PURCHASES')
PLT.SHOW()
```

```
Correlation matrix between recency, web visits, and number of purchases:
                    Recency  NumWebVisitsMonth  Total_Purchases
Recency            1.000000          -0.021445         0.020433
NumWebVisitsMonth -0.021445           1.000000        -0.500218
Total_Purchases    0.020433          -0.500218         1.000000
```

CONCLUSION

1. WHAT IS THE OVERALL RESPONSE RATE TO MARKETING CAMPAIGNS?



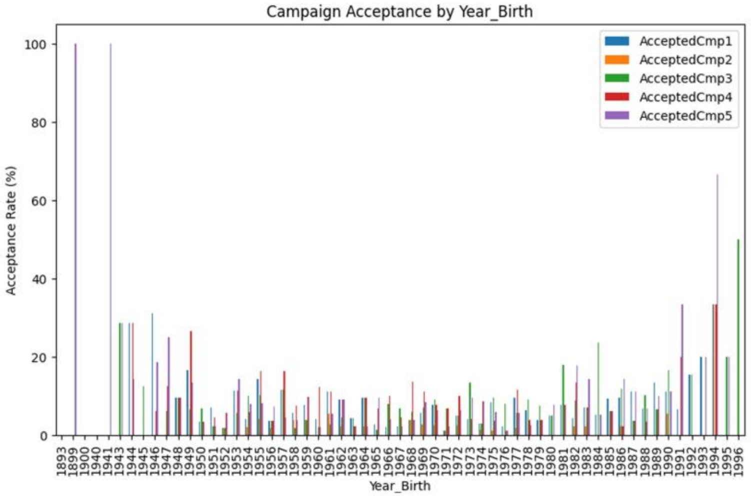
```
# 2. OVERALL ACCEPTANCE RATES FOR EACH MARKETING CAMPAIGN

IMPORT PANDAS AS PD

# CALCULATE THE ACCEPTANCE RATES FOR EACH CAMPAIGN
CAMPAIGN_COLUMNS = ['ACCEPTEDCMP1', 'ACCEPTEDCMP2', 'ACCEPTEDCMP3', 'ACCEPTEDCMP4', 'ACCEPTEDCMP5']
ACCEPTANCE_RATES = DATA[CAMPAIGN_COLUMNS].MEAN() * 100

PRINT("ACCEPTANCE RATES FOR EACH CAMPAIGN:")
PRINT(ACCEPTANCE_RATES)
```

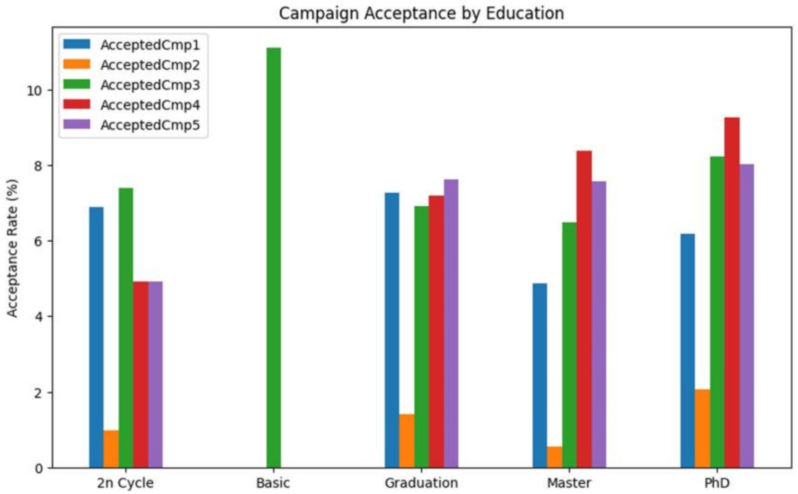
```
Acceptance Rates for Each Campaign:
AcceptedCmp1    6.428571
AcceptedCmp2    1.339286
AcceptedCmp3    7.276786
AcceptedCmp4    7.455357
AcceptedCmp5    7.276786
dtype: float64
```

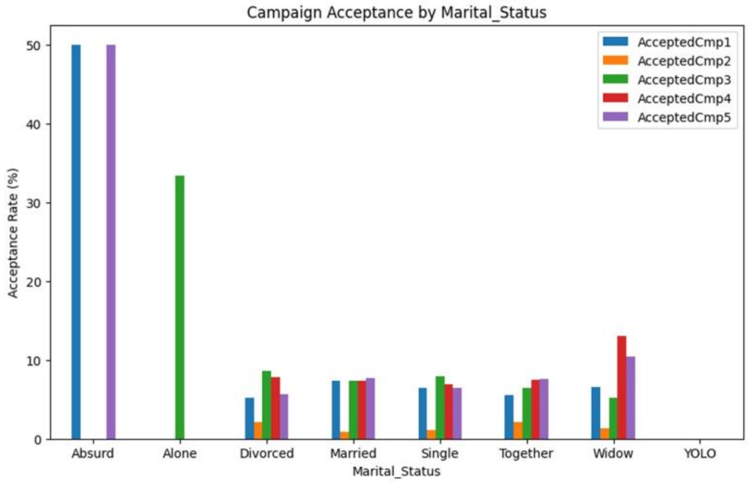
```
# 3. RELATIONSHIP BETWEEN SPENDING PATTERNS AND CAMPAIGN RESPONSES

