In [1]:
```
import pandas as pd
```

In [2]:
```
df = pd.read_csv("marketing_campaign.csv")
```

In [3]:
```
df
```

Out[3]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines	...	NumWeb
0	5524	1957	Graduation	Single	58138.0	0	0	04-09-2012	58	635	...	
1	2174	1954	Graduation	Single	46344.0	1	1	08-03-2014	38	11	...	
2	4141	1965	Graduation	Together	71613.0	0	0	21-08-2013	26	426	...	
3	6182	1984	Graduation	Together	26646.0	1	0	10-02-2014	26	11	...	
4	5324	1981	PhD	Married	58293.0	1	0	19-01-2014	94	173	...	
...	...	...	...	...	...	...	...	...	...	...	...	
2235	10870	1967	Graduation	Married	61223.0	0	1	13-06-2013	46	709	...	
2236	4001	1946	PhD	Together	64014.0	2	1	10-06-2014	56	406	...	
2237	7270	1981	Graduation	Divorced	56981.0	0	0	25-01-2014	91	908	...	
2238	8235	1956	Master	Together	69245.0	0	1	24-01-2014	8	428	...	
2239	9405	1954	PhD	Married	52869.0	1	1	15-10-2012	40	84	...	

2240 rows × 29 columns

In [4]:
```
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
```

In [5]:
```
df.describe()
```

Out[5]:

	ID	Year_Birth	Income	Kidhome	Teenhome	Recency	MntWines	MntFruits	MntMeatProduc
count	2240.000000	2240.000000	2216.000000	2240.000000	2240.000000	2240.000000	2240.000000	2240.000000	2240.0000
mean	5592.159821	1968.805804	52247.251354	0.444196	0.506250	49.109375	303.935714	26.302232	166.9500
std	3246.662198	11.984069	25173.076661	0.538398	0.544538	28.962453	336.597393	39.773434	225.7153
min	0.000000	1893.000000	1730.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0000
25%	2828.250000	1959.000000	35303.000000	0.000000	0.000000	24.000000	23.750000	1.000000	16.0000
50%	5458.500000	1970.000000	51381.500000	0.000000	0.000000	49.000000	173.500000	8.000000	67.0000
75%	8427.750000	1977.000000	68522.000000	1.000000	1.000000	74.000000	504.250000	33.000000	232.0000
max	11191.000000	1996.000000	666666.000000	2.000000	2.000000	99.000000	1493.000000	199.000000	1725.0000

8 rows × 26 columns

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
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
```

In [7]:
```
df.shape[0]
```

Out[7]:
```
2240
```

In [8]:
```
df.columns
```

Out[8]:
```
Index(['ID', 'Year_Birth', 'Education', 'Marital_Status', 'Income', 'Kidhome',
       'Teenhome', 'Dt_Customer', 'Recency', 'MntWines', 'MntFruits',
       'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts',
       'MntGoldProds', 'NumDealsPurchases', 'NumWebPurchases',
       'NumCatalogPurchases', 'NumStorePurchases', 'NumWebVisitsMonth',
       'AcceptedCmp3', 'AcceptedCmp4', 'AcceptedCmp5', 'AcceptedCmp1',
       'AcceptedCmp2', 'Complain', 'Z_CostContact', 'Z_Revenue', 'Response'],
      dtype='object')
```

In [9]:
```
type(df)
```

Out[9]:
```
pandas.core.frame.DataFrame
```

In [10]:
```
len(df.columns)
```

Out[10]:
```
29
```

In [11]:
```
df.drop_duplicates()
```

Out[11]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines	...	NumWeb
0	5524	1957	Graduation	Single	58138.0	0	0	04-09-2012	58	635	...	
1	2174	1954	Graduation	Single	46344.0	1	1	08-03-2014	38	11	...	
2	4141	1965	Graduation	Together	71613.0	0	0	21-08-2013	26	426	...	
3	6182	1984	Graduation	Together	26646.0	1	0	10-02-2014	26	11	...	
4	5324	1981	PhD	Married	58293.0	1	0	19-01-2014	94	173	...	
...	...	...	...	...	...	...	...	...	...	...	...	
2235	10870	1967	Graduation	Married	61223.0	0	1	13-06-2013	46	709	...	
2236	4001	1946	PhD	Together	64014.0	2	1	10-06-2014	56	406	...	
2237	7270	1981	Graduation	Divorced	56981.0	0	0	25-01-2014	91	908	...	
2238	8235	1956	Master	Together	69245.0	0	1	24-01-2014	8	428	...	
2239	9405	1954	PhD	Married	52869.0	1	1	15-10-2012	40	84	...	

2240 rows × 29 columns

In [12]:
```
df['Marital_Status'].unique()
```

Out[12]:
```
array(['Single', 'Together', 'Married', 'Divorced', 'Widow', 'Alone',
       'Absurd', 'YOLO'], dtype=object)
```

In [13]:
```
# Define mapping for standardization
marital_map = {
    'Single': 'Single',
    'Alone': 'Single',
    'YOLO': 'Single',
    'Absurd': 'Single',

