


Step 1: Import Libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb
```

```
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.cluster import KMeans
```

Step 2: Load the Dataset


```
df = pd.read_csv('/content/new.csv')
df.head()
```



| | ID | Year_Birth | Education | Marital_Status | Income | Kidhome | Teenhome | Dt_Customer | Recency | MntWines | ... | NumWebVisitsMonth | AcceptedCmp: |
|---|------|------------|------------|----------------|---------|---------|----------|-------------|---------|----------|-----|-------------------|--------------|
| 0 | 5524 | 1957 | Graduation | Single | 58138.0 | 0 | 0 | 04-09-2012 | 58 | 635 | ... | 7 | (|
| 1 | 2174 | 1954 | Graduation | Single | 46344.0 | 1 | 1 | 08-03-2014 | 38 | 11 | ... | 5 | (|
| 2 | 4141 | 1965 | Graduation | Together | 71613.0 | 0 | 0 | 21-08-2013 | 26 | 426 | ... | 4 | (|
| 3 | 6182 | 1984 | Graduation | Together | 26646.0 | 1 | 0 | 10-02-2014 | 26 | 11 | ... | 6 | (|
| 4 | 5324 | 1981 | PhD | Married | 58293.0 | 1 | 0 | 19-01-2014 | 94 | 173 | ... | 5 | (|

5 rows × 29 columns

```
df.shape
```

 (2240, 29)

Step 3: Data Preprocessing

```
df.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

```

```
df.describe().T
```



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

| | | | | | | | | |
|------------------|--------|-----------|----------|------|-------|------|-------|------|
| Z_Revenue | 2240.0 | 11.000000 | 0.000000 | 11.0 | 11.00 | 11.0 | 11.00 | 11.0 |
| Response | 2240.0 | 0.149107 | 0.356274 | 0.0 | 0.00 | 0.0 | 0.00 | 1.0 |

```
for col in df.columns:
    temp = df[col].isnull().sum()
    if temp > 0:
        print(f'Column {col} contains {temp} null values.')
```

➞ Column Income contains 24 null values.

```
df = df.dropna()
print("Total values in the dataset after removing the null values:", len(df))
```

➞ Total values in the dataset after removing the null values: 2216

```
df.nunique()
```



0

| | |
|----------------------------|------|
| ID | 2216 |
| Year_Birth | 59 |
| Education | 5 |
| Marital_Status | 8 |
| Income | 1974 |
| Kidhome | 3 |
| Teenhome | 3 |
| Dt_Customer | 662 |
| Recency | 100 |
| MntWines | 776 |
| MntFruits | 158 |
| MntMeatProducts | 554 |
| MntFishProducts | 182 |
| MntSweetProducts | 176 |
| MntGoldProds | 212 |
| NumDealsPurchases | 15 |
| NumWebPurchases | 15 |
| NumCatalogPurchases | 14 |
| NumStorePurchases | 14 |
| NumWebVisitsMonth | 16 |
| AcceptedCmp3 | 2 |
| AcceptedCmp4 | 2 |
| AcceptedCmp5 | 2 |
| AcceptedCmp1 | 2 |

| | |
|----------------------|---|
| AcceptedCmp2 | 2 |
| Complain | 2 |
| Z_CostContact | 1 |
| Z_Revenue | 1 |
| Response | 2 |

dtype: int64

```
parts = df["Dt_Customer"].str.split("-", n=3, expand=True)
df["day"] = parts[0].astype('int')
df["month"] = parts[1].astype('int')
df["year"] = parts[2].astype('int')
```