IMPORT NUMPY AS NP

# DEFINE PRODUCT SPENDING COLUMNS
SPENDING_COLUMNS = ['MNTWINES', 'MNTFRUITS', 'MNTMEATPRODUCTS', 'MNTFISHPRODUCTS', 'MNTSWEETPRODUCTS', 'MNTGOLDPRODS']

# CALCULATE AVERAGE SPENDING FOR THOSE WHO ACCEPTED ANY CAMPAIGN VS. THOSE WHO DID NOT
DATA['ACCEPTEDANYCAMPAIGN'] = DATA[CAMPAIGN_COLUMNS].ANY(AXIS=1)
SPENDING_BY_CAMPAIGN_RESPONSE = DATA.GROUPBY('ACCEPTEDANYCAMPAIGN')[SPENDING_COLUMNS].MEAN()

PRINT("AVERAGE SPENDING BY CAMPAIGN RESPONSE:")
PRINT(SPENDING_BY_CAMPAIGN_RESPONSE)
```

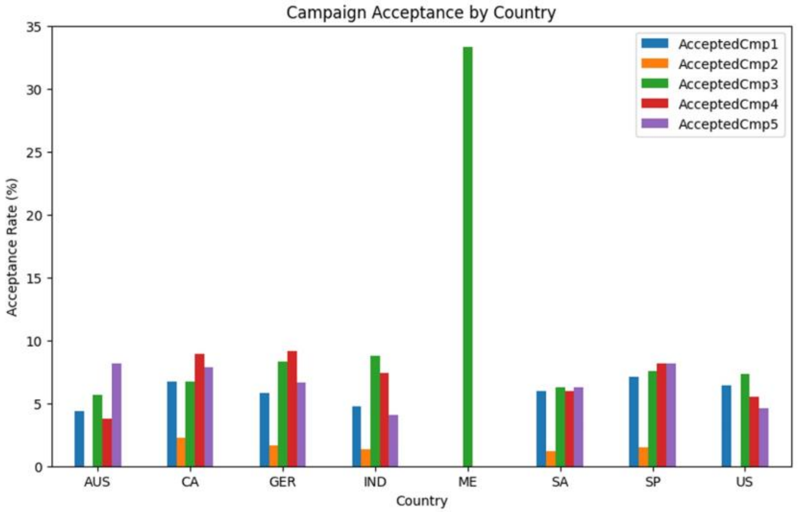
```
Average Spending by Campaign Response:
                       MntWines  MntFruits  MntMeatProducts  MntFishProducts
AcceptedAnyCampaign
False                223.855937  23.613393       134.739449        32.948790
True                 611.282937  36.622030       290.574514        55.090713

                     MntSweetProducts  MntGoldProds
AcceptedAnyCampaign
False                       23.683737     38.963984
True                        40.032397     63.434125
```

4. INFLUENCE OF DEMOGRAPHIC FACTORS ON CAMPAIGN ACCEPTANCE

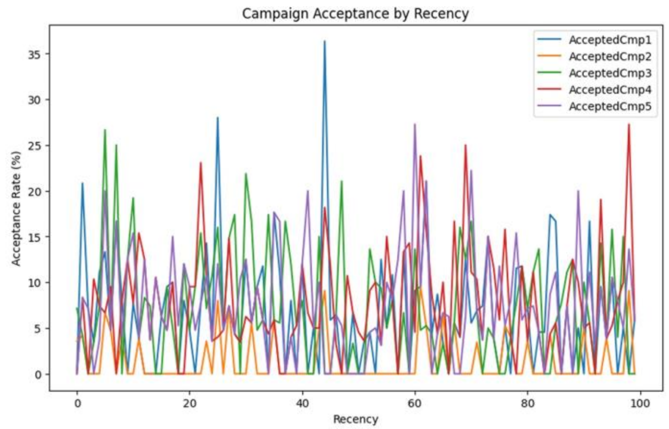


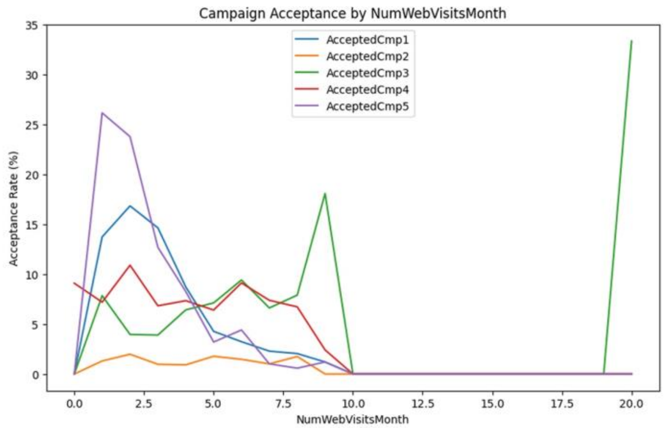


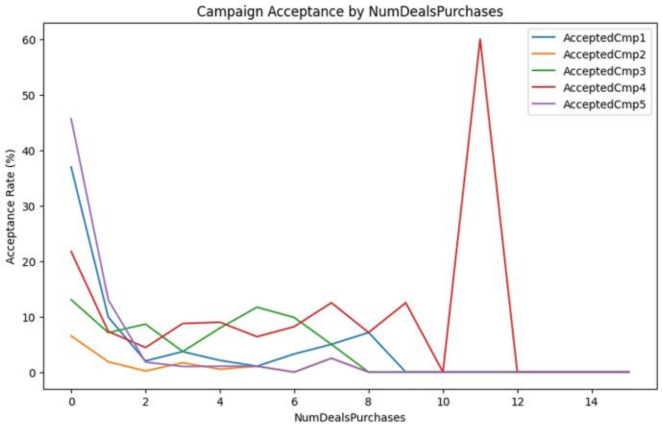


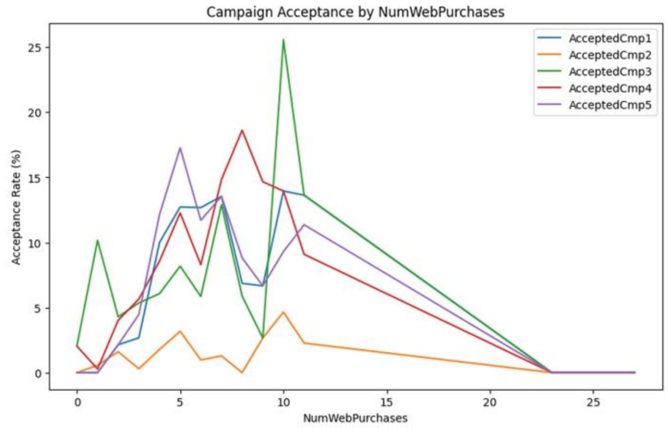


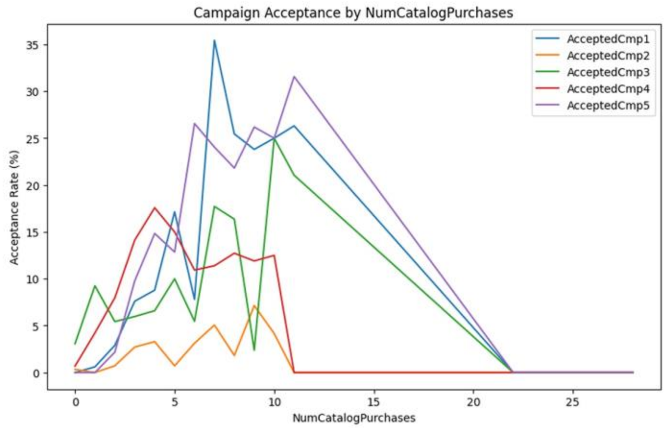
5. EFFECT OF CUSTOMER ENGAGEMENT ON CAMPAIGN RESPONSIVENESS

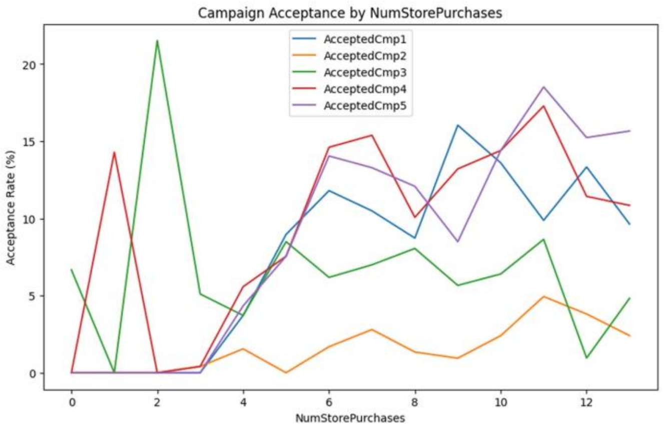






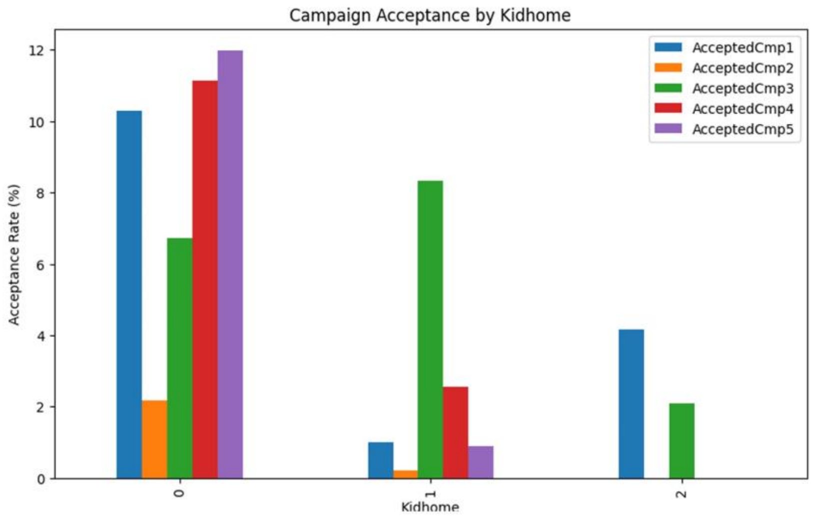


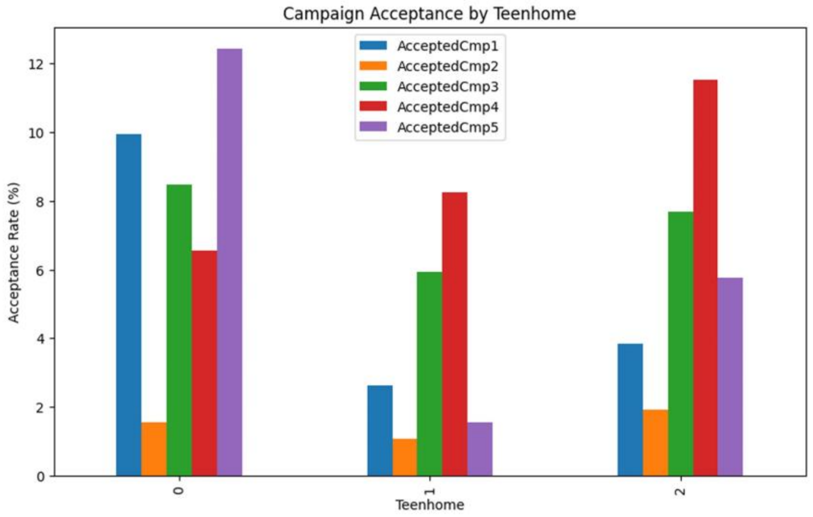




6. IMPACT OF HOUSEHOLD COMPOSITION AND INCOME ON CAMPAIGN ACCEPTANCE

```
Campaign Acceptance by Kidhome:
         AcceptedCmp1  AcceptedCmp2  AcceptedCmp3  AcceptedCmp4  AcceptedCmp5
Kidhome
0           10.286156      2.165507      6.728538     11.136891     11.987626
1            1.001112      0.222469      8.342603      2.558398      0.889878
2            4.166667      0.000000      2.083333      0.000000      0.000000

Campaign Acceptance by Teenhome:
          AcceptedCmp1  AcceptedCmp2  AcceptedCmp3  AcceptedCmp4  AcceptedCmp5
Teenhome
0             9.930915      1.554404      8.462867      6.563040     12.435233
1             2.621359      1.067961      5.922330      8.252427      1.553398
2             3.846154      1.923077      7.692308     11.538462      5.769231

Campaign Acceptance by Income:
              AcceptedCmp1  AcceptedCmp2  AcceptedCmp3  AcceptedCmp4  \
Income
$1,730.00              0.0           0.0           0.0           0.0
$10,245.00             0.0           0.0           0.0           0.0
$10,404.00             0.0           0.0           0.0           0.0
$10,979.00             0.0           0.0           0.0           0.0
$101,970.00          100.0           0.0           0.0         100.0
...                    ...           ...           ...           ...
$95,529.00             0.0           0.0           0.0           0.0
$96,547.00           100.0           0.0         100.0           0.0
$96,843.00             0.0           0.0           0.0           0.0
$96,876.00           100.0           0.0           0.0         100.0
$98,777.00             0.0           0.0           0.0         100.0

              AcceptedCmp5
Income
$1,730.00              0.0
$10,245.00             0.0
$10,404.00             0.0
$10,979.00             0.0
$101,970.00          100.0
...                    ...
$95,529.00           100.0
$96,547.00           100.0
$96,843.00           100.0
$96,876.00           100.0
$98,777.00             0.0

[1974 rows x 5 columns]
```





THANK YOU