    'Married': 'Married',
    'Together': 'Married',

    'Divorced': 'Divorced',
    'Widow': 'Widowed'
}
```

In [14]:
```
marital_map
```

Out[14]:
```
{'Single': 'Single',
 'Alone': 'Single',
 'YOLO': 'Single',
 'Absurd': 'Single',
 'Married': 'Married',
 'Together': 'Married',
 'Divorced': 'Divorced',
 'Widow': 'Widowed'}
```

In [15]:
```
df['Marital_Status'] = df['Marital_Status'].map(marital_map)
```

In [16]:
```
df['Marital_Status'].unique()
```

Out[16]:
```
array(['Single', 'Married', 'Divorced', 'Widowed'], dtype=object)
```

In [17]:
```
df.columns = df.columns.str.lower()
```

In [18]:
```
df.columns
```

Out[18]:
```
Index(['id', 'year_birth', 'education', 'marital_status', 'income', 'kidhome',
       'teenhome', 'dt_customer', 'recency', 'mntwines', 'mntfruits',
       'mntmeatproducts', 'mntfishproducts', 'mntsweetproducts',
       'mntgoldprods', 'numdealspurchases', 'numwebpurchases',
       'numcatalogpurchases', 'numstorepurchases', 'numwebvisitsmonth',
       'acceptedcmp3', 'acceptedcmp4', 'acceptedcmp5', 'acceptedcmp1',
       'acceptedcmp2', 'complain', 'z_costcontact', 'z_revenue', 'response'],
      dtype='object')
```

In [19]:
```
# Top 15 income
top_15_income = df.sort_values(by='income' , ascending=False).head(15)
top_15_income
```

Out[19]:

	id	year_birth	education	marital_status	income	kidhome	teenhome	dt_customer	recency	mntwines	...	numwebvi
2233	9432	1977	Graduation	Married	666666.0	1	0	02-06-2013	23	9	...	
617	1503	1976	PhD	Married	162397.0	1	1	03-06-2013	31	85	...	
687	1501	1982	PhD	Married	160803.0	0	0	04-08-2012	21	55	...	
1300	5336	1971	Master	Married	157733.0	1	0	04-06-2013	37	39	...	
164	8475	1973	PhD	Married	157243.0	0	1	01-03-2014	98	20	...	
1653	4931	1977	Graduation	Married	157146.0	0	0	29-04-2013	13	1	...	
2132	11181	1949	PhD	Married	156924.0	0	0	29-08-2013	85	2	...	
655	5555	1975	Graduation	Divorced	153924.0	0	0	07-02-2014	81	1	...	
1898	4619	1945	PhD	Single	113734.0	0	0	28-05-2014	9	6	...	
646	4611	1970	Graduation	Married	105471.0	0	0	21-01-2013	36	1009	...	
252	10089	1974	Graduation	Divorced	102692.0	0	0	05-04-2013	5	168	...	
203	2798	1977	PhD	Married	102160.0	0	0	02-11-2012	54	763	...	
124	7215	1983	Graduation	Single	101970.0	0	0	12-03-2013	69	722	...	
1113	7451	1960	Master	Single	98777.0	0	0	17-02-2014	23	1000	...	
650	4248	1960	Master	Single	98777.0	0	0	17-02-2014	23	1000	...	

15 rows × 29 columns

In [20]:
```
import seaborn as sns
import matplotlib as mat
import matplotlib.pyplot as plt
%matplotlib inline