```
df.drop(['Z_CostContact', 'Z_Revenue', 'Dt_Customer'],
        axis=1,
        inplace=True)
```

Step 4: Data Visualization and Analysis

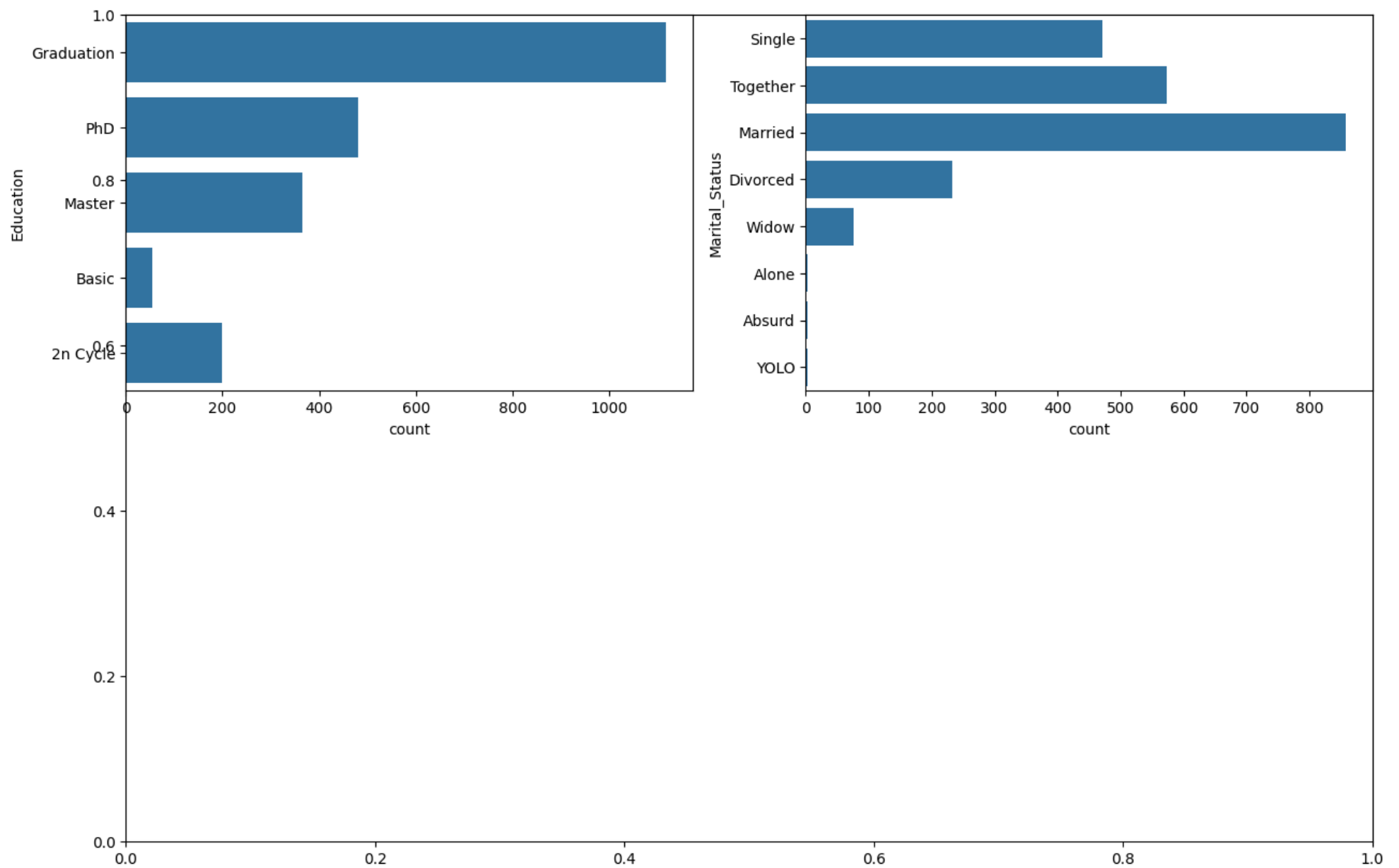
```
floats, objects = [], []
for col in df.columns:
    if df[col].dtype == object:
        objects.append(col)
    elif df[col].dtype == float:
        floats.append(col)
```

```
print(objects)
print(floats)
```

```
↔ ['Education', 'Marital_Status']
   ['Income']
```

```
plt.subplots(figsize=(15, 10))
for i, col in enumerate(objects):
    plt.subplot(2, 2, i + 1)
```

```
sb.countplot(df[col])  
plt.show()
```



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

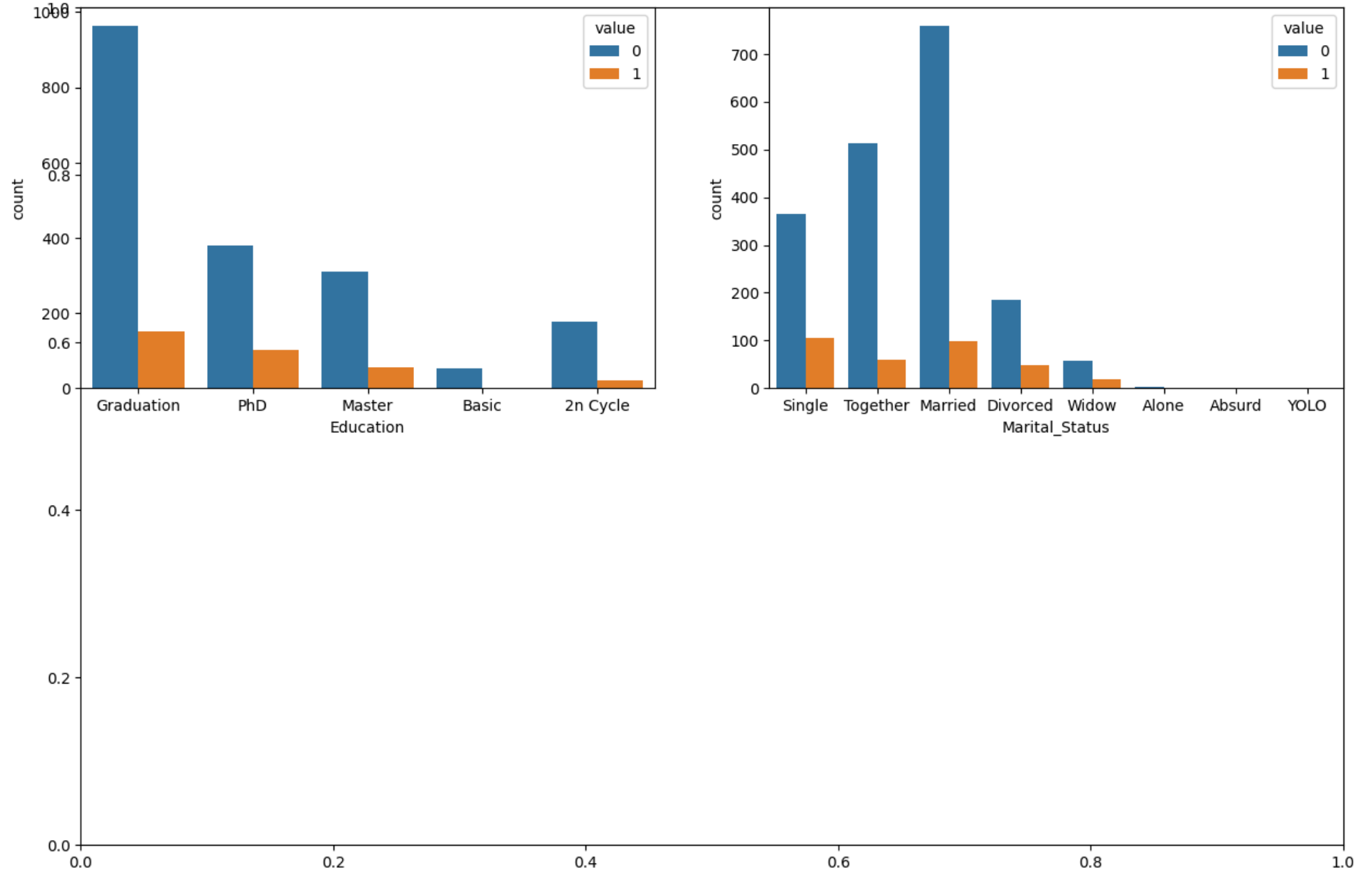


| Marital_Status | count |
|----------------|-------|
| Married | 857 |
| Together | 573 |
| Single | 471 |
| Divorced | 232 |
| Widow | 76 |
| Alone | 3 |
| Absurd | 2 |
| YOLO | 2 |

dtype: int64

```
plt.subplots(figsize=(15, 10))
for i, col in enumerate(objects):
    plt.subplot(2, 2, i + 1)

    df_melted = df.melt(id_vars=[col], value_vars=['Response'], var_name='hue')
    sb.countplot(x=col, hue='value', data=df_melted)
plt.show()
```

```
for col in df.columns:
    if df[col].dtype == object:
        le = LabelEncoder()
        df[col] = le.fit_transform(df[col])

plt.figure(figsize=(15, 15))
sb.heatmap(df.corr() > 0.8, annot=True, cbar=False)
plt.show()
```



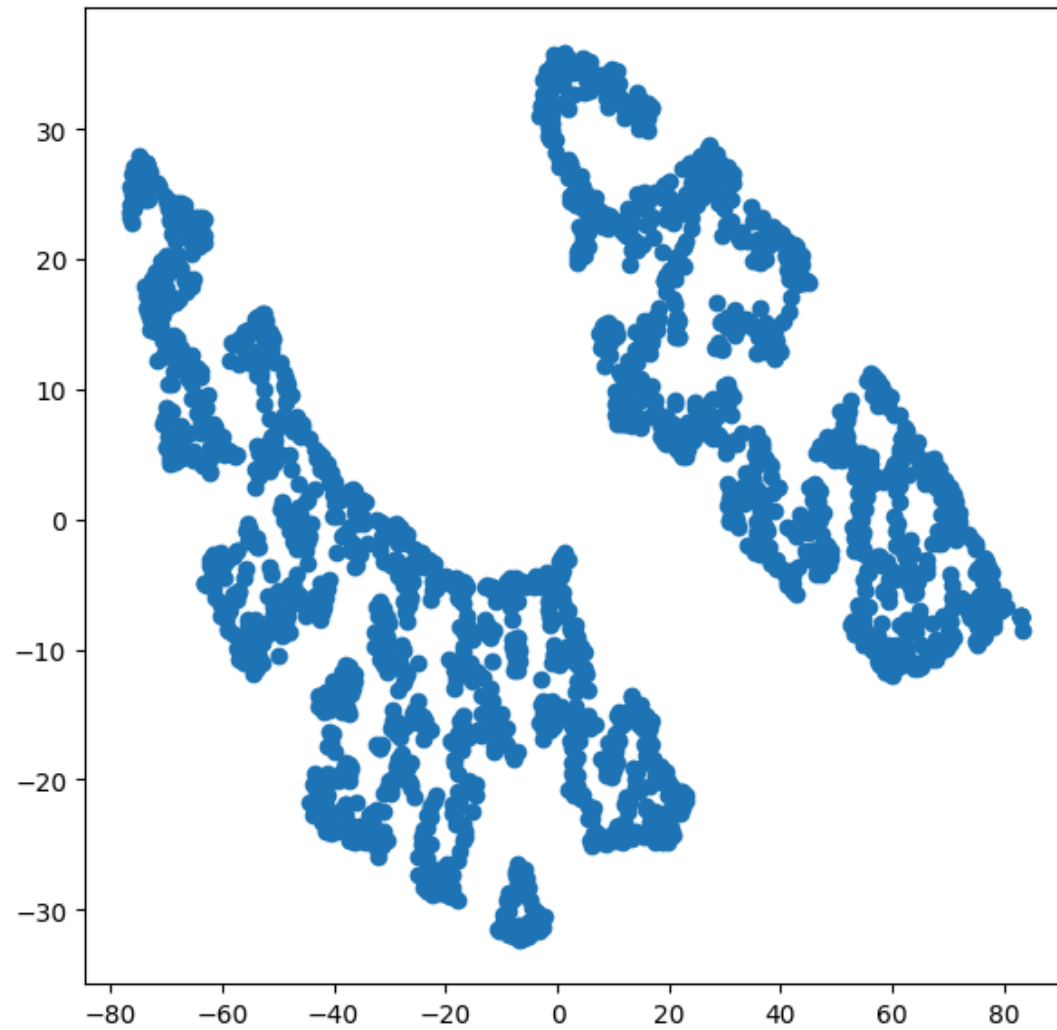
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ID | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Year_Birth | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Education | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Marital_Status | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Income | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Kidhome | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Teenhome | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Recency | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| MntWines | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| MntFruits | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| MntMeatProducts | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| MntFishProducts | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| MntSweetProducts | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| MntGoldProds | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| NumDealsPurchases | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| NumWebPurchases | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| NumCatalogPurchases | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| NumStorePurchases | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| NumWebVisitsMonth | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| AcceptedCmp3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| AcceptedCmp4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| AcceptedCmp5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| AcceptedCmp1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| AcceptedCmp2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Complain | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

[illegible]

```
scaler = StandardScaler()
data = scaler.fit_transform(df)
```

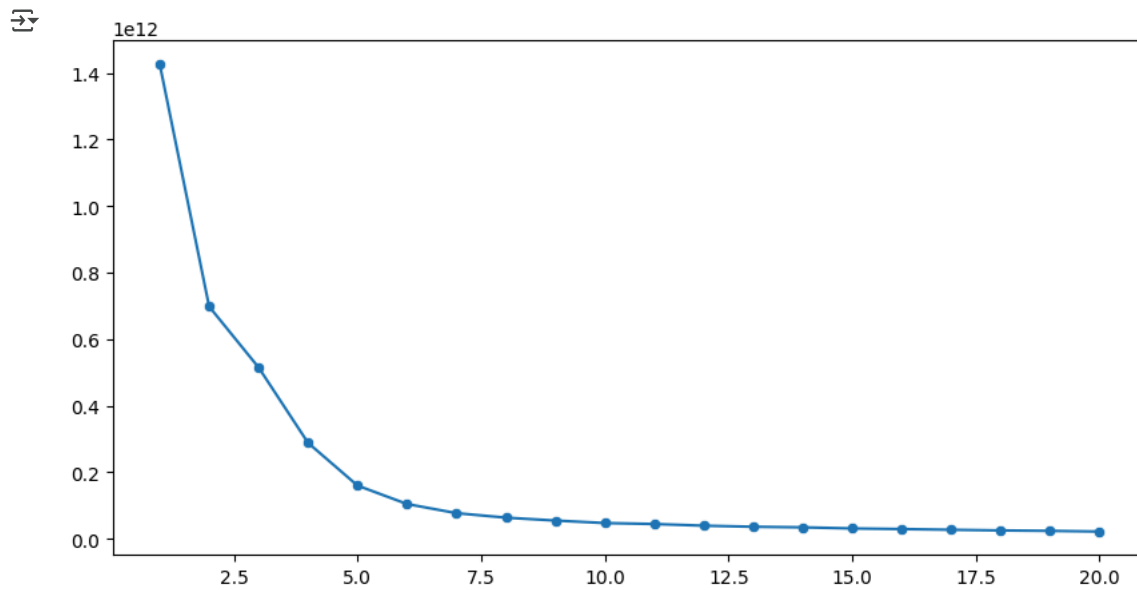
Step 5: Segmentation

```
from sklearn.manifold import TSNE
model = TSNE(n_components=2, random_state=0)
tsne_data = model.fit_transform(df)
plt.figure(figsize=(7, 7))
plt.scatter(tsne_data[:, 0], tsne_data[:, 1])
plt.show()
```



```
error = []
for n_clusters in range(1, 21):
    model = KMeans(init='k-means++',
                    n_clusters=n_clusters,
                    max_iter=500,
                    random_state=22)
    model.fit(df)
    error.append(model.inertia_)
```

```
plt.figure(figsize=(10, 5))
sb.lineplot(x=range(1, 21), y=error)
sb.scatterplot(x=range(1, 21), y=error)
plt.show()
```



```
model = KMeans(init='k-means++',
               n_clusters=5,
               max_iter=500,
               random_state=22)
segments = model.fit_predict(df)

plt.figure(figsize=(7, 7))

df_tsne = pd.DataFrame({'x': tsne_data[:, 0], 'y': tsne_data[:, 1], 'segment': segments})

sb.scatterplot(x='x', y='y', hue='segment', data=df_tsne)
plt.show()
```