sns.set_style('darkgrid')
mat.rcParams['font.size'] = 14
mat.rcParams['figure.figsize'] = (12,6)
mat.rcParams['figure.facecolor'] = '#00000000'
```

In [21]:
```
dff = df.head(10)
dff
```

Out[21]:

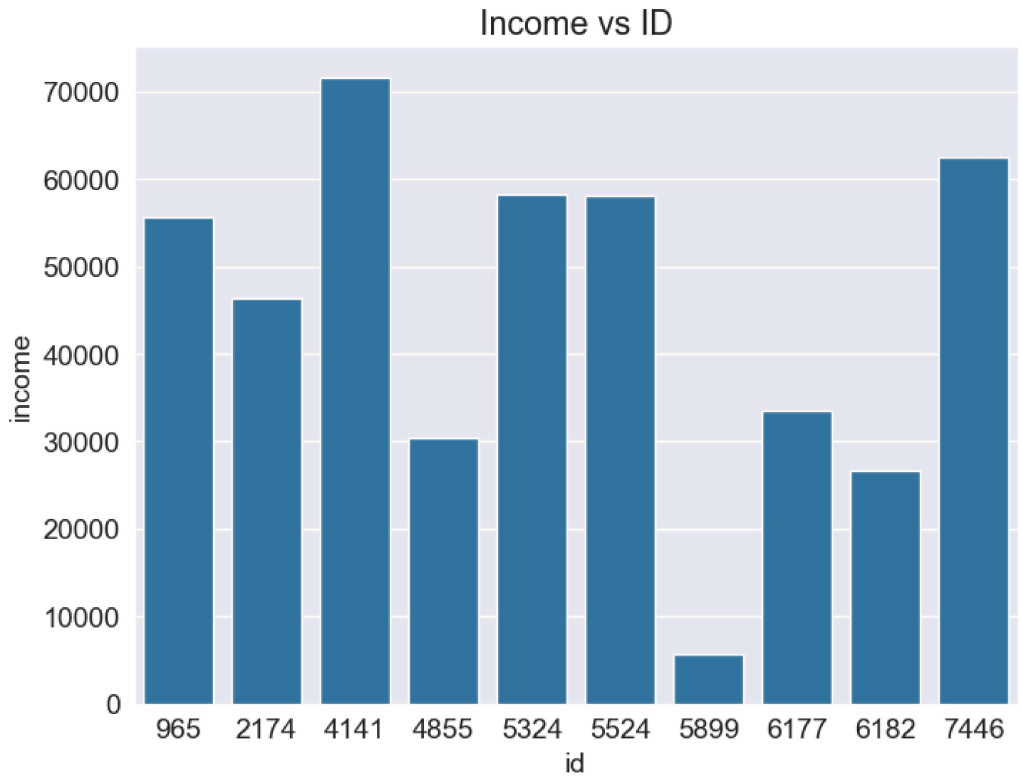
	id	year_birth	education	marital_status	income	kidhome	teenhome	dt_customer	recency	mntwines	...	numwebvisitsmo
0	5524	1957	Graduation	Single	58138.0	0	0	04-09-2012	58	635	...	
1	2174	1954	Graduation	Single	46344.0	1	1	08-03-2014	38	11	...	
2	4141	1965	Graduation	Married	71613.0	0	0	21-08-2013	26	426	...	
3	6182	1984	Graduation	Married	26646.0	1	0	10-02-2014	26	11	...	
4	5324	1981	PhD	Married	58293.0	1	0	19-01-2014	94	173	...	
5	7446	1967	Master	Married	62513.0	0	1	09-09-2013	16	520	...	
6	965	1971	Graduation	Divorced	55635.0	0	1	13-11-2012	34	235	...	
7	6177	1985	PhD	Married	33454.0	1	0	08-05-2013	32	76	...	
8	4855	1974	PhD	Married	30351.0	1	0	06-06-2013	19	14	...	
9	5899	1950	PhD	Married	5648.0	1	1	13-03-2014	68	28	...	

10 rows × 29 columns

In [23]:
```
import seaborn as sns
import matplotlib.pyplot as plt

mat.rcParams['figure.figsize'] = (8, 6)
plt.title('Income vs ID')

# Plot 'income' on x-axis and 'id' on y-axis
sns.barplot(x='id', y='income', data=dff)
plt.show()
```


Income vs ID
70000
60000
50000
40000
30000
20000
10000
0
income
965
2174
4141
4855
5324
5524
5899
6177
6182
7446
id

In [ ]:

In [ ]: